Some Methods in Multivariate Exploratory Data Analysis

Warren F. Kuhfeld and Forrest W. Young
The L. L. Thurstone Psychometric Laboratory
The University of North Carolina at Chapel Hill

Abstract

SAS Institute is funding a software research and development project at UNC which develops new statistical procedures. This paper describes experimental version 6 software developed at the University of North Carolina (UNC), under contract from SAS Institute Inc., Cary, NC 27511 USA. The CORRESP, PRINQUAL, and TRANSREG procedures were developed by Forrest Young and Warren Kuhfeld, and programmed by Warren Kuhfeld. Warren Kuhfeld is visiting the Department of Data Theory, Leiden University, Middelsteegstraat 4, 2312 TW Leiden, The Netherlands, until April 18, 1987, and after that may be reached at SAS Institute Inc., PO Box 8000, Cary, NC 27511 USA. The authors wish to thank the faculty of the Department of Data Theory for their support during the first author’s visit. Forrest Young may be reached at The L. L. Thurstone Psychometric Laboratory, Davie Hall 013 - A, The University of North Carolina, Chapel Hill NC 27514 USA. If you wish to test these procedures, contact Forrest Young. Design suggestions and comments are welcome.

SAS is a registered trademark of SAS Institute Inc., Cary NC USA.

canonical correlation analysis; and response surface regression. PROC PRINQUAL finds transformations that optimize properties of covariance matrices, including the fit of the data to a principal components model. PROC CORRESP performs simple and multiple correspondence analysis. These procedures add powerful new capabilities for multivariate exploratory data analysis to the SAS system.

The CORRESP Procedure

The CORRESP procedure (Young and Kuhfeld, 1986) performs correspondence analysis (Lebart, Morineau, and Warwick, 1984; Greenacre, 1984). PROC CORRESP can also be used to perform small multiple correspondence analyses. Input may be raw category responses on two or more classification variables, category frequency data on two or more classification variables, a binary category response indicator matrix, or a two-way contingency table. Burt tables may be directly input or created from categorical variables. Input to PROC CORRESP may come directly from PROC FREQ. Supplemental variables and observations may also be input. PROC CORRESP can produce an output data set that may be used for plotting by procedures such as VISUALS, IDPLOT, GPLOT, G3D or PLOT.

The statements available in PROC CORRESP are:

PROC CORRESP [options];

TABLES variables [ , variables];

VAR variables;

WEIGHT variable;

SUPVAR variables;

ID variable;

BY variables;

The PROC CORRESP statement has a number of options. The data set options can be used to: name the input SAS data set; create an output coordinate SAS data set that contains the row, column, supplemental observation, and supplemental variable coordinates; and create an output frequency SAS data set that contains the contingency table, row and column profiles, chi-square expected values, deviations from chi-squared expected values, and contributions to the total chi-square.
The printing options control what is printed. The printed output may contain: Euclidean coordinates for both the supplemental and active rows and columns, the singular values, the total inertia of the data matrix, the principal inertias of each dimension, a horizontal bar chart of the principal inertias, the cell, row, column and total chi-square contributions, the chi-square expected and observed minus expected values, the mass and inertia of each point, the squared cosines, and the partial contributions of each coordinate to inertia. Frequency and probability information may be printed in natural units, or in percents, or both.

The coordinate options can be used to specify the number of dimensions to display, specify how the coordinates are to be normalized (more will be said on this later), and to request a multiple correspondence analysis.

The table construction options can be used to specify how table categories are to be constructed - by factorially combining individual variable categories, or by creating a Burt table. Also, these options are used to control how missing values are handled. There are two separate forms of input to PROC CORRESP. One form is specified by the TABLES statement, the other by the VAR statement.

The TABLES statement is used to create a table from categorical data. A WEIGHT statement may be used to specify category frequencies. Supplemental observations and variables may be entered with TABLES input by including a WEIGHT variable with negative frequencies. With TABLES input, the rows and columns of the contingency table are constructed by either concatenating or crossing the values of the variables. When there is just one variable in a list, the categories of the variable form the rows (or columns) of the contingency table. When there are several variables in the list, the rows (or columns) of the contingency table are formed by either factorially combining or concatenating the categories of the variables. For example, consider three variables, SEX with values 'male' and 'female', AGE with values 'young' and 'old', and RACE with values 'black', 'white', and 'other'.

PROC CORRESP;
   TABLES SEX,AGE;
creates a table with two rows ('female' and 'male') and two columns ('old' and 'young').

PROC CORRESP CROSS=ROW;
   TABLES RACE SEX,SEX AGE;
creates a table with six rows ('black * female', 'black * male', 'other * female', 'other * male', 'white * female', and 'white * male') and four columns ('female', 'male', 'old', and 'young').

PROC CORRESP MCA;
   TABLES RACE AGE SEX;
creates a Burt Table and performs a multiple correspondence analysis. Both the rows and the columns have the same with seven categories ('black', 'other', 'white', 'old', 'young', 'female', and 'male').

With the VAR statement, the contingency table rows correspond to the observations of the DATA= data set and the contingency table columns correspond to the variables in the DATA= data set that are listed on the VAR statement. The entries in the DATA= data set are the frequencies in the contingency table. Thus, the VAR statement lets you input a two-way contingency table directly. Supplemental observations may be entered by including a WEIGHT statement with negative weights. Supplemental variables may be entered by including a SUPVAR statement.

PROC CORRESP Coordinate Computations

Let N be the contingency table formed from those observations and variables that are not supplemental. This table is a D_r by D_c (rank r) matrix of nonnegative numbers with nonzero row and column sums. Define P = (1/f)N where f is the total frequency (grand sum) of the elements of N. Let I be a vector of ones of the appropriate dimension, diag() be a matrix valued function that creates a diagonal matrix from a vector. Let r = P1, c = P'1, D_r = diag(r), and D_c = diag(c). The generalized singular value decomposition of P is P = AD_rB where A'D_r^-1A = B'D_c^-1B = I, where I is an identity matrix. A is the matrix of left singular vectors, having r rows and c columns; D_r is a diagonal matrix of singular values, having c rows and columns; and B is a rectangular matrix of right singular vectors having d_c rows and r columns. The first (trivial) column of A and B, and the first singular value are discarded before any results are displayed.

The PROFILE=, ROWS=, and COLUMNS= options are used to standardize the coordinates before they are printed and placed in the output data set. There are four choices of row coordinates. Specify: ROWS=A for row coordinates A,
There are several combinations of ROWS= and COLUMNS= of special interest. The default normalization is ROWS=DAD and COLUMNS=DBD. The PROFILE= option can be used as a short-hand notation for several of the combinations. When PROFILE=ROWS (ROWS=DAD and COLUMNS=B), the row coordinates $D_{-1}AD$, and column coordinates $B$, are a decomposition of the variable partition of the table. Similarly, when PROFILE=COLUMNS (COLUMNS=DBD and ROWS=A), the column coordinates $D_{-1}BD$, and row coordinates $A$, are a decomposition of the supplemental observations, and the column coordinates are weighted centroids of the row coordinates. Similarly, when ROWS=DA and COLUMNS=DBD, the column coordinates are weighted centroids of the row coordinates.

The formulas that are used to compute coordinates for the supplemental rows and columns also depend on the PROFILE= or ROWS= and COLUMNS= options. Let $S_{\nu}$ be the matrix whose rows contain the supplemental observations, and $S_{\nu}$ be a matrix whose rows contain the supplemental variables. Note that $S_{\nu}$ is the transpose of the supplemental variable partition of the table. Let $R_{\nu} = diag(S_{\nu}v) = 1$ be the supplemental observation profile matrix and $C_{\nu} = diag(S_{\nu}) = 1$ be the supplemental variable profile matrix. When ROWS=DAD, the coordinates for the supplemental observations are $R_{-1}AD$, ROWS=DA means $R_{-1}BD$, and ROWS=AD means $R_{-1}BD$. Similarly, when COLUMNS=DBD, the supplemental column coordinates are $C_{-1}BD$, COLUMNS=DB means $C_{-1}BD$, and COLUMNS=BD means $C_{-1}BD$. The quality mass inertia, squared cosines, and contributions to inertia are not dependent on the values specified for the ROWS=, COLUMNS=, and PROFILE= options. Let $sq()$ be a matrix valued function denoting element-wise squaring of the argument matrix. The partial contributions to inertia for the row points are $D_{-1}sq(A)$. The partial contributions to inertia for the column points are $D_{-1}sq(B)$. The squared cosines for the row points are $diag(sq(AD_{-1}))^{-1}sq(AD_{-1})$ and the squared cosines for the column points are $diag(sq(BD_{-1}))^{-1}sq(BD_{-1})$. The mass of the row points is $r$ and the mass of the column points is $c$. Let $t$ be the total inertia (the sum of the elements in $D_{-1}$). The inertia for the row points is $(1/t)D_{-1}sq(AD_{-1})$ and the inertia for the column points is $(1/t)D_{-1}sq(BD_{-1})$. The quality of a row or column point is the sum of its squared cosines, over only as many dimensions as are being kept.

The squared cosines for the supplemental observations are defined as $diag(sq(1c' - R_{\nu}D_{-1}^{-1})^{-1}sq(R_{\nu}D_{-1}^{-1}B)$. The squared cosines for supplemental variables are $diag(sq(1r' - C_{\nu}D_{-1}^{-1})^{-1}sq(C_{\nu}D_{-1}^{-1}A)$. The quality for a supplemental point is the sum of its squared cosines. Inertia and mass are not defined for supplemental points.

PROC CORRESP can be used to perform a multiple correspondence analysis when the contingency table is a Burt table by specifying the MCA option. This option simply does two things. It causes the observed singular values to be replaced by their square roots before they are used or reported, and it suppresses the printing and output of all row information. The resulting inertias and column coordinates are the appropriate inertias and coordinates for a multiple correspondence analysis. The output is equivalent to the column results of a simple correspondence analysis of the design matrix whose inner product is the Burt table.

The TRANSREG Procedure

PROC TRANSREG (Young and Kuhfeld, 1987) obtains linear and nonlinear transformations of variables to optimize the least-squares fit of the data to a variety of linear models including: simple, multiple, and multivariate regression; ANOVA models such as simple and generalized conjoint analysis; vector and ideal point preference regression (external unfolding); redundancy analysis; canonical correlation analysis; and response surface regression.

PROC TRANSREG is invoked by the following statements:

PROC TRANSREG options;

MODEL [t1(dependents) ... ] = [t2(dependents) ... ] [t3(independents) ... ];
POLYNOMIAL variables [/ options ];
ID variables;
BY variables;

The independent and dependent variables may be categorical, ordinal or quantitative. Any mix is allowed. Categorical variables may be transformed by scoring the categories to minimize least-squares error, or may be expanded into dummy variables. Ordinal variables may be transformed monotonically by scoring the ordered categories so that order is preserved and least-squares error is minimized. Ties may be optimally untied or left tied. Ordinal variables may instead be transformed to ranks. Quantitative variables may be smoothly transformed using splines or monotone spline transformations, or may be linearly transformed. Missing data may be estimated without constraint, with category constraints, and with order constraints.

PROC TRANSREG is a scoring procedure. It generates very little printed output. It produces an output data set which contains the transformed (optimally scored) variables. These variables may be input to PROCs REG, CANCORR, ANOVA or GLM to obtain final analyses; or input to a plotting procedure for obtaining preference biplots, transformations plots, and so on.

PROC TRANSREG is an alternating least-squares optimal scaling procedure. This means its algorithms are convergent. At every step, the algorithms decrease least-squares error, and hence maximize (depending on which iterative algorithm is specified) a squared multiple correlation (UNIVARIATE and MORALS), the average of squared multiple correlations (REDUNDANCY), or the of average squared canonical correlations (CANALS).

For more background on alternating least-squares optimal scaling methods and transformation regression methods see Young, de Leeuw, and Takane (1976); Gifi (1981), van der Burg and de Leeuw (1983), Young (1981), Israels (1984), de Leeuw (1986), de Leeuw and Kuhfeld (in press), Spelman and Friedman (1985), Hastie and Tibshirani (1986), and Schiffman, Reynolds, and Young (1981), to name a few of the many relevant works. Also see the PROC TRANSREG documentation (Young and Kuhfeld, 1987). The PROC TRANSREG variable expansions and least-squares transformation families will be described first, then the iterative algorithms will be described.

PROC TRANSREG Variable Expansions

The following are ways in which variables may be preprocessed prior to the start of the iterative algorithms. The variable expansions either replace the original variable with one (RANK) or more (CLASS) new columns, or add new variables (POINT, EPOINT, and QPOINT) to the same set (independent or dependent variables) as the originals. The POINT, EPOINT and QPOINT functions are used in preference (PREFMAP or external unfolding, Carroll, 1972) analyses, for ideal point regression analyses, and for response surface regressions. POINT, EPOINT, and QPOINT, create circular, elliptical, and quadratic response or preference surfaces. These variables are not transformed by the iterative algorithms after the initial preprocessing (except for a linear scaling). Observations with missing values for these types of variables are excluded from the analysis.

POINT - names those continuous, numeric variables that are used for a circular response surface regression or circular ideal point regression. POINT creates a new variable whose value for each observation is the sum of squares of all the POINT variables. This new variable is added to the set of variables and is used in the regression analysis.

EPOINT - names those continuous, numeric variables that are used for an elliptical response surface regression or elliptical ideal point regression. EPOINT creates a new variable whose value for each observation is the sum of squares of all the EPOINT variables. The value of each new variable is the square of each observed value for the corresponding original variable. Both sets of variables (original and squared) are then used in the regression analysis.

QPOINT - names those continuous, numeric variables that are used for a quadratic response surface regression or quadratic ideal point regression. QPOINT creates a set of new variables by crossing the QPOINT variables. If there are m QPOINT variables, the m(m + 1)/2 unique pairs of variable 1 times (element-wise product) variable j (where j may equal i) are created. Both sets (original and crossed) are combined to be used in the regression analysis.

CLASS - names those character or numeric, discrete variables that are classification variables. When a variable is named as a classification variable, it is expanded to dummy variables. Up to the first eight characters of the formatted variable's name are used to determine class membership.

RANK - names those numeric variables that are to be transformed to ranks before the analysis is begun. Ranks are averaged within ties.
Variable Transformation Families

The following are ways in which variables may be iteratively transformed. All variable transformations can be used with both the independent and dependent variables. Missing values for these types of variables may be optimally estimated.

**OPSCORE** - names those character or numeric, discrete variables that are to be optimally scored. OPSCORE assigns scores to each class (level) of the variable. Fisher's (1938) optimal scoring method is used.

**MONOTONE** - names those numeric, usually discrete, variables that are to be transformed monotonically with the restriction that ties are preserved. The Kruskal (1964) secondary least-squares monotonic transformation method is used. This transformation weakly preserves order and category membership (ties).

**UNTIE** - names those numeric, usually discrete, variables that are to be transformed monotonically without the restriction that ties be preserved. The Kruskal (1964) primary least-squares monotonic transformation method is used. This transformation weakly preserves order but not category membership (it unties tied values).

**LINEAR** - names those numeric variables that are subject to a linear transformation - change of origin and scale only.

**SPLINE** - names those numeric variables that are subject to a piece-wise polynomial B-spline (de Boor, 1978) transformation. By default a cubic polynomial transformation is used. Knots and other degrees may be specified.

**ISPLINE** - names those numeric, continuous variables that are subject to a B-spline transformation with monotonically increasing coefficients. By default a quadratic polynomial is used. Knots and other degrees may be specified.

PROC TRANSREG Expansion and Transformation Name Usage

The expansion and transformation names for a list of variables are specified on a MODEL statement. There are many typical MODEL statement forms. Any mix of the OPSCORE, LINEAR, MONOTONE, and UNTIE functions can be used to fit a mixed measurement level model. Often times, only one function will be specified for all independent variables, and that function will be one of the following: LINEAR, CLASS, POINT, OPPOINT, or EPOINT. In these cases the independent variables are not transformed. Rather, they are expanded and then remain unchanged throughout the analysis. For example in a regression algorithm:

MODEL LINEAR(Y1-Y5) = LINEAR(X1-X4);
performs an ordinary multivariate multiple regression analysis,

MODEL MONOTONE(Y1-Y2) OPSCORE(Y3-Y5) = LINEAR(X1) UNTIE(X2) MONOTONE(X3-X4);
performs a mixed measurement level multiple regression analysis, and

MODEL MONOTONE(Y1) = CLASS(X1-X2);
performs a simple conjoint analysis.

The PROC TRANSREG UNIVARIATE Algorithm

METHOD=UNIVARIATE (based on Young, de Leeuw, and Takane, 1976) specifies that each dependent variable be transformed to maximize the squared multiple correlation while the independent variables are not transformed. The METHOD=UNIVARIATE algorithm is a generalization of the ordinary univariate general linear model to allow for dependent variable transformations. This algorithm is used for conjoint analysis, metric and nonmetric PREFMAP and external unfolding analyses, and multiple regression analyses with dependent variable transformations.

The PROC TRANSREG MORALS Algorithm

METHOD=MORALS (Multiple Optimal Regression using Alternating Least Squares based on Young, de Leeuw, and Takane, 1976) specifies that each dependent variable be transformed along with the set of independent variables to maximize the squared multiple correlation. The METHOD=MORALS algorithm is a univariate generalization of the METHOD=UNIVARIATE algorithm to allow for both dependent and independent variable transformations. While more than one (say m) dependent variables may be specified on the MODEL statement, METHOD=MORALS is a univariate generalization because m linear models are fit, with each linear model containing only one dependent variable at a time.

The PROC TRANSREG REDUNDANCY Algorithm

METHOD=REDUNDANCY (an extension of Young, de Leeuw, and Takane, 1976) specifies that all dependent variables
and independent variables be jointly transformed to maximize the average of the squared multiple correlations. The METHOD=REDUNDANCY algorithm is a multivariate generalization of the METHOD=UNIVARIATE algorithm to allow for both dependent and independent variable transformations. One linear model is fit which transforms each variable only once, maximizing the average squared multiple correlation.

The PROC TRANSREG CANALS Algorithm

METHOD=CANALS (C ANonical correlation analysis with Alternating Least Squares based on the CANALS method of van der Burg and de Leeuw, 1983) specifies that all dependent variables and independent variables be jointly transformed to maximize the average of the first r canonical correlations. Like METHOD=REDUNDANCY, METHOD=CANALS is a multivariate method which transforms all of the variables together, finding only one transformation of each variable.

PROC TRANSREG Options

There are a number of options available in PROC TRANSREG. A very brief description of the options will be provided here. The data set options are: DATA=SASdata set and OUT=SASdata set.

The options that control the iterative algorithm are: METHOD=name specifies the iterative algorithm, MAXITER=n specifies the maximum number of iterations, CONVERGE=n specifies convergence criterion, NCAN=n specifies the number of canonical variables, and SINGULAR=n specifies the singularity criterion.

The output data set score partition options control what is placed in the output data set. The options are: REPLACE specifies independent and dependent variables are replaced by their transformed values, IREPLACE specifies independent variables are replaced by their transformed values, DREPLACE specifies dependent variables are replaced by their transformed values, APPROXIMATIONS specifies that both dependent and independent variable approximations are output, IAPPROXIMATIONS specifies that independent variable approximations are output, DAPPROXIMATIONS specifies that dependent variable approximations are output, and NOSCORES specifies that no score partition information be output. Both the output data set coefficient partition options are: COEFFICIENTS specifies that the linear model coefficients be included in the output data, COORDINATES specifies that external unfolding ideal point coordinates be output, MEANS specifies that marginal means for CLASS variable columns be output, UTILITIES specifies that conjoint utilities be output, MRC specifies that multiple regression coefficients be output, MFC specifies that point model ideal point coordinates be output, MEC specifies that elliptical point model ideal point coordinates be output, MOC specifies that quadratic point model ideal point coordinates be output, CCC specifies that canonical coefficients be output, CPC specifies that canonical point model ideal point coordinates be output, CEC specifies that canonical elliptical point model ideal point coordinates be output, and CDC specifies that canonical quadratic point model ideal point coordinates be output.

The output data set variable name prefixes provide prefixes for naming the transformed and approximation variables in the output data set. TDPREFIX=name specifies the transformed dependent variables prefix, TIPREFIX=name specifies the transformed independent variables prefix, ADPREFIX=name specifies the approximations to the dependent variables prefix, AIPREFIX=name specifies the approximations to the independent variables prefix, and CPREFIX=n specifies the number of first characters of a CLASS variable's name that are to be used in constructing binary variable names for the output data set.

The design matrix options are used to specify the details of how the intercept and binary CLASS variables are created. The options are: NOINT specifies that no intercept term be added to the independent variables, DINT specifies that an intercept term be added to the dependent variables, NODINT specifies that an intercept term not be added to the dependent variables, and ALLCATS specifies that there should be one binary variable created per category of each CLASS variable.

The remaining options are: TSTANDARD= specifies how the means and variances of the transformed variables should be set, NOMISS specifies that all observations with missing values be excluded from the analysis, NOPRINT specifies that there be no printed output, ORDCAT=name specifies those special missing values that are to be estimated with within variable order constraints. A POLYNOMIAL statement may be used to specify knots and polynomial degrees for variables that may be subjected to a SPLINE or ISPLINE transformation.
DEGREE=n specifies the degree of the polynomial spline transformation, KNOTS=numberlist specifies the interior knots or break points, NKNOTS=n specifies that PROC TRANSREG should create 0 knots at evenly spaced percentiles. Knots may be repeated to indicate breaks in lower order derivatives.

Finally, an ID statement can be used to list additional character and/or numeric variables that are to be included in the output data set, and a BY statement can be used to obtain separate analyses on observation groups.

PROC TRANSREG Missing Value Estimation

PROC TRANSREG has very flexible missing value estimation capabilities. Missing values within OPSCORE, MONOTONE, UNTIE, LINEAR, SPLINE, and ISPLINE transformation variables may be estimated so that the variance accounted for by the linear model is maximized. A variety of estimation restrictions are provided. No category or order restrictions are placed on the estimates of ordinary missing values (.). The 27 special missing values (-., ., , .A through .2) can be used to indicate categorical missing values, whose estimates, within class and variable, must be identical. In addition, some missing values can be ordered. The ORDCAT=n option can be used to indicate a range of special missing values from the list .A to .2, whose estimates must be weakly ordered within each variable that they appear.

The OPSCORE, MONOTONE, UNTIE, LINEAR, SPLINE, and ISPLINE functions, can be combined with nonmissing values, ordinary missing values, special missing values, and ordered categorical missing values, in any way, as long as there are nonmissing values within each variable that have some variance. The missing value estimation facilities allow for partitioned variables. For example, a LINEAR variable with special missing and ordered categorical missing can be part interval, part ordinal, and part nominal. A MONOTONE variable may have two independent ordinal parts and nominal classes. An UNTIE variable may have an ordered categorical part and an ordered part without category restrictions. There are many more possible examples.

A PROC TRANSREG Example

PROC TRANSREG is easy to use. This example illustrates using PROC TRANSREG in simple regression to find the optimal linear regression line, a nonlinear but monotone regression line, and a nonlinear nonmonotone regression line. A linear regression line can be found by specifying:

```
PROC TRANSREG APPROXIMATIONS DATA=A
   OUT=B;
   MODEL LINEAR(Y) = LINEAR(X);
```

A monotone regression line can be found by requesting an ISPLINE transformation of the independent variable.

```
PROC TRANSREG APPROXIMATIONS DATA=A
   OUT=B;
   MODEL LINEAR(Y) = ISPLINE(X);
   POLYNOMIAL X / NKNOTS=9;
```

The monotonicity restriction may be relaxed by requesting a SPLINE transformation of the independent variable.

```
PROC TRANSREG APPROXIMATIONS DATA=A
   OUT=B;
   MODEL LINEAR(Y) = SPLINE(X);
   POLYNOMIAL X / NKNOTS=9;
```

In all cases, the results can be displayed by running PROC PLOT with a PLOT statement like:

```
PLOT AX*Y='r' Y*X='*' / OVERLAY;
```

The ordinate values of the regression line are contained in the dependent variable approximation and are indicated by the 'r' characters in the plots. The results are shown below.

Nonlinear Scatterplot

```
Y

* 
**** *
****** *****
*** ** ********
**** ** **
** *

X
```

Linear Regression R^2 = .16

```
Y

* 
**** *
****** *****
rrrrrrrrrrrrrrrrr *** *** rrrrrr
** *

X
```
The squared correlation is only 0.16 for the linear regression, showing that a simple linear regression model is not optimal for these data. By relaxing the constraints placed on the regression line the proportion of variance accounted for increases from 0.16 (linear) to 0.64 (monotone) to 0.92 (nonmonotone). Relaxing the linearity constraint allows the regression line to bend and more closely follow the right portion of the scatterplot. Relaxing the monotonicity constraint allows the regression line to more closely follow the periodic portion of the left side of the plot. The nonlinear ISPLINE transformation is a quadratic polynomial spline with knots at the deciles. The nonlinear monotonic SPLINE transformation is a cubic spline with knots at the deciles.

Different knots and different polynomial degrees would produce different results. The two nonlinear regression lines could be closely approximated by simpler piece-wise linear regression lines. The monotone line could be approximated by a piece line with a single knot at the elbow. The nonmonotone line could be approximated by a five piece line with knots at the four elbows.

With this sort of problem, PROC TRANSREG always iterates exactly twice (although only one iteration is necessary). The first iteration reports the $R^2$ for the linear regression line, and finds the optimal transformation of X. Since the data change in the first iteration, a second iteration is performed that reports the $R^2$ for the final nonlinear regression line, and zero data change. The approximation of Y, which is a linear combination of the optimal transformation of X contains the y-coordinates for the nonlinear regression line. The variance of the approximation of Y divided by the variance of Y is the $R^2$ for the fit of the nonlinear regression line. When X is monotonically transformed, the transformation of X is always monotonically increasing, but the approximation to Y, and hence the regression line, is increasing if the correlation is positive and decreasing for negative correlations.

### The PRINQUAL Procedure

The PRINQUAL procedure (Young and Kuhfeld, 1986) has three algorithms for transforming a set of qualitative and quantitative variables. One algorithm follows from a principal components model, the second from a regression model, and the third from a regression-like model. The qualitative variables are transformed to reduce the rank of the data set. Any mixture of qualitative and quantitative variables may be included in the analysis.

The PRINQUAL procedure has many uses. It can be viewed as a generalization of ordinary principal components analysis to a method capable of analyzing data that are not quantitative. It can be used to perform metric and nonmetric MDPREF analyses Carroll (1972). It can be viewed as a data preprocessor - used to transform data prior to their use in any other data analysis. In addition, PRINQUAL has powerful capabilities for estimating missing data.

The statements available in PROC PRINQUAL are:

```
PROC PRINQUAL [options];
VAR t1(variables)
    [t2(variables) ... ];
BY variables;
ID charactervariable;
```

The PROC PRINQUAL statement has a number of options. PROC PRINQUAL is a scoring procedure - its function is to create an output data set that can be input to other procedures. The data set options can be used to name the input and output SAS data sets. Other options control the contents and structure of the output data set. The output data set can contain original data, transformed data, component scores, data approximations, correlations, and component structure. There are a number of options that control the details of the analysis; covariances or correlations may be used.
the number of components, maximum number of iterations, convergence criterion, singularity criterion, initialization method, data and score standardization, and time/accuracy trade-offs may all be specified.

There are three iterative algorithms available in PROC PRINQUAL. The maximum total variance (MTV) method (Young, Takane, and de Leeuw, 1978) transforms the qualitative variables to optimize variables the amount of variance accounted for in all the variables (including the quantitative ones) by a few components. It alternates on each iteration, classical principal components analysis with optimal scaling. This method can be used to perform Carroll's (1972) MREF method, data and standardization. This method can be initialized using the method suggested by Tenenhaus and Vachette (1977).

The minimum generalized variance (MGV) method (Kuhfeld, Sarle, and Young, 1985) transforms the variables to minimize the determinant of the covariance matrix. This method, alternates for each variable multiple regression and optimal scaling. The multiple regression involves predicting the selected variable from all other variables. The iterations can be initialized using a variation of the Tenenhaus and Vachette (1977) method that is appropriate to the 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 mixed measurement level data matrix whose qualitative variables may be nonlinearly transformed. This method tries to create a low rank data matrix that contains the representation of each variable that is most like what the other variables would predict. Under the assumption of multivariate normality, the MGV method can be used as an EM missing data estimation algorithm.

The maximum average correlation (MAC) method (de Leeuw, 1985) transforms the variables to maximize the average of the correlation coefficients. Variables are alternates for each variable, computing a linear combination of the other variables (with all weights equal) and optimal scaling. This method is similar to the MGV method since each qualitative variable is scaled to be like a linear combination of the other variables, but optimal weights are not computed. This method can safely be used when all variables are nominal, or when all variables are positively correlated. It should be used only with extreme care when some ordinal and interval variables are negatively correlated since signs are not taken into account. This method can be used as an iteration initialization method for the MTV and MGV methods.

PROC PRINQUAL has powerful missing data estimation facilities. All of the missing data features available in PROC TRANSREG either are already available in PROC PRINQUAL, or will be available in the next test version.

The VAR statement lists the variables to be analyzed, and the transformation family for each variable. The format of the statement is: VAR v1(variables) [v2(variables) ... ]; where v1, v2, ..., name the transformation families of the parenthesized variable lists. There are currently four valid families: OPSCORE, MONOTONE, UNTIE, and LINEAR. SPLINE and ISPLINE will be available in the next release. All of these transformation families are discussed with PROC TRANSREG.

PROC PRINQUAL produces very little printed output (only an iteration history summary table). In order to examine the results of the analysis in more detail, the information in the output data set can be analyzed with other SAS procedures. PROC FACTOR can be used to perform a components analysis on the transformed data, PROC PLOT can be used to show the transformations, etc.

Final Remarks

During the middle of 1984 a three year SAS software development project began at the University of North Carolina at Chapel Hill, under the direction of Forrest Young. The goal of the project was to make available within the SAS system, exploratory multivariate data analysis methods such as correspondence analysis, transformation regression, and qualitative principal components analysis. In addition, research was planned into interactive hyperdimensional graphics, and user interfaces. Now the three years are almost over and this is the last paper reporting on the project that will be presented during the duration of the project. We are pleased to say that PROC's PRINQUAL, TRANSREG and CORRESP are working in SAS 6.02 as of this writing, and VISUALS (Young, Kent, and Kuhfeld, 1987) is running as a stand-alone program and under 6.03. In addition, PROC IDPLOT (Kuhfeld, 1986) and the enhanced PRINQUAL macro (Young and Kuhfeld, 1985) are running under version 5. Work on PROC's PROXSCAL (multidimensional scaling) and COSAN (covariance structure analysis) in
version 6 is underway. It is our hope that these new procedures will provide SAS users with valuable new tools for exploratory multivariate data analysis.

References


Young, F.W., and Kuhfeld, W.F. (1986), The CORRESP Procedure: Experimental Software for Correspondence

