CHAPTER 16
FACTOR ANALYSIS

16.1 INTRODUCTION

Factor analysis (FA) is similar to principal component analysis in the sense that it leads to the deduction of a new, smaller set of variables that almost describes the behavior given by the original set. However, FA is different because it does not attempt to find transformations for the given variables but aims to discover internal or hidden *factors,* which would have resulted in the current set of variables (Johnson and Wichern 2001).

Historically, factor analysis was proposed as a method to investigate intelligence as the hidden factor behind the ability of students to achieve certain levels of learning in fields such as mathematics, music, and languages. It was controversial at the time because of the interpretations' specifics, which were attached to the *discovered* factors. However, most of these arguments have been abandoned and FA is used systematically in a wide range of problems, including our subject—data reduction. Another important use of factor analysis is as a process of data exploration, which is simply called *exploratory factor analysis* (Johnson and Wichern 2001). We do not deal with the exploratory aspect of FA, but limit our presentation to its use in data reduction.

16.1.1 BASIC MODEL

The basic idea behind factor analysis is to attempt to find a set of hidden *factors* such that the currently observed variables are recovered by performing a set of *linear transformations* on these factors. Mathematically, given the set of *observed* variables $x_1, x_2, \ldots, x_p$, factor analysis attempts to find the set of factors $f_1, f_2, \ldots, f_m$, such that

$$\begin{aligned} x_1 - \mu_1 &= l_{11} f_1 + l_{12} f_2 + \cdots + l_{1m} f_m + \varepsilon_1 \\ x_2 - \mu_2 &= l_{21} f_1 + l_{22} f_2 + \cdots + l_{2m} f_m + \varepsilon_2 \\ &\vdots \quad \vdots \\ x_p - \mu_p &= l_{p1} f_1 + l_{p2} f_2 + \cdots + l_{pm} f_m + \varepsilon_p, \end{aligned} \tag{16.1}$$

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

where $\mu_1, \mu_2, \ldots, \mu_p$ are the means of the variables $x_1, x_2, \ldots, x_p$, and the terms $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p$ represent the unobservable part of variables $x_1, x_2, \ldots$, which are also called *specific factors.* The terms l_{ij}, $i = 1, \ldots, p, j = 1, \ldots, m$ are known as the loadings. The factors $f_1, f_2, \ldots, f_m$ are known as the *common factors.*

Equation 16.1 can be written in matrix form as

$$\mathbf{X} - \boldsymbol{\mu} = \mathbf{LF} + \boldsymbol{\varepsilon}. \tag{16.2}$$

Therefore, one could state the factor analysis problem as follows: Given the observed variables $\mathbf{X}$, along with their mean μ, we attempt to find the set of factors $\mathbf{F}$ and the associated loadings $\mathbf{L}$, such that Equation 16.2 is valid.

The different methods of estimating the loading matrix $\mathbf{L}$ and the vector of factors $\mathbf{F}$ rely on imposing some restrictions on their statistical properties. The following are the most common restrictions.

1. All the factors are independent, with zero mean and variance of unity.
2. All the error terms are also independent, with zero mean and constant variance.
3. The errors are independent of the factors.

With these three restrictions, the factor model of Equation 16.2 constitutes what is known as the *orthogonal factor model.*

Denoting the variance of the ith specific factor, ε_i, by ψ_i, it can be shown that the variance of the ith variable, x_i, is given as

$$\underbrace{\sigma_{ii}}_{\text{Var}(x_i)} = \underbrace{l_{i1}^2 + l_{i2}^2 + \cdots + l_{im}^2}_{\text{Communality}} + \underbrace{\psi_i}_{\substack{\text{Specific} \\ \text{Variance}}}. \tag{16.3}$$

Equation 16.3 shows that the variance of each observable variable is decomposed to the *communality* and the variance of the specific factor. The communality is usually given the symbol h_i^2 for the ith variable, namely,

$$h_i^2 = l_{i1}^2 + l_{i2}^2 + \cdots + l_{im}^2. \tag{16.4}$$

In evaluating the factor model, one usually performs two checks. First, the analysis of the communality shows how much of the variance of the variable is explained by the proposed set of factors. When a small proportion of the variance is explained by the factors, the factor model is suspect, to say the least. The second check is to examine the factors and try to interpret them in terms of their ability to explain the correlation between the variables.

High values of the factor loadings indicate a strong relationship between that factor and the associated variables. A cut-off value of 0.4 is commonly used to determine the significance of a factor in explaining observed variables. However, sometimes the resulting factors show no particular pattern; in such a case, *rotating* the factors is attempted. This method of *factor rotation* is explained next.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

16.1.2 Factor Rotation

Any orthogonal matrix $\mathbf{T}$ such that $\mathbf{T}'\mathbf{T} = \mathbf{T}\mathbf{T}' = \mathbf{I}$ could be introduced into Equation 16.2 without changing the model structure. In this case, we may write the factor model as

$$\mathbf{X} - \boldsymbol{\mu} = \mathbf{L}\mathbf{F} + \boldsymbol{\varepsilon} = \mathbf{L}\mathbf{T}\mathbf{T}'\mathbf{F} + \boldsymbol{\varepsilon} = \mathbf{L}^*\mathbf{F}^* + \boldsymbol{\varepsilon}, \tag{16.5}$$

where $\mathbf{L}^*$ and $\mathbf{F}^*$ are the new loading matrix and vector of factors defined as

$$\mathbf{L}^* = \mathbf{L}\mathbf{T} \quad \text{and} \quad \mathbf{F}^* = \mathbf{T}'\mathbf{F}.$$

An orthogonal transformation matrix, such as $\mathbf{T}$, represents *rotation* of axes. Such rotation could be designed to allow the factors to have certain desired properties. These rotations could lead to easier interpretation of the model.

16.1.3 Estimation Methods

There are two methods for solving the factor model equations for the matrix $\mathbf{L}$ and the factors $\mathbf{F}$: (1) the maximum likelihood (ML) method and (2) the principal component method. The maximum likelihood method assumes that both the common and specific factors, and therefore the observed variables, are normally distributed. The validity of this assumption cannot be guaranteed for all datasets. Another limitation of the ML method is that it is computationally expensive. In fact, the SAS documentation mentions that it takes on average *100 times* the time needed to solve a problem as the principal component method takes.

The computational algorithm involves iterative procedures that may not even converge (see example in Section 16.1.5). However, the ML method allows the evaluation of the number factors' significance and is based on robust theoretical foundations. These two advantages usually make it the method of choice for statisticians. On the other hand, the principal component method is very fast, easy to interpret, and guaranteed to find a solution for all datasets.

16.1.4 Variable Standardization

Because we are not really interested in the scale of magnitude of each variable, but rather in the presence or absence of correlation between the variables, factor analysis starts by standardizing the variables. The variables are standardized such that they have a zero mean and a variance of one. In this way, the total variance of p observed variables is also equal to p. For example, starting with 7 observed variables, `PROC FACTOR` will standardize them such that when the results of the factor analysis are displayed, we speak of explaining, say 6.5, of the total variance of 7.0.

16.1.5 Illustrative Example

Recall the banking dataset of Table 15.1. The dataset contains the monthly summary of credit card average balance (`AvgBalance`), average value of the credit card transactions

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

(AvgTransValue), average value of monthly interest (AvgInt), and average balance of checking account (CheckBalance).

Note that the first three variables relate to credit card account transactions, and the last variable relates to the checking account for each customer. We invoke PROC FACTOR, using the principal component method. We are searching for a model that can represent all the variables. The SAS code as well as the results are as follows:

```
proc factor data=Bank method=PRIN;
 var AvgBalance AvgTransValue AvgInt CheckBalance;
run;
```

```
                    The FACTOR Procedure
          Initial Factor Method: Principal Components
              Prior Communality Estimates: ONE

  Eigenvalues of the Correlation Matrix: Total = 4  Average = 1

         Eigenvalue    Difference    Proportion    Cumulative
    1    2.77045385    1.72791808        0.6926        0.6926
    2    1.04253577    0.86144659        0.2606        0.9532
    3    0.18108918    0.17516798        0.0453        0.9985
    4    0.00592120                      0.0015        1.0000

    2 factors will be retained by the MINEIGEN criterion.

                      Factor Pattern

                           Factor1         Factor2
        AvgBalance         0.98266         0.06501
        AvgTransValue      0.90873        -0.23877
        AvgInt             0.98395         0.05059
        CheckBalance       0.10443         0.98931

             Variance Explained by Each Factor

                  Factor1         Factor2
                2.7704539       1.0425358

        Final Communality Estimates: Total = 3.812990
  AvgBalance      AvgTransValue   AvgInt          CheckBalance
  0.96983992      0.88279765      0.97070795      0.98964410
```

Remember that PROC FACTOR had standardized the variables such that the total variance of the four analysis variables is 4.0. The output shows that we may use two principal components to calculate two factors, which could explain a variance of 3.8129. This represents 95.32% of the total variance. This result could be read directly from the cummulative sum of the eigenvalues. Using two eigenvalues of the correlation matrix, we keep 0.9532 of the total variance.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

The communality of each variable is simply the summation of the square of the corresponding common loadings of the variable. For example, the communality of the variable AvgBalance is calculated as

$$(0.98266)^2 + (0.06501)^2 = 0.96984.$$

We now turn our attention to the common loadings. We observe that the loadings of the first factor for the variables AvgBalance, AvgTransValue, and AvgInt are all high (i.e., larger than 0.4). This indicates that the first factor is a summary of the *credit card* transactions, which is a distinct business entity from the second factor, which represents the balance of the checking account.

Let us now experiment with factor rotation. One method of factor rotation is the *varimax* method. In this method, a criterion is maximized (the varimax criterion) such that the absolute value of the loadings in any column is, as much as possible, either high or low. For example, examining the loadings of the second factor in the banking example, we find a high value of 0.98931 and an intermediate value of −0.23877. We attempt rotating the factors to reduce intermediate values, either by increasing the high value or, more realistically in this case, by *spreading* this intermediate value over several other factors. The SAS code in this case is as follows:

```
proc factor data=Bank method=PRIN rotate=varimax;
 var AvgBalance AvgTransValue AvgInt CheckBalance;
run;
```

The result of the code is both the original factor model and the rotated factors with the resulting new factor loadings. The second part with the rotation effects follows.

```
          Orthogonal Transformation Matrix

                1             2
    1     0.99731       0.07337
    2    -0.07337       0.99731

            Rotated Factor Pattern

                          Factor1         Factor2
AvgBalance                0.97524         0.13693
AvgTransValue             0.92380        -0.17145
AvgInt                    0.97758         0.12265
CheckBalance              0.03156         0.99431

       Variance Explained by Each Factor

            Factor1         Factor2
          2.7611530       1.0518366

       Final Communality Estimates: Total = 3.812990

AvgBalance      AvgTransValue   AvgInt          CheckBalance
0.96983992      0.88279765      0.97070795      0.98964410
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

The printout shows that the transformation was successful in increasing the high values and reducing the smaller ones. For example, in the loadings of the second factor, the intermediate value of −0.23877 is replaced with a smaller value of −0.17145. Note that some of the loadings have increased as a result of *spreading* the intermediate values. The factors now present an even more clear picture of the two lines of business the bank is pursuing, namely, checking accounts and credit cards.

Before we turn our attention to the automation of this process to include it as a tool for data reduction, we illustrate the occasional vulnerability of the ML method. When we attempt to solve the same factor model using the maximum likelihood, we implement the option `METHOD=ML` as follows:

```
proc factor data=Bank method=ML rotate=varimax;
 var AvgBalance AvgTransValue AvgInt CheckBalance;
run;
```

The code generates an error, execution stops, and the SAS engine writes the error message to the SAS Log as follows:

```
---- The SAS log ----
NOTE: 1 factor will be retained by the PROPORTION criterion.
ERROR: Communality greater than 1.0.
```

Furthermore, the preceding results in the following code in the SAS output.

```
---- The SAS Output ----

                    The FACTOR Procedure
          Initial Factor Method: Maximum Likelihood

              Prior Communality Estimates: SMC

   AvgBalance      AvgTransValue   AvgInt          CheckBalance
   0.98822342      0.73267380      0.98835067      0.18701597

Preliminary Eigenvalues: Total = 171.726956  Average = 42.931739

      Eigenvalue    Difference    Proportion    Cumulative
 1    171.829290    171.145087        1.0006        1.0006
 2      0.684203      0.971307        0.0040        1.0046
 3     -0.287104      0.212328       -0.0017        1.0029
 4     -0.499432                     -0.0029        1.0000

 1 factor will be retained by the PROPORTION criterion.

Iteration
Criterion   Ridge        Change            Communalities
1           0.1883987    0.0000 0.1675     0.98834 0.68228 0.99964 0.01948
2           0.1883510    0.0000 0.0008     0.98788 0.68299 1.00010 0.01865

        ERROR: Communality greater than 1.0.
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

This introduction barely scratches the surface of the subject of factor analysis. You can consult many books on the subject for a deeper discussion and the various applications of factor analysis. This book focuses on the use of factor analysis in variable reduction. Therefore, we attempt to replace the original set of variables with a smaller set such that the new set would better represent the underlying structure. In this respect, factor analysis is very similar to principal component analysis (PCA). The next section discusses briefly the differences between the two techniques.

16.2 Relationship Between PCA and FA

As discussed, in factor analysis and specifically when using it as a data reduction technique, attention is focused on finding a *few* factors that can represent *most* of the information in the observable variables. In this regard, factor analysis is conceptually similar to principal component analysis. However, they are different in the following aspects.

1. Factor analysis assumes an underlying structure that relates the factors to the observed variables. PCA does not assume any structure.
2. PCA tries to rotate the axis of the *original* variables, using a set of linear transformations, to explain the variance. Factor analysis, on the other hand, creates a new set of variables to explain the *covariances* and correlations between the observed variables.
3. In PCA, when we use the first two principal components and then revise the model by also using the third component, the first two components do not change. This is not the case in factor analysis where, in general, a two-factor model is completely different from a three-factor model.
4. The calculations involved in PCA are straightforward and usually very fast. PCA also converges for all datasets. In factor analysis, there is a variety of methods. Furthermore, the calculations, especially in the case of the ML method, are involved and complicated. In some cases, the ML algorithm may not even converge.

The differences between factor analysis and PCA should highlight the meaning of each of the two techniques. However, in many data mining applications the meaning of the variables is not vital. Sometimes, we are interested only in creating a smaller number of variables that contain most of the information in the data, regardless of what we mean by *information.* In such cases, PCA may be more suitable because it is easier to use and interpret.

16.3 Implementation of Factor Analysis

`PROC FACTOR` and `PROC SCORE` of SAS implement the algorithms needed to perform factor analysis and score new datasets to generate the reduced set of variables (factors

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

in this case). Our SAS implementation will be a set of macros that wrap these two procedures to facilitate their deployment.

PROC FACTOR contains many options for selecting both the estimation method and factor rotation. In our implementation, we have made the following choices.

1. The principal component method will always be used. This is motivated by its speed and the reliability of obtaining results in *all* cases.
2. We will always use factor rotation, with the varimax criterion, in an attempt to make the factors *sharply defined.*
3. No systematic attempt will be made to explain the factor model. If in a particular case we can find a good interpretation for the factors, we treat this event as a bonus, not a necessity.

16.3.1 Obtaining the Factors

The macro implementation using PROC FACTOR is straightforward. We set the option METHOD to the value PRIN to use the principal component method and store the results in the dataset DSFact. The PROption is used to control printing of the result of PROC FACTOR. Setting it to zero suppresses all output.

Table 16.1 Parameters of macro Factor().

Header	Factor(DSin, VarList, NFactors, DSFact, PROption);
Parameter	*Description*
DSin	Input dataset
VarList	List of variables
NFactors	Number of required factors
DSFactors	Dataset containing the results of the factor analysis
PROptions	Printing options (0 = no printing)

The following is the code of the macro, along with how to use it to model the data of the last example Bank (see Table 16.1).

```
%macro Factor(DSin, VarList, NFactors, DSFact, PROption);
PROC FACTOR DATA=&DSin
            Method=PRIN
            Rotate=Varimax
            nfactors=&Nfactors
outstat=&DSFact score
 %if &PROption = 0 %then noprint; ;
 var &varlist;
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

```
run;
%mend;
%let dsin=Bank;
%let varlist=AvgBalance AvgTransValue AvgInt CheckBalance;
%let nfactors=2;
%let DSFact=BankFact;
%let PROption=0;
%Factor(&DSin, &VarList, &NFactors, &DSFact, &PROption);
```

This macro requires the user to know in advance the number of factors to use. Ideally, this number would be obtained through several trials and by examining the meaning of the factors. However, when the number of observed variables is large, one can always use a smaller arbitrary number of factors without investigating the meaning of the factors. In addition, one must focus on the proportion of the explained variance to make sure that the resulting factor model *does* represent the observed variables. In most cases, this approach results in factors that have no clear interpretation, but at least they are a smaller set.

16.3.2 Factor Scores

The scoring macro used to produce the factor scores is a wrapper of `PROC SCORE`. The only requirement is that the dataset contain the factor model. The following is the code of the macro, along with an example on its implementation (see Table 16.2).

```
%macro FactScore(DSin, VarList, DSFact, DSout);
proc score data=&dsin  score=&DSFact out=&DSout;
      var &VarList;
      run;
%mend;

%let DSin=bank;
%let DSFact=BankFact;
%let VarList=AvgBalance AvgTransValue AvgInt CheckBalance;
%let DSout=Bank_Factors;
%FactScore(&DSin, &VarList, &DSFact, &DSout);
```

Table 16.2 Parameters of macro `FactScore()`.

Header	`FactScore(DSin, VarList, DSFact, DSout);`
Parameter	*Description*
`DSin`	Input dataset
`VarList`	List of variables
`DSFact`	Dataset containing the results of `Factor()` macro
`DSout`	Output dataset with the factors added to it

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.

The result of this scoring macro is a dataset that contains the original analysis variables, in addition to the specified number of factors. The factors are called `Factor1`, `Factor2`, and so on. To rename these new variables, a `DATA` step with a `RENAME` option could be used.

The following macro automates this process (see Table 16.3).

```
%macro FactRen(DSin, Nfactors, Prefix, DSout);
Data &DSout;
 SET &DSin (Rename=(
  %do i=1 %to &NFactors;
   Factor&i= &Prefix&i
   %end; ));
run;
%mend;
```

For example, in order to rename the variables `Factor1` and `Factor2` to `F1` and `F2`, respectively, we invoke macro `FactRen()` as follows:

```
%let DSin=Bank_Factors;
%let DSout=Bank_Factors;
%let Nfactors=2;
%let Prefix=F;
%FactRen(&DSin, &Nfactors,&Prefix,&DSout);
```

Table 16.3 Parameters of macro `FactRen()`.

Header	`FactRen(DSin, Nfactors, Prefix, DSout);`
Parameter	*Description*
`DSin`	Input dataset
`Nfactors`	Number of factors to be renamed
`Prefix`	Prefix used in the new variable names
`DSout`	Output dataset with the factors renamed

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-12-10 05:51:59.