SAS/STAT® 15.1 User's Guide

The PRINQUAL Procedure

This document is an individual chapter from *SAS/STAT® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/STAT® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 15.1 User's Guide

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to http://support.sas.com/thirdpartylicenses.

Chapter 96
The PRINQUAL Procedure

Contents

Overview: PRINQUAL Procedure **7934**
Getting Started: PRINQUAL Procedure **7936**
Syntax: PRINQUAL Procedure **7940**
PROC PRINQUAL Statement 7940
BY Statement 7948
FREQ Statement 7949
ID Statement 7949
TRANSFORM Statement 7949
WEIGHT Statement 7957
Details: PRINQUAL Procedure **7957**
The Three Methods of Variable Transformation 7957
Understanding How PROC PRINQUAL Works 7958
Splines 7962
Missing Values 7963
Controlling the Number of Iterations 7963
Performing a Principal Component Analysis of Transformed Data 7964
Using the MAC Method 7964
Output Data Set 7965
Avoiding Constant Transformations 7968
Constant Variables 7968
Character OPSCORE Variables 7968
REITERATE Option Usage 7968
Passive Observations 7969
Computational Resources 7970
Displayed Output 7971
ODS Table Names 7972
ODS Graphics 7972
Examples: PRINQUAL Procedure **7973**
Example 96.1: Multidimensional Preference Analysis of Automobile Data 7973
Example 96.2: Principal Components of Basketball Rankings 7981
References **7989**

Overview: PRINQUAL Procedure

The PRINQUAL procedure performs principal component analysis (PCA) of qualitative, quantitative, or mixed data. PROC PRINQUAL is based on the work of Kruskal and Shepard (1974); Young, Takane, and de Leeuw (1978); Young (1981); Winsberg and Ramsay (1983). PROC PRINQUAL finds linear and nonlinear transformations of variables, using the method of alternating least squares, that optimize properties of the transformed variables' correlation or covariance matrix. Nonoptimal transformations such as logarithm and rank are also available. You can use ODS Graphics to display the results. You can use PROC PRINQUAL to do the following:

- fit metric and nonmetric principal component analyses
- perform metric and nonmetric multidimensional preference (MDPREF) analyses (Carroll 1972)
- transform data prior to their use in other analyses
- reduce the number of variables for subsequent use in regression analyses, cluster analyses, and other analyses
- detect nonlinear relationships

PROC PRINQUAL provides three methods, each of which seeks to optimize a different property of the transformed variables' covariance or correlation matrix. These methods are as follows:

- maximum total variance, or MTV
- minimum generalized variance, or MGV
- maximum average correlation, or MAC

The MTV method is based on a PCA model, and it is the most commonly used method. All three methods attempt to find transformations that decrease the rank of the covariance matrix computed from the transformed variables. Transforming the variables to maximize the total variance accounted for by a few linear combinations locates the observations in a space with a dimensionality that approximates the stated number of linear combinations as much as possible, given the transformation constraints. Transforming the variables to minimize their generalized variance or maximize the average correlations also reduces the dimensionality, but without a stated target for the final dimensionality. See the section "The Three Methods of Variable Transformation" on page 7957 for more information about all three methods.

The data can contain variables measured on nominal, ordinal, interval, and ratio scales of measurement (Siegel 1956). Any mix is allowed with all methods. PROC PRINQUAL can do the following:

- transform nominal variables by optimally scoring the categories (Fisher 1938)
- transform ordinal variables monotonically by scoring the ordered categories so that order is weakly preserved (adjacent categories can be merged) and the covariance matrix is optimized. You can undo ties optimally or leave them tied (Kruskal 1964). You can also transform ordinal variables to ranks.
- transform interval and ratio scale of measurement variables linearly, or transform them nonlinearly with spline transformations (De Boor 1978; Van Rijckevorsel 1982) or monotone spline transformations (Winsberg and Ramsay 1983). In addition, nonoptimal transformations for logarithm, rank, exponential, power, logit, and inverse trigonometric sine are available.
- estimate missing data without constraint, with category constraints (missing values within the same group get the same value), and with order constraints (missing value estimates in adjacent groups can be tied to preserve a specified ordering). See Gifi (1990) and Young (1981).

The transformed qualitative (nominal and ordinal) variables can be thought of as being quantified by the analysis, with the quantification done in the context set by the algorithm. The data are quantified so that the proportion of variance accounted for by a stated number of principal components is locally maximized, the generalized variance of the variables is locally minimized, or the average of the correlations is locally maximized.

The PROC PRINQUAL iterations produce a set of transformed variables. Each variable's new scoring satisfies a set of constraints based on the original scoring of the variable and the specified transformation type. First, all variables are required to satisfy standardization constraints; that is, all variables have a fixed mean and variance. The other constraints include linear constraints, weak order constraints, category constraints, and smoothness constraints. The new set of scores is selected from the sets of possible scorings that do not violate the constraints so that the method criterion is locally optimized.

Getting Started: PRINQUAL Procedure

PROC PRINQUAL can be used to fit a principal component model with nonlinear transformations of the variables and graphically display the results. This example finds monotonic transformations of ratings of automobiles.

```
title 'Ratings for Automobiles Manufactured in 1980';

data cars;
   input Origin $ 1-8 Make $ 10-19 Model $ 21-36
         (MPG Reliability Acceleration Braking Handling Ride
          Visibility Comfort Quiet Cargo) (1.);
   datalines;
GMC      Buick      Century          3334444544
GMC      Buick      Electra          2434453555
GMC      Buick      Lesabre          2354353545
GMC      Buick      Regal            3244443424

   ... more lines ...

GMC      Pontiac    Sunbird          3134533234
;

ods graphics on;

proc prinqual data=cars plots=all maxiter=100;
   transform monotone(mpg -- cargo);
   id model;
run;
```

The PROC PRINQUAL statement names the input data set Cars. The ODS GRAPHICS statement, along with the PLOTS=ALL option, requests all graphical displays. The PLOTS=ALL option implies the MDPREF option and requests a PCA plot with the scores (automobiles) represented as points and the structure (variables) represented as vectors. By default, the vector lengths are increased by a factor of 2.5 to produce a better graphical display. If instead you were to specify MDPREF=1, you would get the actual vectors, and they would all be short and would end near the origin where there are a lot of points. It is often the case that increasing the vector lengths by a factor of 2 or 3 makes a better graphical display, so by default the vector lengths are increased by a factor of 2.5. Up to 100 iterations are requested with the MAXITER= option. All of the numeric variable are specified with a MONOTONE transformation, so their original values, 1 to 5, are optimally rescored to maximize fit to a two-component model while preserving the original order. The Model variable provides the labels for the row points in the plot.

The iteration history table is shown in Figure 96.1. The monotonic transformations allow the PCA to account for 5% more variance in two principal components than the ordinary PCA model applied to the untransformed data.

Figure 96.1 Automobile Ratings Iteration History

Ratings for Automobiles Manufactured in 1980

The PRINQUAL Procedure

PRINQUAL MTV Algorithm Iteration History

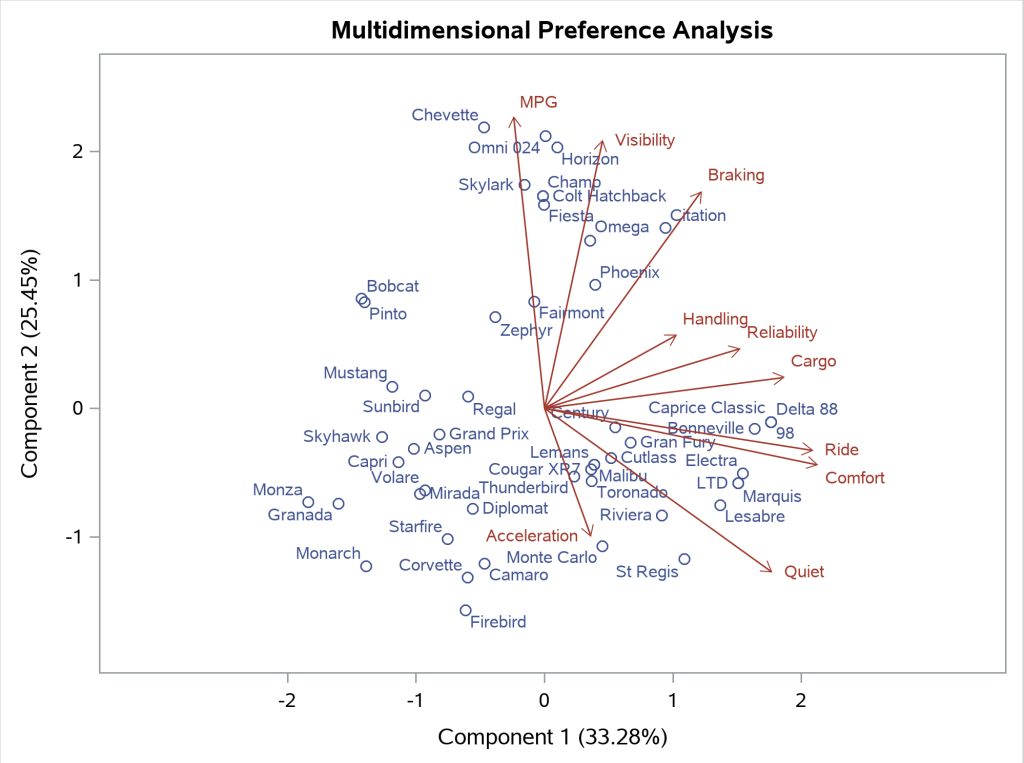
Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.18087	1.24219	0.53742		
2	0.06916	0.77503	0.57244	0.03502	
3	0.04653	0.38237	0.57978	0.00734	
4	0.03387	0.18682	0.58300	0.00321	
5	0.02661	0.13506	0.58484	0.00185	
6	0.01730	0.09213	0.58600	0.00115	
7	0.00969	0.07107	0.58660	0.00061	
8	0.00705	0.04798	0.58685	0.00025	
9	0.00544	0.03482	0.58699	0.00014	
10	0.00442	0.02641	0.58708	0.00009	
11	0.00363	0.02062	0.58714	0.00006	
12	0.00298	0.01643	0.58717	0.00004	
13	0.00245	0.01325	0.58720	0.00002	
14	0.00201	0.01077	0.58721	0.00002	
15	0.00165	0.00880	0.58723	0.00001	
16	0.00136	0.00721	0.58723	0.00001	
17	0.00112	0.00591	0.58724	0.00001	
18	0.00092	0.00485	0.58724	0.00000	
19	0.00075	0.00399	0.58724	0.00000	
20	0.00062	0.00328	0.58725	0.00000	
21	0.00051	0.00269	0.58725	0.00000	
22	0.00042	0.00221	0.58725	0.00000	
23	0.00035	0.00182	0.58725	0.00000	
24	0.00028	0.00149	0.58725	0.00000	
25	0.00023	0.00123	0.58725	0.00000	
26	0.00019	0.00101	0.58725	0.00000	
27	0.00016	0.00083	0.58725	0.00000	
28	0.00013	0.00068	0.58725	0.00000	
29	0.00011	0.00056	0.58725	0.00000	
30	0.00009	0.00046	0.58725	0.00000	
31	0.00007	0.00038	0.58725	0.00000	
32	0.00006	0.00031	0.58725	0.00000	
33	0.00005	0.00025	0.58725	0.00000	
34	0.00004	0.00021	0.58725	0.00000	
35	0.00003	0.00017	0.58725	0.00000	
36	0.00003	0.00014	0.58725	0.00000	
37	0.00002	0.00012	0.58725	0.00000	
38	0.00002	0.00010	0.58725	0.00000	
39	0.00001	0.00008	0.58725	0.00000	
40	0.00001	0.00006	0.58725	0.00000	
41	0.00001	0.00005	0.58725	0.00000	
42	0.00001	0.00004	0.58725	0.00000	Converged

Figure 96.1 *continued*

Algorithm converged.

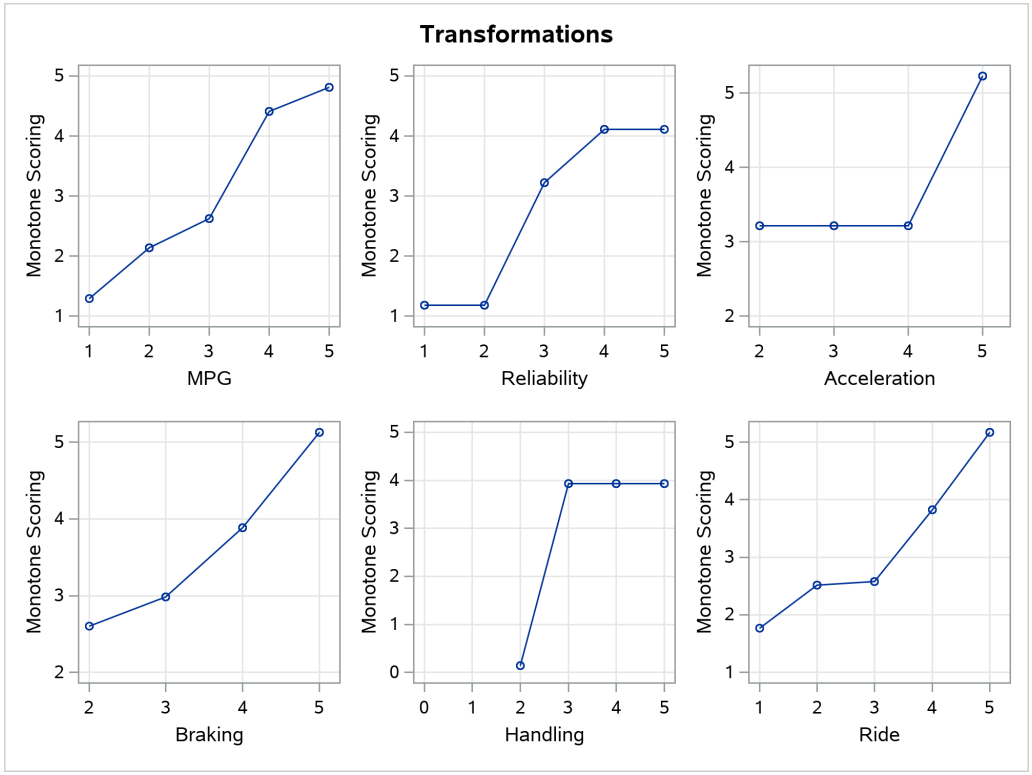
The PCA biplot in Figure 96.2 shows the transformed automobile ratings projected into the two-dimensional plane of the analysis. The automobiles on the left tend to be smaller than the autos on the right, and the autos at the top tend to be cheaper than the autos at the bottom. The vectors can help you interpret the plot of the scores. Longer vectors show the variables that better fit the two-dimensional model. A larger component of them is in the plane of the plot. In contrast, shorter vectors show the variables that do not fit the two-dimensional model as well. They tend to be located less in the plot and more away from the plot; hence their projection into the plot is shorter. To envision this, lay a pencil on your desk directly under a light, and slowly rotate it up to form a 90-degree angle with your desk. As you do so, the shadow or projection of the pencil onto your desk will get progressively shorter. The results show, for example, that the Chevette would be expected to do well on gas mileage but not well on quiet and acceleration. In contrast, the Corvette and the Firebird have the opposite pattern.

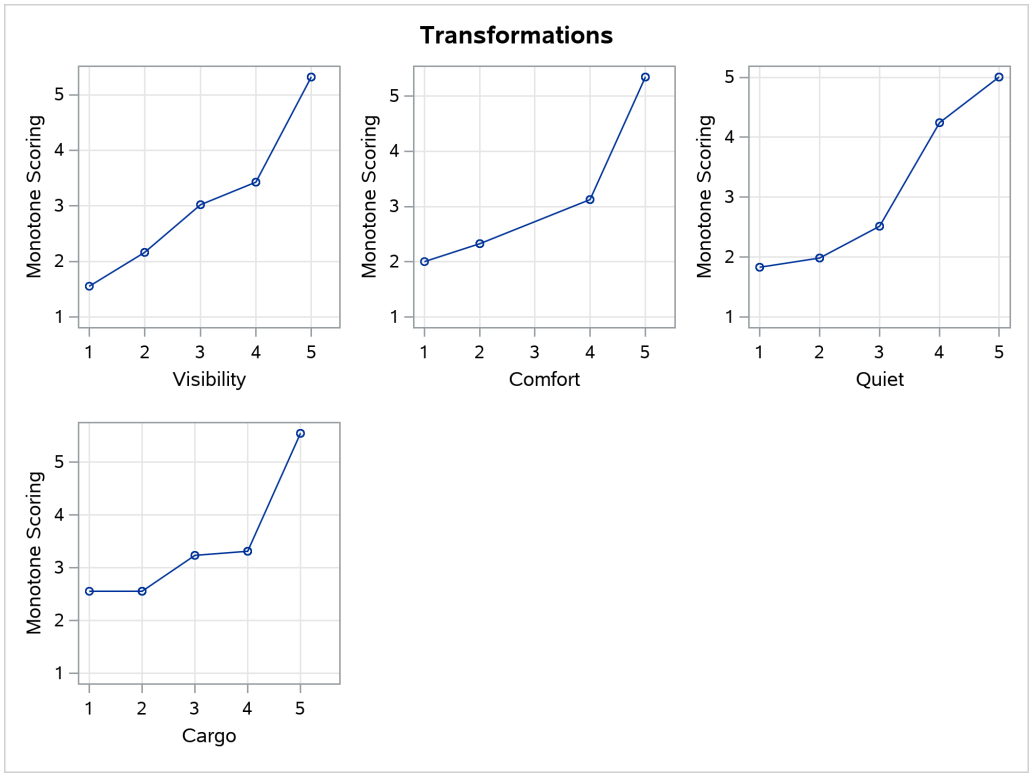
Figure 96.2 Automobile Ratings PCA Biplot



There are many patterns shown in the transformations in Figure 96.3. The transformation of Braking, for example, is not very different from the original scoring. The optimal scoring for other variables, such as Acceleration and Handling, is binary. Automobiles are differentiated by high versus everything else or low versus everything else.

Figure 96.3 Automobile Ratings Transformations





Syntax: PRINQUAL Procedure

The following statements are available in the PRINQUAL procedure:

```
PROC PRINQUAL < options > ;
   TRANSFORM   transform(variables < / t-options >)
               < transform(variables < / t-options >)... > ;
   ID variables ;
   FREQ variable ;
   WEIGHT variable ;
   BY variables ;
```

To use PROC PRINQUAL, you need the PROC PRINQUAL and TRANSFORM statements. You can abbreviate all *options* and *t-options* to their first three letters. This is a special feature of PROC PRINQUAL and is not generally true of other SAS/STAT procedures.

The rest of this section provides detailed syntax information about each of the preceding statements, beginning with the PROC PRINQUAL statement. The remaining statements are described in alphabetical order.

PROC PRINQUAL Statement

PROC PRINQUAL < *options* > ;

The PROC PRINQUAL statement invokes the PRINQUAL procedure. Optionally, this statement identifies an input data set, creates an output data set, specifies the algorithm and other computational details, and controls displayed output. Table 96.1 summarizes the options available in the PROC PRINQUAL statement.

Table 96.1 Summary of PROC PRINQUAL Statement Options

Option	Description
Input Data Set Options	
DATA=	Specifies input SAS data set
Output Data Set Details	
APPROXIMATIONS	Outputs approximations to transformed variables
APREFIX=	Specifies prefix for approximation variables
CORRELATIONS	Outputs correlations and component structure matrix
MDPREF=	Specifies a multidimensional preference analysis
OUT=	Specifies output data set
PREFIX=	Specifies prefix for principal component scores
REPLACE	Replaces raw data with transformed data
SCORES	Outputs principal component scores
STANDARD	Standardizes principal component scores
TPREFIX=	Specifies prefix for transformed variables
TSTANDARD=	Specifies transformation standardization

Table 96.1 *continued*

Option	Description
Method and Iterations	
CCONVERGE=	Specifies minimum criterion change
CHANGE=	Specifies number of first iteration to be displayed
CONVERGE=	Specifies minimum data change
COVARIANCE	Analyzes covariances
DUMMY	Initializes using dummy variables
INITITER=	Specifies number of MAC initialization iterations
MAXITER=	Specifies maximum number of iterations
METHOD=	Specifies iterative algorithm
NOCHECK	Suppresses numerical error checking
N=	Specifies number of principal components
REFRESH=	Specifies number of MGV models before refreshing
REITERATE	Restarts iterations
SINGULAR=	Specifies singularity criterion
TYPE=	Specifies input observation type
Missing Data Handling	
MONOTONE=	Includes monotone special missing values
NOMISS	Excludes observations with missing values
UNTIE=	Unties special missing values
Control Displayed Output	
NOPRINT	Suppresses displayed output
PLOTS=	Specifies ODS Graphics details

The following list describes these options in alphabetical order.

APREFIX=*name*

APR=*name*

specifies a prefix for naming the approximation variables. By default, APREFIX=A. Specifying the APREFIX= option also implies the APPROXIMATIONS option.

APPROXIMATIONS

APPROX

APP

includes principal component approximations to the transformed variables (Eckart and Young 1936) in the output data set. Variable names are constructed from the value of the APREFIX= option and the input variable names. If you specify the APREFIX= option, then approximations are automatically included. If you specify the APPROXIMATIONS option and not the APREFIX= option, then the APPROXIMATIONS option uses the default, APREFIX=A, to construct the variable names.

CCONVERGE=*n*

CCO=*n*

specifies the minimum change in the criterion being optimized that is required to continue iterating. By default, CCONVERGE=0.0. The CCONVERGE= option is ignored for METHOD=MAC. For the MGV method, specify CCONVERGE=–2 to ensure data convergence.

CHANGE=*n*

CHA=*n*

specifies the number of the first iteration to be displayed in the iteration history table. The default is CHANGE=1. When you specify a larger value for *n*, the first $n-1$ iterations are not displayed, thus speeding up the analysis. The CHANGE= option is most useful with the MGV method, which is much slower than the other methods.

CONVERGE=*n*

CON=*n*

specifies the minimum average absolute change in standardized variable scores that is required to continue iterating. By default, CONVERGE=0.00001. Average change is computed over only those variables that can be transformed by the iterations—that is, all LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, and SSPLINE variables and nonoptimal transformation variables with missing values. For more information, see the section "Optimal Transformations" on page 7952.

COVARIANCE

COV

computes the principal components from the covariance matrix. The variables are always centered to mean zero. If you do not specify the COVARIANCE option, the variables are also standardized to variance one, which means the analysis is based on the correlation matrix.

CORRELATIONS

COR

includes correlations and the component structure matrix in the output data set. By default, this information is not included.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. The data set must be an ordinary SAS data set; it cannot be a TYPE=CORR or TYPE=COV data set. If you omit the DATA= option, PROC PRINQUAL uses the most recently created SAS data set.

DUMMY

DUM

expands variables specified for OPSCORE optimal transformations to dummy variables for the initialization (Tenenhaus and Vachette 1977). By default, the initial values of OPSCORE variables are the actual data values. The dummy variable nominal initialization requires considerable time and memory, so it might not be possible to use the DUMMY option with large data sets. No separate report of the initialization is produced. Initialization results are incorporated into the first iteration displayed in the iteration history table. For details, see the section "Optimal Transformations" on page 7952.

INITITER=*n*

INI=*n*

specifies the number of MAC iterations required to initialize the data before starting MTV or MGV iterations. By default, INITITER=0. The INITITER= option is ignored if METHOD=MAC.

MAXITER=*n*

MAX=*n*

specifies the maximum number of iterations. By default, MAXITER=30.

MDPREF<=*n***>**

MDP<=*n***>**

specifies a multidimensional preference analysis by implying the STANDARD, SCORES, and CORRELATIONS options. This option also suppresses warnings when there are more variables than observations.

When ODS Graphics is enabled, an MDPREF plot is produced with points for each row and vectors for each column. Often, the vectors are short, and a better graphical display is produced when the vectors are stretched. The absolute lengths of each vector can optionally be changed by specifying MDPREF=*n*. Then the vector coordinates are all multiplied by *n*. Usually, *n* will be a value such as 2, 2.5, or 3. The default is 2.5. Specify MDPREF=1 to see the vectors without any stretching. The relative lengths of the different vectors is important and interpretable, and these are preserved by the stretching.

METHOD=MAC | MGV | MTV

MET=MAC | MGV | MTV

specifies the optimization method. By default, METHOD=MTV. Values of the METHOD= option are MTV, for maximum total variance; MGV, for minimum generalized variance; and MAC, for maximum average correlation. You can use the MAC method when all variables are positively correlated or when no monotonicity constraints are placed on any transformations. See the section "The Three Methods of Variable Transformation" on page 7957 for more information.

MONOTONE=*two-letters*

MON=*two-letters*

specifies the first and last special missing value in the list of those special missing values to be estimated using within-variable order and category constraints. By default, there are no order constraints on missing value estimates. The *two-letters* value must consist of two letters in alphabetical order. For example, MONOTONE=DF means that the estimate of .D must be less than or equal to the estimate of .E, which must be less than or equal to the estimate of .F; no order constraints are placed on estimates of ._, .A through .C, and .G through .Z. For details, see the sections "Missing Values" on page 7963 and "Optimal Scaling" on page 10357 in Chapter 123, "The TRANSREG Procedure."

N=*n*

specifies the number of principal components to be computed. By default, N=2.

NOCHECK

NOC

turns off computationally intensive numerical error checking for the MGV method. If you do not specify the NOCHECK option, the procedure computes R square from the squared length of the predicted values vector and compares this value to the R square computed from the error sum of squares that is a byproduct of the sweep algorithm (Goodnight 1978). If the two values of R square differ by more than the square root of the value of the SINGULAR= option, a warning is displayed, the value of the REFRESH= option is halved, and the model is refit after refreshing. Specifying the NOCHECK option slightly speeds up the algorithm. Note that other less computationally intensive error checking is always performed.

NOMISS

NOM

excludes all observations with missing values from the analysis, but does not exclude them from the OUT= data set. If you omit the NOMISS option, PROC PRINQUAL simultaneously computes the optimal transformations of the nonmissing values and estimates the missing values that minimize squared error.

Casewise deletion of observations with missing values occurs when you specify the NOMISS option, when there are missing values in IDENTITY variables, when there are weights less than or equal to 0, or when there are frequencies less than 1. Excluded observations are output with a blank value for the _TYPE_ variable, and they have a weight of 0. They do not contribute to the analysis but are scored and transformed as *supplementary* or passive observations. See the sections "Passive Observations" on page 7969 and "Missing Values" on page 7963 for more information about excluded observations and missing data.

NOPRINT

NOP

suppresses the display of all output. This option disables the Output Delivery System (ODS), including ODS Graphics, for the duration of the procedure. For more information, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

specifies an output SAS data set that contains results of the analysis. If you omit the OUT= option, PROC PRINQUAL still creates an output data set and names it by using the DATA*n* convention. If you want to create a SAS data set in a permanent library, you must specify a two-level name. For more information about permanent libraries and SAS data sets, see *SAS Language Reference: Concepts*. You can use the REPLACE, APPROXIMATIONS, SCORES, and CORRELATIONS options to control what information is included in the output data set. For details, see the section "Output Data Set" on page 7965.

PLOTS < (*global-plot-options*) > < = *plot-request* < (*options*) > >

PLOTS < (*global-plot-options*) > < = (*plot-request* < (*options*) > < ... *plot-request* < (*options*) > >) >

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses from around the plot request. Here are some examples:

```
plots=none
plots=transformation
plots(unpack)=transformation
```

ODS Graphics must be enabled before plots can be requested. For example:

```
ods graphics on;

proc prinqual plots=all;
   transformation spline(x1-x10);
run;

ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 623 in Chapter 21, "Statistical Graphics Using ODS."

If ODS Graphics is enabled, but do not specify the PLOTS= option, then PROC PRINQUAL produces an MDPREF plot when the MDPREF option is specified.

The global plot options include the following:

FLIP

FLI

flips or interchanges the X-axis and Y-axis dimensions for MDPREF plots. The FLIP option can be specified either as a global plot option (for example, PLOTS(FLIP)) or with the MDPREF option (for example, PLOTS=MDPREF(FLIP)).

INTERPOLATE

INT

uses observations that are excluded from the analysis for interpolation in the fit and transformation plots. By default, observations with zero weight are excluded from all plots. These include observations with a zero, negative, or missing weight or frequency and observations excluded due to missing and invalid values. You can specify PLOTS(INTERPOLATE)=(*plot-requests*) to include some of these observations in the plots. You can use this option, for example, with sparse data sets to show smoother functions over the range of the data (see the section "The PLOTS(INTERPOLATE) Option" on page 10396 in Chapter 123, "The TRANSREG Procedure").

ONLY

ONL

suppresses the default plots. Only plots specifically requested are displayed.

UNPACKPANEL

UNPACK

UNP

suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel.

The plot requests include the following:

ALL

produces all appropriate plots.

TRANSFORMATION

TRA

TRANSFORMATION(UNPACK)

TRA(UNP)

plots the variable transformations. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to display each plot in a separate panel.

MDPREF

MDP

plots multidimensional preference analysis results. The MDPREF plot can also be requested by specifying the MDPREF option in the PROC PRINQUAL statement outside the PLOTS= option.

NONE

suppresses all plots.

PREFIX=*name*

PRE=*name*

specifies a prefix for naming the principal components. By default, PREFIX=Prin. As a result, the principal component default names are Prin1, Prin2,..., Prin*n*.

REFRESH=*n*

REF=*n*

specifies the number of variables to scale in the MGV method before computing a new inverse. By default, REFRESH=5. PROC PRINQUAL uses the REFRESH= option in the sweep algorithm of the MGV method. Large values for the REFRESH= option make the method run faster but with more numerical error. Small values make the method run more slowly but with more numerical accuracy.

REITERATE

REI

enables PROC PRINQUAL to use previous transformations as starting points. The REITERATE option affects only variables that are iteratively transformed (specified as LINEAR, SPLINE, MSPLINE, SSPLINE, UNTIE, OPSCORE, and MONOTONE). For iterative transformations, the REITERATE option requests a search in the input data set for a variable that consists of the value of the TPREFIX= option followed by the original variable name. If such a variable is found, it is used to provide the initial values for the first iteration. The final transformation is a member of the transformation family defined by the original variable, not the transformation family defined by the initialization variable. See the section "REITERATE Option Usage" on page 7968.

REPLACE
REP

replaces the original data with the transformed data in the output data set. The names of the transformed variables in the output data set correspond to the names of the original variables in the input data set. If you do not specify the REPLACE option, both original variables and transformed variables (with names constructed from the TPREFIX= option and the original variable names) are included in the output data set.

SCORES
SCO

includes principal component scores in the output data set. By default, scores are not included.

SINGULAR=*n*
SIN=*n*

specifies the largest value within rounding error of zero. By default, SINGULAR=1E–8. PROC PRINQUAL uses the value of the SINGULAR= option for checking $(1 - \mathrm{R}^2)$ when constructing full-rank matrices of predictor variables, checking denominators before dividing, and so on.

STANDARD
STD

standardizes the principal component scores in the output data set to mean zero and variance one instead of the default mean zero and variance equal to the corresponding eigenvalue. See the SCORES option.

TPREFIX=*name*
TPR=*name*

specifies a prefix for naming the transformed variables. By default, TPREFIX=T. The TPREFIX= option is ignored if you specify the REPLACE option.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z
TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables in the OUT= data set. By default, TSTANDARD=ORIGINAL. When you specify the TSTANDARD= option in the PROC PRINQUAL statement, it the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization just for selected variables.

CENTER centers the output variables to mean zero, but the variances are the same as the variances of the input variables.

NOMISS sets the means and variances of the transformed variables in the OUT= data set, computed over all output values that correspond to nonmissing values in the input data set, to the means and variances computed from the nonmissing observations of the original variables. The TSTANDARD=NOMISS specification is useful with missing data. When a variable is linearly transformed, the final variable contains the original nonmissing values and the missing value estimates. In other words, the nonmissing values are unchanged. If your data have no missing values, TSTANDARD=NOMISS and TSTANDARD=ORIGINAL produce the same results.

ORIGINAL sets the means and variances of the transformed variables to the means and variances of the original variables. This is the default.

Z standardizes the variables to mean zero, variance one.

For nonoptimal variable transformations, the means and variances of the original variables are actually the means and variances of the nonlinearly transformed variables, unless you specify the ORIGINAL nonoptimal *t-option* in the TRANSFORM statement. For example, if a variable X with no missing values is specified as LOG, then, by default, the final transformation of X is simply LOG(X), not LOG(X) standardized to the mean of X and variance of X.

TYPE='*text* '|*name*

TYP='*text* '|*name*

specifies the valid value for the _TYPE_ variable in the input data set. If PROC PRINQUAL finds an input _TYPE_ variable, it uses only observations with a _TYPE_ value that matches the TYPE= value. This enables a PROC PRINQUAL OUT= data set containing correlations to be used as input to PROC PRINQUAL without requiring a WHERE statement to exclude the correlations. If a _TYPE_ variable is not in the data set, all observations are used. The default is TYPE='SCORE', so if you do not specify the TYPE= option, only observations with _TYPE_ = 'SCORE' are used.

PROC PRINQUAL displays a note when it reads observations with blank values of _TYPE_, but it does not automatically exclude those observations. Data sets created by the TRANSREG and PRINQUAL procedures have blank _TYPE_ values for those observations that were excluded from the analysis due to nonpositive weights, nonpositive frequencies, or missing data. When these observations are read again, they are excluded for the same reason that they were excluded from their original analysis, not because their _TYPE_ value is blank.

UNTIE=*two-letters*

UNT=*two-letters*

specifies the first and last special missing values in the list of those special missing values that are to be estimated with within-variable order constraints but no category constraints. The *two-letters* value must consist of two letters in alphabetical order. By default, there are category constraints but no order constraints on special missing value estimates. For details, see the section "Missing Values" on page 7963. Also, see the section "Optimal Scaling" on page 10357 in Chapter 123, "The TRANSREG Procedure."

BY Statement

BY *variables* **;**

You can specify a BY statement in PROC PRINQUAL to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement in the PRINQUAL procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged

in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, list the variable's name in a FREQ statement. PROC PRINQUAL then treats the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ statement variable is greater than or equal to 1.

ID Statement

ID *variables* **;**

The ID statement includes additional character or numeric variables in the output data set. The variables must be contained in the input data set.

TRANSFORM Statement

TRANSFORM *transform*(*variables* < / *t-options* >)< *transform*(*variables* < / *t-options* >)... > **;**

The TRANSFORM statement lists the variables to be analyzed (*variables*) and specifies the transformation (*transform*) to apply to each variable listed. You must specify a transformation for each variable list in the TRANSFORM statement. The variables are variables in the data set. The *t-options* are transformation options that provide details for the transformation; these depend on the *transform* chosen. The *t-options* are listed after a slash in the parentheses that enclose the variables.

For example, the following statements find a quadratic polynomial transformation of all variables in the data set:

```
proc prinqual;
   transform spline(_all_ / degree=2);
run;
```

Or, if N1 through N10 are nominal variables and M1 through M10 are ordinal variables, you can use the following statements:

```
proc prinqual;
   transform opscore(N1-N10) monotone(M1-M10);
run;
```

The following sections describe the transformations available (specified with *transform*) and the options available for some of the transformations (specified with *t-options*).

Families of Transformations

There are three types of transformation families: nonoptimal, optimal, and other. The families are described as follows:

Nonoptimal transformations	preprocess the specified variables, replacing each one with a single new nonoptimal, nonlinear transformation.
Optimal transformations	replace the specified variables with new, iteratively derived optimal transformation variables that fit the specified model better than the original variable (except in contrived cases where the transformation fits the model exactly as well as the original variable).
Other transformations	are the IDENTITY and SSPLINE transformations. These do not fit into either of the preceding categories.

Table 96.2 summarizes the transformations in each family.

Table 96.2 Transformation Families

Transformation	Description
Nonoptimal Transformations	
ARSIN	Inverse trigonometric sine
EXP	Exponential
LOG	Logarithm
LOGIT	Logit
POWER	Raises variables to specified power
RANK	Transforms to ranks
Optimal Transformations	
LINEAR	Linear
MONOTONE	Monotonic, ties preserved
MSPLINE	Monotonic B-spline
OPSCORE	Optimal scoring
SPLINE	B-spline
UNTIE	Monotonic, ties not preserved
Other Transformations	
IDENTITY	Identity, no transformation
SSPLINE	Iterative smoothing spline

The *transform* is followed by a variable (or list of variables) enclosed in parentheses. Optionally, depending on the *transform*, the parentheses can also contain *t-options*, which follow the variables and a slash. For example, the following statement computes the LOG transformation of X and Y:

```
transform log(X Y);
```

A more complex example follows:

```
transform spline(Y / nknots=2) log(X1 X2 X3);
```

The preceding statement uses the SPLINE transformation of the variable Y and the LOG transformation of the variables X1, X2, and X3. In addition, it uses the NKNOTS= option with the SPLINE transformation and specifies two knots.

The rest of this section provides syntax details for members of the three families of transformations. The *t-options* are discussed in the section "Transformation Options (t-options)" on page 7953.

Nonoptimal Transformations

Nonoptimal transformations are computed before the iterative algorithm begins. Nonoptimal transformations create a single new transformed variable that replaces the original variable. The new variable is not transformed by the subsequent iterative algorithms (except for a possible linear transformation and missing value estimation).

The following list provides syntax and details for nonoptimal variable transformations.

ARSIN

ARS

finds an inverse trigonometric sine transformation. Variables specified in the ARSIN *transform* must be numeric and in the interval ($-1.0 \leq x \leq 1.0$), and they are typically continuous.

EXP

exponentiates variables (x is transformed to a^x). To specify the value of a, use the PARAMETER= *t-option*. By default, a is the mathematical constant $e = 2.718\ldots$. Variables specified with the EXP *transform* must be numeric, and they are typically continuous.

LOG

transforms variables to logarithms (x is transformed to $\log_a(x)$). To specify the base of the logarithm, use the PARAMETER= *t-option*. The default is a natural logarithm with base $e = 2.718\ldots$. Variables specified with the LOG *transform* must be numeric and positive, and they are typically continuous.

LOGIT

finds a logit transformation on the variables. The logit of x is $\log(x/(1-x))$. Unlike other transformations, LOGIT does not have a three-letter abbreviation. Variables specified with the LOGIT *transform* must be numeric and in the interval ($0.0 < x < 1.0$), and they are typically continuous.

POWER

POW

raises variables to a specified power (x is transformed to x^a). You must specify the power parameter a by specifying the PARAMETER= *t-option* following the variables.

```
power(variable / parameter=number)
```

You can use POWER for squaring variables (PARAMETER=2), reciprocal transformations (PARAMETER=–1), square roots (PARAMETER=0.5), and so on. Variables specified with the POWER *transform* must be numeric, and they are typically continuous.

RANK

RAN

transforms variables to ranks. Ranks are averaged within ties. The smallest input value is assigned the smallest rank. Variables specified with the RANK *transform* must be numeric.

Optimal Transformations

Optimal transformations are iteratively derived. Missing values for these types of variables can be optimally estimated (see the section "Missing Values" on page 7963). See the sections "OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations" on page 10357 and "SPLINE and MSPLINE Transformations" on page 10359 in Chapter 123, "The TRANSREG Procedure," for more information about the optimal transformations.

The following list provides syntax and details for optimal transformations.

LINEAR

LIN

finds an optimal linear transformation of each variable. For variables with no missing values, the transformed variable is the same as the original variable. For variables with missing values, the transformed nonmissing values have a different scale and origin than the original values. Variables specified with the LINEAR *transform* must be numeric.

MONOTONE

MON

finds a monotonic transformation of each variable, with the restriction that ties are preserved. The Kruskal (1964) secondary least squares monotonic transformation is used. This transformation weakly preserves order and category membership (ties). Variables specified with the MONOTONE *transform* must be numeric, and they are typically discrete.

MSPLINE

MSP

finds a monotonically increasing B-spline transformation with monotonic coefficients (De Boor 1978; De Leeuw 1986) of each variable. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY= *t-options* with MSPLINE. By default, PROC PRINQUAL uses a quadratic spline. Variables specified with the MSPLINE *transform* must be numeric, and they are typically continuous.

OPSCORE

OPS

finds an optimal scoring of each variable. The OPSCORE transformation assigns scores to each class (level) of the variable. The Fisher (1938) optimal scoring method is used. Variables specified with the OPSCORE *transform* can be either character or numeric; numeric variables should be discrete.

SPLINE

SPL

finds a B-spline transformation (De Boor 1978) of each variable. By default, PROC PRINQUAL uses a cubic polynomial transformation. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with SPLINE. Variables specified with the SPLINE *transform* must be numeric, and they are typically continuous.

UNTIE

UNT

finds a monotonic transformation of each variable without the restriction that ties are preserved. PROC PRINQUAL uses the Kruskal (1964) primary least squares monotonic transformation method. This transformation weakly preserves order but not category membership (it might untie some previously tied values). Variables specified with the UNTIE *transform* must be numeric, and they are typically discrete.

Other Transformations

IDENTITY

IDE

specifies variables that are not changed by the iterations. The IDENTITY transformation is used for variables when no transformation and no missing data estimation are desired. However, the REFLECT, ADDITIVE, TSTANDARD=Z, and TSTANDARD=CENTER options can linearly transform all variables, including IDENTITY variables, after the iterations. Observations with missing values in IDENTITY variables are excluded from the analysis, and no optimal scores are computed for missing values in IDENTITY variables. Variables specified with the IDENTITY *transform* must be numeric.

SSPLINE

SSP

finds an iterative smoothing spline transformation of each variable. The SSPLINE transformation does not generally minimize squared error. You can specify the smoothing parameter with either the SM= *t-option* or the PARAMETER= *t-option*. The default smoothing parameter is SM=0. Variables specified with the SSPLINE *transform* must be numeric, and they are typically continuous.

Transformation Options (t-options)

If you use a nonoptimal, optimal, or other transformation, you can use *t-options*, which specify additional details of the transformation. The *t-options* are specified within the parentheses that enclose variables and are listed after a slash. For example:

```
proc prinqual;
   transform spline(X Y / nknots=3);
run;
```

The preceding statements find an optimal variable transformation (SPLINE) of the variables X and Y and use a *t-option* to specify the number of knots (NKNOTS=). The following is a more complex example:

```
proc prinqual;
   transform spline(Y / nknots=3) spline(X1 X2 / nknots=6);
run;
```

These statements use the SPLINE transformation for all three variables and use *t-options* as well; the NKNOTS= option specifies the number of knots for the spline.

The following sections discuss the *t-options* available for nonoptimal, optimal, and other transformations.

Table 96.3 summarizes the *t-options*.

Table 96.3 Transformation Options

Option	Description
Nonoptimal Transformation	
ORIGINAL	Uses original mean and variance
Parameter Specification	
PARAMETER=	Specifies miscellaneous parameters
SM	Specifies smoothing parameter
Spline	
DEGREE=	Specifies the degree of the spline
EVENLY	Spaces the knots evenly
KNOTS=	Specifies the interior knots or break points
NKNOTS=	Creates *n* knots
Other t-options	
NAME=	Renames variables
REFLECT	Reflects the variable around the mean
TSTANDARD=	Specifies transformation standardization

Nonoptimal Transformation t-options

ORIGINAL

ORI

matches the variable's final mean and variance to the mean and variance of the original variable. By default, the mean and variance are based on the transformed values. The ORIGINAL *t-option* is available for all of the nonoptimal transformations.

Parameter t-options

PARAMETER=*number*

PAR=*number*

specifies the transformation parameter. The PARAMETER= *t-option* is available for the EXP, LOG, POWER, SMOOTH, and SSPLINE transformations. For EXP, the parameter is the value to be exponentiated; for LOG, the parameter is the base value; and for POWER, the parameter is the power. For SMOOTH and SSPLINE, the parameter is the raw smoothing parameter. (See the SM= option for an alternative way to specify the smoothing parameter.) The default for the PARAMETER= *t-option* for the LOG and EXP transformations is $e = 2.718\ldots$. The default parameter for SSPLINE is computed from SM=0. For the POWER transformation, you must specify the PARAMETER= *t-option*; there is no default.

SM=*n*

specifies a smoothing parameter in the range 0 to 100, just like PROC GPLOT uses. For example, SM=50 in PROC PRINQUAL is equivalent to I=SM50 on the SYMBOL statement with PROC GPLOT. You can specify the SM= *t-option* only with the SSPLINE transformation. The smoothness of the function increases as the value of the smoothing parameter increases. By default, SM=0.

Spline t-options

The following *t-options* are available with the SPLINE and MSPLINE optimal transformations.

DEGREE=*n*

DEG=*n*

specifies the degree of the B-spline transformation. The degree must be a nonnegative integer. The defaults are DEGREE=3 for SPLINE variables and DEGREE=2 for MSPLINE variables.

The polynomial degree should be a small integer, usually 0, 1, 2, or 3. Larger values are rarely useful. If you have any doubt as to what degree to specify, use the default.

EVENLY<=*n***>**

EVE<=*n***>**

is used with the NKNOTS= *t-option* to space the knots evenly. The differences between adjacent knots are constant. If you specify NKNOTS=*k*, *k* knots are created at

$$\text{minimum} + i((\text{maximum} - \text{minimum})/(k+1))$$

for $i = 1, \ldots, k$. For example, if you specify

```
spline(X / knots=2 evenly)
```

and the variable X has a minimum of 4 and a maximum of 10, then the two interior knots are 6 and 8. Without the EVENLY *t-option*, the NKNOTS= *t-option* places knots at percentiles, so the knots are not evenly spaced.

By default for the SPLINE and MSPLINE transformations, the smaller exterior knots are all the same and just a little smaller than the minimum. Similarly, by default, the larger exterior knots are all the same and just a little larger than the maximum. However, if you specify EVENLY=*n*, then the *n* exterior knots are evenly spaced as well. The number of exterior knots must be greater than or equal to the degree. You can specify values larger than the degree when you want to interpolate slightly beyond the range or your data. The exterior knots must be less than the minimum or greater than the maximum, and hence the knots across all sets are not precisely equally spaced. For example, with data ranging from 0 to 10, and with EVENLY=3 and NKNOTS=4, the first exterior knots are –4.000000000001, –2.000000000001, and –0.000000000001, the interior knots are 2, 4, 6, and 8, and the second exterior knots are 10.000000000001, 12.000000000001, and 14.000000000001.

KNOTS=*number-list* | *n* **TO** *m* **BY** *p*

KNO=*number-list* | *n* **TO** *m* **BY** *p*

specifies the interior knots or break points. By default, there are no knots. The first time you specify a value in the knot list, it indicates a discontinuity in the *n*th (from DEGREE=*n*) derivative of the transformation function at the value of the knot. The second mention of a value indicates a discontinuity in the $(n-1)$ derivative of the transformation function at the value of the knot. Knots can be repeated

any number of times to decrease the smoothness at the break points, but the values in the knot list can never decrease.

You cannot use the KNOTS= *t-option* with the NKNOTS= *t-option*. You should keep the number of knots small. (See the section "Specifying the Number of Knots" on page 10360 in Chapter 123, "The TRANSREG Procedure.")

NKNOTS=*n*

NKN=*n*

creates *n* knots, the first at the $100/(n+1)$th percentile, the second at the $200/(n+1)$th percentile, and so on. Knots are always placed at data values; there is no interpolation. For example, if NKNOTS=3, knots are placed at the 25th percentile, the median, and the 75th percentile. By default, NKNOTS=0. The NKNOTS= *t-option* must be ≥ 0.

You cannot use the NKNOTS= *t-option* with the KNOTS= *t-option*. You should keep the number of knots small. (See the section "Specifying the Number of Knots" on page 10360 in Chapter 123, "The TRANSREG Procedure.")

Other t-options

The following *t-options* are available for all transformations.

NAME=(*variable-list***)**

NAM=(*variable-list***)**

renames variables as they are used in the TRANSFORM statement. This option allows a variable to be used more than once. For example, if the variable X is a character variable, then the following step stores both the original character variable X and a numeric variable XC that contains category numbers in the output data set.

```
proc prinqual data=A n=1 out=B;
   transform linear(Y) opscore(X / name=(XC));
   id X;
run;
```

REFLECT

REF

reflects the transformation

$$y = -(y - \bar{y}) + \bar{y}$$

after the iterations are completed and before the final standardization and results calculations.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z

TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables in the OUT= data set. By default, TSTANDARD=ORIGINAL. When the TSTANDARD= option is specified in the PROC PRINQUAL statement, it specifies the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization only for selected variables.

WEIGHT Statement

WEIGHT *variable* **;**

When you use a WEIGHT statement, a weighted residual sum of squares is minimized. The WEIGHT statement has no effect on degrees of freedom or number of observations, but the weights affect most other calculations. The observation is used in the analysis only if the value of the WEIGHT statement variable is greater than 0.

Details: PRINQUAL Procedure

The Three Methods of Variable Transformation

The three methods of variable transformation provided by PROC PRINQUAL are discussed in the following sections.

The Maximum Total Variance (MTV) Method

The MTV method (Young, Takane, and de Leeuw 1978) is based on the principal component model, and it attempts to maximize the sum of the first *r* eigenvalues of the covariance matrix. This method transforms variables to be (in a least squares sense) as similar to linear combinations of *r* principal component score variables as possible, where *r* can be much smaller than the number of variables. This maximizes the total variance of the first *r* components (the trace of the covariance matrix of the first *r* principal components). See *SAS Technical Report R-108*.

On each iteration, the MTV algorithm alternates classical principal component analysis (Hotelling 1933) with optimal scaling (Young 1981). When all variables are ordinal preference ratings, this corresponds to MDPREF analysis (Carroll 1972). You can request the dummy variable initialization method suggested by Tenenhaus and Vachette (1977), who independently propose the same iterative algorithm for nominal and interval scale-of-measurement variables.

The Minimum Generalized Variance (MGV) Method

The MGV method (Sarle 1984) uses an iterated multiple regression algorithm in an attempt to minimize the determinant of the covariance matrix of the transformed variables. This method transforms each variable to be (in a least squares sense) as similar to linear combinations of the remaining variables as possible. This locally minimizes the generalized variance of the transformed variables, the determinant of the covariance matrix, the volume of the parallelepiped defined by the transformed variables, and the sphericity (the extent to which a quadratic form in the optimized covariance matrix defines a sphere). See *SAS Technical Report R-108*.

On each iteration for each variable, the MGV algorithm alternates multiple regression with optimal scaling. The multiple regression involves predicting the selected variable from all other variables. You can request a dummy variable initialization by using a modification of the Tenenhaus and Vachette (1977) method that is appropriate with a regression algorithm. This method can be viewed as a way of investigating the nature of the linear and nonlinear dependencies in, and the rank of, a data matrix containing variables that can be

nonlinearly transformed. This method tries to create a less-than-full-rank data matrix. The matrix contains the transformation of each variable that is most similar to what the other transformed variables predict.

The Maximum Average Correlation (MAC) Method

The MAC method (De Leeuw 1985) uses an iterated constrained multiple regression algorithm in an attempt to maximize the average of the elements of the correlation matrix. This method transforms each variable to be (in a least squares sense) as similar to the average of the remaining variables as possible.

On each iteration for each variable, the MAC algorithm alternates computing an equally weighted average of the other variables with optimal scaling. The MAC method is similar to the MGV method in that each variable is scaled to be as similar to a linear combination of the other variables as possible, given the constraints on the transformation. However, optimal weights are not computed. You can use the MAC method when all variables are positively correlated or when no monotonicity constraints are placed on any transformations. Do not use this method with negatively correlated variables when some optimal transformations are constrained to be increasing because the signs of the correlations are not taken into account. The MAC method is useful as an initialization method for the MTV and MGV methods.

Understanding How PROC PRINQUAL Works

In the following example, PROC PRINQUAL uses the MTV method to linearize a curved scatter plot. Let

$$
\begin{aligned}
X_1 &= -1 \text{ to } 1 \text{ by } 0.02 \\
X_2 &= X_1^3 + \epsilon \\
X_3 &= X_2^5 + \epsilon
\end{aligned}
$$

where ϵ is normal error.

These three variables define a curved swarm of points in three-dimensional space. First, the SGSCATTER procedure is used to display two-dimensional views of these data. Next, PROC PRINQUAL is used to straighten the scatter plot, making it more one-dimensional by finding a smooth transformation of each variable. The N=1 option in the PROC PRINQUAL statement requests one principal component. The TRANSFORM statement requests a cubic spline transformation with nine knots. *Splines* are curves, which are usually required to be continuous and smooth. See the section "Splines" on page 7962 for more information about splines. See Smith (1979) for an excellent introduction to splines.

PROC PRINQUAL transforms each variable to be as much as possible like the first principal component (or more generally, to be close to the space defined by the first N= principal components). One component accounts for 92 percent of the variance of the untransformed data and over 99 percent of the variance of the transformed data (see Figure 96.5). Note that the results did not converge in the default 50 iterations, so more iterations were requested using the MAXITER= option. The transformations are requested by specifying PLOTS=TRANSFORMATION and are displayed in Figure 96.6.

PROC PRINQUAL creates an output data set that contains both the original and transformed variables. The original variables are named X1, X2, and X3, and the transformed variables are named TX1, TX2, and TX3. The transformed variables are displayed using the SGSCATTER procedure in Figure 96.7.

The following statements produce Figure 96.4 through Figure 96.7:

```
ods graphics on;

* Generate Three-Dimensional Data;
data X;
   do X1 = -1 to 1 by 0.02;
      X2 = X1 ** 3 + 0.05 * normal(7);
      X3 = X1 ** 5 + 0.05 * normal(7);
      output;
   end;
run;

proc sgscatter data=x;
   plot x1*x2 x1*x3 x3*x2;
run;

* Try to Straighten the Scatter Plot;
proc prinqual data=X n=1 maxiter=2000 plots=transformation out=results;
   title 'Linearize the Scatter Plot';
   transform spline(X1-X3 / nknots=9);
run;


* Plot the Linearized Scatter Plot;
proc sgscatter data=results;
   plot tx1*tx2 tx1*tx3 tx3*tx2;
run;
```

The three-dimensional data in Figure 96.4 and Figure 96.7 are displayed in three two-dimensional plots, arrayed as if they were three faces of a cube that was flattened as you might flatten a box.

Figure 96.4 Three-Dimensional Scatter Plot

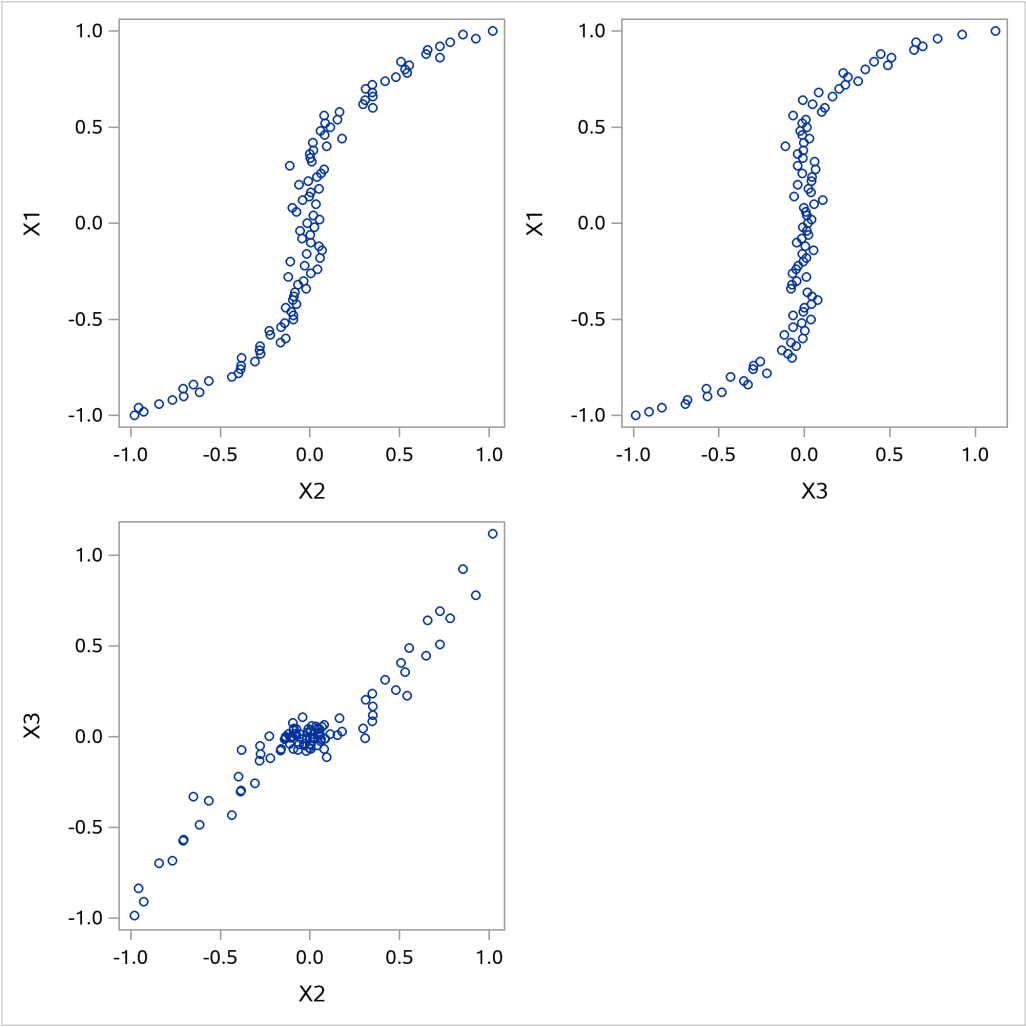


Figure 96.5 PRINQUAL Iteration History

Linearize the Scatter Plot

The PRINQUAL Procedure

PRINQUAL MTV Algorithm Iteration History

Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.15125	0.93453	0.92376		
2	0.04589	0.14682	0.98030	0.05653	
3	0.03154	0.10125	0.98626	0.00596	
4	0.02258	0.06890	0.98890	0.00265	
5	0.01682	0.04777	0.99028	0.00137	
6	0.01297	0.03782	0.99106	0.00078	
7	0.01032	0.03029	0.99154	0.00048	
.					
.					
.					
1670	0.00001	0.00005	0.99371	0.00000	
1671	0.00001	0.00005	0.99371	0.00000	
1672	0.00001	0.00005	0.99371	0.00000	Converged

```
Algorithm converged.
```

Figure 96.6 Transformations

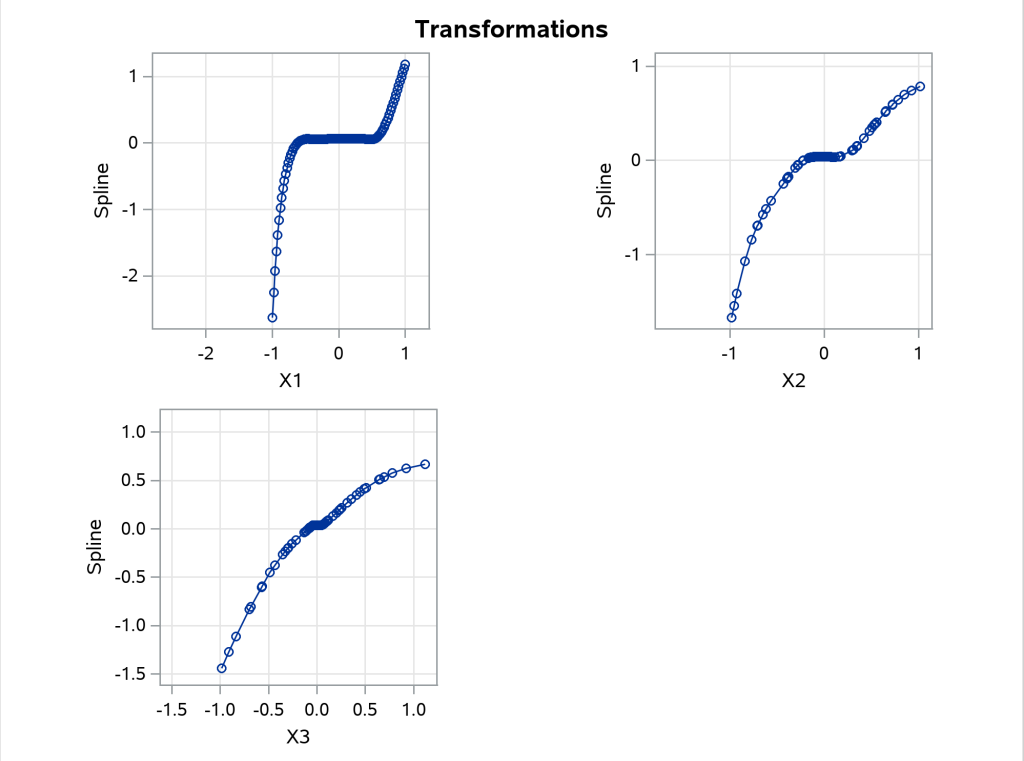
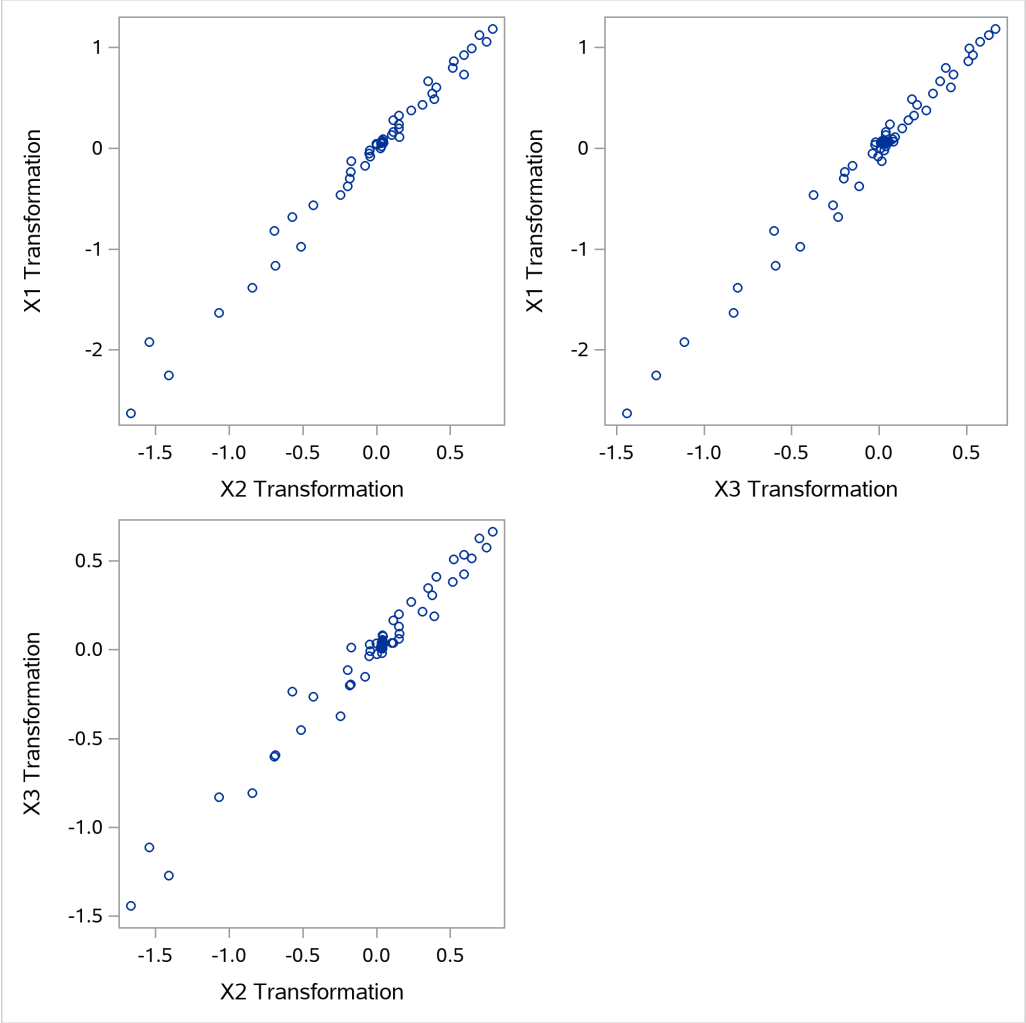


Figure 96.7 Linearized Scatter Plot



Splines

Splines are curves, and they are usually required to be continuous and smooth. Splines are usually defined as piecewise polynomials of degree n with function values and first $n - 1$ derivatives that agree at the points where they join. The abscissa values of the join points are called *knots*. The term "spline" is also used for polynomials (splines with no knots) and piecewise polynomials with more than one discontinuous derivative. Splines with no knots are generally smoother than splines with knots, which are generally smoother than splines with multiple discontinuous derivatives. Splines with few knots are generally smoother than splines with many knots; however, increasing the number of knots usually increases the fit of the spline function to the data. Knots give the curve freedom to bend to follow the data more closely. See Smith (1979) for an excellent introduction to splines. There are many examples and detailed discussions of splines in Chapter 123, "The TRANSREG Procedure." See the sections "Linear and Nonlinear Regression Functions" on page 10309, "Smoothing Splines" on page 10322, "SPLINE and MSPLINE Transformations" on page 10359, "Specifying the Number of Knots" on page 10360, "SPLINE, BSPLINE, and PSPLINE Comparisons" on page 10361,

"Linear and Nonlinear Regression Functions" on page 10309, "Simultaneously Fitting Two Regression Functions" on page 10313, and examples "Using Splines and Knots" on page 10292 and Example 123.1.

Missing Values

PROC PRINQUAL can estimate missing values, subject to optional constraints, so that the covariance matrix is optimized. The procedure provides several approaches for handling missing data. When you specify the NOMISS option in the PROC PRINQUAL statement, observations with missing values are excluded from the analysis. Otherwise, missing data are estimated, using variable means as initial estimates. Missing values for OPSCORE character variables are treated the same as any other category during the initialization. See the section "Missing Values" on page 10347 in Chapter 123, "The TRANSREG Procedure," for more information about missing data estimation.

Controlling the Number of Iterations

Several options in the PROC PRINQUAL statement control the number of iterations performed. Iteration terminates when any one of the following conditions is satisfied:

- The number of iterations equals the value of the MAXITER= option.
- The average absolute change in variable scores from one iteration to the next is less than the value of the CONVERGE= option.
- The criterion change is less than the value of the CCONVERGE= option.

With the MTV method, the change in the proportion of variance criterion can become negative when the data have converged so that it is numerically impossible, within machine precision, to increase the criterion. Because the MTV algorithm is convergent, a negative criterion change is the result of very small amounts of rounding error. The MGV method displays the average squared multiple correlation (which is not the criterion being optimized), so the criterion change can become negative well before convergence. The MAC method criterion (average correlation) is never computed, so the CCONVERGE= option is ignored for METHOD=MAC. You can specify a negative value for either convergence option if you want to define convergence only in terms of the other convergence option.

With the MGV method, iterations minimize the generalized variance (determinant), but the generalized variance is not reported for two reasons. First, in most data sets, the generalized variance is almost always near zero (or will be after one or two iterations), which is its minimum. This does not mean that iteration is complete; it simply means that at least one multiple correlation is at or near one. The algorithm continues minimizing the determinant in $(m-1)$, $(m-2)$ dimensions, and so on. Because the generalized variance is almost always near zero, it does not provide a good indication of how the iterations are progressing. The mean R square provides a better indication of convergence. The second reason for not reporting the generalized variance is that almost no additional time is required to compute R square values for each step. This is because the error sum of squares is a byproduct of the algorithm at each step. Computing the determinant at the end of each iteration adds more computations to an already computationally intensive algorithm.

You can increase the number of iterations to ensure convergence by increasing the value of the MAXITER= option and decreasing the value of the CONVERGE= option. Because the average absolute change in

standardized variable scores seldom decreases below 1E–11, you typically do not specify a value for the CONVERGE= option less than 1E–8 or 1E–10. Most of the data changes occur during the first few iterations, but the data can still change after 50 or even 100 iterations. You can try different combinations of values for the CONVERGE= and MAXITER= options to ensure convergence without extreme overiteration. If the data do not converge with the default specifications, specify the REITERATE option, or try CONVERGE=1E–8 and MAXITER=50, or CONVERGE=1E–10 and MAXITER=200.

Performing a Principal Component Analysis of Transformed Data

PROC PRINQUAL produces an iteration history table that displays (for each iteration) the iteration number, the maximum and average absolute change in standardized variable scores computed over the iteratively transformed variables, the criterion being optimized, and the criterion change. In order to examine the results of the analysis in more detail, you can analyze the information in the output data set by using other SAS procedures.

Specifically, use the PRINCOMP procedure to perform a principal components analysis on the transformed data. PROC PRINCOMP accepts the raw data from PROC PRINQUAL but issues a warning, because the PROC PRINQUAL output data set has _NAME_ and _TYPE_ variables but is not a TYPE=CORR data set. You can ignore this warning.

If the output data set contains both scores and correlations, you must subset it for analysis with PROC PRINCOMP. Otherwise, the correlation observations are treated as ordinary observations and the PROC PRINCOMP results are incorrect. For example, consider the following statements:

```
proc prinqual data=a out=b correlations replace;
   transform spline(var1-var50 / nknots=3);
run;

proc princomp data=b;
   where _TYPE_='SCORE';
run;
```

Also note that the proportion of variance accounted for, as reported by PROC PRINCOMP, can exceed the proportion of variance accounted for in the last PROC PRINQUAL iteration. This is because PROC PRINQUAL reports the variance accounted for by the components analysis that generated the current scaling of the data, not a components analysis of the current scaling of the data.

Using the MAC Method

You can use the MAC algorithm alone by specifying METHOD=MAC, or you can use it as an initialization algorithm for METHOD=MTV and METHOD=MGV analyses by specifying the iteration option INITITER=. If any variables are negatively correlated, do not use the MAC algorithm with monotonic transformations (MONOTONE, UNTIE, and MSPLINE) because the signs of the correlations among the variables are not used when computing variable approximations. If an approximation is negatively correlated with the original variable, monotone constraints would make the optimally scaled variable a constant, which is not allowed (see the section "Avoiding Constant Transformations" on page 7968). When used with other transformations, the MAC algorithm can reverse the scoring of the variables. So, for example, if variable X is designated LOG(X) with METHOD=MAC and TSTANDARD=ORIGINAL, the final transformation (for example, TX)

might not be LOG(X). If TX is not LOG(X), it has the same mean as LOG(X) and the same variance as LOG(X), and it is perfectly negatively correlated with LOG(X). PROC PRINQUAL displays a note for every variable that is reversed in this manner.

You can use the METHOD=MAC algorithm to reverse the scorings of some rating variables before a factor analysis. The correlations among bipolar ratings such as 'like - dislike', 'hot - cold', and 'fragile - monumental' are typically both positive and negative. If some items are reversed to say 'dislike - like', 'cold - hot', and 'monumental - fragile', some of the negative signs can be eliminated, and the factor pattern matrix would be cleaner. You can use PROC PRINQUAL with METHOD=MAC and LINEAR transformations to reverse some items, maximizing the average of the intercorrelations.

Output Data Set

PROC PRINQUAL produces an output data set by default. By specifying the OUT=, APPROXIMATIONS, SCORES, REPLACE, and CORRELATIONS options in the PROC PRINQUAL statement, you can name this data set and control its contents.

By default, the procedure creates an output data set that contains variables with _TYPE_='SCORE'. These observations contain original variables, transformed variables, components, or data approximations. If you specify the CORRELATIONS option in the PROC PRINQUAL statement, the data set also contains observations with _TYPE_='CORR'; these observations contain correlations or component structure information.

Structure and Content

The output data set can have 16 different forms, depending on the specified combinations of the REPLACE, SCORES, APPROXIMATIONS, and CORRELATIONS options. You can specify any combination of these options. To illustrate, assume that the data matrix consists of N observations and m variables, and n components are computed. Then define the following:

D the $N \times m$ matrix of original data with variable names that correspond to the names of the variables in the input data set. However, when you use the OPSCORE transformation on character variables, those variables are replaced by numeric variables that contain category numbers.

T the $N \times m$ matrix of transformed data with variable names constructed from the value of the TPREFIX= option (if you do not specify the REPLACE option) and the names of the variables in the input data set

S the $N \times n$ matrix of component scores with variable names constructed from the value of the PREFIX= option and integers

A the $N \times m$ matrix of data approximations with variable names constructed from the value of the APREFIX= option and the names of the variables in the input data set

$\mathbf{R_{TD}}$ the $m \times m$ matrix of correlations between the transformed variables and the original variables with variable names that correspond to the names of the variables in the input data set. When missing values exist, casewise deletion is used to compute the correlations.

$\mathbf{R_{TT}}$ the $m \times m$ matrix of correlations among the transformed variables with the variable names constructed from the value of the TPREFIX= option (if you do not specify the REPLACE option) and the names of the variables in the input data set

$\mathbf{R_{TS}}$ the $m \times n$ matrix of correlations between the transformed variables and the principal component scores (component structure matrix) with variable names constructed from the value of the PREFIX= option and integers

$\mathbf{R_{TA}}$ the $m \times m$ matrix of correlations between the transformed variables and the variable approximations with variable names constructed from the value of the APREFIX= option and the names of the variables in the input data set

To create a data set Work.A that contains all information, specify the following options in the PROC PRINQUAL statement:

```
proc prinqual scores approximations correlations out=a;
```

Also use a TRANSFORM statement appropriate for your data. Then the Work.A data set contains the following:

D	T	S	A
$\mathbf{R_{TD}}$	$\mathbf{R_{TT}}$	$\mathbf{R_{TS}}$	$\mathbf{R_{TA}}$

To eliminate the bottom partitions that contain the correlations and component structure, do not specify the CORRELATIONS option. For example, use the following PROC PRINQUAL statement with an appropriate TRANSFORM statement:

```
proc prinqual scores approximations out=a;
```

Then the Work.A data set contains the following:

D	T	S	A

Suppose you use the following PROC PRINQUAL statement (with an appropriate TRANSFORM statement):

```
proc prinqual out=a;
```

This creates a data set Work.A of the following form:

D	T

To output transformed data and component scores only, specify the following options in the PROC PRINQUAL statement:

```
proc prinqual replace scores out=a;
```

Then the Work.A data set contains the following:

T	S

TYPE and _NAME_ Variables

In addition to the preceding information, the output data set contains two character variables, the variable _TYPE_ (length 8) and the variable _NAME_ (the length depends on the VALIDVARNAME= system option).

The _TYPE_ variable has the value 'SCORE' if the observation contains variables, transformed variables, components, or data approximations; the _TYPE_ variable has the value 'CORR' if the observation contains correlations or component structure.

By default, the _NAME_ variable has values 'ROW1', 'ROW2', and so on, for the observations with _TYPE_='SCORE'. If you use an ID statement, the variable _NAME_ contains the formatted ID variable for SCORES observations. The values of the variable _NAME_ for observations with _TYPE_='CORR' are the names of the transformed variables.

Certain procedures, such as PROC PRINCOMP, which can use the PROC PRINQUAL output data set, issue a warning that the PROC PRINQUAL data set contains _NAME_ and _TYPE_ variables but is not a TYPE=CORR data set. You can ignore this warning.

Variable Names

The TPREFIX=, APREFIX=, and PREFIX= options specify prefixes for the transformed and approximation variable names and for principal component score variables, respectively. PROC PRINQUAL constructs transformed and approximation variable names from a prefix and the first characters of the original variable name. The number of characters in the prefix plus the number of characters in the original variable name (including the final digits, if any) required to uniquely designate the new variables should not exceed v, where v is the length from the VALIDVARNAME= system option (for example, 32 for VALIDVARNAME=V7). If the APREFIX= parameter that you specify is one character, PROC PRINQUAL adds the first $v - 1$ characters of the original variable name; if your prefix is four characters, only the first $v - 4$ characters of the original variable name are added.

Effect of the TSTANDARD= and COVARIANCE Options

The values in the output data set are affected by the TSTANDARD= and COVARIANCE options. If you specify TSTANDARD=NOMISS, the NOMISS standardization is performed on the transformed data after the iterations have been completed, but before the output data set is created. The new means and variances are used in creating the output data set. Then, if you do not specify the COVARIANCE option, the data are transformed to mean zero and variance one. The principal component scores and data approximations are computed from the resulting matrix. The data are then linearly transformed to have the mean and variance specified by the TSTANDARD= option. The data approximations are transformed so that the means within each pair of a transformed variable and its approximation are the same. The ratio of the variance of a variable approximation to the variance of the corresponding transformed variable equals the proportion of the variance of the variable that is accounted for by the components model.

If you specify the COVARIANCE option and do not specify TSTANDARD=Z, you can input the transformed data to PROC PRINCOMP, again specifying the COVARIANCE option, to perform a components analysis of the results of PROC PRINQUAL. Similarly, if you do not specify the COVARIANCE option with PROC PRINQUAL and you input the transformed data to PROC PRINCOMP without the COVARIANCE option, you receive the same report. However, some combinations of PROC PRINQUAL options, such as COVARIANCE and TSTANDARD=Z, while valid, produce approximations and scores that cannot be reproduced by PROC PRINCOMP.

The component scores in the output data set are computed from the correlations among the transformed variables, or from the covariances if you specified the COVARIANCE option. The component scores are computed after the TSTANDARD=NOMISS transformation, if specified. The means of the component scores in the output data set are always zero. The variances equal the corresponding eigenvalues, unless you specify the STANDARD option; then the variances are set to one.

Avoiding Constant Transformations

There are times when the optimal scaling produces a constant transformed variable. This can happen with the MONOTONE, UNTIE, and MSPLINE transformations when the target is negatively correlated with the original input variable. It can happen with all transformations when the target is uncorrelated with the original input variable. When this happens, the procedure modifies the target to avoid a constant transformation. This strategy avoids certain nonoptimal solutions.

If the transformation is monotonic and a constant transformed variable results, the procedure multiplies the target by –1 and tries the optimal scaling again. If the transformation is not monotonic or if the multiplication by –1 did not help, the procedure tries using a random target. If the transformation is still constant, the previous nonconstant transformation is retained. When a constant transformation is avoided by any strategy, this message is displayed: "A constant transformation was avoided for *name*."

Constant Variables

Constant and almost constant variables are zeroed and ignored.

Character OPSCORE Variables

Character OPSCORE variables are replaced by a numeric variable containing category numbers before the iterations, and the character values are discarded. Only the first eight characters are considered in determining category membership. If you want the original character variable in the output data set, give it a different name in the OPSCORE specification (OPSCORE(x / name=(x2)) and name the original variable in the ID statement (ID x;).

REITERATE Option Usage

You can use the REITERATE option to perform additional iterations when PROC PRINQUAL stops before the data have adequately converged. For example, suppose you execute the following code:

```
proc prinqual data=A cor out=B;
   transform mspline(X1-X5);
run;
```

If the transformations do not converge in the default 30 iterations, you can perform more iterations without repeating the first 30 iterations, as follows:

```
proc prinqual data=B reiterate cor out=B;
   transform mspline(X1-X5);
run;
```

Note that a WHERE statement is not necessary to exclude the correlation observations. They are automatically excluded because their _TYPE_ variable value is not 'SCORE'.

You can also use the REITERATE option to specify starting values other than the original values for the transformations. Providing alternate starting points might avoid local optima. Here are two examples.

```
proc prinqual data=A out=B;
   transform rank(X1-X5);
run;

proc prinqual data=B reiterate out=C;
   /* Use ranks as the starting point. */
   transform monotone(X1-X5);
run;

data B;
   set A;
   array TXS[5] TX1-TX5;
   do j = 1 to 5;
      TXS[j] = normal(0);
   end;
run;

proc prinqual data=B reiterate out=C;
   /* Use a random starting point. */
   transform monotone(X1-X5);
run;
```

Note that divergence with the REITERATE option, particularly in the second iteration, is not an error since the initial transformation is not required to be a valid member of the transformation family. When you specify the REITERATE option, the iteration does not terminate when the criterion change is negative during the first 10 iterations.

Passive Observations

Observations can be excluded from the analysis for several reasons, including zero weight, zero frequency, missing values in variables designated as IDENTITY, or missing values with the NOMISS option specified. These observations are passive in that they do not contribute to determining transformations, R square, total variance, and so on. However, some information can be computed for them, such as approximations, principal component scores, and transformed values. Passive observations in the output data set have a blank value for the variable _TYPE_.

Missing value estimates for passive observations might converge slowly with METHOD=MTV. In the following example, the missing value estimates should be 2, 5, and 8. Since the nonpassive observations do not change, the procedure converges in one iteration but the missing value estimates do not converge. The extra iterations produced by specifying CONVERGE=–1 and CCONVERGE=–1, as shown in the second PROC PRINQUAL step that follows, generate the expected results.

```
data A;
   input X Y;
   datalines;
1 1
2 .
3 3
4 4
5 .
6 6
7 7
8 .
9 9
;

proc prinqual nomiss data=A nomiss n=1 out=B method=mtv;
   transform lin(X Y);
run;

proc print;
run;

proc prinqual nomiss data=A nomiss n=1 out=B method=mtv converge=-1 cconverge=-1;
   transform lin(X Y);
run;

proc print;
run;
```

Computational Resources

This section provides information about the computational resources required to run PROC PRINQUAL.

Let

$$
\begin{aligned}
N &= \text{number of observations} \\
m &= \text{number of variables} \\
n &= \text{number of principal components} \\
k &= \text{maximum spline degree} \\
p &= \text{maximum number of knots}
\end{aligned}
$$

- For the MTV algorithm, more than

$$56m + 8Nm + 8\left(6N + (p + k + 2)(p + k + 11)\right)$$

bytes of array space are required.

- For the MGV and MAC algorithms, more than $56m$ plus the maximum of the data matrix size and the optimal scaling work space bytes of array space are required. The data matrix size is $8Nm$ bytes. The optimal scaling work space requires less than $8\,(6N + (p + k + 2)(p + k + 11))$ bytes.
- For the MTV and MGV algorithms, more than $56m + 4m(m + 1)$ bytes of array space are required.
- PROC PRINQUAL tries to store the original and transformed data in memory. If there is not enough memory, a utility data set is used, potentially resulting in a large increase in execution time. The amount of memory for the preceding data formulas is an underestimate of the amount of memory needed to handle most problems. These formulas give an absolute minimum amount of memory required. If a utility data set is used, and if memory could be used with perfect efficiency, then roughly the amount of memory stated previously would be needed. In reality, most problems require at least two or three times the minimum.
- PROC PRINQUAL sorts the data once. The sort time is roughly proportional to $mN^{3/2}$.
- For the MTV algorithm, the time required to compute the variable approximations is roughly proportional to $2Nm^2 + 5m^3 + nm^2$.
- For the MGV algorithm, one regression analysis per iteration is required to compute model parameter estimates. The time required to accumulate the crossproduct matrix is roughly proportional to Nm^2. The time required to compute the regression coefficients is roughly proportional to m^3. For each variable for each iteration, the swept crossproduct matrix is updated with time roughly proportional to m(N+m). The swept crossproduct matrix is updated for each variable with time roughly proportional to m^2, until computations are refreshed, requiring all sweeps to be performed again.
- The only computationally intensive part of the MAC algorithm is the optimal scaling, since variable approximations are simple averages.
- Each optimal scaling is a multiple regression problem, although some transformations are handled with faster special-case algorithms. The number of regressors for the optimal scaling problems depends on the original values of the variable and the type of transformation. For each monotone spline transformation, an unknown number of multiple regressions is required to find a set of coefficients that satisfies the constraints. The B-spline basis is generated twice for each SPLINE and MSPLINE transformation for each iteration. The time required to generate the B-spline basis is roughly proportional to Nk^2.

Displayed Output

The main output from PROC PRINQUAL is the output data set. However, the procedure does produce displayed output in the form of an iteration history table that includes the following:

- iteration number
- the criterion being optimized
- criterion change
- maximum and average absolute change in standardized variable scores computed over variables that can be iteratively transformed
- notes
- final convergence status

ODS Table Names

PROC PRINQUAL assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 96.4 along with the PROC PRINQUAL statement options needed to produce the table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 96.4 ODS Tables Produced by PROC PRINQUAL

ODS Table Name	Description	Option
ConvergenceStatus	Convergence Status	default
Footnotes	Iteration History Footnotes	default
MAC	MAC Iteration History	METHOD=MAC
MGV	MGV Iteration History	METHOD=MGV
MTV	MTV Iteration History	METHOD=MTV
PctVar	Percentage of Variance	nonprinting

The nonprinting PctVar table is not displayed and does not appear in the ODS trace output unless you specify it in an ODS OUTPUT statement, as in the following example:

```
ods output pctvar=pvardataset;
```

ODS Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS."

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 623 in Chapter 21, "Statistical Graphics Using ODS."

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section "A Primer on ODS Statistical Graphics" on page 622 in Chapter 21, "Statistical Graphics Using ODS."

The plots are produced only when you specify the options shown in the table. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC PRINQUAL generates are listed in Table 96.5, along with the required statements and options.

Table 96.5 Graphs Produced by PROC PRINQUAL

ODS Graph Name	Plot Description	Option
MDPrefPlot	Multidimensional preference analysis	MDPREF
TransformationPlot	Variable transformation	PLOTS=TRANSFORMATION

Examples: PRINQUAL Procedure

Example 96.1: Multidimensional Preference Analysis of Automobile Data

This example uses PROC PRINQUAL to perform a nonmetric multidimensional preference (MDPREF) analysis (Carroll 1972). MDPREF analysis is a principal component analysis of a data matrix with columns that correspond to people and rows that correspond to objects. The data are ratings or rankings of each person's preference for each object. The data are the transpose of the usual multivariate data matrix. (In other words, the columns are people; in the more typical matrix the rows represent people.) The final result of an MDPREF analysis is a biplot (Gabriel 1981) of the resulting preference space. A biplot displays the judges and objects in a single plot by projecting them onto the plane in the transformed variable space that accounts for the most variance.

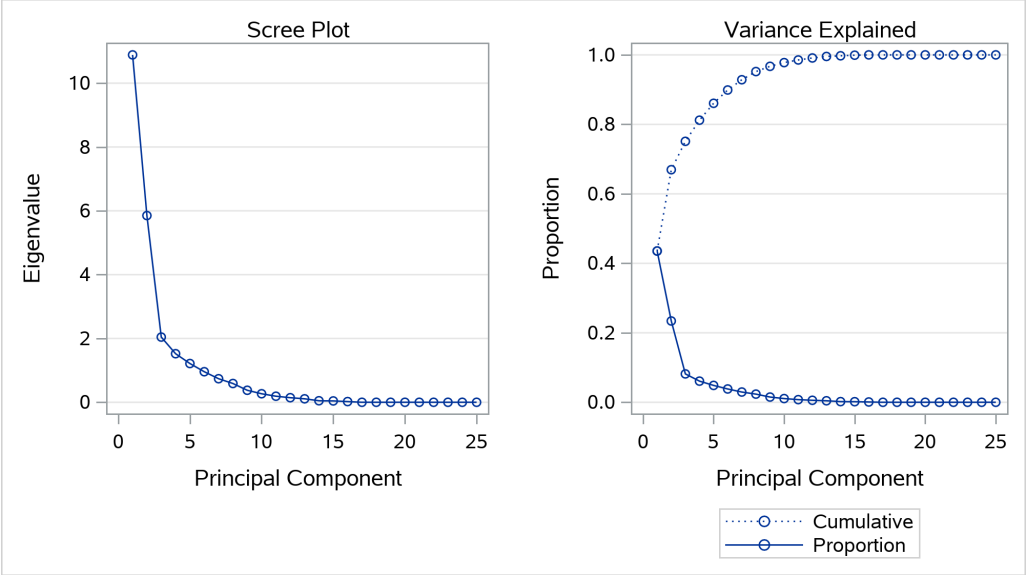
In 1980, 25 judges gave their preferences for each of 17 new automobiles. The ratings were made on a 0 to 9 scale, with 0 meaning very weak preference and 9 meaning very strong preference for the automobile. The following statements create a SAS data set with the manufacturer and model of each automobile along with the ratings:

```
title 'Preference Ratings for Automobiles Manufactured in 1980';

options validvarname=any;

data CarPref;
   input Make $ 1-10 Model $ 12-22 @25 ('1'n-'25'n) (1.);
   datalines;
Cadillac   Eldorado     800799049124050897109380 9
Chevrolet  Chevette     005120042345104300351569 8
Chevrolet  Citation     405330581416164354474779 5
Chevrolet  Malibu       602740072312134554566865 8
Ford       Fairmont     202400671502144353064865 5
Ford       Mustang      500719770502110185065755 6
Ford       Pinto        002100030303020150051407 8
Honda      Accord       595689760969995299897507 8
Honda      Civic        483670950748885256776507 5
Lincoln    Continental  700899059223040996209190 9
Plymouth   Gran Fury    700600043410110733345870 8
Plymouth   Horizon      300500563546130244467565 5
Plymouth   Volare       400500361402160275447655 5
Pontiac    Firebird     010789561320120695826590 7
Volkswagen Dasher       485869650887779537789500 0
Volkswagen Rabbit       485850970969579548788500 0
Volvo      DL           998999890999998798991900 0
;
```

The following statements run PROC PRINCOMP and create a scree plot. The results of this step are shown in Output 96.1.1.

```
ods graphics on;
* Principal Component Analysis of the Original Data;
proc princomp data=CarPref;
   ods select ScreePlot;
   var '1'n-'25'n;
run;
```

Output 96.1.1 Eigenvalue Plot



The scree or eigenvalue plot in Output 96.1.1 shows that two principal components should be retained. There is a clear separation between the first two components and the remaining components. There are eight eigenvalues that are precisely zero because there are eight fewer observations than variables in the data matrix. One additional eigenvalue is zero, for a total of nine zero eigenvalues, since the correlation matrix is based on centered data. The following statements create the data set and perform a principal component analysis of the original data.

PROC PRINQUAL fits the nonmetric MDPREF model. PROC PRINQUAL monotonically transforms the raw judgments to maximize the proportion of variance accounted for by the first two principal components. The MONOTONE option is specified in the TRANSFORM statement to request a nonmetric MDPREF analysis; alternatively, you can instead specify the IDENTITY option for a metric analysis. Several options are used in the PROC PRINQUAL statement. The option DATA=CarPref specifies the input data set, OUT=Results creates an output data set, and N=2 and the default METHOD=MTV transform the data to better fit a two-component model. The REPLACE option replaces the original data with the monotonically transformed data in the OUT= data set. The MDPREF option standardizes the component scores to variance one so that the geometry of the biplot is correct, and it creates two variables in the OUT= data set named Prin1 and Prin2. These variables contain the standardized principal component scores and structure matrix, which are used to make the biplot. If the variables in data matrix $\mathbf{X}$ are standardized to mean zero and variance one, and n is the number of rows in $\mathbf{X}$, then $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{W}'$ is the principal component model, where $\mathbf{X}'\mathbf{X}/(n-1) = \mathbf{W}\mathbf{\Lambda}\mathbf{W}'$. The $\mathbf{W}$ and $\mathbf{\Lambda}$ contain the eigenvectors and eigenvalues of the correlation matrix of $\mathbf{X}$. The first two columns of $\mathbf{V}$, the standardized component scores, and $\mathbf{W}\mathbf{\Lambda}^{1/2}$, which is the structure matrix, are output. The advantage of creating a biplot based on principal components is that coordinates do not depend on the sample size. The following statements transform the data and produce Output 96.1.2.

```
* Transform the Data to Better Fit a Two Component Model;
proc prinqual data=CarPref out=Results n=2 replace mdpref;
   title2 'Multidimensional Preference (MDPREF) Analysis';
   title3 'Optimal Monotonic Transformation of Preference Data';
   id model;
   transform monotone('1'n-'25'n);
run;
```

Output 96.1.2 PRINQUAL Iteration History

Preference Ratings for Automobiles Manufactured in 1980
Multidimensional Preference (MDPREF) Analysis
Optimal Monotonic Transformation of Preference Data

The PRINQUAL Procedure

PRINQUAL MTV Algorithm Iteration History

Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.24994	1.28017	0.66946		
2	0.07223	0.36958	0.80194	0.13249	
3	0.04522	0.29026	0.81598	0.01404	
4	0.03096	0.25213	0.82178	0.00580	
5	0.02182	0.23045	0.82493	0.00315	
6	0.01602	0.19017	0.82680	0.00187	
7	0.01219	0.14748	0.82793	0.00113	
8	0.00953	0.11031	0.82861	0.00068	
9	0.00737	0.06461	0.82904	0.00043	
10	0.00556	0.04469	0.82930	0.00026	
11	0.00445	0.04087	0.82944	0.00014	
12	0.00381	0.03706	0.82955	0.00011	
13	0.00319	0.03348	0.82965	0.00009	
14	0.00255	0.02999	0.82971	0.00006	
15	0.00213	0.02824	0.82976	0.00005	
16	0.00183	0.02646	0.82980	0.00004	
17	0.00159	0.02472	0.82983	0.00003	
18	0.00139	0.02305	0.82985	0.00003	
19	0.00123	0.02145	0.82988	0.00002	
20	0.00109	0.01993	0.82989	0.00002	
21	0.00096	0.01850	0.82991	0.00001	
22	0.00086	0.01715	0.82992	0.00001	
23	0.00076	0.01588	0.82993	0.00001	
24	0.00067	0.01440	0.82994	0.00001	
25	0.00059	0.00871	0.82994	0.00001	
26	0.00050	0.00720	0.82995	0.00000	
27	0.00043	0.00642	0.82995	0.00000	
28	0.00037	0.00573	0.82995	0.00000	
29	0.00031	0.00510	0.82995	0.00000	
30	0.00027	0.00454	0.82995	0.00000	Not Converged

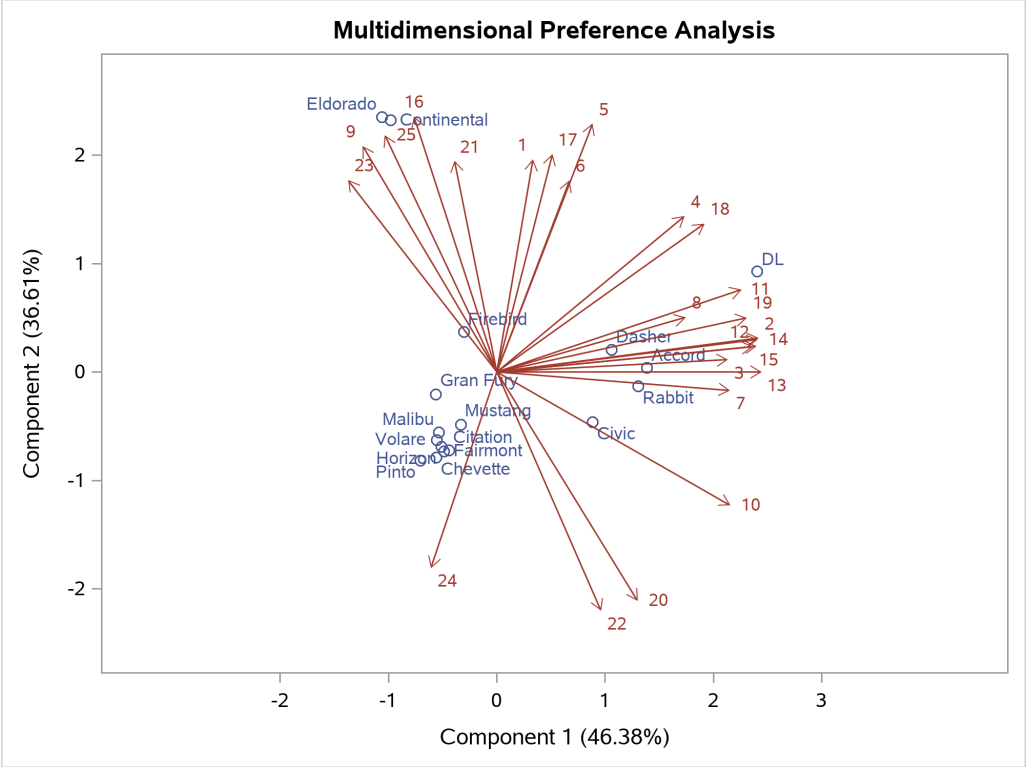
WARNING: Failed to converge, however criterion change is less than 0.0001.

The iteration history displayed by PROC PRINQUAL indicates that the proportion of variance is increased from an initial 0.66946 to 0.82995. The proportion of variance accounted for by PROC PRINQUAL on the first iteration equals the cumulative proportion of variance shown by PROC PRINCOMP for the first two principal components. PROC PRINQUAL's initial iteration performs a standard principal component analysis of the raw data. The columns labeled Average Change, Maximum Change, and Criterion Change contain values that always decrease, indicating that PROC PRINQUAL is improving the transformations at a monotonically decreasing rate over the iterations. This does not always happen, and when it does not, it

suggests that the analysis might be converging to a degenerate solution. See Example 96.2 for a discussion of a degenerate solution. The algorithm does not converge in 30 iterations. However, the criterion change is small, indicating that more iterations are unlikely to have much effect on the results.

The biplot, shown in Output 96.1.3, is automatically displayed by PROC PRINQUAL when ODS Graphics is enabled and the MDPREF option is specified.

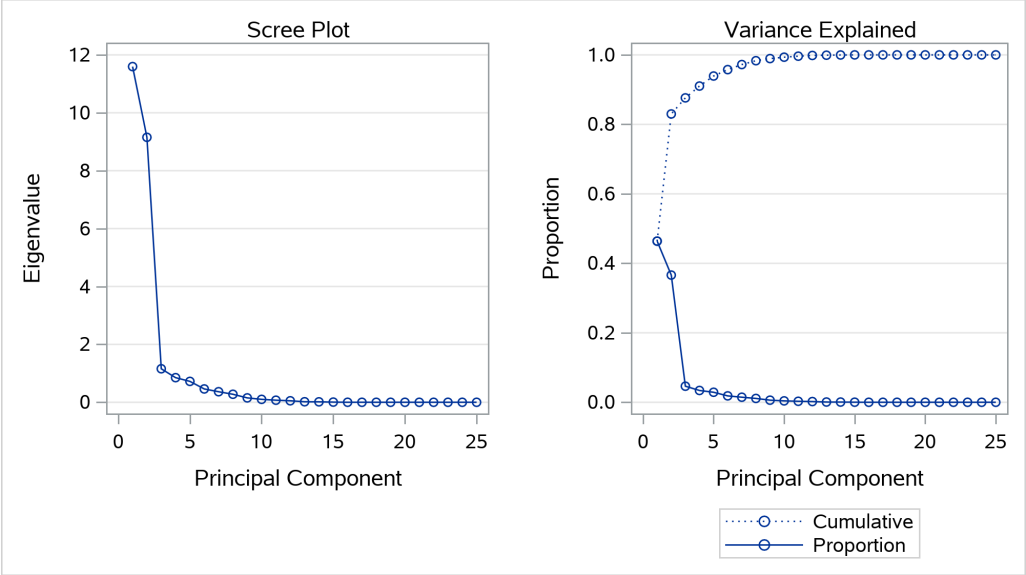
Output 96.1.3 Biplot Made with PRINQUAL



The second PROC PRINCOMP analysis is performed on the transformed data. The WHERE statement is used to retain only the monotonically transformed judgments. The scree plot shows that the first two eigenvalues are now much larger than the remaining smaller eigenvalues. The second eigenvalue has increased markedly at the expense of the next several eigenvalues. Two principal components seem to be necessary and sufficient to adequately describe these judges' preferences for these automobiles. The cumulative proportion of variance displayed by PROC PRINCOMP for the first two principal components is 0.83. The following statements perform the analysis and produce Output 96.1.4:

```
* Final Principal Component Analysis;
proc princomp data=Results;
   ods select ScreePlot;
   var '1'n-'25'n;
   where _TYPE_='SCORE';
run;
```

Output 96.1.4 Transformed Data Eigenvalue Plot



The remainder of the example discusses the MDPREF biplot. A biplot is a plot that displays the relation between the row points and the columns of a data matrix. The rows of $\mathbf{V}$, the standardized component scores, and $\mathbf{W}\mathbf{\Lambda}^{1/2}$, the structure matrix, contain enough information to reproduce $\mathbf{X}$. The (i, j) element of $\mathbf{X}$ is the product of row i of $\mathbf{V}$ and row j of $\mathbf{W}\mathbf{\Lambda}^{1/2}$. If all but the first two columns of $\mathbf{V}$ and $\mathbf{W}\mathbf{\Lambda}^{1/2}$ are discarded, the (i, j) element of $\mathbf{X}$ is approximated by the product of row i of $\mathbf{V}$ and row j of $\mathbf{W}\mathbf{\Lambda}^{1/2}$.

Since the MDPREF analysis is based on a principal component model, the dimensions of the MDPREF biplot are the first two principal components. The first principal component is the longest dimension through the MDPREF biplot. The first principal component is overall preference, which is the most salient dimension in the preference judgments. One end points in the direction that is on the average preferred most by the judges, and the other end points in the least preferred direction. The second principal component is orthogonal to the first principal component, and it is the orthogonal direction that is the second most salient. The interpretation of the second dimension varies from example to example.

With an MDPREF biplot, it is geometrically appropriate to represent each automobile (object) by a point and each judge by a vector. The automobile points have coordinates that are the scores of the automobile on the first two principal components. The judge vectors emanate from the origin of the space and go through a point whose coordinates are the coefficients of the judge (variable) on the first two principal components.

The absolute length of a vector is arbitrary. However, the relative lengths of the vectors indicate fit, with the squared lengths being proportional to the communalities that you can get in PROC FACTOR output. The direction of the vector indicates the direction that is most preferred by the individual judge, with preference increasing as the vector moves from the origin. Let $\mathbf{v}'$ be row i of $\mathbf{V}$, $\mathbf{u}'$ be row j of $\mathbf{U} = \mathbf{W}\mathbf{\Lambda}^{1/2}$, $\|\mathbf{v}\|$ be the length of $\mathbf{v}$, $\|\mathbf{u}\|$ be the length of $\mathbf{u}$, and θ be the angle between $\mathbf{v}$ and $\mathbf{u}$. The predicted degree of preference that an individual judge has for an automobile is $\mathbf{u}'\mathbf{v} = \|\mathbf{u}\|\,\|\mathbf{v}\|\cos\theta$. Each automobile point can be orthogonally projected onto the vector. The projection of automobile i on vector j is $\mathbf{u}((\mathbf{u}'\mathbf{v})/(\mathbf{u}'\mathbf{u}))$, and the length of this projection is $\|\mathbf{v}\|\cos\theta$. The automobile that projects farthest along a vector in the direction it points is that judge's most preferred automobile, since the length of this projection, $\|\mathbf{v}\|\cos\theta$, differs from the predicted preference, $\|\mathbf{u}\|\,\|\mathbf{v}\|\cos\theta$, only by $\|\mathbf{u}\|$, which is constant for each judge.

To interpret the biplot, look for directions through the plot that show a continuous change in some attribute of the automobiles, or look for regions in the plot that contain clusters of automobile points and determine what attributes the automobiles have in common. Points that are tightly clustered in a region of the plot represent automobiles that have the same preference patterns across the judges. Vectors that point in roughly the same direction represent judges who have similar preference patterns.

In the biplot, American automobiles are located at the left of the space, while European and Japanese automobiles are located at the right. At the top of the space are expensive American automobiles (Cadillac Eldorado, Lincoln Continental), while at the bottom are inexpensive ones (Ford Pinto, Chevrolet Chevette). The first principal component differentiates American from imported automobiles, and the second arranges automobiles by price and other associated characteristics.

The two expensive American automobiles form a cluster, the sporty automobile (Pontiac Firebird) is by itself, the Volvo DL is by itself, and the remaining imported autos form a cluster, as do the remaining American autos. It seems there are 5 prototypical automobiles in this set of 17, in terms of preference patterns among the 25 judges.

Most of the judges prefer the imported automobiles, especially the Volvo. There is also a fairly large minority that prefer the expensive autos, whether or not they are American (those with vectors that point toward one o'clock), or simply prefer expensive American automobiles (vectors that point toward eleven o'clock). There are two judges who prefer anything except expensive American autos (five o'clock vectors), and one who prefers inexpensive American autos (seven o'clock vector).

Several vectors point toward the upper-right corner of the plot, toward a region with no automobiles. This is the region between the European and Japanese autos at the right and the luxury autos at the top. This suggests that there is a market for luxury Japanese and European automobiles.

The next part of this example modifies the graph template for the MDPREF plot to display group information (the make of the automobile) in the MDPREF plot. First you need to run the PROC PRINQUAL step with ODS trace output enabled to find the name of the graph template for the MDPREF plot:

```
ods trace on;
proc prinqual data=CarPref out=Results n=2 replace mdpref;
   title2 'Multidimensional Preference (MDPREF) Analysis';
   title3 'Optimal Monotonic Transformation of Preference Data';
   id model;
   transform monotone('1'n-'25'n);
run;
```

The results are as follows:

```
Output Added:
-------------
Name:        MDPrefPlot
Label:       2 by 1
Template:    Stat.Prinqual.Graphics.MDPref
Path:        Prinqual.MDPREF.MDPrefPlot
-------------
```

The following step displays the template:

```
proc template;
   source Stat.Prinqual.Graphics.MDPref;
run;
```

The following template is displayed:

```
define statgraph Stat.Prinqual.Graphics.MDPref;
   notes "Multidimensional Preference Analysis Plot";
   dynamic xVar yVar xVec yVec ylab xlab yshortlab xshortlab xOri yOri
      stretch;
   begingraph;
      entrytitle "Multidimensional Preference Analysis";
      layout overlayequated / equatetype=fit xaxisopts=(label=XLAB shortlabel
         =XSHORTLAB offsetmin=0.1 offsetmax=0.1) yaxisopts=(label=YLAB
         shortlabel=YSHORTLAB offsetmin=0.1 offsetmax=0.1);
         scatterplot y=YVAR x=XVAR / datalabel=IDLAB1 rolename=(_tip1=
            OBSNUMVAR _id2=IDLAB2 _id3=IDLAB3 _id4=IDLAB4 _id5=IDLAB5 _id6=
            IDLAB6 _id7=IDLAB7 _id8=IDLAB8 _id9=IDLAB9 _id10=IDLAB10 _id11=
            IDLAB11 _id12=IDLAB12 _id13=IDLAB13 _id14=IDLAB14 _id15=IDLAB15
            _id16=IDLAB16 _id17=IDLAB17 _id18=IDLAB18 _id19=IDLAB19 _id20=
            IDLAB20) tip=(y x datalabel _tip1 _id2 _id3 _id4 _id5 _id6 _id7
            _id8 _id9 _id10 _id11 _id12 _id13 _id14 _id15 _id16 _id17 _id18
            _id19 _id20) datalabelattrs=GRAPHVALUETEXT (color=
            GraphData1:ContrastColor) markerattrs=GRAPHDATA1;
         vectorplot y=YVEC x=XVEC xorigin=0 yorigin=0 / datalabel=LABEL2VAR
            shaftprotected=false rolename=(_tip1=VNAME _tip2=VLABEL _tip3=
            YORI _tip4=XORI _tip5=LENGTH _tip6=LENGTH2) tip=(y x datalabel
            _tip1 _tip2 _tip3 _tip4 _tip5 _tip6) datalabelattrs=
            GRAPHVALUETEXT (color=GraphData2:ContrastColor) lineattrs=
            GRAPHDATA2 (pattern=solid) primary=true;
         if (0)
            entry "Vector Stretch = " STRETCH / autoalign=(topright topleft
               bottomright bottomleft right left top bottom);
         endif;
      endlayout;
   endgraph;
end;
```

The following step adds a PROC PRINQUAL statement and a RUN statement, removes attribute options from the SCATTERPLOT statement, adds a GROUP=IDLAB2 option to use the second ID variable as a group variable, and adds a NAME='s' option along with a DISCRETELEGEND statement to display the groups in a legend:

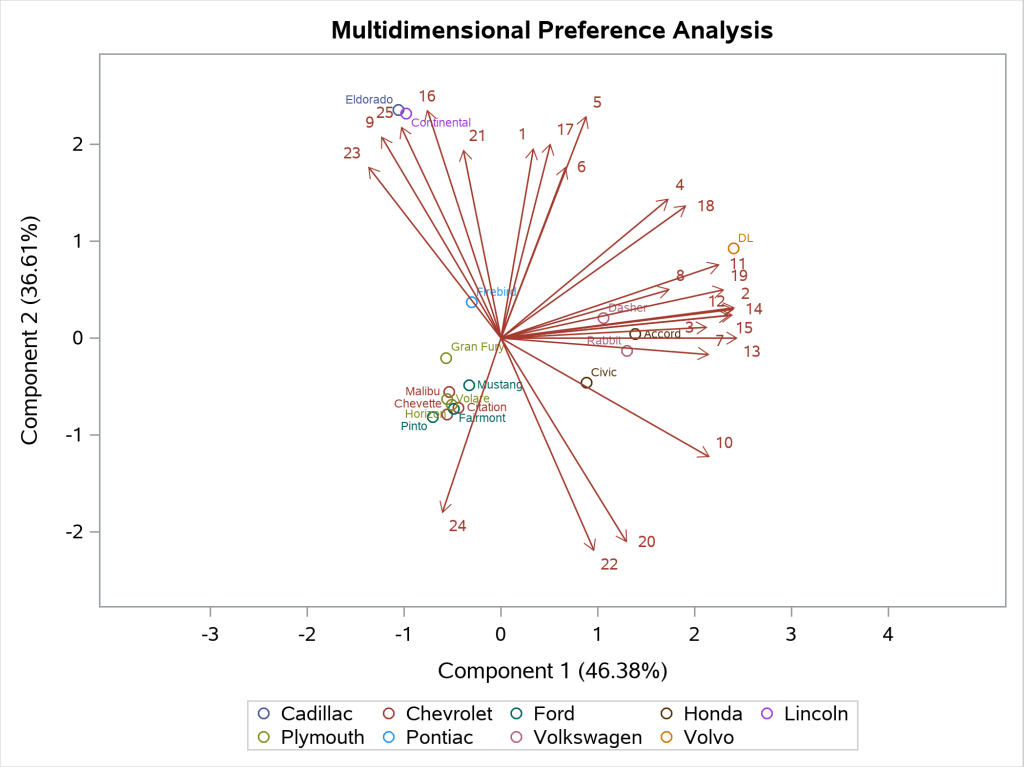
```
proc template;
   define statgraph Stat.Prinqual.Graphics.MDPref;
      notes "Multidimensional Preference Analysis Plot";
      dynamic xVar yVar xVec yVec ylab xlab yshortlab xshortlab xOri yOri
         stretch;
      begingraph;
         entrytitle "Multidimensional Preference Analysis";
         layout overlayequated / equatetype=fit xaxisopts=(label=XLAB shortlabel
            =XSHORTLAB offsetmin=0.1 offsetmax=0.1) yaxisopts=(label=YLAB
            shortlabel=YSHORTLAB offsetmin=0.1 offsetmax=0.1);
            scatterplot y=YVAR x=XVAR / datalabel=IDLAB1 rolename=(_tip1=
               OBSNUMVAR _id2=IDLAB2 _id3=IDLAB3 _id4=IDLAB4 _id5=IDLAB5 _id6=
               IDLAB6 _id7=IDLAB7 _id8=IDLAB8 _id9=IDLAB9 _id10=IDLAB10 _id11=
               IDLAB11 _id12=IDLAB12 _id13=IDLAB13 _id14=IDLAB14 _id15=IDLAB15
               _id16=IDLAB16 _id17=IDLAB17 _id18=IDLAB18 _id19=IDLAB19 _id20=
               IDLAB20) tip=(y x datalabel _tip1 _id2 _id3 _id4 _id5 _id6 _id7
               _id8 _id9 _id10 _id11 _id12 _id13 _id14 _id15 _id16 _id17 _id18
               _id19 _id20)

               group=idlab2 name='s'; *<==== add the group variable ====<<<<;

            vectorplot y=YVEC x=XVEC xorigin=0 yorigin=0 / datalabel=LABEL2VAR
               shaftprotected=false rolename=(_tip1=VNAME _tip2=VLABEL _tip3=
               YORI _tip4=XORI _tip5=LENGTH _tip6=LENGTH2) tip=(y x datalabel
               _tip1 _tip2 _tip3 _tip4 _tip5 _tip6) datalabelattrs=
               GRAPHVALUETEXT (color=GraphData2:ContrastColor) lineattrs=
               GRAPHDATA2 (pattern=solid) primary=true;

            discretelegend 's'; *<==== add a legend ====<<<<;

            if (0)
               entry "Vector Stretch = " STRETCH / autoalign=(topright topleft
                  bottomright bottomleft right left top bottom);
            endif;
         endlayout;
      endgraph;
   end;
run;
```

The following step creates the MDPREF plot with the make of the automobile added as a second ID variable and displayed in the graph as a group variable:

```
proc prinqual data=CarPref out=Results n=2 replace mdpref;
   title2 'Multidimensional Preference (MDPREF) Analysis';
   title3 'Optimal Monotonic Transformation of Preference Data';
   id model make;
   transform monotone('1'n-'25'n);
run;
```

The results are displayed in Output 96.1.5.

Output 96.1.5 Biplot with a Group Variable



The following step restores the default template:

```
proc template;
   delete Stat.Prinqual.Graphics.MDPref / store=sasuser.templat;
run;
```

Example 96.2: Principal Components of Basketball Rankings

The data in this example are 1985–1986 preseason rankings of 35 U.S. college basketball teams by 10 different news services. The services do not all rank the same teams or the same number of teams, so there are missing values in these data. Each of the 35 teams in the data set is ranked by at least one news service. One way of summarizing these data is with a principal component analysis, since the rankings should all be related to a single underlying variable, the first principal component.

You can use PROC PRINQUAL to estimate the missing ranks and compute scores for all observations. You can formulate a PROC PRINQUAL analysis that assumes that the observed ranks are ordinal variables and replaces the ranks with new numbers that are monotonic with the ranks and better fit the one principal component model. The missing rank estimates need to be constrained since a news service would have positioned the unranked teams below the teams it ranked. PROC PRINQUAL should impose order constraints within the nonmissing values and between the missing and nonmissing values, but not within the missing values. PROC PRINQUAL has sophisticated missing data handling facilities; however, these facilities cannot directly handle this problem. The solution requires reformulating the problem.

By performing some preliminary data manipulations, specifying the N=1 option in the PROC PRINQUAL statement, and specifying the UNTIE transformation in the TRANSFORM statement, you can make the missing value estimates conform to the requirements. The PROC MEANS step finds the largest rank for each variable. The next DATA step replaces missing values with a value that is one larger than the largest observed rank. The PROC PRINQUAL N=1 option specifies that the variables should be transformed to make them as one-dimensional as possible. The UNTIE transformation in the TRANSFORM statement monotonically transforms the ranks, untying any ties in an optimal way. Because the only ties are for the values that replace the missing values, and because these values are larger than the observed values, the rescoring of the data satisfies the preceding requirements.

The following statements create the data set and perform the transformations discussed previously. These statements produce Output 96.2.1 and Output 96.2.2.

```
*  Preseason 1985 College Basketball Rankings
*  (rankings of 35 teams by 10 news services)
*
*  Note:(a) Various news services rank varying numbers of teams.
*       (b) Not all 35 teams are ranked by all news services.
*       (c) Each team is ranked by at least one service.
*       (d) Rank 20 is missing for UPI.;

title1 '1985 Preseason College Basketball Rankings';

data bballm;
   input School $13. CSN DurhamSun DurhamHerald WashingtonPost
         USA_Today SportMagazine InsideSports UPI AP SportsIllustrated;
   label CSN               = 'Community Sports News (Chapel Hill, NC)'
         DurhamSun         = 'Durham Sun'
         DurhamHerald      = 'Durham Morning Herald'
         WashingtonPost    = 'Washington Post'
         USA_Today         = 'USA Today'
         SportMagazine     = 'Sport Magazine'
         InsideSports      = 'Inside Sports'
         UPI               = 'United Press International'
         AP                = 'Associated Press'
         SportsIllustrated = 'Sports Illustrated'
         ;
   format CSN--SportsIllustrated 5.1;
   datalines;
Louisville     1  8  1  9  8  9  6 10  9  9
Georgia Tech   2  2  4  3  1  1  1  2  1  1
Kansas         3  4  5  1  5 11  8  4  5  7
Michigan       4  5  9  4  2  5  3  1  3  2
Duke           5  6  7  5  4 10  4  5  6  5
UNC            6  1  2  2  3  4  2  3  2  3
Syracuse       7 10  6 11  6  6  5  6  4 10
Notre Dame     8 14 15 13 11 20 18 13 12  .
Kentucky       9 15 16 14 14 19 11 12 11 13
LSU           10  9 13  . 13 15 16  9 14  8
DePaul        11  . 21 15 20  . 19  .  . 19
Georgetown    12  7  8  6  9  2  9  8  8  4
Navy          13 20 23 10 18 13 15  . 20  .
Illinois      14  3  3  7  7  3 10  7  7  6
```

```
Iowa          15 16  .  . 23  .  . 14  . 20
Arkansas      16  .  .  . 25  .  .  .  . 16
Memphis State 17  . 11  . 16  8 20  . 15 12
Washington    18  .  .  .  .  .  . 17  .  .
UAB           19 13 10  . 12 17  . 16 16 15
UNLV          20 18 18 19 22  . 14 18 18  .
NC State      21 17 14 16 15  . 12 15 17 18
Maryland      22  .  .  . 19  .  .  . 19 14
Pittsburgh    23  .  .  .  .  .  .  .  .  .
Oklahoma      24 19 17 17 17 12 17  . 13 17
Indiana       25 12 20 18 21  .  .  .  .  .
Virginia      26  . 22  .  . 18  .  .  .  .
Old Dominion  27  .  .  .  .  .  .  .  .  .
Auburn        28 11 12  8 10  7  7 11 10 11
St. Johns     29  .  .  .  . 14  .  .  .  .
UCLA          30  .  .  .  .  .  . 19  .  .
St. Joseph's   .  . 19  .  .  .  .  .  .  .
Tennessee      .  . 24  .  . 16  .  .  .  .
Montana        .  .  . 20  .  .  .  .  .  .
Houston        .  .  .  . 24  .  .  .  .  .
Virginia Tech  .  .  .  .  .  . 13  .  .  .
;

* Find maximum rank for each news service and replace
* each missing value with the next highest rank.;

proc means data=bballm noprint;
   output out=maxrank
      max=mcsn mdurs mdurh mwas musa mspom mins mupi map mspoi;
run;

data bball;
   set bballm;
   if _n_=1 then set maxrank;
   array services[10] CSN--SportsIllustrated;
   array maxranks[10] mcsn--mspoi;
   keep School CSN--SportsIllustrated;
   do i=1 to 10;
      if services[i]=. then services[i]=maxranks[i]+1;
   end;
run;

* Assume that the ranks are ordinal and that unranked teams would have
* been ranked lower than ranked teams.  Monotonically transform all ranked
* teams while estimating the unranked teams.  Enforce the constraint that
* the missing ranks are estimated to be less than the observed ranks.
* Order the unranked teams optimally within this constraint.  Do this so
* as to maximize the variance accounted for by one linear combination.
* This makes the data as nearly rank one as possible, given the constraints.
*
* NOTE: The UNTIE transformation should be used with caution.
* It frequently produces degenerate results.;

ods graphics on;

proc prinqual data=bball out=tbball scores n=1 tstandard=z
```

```
   plots=transformations;
   title2 'Optimal Monotonic Transformation of Ranked Teams';
   title3 'with Constrained Estimation of Unranked Teams';
   transform untie(CSN -- SportsIllustrated);
   id School;
run;
```

Output 96.2.1 PRINQUAL Iteration History

1985 Preseason College Basketball Rankings
Optimal Monotonic Transformation of Ranked Teams
with Constrained Estimation of Unranked Teams

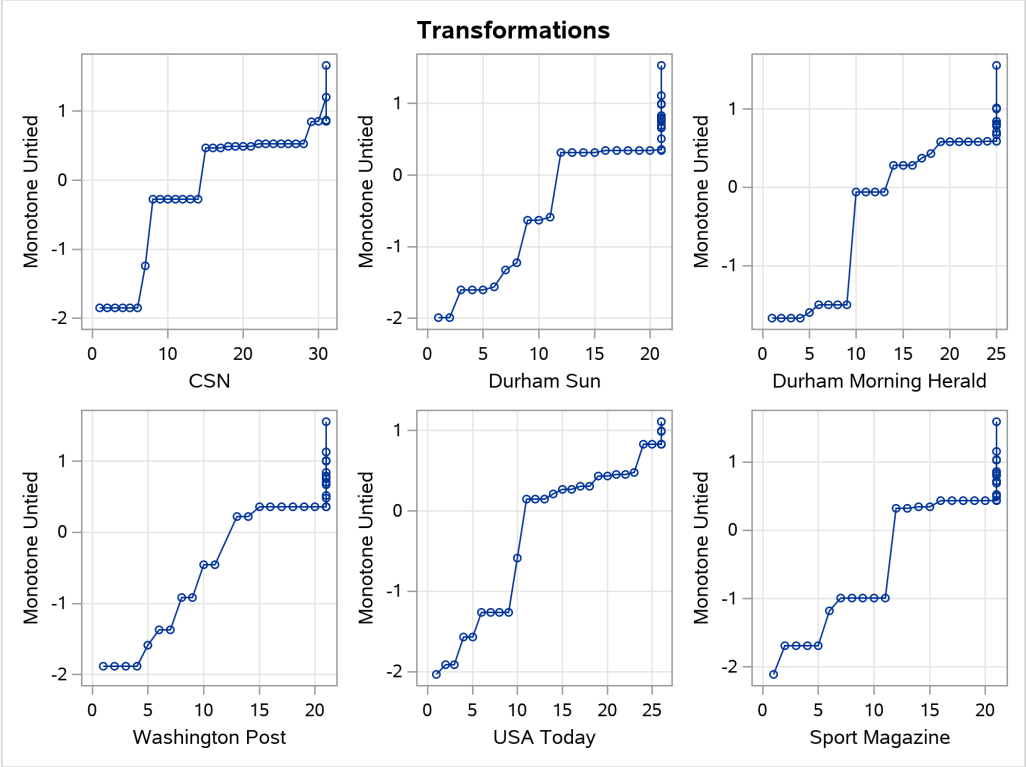
The PRINQUAL Procedure

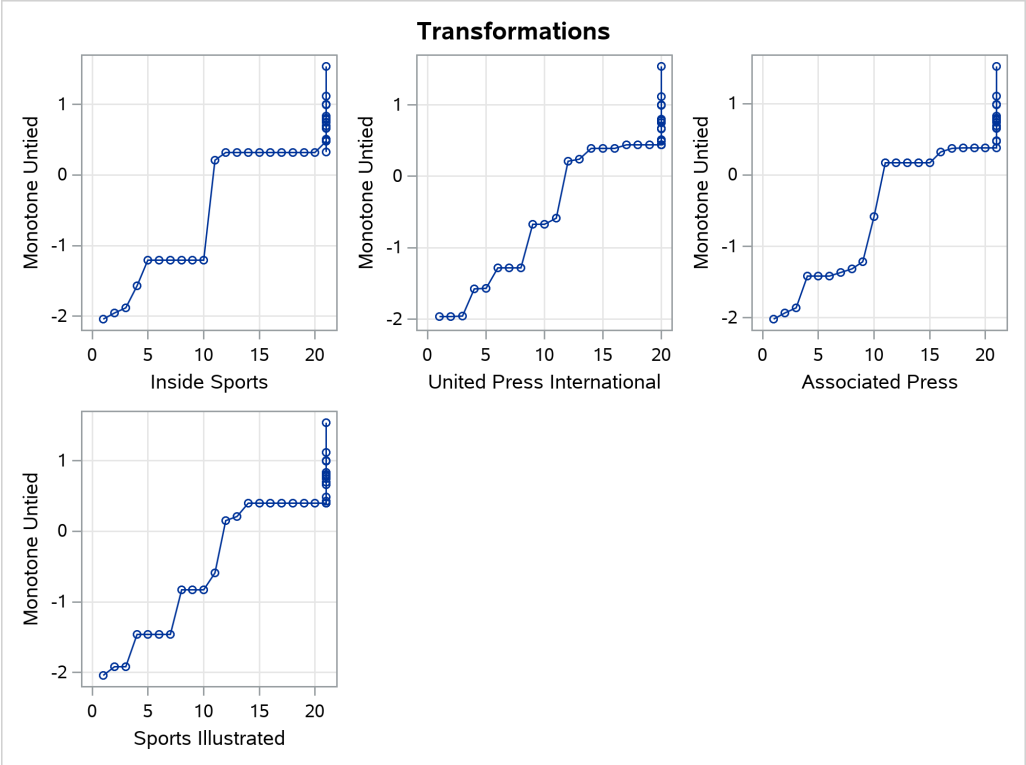
PRINQUAL MTV Algorithm Iteration History

Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.18563	0.76531	0.85850		
2	0.03225	0.14627	0.94362	0.08512	
3	0.02126	0.10530	0.94669	0.00307	
4	0.01467	0.07526	0.94801	0.00132	
5	0.01067	0.05282	0.94865	0.00064	
6	0.00800	0.03669	0.94899	0.00034	
7	0.00617	0.02862	0.94919	0.00020	
8	0.00486	0.02636	0.94932	0.00013	
9	0.00395	0.02453	0.94941	0.00009	
10	0.00327	0.02300	0.94947	0.00006	
11	0.00275	0.02166	0.94952	0.00005	
12	0.00236	0.02041	0.94956	0.00004	
13	0.00205	0.01927	0.94959	0.00003	
14	0.00181	0.01818	0.94962	0.00003	
15	0.00162	0.01719	0.94964	0.00002	
16	0.00147	0.01629	0.94966	0.00002	
17	0.00136	0.01546	0.94968	0.00002	
18	0.00128	0.01469	0.94970	0.00002	
19	0.00121	0.01398	0.94971	0.00001	
20	0.00115	0.01332	0.94973	0.00001	
21	0.00111	0.01271	0.94974	0.00001	
22	0.00105	0.01213	0.94975	0.00001	
23	0.00099	0.01155	0.94976	0.00001	
24	0.00095	0.01095	0.94977	0.00001	
25	0.00091	0.01038	0.94978	0.00001	
26	0.00088	0.00986	0.94978	0.00001	
27	0.00084	0.00936	0.94979	0.00001	
28	0.00081	0.00889	0.94980	0.00001	
29	0.00077	0.00846	0.94980	0.00000	
30	0.00073	0.00805	0.94980	0.00000	Not Converged

WARNING: Failed to converge, however criterion change is less than 0.0001.

Output 96.2.2 Transformations





An alternative approach is to use the pairwise deletion option of the CORR procedure to compute the correlation matrix and then use PROC PRINCOMP or PROC FACTOR to perform the principal component analysis. This approach has several disadvantages. The correlation matrix might not be positive semidefinite (PSD), an assumption required for principal component analysis. PROC PRINQUAL always produces a PSD correlation matrix. Even with pairwise deletion, PROC CORR removes the six observations that have only a single nonmissing value from this data set. Finally, it is still not possible to calculate scores on the principal components for those teams that have missing values.

You can compute the composite ranking by using PROC PRINCOMP and some preliminary data manipulations, similar to those discussed previously.

Chapter 95, "The PRINCOMP Procedure," contains an example where the average of the unused ranks in each poll is substituted for the missing values, and each observation is weighted by the number of nonmissing values. This method has much to recommend it. It is much faster and simpler than using PROC PRINQUAL. It is also much less prone to degeneracies and capitalization on chance. However, PROC PRINCOMP does not allow the nonmissing ranks to be monotonically transformed and the missing values untied to optimize fit.

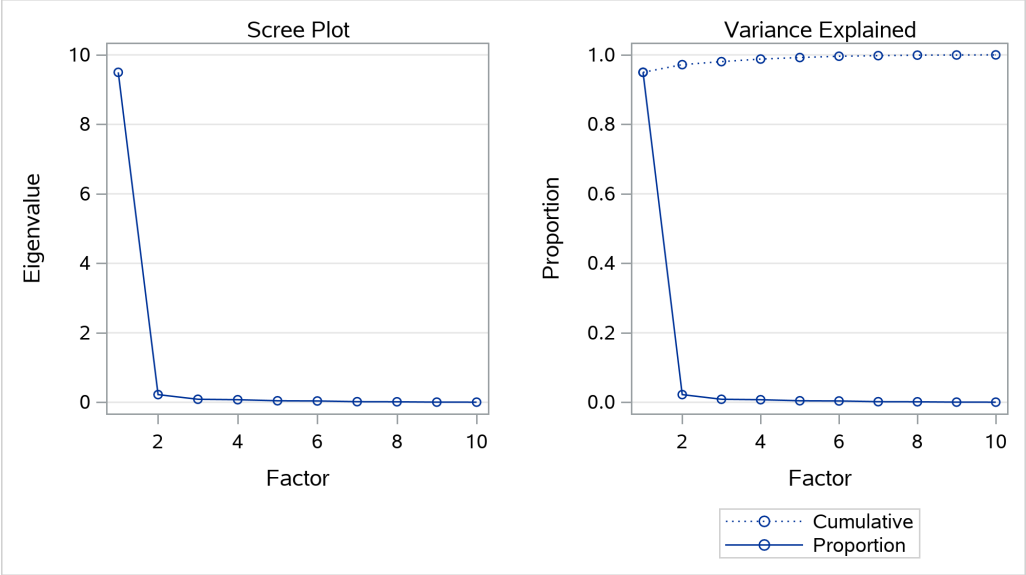
PROC PRINQUAL monotonically transforms the observed ranks and estimates the missing ranks (within the constraints given previously) to account for almost 95 percent of the variance of the transformed data by just one dimension. PROC FACTOR is then used to report details of the principal component analysis of the transformed data. As shown by the Factor Pattern values in Output 96.2.3, nine of the ten news services have a correlation of 0.95 or larger with the scores on the first principal component after the data are optimally transformed. The scores are sorted and the composite ranking is displayed following the PROC FACTOR output. More confidence can be placed in the stability of the scores for teams that are ranked by the majority of the news services than in scores for teams that are seldom ranked.

The monotonic transformations are plotted for each of the ten news services. See Output 96.2.2. These plots show the values of the raw ranks (with the missing ranks replaced by the maximum rank plus one) versus the rescored (transformed) ranks. The transformations are the step functions that maximize the fit of the data to the principal component model. Smoother transformations could be found by using MSPLINE transformations, but MSPLINE transformations would not correctly handle the missing data problem.

The following statements perform the final analysis and produce Output 96.2.3:

```
* Perform the Final Principal Component Analysis;
proc factor nfactors=1 plots=scree;
   title4 'Principal Component Analysis';
   ods select factorpattern screeplot;
   var TCSN -- TSportsIllustrated;
run;

proc sort;
   by Prin1;
run;

* Display Scores on the First Principal Component;
proc print;
   title4 'Teams Ordered by Scores on First Principal Component';
   var School Prin1;
run;
```

Output 96.2.3 Principal Components of College Basketball Rankings



Output 96.2.3 *continued*

Factor Pattern

		Factor1
TCSN	CSN Transformation	0.91136
TDurhamSun	DurhamSun Transformation	0.98887
TDurhamHerald	DurhamHerald Transformation	0.97402
TWashingtonPost	WashingtonPost Transformation	0.97408
TUSA_Today	USA_Today Transformation	0.98867
TSportMagazine	SportMagazine Transformation	0.95331
TInsideSports	InsideSports Transformation	0.98521
TUPI	UPI Transformation	0.98534
TAP	AP Transformation	0.99590
TSportsIllustrated	SportsIllustrated Transformation	0.98615

Output 96.2.3 *continued*

1985 Preseason College Basketball Rankings
Optimal Monotonic Transformation of Ranked Teams
with Constrained Estimation of Unranked Teams
Teams Ordered by Scores on First Principal Component

Obs	School	Prin1
1	Georgia Tech	-6.20315
2	UNC	-5.93314
3	Michigan	-5.71034
4	Kansas	-4.78699
5	Duke	-4.75896
6	Illinois	-4.19220
7	Georgetown	-4.02861
8	Louisville	-3.73087
9	Syracuse	-3.47497
10	Auburn	-1.78429
11	LSU	-0.35928
12	Memphis State	0.46737
13	Kentucky	0.63661
14	Notre Dame	0.71919
15	Navy	0.76187
16	UAB	0.98316
17	DePaul	1.09891
18	Oklahoma	1.12012
19	NC State	1.15144
20	UNLV	1.28766
21	Iowa	1.45260
22	Indiana	1.48123
23	Maryland	1.54935
24	Virginia	2.01385
25	Arkansas	2.02718
26	Washington	2.10878
27	Tennessee	2.27770
28	Virginia Tech	2.36103
29	St. Johns	2.37387
30	Montana	2.43502
31	UCLA	2.52481
32	Pittsburgh	3.00907
33	Old Dominion	3.03324
34	St. Joseph's	3.39259
35	Houston	4.69614

The ordinary PROC PRINQUAL missing data handling facilities do not work for these data because they do not constrain the missing data estimates properly. If you code the missing ranks as missing and specify linear transformations, then you can compute least squares estimates of the missing values without transforming the observed values. The first principal component then accounts for 92 percent of the variance after 20 iterations. However, Virginia Tech is ranked number 11 by its score even though it appeared in only one poll (Inside Sports ranked it number 13, anchoring it firmly in the middle). Specifying monotone transformations is also inappropriate since they too allow unranked teams to move in between ranked teams.

With these data, the combination of monotone transformations and the freedom to score the missing ranks without constraint leads to degenerate transformations. PROC PRINQUAL tries to merge the 35 points into two points, producing a perfect fit in one dimension. There is evidence for this after 20 iterations when the Average Change, Maximum Change, and Criterion Change values are all increasing, instead of the more stable decreasing change rate seen in the analysis shown. The change rates all stop increasing after 41 iterations, and it is clear by 70 or 80 iterations that one component will account for 100 percent of the transformed variables variance after sufficient iteration. While this might seem desirable (after all, it is a perfect fit), you should, in fact, be on guard when this happens. Whenever convergence is slow, the rates of change increase, or the final data perfectly fit the model, the solution is probably degenerating because of too few constraints on the scorings.

PROC PRINQUAL can account for 100 percent of the variance by scoring Montana and UCLA with one positive value on all variables and scoring all the other teams with one negative value on all variables. This inappropriate analysis suggests that all ranked teams are equally good except for two teams that are less good. Both of these two teams are ranked by only one news service, and their only nonmissing rank is last in the poll. This accounts for the degeneracy.

References

Carroll, J. D. (1972). "Individual Differences and Multidimensional Scaling." In *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences*, vol. 1, edited by R. N. Shepard, A. K. Romney, and S. B. Nerlove, 105–155. New York: Seminar Press.

De Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.

De Leeuw, J. (1985). Personal communication, Leiden, Netherlands.

De Leeuw, J. (1986). *Regression with Optimal Scaling of the Dependent Variable*. Leiden, Netherlands: Department of Data Theory, University of Leiden.

Eckart, C., and Young, G. (1936). "The Approximation of One Matrix by Another of Lower Rank." *Psychometrika* 1:211–218.

Fisher, R. A. (1938). *Statistical Methods for Research Workers*. 10th ed. Edinburgh: Oliver & Boyd.

Gabriel, K. R. (1981). "Biplot Display of Multivariate Matrices for Inspection of Data and Diagnosis." In *Interpreting Multivariate Data*, edited by V. Barnett, 571–572. Chichester, UK: John Wiley & Sons.

Gifi, A. (1990). *Nonlinear Multivariate Analysis*. New York: John Wiley & Sons.

Goodnight, J. H. (1978). *The SWEEP Operator: Its Importance in Statistical Computing*. Technical Report R-106, SAS Institute Inc., Cary, NC.

Hotelling, H. (1933). "Analysis of a Complex of Statistical Variables into Principal Components." *Journal of Educational Psychology* 24:417–441, 498–520.

Kruskal, J. B. (1964). "Nonmetric Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis." *Psychometrika* 29:1–27.

Kruskal, J. B., and Shepard, R. N. (1974). "A Nonmetric Variety of Linear Factor Analysis." *Psychometrika* 38:123–157.

Sarle, W. S. (1984). Personal communication, Cary, NC.

Siegel, S. (1956). *Nonparametric Statistics*. New York: McGraw-Hill.

Smith, P. L. (1979). "Splines as a Useful and Convenient Statistical Tool." *American Statistician* 33:57–62.

Tenenhaus, M., and Vachette, J. L. (1977). "PRINQUAL: Un programme d'analyse en composantes principales d'un ensemble de variables nominales ou numériques." *Les Cahiers de Recherche* 68. CESA, Jouy-en-Josas, France.

Van Rijckevorsel, J. L. (1982). "Canonical Analysis with B-Splines." In *COMPUSTAT 1982, Part I*, edited by H. Caussinus, P. Ettinger, and R. Tomassone, 393–398. Vienna: Physica-Verlag.

Winsberg, S., and Ramsay, J. O. (1983). "Monotone Spline Transformations for Dimension Reduction." *Psychometrika* 48:575–595.

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

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

Subject Index

biplot
    PRINQUAL procedure, 7973

casewise deletion
    PRINQUAL procedure, 7944
character OPSCORE variables
    PRINQUAL procedure, 7968
constant transformations
    avoiding (PRINQUAL), 7968
constant variables
    PRINQUAL procedure, 7968

excluded observations
    PRINQUAL procedure, 7948, 7969

frequency variable
    PRINQUAL procedure, 7949

identity transformation
    PRINQUAL procedure, 7953
initialization
    random (PRINQUAL), 7968
iterations
    PRINQUAL procedure, 7963
    restarting (PRINQUAL), 7946, 7968

knots
    PRINQUAL procedure, 7955, 7956

linear transformation
    PRINQUAL procedure, 7952

MAC method
    PRINQUAL procedure, 7958, 7964
maximum average correlation method
    PRINQUAL procedure, 7958, 7964
maximum total variance method
    PRINQUAL procedure, 7957
MDPREF analysis
    PRINQUAL procedure, 7973
MGV method
    PRINQUAL procedure, 7957
minimum generalized variance method
    PRINQUAL procedure, 7957
missing values
    character (PRINQUAL), 7952
    PRINQUAL procedure, 7944, 7963, 7969
monotonic
    transformation (PRINQUAL), 7952, 7953
    transformation, B-spline (PRINQUAL), 7952
MTV method
    PRINQUAL procedure, 7957
multidimensional preference analysis
    PRINQUAL procedure, 7973

nonoptimal transformations
    PRINQUAL procedure, 7951

ODS graph names
    PRINQUAL procedure, 7972
optimal
    scoring (PRINQUAL), 7952
    transformations (PRINQUAL), 7952
output data sets
    PRINQUAL procedure, 7965
output table names
    PRINQUAL procedure, 7972

passive observations
    PRINQUAL procedure, 7969
principal component analysis
    PRINQUAL procedure, 7964
PRINQUAL procedure
    biplot, 7973
    casewise deletion, 7944
    character OPSCORE variables, 7968
    constant transformations, avoiding, 7968
    constant variables, 7968
    excluded observations, 7948, 7969
    frequency variable, 7949
    identity transformation, 7953
    iterations, 7946, 7963, 7968
    knots, 7955, 7956
    linear transformation, 7952
    MAC method, 7958, 7964
    maximum average correlation method, 7958, 7964
    maximum total variance method, 7957
    MDPREF analysis, 7973
    MGV method, 7957
    minimum generalized variance method, 7957
    missing character values, 7952
    missing values, 7944, 7963, 7969
    monotonic B-spline transformation, 7952
    monotonic transformation, 7952, 7953
    MTV method, 7957
    multidimensional preference analysis, 7973
    nonoptimal transformations, 7951

ODS graph names, 7972
optimal scoring, 7952
optimal transformations, 7952
output data sets, 7965
output table names, 7972
passive observations, 7969
principal component analysis, 7964
random initializations, 7968
reflecting the transformation, 7956
renaming variables, 7956
reusing variables, 7956
smoothing spline transformation, 7953
spline t-options, 7955
spline transformation, 7953
standardization, 7967
transformation options, 7953
variable names, 7967
weight variable, 7957

reflecting the transformation
PRINQUAL procedure, 7956
renaming and reusing variables
PRINQUAL procedure, 7956

smoothing spline transformation
PRINQUAL procedure, 7953
spline t-options
PRINQUAL procedure, 7955
spline transformation
PRINQUAL procedure, 7953

transformation options
PRINQUAL procedure, 7953

variables
frequency (PRINQUAL), 7949
renaming (PRINQUAL), 7956
reusing (PRINQUAL), 7956
weight (PRINQUAL), 7957

weight variable
PRINQUAL procedure, 7957

Syntax Index

APPROXIMATIONS option
    PROC PRINQUAL statement, 7941
APREFIX= option
    PROC PRINQUAL statement, 7941
ARSIN transformation
    TRANSFORM statement (PRINQUAL), 7951

BY statement
    PRINQUAL procedure, 7948

CCONVERGE= option
    PROC PRINQUAL statement, 7942
CHANGE= option
    PROC PRINQUAL statement, 7942
CONVERGE= option
    PROC PRINQUAL statement, 7942
CORRELATIONS option
    PROC PRINQUAL statement, 7942
COVARIANCE option
    PROC PRINQUAL statement, 7942

DATA= option
    PROC PRINQUAL statement, 7942
DEGREE= option
    TRANSFORM statement (PRINQUAL), 7955
DUMMY option
    PROC PRINQUAL statement, 7942

EVENLY option
    TRANSFORM statement (PRINQUAL), 7955
EXP transformation
    TRANSFORM statement (PRINQUAL), 7951

FREQ statement
    PRINQUAL procedure, 7949

ID statement
    PRINQUAL procedure, 7949
IDENTITY transformation
    TRANSFORM statement (PRINQUAL), 7953
INITITER= option
    PROC PRINQUAL statement, 7943

KNOTS= option
    TRANSFORM statement (PRINQUAL), 7955

LINEAR transformation
    TRANSFORM statement (PRINQUAL), 7952
LOG transformation
    TRANSFORM statement (PRINQUAL), 7951
LOGIT transformation
    TRANSFORM statement (PRINQUAL), 7951

MAXITER= option
    PROC PRINQUAL statement, 7943
MDPREF= option
    PROC PRINQUAL statement, 7943
METHOD= option
    PROC PRINQUAL statement, 7943
MONOTONE transformation
    TRANSFORM statement (PRINQUAL), 7952
MONOTONE= option
    PROC PRINQUAL statement, 7943
MSPLINE transformation
    TRANSFORM statement (PRINQUAL), 7952

N= option
    PROC PRINQUAL statement, 7943
NAME= option
    TRANSFORM statement (PRINQUAL), 7956
NKNOTS= option
    TRANSFORM statement (PRINQUAL), 7956
NOCHECK option
    PROC PRINQUAL statement, 7944
NOMISS option
    PROC PRINQUAL statement, 7944
NOPRINT option
    PROC PRINQUAL statement, 7944

OPSCORE transformation
    TRANSFORM statement (PRINQUAL), 7952
ORIGINAL option
    TRANSFORM statement (PRINQUAL), 7954
OUT= option
    PROC PRINQUAL statement, 7944

PARAMETER= option
    TRANSFORM statement (PRINQUAL), 7954
PLOTS= option
    PROC PRINQUAL statement, 7945
POWER transformation
    TRANSFORM statement (PRINQUAL), 7951
PREFIX= option
    PROC PRINQUAL statement, 7946
PRINQUAL procedure
    syntax, 7940
PRINQUAL procedure, BY statement, 7948
PRINQUAL procedure, FREQ statement, 7949
PRINQUAL procedure, ID statement, 7949

PRINQUAL procedure, PROC PRINQUAL statement, 7940
APPROXIMATIONS option, 7941
APREFIX= option, 7941
CCONVERGE= option, 7942
CHANGE= option, 7942
CONVERGE= option, 7942
CORRELATIONS option, 7942
COVARIANCE option, 7942
DATA= option, 7942
DUMMY option, 7942
INITITER= option, 7943
MAXITER= option, 7943
MDPREF= option, 7943
METHOD= option, 7943
MONOTONE= option, 7943
N= option, 7943
NOCHECK option, 7944
NOMISS option, 7944
NOPRINT option, 7944
OUT= option, 7944
PLOTS= option, 7945
PREFIX= option, 7946
REFRESH= option, 7946
REITERATE option, 7946
REPLACE option, 7947
SCORES option, 7947
SINGULAR= option, 7947
STANDARD option, 7947
TPREFIX= option, 7947
TSTANDARD= option, 7947
TYPE= option, 7948
UNTIE= option, 7948
PRINQUAL procedure, TRANSFORM statement
ARSIN transformation, 7951
DEGREE= option, 7955
EVENLY option, 7955
EXP transformation, 7951
IDENTITY transformation, 7953
KNOTS= option, 7955
LINEAR transformation, 7952
LOG transformation, 7951
LOGIT transformation, 7951
MONOTONE transformation, 7952
MSPLINE transformation, 7952
NAME= option, 7956
NKNOTS= option, 7956
OPSCORE transformation, 7952
ORIGINAL option, 7954
PARAMETER= option, 7954
POWER transformation, 7951
RANK transformation, 7952
REFLECT option, 7956
SM= option, 7955
SPLINE transformation, 7953
SSPLINE transformation, 7953
TSTANDARD= option, 7956
UNTIE transformation, 7953
PRINQUAL procedure, WEIGHT statement, 7957
PROC PRINQUAL statement, *see* PRINQUAL procedure

RANK transformation
TRANSFORM statement (PRINQUAL), 7952
REFLECT option
TRANSFORM statement, 7956
REFRESH= option
PROC PRINQUAL statement, 7946
REITERATE option
PROC PRINQUAL statement, 7946
REPLACE option
PROC PRINQUAL statement, 7947

SCORES option
PROC PRINQUAL statement, 7947
SINGULAR= option
PROC PRINQUAL statement, 7947
SM= option
TRANSFORM statement, 7955
SPLINE transformation
TRANSFORM statement (PRINQUAL), 7953
SSPLINE transformation
TRANSFORM statement (PRINQUAL), 7953
STANDARD option
PROC PRINQUAL statement, 7947

TPREFIX= option
PROC PRINQUAL statement, 7947
TRANSFORM statement (PRINQUAL)
ARSIN transformation, 7951
DEGREE= option, 7955
EVENLY option, 7955
EXP transformation, 7951
IDENTITY transformation, 7953
KNOTS= option, 7955
LINEAR transformation, 7952
LOG transformation, 7951
LOGIT transformation, 7951
MONOTONE transformation, 7952
MSPLINE transformation, 7952
NAME= option, 7956
NKNOTS= option, 7956
OPSCORE transformation, 7952
ORIGINAL option, 7954
PARAMETER= option, 7954
POWER transformation, 7951
RANK transformation, 7952
SPLINE transformation, 7953
SSPLINE transformation, 7953

TSTANDARD= option, 7956
UNTIE transformation, 7953
TSTANDARD= option
PROC PRINQUAL statement, 7947
TRANSFORM statement (PRINQUAL), 7956
TYPE= option
PROC PRINQUAL statement, 7948

UNTIE transformation
TRANSFORM statement (PRINQUAL), 7953
UNTIE= option
PROC PRINQUAL statement, 7948

WEIGHT statement
PRINQUAL procedure, 7957