SAS/STAT® 12.3 User's Guide
The FACTOR Procedure
(Chapter)

Chapter 35
The FACTOR Procedure

Contents

Overview: FACTOR Procedure

The FACTOR procedure performs a variety of common factor and component analyses and rotations. Input can be multivariate data, a correlation matrix, a covariance matrix, a factor pattern, or a matrix of scoring coefficients. The procedure can factor either the correlation or covariance matrix, and you can save most results in an output data set.

PROC FACTOR can process output from other procedures. For example, it can rotate the canonical coefficients from multivariate analyses in the GLM procedure.

The methods for factor extraction are principal component analysis, principal factor analysis, iterated principal factor analysis, unweighted least squares factor analysis, maximum likelihood (canonical) factor analysis, alpha factor analysis, image component analysis, and Harris component analysis. A variety of methods for prior communality estimation is also available.

Specific methods for orthogonal rotation are varimax, quartimax, biquartimax, equamax, parsimax, and factor parsimax. Oblique versions of these methods are also available. In addition, quartimin, biquartimin, and covarimin methods for (direct) oblique rotation are available. General methods for orthogonal rotation are orthomax with user-specified gamma, Crawford-Ferguson family with user-specified weights on variable parsimony and factor parsimony, and generalized Crawford-Ferguson family with user-specified weights. General methods for oblique rotation are direct oblimin with user-specified tau, Crawford-Ferguson family with user-specified weights on variable parsimony and factor parsimony, generalized Crawford-Ferguson family with user-specified weights, promax with user-specified exponent, Harris-Kaiser case II with user-specified exponent, and Procrustes with a user-specified target pattern.

Output includes means, standard deviations, correlations, Kaiser's measure of sampling adequacy, eigenvalues, a scree plot, eigenvectors, prior and final communality estimates, the unrotated factor pattern, residual and partial correlations, the rotated primary factor pattern, the primary factor structure, interfactor correlations, the reference structure, reference axis correlations, the variance explained by each factor both ignoring and eliminating other factors, plots of both rotated and unrotated factors, squared multiple correlation of each factor with the variables, standard error estimates, confidence limits, coverage displays, and scoring coefficients.

The FACTOR procedure uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Any topics that are not given explicit references are discussed in Mulaik (1972) or Harman (1976).

Background

See Chapter 73, "The PRINCOMP Procedure," for a discussion of principal component analysis. See Chapter 27, "The CALIS Procedure," for a discussion of confirmatory factor analysis.

Common factor analysis was invented by Spearman (1904). Kim and Mueller (1978a, b) provide a very elementary discussion of the common factor model. Gorsuch (1974) presents a broad survey of factor analysis, and Gorsuch (1974) and Cattell (1978) are useful as guides to practical research methodology. Harman (1976) gives a lucid discussion of many of the more technical aspects of factor analysis, especially oblique rotation. Morrison (1976) and Mardia, Kent, and Bibby (1979) provide excellent statistical treatments of common factor analysis. Mulaik (1972) provides the most thorough and authoritative general reference on

factor analysis and is highly recommended to anyone familiar with matrix algebra. Stewart (1981) gives a nontechnical presentation of some issues to consider when deciding whether or not a factor analysis might be appropriate.

A frequent source of confusion in the field of factor analysis is the term *factor*. It sometimes refers to a hypothetical, unobservable variable, as in the phrase *common factor*. In this sense, *factor analysis* must be distinguished from component analysis since a component is an observable linear combination. *Factor* is also used in the sense of *matrix factor,* in that one matrix is a factor of a second matrix if the first matrix multiplied by its transpose equals the second matrix. In this sense, *factor analysis* refers to all methods of data analysis that use matrix factors, including component analysis and common factor analysis.

A *common factor* is an unobservable, hypothetical variable that contributes to the variance of at least two of the observed variables. The unqualified term "factor" often refers to a common factor. A *unique factor* is an unobservable, hypothetical variable that contributes to the variance of only one of the observed variables. The model for common factor analysis posits one unique factor for each observed variable.

The equation for the common factor model is

$$y_{ij} = x_{i1}b_{1j} + x_{i2}b_{2j} + \cdots + x_{iq}b_{qj} + e_{ij}$$

where

y_{ij}	is the value of the *i*th observation on the *j*th variable
x_{ik}	is the value of the *i*th observation on the *k*th common factor
b_{kj}	is the regression coefficient of the *k*th common factor for predicting the *j*th variable
e_{ij}	is the value of the *i*th observation on the *j*th unique factor
q	is the number of common factors

It is assumed, for convenience, that all variables have a mean of 0. In matrix terms, these equations reduce to

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

In the preceding equation, $\mathbf{X}$ is the matrix of factor scores, and $\mathbf{B}'$ is the factor pattern.

There are two critical assumptions:

- The unique factors are uncorrelated with each other.

- The unique factors are uncorrelated with the common factors.

In principal component analysis, the residuals are generally correlated with each other. In common factor analysis, the unique factors play the role of residuals and are defined to be uncorrelated both with each other and with the common factors. Each common factor is assumed to contribute to at least two variables; otherwise, it would be a unique factor.

When the factors are initially extracted, it is also assumed, for convenience, that the common factors are uncorrelated with each other and have unit variance. In this case, the common factor model implies that the covariance s_{jk} between the *j*th and *k*th variables, $j \neq k$, is given by

$$s_{jk} = b_{1j}b_{1k} + b_{2j}b_{2k} + \cdots + b_{qj}b_{qk}$$

or

$$S = B'B + U^2$$

where S is the covariance matrix of the observed variables, and U^2 is the diagonal covariance matrix of the unique factors.

If the original variables are standardized to unit variance, the preceding formula yields correlations instead of covariances. It is in this sense that common factors explain the correlations among the observed variables. When considering the diagonal elements of standardized S, the variance of the *j*th variable is expressed as

$$s_{jj} = 1 = b_{1j}^2 + b_{2j}^2 + \cdots + b_{qj}^2 + [U^2]_{jj}$$

where $b_{1j}^2 + b_{2j}^2 + \cdots + b_{qj}^2$ and $[U^2]_{jj}$ are the communality and uniqueness, respectively, of the *j*th variable. The communality represents the extent of the overlap with the common factors. In other words, it is the proportion of variance accounted for by the common factors.

The difference between the correlation predicted by the common factor model and the actual correlation is the *residual correlation*. A good way to assess the goodness of fit of the common factor model is to examine the residual correlations.

The common factor model implies that the partial correlations among the variables, removing the effects of the common factors, must all be zero. When the common factors are removed, only unique factors, which are by definition uncorrelated, remain.

The assumptions of common factor analysis imply that the common factors are, in general, not linear combinations of the observed variables. In fact, even if the data contain measurements on the entire population of observations, you cannot compute the scores of the observations on the common factors. Although the common factor scores cannot be computed directly, they can be estimated in a variety of ways.

The problem of factor score indeterminacy has led several factor analysts to propose methods yielding components that can be considered approximations to common factors. Since these components are defined as linear combinations, they are computable. The methods include Harris component analysis and image component analysis. The advantage of producing determinate component scores is offset by the fact that, even if the data fit the common factor model perfectly, component methods do not generally recover the correct factor solution. You should not use any type of component analysis if you really want a common factor analysis (Dziuban and Harris 1973; Lee and Comrey 1979).

After the factors are estimated, it is necessary to interpret them. Interpretation usually means assigning to each common factor a name that reflects the *salience* of the factor in predicting each of the observed variables—that is, the coefficients in the pattern matrix corresponding to the factor. Factor interpretation is a subjective process. It can sometimes be made less subjective by *rotating* the common factors—that is, by applying a nonsingular linear transformation. A rotated pattern matrix in which all the coefficients are close to 0 or ±1 is easier to interpret than a pattern with many intermediate elements. Therefore, most rotation methods attempt to optimize a simplicity function of the rotated pattern matrix that measures, in some sense, how close the elements are to 0 or ±1. Because the loading estimates are subject to sampling variability, it is useful to obtain the standard error estimates for the loadings for assessing the uncertainty due to random sampling. Notice that the *salience* of a factor loading refers to the magnitude of the loading, while statistical *significance* refers to the statistical evidence against a particular hypothetical value. A loading significantly different from 0 does not automatically mean it must be salient. For example, if salience is defined as a

magnitude larger than 0.4 while the entire 95% confidence interval for a loading lies between 0.1 and 0.3, the loading is statistically significant larger than 0 but it is not salient. Under the maximum likelihood method, you can obtain standard errors and confidence intervals for judging the salience of factor loadings.

After the initial factor extraction, the common factors are uncorrelated with each other. If the factors are rotated by an *orthogonal transformation,* the rotated factors are also uncorrelated. If the factors are rotated by an *oblique transformation,* the rotated factors become correlated. Oblique rotations often produce more useful patterns than do orthogonal rotations. However, a consequence of correlated factors is that there is no single unambiguous measure of the importance of a factor in explaining a variable. Thus, for oblique rotations, the pattern matrix does not provide all the necessary information for interpreting the factors; you must also examine the *factor structure* and the *reference structure*.

Rotating a set of factors does not change the statistical explanatory power of the factors. You cannot say that any rotation is better than any other rotation from a statistical point of view; all rotations, orthogonal or oblique, are equally good statistically. Therefore, the choice among different rotations must be based on nonstatistical grounds. For most applications, the preferred rotation is that which is most easily interpretable, or most compatible with substantive theories.

If two rotations give rise to different interpretations, those two interpretations must not be regarded as conflicting. Rather, they are two different ways of looking at the same thing, two different points of view in the common-factor space. Any conclusion that depends on one and only one rotation being correct is invalid.

Outline of Use

Principal Component Analysis

One important type of analysis performed by the FACTOR procedure is principal component analysis. The following statements result in a principal component analysis:

```
proc factor;
   run;
```

The output includes all the eigenvalues and the pattern matrix for eigenvalues greater than one.

Most applications require additional output. For example, you might want to compute principal component scores for use in subsequent analyses or obtain a graphical aid to help decide how many components to keep. You can save the results of the analysis in a permanent SAS data library by using the OUTSTAT= option. For more information about permanent libraries and SAS data sets, see *SAS Language Reference: Concepts*. Assuming that your SAS data library has the libref save and that the data are in a SAS data set called raw, you could do a principal component analysis as follows:

```
proc factor data=raw method=principal scree mineigen=0 score
      outstat=save.fact_all;
   run;
```

The SCREE option produces a plot of the eigenvalues that is helpful in deciding how many components to use. Alternative, you can use the PLOTS=SCREE option to produce high-quality scree plots. The MINEIGEN=0 option causes all components with variance greater than zero to be retained. The SCORE option requests that scoring coefficients be computed. The OUTSTAT= option saves the results in a specially

structured SAS data set. The name of the data set, in this case fact_all, is arbitrary. To compute principal component scores, use the SCORE procedure:

```
proc score data=raw score=save.fact_all out=save.scores;
run;
```

The SCORE procedure uses the data and the scoring coefficients that are saved in save.fact_all to compute principal component scores. The component scores are placed in variables named Factor1, Factor2, ..., Factor*n* and are saved in the data set save.scores. If you know ahead of time how many principal components you want to use, you can obtain the scores directly from PROC FACTOR by specifying the NFACTORS= and OUT= options. To get scores from three principal components, specify the following:

```
proc factor data=raw method=principal
    nfactors=3 out=save.scores;
run;
```

To plot the scores for the first three components, use the SGSCATTER procedure:

```
proc sgscatter;
   matrix factor1-factor3;
run;
```

Principal Factor Analysis

The simplest and computationally most efficient method of common factor analysis is principal factor analysis, which is obtained in the same way as principal component analysis except for the use of the PRIORS= option. The usual form of the initial analysis is as follows:

```
proc factor data=raw method=principal scree
    mineigen=0 priors=smc outstat=save.fact_all;
run;
```

The squared multiple correlations (SMC) of each variable with all the other variables are used as the prior communality estimates. If your correlation matrix is singular, you should specify PRIORS=MAX instead of PRIORS=SMC. The SCREE and MINEIGEN= options serve the same purpose as in the preceding principal component analysis. Saving the results with the OUTSTAT= option enables you to examine the eigenvalues and scree plot before deciding how many factors to rotate and to try several different rotations without re-extracting the factors. The OUTSTAT= data set is automatically marked TYPE=FACTOR, so the FACTOR procedure realizes that it contains statistics from a previous analysis instead of raw data.

After looking at the eigenvalues to estimate the number of factors, you can try some rotations. Two and three factors can be rotated with the following statements:

```
proc factor data=save.fact_all method=principal n=2
    rotate=promax reorder score outstat=save.fact_2;
run;
proc factor data=save.fact_all method=principal n=3
    rotate=promax reorder score outstat=save.fact_3;
run;
```

The output data set from the previous run is used as input for these analyses. The options N=2 and N=3 specify the number of factors to be rotated. The specification ROTATE=PROMAX requests a promax rotation, which has the advantage of providing both orthogonal and oblique rotations with only one invocation of

PROC FACTOR. The REORDER option causes the variables to be reordered in the output so that variables associated with the same factor appear next to each other.

You can now compute and plot factor scores for the two-factor promax-rotated solution as follows:

```
proc score data=raw score=save.fact_2 out=save.scores;
run;

proc sgplot;
   scatter y=factor2 x=factor1;
run;
```

Maximum Likelihood Factor Analysis

Although principal factor analysis is perhaps the most commonly used method of common factor analysis, most statisticians prefer maximum likelihood (ML) factor analysis (Lawley and Maxwell 1971). The ML method of estimation has desirable asymptotic properties (Bickel and Doksum 1977) and produces better estimates than principal factor analysis in large samples. You can test hypotheses about the number of common factors by using the ML method. You can also obtain standard error and confidence interval estimates for many classes of rotated or unrotated factor loadings, factor correlations, and structure loadings under the ML theory.

The unrotated ML solution is equivalent to Rao's canonical factor solution (Rao 1955) and Howe's solution maximizing the determinant of the partial correlation matrix (Morrison 1976). Thus, as a descriptive method, ML factor analysis does not require a multivariate normal distribution. The validity of Bartlett's χ^2 test for the number of factors does require approximate normality plus additional regularity conditions that are usually satisfied in practice (Geweke and Singleton 1980). Bartlett's test of sphericity in the context of factor analysis is equivalent to Bartlett's χ^2 test for zero common factors. This test is routinely displayed in the maximum likelihood factor analysis output.

Lawley and Maxwell (1971) derive the standard error formulas for unrotated loadings, while Archer and Jennrich (1973) and Jennrich (1973, 1974) derive the standard error formulas for several classes of rotated solutions. Extended formulas for computing standard errors in various situations appear in Browne et al. (2008); Hayashi and Yung (1999); Yung and Hayashi (2001). A combination of these methods is used in PROC FACTOR to compute standard errors in an efficient manner. Confidence intervals are computed by using the asymptotic normality of the estimates. To ensure that the confidence intervals fall within the admissible parameter range, transformation methods due to Browne (1982) are used. The validity of the standard error estimates and confidence limits requires the assumptions of multivariate normality and a fixed number of factors.

The ML method is more computationally demanding than principal factor analysis for two reasons. First, the communalities are estimated iteratively, and each iteration takes about as much computer time as principal factor analysis. The number of iterations typically ranges from about five to twenty. Second, if you want to extract different numbers of factors, as is often the case, you must run the FACTOR procedure once for each number of factors. Therefore, an ML analysis can take 100 times as long as a principal factor analysis. This does not include the time for computing standard error estimates, which is even more computationally demanding. For analyses with fewer than 35 variables, the computing time for the ML method, including the computation of standard errors, usually ranges from a few seconds to well under a minute. This seems to be a reasonable performance.

You can use principal factor analysis to get a rough idea of the number of factors before doing an ML analysis. If you think that there are between one and three factors, you can use the following statements for the ML analysis:

```
proc factor data=raw method=ml n=1
     outstat=save.fact1;
run;
proc factor data=raw method=ml n=2 rotate=promax
     outstat=save.fact2;
run;
proc factor data=raw method=ml n=3 rotate=promax
     outstat=save.fact3;
run;
```

The output data sets can be used for trying different rotations, computing scoring coefficients, or restarting the procedure in case it does not converge within the allotted number of iterations.

If you can determine how many factors should be retained before an analysis, as in the following statements, you can get the standard errors and confidence limits to aid interpretations for the ML analysis:

```
proc factor data=raw method=ml n=3 rotate=quartimin se
   cover=.4;
run;
```

In this analysis, you specify the quartimin rotation in the ROTATE= option. The SE option requests the computation of standard error estimates. In the COVER= option, you require absolute values of 0.4 or greater in order for loadings to be salient. In the output of coverage display, loadings that are salient would have their entire confidence intervals spanning beyond the 0.4 mark (or the –0.4 mark in the opposite direction). Only those salient loadings should be used for interpreting the factors. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 2243 for more details.

The ML method cannot be used with a singular correlation matrix, and it is especially prone to Heywood cases. See the section "Heywood Cases and Other Anomalies about Communality Estimates" on page 2248 for a discussion of Heywood cases. If you have problems with ML, the best alternative is to use the METHOD=ULS option for unweighted least squares factor analysis.

Factor Rotation

After the initial factor extraction, the factors are uncorrelated with each other. If the factors are rotated by an *orthogonal transformation,* the rotated factors are also uncorrelated. If the factors are rotated by an *oblique transformation,* the rotated factors become correlated. Oblique rotations often produce more useful patterns than orthogonal rotations do. However, a consequence of correlated factors is that there is no single unambiguous measure of the importance of a factor in explaining a variable. Thus, for oblique rotations, the pattern matrix does not provide all the necessary information for interpreting the factors; you must also examine the *factor structure* and the *reference structure*.

Nowadays, most rotations are done analytically. There are many choices for orthogonal and oblique rotations. An excellent summary of a wide class of analytic rotations is in Crawford and Ferguson (1970). The Crawford-Ferguson family of orthogonal rotations includes the orthomax rotation as a subclass and the popular varimax rotation as a special case. To illustrate these relationships, the following four specifications for orthogonal rotations with different ROTATE= options will give the same results for a data set with nine observed variables:

```
    /* Orthogonal Crawford-Ferguson Family with
       variable parsimony weight = nvar - 1 = 8, and
       factor parsimony weight = 1  */
proc factor data=raw n=3 rotate=orthcf(8,1);
run;


    /* Orthomax without the GAMMA= option */
proc factor data=raw n=3 rotate=orthomax(1);
run;


    /* Orthomax with the GAMMA= option */
proc factor data=raw n=3 rotate=orthomax gamma=1;
run;


    /* Varimax */
proc factor data=raw n=3 rotate=varimax;
run;
```

You can also get the oblique versions of the varimax in two equivalent ways:

```
    /* Oblique Crawford-Ferguson Family with
       variable parsimony weight = nvar - 1 = 8, and
       factor parsimony weight = 1; */
proc factor data=raw n=3 rotate=oblicf(8,1);
run;


    /* Oblique Varimax */
proc factor data=raw n=3 rotate=obvarimax;
run;
```

Jennrich (1973) proposes a generalized Crawford-Ferguson family that includes the Crawford-Ferguson family and the (direct) oblimin family (see Harman 1976) as subclasses. The better-known quartimin rotation is a special case of the oblimin class, and hence a special case of the generalized Crawford-Ferguson family. For example, the following four specifications of oblique rotations are equivalent:

```
    /* Oblique generalized Crawford-Ferguson Family
       with weights 0, 1, 0, -1 */
proc factor data=raw n=3 rotate=obligencf(0,1,0,-1);
run;


    /* Oblimin family without the TAU= option */
proc factor data=raw n=3 rotate=oblimin(0);
run;


    /* Oblimin family with the TAU= option */
proc factor data=raw n=3 rotate=oblimin tau=0;
run;


    /* Quartimin */
proc factor data=raw n=3 rotate=quartimin;
run;
```

In addition to the generalized Crawford-Ferguson family, the available oblique rotation methods in PROC FACTOR include Harris-Kaiser, promax, and Procrustes. See the section "Simplicity Functions for Rotations" on page 2244 for details about the definitions of various rotations. See Harman (1976) and Mulaik (1972) for further information.

Getting Started: FACTOR Procedure

The following example demonstrates how you can use the FACTOR procedure to perform common factor analysis and factor rotation.

In this example, 103 police officers were rated by their supervisors on 14 scales (variables). You conduct a common factor analysis on these variables to see what latent factors are operating behind these ratings. The overall rating variable is excluded from the factor analysis.

The following DATA step creates the SAS data set jobratings:

```
options validvarname=any;
data jobratings;
   input ('Communication Skills'n
          'Problem Solving'n
          'Learning Ability'n
          'Judgment Under Pressure'n
          'Observational Skills'n
          'Willingness to Confront Problems'n
          'Interest in People'n
          'Interpersonal Sensitivity'n
          'Desire for Self-Improvement'n
          'Appearance'n
          'Dependability'n
          'Physical Ability'n
          'Integrity'n
          'Overall Rating'n) (1.);
   datalines;
26838853879867
74758876857667
56757863775875
67869777988997
99997798878888
89897899888799
89999889899798
87794798468886

   ... more lines ...

99899899899899
76656399567486
;
```

The following statements invoke the FACTOR procedure:

```
proc factor data=jobratings(drop='Overall Rating'n) priors=smc
   rotate=varimax;
run;
```

The DATA= option in PROC FACTOR specifies the SAS data set jobratings as the input data set. The DROP= option drops the Overall Rating variable from the analysis. To conduct a common factor analysis, you need to set the prior communality estimate to less than one for each variable. Otherwise, the factor solution would simply be a recast of the principal components solution, in which "factors" are linear combinations of observed variables. However, in the common factor model you always assume that observed variables are functions of underlying factors. In this example, the PRIORS= option specifies that the squared multiple correlations (SMC) of each variable with all the other variables are used as the prior communality estimates. Note that squared multiple correlations are usually less than one. By default, the principal factor extraction is used if the METHOD= option is not specified. To facilitate interpretations, the ROTATE= option specifies the VARIMAX orthogonal factor rotation to be used.

The output from the factor analysis is displayed in Figure 35.1 through Figure 35.5.

As displayed in Figure 35.1, the prior communality estimates are set to the squared multiple correlations. Figure 35.1 also displays the table of eigenvalues (the variances of the principal factors) of the reduced correlation matrix. Each row of the table pertains to a single eigenvalue. Following the column of eigenvalues are three measures of each eigenvalue's relative size and importance. The first of these displays the difference between the eigenvalue and its successor. The last two columns display the individual and cumulative proportions that the corresponding factor contributes to the total variation. The last line displayed in Figure 35.1 states that three factors are retained, as determined by the PROPORTION criterion.

Figure 35.1 Table of Eigenvalues from PROC FACTOR

```
                        The FACTOR Procedure
                Initial Factor Method: Principal Factors


                  Prior Communality Estimates: SMC


                                        Judgment
 Communication        Problem        Learning           Under        Observational
       Skills         Solving         Ability        Pressure               Skills

   0.62981394      0.58657431      0.61009871      0.63766021           0.67187583


     Willingness
     to Confront        Interest    Interpersonal          Desire for
        Problems       in People      Sensitivity      Self-Improvement

      0.64779805      0.75641519       0.75584891          0.57460176


                                        Physical
       Appearance     Dependability       Ability          Integrity

       0.45505304        0.63449045     0.42245324        0.68195454
```

Figure 35.1 *continued*

```
        Eigenvalues of the Reduced Correlation Matrix:
             Total = 8.06463816  Average = 0.62035678

          Eigenvalue    Difference    Proportion    Cumulative

     1     6.17760549   4.71531946       0.7660        0.7660
     2     1.46228602   0.90183348       0.1813        0.9473
     3     0.56045254   0.28093933       0.0695        1.0168
     4     0.27951322   0.04766016       0.0347        1.0515
     5     0.23185305   0.16113428       0.0287        1.0802
     6     0.07071877   0.07489624       0.0088        1.0890
     7    -.00417747    0.03387533      -0.0005        1.0885
     8    -.03805279    0.04776534      -0.0047        1.0838
     9    -.08581814    0.02438060      -0.0106        1.0731
    10    -.11019874    0.01452741      -0.0137        1.0595
    11    -.12472615    0.02356465      -0.0155        1.0440
    12    -.14829080    0.05823605      -0.0184        1.0256
    13    -.20652684                    -0.0256        1.0000


     3 factors will be retained by the PROPORTION criterion.
```

Figure 35.2 displays the initial factor pattern matrix. The factor pattern matrix represents standardized regression coefficients for predicting the variables by using the extracted factors. Because the initial factors are uncorrelated, the pattern matrix is also equal to the correlations between variables and the common factors.

Figure 35.2 Factor Pattern Matrix from PROC FACTOR

```
                             Factor Pattern

                                      Factor1        Factor2        Factor3

Communication Skills                  0.75441        0.07707       -0.25551
Problem Solving                       0.68590        0.08026       -0.34788
Learning Ability                      0.65904        0.34808       -0.25249
Judgment Under Pressure               0.73391       -0.21405       -0.23513
Observational Skills                  0.69039        0.45292        0.10298
Willingness to Confront Problems      0.66458        0.47460        0.09210
Interest in People                    0.70770       -0.53427        0.10979
Interpersonal Sensitivity            0.64668        -0.61284       -0.07582
Desire for Self-Improvement           0.73820        0.12506        0.09062
Appearance                            0.57188        0.20052        0.16367
Dependability                         0.79475       -0.04516        0.16400
Physical Ability                      0.51285        0.10251        0.34860
Integrity                             0.74906       -0.35091        0.18656
```

The pattern matrix suggests that Factor1 represents general ability. All loadings for Factor1 in the Factor Pattern are at least 0.5. Factor2 consists of high positive loadings on certain task-related skills (Willingness to Confront Problems, Observational Skills, and Learning Ability) and high negative loadings on some interpersonal skills (Interpersonal Sensitivity, Interest in People, and Integrity). This factor measures individuals' relative strength in these skills. Theoretically, individuals with high positive scores on this factor would exhibit better task-related skills than interpersonal skills. Individuals with high negative scores would exhibit better interpersonal skills than task-related skills. Individuals with scores near zero would have those skills balanced. Factor3 does not have a cluster of very high or very low factor loadings. Therefore, interpreting this factor is difficult.

Figure 35.3 displays the proportion of variance explained by each factor and the final communality estimates, including the total communality. The final communality estimates are the proportion of variance of the variables accounted for by the common factors. When the factors are orthogonal, the final communalities are calculated by taking the sum of squares of each row of the factor pattern matrix.

Figure 35.3 Variance Explained and Final Communality Estimates

```
                  Variance Explained by Each Factor

               Factor1           Factor2           Factor3

             6.1776055         1.4622860         0.5604525


           Final Communality Estimates: Total = 8.200344

                                                Judgment
   Communication         Problem        Learning      Under       Observational
       Skills            Solving         Ability     Pressure          Skills

     0.64036292        0.59791844      0.61924167   0.63972863        0.69237485

      Willingness
      to Confront        Interest      Interpersonal              Desire for
        Problems         in People      Sensitivity            Self-Improvement

      0.67538695        0.79833968      0.79951357                0.56879171

                                        Physical
        Appearance      Dependability    Ability         Integrity

        0.39403630       0.66056907     0.39504805       0.71903222
```

Figure 35.4 displays the results of the VARIMAX rotation of the three extracted factors and the corresponding orthogonal transformation matrix. The rotated factor pattern matrix is calculated by postmultiplying the original factor pattern matrix (Figure 35.4) by the transformation matrix.

Figure 35.4 Transformation Matrix and Rotated Factor Pattern

```
              Orthogonal Transformation Matrix

                     1              2              3

          1        0.59125        0.59249        0.54715
          2       -0.80080        0.51170        0.31125
          3        0.09557        0.62219       -0.77701


              Rotated Factor Pattern

                                  Factor1        Factor2        Factor3

Communication Skills              0.35991        0.32744        0.63530
Problem Solving                   0.30802        0.23102        0.67058
Learning Ability                  0.08679        0.41149        0.66512
Judgment Under Pressure           0.58287        0.17901        0.51764
Observational Skills              0.05533        0.70488        0.43870
Willingness to Confront Problems  0.02168        0.69391        0.43978
Interest in People                0.85677        0.21422        0.13562
Interpersonal Sensitivity         0.86587        0.02239        0.22200
Desire for Self-Improvement       0.34498        0.55775        0.37242
Appearance                        0.19319        0.54327        0.24814
Dependability                     0.52174        0.54981        0.29337
Physical Ability                  0.25445        0.57321        0.04165
Integrity                         0.74172        0.38033        0.15567
```

The rotated factor pattern matrix is somewhat simpler to interpret. If a magnitude of at least 0.5 is required to indicate a salient variable-factor relationship, Factor1 now represents interpersonal skills (Interpersonal Sensitivity, Interest in People, Integrity, Judgment Under Pressure, and Dependability). Factor2 measures physical skills and job enthusiasm (Observational Skills, Willingness to Confront Problems, Physical Ability, Desire for Self-Improvement, Dependability, and Appearance). Factor3 measures cognitive skills (Communication Skills, Problem Solving, Learning Ability, and Judgment Under Pressure).

However, using 0.5 for determining a salient variable-factor relationship does not take sampling variability into account. If the underlying assumptions for the maximum likelihood estimation are approximately satisfied, you can output standard error estimates and the confidence intervals with METHOD=ML. You can then determine the salience of the variable-factor relationship by using the coverage displays. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 2243 for more details.

Figure 35.5 displays the variance explained by each factor and the final communality estimates after the orthogonal rotation. Even though the variances explained by the rotated factors are different from that of the unrotated factor (compare with Figure 35.3), the cumulative variance explained by the common factors remains the same. Note also that the final communalities for variables, as well as the total communality, remain unchanged after rotation. Although rotating a factor solution will not increase or decrease the statistical quality of the factor model, it can simplify the interpretations of the factors and redistribute the variance explained by the factors.

Figure 35.5 Variance Explained and Final Communality Estimates after Rotation

```
                    Variance Explained by Each Factor

                Factor1              Factor2              Factor3

                3.1024330            2.7684489            2.3294622


              Final Communality Estimates: Total = 8.200344

                                                Judgment
  Communication        Problem        Learning        Under        Observational
     Skills            Solving         Ability       Pressure           Skills

    0.64036292       0.59791844      0.61924167      0.63972863       0.69237485

     Willingness
     to Confront        Interest      Interpersonal           Desire for
       Problems        in People       Sensitivity         Self-Improvement

     0.67538695       0.79833968       0.79951357            0.56879171

                                     Physical
       Appearance      Dependability     Ability         Integrity

       0.39403630       0.66056907     0.39504805        0.71903222
```

Syntax: FACTOR Procedure

The following statements are available in the FACTOR procedure:

PROC FACTOR < *options* > ;
    VAR *variables* ;
    PRIORS *communalities* ;
    PARTIAL *variables* ;
    FREQ *variable* ;
    WEIGHT *variable* ;
    BY *variables* ;

Usually only the VAR statement is needed in addition to the PROC FACTOR statement. The descriptions of the BY, FREQ, PARTIAL, PRIORS, VAR, and WEIGHT statements follow the description of the PROC FACTOR statement in alphabetical order.

PROC FACTOR Statement

    PROC FACTOR < *options* > ;

The PROC FACTOR statement invokes the FACTOR procedure. The options listed in Table 35.1 are available in the PROC FACTOR statement.

Table 35.1 Options Available in the PROC FACTOR Statement

Option	Description
Data Set Options	
DATA=	Specifies input SAS data set
OUT=	Specifies output SAS data set
OUTSTAT=	Specifies output data set containing statistical results
TARGET=	Specifies input data set containing the target pattern for rotation
Factor Extraction and Communalities	
HEYWOOD	Sets to 1 any communality greater than 1
METHOD=	Specifies the estimation method
PRIORS=	Specifies the method for computing prior communality estimates
RANDOM=	Specifies the seed for pseudo-random number generation
ULTRAHEYWOOD	Allows communalities to exceed 1
Number of Factors	
MINEIGEN=	Specifies the smallest eigenvalue for retaining a factor
NFACTORS=	Specifies the number of factors to retain
PROPORTION=	Specifies the proportion of common variance in extracted factors
Data Analysis Options	
ALPHA=	Specifies the confidence level for interval construction
COVARIANCE	Requests factoring of the covariance matrix
COVER=	Computes the confidence interval and specifies the coverage reference point
NOINT	Omits the intercept from computing covariances or correlations
SE	Requests the standard error estimates in ML estimation
VARDEF=	Specifies the divisor used in calculating covariances or correlations
WEIGHT	Factors a weighted correlation or covariance matrix
Rotation Method and Properties	
GAMMA=	Specifies the orthomax weight
HKPOWER=	Specifies the power in Harris-Kaiser rotation
NORM=	Specifies the method for row normalization in rotation
NOPROMAXNORM	Turns off row normalization in promax rotation
POWER=	Specifies the power to be used in promax rotation
PREROTATE=	Specifies the prerotation method in promax rotation
RCONVERGE=	Specifies the convergence criterion for rotation cycles
RITER=	Specifies the maximum number of cycles for rotation
ROTATE=	Specifies the rotation method
TAU=	Specifies the oblimin weight

Table 35.1 *continued*

Option	Description
ODS Graphics	
PLOTS=	Specifies ODS Graphics selection
Control Display Output	
ALL	Displays all optional output except plots
CORR	Displays the (partial) correlation matrix
EIGENVECTORS	Displays the eigenvectors of the reduced correlation matrix
FLAG=	Specifies the minimum absolute value to be flagged in the correlation and loading matrices
FUZZ=	Specifies the maximum absolute value to be displayed as missing in the correlation and loading matrices
MSA	Computes Kaiser's measure of sampling adequacy and the related partial correlations
NOPRINT	Suppresses the display of all output
NPLOT=	Specifies the number of factors to be plotted
PLOT	Plots the rotated factor pattern
PLOTREF	Plots the reference structure
PREPLOT	Plots the factor pattern before rotation
PRINT	Displays the input factor pattern or scoring coefficients and related statistics
REORDER	Reorders the rows (variables) of various factor matrices
RESIDUALS	Displays the residual correlation matrix and the associated partial correlation matrix
ROUND	Prints correlation and loading matrices with rounded values
SCORE	Displays the factor scoring coefficients
SCREE	Displays the scree plot of the eigenvalues
SIMPLE	Displays means, standard deviations, and number of observations
Numerical Properties	
CONVERGE=	Specifies the convergence criterion
MAXITER=	Specifies the maximum number of iterations
SINGULAR=	Specifies the singularity criterion
Miscellaneous	
NOCORR	Excludes the correlation matrix from the OUTSTAT= data set
NOBS=	Specifies the number of observations
PARPREFIX=	Specifies the prefix for the residual variables in the output data sets
PREFIX=	Specifies the prefix for naming factors

ALL
> displays all optional output except plots. When the input data set is TYPE=CORR, TYPE=UCORR, TYPE=COV, TYPE=UCOV, or TYPE=FACTOR, simple statistics, correlations, and MSA are not displayed.

ALPHA=*p*
> specifies the level of confidence $1 - p$ for interval construction. By default, $p = 0.05$, corresponding to $1 - p = 95\%$ confidence intervals. If p is greater than one, it is interpreted as a percentage and divided by 100. With multiple confidence intervals to be constructed, the ALPHA= value is applied to each interval construction one at a time. This will not control the coverage probability of the intervals simultaneously. To control familywise coverage probability, you might consider supplying a nonconventional p by using methods such as Bonferroni adjustment.

CONVERGE=*p*

CONV=*p*
> specifies the convergence criterion for the METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, or METHOD=ML option. Iteration stops when the maximum change in the communalities is less than the value of the CONVERGE= option. The default value is 0.001. Negative values are not allowed.

CORR

C
> displays the correlation matrix or partial correlation matrix.

COVARIANCE

COV
> requests factoring of the covariance matrix instead of the correlation matrix. The COV option is effective only with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. For other methods, PROC FACTOR produces the same results with or without the COV option.

COVER < =*p* >

CI < =*p* >
> computes the confidence intervals and optionally specifies the value of factor loading for coverage detection. By default, $p = 0$. The specified value is represented by an asterisk (*) in the coverage display. This is useful for determining the salience of loadings. For example, if COVER=0.4, a display '0*[ ]' indicates that the entire confidence interval is above 0.4, implying strong evidence for the salience of the loading. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 2243 for more details.

DATA=*SAS-data-set*
> specifies the input data set, which can be an ordinary SAS data set or a specially structured SAS data set as described in the section "Input Data Set" on page 2239. If the DATA= option is omitted, the most recently created SAS data set is used.

EIGENVECTORS

EV

displays the eigenvectors of the reduced correlation matrix, of which the diagonal elements are replaced with the communality estimates. When METHOD=ML, the eigenvectors are for the weighted reduced correlation matrix. PROC FACTOR chooses the solution that makes the sum of the elements of each eigenvector nonnegative. If the sum of the elements is equal to zero, then the sign depends on how the number is rounded off.

FLAG=p

flags absolute values larger than p with an asterisk in the correlation and loading matrices. Negative values are not allowed for p. Values printed in the matrices are multiplied by 100 and rounded to the nearest integer (see the ROUND option). The FLAG= option has no effect when standard errors or confidence intervals are also printed.

FUZZ=p

prints correlations and factor loadings with absolute values less than p printed as missing. For partial correlations, the FUZZ= value is divided by 2. For residual correlations, the FUZZ= value is divided by 4. The exact values in any matrix can be obtained from the OUTSTAT= and ODS output data sets. Negative values are not allowed. The FUZZ= option has no effect when standard errors or confidence intervals are also printed.

GAMMA=p

specifies the orthomax weight used with the option ROTATE=ORTHOMAX or PRE-ROTATE=ORTHOMAX. Alternatively, you can use ROTATE=ORTHOMAX(p) with p representing the orthomax weight. There is no restriction on valid values for the orthomax weight, although the most common values are between zero and the number of variables. The default GAMMA= value is one, resulting in the varimax rotation. See the section "Simplicity Functions for Rotations" on page 2244 for more details.

HEYWOOD

HEY

sets to 1 any communality greater than 1, allowing iterations to proceed. See the section "Heywood Cases and Other Anomalies about Communality Estimates" on page 2248 for a discussion of Heywood cases.

HKPOWER=p

HKP=p

specifies the power of the square roots of the eigenvalues used to rescale the eigenvectors for Harris-Kaiser (ROTATE=HK) rotation, assuming that the factors are extracted by the principal factor method. If the principal factor method is not used for factor extraction, the eigenvectors are replaced by the normalized columns of the unrotated factor matrix, and the eigenvalues are replaced by the column normalizing constants. HKPOWER= values between 0.0 and 1.0 are reasonable. The default value is 0.0, yielding the independent cluster solution, in which each variable tends to have a large loading on only one factor. An HKPOWER= value of 1.0 is equivalent to an orthogonal rotation, with the varimax rotation as the default. You can also specify the HKPOWER= option with ROTATE=QUARTIMAX, ROTATE=BIQUARTIMAX, ROTATE=EQUAMAX, or ROTATE=ORTHOMAX, and so on. The only restriction is that the Harris-Kaiser rotation must be associated with an orthogonal rotation.

MAXITER=*n*

    specifies the maximum number of iterations for factor extraction. You can use the MAXITER= option with the PRINIT, ULS, ALPHA, or ML method. The default is 30.

METHOD=*name*

M=*name*

    specifies the method for extracting factors. The default is METHOD=PRINCIPAL unless the DATA= data set is TYPE=FACTOR, in which case the default is METHOD=PATTERN. Valid values for *name* are as follows:

ALPHA | A   produces alpha factor analysis.

HARRIS | H   yields Harris component analysis of $\mathbf{S}^{-1}\mathbf{RS}^{-1}$ (Harris 1962), a noniterative approximation to canonical component analysis.

IMAGE | I   yields principal component analysis of the image covariance matrix, not the image analysis of Kaiser (1963, 1970) or Kaiser and Rice (1974). A nonsingular correlation matrix is required.

ML | M   performs maximum likelihood factor analysis with an algorithm due, except for minor details, to Fuller (1987). The option METHOD=ML requires a nonsingular correlation matrix.

PATTERN   reads a factor pattern from a TYPE=FACTOR, TYPE=CORR, TYPE=UCORR, TYPE=COV, or TYPE=UCOV data set. If you create a TYPE=FACTOR data set in a DATA step, only observations containing the factor pattern (_TYPE_='PATTERN') and, if the factors are correlated, the interfactor correlations (_TYPE_='FCORR') are required.

PRINCIPAL | PRIN | P   yields principal component analysis if no PRIORS option or statement is used or if you specify PRIORS=ONE; if you specify a PRIORS statement or a PRIORS= value other than PRIORS=ONE, a principal factor analysis is performed.

PRINIT   yields iterated principal factor analysis.

SCORE   reads scoring coefficients (_TYPE_='SCORE') from a TYPE=FACTOR, TYPE=CORR, TYPE=UCORR, TYPE=COV, or TYPE=UCOV data set. The data set must also contain either a correlation or a covariance matrix. Scoring coefficients are also displayed if you specify the OUT= option.

ULS | U   produces unweighted least squares factor analysis.

MINEIGEN=*p*

MIN=*p*

    specifies the smallest eigenvalue for which a factor is retained. If you specify two or more of the MINEIGEN=, NFACTORS=, and PROPORTION= options, the number of factors retained is the minimum number satisfying any of the criteria. The MINEIGEN= option cannot be used with either the METHOD=PATTERN or the METHOD=SCORE option. Negative values are not allowed. The default is 0 unless you omit both the NFACTORS= and the PROPORTION= options and one of the following conditions holds:

    ● If you specify the METHOD=ALPHA or METHOD=HARRIS option, then MINEIGEN=1.

- If you specify the METHOD=IMAGE option, then

$$\text{MINEIGEN} = \frac{\text{total image variance}}{\text{number of variables}}$$

- For any other METHOD= specification, if prior communality estimates of 1.0 are used, then

$$\text{MINEIGEN} = \frac{\text{total weighted variance}}{\text{number of variables}}$$

When an unweighted correlation matrix is factored, this value is 1.

MSA

produces the partial correlations between each pair of variables controlling for all other variables (the negative anti-image correlations) and Kaiser's measure of sampling adequacy (Kaiser 1970; Kaiser and Rice 1974; Cerny and Kaiser 1977).

NFACTORS=*n*
NFACT=*n*
N=*n*

specifies the maximum number of factors to be extracted and determines the amount of memory to be allocated for factor matrices. The default is the number of variables. Specifying a number that is small relative to the number of variables can substantially decrease the amount of memory required to run PROC FACTOR, especially with oblique rotations. If you specify two or more of the NFACTORS=, MINEIGEN=, and PROPORTION= options, the number of factors retained is the minimum number satisfying any of the criteria. If you specify the option NFACTORS=0, eigenvalues are computed, but no factors are extracted. If you specify the option NFACTORS=–1, neither eigenvalues nor factors are computed. You can use the NFACTORS= option with the METHOD=PATTERN or METHOD=SCORE option to specify a smaller number of factors than are present in the data set.

NOBS=*n*

specifies the number of observations. If the DATA= input data set is a raw data set, *nobs* is defined by default to be the number of observations in the raw data set. The NOBS= option overrides this default definition. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, the number of observations can be specified either by using the NOBS= option in the PROC FACTOR statement or by including a _TYPE_='N' observation in the DATA= input data set.

NOCORR

prevents the correlation matrix from being transferred to the OUTSTAT= data set when you specify the METHOD=PATTERN option. The NOCORR option greatly reduces memory requirements when there are many variables but few factors. The NOCORR option is not effective if the correlation matrix is required for other requested output; for example, if the scores or the residual correlations are displayed (for example, by using the SCORE, RESIDUALS, or ALL option).

NOINT

omits the intercept from the analysis; covariances or correlations are not corrected for the mean.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

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

NORM=COV | KAISER | NONE | RAW | WEIGHT
 specifies the method for normalizing the rows of the factor pattern for rotation. If you specify the option NORM=KAISER, Kaiser's normalization is used ($\sum_j p_{ij}^2 = 1$). If you specify the option NORM=WEIGHT, the rows are weighted by the Cureton-Mulaik technique (Cureton and Mulaik 1975). If you specify the option NORM=COV, the rows of the pattern matrix are rescaled to represent covariances instead of correlations. If you specify the option NORM=NONE or NORM=RAW, normalization is not performed. The default is NORM=KAISER.

NPLOTS | NPLOT=*n*
 specifies the number of factors to be plotted. The default is to plot all factors. The smallest allowable value is 2. If you specify the option NPLOTS=*n*, all pairs of the first *n* factors are plotted, producing a total of $n(n-1)/2$ plots.

OUT=*SAS-data-set*
 creates a data set containing all the data from the DATA= data set plus variables called Factor1, Factor2, and so on, containing estimated factor scores. The DATA= data set must contain multivariate data, not correlations or covariances. You must also specify the NFACTORS= option to determine the number of factor score variables. If you specify partial variables in the PARTIAL statement, the OUT= data set will also contain the residual variables that are used for factor analysis. The output data set is described in detail in the section "Output Data Sets" on page 2241. 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*.

OUTSTAT=*SAS-data-set*
 specifies an output data set containing most of the results of the analysis. The output data set is described in detail in the section "Output Data Sets" on page 2241. 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*.

PARPREFIX=*name*
 specifies the prefix for the residual variables in the OUT= and the OUTSTAT= data sets when partial variables are specified in the PARTIAL statement.

PLOT
 plots the factor pattern after rotation. This option produces printer plots. High-quality ODS graphical plots for factor patterns can be requested with the PLOTS=LOADINGS or PLOTS=INITLOADINGS option.

PLOTREF
 plots the reference structure instead of the default factor pattern after oblique rotation.

PLOTS < (*global-plot-options*) > = *plot-request* < (*options*) >
PLOTS < (*global-plot-options*) > = (*plot-request* < (*options*) > < ... *plot-request* < (*options*) > >)
 specifies one or more ODS graphical plots in PROC FACTOR. When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*. Here are some examples:

```
plots=all
plots(flip)=loadings
plots=(loadings(flip) scree(unpack))
```

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

```
ods graphics on;
proc factor plots=all;
run;
ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 600 in Chapter 21, "Statistical Graphics Using ODS."

For an example containing graphical displays of factor analysis results, see Example 35.2.

The following table shows the available *plot-requests* and their available suboptions:

Plot-Request	**Plot Description**	**Suboptions**
ALL	all available plots	all
INITLOADINGS	unrotated factor loadings	CIRCLE=, FLIP, NPLOTS=, PLOTREF, and VECTOR
LOADINGS	rotated factor loadings	CIRCLE=, FLIP, NPLOTS=, PLOTREF, and VECTOR
NONE	no ODS graphical plots	
PRELOADINGS	prerotated factor loadings	CIRCLE=, FLIP, NPLOTS=, PLOTREF, and VECTOR
SCREE	scree and variance explained	UNPACK

The following are the available *global-plot-options* or *options* for plots:

CIRCLE | CIRCLES < = *numbers* >   draws circular reference lines in scatter plots or vector plots of factor loadings. You can specify the locations of the circular reference lines in the *numbers* list. Each number indicates the proportion or percentage of area of the unit circle that is enclosed by the specified circle. Each of the *numbers* must lie between 0 and 100, inclusively. When a number is between 0 and 1 (inclusively), it is interpreted as a proportion; otherwise, it is interpreted as a percentage. The maximum number of circles is 5.

The CIRCLE option applies to the scatter or vector plots requested by the INIT-LOADINGS, LOADINGS, and PRELOADINGS options. By default, a unit-circle, which represents 100% of the total area, is drawn for the vector plots. However, no circle will be drawn for scatter plots unless the CIRCLE option is specified. Two special cases for this option are: (1) With no *numbers* following the CIRCLE option, a 100% circle will be drawn. (2) With CIRCLE=0, no circle will be drawn. This special case is primarily used to turn off the default unit-circle in vector plots.

FLIP   switches the X and Y axes. It applies to the INITLOADINGS, LOADINGS, and PRELOADINGS *plot-requests*.

NPLOT | NPLOTS=*n* specifies the number of factors *n* ($n \geq 2$) to be plotted. It applies to the INIT-LOADINGS, LOADINGS, and PRELOADINGS *plot-requests*. Since this option can also be specified in the PROC FACTOR statement, the final value of *n* is determined by the following steps. The NPLOTS= value of the PROC FACTOR is read first. If the NPLOTS= option is specified as a *global-plot-option*, the value of *n* will be updated. Then, if the NPLOTS= option is again specified in an individual *plot-request*, the value will be updated again for that individual *plot-request*. For example, in the following statement, four factors are extracted with the N=4 option:

```
proc factor n=4 nplots=3 plots(nplots=4)=
                 (loadings preloadings(nplots=2));
```

Initially, plots of the first three factors are specified with the NPLOTS=3 option. When you are producing ODS graphical plots, the *global-plot-option* NPLOTS=4 is used. As a result, the LOADINGS *plot-request* will produce plots for all pairs of the first 4 factors. However, because the NPLOTS=2 is specified locally for the PRELOADINGS *plot-request*, it will produce a prerotated factor loading plot for the first two factors only.

The default NPLOTS= value is 5 or the total number of factors (*m*), whichever is smaller. If you specify an NPLOTS= value that is greater than *m*, NPLOTS=*m* will be used.

PLOTREF plots the reference structures rather than the factor pattern loadings. It applies to the INITLOADINGS, LOADINGS, and PRELOADINGS *plot-requests* when the factor solution is oblique. This option can also be set globally as an option in the PROC FACTOR statement.

UNPACK plots component graphs separately. It applies to the SCREE *plot-request* only.

VECTOR plots loadings in vector form. It applies to the INITLOADINGS, LOADINGS, and PRELOADINGS *plot-requests* when the factor solution is orthogonal. For oblique solutions, the VECTOR option is ignored and the default scatter plots for factor loadings or reference structures are displayed.

Be aware that the PLOT option in the PROC FACTOR statement requests only the printer plots of factor loadings. The current option PLOTS= or PLOT=, however, is for ODS graphical plots.

You can specify options for the requested ODS graphical plots as *global-plot-options* or as local *options*. *Global-plot-options* apply to all appropriate individual *plot-requests* specified. For example, because the SCREE plot is not subject to axes flipping, the following two specifications are equivalent:

```
plots(flip)=(loadings preloadings scree)
plots=(loadings(flip) preloadings(flip) scree)
```

Options specified locally after each *plot-request* apply to that plot-request only. For example, consider the following specification:

```
plots=(scree(unpack) loadings(plotref) preloadings(flip))
```

The FLIP option applies to the PRELOADINGS *plot-request* but not the LOADINGS *plot-request*; the PLOTREF option applies to the LOADINGS *plot-request* but not the PRELOADINGS *plot-request*; and the UNPACK option applies to the SCREE *plot-request* only.

POWER=*n*

specifies the power to be used in computing the target pattern for the option ROTATE=PROMAX. Valid values must be integers ≥ 1. The default value is 3. You can also specify the POWER= value in the ROTATE= option—for example, ROTATE=PROMAX(4).

PREFIX=*name*

specifies a prefix for naming the factors. By default, the names are Factor1, Factor2, ..., Factor*n*. If you specify PREFIX=ABC, the factors are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the current name length defined by the VALIDVARNAME= system option.

PREPLOT

plots the factor pattern before rotation. This option produces printer plots. High-quality ODS graphical plots for factor patterns can be requested with the PLOTS=PRELOADINGS option.

PREROTATE=*name*

PRE=*name*

specifies the prerotation method for the option ROTATE=PROMAX. Any rotation method other than PROMAX or PROCRUSTES can be used. The default is PREROTATE=VARIMAX. If a previously rotated pattern is read using the option METHOD=PATTERN, you should specify the PREROTATE=NONE option.

PRINT

displays the input factor pattern or scoring coefficients and related statistics. In oblique cases, the reference and factor structures are computed and displayed. The PRINT option is effective only with the option METHOD=PATTERN or METHOD=SCORE.

PRIORS=*name*

specifies a method for computing prior communality estimates. You can specify numeric values for the prior communality estimates by using the PRIORS statement. Valid values for *name* are as follows:

ASMC | A   sets the prior communality estimates proportional to the squared multiple correlations but adjusted so that their sum is equal to that of the maximum absolute correlations (Cureton 1968).

INPUT | I   reads the prior communality estimates from the first observation with either _TYPE_='PRIORS' or _TYPE_='COMMUNAL' in the DATA= data set (which cannot be TYPE=DATA).

MAX | M   sets the prior communality estimate for each variable to its maximum absolute correlation with any other variable.

ONE | O   sets all prior communalities to 1.0.

RANDOM | R   sets the prior communality estimates to pseudo-random numbers uniformly distributed between 0 and 1.

SMC | S   sets the prior communality estimate for each variable to its squared multiple correlation with all other variables.

The default prior communality estimates are as follows:

METHOD=	PRIORS=
PRINCIPAL	ONE
PRINIT	ONE
ALPHA	SMC
ULS	SMC
ML	SMC
HARRIS	(not applicable)
IMAGE	(not applicable)
PATTERN	(not applicable)
SCORE	(not applicable)

By default, the options METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, and METHOD=ML stop iterating and set the number of factors to 0 if an estimated communality exceeds 1. The options HEYWOOD and ULTRAHEYWOOD allow processing to continue.

PROPORTION=p

PERCENT=p

P=p

specifies the proportion of common variance to be accounted for by the retained factors. The proportion of common variance is computed using the total prior communality estimates as the basis. If the value is greater than one, it is interpreted as a percentage and divided by 100. The options PROPORTION=0.75 and PERCENT=75 are equivalent. The default value is 1.0 or 100%. You cannot specify the PROPORTION= option with the METHOD=PATTERN or METHOD=SCORE option. If you specify two or more of the PROPORTION=, NFACTORS=, and MINEIGEN= options, the number of factors retained is the minimum number satisfying any of the criteria.

RANDOM=n

specifies a positive integer as a starting value for the pseudo-random number generator for use with the option PRIORS=RANDOM. If you do not specify the RANDOM= option, the time of day is used to initialize the pseudo-random number sequence. Valid values must be integers ≥ 1.

RCONVERGE=p

RCONV=p

specifies the convergence criterion for rotation cycles. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K = \max(1, |f_{old}|)$ is a scaling factor, and ϵ is 1E–9 by default and is modified by the RCONVERGE= value.

REORDER

RE

causes the rows (variables) of various factor matrices to be reordered on the output. Variables with their highest absolute loading (reference structure loading for oblique rotations) on the first factor are displayed first, from largest to smallest loading, followed by variables with their highest absolute loading on the second factor, and so on. The order of the variables in the output data set is not affected. The factors are not reordered.

RESIDUALS

RES

displays the residual correlation matrix and the associated partial correlation matrix. The diagonal elements of the residual correlation matrix are the unique variances.

RITER=*n*

specifies the maximum number of cycles *n* for factor rotation. Except for promax and Procrustes, you can use the RITER= option with all rotation methods. The default *n* is the maximum between 10 times the number of variables and 100.

ROTATE=*name*

R=*name*

specifies the rotation method. The default is ROTATE=NONE.

Valid *names* for orthogonal rotations are as follows:

BIQUARTIMAX | BIQMAX   specifies orthogonal biquartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(.5).

EQUAMAX | E   specifies orthogonal equamax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of factors*/2.

FACTORPARSIMAX | FPA   specifies orthogonal factor parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of variables*.

NONE | N   specifies that no rotation be performed, leaving the original orthogonal solution.

ORTHCF(*p1,p2*) | ORCF(*p1,p2*)   specifies the orthogonal Crawford-Ferguson rotation with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 2244.

ORTHGENCF(*p1,p2,p3,p4*) | ORGENCF(*p1,p2,p3,p4*)   specifies the orthogonal generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 2244.

ORTHOMAX<(*p*)> | ORMAX<(*p*)>   specifies the orthomax rotation with orthomax weight *p*. If ROTATE=ORTHOMAX is used, the default *p* value is 1 unless specified otherwise in the GAMMA= option. Alternatively, ROTATE=ORTHOMAX(*p*) specifies *p* as the orthomax weight or the GAMMA= value. See the definition of the orthomax weight in the section "Simplicity Functions for Rotations" on page 2244.

PARSIMAX | PA   specifies orthogonal parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with

$$\text{GAMMA} = \frac{nvar \times (nfact - 1)}{nvar + nfact - 2}$$

where *nvar* is the number of variables and *nfact* is the number of factors.

QUARTIMAX | QMAX | Q   specifies orthogonal quartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(0).

VARIMAX | V   specifies orthogonal varimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=1.

Valid *names* for oblique rotations are as follows:

BIQUARTIMIN | BIQMIN   specifies biquartimin rotation. It corresponds to the specification ROTATE=OBLIMIN(.5) or ROTATE=OBLIMIN with TAU=0.5.

COVARIMIN | CVMIN   specifies covarimin rotation. It corresponds to the specification ROTATE=OBLIMIN(1) or ROTATE=OBLIMIN with TAU=1.

HK<(*p*)> | H<(*p*)>   specifies Harris-Kaiser case II orthoblique rotation. When specifying this option, you can use the HKPOWER= option to set the power of the square roots of the eigenvalues by which the eigenvectors are scaled, assuming that the factors are extracted by the principal factor method. For other extraction methods, the unrotated factor pattern is column normalized. The power is then applied to the column normalizing constants, instead of the eigenvalues. You can also use RO-TATE=HK(*p*), with *p* representing the HKPOWER= value. The default associated orthogonal rotation with ROTATE=HK is the varimax rotation without Kaiser normalization. You might associate the Harris-Kaiser with other orthogonal rotations by using the ROTATE= option together with the HKPOWER= option.

OBBIQUARTIMAX | OBIQMAX   specifies oblique biquartimax rotation.

OBEQUAMAX | OE   specifies oblique equamax rotation.

OBFACTORPARSIMAX | OFPA   specifies oblique factor parsimax rotation.

OBLICF(*p1*,*p2*) | OBCF(*p1*,*p2*)   specifies the oblique Crawford-Ferguson rotation with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 2244.

OBLIGENCF(*p1*,*p2*,*p3*,*p4*) | OBGENCF(*p1*,*p2*,*p3*,*p4*)   specifies the oblique generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 2244.

OBLIMIN<(*p*)> | OBMIN<(*p*)>   specifies the oblimin rotation with oblimin weight *p*. If ROTATE=OBLIMIN is used, the default *p* value is zero unless specified otherwise in the TAU= option. Alternatively, ROTATE=OBLIMIN(*p*) specifies *p* as the oblimin weight or the TAU= value. See the definition of the oblimin weight in the section "Simplicity Functions for Rotations" on page 2244.

OBPARSIMAX | OPA   specifies oblique parsimax rotation.

OBQUARTIMAX | OQMAX   specifies oblique quartimax rotation. This is the same as the QUAR-
TIMIN method.

OBVARIMAX | OV   specifies oblique varimax rotation.

PROCRUSTES   specifies oblique Procrustes rotation with the target pattern provided by the TAR-
GET= data set. The unrestricted least squares method is used with factors scaled
to unit variance after rotation.

PROMAX<(p)> | P<(p)>   specifies oblique promax rotation. You can use the PREROTATE= op-
tion to set the desirable prerotation method, orthogonal or oblique. When used
with ROTATE=PROMAX, the POWER= option lets you specify the power for
forming the target. You can also use ROTATE=PROMAX(p), where p represents
the POWER= value.

QUARTIMIN | QMIN   specifies quartimin rotation.   It is the same as the oblique quartimax
method.   It also corresponds to the specification ROTATE=OBLIMIN(0) or
ROTATE=OBLIMIN with TAU=0.

ROUND

prints correlation and loading matrices with entries multiplied by 100 and rounded to the nearest
integer. The exact values can be obtained from the OUTSTAT= and ODS output data sets. The
ROUND option also flags absolute values larger than the FLAG= value with an asterisk in correlation
and loading matrices (see the FLAG= option). If the FLAG= option is not specified, the root mean
square of all the values in the matrix printed is used as the default FLAG= value. The ROUND option
has no effect when standard errors or confidence intervals are also printed.

SCORE

displays the factor scoring coefficients. The squared multiple correlation of each factor with the vari-
ables is also displayed except in the case of unrotated principal components. The SCORE option also
outputs the factor scoring coefficients in the _TYPE_=SCORE or _TYPE_=USCORE observations
in the OUTSTAT= data set. Unless you specify the NOINT option in PROC FACTOR, the scoring
coefficients should be applied to standardized variables—variables that are centered by subtracting
the original variable means and then divided by the original variable standard deviations. With the
NOINT option, the scoring coefficients should be applied to data without centering.

SCREE

displays a scree plot of the eigenvalues (Cattell 1966, 1978; Cattell and Vogelman 1977; Horn and
Engstrom 1979). This option produces printer plots. High-quality scree plots can be requested with
the PLOTS=SCREE option.

SE
STDERR

computes standard errors for various classes of unrotated and rotated solutions under the maximum
likelihood estimation.

SIMPLE
S

displays means, standard deviations, and the number of observations.

SINGULAR=p

SING=p

>   specifies the singularity criterion, where $0 < p < 1$. The default value is 1E–8.

TARGET=SAS-data-set

>   specifies an input data set containing the target pattern for Procrustes rotation (see the description of the ROTATE= option). The TARGET= data set must contain variables with the same names as those being factored. Each observation in the TARGET= data set becomes one column of the target factor pattern. Missing values are treated as zeros. The _NAME_ and _TYPE_ variables are not required and are ignored if present.

TAU=p

>   specifies the oblimin weight used with the option ROTATE=OBLIMIN or PREROTATE=OBLIMIN. Alternatively, you can use ROTATE=OBLIMIN(p) with p representing the oblimin weight. There is no restriction on valid values for the oblimin weight, although for practical purposes a negative or zero value is recommended. The default TAU= value is 0, resulting in the quartimin rotation. See the section "Simplicity Functions for Rotations" on page 2244 for more details.

ULTRAHEYWOOD

ULTRA

>   allows communalities to exceed 1. The ULTRAHEYWOOD option can cause convergence problems because communalities can become extremely large, and ill-conditioned Hessians might occur. See the section "Heywood Cases and Other Anomalies about Communality Estimates" on page 2248 for a discussion of Heywood cases.

VARDEF=DF | N | WDF | WEIGHT | WGT

>   specifies the divisor used in the calculation of variances and covariances. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table where $i=0$ if the NOINT option is used and $i=1$ otherwise, and where k is the number of partial variables specified in the PARTIAL statement.

Value	Description	Divisor
DF	degrees of freedom	$n - k - i$
N	number of observations	$n - k$
WDF	sum of weights DF	$\sum_i w_i - k - i$
WEIGHT | WGT	sum of weights	$\sum_i w_i - k$

WEIGHT

>   factors a weighted correlation or covariance matrix. The WEIGHT option can be used only with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. The input data set must be of type CORR, UCORR, COV, UCOV, or FACTOR, and the variable weights are obtained from an observation with _TYPE_='WEIGHT'.

BY Statement

> **BY** *variables* **;**

You can specify a BY statement with PROC FACTOR 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 for the FACTOR 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).

If you specify the TARGET= option and the TARGET= data set does not contain any of the BY variables, then the entire TARGET= data set is used as a Procrustean target for each BY group in the DATA= data set.

If the TARGET= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the TARGET= data set as in the DATA= data set, then PROC FACTOR displays an error message and stops.

If all the BY variables appear in the TARGET= data set with the same type and length as in the DATA= data set, then each BY group in the TARGET= data set is used as a Procrustean target for the corresponding BY group in the DATA= data set. The BY groups in the TARGET= data set must be in the same order as in the DATA= data set. If you specify the NOTSORTED option in the BY statement, there must be identical BY groups in the same order in both data sets. If you do not specify the NOTSORTED option, some BY groups can appear in one data set but not in the other.

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 a variable in the data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears *n* times, where *n* is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, the value is truncated to an integer.

The WEIGHT and FREQ statements have a similar effect, except in determining the number of observations for significance tests.

PARTIAL Statement

> **PARTIAL** *variables* ;

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables that are used to partial out the variables in the analysis.

PRIORS Statement

> **PRIORS** *communalities* ;

The PRIORS statement specifies numeric values between 0.0 and 1.0 for the prior communality estimates for each variable. The first numeric value corresponds to the first variable in the VAR statement, the second value to the second variable, and so on. The number of numeric values must equal the number of variables. For example:

```
proc factor;
   var      x  y  z;
   priors .7 .8 .9;
run;
```

You can specify various methods for computing prior communality estimates with the PRIORS= option in the PROC FACTOR statement. See the description of that option for more information about the default prior communality estimates.

VAR Statement

> **VAR** *variables* ;

The VAR statement specifies the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not specified in other statements are analyzed.

WEIGHT Statement

> **WEIGHT** *variable* ;

If you want to use relative weights for each observation in the input data set, specify a variable containing weights in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. If a variable value is negative or is missing, it is excluded from the analysis.

Details: FACTOR Procedure

Input Data Set

The FACTOR procedure can read an ordinary SAS data set containing raw data or a special data set specified as a TYPE=CORR, TYPE=UCORR, TYPE=SSCP, TYPE=COV, TYPE=UCOV, or TYPE=FACTOR data set containing previously computed statistics. A TYPE=CORR data set can be created by the CORR procedure or various other procedures such as the PRINCOMP procedure. It contains means, standard deviations, the sample size, the correlation matrix, and possibly other statistics if it is created by some procedure other than PROC CORR. A TYPE=COV data set is similar to a TYPE=CORR data set but contains a covariance matrix. A TYPE=UCORR or TYPE=UCOV data set contains a correlation or covariance matrix that is not corrected for the mean. The default VAR variable list does not include Intercept if the DATA= data set is TYPE=SSCP. If the Intercept variable is explicitly specified in the VAR statement with a TYPE=SSCP data set, the NOINT option is activated. A TYPE=FACTOR data set can be created by the FACTOR procedure and is described in the section "Output Data Sets" on page 2241.

If your data set has many observations and you plan to run FACTOR several times, you can save computer time by first creating a TYPE=CORR data set and using it as input to PROC FACTOR, as in the following statements:

```
proc corr data=raw out=correl;      /* create TYPE=CORR data set */
proc factor data=correl method=ml; /* maximum likelihood        */
proc factor data=correl;            /* principal components      */
```

The data set created by the CORR procedure is automatically given the TYPE=CORR data set option, so you do not have to specify TYPE=CORR. However, if you use a DATA step with a SET statement to modify the correlation data set, you must use the TYPE=CORR attribute in the new data set. You can use a VAR statement with PROC FACTOR when reading a TYPE=CORR data set to select a subset of the variables or change the order of the variables.

Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each correlation from all observations that have values present for the pair of variables involved (pairwise deletion). The resulting correlation matrix might have negative eigenvalues. If you specify the NOMISS option with the CORR procedure, observations with any missing values are completely omitted from the calculations (listwise deletion), and there is no danger of negative eigenvalues.

PROC FACTOR can also create a TYPE=FACTOR data set, which includes all the information in a TYPE=CORR data set, and use it for repeated analyses. For a TYPE=FACTOR data set, the default value of the METHOD= option is PATTERN. The following PROC FACTOR statements produce the same results as the previous example:

```
proc factor data=raw method=ml outstat=fact; /* max. likelihood */
proc factor data=fact method=prin;      /* principal components */
```

You can use a TYPE=FACTOR data set to try several different rotation methods on the same data without repeatedly extracting the factors. In the following example, the second and third PROC FACTOR statements use the data set fact created by the first PROC FACTOR statement:

```
proc factor data=raw outstat=fact; /* principal components */
proc factor rotate=varimax;          /* varimax rotation     */
proc factor rotate=quartimax;        /* quartimax rotation   */
```

You can create a TYPE=CORR, TYPE=UCORR, or TYPE=FACTOR data set in a DATA step for PROC FACTOR to read as input. For example, in the following a TYPE=CORR data set is created and is read as input data set by the subsequent PROC FACTOR statement:

```
data correl(type=corr);
   _TYPE_='CORR';
   input _NAME_ $ x y z;
   datalines;
x  1.0  .   .
y   .7 1.0  .
z   .5   .4 1.0
;
proc factor;
run;
```

Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement and include the _TYPE_ and _NAME_ variables. In a TYPE=CORR data set, only the correlation matrix (_TYPE_='CORR') is necessary. It can contain missing values as long as every pair of variables has at least one nonmissing value.

You can also create a TYPE=FACTOR data set containing only a factor pattern (_TYPE_='PATTERN') and use the FACTOR procedure to rotate it, as these statements show:

```
data pat(type=factor);
   _TYPE_='PATTERN';
   input _NAME_ $ x y z;
   datalines;
factor1  .5  .7  .3
factor2  .8  .2  .8
;
proc factor rotate=promax prerotate=none;
run;
```

If the input factors are oblique, you must also include the interfactor correlation matrix with _TYPE_='FCORR', as shown here:

```
data pat(type=factor);
   input _TYPE_ $ _NAME_ $ x y z;
   datalines;
pattern factor1  .5  .7  .3
pattern factor2  .8  .2  .8
fcorr   factor1 1.0  .2  .
fcorr   factor2  .2 1.0  .
;
proc factor rotate=promax prerotate=none;
run;
```

Some procedures, such as the PRINCOMP and CANDISC procedures, produce TYPE=CORR or TYPE=UCORR data sets containing scoring coefficients (_TYPE_='SCORE' or _TYPE_= 'USCORE'). These coefficients can be input to PROC FACTOR and rotated by using the METHOD=SCORE option, as in the following statements:

```
proc princomp data=raw n=2 outstat=prin;
run;
proc factor data=prin method=score rotate=varimax;
run;
```

Notice that the input data set prin must contain the correlation matrix as well as the scoring coefficients.

Output Data Sets

The OUT= Data Set

The OUT= data set contains all the data in the DATA= data set plus new variables called Factor1, Factor2, and so on, containing estimated factor scores. Each estimated factor score is computed as a linear combination of the standardized values of the variables that are factored. The coefficients are always displayed if the OUT= option is specified, and they are labeled "Standardized Scoring Coefficients."

If partial variables are specified in the PARTIAL statement, the factor analysis is on the residuals of the variables, which are regressed on the partial variables. In this case, the OUT= data set also contains the (unstandardized) residuals, which are prefixed by R_ by default. For example, the residual of variable X is named R_X in the OUT= data set. You might also assign the prefix by the PARPREFIX= option. Because the residuals are factor-analyzed, the estimated factor scores are computed as linear combinations of the standardized values of the residuals, but not the original variables.

The OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR or TYPE=UCORR data set produced by the CORR procedure, but it is a TYPE=FACTOR data set and it contains many results in addition to those produced by PROC CORR. The OUTSTAT= data set contains observations with _TYPE_='UCORR' and _TYPE_='USTD' if you specify the NOINT option.

The output data set contains the following variables:

- the BY variables, if any

- two new character variables, _TYPE_ and _NAME_

- the variables analyzed—those in the VAR statement, or, if there is no VAR statement, all numeric variables not listed in any other statement. If partial variables are specified in the PARTIAL statement, the residuals are included instead. By default, the residual variable names are prefixed by R_, unless you specify something different in the PARPREFIX= option.

Each observation in the output data set contains some type of statistic as indicated by the _TYPE_ variable. The _NAME_ variable is blank except where otherwise indicated. The values of the _TYPE_ variable are as follows:

MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations

N	sample size
CORR	correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the correlation matrix.
UCORR	uncorrected correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the uncorrected correlation matrix.
IMAGE	image coefficients. The _NAME_ variable contains the name of the variable corresponding to each row of the image coefficient matrix.
IMAGECOV	image covariance matrix. The _NAME_ variable contains the name of the variable corresponding to each row of the image covariance matrix.
COMMUNAL	final communality estimates
PRIORS	prior communality estimates, or estimates from the last iteration for iterative methods
WEIGHT	variable weights
SUMWGT	sum of the variable weights
EIGENVAL	eigenvalues
UNROTATE	unrotated factor pattern. The _NAME_ variable contains the name of the factor.
SE_UNROT	standard error estimates for the unrotated loadings. The _NAME_ variable contains the name of the factor.
RESIDUAL	residual correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the residual correlation matrix.
PRETRANS	transformation matrix from prerotation. The _NAME_ variable contains the name of the factor.
PREFCORR	prerotated interfactor correlations. The _NAME_ variable contains the name of the factor.
SE_PREFC	standard error estimates for prerotated interfactor correlations. The _NAME_ variable contains the name of the factor.
PREROTAT	prerotated factor pattern. The _NAME_ variable contains the name of the factor.
SE_PREPA	standard error estimates for the prerotated loadings. The _NAME_ variable contains the name of the factor.
PRERCORR	prerotated reference axis correlations. The _NAME_ variable contains the name of the factor.
PREREFER	prerotated reference structure. The _NAME_ variable contains the name of the factor.
PRESTRUC	prerotated factor structure. The _NAME_ variable contains the name of the factor.
SE_PREST	standard error estimates for prerotated structure loadings. The _NAME_ variable contains the name of the factor.
PRESCORE	prerotated scoring coefficients. The _NAME_ variable contains the name of the factor.
TRANSFOR	transformation matrix from rotation. The _NAME_ variable contains the name of the factor.
FCORR	interfactor correlations. The _NAME_ variable contains the name of the factor.
SE_FCORR	standard error estimates for interfactor correlations. The _NAME_ variable contains the name of the factor.

PATTERN	factor pattern. The _NAME_ variable contains the name of the factor.
SE_PAT	standard error estimates for the rotated loadings. The _NAME_ variable contains the name of the factor.
RCORR	reference axis correlations. The _NAME_ variable contains the name of the factor.
REFERENC	reference structure. The _NAME_ variable contains the name of the factor.
STRUCTUR	factor structure. The _NAME_ variable contains the name of the factor.
SE_STRUC	standard error estimates for structure loadings. The _NAME_ variable contains the name of the factor.
SCORE	scoring coefficients to be applied to standardized variables if the SCORE option is specified on the PROC FACTOR statement. The _NAME_ variable contains the name of the factor.
USCORE	scoring coefficients to be applied without subtracting the mean from the raw variables if the SCORE option is specified on the PROC FACTOR statement. The _NAME_ variable contains the name of the factor.

Confidence Intervals and the Salience of Factor Loadings

The traditional approach to determining salient loadings (loadings that are considered large in absolute values) employs rules of thumb such as 0.3 or 0.4. However, this does not use the statistical evidence efficiently. The asymptotic normality of the distribution of factor loadings enables you to construct confidence intervals to gauge the salience of factor loadings. To guarantee the range-respecting properties of confidence intervals, a transformation procedure such as in CEFA (Browne et al. 2008) is used. For example, because the orthogonal rotated factor loading θ must be bounded between –1 and +1, the Fisher transformation

$$\varphi = \frac{1}{2} \log(\frac{1 + \theta}{1 - \theta})$$

is employed so that φ is an unbounded parameter. Assuming the asymptotic normality of $\hat{\varphi}$, a symmetric confidence interval for φ is constructed. Then, a back-transformation on the confidence limits yields an asymmetric confidence interval for θ. Applying the results of Browne (1982), a $(1-\alpha)100\%$ confidence interval for the orthogonal factor loading θ is

$$(\hat{\theta}_l = \frac{a/b - 1}{a/b + 1}, \quad \hat{\theta}_u = \frac{a \times b - 1}{a \times b + 1})$$

where

$$a = \frac{1 + \hat{\theta}}{1 - \hat{\theta}}, \quad b = \exp(z_{\alpha/2} \times \frac{2\hat{\sigma}}{1 - \hat{\theta}^2})$$

and $\hat{\theta}$ is the estimated factor loading, $\hat{\sigma}$ is the standard error estimate of the factor loading, and $z_{\alpha/2}$ is the $(1 - \alpha/2)100$ percentile point of a standard normal distribution.

Once the confidence limits are constructed, you can use the corresponding coverage displays for determining the salience of the variable-factor relationship. In a coverage display, the COVER= value is represented by an asterisk (*). The following table summarizes various displays and their interpretations.

Table 35.2 Interpretations of the Coverage Displays

Positive Estimate	Negative Estimate	COVER=0 Specified	Interpretation
[0]*	*[0]		The estimate is not significantly different from zero, and the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the nonsalience of the variable-factor relationship.
0[ ]*	*[ ]0		The estimate is significantly different from zero, but the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the nonsalience of the variable-factor relationship.
[0*]	[*0]	[0]	The estimate is not significantly different from zero or the COVER= value. The population value might have been larger or smaller in magnitude than the COVER= value. There is no statistical evidence for the salience of the variable-factor relationship.
0[*]	[*]0		The estimate is significantly different from zero but not from the COVER= value. This is marginal statistical evidence for the salience of the variable-factor relationship.
0*[ ]	[ ]*0	0[ ] or [ ]0	The estimate is significantly different from zero, and the CI covers a region of values that are larger in magnitude than the COVER= value. This is strong statistical evidence for the salience of the variable-factor relationship.

See Example 35.4 for an illustration of the use of confidence intervals for interpreting factors.

Simplicity Functions for Rotations

To rotate a factor pattern is to apply a nonsingular linear transformation to the unrotated factor pattern matrix. An optimal transformation is usually defined as a minimum or maximum point of a simplicity function. Different rotation methods are based on different simplicity functions employed.

For the promax or the Procrustes rotation, the simplicity function used is the sum of squared differences between the rotated factor pattern and the target matrix. The optimal transformation is obtained by minimizing this simplicity function with respect to the choices of all possible transformation.

For the class of the generalized Crawford-Ferguson family Jennrich (1973), the simplicity function being optimized is

$$f = k_1 Z + k_2 H + k_3 V + k_4 Q$$

where

$$Z = (\sum_j \sum_i b_{ij}^2)^2, \quad H = \sum_i (\sum_j b_{ij}^2)^2$$

$$V = \sum_j (\sum_i b_{ij}^2)^2, \quad Q = \sum_j \sum_i b_{ij}^4$$

k_1, k_2, k_3, and k_4 are constants, and b_{ij} represents an element of the rotated pattern matrix. Except for specialized research purposes, it is uncommon in practice to use this simplicity function directly for rotation. However, this simplicity function reduces to many well-known classes of rotations. One of these is the Crawford-Ferguson family Crawford and Ferguson (1970), which minimizes

$$f_{cf} = c_1(H - Q) + c_2(V - Q)$$

where c_1 and c_2 are constants, $(H - Q)$ represents variable (row) parsimony, and $(V - Q)$ represents factor (column) parsimony. Therefore, the relative importance of both the variable parsimony and of the factor parsimony is adjusted using the constants c_1 and c_2. The orthomax class (see Harman 1976) maximizes the function

$$f_{or} = pQ - \gamma V$$

where γ is the orthomax weight and is usually between 0 and the number of variables p. The oblimin class minimizes the function

$$f_{ob} = p(H - Q) - \tau(Z - V)$$

where τ is the oblimin weight. For practical purposes, a negative or zero value for τ is recommended.

All of the preceding definitions are for rotations without row normalization. For rotations with Kaiser normalization, the definition of b_{ij} is replaced by b_{ij}/h_i, where h_i is the communality estimate of variable i.

Missing Values

If the DATA= data set contains data (rather than a matrix or factor pattern), then observations with missing values for any variables in the analysis are omitted from the computations. If a correlation or covariance matrix is read, it can contain missing values as long as every pair of variables has at least one nonmissing entry. Missing values in a pattern or scoring coefficient matrix are treated as zeros.

Cautions

- The amount of time that FACTOR takes is roughly proportional to the cube of the number of variables. Factoring 100 variables, therefore, takes about 1,000 times as long as factoring 10 variables. Iterative methods (PRINIT, ALPHA, ULS, ML) can also take 100 times as long as noniterative methods (PRINCIPAL, IMAGE, HARRIS).

- No computer program is capable of reliably determining the optimal number of factors, since the decision is ultimately subjective. You should not blindly accept the number of factors obtained by default; instead, use your own judgment to make a decision.

- Singular correlation matrices cause problems with the options PRIORS=SMC and METHOD=ML. Singularities can result from using a variable that is the sum of other variables, coding too many dummy variables from a classification variable, or having more variables than observations.

- If you use the CORR procedure to compute the correlation matrix and there are missing data and the NOMISS option is not specified, then the correlation matrix might have negative eigenvalues.

- If a TYPE=CORR, TYPE=UCORR, or TYPE=FACTOR data set is copied or modified using a DATA step, the new data set does not automatically have the same TYPE as the old data set. You must specify the TYPE= data set option in the DATA statement. If you try to analyze a data set that has lost its TYPE=CORR attribute, PROC FACTOR displays a warning message saying that the data set contains _NAME_ and _TYPE_ variables but analyzes the data set as an ordinary SAS data set.

- For a TYPE=FACTOR data set, the default is METHOD=PATTERN, not METHOD=PRIN.

Factor Scores

The FACTOR procedure can compute estimated factor scores directly if you specify the NFACTORS= and OUT= options, or indirectly using the SCORE procedure. The latter method is preferable if you use the FACTOR procedure interactively to determine the number of factors, the rotation method, or various other aspects of the analysis. To compute factor scores for each observation by using the SCORE procedure, do the following:

- Use the SCORE option in the PROC FACTOR statement.

- Create a TYPE=FACTOR output data set with the OUTSTAT= option.

- Use the SCORE procedure with both the raw data and the TYPE=FACTOR data set.

- Do not use the TYPE= option in the PROC SCORE statement.

For example, the following statements could be used:

```
proc factor data=raw score outstat=fact;
run;
proc score data=raw score=fact out=scores;
run;
```

or

```
proc corr   data=raw out=correl;
run;
proc factor data=correl score outstat=fact;
run;
proc score  data=raw score=fact out=scores;
run;
```

For a more detailed example, see Example 82.1 in Chapter 82, "The SCORE Procedure."

A component analysis (principal, image, or Harris) produces scores with mean zero and variance one. If you have done a common factor analysis, the true factor scores have mean zero and variance one, but the computed factor scores are only estimates of the true factor scores. These estimates have mean zero but variance equal to the squared multiple correlation of the factor with the variables. The estimated factor scores might have small nonzero correlations even if the true factors are uncorrelated.

Variable Weights and Variance Explained

A principal component analysis of a correlation matrix treats all variables as equally important. A principal component analysis of a covariance matrix gives more weight to variables with larger variances. A principal component analysis of a covariance matrix is equivalent to an analysis of a weighted correlation matrix, where the weight of each variable is equal to its variance. Variables with large weights tend to have larger loadings on the first component and smaller residual correlations than variables with small weights.

You might want to give weights to variables by using values other than their variances. Mulaik (1972) explains how to obtain a maximally reliable component by means of a weighted principal component analysis. With the FACTOR procedure, you can indirectly give arbitrary weights to the variables by using the COV option and rescaling the variables to have variance equal to the desired weight, or you can give arbitrary weights directly by using the WEIGHT option and including the weights in a TYPE=CORR data set.

Arbitrary variable weights can be used with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. Alpha and ML factor analyses compute variable weights based on the communalities (Harman 1976, pp. 217–218). For alpha factor analysis, the weight of a variable is the reciprocal of its communality. In ML factor analysis, the weight is the reciprocal of the uniqueness. Harris component analysis uses weights equal to the reciprocal of one minus the squared multiple correlation of each variable with the other variables.

For uncorrelated factors, the variance explained by a factor can be computed with or without taking the weights into account. The usual method for computing variance accounted for by a factor is to take the sum of squares of the corresponding column of the factor pattern, yielding an unweighted result. If the square of each loading is multiplied by the weight of the variable before the sum is taken, the result is the weighted variance explained, which is equal to the corresponding eigenvalue except in image analysis. Whether the weighted or unweighted result is more important depends on the purpose of the analysis.

In the case of correlated factors, the variance explained by a factor can be computed with or without taking the other factors into account. If you want to ignore the other factors, the variance explained is given by the weighted or unweighted sum of squares of the appropriate column of the factor structure since the factor structure contains simple correlations. If you want to subtract the variance explained by the other factors from the amount explained by the factor in question (the Type II variance explained), you can take the weighted or unweighted sum of squares of the appropriate column of the reference structure because the reference structure contains semipartial correlations. There are other ways of measuring the variance explained. For example, given a prior ordering of the factors, you can eliminate from each factor the variance explained by previous factors and compute a Type I variance explained. Harman (1976, pp. 268–270) provides another method, which is based on direct and joint contributions.

Heywood Cases and Other Anomalies about Communality Estimates

Since communalities are squared correlations, you would expect them always to lie between 0 and 1. It is a mathematical peculiarity of the common factor model, however, that final communality estimates might exceed 1. If a communality equals 1, the situation is referred to as a Heywood case, and if a communality exceeds 1, it is an ultra-Heywood case. An ultra-Heywood case implies that some unique factor has negative variance, a clear indication that something is wrong. Possible causes include the following:

- bad prior communality estimates

- too many common factors

- too few common factors

- not enough data to provide stable estimates

- the common factor model is not an appropriate model for the data

An ultra-Heywood case renders a factor solution invalid. Factor analysts disagree about whether or not a factor solution with a Heywood case can be considered legitimate.

With METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, or METHOD=ML, the FACTOR procedure, by default, stops iterating and sets the number of factors to 0 if an estimated communality exceeds 1. To enable processing to continue with a Heywood or ultra-Heywood case, you can use the HEYWOOD or ULTRAHEYWOOD option in the PROC FACTOR statement. The HEYWOOD option sets the upper bound of any communality to 1, while the ULTRAHEYWOOD option allows communalities to exceed 1.

Theoretically, the communality of a variable should not exceed its reliability. Violation of this condition is called a quasi-Heywood case and should be regarded with the same suspicion as an ultra-Heywood case.

Elements of the factor structure and reference structure matrices can exceed 1 only in the presence of an ultra-Heywood case. On the other hand, an element of the factor pattern might exceed 1 in an oblique rotation.

The maximum likelihood method is especially susceptible to quasi- or ultra-Heywood cases. During the iteration process, a variable with high communality is given a high weight; this tends to increase its communality, which increases its weight, and so on.

It is often stated that the squared multiple correlation of a variable with the other variables is a lower bound to its communality. This is true if the common factor model fits the data perfectly, but it is not generally the case with real data. A final communality estimate that is less than the squared multiple correlation can, therefore, indicate poor fit, possibly due to not enough factors. It is by no means as serious a problem as an ultra-Heywood case. Factor methods that use the Newton-Raphson method can actually produce communalities less than 0, a result even more disastrous than an ultra-Heywood case.

The squared multiple correlation of a factor with the variables might exceed 1, even in the absence of ultra-Heywood cases. This situation is also cause for alarm. Alpha factor analysis seems to be especially prone to this problem, but it does not occur with maximum likelihood. If a squared multiple correlation is negative, there are too many factors retained.

With data that do not fit the common factor model perfectly, you can expect some of the eigenvalues to be negative. If an iterative factor method converges properly, the sum of the eigenvalues corresponding to rejected factors should be 0; hence, some eigenvalues are positive and some negative. If a principal factor analysis fails to yield any negative eigenvalues, the prior communality estimates are probably too large. Negative eigenvalues cause the cumulative proportion of variance explained to exceed 1 for a sufficiently large number of factors. The cumulative proportion of variance explained by the retained factors should be approximately 1 for principal factor analysis and should converge to 1 for iterative methods. Occasionally, a single factor can explain more than 100 percent of the common variance in a principal factor analysis, indicating that the prior communality estimates are too low.

If a squared canonical correlation or a coefficient alpha is negative, there are too many factors retained.

Principal component analysis, unlike common factor analysis, has none of these problems if the covariance or correlation matrix is computed correctly from a data set with no missing values. Various methods for missing value correlation or severe rounding of the correlations can produce negative eigenvalues in principal components.

Time Requirements

n = number of observations
v = number of variables
f = number of factors
i = number of iterations during factor extraction
r = length of iterations during factor rotation

The time required to compute...	is roughly proportional to
an overall factor analysis	iv^3
the correlation matrix	nv^2
PRIORS=SMC or ASMC	v^3
PRIORS=MAX	v^2
eigenvalues	v^3
final eigenvectors	fv^2
generalized Crawford-Ferguson family of rotations, PROMAX, or HK	rvf^2
ROTATE=PROCRUSTES	vf^2

Each iteration in the PRINIT or ALPHA method requires computation of eigenvalues and f eigenvectors.

Each iteration in the ML or ULS method requires computation of eigenvalues and $v - f$ eigenvectors.

The amount of time that PROC FACTOR takes is roughly proportional to the cube of the number of variables. Factoring 100 variables, therefore, takes about 1000 times as long as factoring 10 variables. Iterative methods (PRINIT, ALPHA, ULS, ML) can also take 100 times as long as noniterative methods (PRINCIPAL, IMAGE, HARRIS).

Displayed Output

PROC FACTOR output includes the following.

- Input data type, numbers of records read and used for raw data input, number of observations (NOBS=) set in the PROC FACTOR statements, and the number of observations used in significance tests

- Mean and Std Dev (standard deviation) of each variable and the number of observations, if you specify the SIMPLE option

- Correlations, if you specify the CORR option

- Inverse Correlation Matrix, if you specify the ALL option

- Partial Correlations Controlling all other Variables (negative anti-image correlations), if you specify the MSA option. If the data are appropriate for the common factor model, the partial correlations should be small.

- Kaiser's Measure of Sampling Adequacy (Kaiser 1970; Kaiser and Rice 1974; Cerny and Kaiser 1977), both overall and for each variable, if you specify the MSA option. The MSA is a summary of how small the partial correlations are relative to the ordinary correlations. Values greater than 0.8 can be considered good. Values less than 0.5 require remedial action, either by deleting the offending variables or by including other variables related to the offenders.

- Prior Communality Estimates, unless 1.0s are used or unless you specify the METHOD=IMAGE, METHOD=HARRIS, METHOD=PATTERN, or METHOD=SCORE option

- Squared Multiple Correlations of each variable with all the other variables, if you specify the METHOD=IMAGE or METHOD=HARRIS option

- Image Coefficients, if you specify the METHOD=IMAGE option

- Image Covariance Matrix, if you specify the METHOD=IMAGE option

- Preliminary Eigenvalues based on the prior communalities, if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option. The table produced includes the Total and the Average of the eigenvalues, the Difference between successive eigenvalues, the Proportion of variation represented, and the Cumulative proportion of variation.

- the number of factors that are retained, unless you specify the METHOD=PATTERN or METHOD=SCORE option

- the Scree Plot of Eigenvalues, if you specify the SCREE option. The preliminary eigenvalues are used if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option.

- the iteration history, if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option. The table produced contains the iteration number (Iter); the Criterion being optimized (Jöreskog 1977); the Ridge value for the iteration if you specify the METHOD=ML or METHOD=ULS option; the maximum Change in any communality estimate; and the Communalities.

- Significance tests, if you specify the option METHOD=ML, including Bartlett's chi-square, df, and Prob $> \chi^2$ for H_0: No common factors and H_0: factors retained are sufficient to explain the correlations. The H_0 test for no common factors is equivalent to Bartlett's test of sphericity. The variables should have an approximate multivariate normal distribution for the probability levels to be valid. Lawley and Maxwell (1971) suggest that the number of observations should exceed the number of variables by 50 or more, although Geweke and Singleton (1980) claim that as few as 10 observations are adequate with five variables and one common factor. Certain regularity conditions must also be satisfied for Bartlett's χ^2 test to be valid (Geweke and Singleton 1980), but in practice these conditions usually are satisfied. The notation Prob>chi**2 means "the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed." The chi-square value is displayed with and without Bartlett's correction.

- Akaike's Information Criterion, if you specify the METHOD=ML option. Akaike's information criterion (AIC) (Akaike 1973, 1974, 1987) is a general criterion for estimating the best number of parameters to include in a model when maximum likelihood estimation is used. The number of factors that yields the smallest value of AIC is considered best. Like the chi-square test, AIC tends to include factors that are statistically significant but inconsequential for practical purposes.

- Schwarz's Bayesian Criterion, if you specify the METHOD=ML option. Schwarz's Bayesian Criterion (SBC) (Schwarz 1978) is another criterion, similar to AIC, for determining the best number of parameters. The number of factors that yields the smallest value of SBC is considered best; SBC seems to be less inclined to include trivial factors than either AIC or the chi-square test.

- Tucker and Lewis's Reliability Coefficient, if you specify the METHOD=ML option (Tucker and Lewis 1973)

- Squared Canonical Correlations, if you specify the METHOD=ML option. These are the same as the squared multiple correlations for predicting each factor from the variables.

- Coefficient Alpha for Each Factor, if you specify the METHOD=ALPHA option

- Eigenvectors, if you specify the EIGENVECTORS or ALL option, unless you also specify the METHOD=PATTERN or METHOD=SCORE option

- Eigenvalues of the (Weighted) (Reduced) (Image) Correlation or Covariance Matrix, unless you specify the METHOD=PATTERN or METHOD=SCORE option. Included are the Total and the Average of the eigenvalues, the Difference between successive eigenvalues, the Proportion of variation represented, and the Cumulative proportion of variation.

- the Factor Pattern, which is equal to both the matrix of standardized regression coefficients for predicting variables from common factors and the matrix of correlations between variables and common factors since the extracted factors are uncorrelated. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.

- Variance explained by each factor, both Weighted and Unweighted, if variable weights are used

- Final Communality Estimates, including the Total communality; or Final Communality Estimates and Variable Weights, including the Total communality, both Weighted and Unweighted, if variable weights are used. Final communality estimates are the squared multiple correlations for predicting the variables from the estimated factors, and they can be obtained by taking the sum of squares of each row of the factor pattern, or a weighted sum of squares if variable weights are used.

- Residual Correlations with Uniqueness on the Diagonal, if you specify the RESIDUAL or ALL option

- Root Mean Square Off-diagonal Residuals, both Over-all and for each variable, if you specify the RESIDUAL or ALL option

- Partial Correlations Controlling Factors, if you specify the RESIDUAL or ALL option

- Root Mean Square Off-diagonal Partials, both Over-all and for each variable, if you specify the RESIDUAL or ALL option

- Plots of Factor Pattern for unrotated factors, if you specify the PREPLOT option. The number of plots is determined by the NPLOT= option.

- Variable Weights for Rotation, if you specify the NORM=WEIGHT option

- Factor Weights for Rotation, if you specify the HKPOWER= option

- Orthogonal Transformation Matrix, if you request an orthogonal rotation

- Rotated Factor Pattern, if you request an orthogonal rotation. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.

- Variance explained by each factor after rotation. If you request an orthogonal rotation and if variable weights are used, both weighted and unweighted values are produced.

- Target Matrix for Procrustean Transformation, if you specify the ROTATE=PROMAX or ROTATE=PROCRUSTES option

- the Procrustean Transformation Matrix, if you specify the ROTATE=PROMAX or ROTATE=PROCRUSTES option

- the Normalized Oblique Transformation Matrix, if you request an oblique rotation, which, for the option ROTATE=PROMAX, is the product of the prerotation and the Procrustes rotation

- Inter-factor Correlations, if you specify an oblique rotation. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.

- Rotated Factor Pattern (Std Reg Coefs), if you specify an oblique rotation, giving standardized regression coefficients for predicting the variables from the factors. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.

- Reference Axis Correlations if you specify an oblique rotation. These are the partial correlations between the primary factors when all factors other than the two being correlated are partialed out.

- Reference Structure (Semipartial Correlations), if you request an oblique rotation. The reference structure is the matrix of semipartial correlations (Kerlinger and Pedhazur 1973) between variables and common factors, removing from each common factor the effects of other common factors. If the common factors are uncorrelated, the reference structure is equal to the factor pattern.

- Variance explained by each factor eliminating the effects of all other factors, if you specify an oblique rotation. Both Weighted and Unweighted values are produced if variable weights are used. These variances are equal to the (weighted) sum of the squared elements of the reference structure corresponding to each factor.

- Factor Structure (Correlations), if you request an oblique rotation. The (primary) factor structure is the matrix of correlations between variables and common factors. If the common factors are uncorrelated, the factor structure is equal to the factor pattern. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.

- Variance explained by each factor ignoring the effects of all other factors, if you request an oblique rotation. Both Weighted and Unweighted values are produced if variable weights are used. These variances are equal to the (weighted) sum of the squared elements of the factor structure corresponding to each factor.

- Final Communality Estimates for the rotated factors if you specify the ROTATE= option. The estimates should equal the unrotated communalities.

- Squared Multiple Correlations of the Variables with Each Factor, if you specify the SCORE or ALL option, except for unrotated principal components

- Standardized Scoring Coefficients, if you specify the SCORE or ALL option

- Plots of the Factor Pattern for rotated factors, if you specify the PLOT option and you request an orthogonal rotation. The number of plots is determined by the NPLOT= option.

- Plots of the Reference Structure for rotated factors, if you specify the PLOT option and you request an oblique rotation. The number of plots is determined by the NPLOT= option. Included are the Reference Axis Correlation and the Angle between the Reference Axes for each pair of factors plotted.

If you specify the ROTATE=PROMAX option, the output includes results for both the prerotation and the Procrustes rotation.

ODS Table Names

PROC FACTOR 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 the Table 35.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 35.3 ODS Tables Produced by PROC FACTOR

ODS Table Name	Description	Option
AlphaCoef	Coefficient alpha for each factor	METHOD=ALPHA
CanCorr	Squared canonical correlations	METHOD=ML
CondStdDev	Conditional standard deviations	SIMPLE with PARTIAL
ConvergenceStatus	Convergence status	METHOD=PRINIT, ALPHA, ML, or ULS
Corr	Correlations	CORR
Eigenvalues	Eigenvalues	default, SCREE
Eigenvectors	Eigenvectors	EIGENVECTORS
FactorWeightRotate	Factor weights for rotation	HKPOWER=
FactorPattern	Factor pattern	default
FactorStructure	Factor structure	ROTATE= any oblique rotation
FinalCommun	Final communalities	default
FinalCommunWgt	Final communalities with weights	METHOD=ML or ALPHA
FitMeasures	Measures of fit	METHOD=ML
ImageCoef	Image coefficients	METHOD=IMAGE
ImageCov	Image covariance matrix	METHOD=IMAGE
ImageFactors	Image factor matrix	METHOD=IMAGE
InputFactorPattern	Input factor pattern	METHOD=PATTERN with PRINT or ALL
InputScoreCoef	Standardized input scoring coefficients	METHOD=SCORE with PRINT or ALL
InterFactorCorr	Interfactor correlations	ROTATE= any oblique rotation
InvCorr	Inverse correlation matrix	ALL
IterHistory	Iteration history	METHOD=PRINIT, ALPHA, ML, or ULS
MultipleCorr	Squared multiple correlations	METHOD=IMAGE or METHOD=HARRIS
NObs	Number of records and observations, input data type	default

Table 35.3 *continued*

ODS Table Name	Description	Option
NormObliqueTrans	Normalized oblique transformation matrix	ROTATE= any oblique rotation
ObliqueRotFactPat	Rotated factor pattern	ROTATE= any oblique rotation
ObliqueTrans	Oblique transformation matrix	HKPOWER=
OrthRotFactPat	Rotated factor pattern	ROTATE= any orthogonal rotation
OrthTrans	Orthogonal transformation matrix	ROTATE= any orthogonal rotation
ParCorrControlFactor	Partial correlations controlling factors	RESIDUAL
ParCorrControlVar	Partial correlations controlling other variables	MSA
PartialCorr	Partial correlations	MSA, CORR with PARTIAL
PriorCommunalEst	Prior communality estimates	PRIORS=, METHOD=ML or ALPHA
ProcrustesTarget	Target matrix for Procrustean transformation	ROTATE=PROCRUSTES, ROTATE=PROMAX
ProcrustesTrans	Procrustean transformation matrix	ROTATE=PROCRUSTES, ROTATE=PROMAX
RMSOffDiagPartials	Root mean square off-diagonal partials	RESIDUAL
RMSOffDiagResids	Root mean square off-diagonal residuals	RESIDUAL
ReferenceAxisCorr	Reference axis correlations	ROTATE= any oblique rotation
ReferenceStructure	Reference structure	ROTATE= any oblique rotation
ResCorrUniqueDiag	Residual correlations with uniqueness on the diagonal	RESIDUAL
SamplingAdequacy	Kaiser's measure of sampling adequacy	MSA
SignifTests	Significance tests	METHOD=ML
SimpleStatistics	Simple statistics	SIMPLE
StdScoreCoef	Standardized scoring coefficients	SCORE
VarExplain	Variance explained	default
VarExplainWgt	Variance explained with weights	METHOD=ML, or ALPHA
VarFactorCorr	Squared multiple correlations of the variables with each factor	SCORE
VarWeightRotate	Variable weights for rotation	NORM=WEIGHT, ROTATE=

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 600 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 599 in Chapter 21, "Statistical Graphics Using ODS."

The names of the graphs that PROC FACTOR generates are listed in Table 35.4, along with the required statements and options.

Table 35.4   Graphs Produced by PROC FACTOR

ODS Graph Name	Plot Description	Option
PatternPlot	Rotated factor pattern	PLOTS=LOADINGS
InitPatternPlot	Initial factor pattern	PLOTS=INITLOADINGS
InitRefStructurePlot	Initial reference structures	PLOTS=INITLOADINGS and PLOTREF
PrePatternPlot	Prerotated factor pattern	PLOTS=PRELOADINGS
PreRefStructurePlot	Prerotated reference structures	PLOTS=PRELOADINGS and PLOTREF
RefStructurePlot	Rotated reference structures	PLOTS=LOADINGS and PLOTREF
ScreePlot	Scree and variance explained plots	PLOTS=SCREE
VariancePlot	Plot of explained variance	PLOTS=SCREE(UNPACK)

Examples: FACTOR Procedure

Example 35.1: Principal Component Analysis

This example analyzes socioeconomic data provided by Harman (1976). The five variables represent total population (Population), median school years (School), total employment (Employment), miscellaneous professional services (Services), and median house value (HouseValue). Each observation represents one of twelve census tracts in the Los Angeles Standard Metropolitan Statistical Area.

You conduct a principal component analysis by using the following statements:

```
data SocioEconomics;
   input Population School Employment Services HouseValue;
   datalines;
5700     12.8      2500      270       25000
1000     10.9      600       10        10000
3400     8.8       1000      10        9000
3800     13.6      1700      140       25000
4000     12.8      1600      140       25000
8200     8.3       2600      60        12000
1200     11.4      400       10        16000
9100     11.5      3300      60        14000
9900     12.5      3400      180       18000
9600     13.7      3600      390       25000
9600     9.6       3300      80        12000
9400     11.4      4000      100       13000
;

proc factor data=SocioEconomics simple corr;
run;
```

You begin with the specification of the raw data set with 12 observations. Then you use the DATA= option in the PROC FACTOR statement to specify the data set in the analysis. You also set the SIMPLE and CORR options for additional output results, which are shown in Output 35.1.2 and Output 35.1.3, respectively.

By default, PROC FACTOR assumes that all initial communalities are 1, which is the case for the current principal component analysis. If you intend to find common factors instead, use the PRIORS= option or the PRIORS statement to set initial communalities to values less than 1, which results in extracting the principal factors rather than the principal components. See Example 35.2 for the specification of a principal factor analysis.

For the current principal component analysis, the first output table is displayed in the Output 35.1.1.

Output 35.1.1 Principal Component Analysis: Number of Observations

```
                   Five Socioeconomic Variables
          See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
                   Principal Component Analysis

                       The FACTOR Procedure

         Input Data Type                    Raw Data
         Number of Records Read                   12
         Number of Records Used                   12
         N for Significance Tests                 12
```

In Output 35.1.1, the input data type is shown to be raw data. PROC FACTOR also accepts other data type such as correlations and covariances. See Example 35.4 for the use of correlations as input data. For the current raw data set, PROC FACTOR reads in 12 records and all these 12 records are used. When there are missing values in the data set, these two numbers might not match due to the dropping of the records with missing values. The last row of the table shows that $N = 12$ is used in the significance tests conducted in the analysis.

The SIMPLE option specified in the PROC FACTOR statement generates the means and standard deviations of all observed variables in the analysis, as shown in Output 35.1.2.

Output 35.1.2 Principal Component Analysis: Simple Statistics

```
          Means and Standard Deviations from 12 Observations

               Variable              Mean       Std Dev

               Population        6241.667     3439.9943
               School              11.442        1.7865
               Employment        2333.333     1241.2115
               Services           120.833      114.9275
               HouseValue       17000.000     6367.5313
```

The ranges of means and standard deviations for the analysis are quite large. Variables are measured on quite different scales. However, this is not an issue because PROC FACTOR basically analyzes the standardized scales (that is, the correlations) of the variables.

The CORR option specified in the PROC FACTOR statement generates the output of the observed correlations in Output 35.1.3.

Output 35.1.3 Principal Component Analysis: Correlations

```
                                  Correlations

                 Population      School    Employment     Services    HouseValue

Population          1.00000     0.00975       0.97245      0.43887       0.02241
School              0.00975     1.00000       0.15428      0.69141       0.86307
Employment          0.97245     0.15428       1.00000      0.51472       0.12193
Services            0.43887     0.69141       0.51472      1.00000       0.77765
HouseValue          0.02241     0.86307       0.12193      0.77765       1.00000
```

The correlation matrix shown in Output 35.1.3 is analyzed by PROC FACTOR.

The first step of principal component analysis is to look at the eigenvalues of the correlation matrix. The larger eigenvalues are extracted first. Because there are five observed variables, five eigenvalues can be extracted, as shown in Output 35.1.4.

Output 35.1.4 Principal Component Analysis: Eigenvalues

```
     Eigenvalues of the Correlation Matrix: Total = 5  Average = 1

          Eigenvalue    Difference    Proportion    Cumulative

      1    2.87331359    1.07665350      0.5747        0.5747
      2    1.79666009    1.58182321      0.3593        0.9340
      3    0.21483689    0.11490283      0.0430        0.9770
      4    0.09993405    0.08467868      0.0200        0.9969
      5    0.01525537                    0.0031        1.0000
```

In Output 35.1.4, the two largest eigenvalues are 2.8733 and 1.7967, which together account for 93.4% of the standardized variance. Thus, the first two principal components provide an adequate summary of the data for most purposes. Three components, which explain 97.7% of the variation, should be sufficient for almost any application. PROC FACTOR retains the first two components on the basis of the eigenvalues-greater-than-one rule since the third eigenvalue is only 0.2148.

To express the observed variables as functions of the components (or factors, in general), you consult the factor loading matrix as shown in Output 35.1.5.

Output 35.1.5 Principal Component Analysis: Factor Pattern

```
                        Factor Pattern

                      Factor1            Factor2

         Population    0.58096            0.80642
         School        0.76704           -0.54476
         Employment    0.67243            0.72605
         Services      0.93239           -0.10431
         HouseValue    0.79116           -0.55818
```

The factor pattern is often referred to as the factor loading matrix in factor analysis. The elements in the loading matrix are called factor loadings. There are at least two ways you can interpret these factor loadings. First, you can use this table to express the observed variables as functions of the extracted factors (or components, as in the current analysis). Each row of the factor loadings tells you the linear combination of the factor or component scores that would yield the expected value of the associated variable. Second, you can interpret each loading as a correlation between an observed variable and a factor or component, provided that the factor solution is an orthogonal one (that is, factors are uncorrelated), such as the current initial factor solution. Hence, the factor loadings indicate how strongly the variables and the factors or components are related.

In Output 35.1.5, the first component (labeled "Factor1") has large positive loadings for all five variables. Its correlation with Services (0.9324) is especially high. The second component is basically a contrast of Population (0.8064) and Employment (0.7261) against School (–0.5448) and HouseValue (–0.5582), with a very small loading on Services (–0.1043).

The total variance explained by the two components are shown in Output 35.1.6.

Output 35.1.6 Principal Component Analysis: Total Variance Explained by Factors

```
              Variance Explained by Each Factor

                   Factor1              Factor2

                  2.8733136           1.7966601
```

The first and second component account for 2.8733 and 1.7967, respectively, of the total variance of 5. In the initial factor solution, the total variance explained by the factors or components are the same as the eigenvalues extracted. (Compare the total variance with the eigenvalues shown in Output 35.1.4.) Due to the dropping of the less important components, the sum of these two numbers is 4.6700, which is only a little bit less than total variance 5 of the original correlation matrix.

You can also look at the variance explained by the two components for each observed variables in Output 35.1.7.

Output 35.1.7 Principal Component Analysis: Final Communality Estimates

```
              Final Communality Estimates: Total = 4.669974

     Population        School      Employment        Services       HouseValue

     0.98782629    0.88510555      0.97930583      0.88023562       0.93750041
```

In Output 35.1.7, the final communality estimates show that all the variables are well accounted for by the two components, with final communality estimates ranging from 0.8802 for Services to 0.9878 for Population. The sum of the communalities is 4.6700, which is the same as the sum of the variance explained by the two components, as shown in Output 35.1.6.

Principal Component Analysis by PROC FACTOR and PROC PRINCOMP

The principal component analysis by PROC FACTOR emphasizes how the principal components explain the observed variables. The factor loadings in the factor pattern as shown in Output 35.1.5 are the coefficients for combining the factor/component scores to yield the observed variable scores when the expected error residuals are zero. For example, the predicted standardized value of Population given the factor/component scores for Factor1 and Factor2 is given by:

$$\text{Population} = 0.58096 \times \text{Factor1} + 0.80642 \times \text{Factor2}$$

If you are primarily interested in getting the component scores as linear combinations of the observed variables, the factor loading matrix table is not the right one for you. However, you might request the standardized scoring coefficients by adding the SCORE option in the FACTOR statement:

```
proc factor data=SocioEconomics n=5 score;
run;
```

In the preceding PROC FACTOR statement, N=5 is specified for retaining all five components. This is done for comparing the PROC FACTOR results with those of PROC PRINCOMP, which is described later.

The SCORE option requests the display of the standardized scoring coefficients, which are shown in Output 35.1.8.

Output 35.1.8 Principal Component Analysis: Scoring Coefficients for Computing Component Scores

```
                    Standardized Scoring Coefficients

               Factor1       Factor2      Factor3      Factor4      Factor5

Population    0.20219065    0.44884459    0.1284067    0.64542101    5.58240225
School        0.26695219   -0.3032049    1.48611655   -1.1184573    1.41573501
Employment    0.23402646    0.40410834    0.53496241   0.07255759   -5.6513542
Services      0.32450082   -0.0580552    -1.432726    -1.5828806   -0.0010006
HouseValue    0.27534803   -0.3106762    -0.3012889    2.41418899   -0.6673445
```

In Output 35.1.8, each factor/component is expressed as a linear combination of the standardized observed variables. For example, the first principal component or Factor1 is computed as:

$$0.2022 \times \text{Population} + 0.2670 \times \text{School} + 0.2340 \times \text{Employment} + 0.3245 \times \text{Services} + 0.2753 \times \text{HouseValue}$$

Again, when applying this formula you must use the standardized observed variables (with means 0 and standard deviations 1), but not the raw data.

Apart from some scaling differences, the set of scoring coefficients obtained from PROC FACTOR are equivalent to those obtained from PROC PRINCOMP, as specified by the following statement:

```
proc princomp data=SocioEconomics;
run;
```

PROC PRINCOMP displays the scoring coefficients as eigenvectors, which are shown in Output 35.1.9.

Output 35.1.9 Principal Component Analysis by PROC PRINCOMP: Eigenvectors

```
                              Eigenvectors

               Prin1         Prin2        Prin3        Prin4        Prin5

Population    0.342730      0.601629     0.059517     0.204033     0.689497
School        0.452507     -.406414      0.688822    -.353571      0.174861
Employment    0.396695      0.541665     0.247958     0.022937    -.698014
Services      0.550057     -.077817     -.664076     -.500386     -.000124
HouseValue    0.466738     -.416429     -.139649      0.763182     -.082425
```

For example, to get the first principal component score, you use the following formula:

$$0.3427 \times \text{Population} + 0.4525 \times \text{School} + 0.3967 \times \text{Employment} + 0.5500 \times \text{Services} + 0.4667 \times \text{HouseValue}$$

This formula is not exactly the same as the one shown by using PROC FACTOR. All scoring coefficients in PROC FACTOR are smaller, approximately a factor of 0.59 to those coefficients obtained from PROC PRINCOMP. The reason for the scalar difference is that PROC FACTOR assumes all factors/components to

have variance of 1, while PROC PRINCOMP creates components that have variances equal to the eigenvalues. You can do a simple rescaling of the standardized scoring coefficients obtained from PROC FACTOR so that they match the associated eigenvectors from the PROC PRINCOMP. Basically, you need to rescale each column of the standardized scoring coefficients obtained from PROC FACTOR to have the sum of squares equaling one, which is a defining characteristic of eigenvectors. This could be accomplished by dividing each coefficient by the square root of the corresponding column sum of squares.

For the present example, you can use PROC STDIZE to do the rescaling, as shown in the following statements:

```
proc factor data=SocioEconomics n=5 score;
   ods output StdScoreCoef=Coef;
run;

proc stdize method=ustd mult=.44721 data=Coef out=eigenvectors;
   Var Factor1-Factor5;
run;

proc print data=eigenvectors;
run;
```

First, you create an output set Coef for the standardized scoring coefficients by the ODS OUTPUT statement. Note that "StdScoreCoef" is the ODS table that contains the standardized scoring coefficients as shown in Output 35.1.8. (See Table 35.3 for all ODS table names for PROC FACTOR.) Next, you use METHOD=USTD in the PROC STDIZE statement to divide the output coefficients by the corresponding uncorrected (for mean) standard deviations. The following formula shows the relationship between the uncorrected standard deviation and the sum of squares:

$$\text{uncorrected standard deviation} = \sqrt{\text{sum of squares}/N}$$

Recall that what you intend to divide from each coefficient is its square root of the corresponding column sum of squares. Therefore, to adjust for what PROC STDIZE does using METHOD=USTD, you have to multiply each variable by a constant term of $1/\sqrt{N}$ in the standardization. For the current example, this constant term is 0.44721 ($= 1/\sqrt{5}$) and is specified through the MULT= option in the PROC STDIZE statement. With the OUT= option, the rescaled scoring coefficients are saved in the SAS data set eigenvectors. The printout of the data set in Output 35.1.10 shows the rescaled standardized scoring coefficients obtained from PROC FACTOR.

Output 35.1.10 Rescaled Standardized Scoring Coefficients

Obs	Variable	Factor1	Factor2	Factor3	Factor4	Factor5
1	Population	0.34272761	0.60162443	0.05951667	0.20403109	0.68949172
2	School	0.45250304	-0.4064112	0.68881691	-0.3535678	0.17485977
3	Employment	0.39669158	0.54166065	0.24795576	0.02293697	-0.6980081
4	Services	0.5500521	-0.0778162	-0.6640703	-0.5003817	-0.0001236
5	HouseValue	0.4667346	-0.4164256	-0.1396478	0.76317568	-0.0824248

As you can see, these standardized scoring coefficients are essentially the same as those obtained from PROC PRINCOMP, as shown in Output 35.1.9. This example shows that principal component analyses by PROC FACTOR and PROC PRINCOMP are indeed equivalent. PROC PRINCOMP emphasizes more the linear combinations of the variables to form the components, while PROC FACTOR expresses variables as linear combinations of the components in the output. If a principal component analysis of the data is all you need in a particular application, there is no reason to use PROC FACTOR instead of PROC PRINCOMP. Therefore, the following examples focus on common factor analysis for which that you can apply only PROC FACTOR, but not PROC PRINCOMP.

Example 35.2: Principal Factor Analysis

This example uses the data presented in Example 35.1 and performs a principal factor analysis with squared multiple correlations for the prior communality estimates. Unlike Example 35.1, which analyzes the principal components (with default PRIORS=ONE), the current analysis is based on a common factor model. To use a common factor model, you specify PRIORS=SMC in the PROC FACTOR statement, as shown in the following:

```
ods graphics on;

proc factor data=SocioEconomics
   priors=smc msa residual
   rotate=promax reorder
   outstat=fact_all
   plots=(scree initloadings preloadings loadings);
run;

ods graphics off;
```

In the PROC FACTOR statement, you include several other options to help you analyze the results. To help determine whether the common factor model is appropriate, you request the Kaiser's measure of sampling adequacy with the MSA option. You specify the RESIDUALS option to compute the residual correlations and partial correlations.

The ROTATE= and REORDER options are specified to enhance factor interpretability. The ROTATE=PROMAX option produces an orthogonal varimax prerotation (default) followed by an oblique Procrustes rotation, and the REORDER option reorders the variables according to their largest factor loadings. An OUTSTAT= data set is created by PROC FACTOR and displayed in Output 35.2.15.

PROC FACTOR can produce high-quality graphs that are very useful for interpreting the factor solutions. To request these graphs, ODS Graphics must be enabled. All ODS graphs in PROC FACTOR are requested with the PLOTS= option. In this example, you request a scree plot (SCREE) and loading plots for the factor matrix during the following three stages: initial unrotated solution (INITLOADINGS), prerotated (varimax) solution (PRELOADINGS), and promax-rotated solution (LOADINGS). The scree plot helps you determine the number of factors, and the loading plots help you visualize the patterns of factor loadings during various stages of analyses.

Principal Factor Analysis: Kaiser's MSA and Factor Extraction Results

Output 35.2.1 displays the results of the partial correlations and Kaiser's measure of sampling adequacy.

Output 35.2.1 Principal Factor Analysis: Partial Correlations and Kaiser's MSA

```
              Partial Correlations Controlling all other Variables

                 Population      School    Employment      Services     HouseValue

Population        1.00000      -0.54465       0.97083       0.09612        0.15871
School           -0.54465       1.00000       0.54373       0.04996        0.64717
Employment        0.97083       0.54373       1.00000       0.06689       -0.25572
Services          0.09612       0.04996       0.06689       1.00000        0.59415
HouseValue        0.15871       0.64717      -0.25572       0.59415        1.00000


        Kaiser's Measure of Sampling Adequacy: Overall MSA = 0.57536759

      Population          School       Employment        Services       HouseValue

      0.47207897        0.55158839     0.48851137      0.80664365       0.61281377
```

If the data are appropriate for the common factor model, the partial correlations (controlling all other variables) should be small compared to the original correlations. For example, the partial correlation between the variables School and HouseValue is 0.65, slightly less than the original correlation of 0.86 (see Output 35.1.3). The partial correlation between Population and School is –0.54, which is much larger in absolute value than the original correlation; this is an indication of trouble. Kaiser's MSA is a summary, for each variable and for all variables together, of how much smaller the partial correlations are than the original correlations. Values of 0.8 or 0.9 are considered good, while MSAs below 0.5 are unacceptable. The variables Population, School, and Employment have very poor MSAs. Only the Services variable has a good MSA. The overall MSA of 0.58 is sufficiently poor that additional variables should be included in the analysis to better define the common factors. A commonly used rule is that there should be at least three variables per factor. In the following analysis, you determine that there are two common factors in these data. Therefore, more variables are needed for a reliable analysis.

Output 35.2.2 displays the results of the principal factor extraction.

Output 35.2.2 Principal Factor Analysis: Factor Extraction

```
                    Prior Communality Estimates: SMC

      Population          School       Employment        Services       HouseValue

      0.96859160        0.82228514     0.96918082      0.78572440       0.84701921
```

Output 35.2.2 *continued*

```
           Eigenvalues of the Reduced Correlation Matrix:
              Total = 4.39280116   Average = 0.87856023

             Eigenvalue    Difference    Proportion    Cumulative

        1    2.73430084    1.01823217       0.6225        0.6225
        2    1.71606867    1.67650586       0.3907        1.0131
        3    0.03956281    0.06408626       0.0090        1.0221
        4   -.02452345    0.04808427      -0.0056        1.0165
        5   -.07260772                     -0.0165        1.0000
```

The square multiple correlations are shown as prior communality estimates in Output 35.2.2. The PRI-ORS=SMC option basically replaces the diagonal of the original observed correlation matrix by these square multiple correlations. Because the square multiple correlations are usually less than one, the resulting correlation matrix for factoring is called the reduced correlation matrix. In the current example, the SMCs are all fairly large; hence, you expect the results of the principal factor analysis to be similar to those in the principal component analysis.

The first two largest positive eigenvalues of the reduced correlation matrix account for 101.31% of the common variance. This is possible because the reduced correlation matrix, in general, is not necessarily positive definite, and negative eigenvalues for the matrix are possible. A pattern like this suggests that you might not need more than two common factors. The scree and variance explained plots of Output 35.2.3 clearly support the conclusion that two common factors are present. Showing in the left panel of Output 35.2.3 is the scree plot of the eigenvalues of the reduced correlation matrix. A sharp bend occurs at the third eigenvalue, reinforcing the conclusion that two common factors are present. These cumulative proportions of common variance explained by factors are plotted in the right panel of Output 35.2.3, which shows that the curve essentially flattens out after the second factor.

Output 35.2.3 Scree and Variance Explained Plots
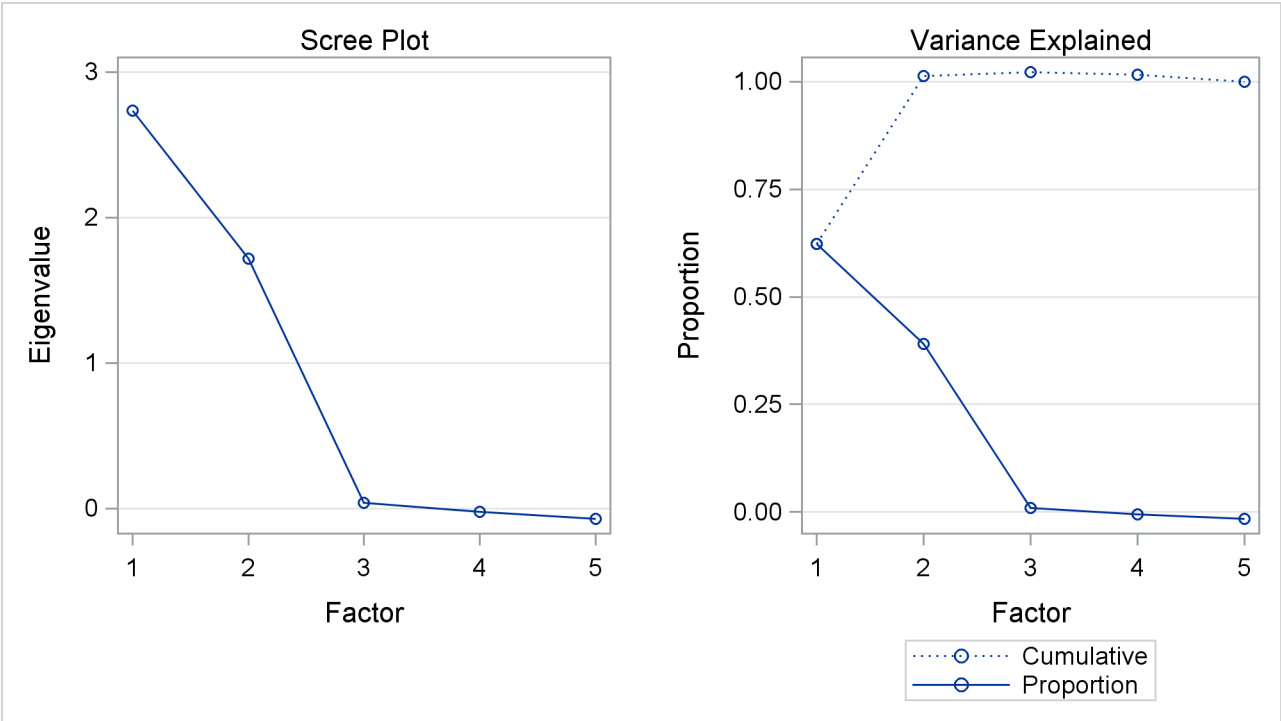
Principal Factor Analysis: Initial Factor Solution

For the current analysis, PROC FACTOR retains two factors by certain default criteria. This decision agrees with the conclusion drawn by inspecting the scree plot. The principal factor pattern with the two factors is displayed in Output 35.2.4. This factor pattern is similar to the principal component pattern seen in Output 35.1.5 of Example 35.1. For example, the variable Services has the largest loading on the first factor, and the Population variable has the smallest. The variables Population and Employment have large positive loadings on the second factor, and the HouseValue and School variables have large negative loadings.

Output 35.2.4 Initial Factor Pattern Matrix and Communalities

Factor Pattern

	Factor1	Factor2
Services	0.87899	−0.15847
HouseValue	0.74215	−0.57806
Employment	0.71447	0.67936
School	0.71370	−0.55515
Population	0.62533	0.76621

Variance Explained by Each Factor

Factor1	Factor2
2.7343008	1.7160687

```
            Final Communality Estimates: Total = 4.450370

   Population          School       Employment         Services        HouseValue

  0.97811334       0.81756387       0.97199928       0.79774304       0.88494998
```

Comparing the current factor loading matrix in Output 35.2.4 with that in Output 35.1.5 in Example 35.1, you notice that the variables are arranged differently in the two output tables. This is due to the use of the REORDER option in the current analysis. The advantage of using this option might not be very obvious in Output 35.2.4, but you can see its value when looking at the rotated solutions, as shown in Output 35.2.7 and Output 35.2.11.

The final communality estimates are all fairly close to the priors (shown in Output 35.2.2). Only the communality for the variable HouseValue increased appreciably, from 0.847 to 0.885. Therefore, you are sure that all the common variance is accounted for.

Output 35.2.5 shows that the residual correlations (off-diagonal elements) are low, the largest being 0.03. The partial correlations are not quite as impressive, since the uniqueness values are also rather small. These results indicate that the squared multiple correlations are good but not quite optimal communality estimates.

Output 35.2.5 Residual and Partial Correlations

```
           Residual Correlations With Uniqueness on the Diagonal

                 Population         School    Employment         Services       HouseValue

Population          0.02189       -0.01118       0.00514          0.01063          0.00124
School             -0.01118        0.18244       0.02151         -0.02390          0.01248
Employment          0.00514        0.02151       0.02800         -0.00565         -0.01561
Services            0.01063       -0.02390      -0.00565          0.20226          0.03370
HouseValue          0.00124        0.01248      -0.01561          0.03370          0.11505


          Root Mean Square Off-Diagonal Residuals: Overall = 0.01693282

   Population          School       Employment         Services        HouseValue

  0.00815307       0.01813027       0.01382764       0.02151737       0.01960158


                   Partial Correlations Controlling Factors

                 Population         School    Employment         Services       HouseValue

Population          1.00000       -0.17693       0.20752          0.15975          0.02471
School             -0.17693        1.00000       0.30097         -0.12443          0.08614
Employment          0.20752        0.30097       1.00000         -0.07504         -0.27509
Services            0.15975       -0.12443      -0.07504          1.00000          0.22093
HouseValue          0.02471        0.08614      -0.27509          0.22093          1.00000
```

Output 35.2.5 *continued*

```
     Root Mean Square Off-Diagonal Partials: Overall = 0.18550132

   Population            School        Employment          Services         HouseValue

   0.15850824        0.19025867        0.23181838        0.15447043         0.18201538
```

As displayed in Output 35.2.6, the unrotated factor pattern reveals two tight clusters of variables, with the variables HouseValue and School at the negative end of Factor2 axis and the variables Employment and Population at the positive end. The Services variable is in between but closer to the HouseValue and School variables. A good rotation would place the axes so that most variables would have zero loadings on most factors. As a result, the axes would appear as though they are put through the variable clusters.

Output 35.2.6 Unrotated Factor Loading Plot
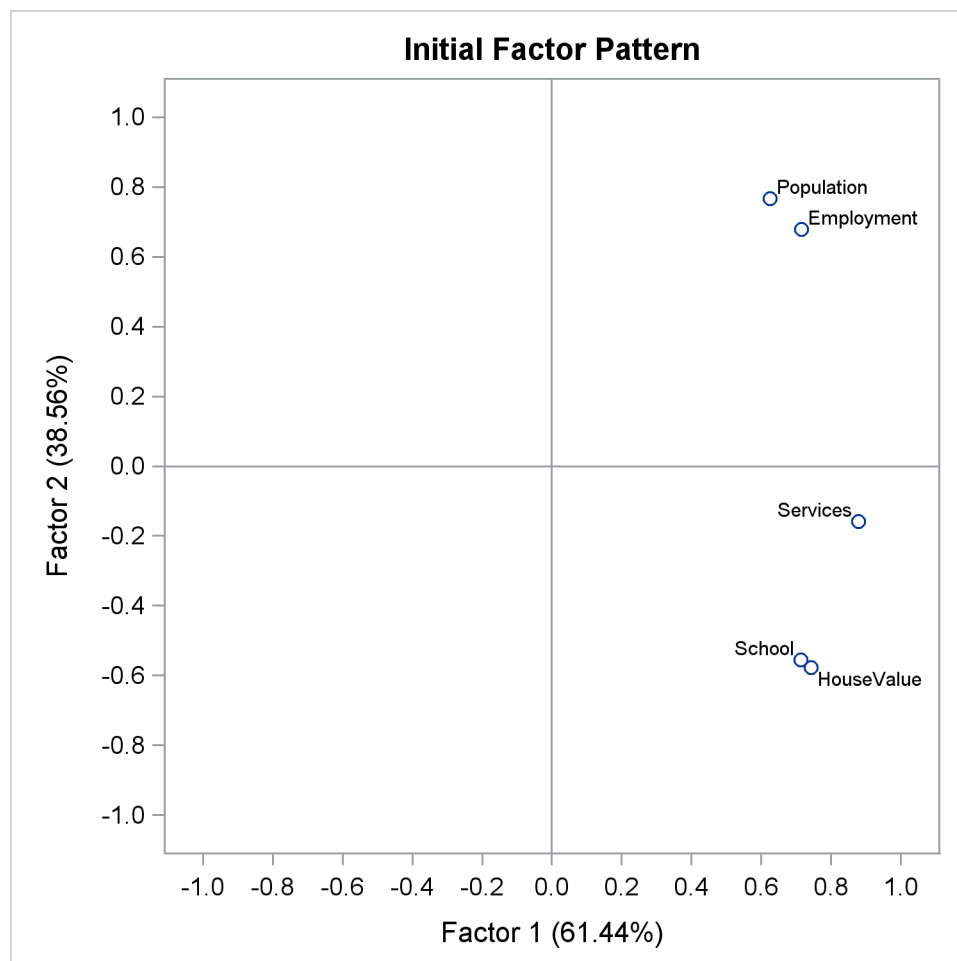
Principal Factor Analysis: Varimax Prerotation

In Output 35.2.7, the results of the varimax prerotation are shown. To yield the varimax-rotated factor loading (pattern), the initial factor loading matrix is postmultiplied by an orthogonal transformation matrix. This orthogonal transformation matrix is shown in Output 35.2.7, followed by the varimax-rotated factor pattern. This rotation or transformation leads to small loadings of Population and Employment on the first factor and small loadings of HouseValue and School on the second factor. Services appears to have a larger loading on the first factor than it has on the second factor, although both loadings are substantial. Hence, Services appears to be factorially complex.

With the REORDER option in effect, you can see the variable clusters clearly in the factor pattern. The first factor is associated more with the first three variables (first three rows of variables): HouseValue, School, and Services. The second factor is associated more with the last two variables (last two rows of variables): Population and Employment.

For orthogonal factor solutions such as the current varimax-rotated solution, you can also interpret the values in the factor loading (pattern) matrix as correlations. For example, HouseValue and Factor 1 have a high correlation at 0.94, while Population and Factor 1 have a low correlation at 0.02.

Output 35.2.7 Varimax Rotation: Transform Matrix and Rotated Pattern

```
                    Orthogonal Transformation Matrix

                                 1                   2

              1              0.78895             0.61446
              2             -0.61446             0.78895


                        Rotated Factor Pattern

                             Factor1              Factor2

        HouseValue           0.94072             -0.00004
        School               0.90419              0.00055
        Services             0.79085              0.41509
        Population           0.02255              0.98874
        Employment           0.14625              0.97499


                    Variance Explained by Each Factor

                    Factor1              Factor2

                  2.3498567            2.1005128


            Final Communality Estimates: Total = 4.450370

     Population          School       Employment        Services        HouseValue

    0.97811334       0.81756387       0.97199928       0.79774304       0.88494998
```

The variance explained by the factors are more evenly distributed in the varimax-rotated solution, as compared with that of the unrotated solution. Indeed, this is a typical fact for any kinds of factor rotation. In the current example, before the varimax rotation the two factors explain 2.73 and 1.72, respectively, of the common variance (see Output 35.2.4). After the varimax rotation the two rotated factors explain 2.35 and 2.10, respectively, of the common variance. However, the total variance accounted for by the factors remains unchanged after the varimax rotation. This invariance property is also observed for the communalities of the variables after the rotation, as evidenced by comparing the current communality estimates in Output 35.2.7 with those in Output 35.2.4.

Output 35.2.8 shows the graphical plot of the varimax-rotated factor loadings. Clearly, HouseValue and School cluster together on the Factor 1 axis, while Population and Employment cluster together on the Factor 2 axis. Service is closer to the cluster of HouseValue and School.

Output 35.2.8 Varimax-Rotated Factor Loadings
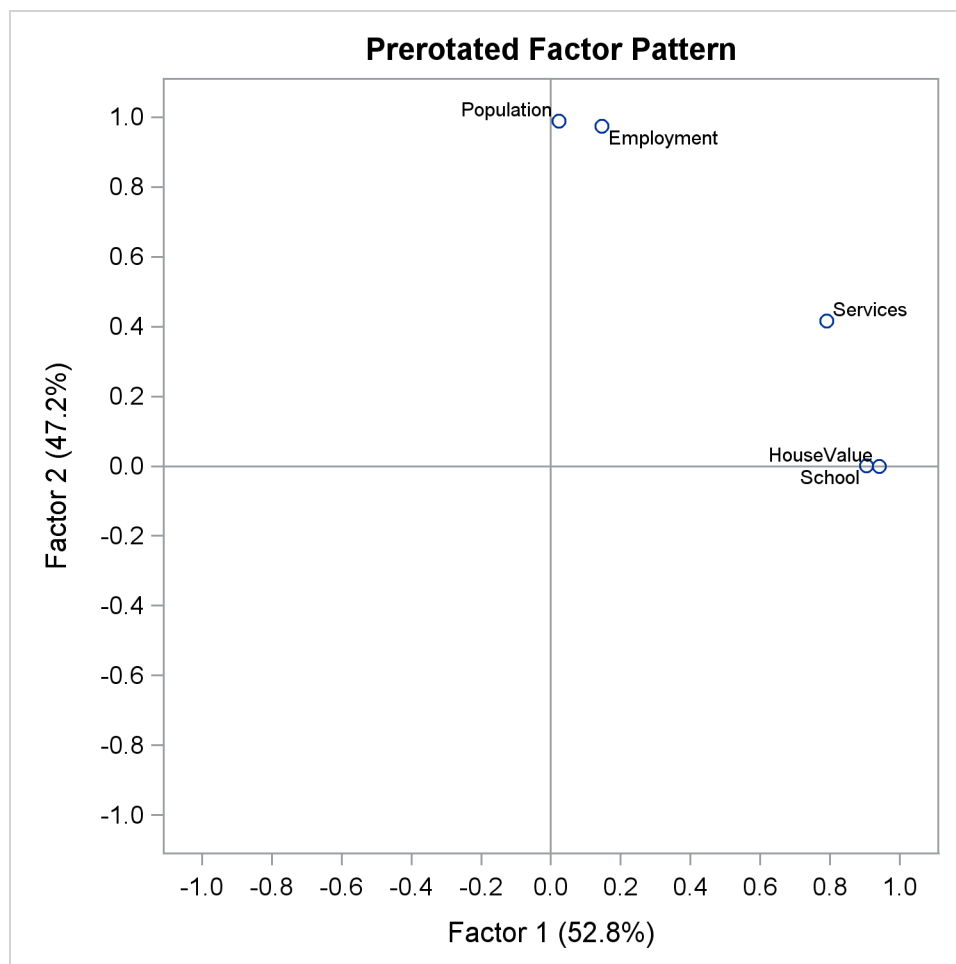
An alternative to the scatter plot of factor loadings is the so-called vector plot of loadings, which is shown in Output 35.2.9. The vector plot is requested with the suboption VECTOR in the PLOTS= option. That is:

```
plots=preloadings(vector)
```

This generates the vector plot of loadings in Output 35.2.9.

Output 35.2.9 Varimax-Rotated Factor Loadings: Vector Plot
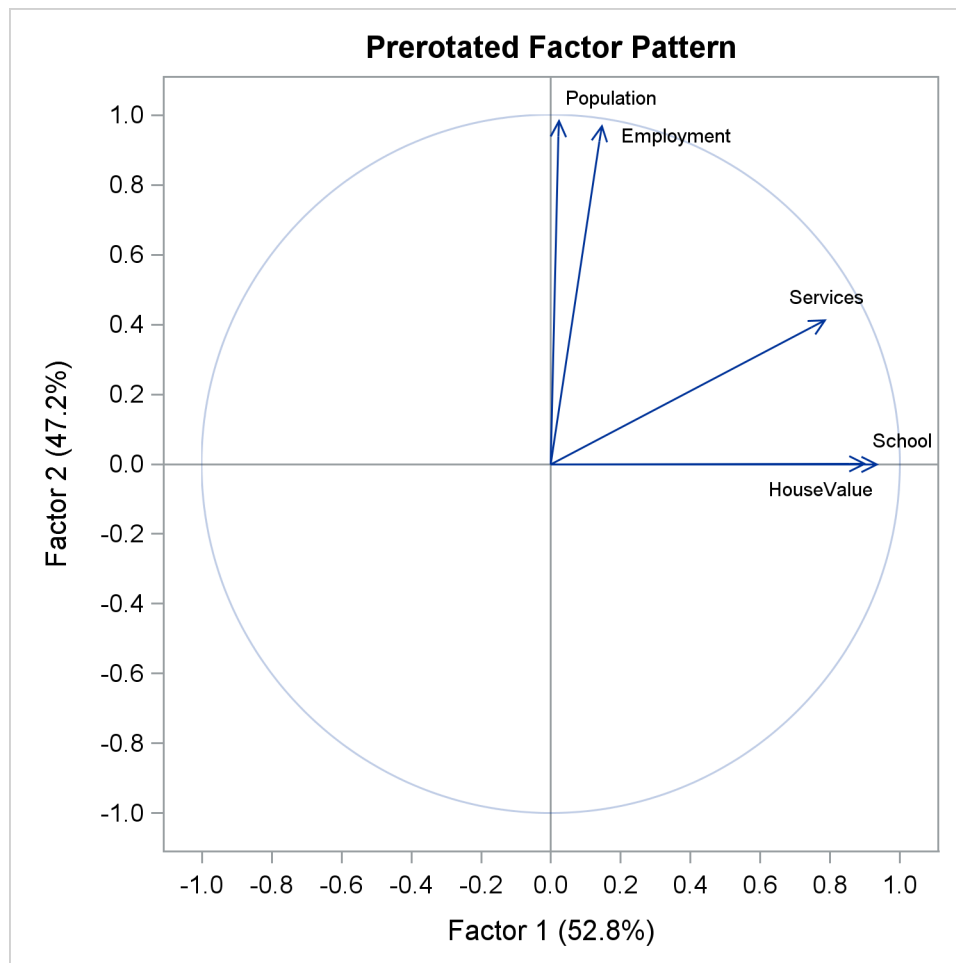
Principal Factor Analysis: Oblique Promax Rotation

For some researchers, the varimax-rotated factor solution in the preceding section might be good enough to provide them useful and interpretable results. For others who believe that common factors are seldom orthogonal, an obliquely rotated factor solution might be more desirable, or at least should be attempted.

PROC FACTOR provides a very large class of oblique factor rotations. The current example shows a particular one—namely, the promax rotation as requested by the ROTATE=PROMAX option.

The results of the promax rotation are shown in Output 35.2.10 and Output 35.2.11. The corresponding plot of factor loadings is shown in Output 35.2.12.

Output 35.2.10 Promax Rotation: Procrustean Target and Transformation

```
         Target Matrix for Procrustean Transformation

                        Factor1              Factor2

      HouseValue         1.00000            -0.00000
      School             1.00000             0.00000
      Services           0.69421             0.10045
      Population          0.00001             1.00000
      Employment          0.00326             0.96793


          Procrustean Transformation Matrix

                          1                    2

            1      1.04116598        -0.0986534
            2     -0.1057226          0.96303019


      Normalized Oblique Transformation Matrix

                          1                    2

            1        0.73803            0.54202
            2       -0.70555            0.86528
```

Output 35.2.10 shows the Procrustean target, to which the varimax factor pattern is rotated, followed by the display of the Procrustean transformation matrix. This is the matrix that transforms the varimax factor pattern so that the rotated pattern is as close as possible to the Procrustean target. However, because the variances of factors have to be fixed at 1 during the oblique transformation, a normalized version of the Procrustean transformation matrix is the one that is actually used in the transformation. This normalized transformation matrix is shown at the bottom of Output 35.2.10. Using this transformation matrix leads to the promax-rotated factor solution, as shown in Output 35.2.11.

Output 35.2.11 Promax Rotation: Factor Correlations and Factor Pattern

```
              Inter-Factor Correlations

                       Factor1              Factor2

      Factor1           1.00000             0.20188
      Factor2           0.20188             1.00000
```

Output 35.2.11 *continued*

```
        Rotated Factor Pattern (Standardized Regression Coefficients)

                                 Factor1              Factor2

             HouseValue        0.95558485          -0.0979201
             School            0.91842142          -0.0935214
             Services          0.76053238           0.33931804
             Population       -0.0790832            1.00192402
             Employment        0.04799             0.97509085
```

After the promax rotation, the factors are no longer uncorrelated. As shown in Output 35.2.11, the correlation of the two factors is now 0.20. In the (initial) unrotated and the varimax solutions, the two factors are not correlated.

In addition to allowing the factors to be correlated, in an oblique factor solution you seek a pattern of factor loadings that is more "differentiated" (referred to as the "simple structures" in the literature). The more differentiated the loadings, the easier the interpretation of the factors.

For example, factor loadings of Services and Population on Factor 2 are 0.415 and 0.989, respectively, in the (orthogonal) varimax-rotated factor pattern (see Output 35.2.7). With the (oblique) promax rotation (see Output 35.2.11), these two loadings become even more differentiated with values 0.339 and 1.002, respectively. Overall, however, the factor patterns before and after the promax rotation do not seem to differ too much. This fact is confirmed by comparing the graphical plots of factor loadings. The plots in Output 35.2.12 (promax-rotated factor loadings) and Output 35.2.8 (varimax-rotated factor loadings) show very similar patterns.

Output 35.2.12 Promax Rotation: Factor Loading Plot
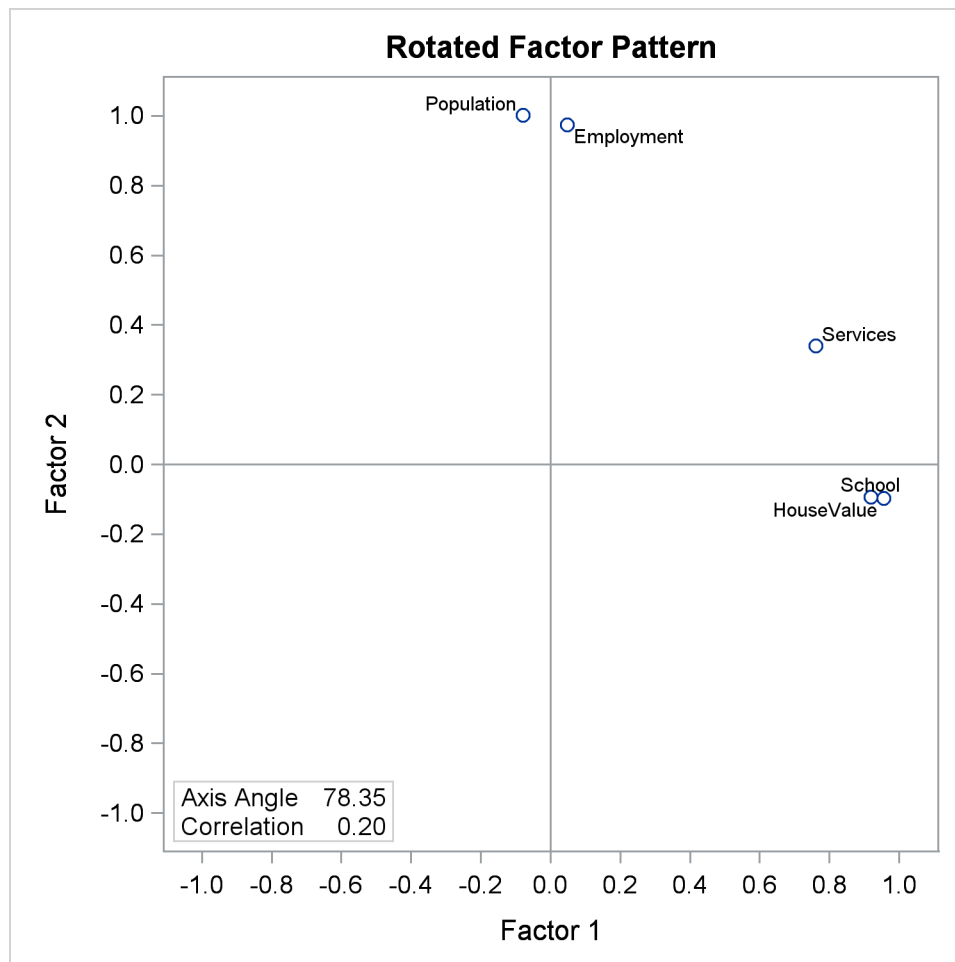
Unlike the orthogonal factor solutions where you can interpret the factor loadings as correlations between variables and factors, in oblique factor solutions such as the promax solution, you have to turn to the factor structure matrix for examining the correlations between variables and factors. Output 35.2.13 shows the factor structures of the promax-rotated solution.

Output 35.2.13 Promax Rotation: Factor Structures and Final Communalities

```
              Factor Structure (Correlations)

                            Factor1          Factor2

        HouseValue          0.93582          0.09500
        School              0.89954          0.09189
        Services            0.82903          0.49286
        Population          0.12319          0.98596
        Employment          0.24484          0.98478
```

Output 35.2.13 *continued*

```
         Variance Explained by Each Factor Ignoring Other Factors

                      Factor1              Factor2

                    2.4473495            2.2022803


            Final Communality Estimates: Total = 4.450370

   Population          School      Employment          Services        HouseValue

   0.97811334      0.81756387      0.97199928        0.79774304        0.88494998
```

Basically, the factor structure matrix shown in Output 35.2.13 reflects a similar pattern to the factor pattern matrix shown in Output 35.2.11. The critical difference is that you can have the correlation interpretation only by using the factor structure matrix. For example, in the factor structure matrix shown in Output 35.2.13, the correlation between Population and Factor 2 is 0.986. The corresponding value shown in the factor pattern matrix in Output 35.2.11 is 1.002, which certainly cannot be interpreted as a correlation coefficient.

Common variance explained by the promax-rotated factors are 2.447 and 2.202, respectively, for the two factors. Unlike the orthogonal factor solutions (for example, the prerotated varimax solution), variance explained by these promax-rotated factors do not sum up to the total communality estimate 4.45. In oblique factor solutions, variance explained by oblique factors cannot be partitioned for the factors. Variance explained by a common factor is computed while ignoring the contributions from the other factors.

However, the communalities for the variables, as shown in the bottom of Output 35.2.13, do not change from rotation to rotation. They are still the same set of communalities in the initial, varimax-rotated, and promax-rotated solutions. This is a basic fact about factor rotations: they only redistribute the variance explained by the factors; the total variance explained by the factors for any variable (that is, the communality of the variable) remains unchanged.

In the literature of exploratory factor analysis, reference axes had been an important tool in factor rotation. Nowadays, rotations are seldom done through the uses of the reference axes. Despite that, results about reference axes do provide additional information for interpreting factor analysis results. For the current example of the promax rotation, PROC FACTOR shows the relevant results about the reference axes in Output 35.2.14.

Output 35.2.14 Promax Rotation: Reference Axis Correlations and Reference Structures

```
                  Reference Axis Correlations

                        Factor1              Factor2

            Factor1    1.00000             −0.20188
            Factor2   −0.20188              1.00000
```

Output 35.2.14 *continued*

```
            Reference Structure (Semipartial Correlations)

                              Factor1            Factor2

        HouseValue            0.93591           -0.09590
        School                0.89951           -0.09160
        Services              0.74487            0.33233
        Population           -0.07745            0.98129
        Employment            0.04700            0.95501


    Variance Explained by Each Factor Eliminating Other Factors

                     Factor1            Factor2

                    2.2480892          2.0030200
```

To explain the results in the reference-axis system, some geometric interpretations of the factor axes are needed. Consider a single factor in a system of n common factors in an oblique factor solution. Taking away the factor under consideration, the remaining $n - 1$ factors span a hyperplane in the factor space of $n - 1$ dimensions. The vector that is orthogonal to this hyperplane is the reference axis (reference vector) of the factor under consideration. Using the same definition for the remaining factors, you have n reference vectors for n factors.

A factor in an oblique factor solution can be considered as the sum of two independent components: its associated reference vector and a component that is overlapped with all other factors. In other words, the reference vector of a factor is a unique part of the factor that is not predictable from all other factors. Thus, the loadings on a reference vector are the unique effects of the corresponding factor, partialling out the effects from all other factors. The variances explained by a reference vector are the unique variances explained by the corresponding factor, partialling out the variances explained by all other factors.

Output 35.2.14 shows the reference axis correlations. The correlation between the reference vectors is – 0.20. Next, Output 35.2.14 shows the loadings on the reference vectors in the table entitled "Reference Structure (Semipartial Correlations)." As explained previously, loadings on a reference vector are also the unique effects of the corresponding factor, partialling out the effects from the all other factors. For example, the unique effect of Factor 1 on HouseValue is 0.936. Another important property of the reference vector system is that loadings on a reference vector are also correlations between the variables and the corresponding factor, partialling out the correlations between the variables and other factors. This means that the loading 0.936 in the reference structure table is the unique correlation between HouseValue and Factor 1, partialling out the correlation between HouseValue with Factor 2. Hence, as suggested by the title of table, all loadings reported in the "Reference Structure (Semipartial Correlations)" can be interpreted as semipartial correlations between variables and factors.

The last table shown in Output 35.2.14 are the variances explained by the reference vectors. As explained previously, these are also unique variances explained by the factors, partialling out the variances explained by all other factors (or eliminating all other factors, as suggested by the title of the table). In the current example, Factor 1 explains 2.248 of the variable variances, partialling out all variable variances explained by Factor 2.

Notice that factor pattern (shown in Output 35.2.11), factor structures (correlations, shown in Output 35.2.13), and reference structures (semipartial correlations, shown in Output 35.2.14) give you different information about the oblique factor solutions such as the promax-rotated solution. However, for orthogonal factor solutions such as the varimax-rotated solution, factor structures and reference structures are all the same as the factor pattern.

Principal Factor Analysis: Factor Rotations with Factor Pattern Input

The promax rotation is one of the many rotations that PROC FACTOR provides. You can specify many different rotation algorithms by using the ROTATE= options. In this section, you explore different rotated factor solutions from the initial principal factor solution. Specifically, you want to examine the factor patterns yielded by the quartimax transformation (an orthogonal transformation) and the Harris-Kaiser (an oblique transformation), respectively.

Rather than analyzing the entire problem again with new rotations, you can simply use the OUTSTAT= data set from the preceding factor analysis results.

First, the OUTSTAT= data set is printed using the following statements:

```
proc print data=fact_all;
run;
```

The output data set is displayed in Output 35.2.15.

Output 35.2.15 Output Data Set

```
                          Factor Output Data Set


                                                               House
 Obs   _TYPE_      _NAME_      Population   School  Employment  Services   Value

  1   MEAN                      6241.67    11.4417    2333.33    120.833  17000.00
  2   STD                       3439.99     1.7865    1241.21    114.928   6367.53
  3   N                           12.00    12.0000      12.00     12.000     12.00
  4   CORR      Population         1.00     0.0098       0.97      0.439      0.02
  5   CORR      School             0.01     1.0000       0.15      0.691      0.86
  6   CORR      Employment         0.97     0.1543       1.00      0.515      0.12
  7   CORR      Services           0.44     0.6914       0.51      1.000      0.78
  8   CORR      HouseValue         0.02     0.8631       0.12      0.778      1.00
  9   COMMUNAL                     0.98     0.8176       0.97      0.798      0.88
 10   PRIORS                       0.97     0.8223       0.97      0.786      0.85
 11   EIGENVAL                     2.73     1.7161       0.04     -0.025     -0.07
 12   UNROTATE  Factor1            0.63     0.7137       0.71      0.879      0.74
 13   UNROTATE  Factor2            0.77    -0.5552       0.68     -0.158     -0.58
 14   RESIDUAL  Population         0.02    -0.0112       0.01      0.011      0.00
 15   RESIDUAL  School            -0.01     0.1824       0.02     -0.024      0.01
 16   RESIDUAL  Employment         0.01     0.0215       0.03     -0.006     -0.02
 17   RESIDUAL  Services           0.01    -0.0239      -0.01      0.202      0.03
 18   RESIDUAL  HouseValue         0.00     0.0125      -0.02      0.034      0.12
 19   PRETRANS  Factor1            0.79    -0.6145         .          .         .
 20   PRETRANS  Factor2            0.61     0.7889         .          .         .
 21   PREROTAT  Factor1            0.02     0.9042       0.15      0.791      0.94
 22   PREROTAT  Factor2            0.99     0.0006       0.97      0.415     -0.00
 23   TRANSFOR  Factor1            0.74    -0.7055         .          .         .
 24   TRANSFOR  Factor2            0.54     0.8653         .          .         .
 25   FCORR     Factor1            1.00     0.2019         .          .         .
 26   FCORR     Factor2            0.20     1.0000         .          .         .
 27   PATTERN   Factor1           -0.08     0.9184       0.05      0.761      0.96
 28   PATTERN   Factor2            1.00    -0.0935       0.98      0.339     -0.10
 29   RCORR     Factor1            1.00    -0.2019         .          .         .
 30   RCORR     Factor2           -0.20     1.0000         .          .         .
 31   REFERENC  Factor1           -0.08     0.8995       0.05      0.745      0.94
 32   REFERENC  Factor2            0.98    -0.0916       0.96      0.332     -0.10
 33   STRUCTUR  Factor1            0.12     0.8995       0.24      0.829      0.94
 34   STRUCTUR  Factor2            0.99     0.0919       0.98      0.493      0.09
```

Various results from the previous factor analysis are saved in this data set, including the initial unrotated solution (its factor pattern is saved in observations with _TYPE_=UNROTATE), the prerotated varimax solution (its factor pattern is saved in observations with _TYPE_=PREROTAT), and the oblique promax solution (its factor pattern is saved in observations with _TYPE_=PATTERN).

When PROC FACTOR reads in an input data set with TYPE=FACTOR, the observations with _TYPE_=PATTERN are treated as the initial factor pattern to be rotated by PROC FACTOR. Hence, it is important that you provide the correct initial factor pattern for PROC FACTOR to read in.

In the current example, you need to provide the unrotated solution from the preceding analysis as the input factor pattern. The following statements create a TYPE=FACTOR data set fact2 from the preceding OUTSTAT= data set fact_all:

```
data fact2(type=factor);
   set fact_all;
   if _TYPE_ in('PATTERN' 'FCORR') then delete;
   if _TYPE_='UNROTATE' then _TYPE_='PATTERN';
   run;
```

In these statements, you delete observations with _TYPE_=PATTERN or _TYPE_=FCORR, which are for the promax-rotated factor solution, and change observations with _TYPE_=UNROTATE to _TYPE_=PATTERN in the new data set fact2. In this way, the initial orthogonal factor pattern matrix is saved in the observations with _TYPE_=PATTERN.

You use this new data set and rotate the initial solution to another oblique solution with the ROTATE=QUARTIMAX option, as shown in the following statements:

```
proc factor data=fact2 rotate=quartimax reorder;
   run;
```

As shown in Output 35.2.16, the input data set is of the FACTOR type for the new rotation.

Output 35.2.16 Quartimax Rotation With Input Factor Pattern

```
           Quartimax Rotation From a TYPE=FACTOR Data Set

                     The FACTOR Procedure

        Input Data Type                        FACTOR
        N Set/Assumed in Data Set                 12
        N for Significance Tests                  12
```

The quartimax-rotated factor pattern is displayed in Output 35.2.17.

Output 35.2.17 Quartimax-Rotated Factor Pattern

```
               Orthogonal Transformation Matrix

                              1                2

              1          0.80138          0.59815
              2         -0.59815          0.80138
```

Output 35.2.17 *continued*

```
                    Rotated Factor Pattern

                          Factor1             Factor2

           HouseValue      0.94052            -0.01933
           School          0.90401            -0.01799
           Services        0.79920             0.39878
           Population       0.04282             0.98807
           Employment       0.16621             0.97179


              Variance Explained by Each Factor

                    Factor1             Factor2

                  2.3699941           2.0803754
```

The quartimax rotation produces an orthogonal transformation matrix shown at the top of Output 35.2.17. After the transformation, the factor pattern is shown next. Compared with the varimax-rotated factor pattern (see Output 35.2.7), the quartimax-rotated factor pattern shows some differences. The loadings of HouseValue and School on Factor 1 drop only slightly in the quartimax factor pattern, while the loadings of Services, Population, and Employment on Factor 1 gain relatively larger amounts. The total variance explained by Factor 1 in the varimax-rotated solution (see Output 35.2.7) is 2.350, while it is 2.370 after the quartimax-rotation. In other words, more variable variances are explained by the first factor in the quartimax factor pattern than in the varimax factor pattern. Although not very strongly demonstrated in the current example, this illustrates a well-known property about the quartimax rotation: it tends to produce a general factor for all variables.

Another oblique rotation is now explored. The Harris-Kaiser transformation weighted by the Cureton-Mulaik technique is applied to the initial factor pattern. To achieve this, you use the ROTATE=HK and NORM=WEIGHT options in the following PROC FACTOR statement:

```
ods graphics on;

proc factor data=fact2 rotate=hk norm=weight reorder
     plots=loadings;
run;

ods graphics off;
```

Output 35.2.18 shows the variable weights in the rotation.

Output 35.2.18 Harris-Kaiser Rotation: Weights

```
                    Variable Weights for Rotation

     Population          School        Employment         Services        HouseValue

     0.95982747        0.93945424      0.99746396        0.12194766        0.94007263
```

While all other variables have weights at least as large as 0.93, the weight for Services is only 0.12. This means that due to its small weight, Services is not as important as the other variables for determining the rotation (transformation). This makes sense when you look at the initial unrotated factor pattern plot in Output 35.2.6. In the plot, there are two main clusters of variables, and Services does not seem to fall into either of the clusters. In order to yield a Harris-Kaiser rotation (transformation) that would gear towards to two clusters, the Cureton-Mulaik weighting essentially downweights the contribution from Services in the factor rotation.

The results of the Harris-Kaiser factor solution are displayed in Output 35.2.19, with a graphical plot of rotated loadings displayed in Output 35.2.20.

Output 35.2.19 Harris-Kaiser Rotation: Factor Correlations and Factor Pattern

```
                    Inter-Factor Correlations

                            Factor1          Factor2

            Factor1         1.00000          0.08358
            Factor2         0.08358          1.00000


    Rotated Factor Pattern (Standardized Regression Coefficients)

                            Factor1          Factor2

            HouseValue      0.94048          0.00279
            School          0.90391          0.00327
            Services        0.75459          0.41892
            Population      -0.06335         0.99227
            Employment      0.06152          0.97885
```

Because the Harris-Kaiser produces an oblique factor solution, you compare the current results with that of the promax (see Output 35.2.11), which also produces an oblique factor solution. The correlation between the factors in the Harris-Kaiser solution is 0.084; this value is much smaller than the same correlation in the promax solution, which is 0.201. However, the Harris-Kaiser rotated factor pattern shown in Output 35.2.19 is more or less the same as that of the promax-rotated factor pattern shown in Output 35.2.11. Which solution would you consider to be more reasonable or interpretable?

From the statistical point of view, the Harris-Kaiser and promax factor solutions are equivalent. They explain the observed variable relationships equally well. From the simplicity point of view, however, you might prefer to interpret the Harris-Kaiser solution because the factor correlation is smaller. In other words, the factors in the Harris-Kaiser solution do not overlap that much conceptually; hence they should be more distinctive to interpret. However, in practice simplicity in factor correlations might not the only principle to consider. Researchers might actually expect to have some factors to be highly correlated based on theoretical or substantive grounds.

Although the Harris-Kaiser and the promax factor patterns are very similar, the graphical plots of the loadings from the two solutions paint slightly different pictures. The plot of the promax-rotated loadings is shown in Output 35.2.12, while the plot of the loadings for the current Harris-Kaiser solution is shown in Output 35.2.20.

Output 35.2.20 Harris-Kaiser Rotation: Factor Loading Plot
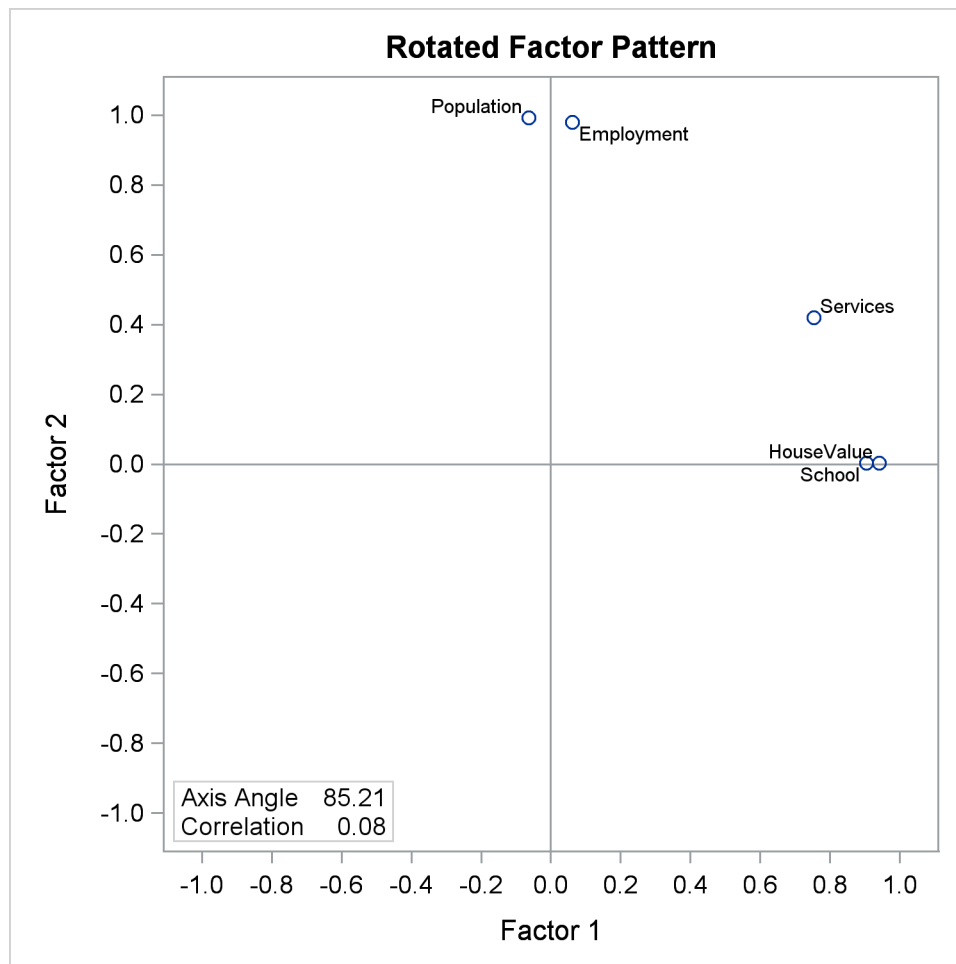
The two factor axes in the Harris-Kaiser rotated pattern (Output 35.2.20) clearly cut through the centers of the two variable clusters, while the Factor 1 axis in the promax solution lies above a variable cluster (Output 35.2.12). The reason for this subtle difference is that in the Harris-Kaiser rotation, the Services is a "loner" that has been downweighted by the Cureton-Mulaik technique (see its relatively small weight in Output 35.2.18). As a result, the rotated axes are basically determined by the two variable clusters in the Harris-Kaiser rotation.

As far as the current discussion goes, it is not recommending one rotation method over another. Rather, it simply illustrates how you could control certain types of characteristics of factor rotation through the many options supported by PROC FACTOR. Should you prefer an orthogonal rotation to an oblique rotation? Should you choose the oblique factor solution with the smallest factor correlations? Should you use a weighting scheme that would enable you to find independent variable clusters? While PROC FACTOR enables you to explore all these alternatives, you must consult advanced textbooks and published articles to get satisfactory and complete answers to these questions.

Example 35.3: Maximum Likelihood Factor Analysis

This example uses maximum likelihood factor analyses for one, two, and three factors. It is already apparent from the principal factor analysis that the best number of common factors is almost certainly two. The one- and three-factor ML solutions reinforce this conclusion and illustrate some of the numerical problems that can occur. The following statements produce Output 35.3.1 through Output 35.3.3:

```
title3 'Maximum Likelihood Factor Analysis with One Factor';
proc factor data=SocioEconomics method=ml heywood n=1;
run;

title3 'Maximum Likelihood Factor Analysis with Two Factors';
proc factor data=SocioEconomics method=ml heywood n=2;
run;

title3 'Maximum Likelihood Factor Analysis with Three Factors';
proc factor data=SocioEconomics method=ml heywood n=3;
run;
```

Output 35.3.1 displays the results of the analysis with one factor.

Output 35.3.1 Maximum Likelihood Factor Analysis

```
          Maximum Likelihood Factor Analysis with One Factor

                     The FACTOR Procedure

          Input Data Type                    Raw Data
          Number of Records Read                   12
          Number of Records Used                   12
          N for Significance Tests                 12




          Maximum Likelihood Factor Analysis with One Factor

                     The FACTOR Procedure
              Initial Factor Method: Maximum Likelihood

               Prior Communality Estimates: SMC

    Population        School       Employment       Services       HouseValue

    0.96859160      0.82228514      0.96918082      0.78572440      0.84701921
```

Output 35.3.1 *continued*

Preliminary Eigenvalues: Total = 76.1165859 Average = 15.2233172

	Eigenvalue	Difference	Proportion	Cumulative
1	63.7010086	50.6462895	0.8369	0.8369
2	13.0547191	12.7270798	0.1715	1.0084
3	0.3276393	0.6749199	0.0043	1.0127
4	−0.3472805	0.2722202	−0.0046	1.0081
5	−0.6195007		−0.0081	1.0000

1 factor will be retained by the NFACTOR criterion.

Iteration	Criterion	Ridge	Change		Communalities		
1	6.5429218	0.0000	0.1033	0.93828	0.72227	1.00000	0.71940
				0.74371			
2	3.1232699	0.0000	0.7288	0.94566	0.02380	1.00000	0.26493
				0.01487			

Convergence criterion satisfied.

Significance Tests Based on 12 Observations

Test	DF	Chi-Square	Pr > ChiSq
H0: No common factors	10	54.2517	<.0001
HA: At least one common factor			
H0: 1 Factor is sufficient	5	24.4656	0.0002
HA: More factors are needed			

Chi-Square without Bartlett's Correction	34.355969
Akaike's Information Criterion	24.355969
Schwarz's Bayesian Criterion	21.931436
Tucker and Lewis's Reliability Coefficient	0.120231

Squared Canonical Correlations

Factor1

1.0000000

Output 35.3.1 *continued*

```
Eigenvalues of the Weighted Reduced Correlation Matrix: Total = 0  Average = 0

                       Eigenvalue     Difference

              1            Infty           Infty
              2       1.92716032      2.15547340
              3       -.22831308      0.56464322
              4       -.79295630      0.11293464
              5       -.90589094


                       Factor Pattern

                                     Factor1

             Population              0.97245
             School                  0.15428
             Employment              1.00000
             Services                0.51472
             HouseValue              0.12193


                 Variance Explained by Each Factor

             Factor          Weighted     Unweighted

             Factor1        17.8010629     2.24926004


             Final Communality Estimates and Variable Weights
      Total Communality: Weighted = 17.801063    Unweighted = 2.249260

             Variable        Communality          Weight

             Population       0.94565561      18.4011648
             School           0.02380349       1.0243839
             Employment       1.00000000           Infty
             Services         0.26493499       1.3604239
             HouseValue       0.01486595       1.0150903
```

The solution on the second iteration is so close to the optimum that PROC FACTOR cannot find a better solution; hence you receive this message:

```
Convergence criterion satisfied.
```

When this message appears, you should try rerunning PROC FACTOR with different prior communality estimates to make sure that the solution is correct. In this case, other prior estimates lead to the same solution or possibly to worse local optima, as indicated by the information criteria or the chi-square values.

The variable Employment has a communality of 1.0 and, therefore, an infinite weight that is displayed next to the final communality estimate as a missing/infinite value. The first eigenvalue is also infinite. Infinite values are ignored in computing the total of the eigenvalues and the total final communality.

Output 35.3.2 displays the results of the analysis with two factors. The analysis converges without incident. This time, however, the Population variable is a Heywood case.

Output 35.3.2 Maximum Likelihood Factor Analysis: Two Factors

```
          Input Data Type                    Raw Data
          Number of Records Read                   12
          Number of Records Used                   12
          N for Significance Tests                 12


                  Prior Communality Estimates: SMC

   Population           School      Employment        Services       HouseValue

  0.96859160        0.82228514      0.96918082      0.78572440       0.84701921


      Preliminary Eigenvalues: Total = 76.1165859  Average = 15.2233172

                Eigenvalue     Difference     Proportion     Cumulative

          1     63.7010086     50.6462895         0.8369         0.8369
          2     13.0547191     12.7270798         0.1715         1.0084
          3      0.3276393      0.6749199         0.0043         1.0127
          4     -0.3472805      0.2722202        -0.0046         1.0081
          5     -0.6195007                        -0.0081         1.0000


        2 factors will be retained by the NFACTOR criterion.



Iteration    Criterion    Ridge    Change                  Communalities

    1        0.3431221    0.0000    0.0471    1.00000    0.80672    0.95058    0.79348
                                              0.89412
    2        0.3072178    0.0000    0.0307    1.00000    0.80821    0.96023    0.81048
                                              0.92480
    3        0.3067860    0.0000    0.0063    1.00000    0.81149    0.95948    0.81677
                                              0.92023
    4        0.3067373    0.0000    0.0022    1.00000    0.80985    0.95963    0.81498
                                              0.92241
    5        0.3067321    0.0000    0.0007    1.00000    0.81019    0.95955    0.81569
                                              0.92187


      Convergence criterion satisfied.
```

Output 35.3.2 *continued*

```
            Significance Tests Based on 12 Observations

                                                      Pr >
                   Test                  DF   Chi-Square   ChiSq

   H0: No common factors                 10     54.2517   <.0001
   HA: At least one common factor
   H0: 2 Factors are sufficient           1      2.1982   0.1382
   HA: More factors are needed


      Chi-Square without Bartlett's Correction      3.3740530
      Akaike's Information Criterion                 1.3740530
      Schwarz's Bayesian Criterion                   0.8891463
      Tucker and Lewis's Reliability Coefficient     0.7292200


                  Squared Canonical Correlations

                    Factor1           Factor2

                  1.0000000         0.9518891


         Eigenvalues of the Weighted Reduced Correlation
         Matrix: Total = 19.7853157  Average = 4.94632893

          Eigenvalue    Difference    Proportion    Cumulative

     1        Infty         Infty
     2    19.7853143    19.2421292        1.0000        1.0000
     3     0.5431851     0.5829564        0.0275        1.0275
     4    -0.0397713     0.4636411       -0.0020        1.0254
     5    -0.5034124                     -0.0254        1.0000


                          Factor Pattern

                            Factor1           Factor2

        Population          1.00000           0.00000
        School              0.00975           0.90003
        Employment          0.97245           0.11797
        Services            0.43887           0.78930
        HouseValue          0.02241           0.95989


               Variance Explained by Each Factor

          Factor        Weighted    Unweighted

          Factor1     24.4329707    2.13886057
          Factor2     19.7853143    2.36835294
```

Output 35.3.2 *continued*

```
            Final Communality Estimates and Variable Weights
       Total Communality: Weighted = 44.218285    Unweighted = 4.507214


                    Variable        Communality          Weight

                    Population       1.00000000            Infty
                    School           0.81014489         5.2682940
                    Employment       0.95957142        24.7246669
                    Services         0.81560348         5.4256462
                    HouseValue       0.92189372        12.7996793
```

The results of the three-factor analysis are shown in Output 35.3.3.

Output 35.3.3 Maximum Likelihood Factor Analysis: Three Factors

```
                Input Data Type                    Raw Data
                Number of Records Read                  12
                Number of Records Used                  12
                N for Significance Tests                12


                    Prior Communality Estimates: SMC

     Population          School       Employment         Services       HouseValue

    0.96859160       0.82228514       0.96918082       0.78572440       0.84701921


       Preliminary Eigenvalues: Total = 76.1165859   Average = 15.2233172

                 Eigenvalue    Difference    Proportion    Cumulative

            1    63.7010086    50.6462895        0.8369        0.8369
            2    13.0547191    12.7270798        0.1715        1.0084
            3     0.3276393     0.6749199        0.0043        1.0127
            4    -0.3472805     0.2722202       -0.0046        1.0081
            5    -0.6195007                      -0.0081        1.0000
```

Output 35.3.3 *continued*

```
            3 factors will be retained by the NFACTOR criterion.


          WARNING: Too many factors for a unique solution.



Iteration    Criterion    Ridge    Change              Communalities

    1        0.1798029    0.0313   0.0501    0.96081   0.84184   1.00000   0.80175
                                             0.89716
    2        0.0016405    0.0313   0.0678    0.98081   0.88713   1.00000   0.79559
                                             0.96500
    3        0.0000041    0.0313   0.0094    0.98195   0.88603   1.00000   0.80498
                                             0.96751
    4        0.0000000    0.0313   0.0006    0.98202   0.88585   1.00000   0.80561
                                             0.96735


        ERROR: Converged, but not to a proper optimum.


               Try a different 'PRIORS' statement.



                 Significance Tests Based on 12 Observations


                                                          Pr >
                 Test                    DF    Chi-Square    ChiSq

       H0: No common factors            10       54.2517    <.0001
       HA: At least one common factor
       H0: 3 Factors are sufficient     -2        0.0000      .
       HA: More factors are needed


          Chi-Square without Bartlett's Correction       0.0000003
          Akaike's Information Criterion                  4.0000003
          Schwarz's Bayesian Criterion                   4.9698136
          Tucker and Lewis's Reliability Coefficient     0.0000000


                      Squared Canonical Correlations

              Factor1          Factor2          Factor3

            1.0000000        0.9751895        0.6894465
```

Output 35.3.3 *continued*

```
          Eigenvalues of the Weighted Reduced Correlation
          Matrix: Total = 41.5254193  Average = 10.3813548

             Eigenvalue    Difference    Proportion    Cumulative

      1         Infty         Infty
      2     39.3054826    37.0854258        0.9465        0.9465
      3      2.2200568     2.2199693        0.0535        1.0000
      4      0.0000875     0.0002949        0.0000        1.0000
      5     -0.0002075                     -0.0000        1.0000


                         Factor Pattern

                      Factor1          Factor2          Factor3

   Population         0.97245         -0.11233         -0.15409
   School             0.15428          0.89108          0.26083
   Employment         1.00000          0.00000          0.00000
   Services           0.51472          0.72416         -0.12766
   HouseValue         0.12193          0.97227         -0.08473


                  Variance Explained by Each Factor

             Factor        Weighted      Unweighted

             Factor1      54.6115241     2.24926004
             Factor2      39.3054826     2.27634375
             Factor3       2.2200568     0.11525433


           Final Communality Estimates and Variable Weights
     Total Communality: Weighted = 96.137063   Unweighted = 4.640858

             Variable      Communality        Weight

             Population     0.98201660     55.6066901
             School         0.88585165      8.7607194
             Employment     1.00000000          Infty
             Services       0.80564301      5.1444261
             HouseValue     0.96734687     30.6251078
```

In the results, a warning message is displayed:

```
WARNING:  Too many factors for a unique solution.
```

The number of parameters in the model exceeds the number of elements in the correlation matrix from which they can be estimated, so an infinite number of different perfect solutions can be obtained. The criterion approaches zero at an improper optimum, as indicated by this message:

```
Converged, but not to a proper optimum.
```

The degrees of freedom for the chi-square test are –2, so a probability level cannot be computed for three factors. Note also that the variable Employment is a Heywood case again.

The probability levels for the chi-square test are 0.0001 for the hypothesis of no common factors, 0.0002 for one common factor, and 0.1382 for two common factors. Therefore, the two-factor model seems to be an adequate representation. Akaike's information criterion and Schwarz's Bayesian criterion attain their minimum values at two common factors, so there is little doubt that two factors are appropriate for these data.

Example 35.4: Using Confidence Intervals to Locate Salient Factor Loadings

This example illustrates how you can use the standard errors and confidence intervals to understand the pattern of factor loadings under the maximum likelihood estimation. There are nine tests and you want a three-factor solution (N=3) for a correlation matrix based on 200 observations. The following statements define the input data set and specify the desirable analysis by the FACTOR procedure:

```
data test(type=corr);
   title 'Quartimin-Rotated Factor Solution with Standard Errors';
   input _name_ $ test1-test9;
   _type_ = 'corr';
   datalines;
Test1      1   .561   .602   .290   .404   .328   .367   .179 -.268
Test2    .561     1   .743   .414   .526   .442   .523   .289 -.399
Test3    .602   .743     1   .286   .343   .361   .679   .456 -.532
Test4    .290   .414   .286     1   .677   .446   .412   .400 -.491
Test5    .404   .526   .343   .677     1   .584   .408   .299 -.466
Test6    .328   .442   .361   .446   .584     1   .333   .178 -.306
Test7    .367   .523   .679   .412   .408   .333     1   .711 -.760
Test8    .179   .289   .456   .400   .299   .178   .711     1 -.725
Test9   -.268 -.399 -.532 -.491 -.466 -.306 -.760 -.725     1
;

title2 'A nine-variable-three-factor example';
proc factor data=test method=ml reorder rotate=quartimin
   nobs=200 n=3 se cover=.45 alpha=.1;
run;
```

In the PROC FACTOR statement, you apply quartimin rotation with (default) Kaiser normalization. You define loadings with magnitudes greater than 0.45 to be salient (COVER=0.45) and use 90% confidence intervals (ALPHA=0.1) to judge the salience. The REORDER option is specified so that variables that have similar loadings with factors are clustered together.

After the quartimin rotation, the correlation matrix for factors is shown in Output 35.4.1.

Output 35.4.1 Quartimin-Rotated Factor Solution with Standard Errors

```
              Inter-Factor Correlations
             With 90% confidence limits
             Estimate/StdErr/LowerCL/UpperCL


                 Factor1          Factor2          Factor3


     Factor1     1.00000          0.41283          0.38304
                 0.00000          0.06267          0.06060
                    .             0.30475          0.27919
                    .             0.51041          0.47804


     Factor2     0.41283          1.00000          0.47006
                 0.06267          0.00000          0.05116
                 0.30475             .             0.38177
                 0.51041             .             0.54986


     Factor3     0.38304          0.47006          1.00000
                 0.06060          0.05116          0.00000
                 0.27919          0.38177             .
                 0.47804          0.54986             .
```

The factors are medium to highly correlated. The confidence intervals seem to be very wide, suggesting that the estimation of factor correlations might not be very accurate for this sample size. For example, the 90% confidence interval for the correlation between Factor1 and Factor2 is (0.30, 0.51), a range of 0.21. You might need a larger sample to get a narrower interval, or you might need a better estimation.

Next, coverage displays for factor loadings are shown in Output 35.4.2.

Output 35.4.2 Using the Rotated Factor Pattern to Interpret the Factors

```
      Rotated Factor Pattern (Standardized Regression Coefficients)
             With 90% confidence limits; Cover |*| = 0.45?
           Estimate/StdErr/LowerCL/UpperCL/Coverage Display


                    Factor1            Factor2            Factor3

      test8         0.86810           -0.05045            0.00114
                    0.03282            0.03185            0.03087
                    0.80271           -0.10265           -0.04959
                    0.91286            0.00204            0.05187
                      0*[]               *[0]               [0]*

      test7         0.73204            0.27296            0.01098
                    0.04434            0.05292            0.03838
                    0.65040            0.18390           -0.05211
                    0.79697            0.35758            0.07399
                      0*[]               0[]*               [0]*

      test9        -0.79654           -0.01230           -0.17307
                    0.03948            0.04225            0.04420
                   -0.85291           -0.08163           -0.24472
                   -0.72180            0.05715           -0.09955
                      []*0               *[0]               *[]0

      test3         0.27715            0.91156           -0.19727
                    0.05489            0.04877            0.02981
                    0.18464            0.78650           -0.24577
                    0.36478            0.96481           -0.14778
                      0[]*               0*[]               *[]0

      test2         0.01063            0.71540            0.20500
                    0.05060            0.05148            0.05496
                   -0.07248            0.61982            0.11310
                    0.09359            0.79007            0.29342
                      [0]*               0*[]               0[]*

      test1        -0.07356            0.63815            0.13983
                    0.04245            0.05380            0.05597
                   -0.14292            0.54114            0.04682
                   -0.00348            0.71839            0.23044
                      *[]0               0*[]               0[]*

      test5         0.00863            0.03234            0.91282
                    0.04394            0.04387            0.04509
                   -0.06356           -0.03986            0.80030
                    0.08073            0.10421            0.96323
                      [0]*               [0]*               0*[]

      test4         0.22357           -0.07576            0.67925
                    0.05956            0.03640            0.05434
                    0.12366           -0.13528            0.57955
                    0.31900           -0.01569            0.75891
                      0[]*               *[]0               0*[]

      test6        -0.04295            0.21911            0.53183
                    0.05114            0.07481            0.06905
                   -0.12656            0.09319            0.40893
                    0.04127            0.33813            0.63578
                      *[0]               0[]*               0[*]
```

The coverage displays in Output 35.4.2 show that Test8, Test7, and Test9 have salient relationships with Factor1. The coverage displays are either '0*[ ]' or '[ ]*0', indicating that the entire 90% confidence intervals for the corresponding loadings are beyond the salience value at 0.45. On the other hand, the coverage display for Test3 on Factor1 is '0[ ]*'. This indicates that even though the loading estimate is significantly larger than zero, it is not large enough to be salient. Similarly, Test3, Test2, and Test1 have salient relationships with Factor2, while Test5 and Test4 have salient relationships with Factor3. For Test6, its relationship with Factor3 is a little bit ambiguous; the 90% confidence interval approximately covers values between 0.40 and 0.64. This means that the population value might have been smaller or larger than 0.45. It is marginal evidence for a salient relationship.

For oblique factor solutions, some researchers prefer to examine the factor structure loadings, which represent correlations, for determining salient relationships. In Output 35.4.3, the factor structure loadings and the associated standard error estimates and coverage displays are shown.

Output 35.4.3 Using the Factor Structure to Interpret the Factors

```
                Factor Structure (Correlations)
          With 90% confidence limits; Cover |*| = 0.45?
         Estimate/StdErr/LowerCL/UpperCL/Coverage Display


                  Factor1            Factor2            Factor3

  test8           0.84771            0.30847            0.30994
                  0.02871            0.06593            0.06263
                  0.79324            0.19641            0.20363
                  0.88872            0.41257            0.40904
                    0*[]               0[]*               0[]*


  test7           0.84894            0.58033            0.41970
                  0.02688            0.05265            0.06060
                  0.79834            0.48721            0.31523
                  0.88764            0.66041            0.51412
                    0*[]               0*[]               0[*]


  test9          -0.86791           -0.42248           -0.48396
                  0.02522            0.06187            0.05504
                 -0.90381           -0.51873           -0.56921
                 -0.81987           -0.31567           -0.38841
                    []*0               [*]0               [*]0


  test3           0.57790            0.93325            0.33738
                  0.05069            0.02953            0.06779
                  0.48853            0.86340            0.22157
                  0.65528            0.96799            0.44380
                    0*[]               0*[]               0[]*


  test2           0.38449            0.81615            0.54535
                  0.06143            0.03106            0.05456
                  0.27914            0.75829            0.44946
                  0.48070            0.86126            0.62883
                    0[*]               0*[]               0[*]


  test1           0.24345            0.67351            0.41162
                  0.06864            0.04284            0.05995
                  0.12771            0.59680            0.30846
                  0.35264            0.73802            0.50522
                    0[]*               0*[]               0[*]


  test5           0.37163            0.46498            0.93132
                  0.06092            0.04979            0.03277
                  0.26739            0.37923            0.85159
                  0.46727            0.54282            0.96894
                    0[*]               0[*]               0*[]


  test4           0.45248            0.33583            0.72927
                  0.05876            0.06289            0.04061
                  0.35072            0.22867            0.65527
                  0.54367            0.43494            0.78941
                    0[*]               0[]*               0*[]


  test6           0.25122            0.45137            0.61837
                  0.07140            0.05858            0.05051
                  0.13061            0.34997            0.52833
                  0.36450            0.54232            0.69465
                    0[]*               0[*]               0*[]
```

The interpretations based on the factor structure matrix do not change much from that based on the factor loadings except for Test3 and Test9. Test9 now has a salient correlation with Factor3. For Test3, it has salient correlations with both Factor1 and Factor2. Fortunately, there are still tests that have salient correlations only with either Factor1 or Factor2 (but not both). This would make interpretations of factors less problematic.

References

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in Petrov and Csaki, eds., *Proceedings of the Second International Symposium on Information Theory*, 267–281.

Akaike, H. (1974), "A New Look at the Statistical Model Identification," *IEEE Transaction on Automatic Control*, AC-19, 716–723.

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.

Archer, C. O. and Jennrich, R. I. (1973), "Standard Errors for Orthogonally Rotated Factor Loadings," *Psychometrika*, 38, 581–592.

Bickel, P. J. and Doksum, K. A. (1977), *Mathematical Statistics*, San Francisco: Holden-Day.

Browne, M. W. (1982), "Covariance Structures," in D. M. Hawkins, ed., *Topics in Applied Multivariate Analysis*, 72–141, Cambridge: Cambridge University Press.

Browne, M. W., Cudeck, R., Tateneni, K., and Mels, G. (2008), "CEFA: Comprehensive Exploratory Factor Analysis, Version 3.02," retrieved from: http://faculty.psy.ohio-state.edu/browne/programs.htm.

Cattell, R. B. (1966), "The Scree Test for the Number of Factors," *Multivariate Behavioral Research*, 1, 245–276.

Cattell, R. B. (1978), *The Scientific Use of Factor Analysis*, New York: Plenum.

Cattell, R. B. and Vogelman, S. (1977), "A Comprehensive Trial of the Scree and KG Criteria for Determining the Number of Factors," *Multivariate Behavioral Research*, 12, 289–325.

Cerny, B. A. and Kaiser, H. F. (1977), "A Study of a Measure of Sampling Adequacy for Factor-Analytic Correlation Matrices," *Multivariate Behavioral Research*, 12, 43–47.

Crawford, C. B. and Ferguson, G. A. (1970), "A General Rotation Criterion and Its Use in Orthogonal Rotation," *Psychometrika*, 35, 321–332.

Cureton, E. E. (1968), *A Factor Analysis of Project TALENT Tests and Four Other Test Batteries*, Interim Report 4 to the U.S. Office of Education, Cooperative Research Project No. 3051.) Palo Alto, CA: Project TALENT Office, American Institutes for Research and University of Pittsburgh.

Cureton, E. E. and Mulaik, S. A. (1975), "The Weighted Varimax Rotation and the Promax Rotation," *Psychometrika*, 40, 183–195.

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

Fuller, W. A. (1987), *Measurement Error Models*, New York: John Wiley & Sons.

Geweke, J. F. and Singleton, K. J. (1980), "Interpreting the Likelihood Ratio Statistic in Factor Models When Sample Size Is Small," *Journal of the American Statistical Association*, 75, 133–137.

Gorsuch, R. L. (1974), *Factor Analysis*, Philadelphia: W. B. Saunders.

Harman, H. H. (1976), *Modern Factor Analysis*, Third Edition, Chicago: University of Chicago Press.

Harris, C. W. (1962), "Some Rao-Guttman Relationships," *Psychometrika*, 27, 247–263.

Hayashi, K. and Yung, Y. F. (1999), "Standard Errors for the Class of Orthomax-Rotated Factor Loadings: Some Matrix Results," *Psychometrika*, 64, 451–460.

Horn, J. L. and Engstrom, R. (1979), "Cattell's Scree Test in Relation to Bartlett's Chi-Square Test and Other Observations on the Number of Factors Problem," *Multivariate Behavioral Research*, 14, 283–300.

Jennrich, R. I. (1973), "Standard Errors for Obliquely Rotated Factor Loadings," *Psychometrika*, 38, 593–604.

Jennrich, R. I. (1974), "Simplified Formulae for Standard Errors in Maximum-Likelihood Factor Analysis," *British Journal of Mathematical and Statistical Psychology*, 27, 122–131.

Jöreskog, K. G. (1977), "Factor Analysis by Least-Squares and Maximum Likelihood Methods," in K. Enslein, A. Ralston, and H. S. Wilf, eds., *Statistical Methods for Digital Computers*, New York: John Wiley & Sons.

Kaiser, H. F. (1963), "Image Analysis," in C. W. Harris, ed., *Problems in Measuring Change*, Madison: University of Wisconsin Press.

Kaiser, H. F. (1970), "A Second Generation Little Jiffy," *Psychometrika*, 35, 401–415.

Kaiser, H. F. and Rice, J. (1974), "Little Jiffy, Mark IV," *Educational and Psychological Measurement*, 34, 111–117.

Kerlinger, F. N. and Pedhazur, E. J. (1973), *Multiple Regression in Behavioral Research*, New York: Holt, Rinehart & Winston.

Kim, J. O. and Mueller, C. W. (1978a), *Factor Analysis: Statistical Methods and Practical Issues*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-014, Beverly Hills: Sage Publications.

Kim, J. O. and Mueller, C. W. (1978b), *Introduction to Factor Analysis: What It Is and How To Do It*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-013, Beverly Hills: Sage Publications.

Lawley, D. N. and Maxwell, A. E. (1971), *Factor Analysis as a Statistical Method*, New York: Macmillan.

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

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

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

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

Rao, C. R. (1955), "Estimation and Tests of Significance in Factor Analysis," *Psychometrika*, 20, 93–111.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Spearman, C. (1904), "General Intelligence Objectively Determined and Measured," *American Journal of Psychology*, 15, 201–293.

Stewart, D. W. (1981), "The Application and Misapplication of Factor Analysis in Marketing Research," *Journal of Marketing Research*, 18, 51–62.

Tucker, L. R. and Lewis, C. (1973), "A Reliability Coefficient for Maximum Likelihood Factor Analysis," *Psychometrika*, 38, 1–10.

Yung, Y. F. and Hayashi, K. (2001), "A Computationally Efficient Method for Obtaining Standard Error Estimates for the Promax and Related Solutions," *British Journal of Mathematical and Statistical Psychology*, 54, 125–138.

Subject Index

Syntax Index

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.

SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.

SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

support.sas.com/publishing

SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

support.sas.com/spn

§sas. | THE POWER TO KNOW®