CHAPTER 15
PRINCIPAL COMPONENT ANALYSIS

15.1 INTRODUCTION

Principal component analysis (PCA) is one of the oldest and most used methods for the reduction of multidimensional data (Johnson and Wichern 2001). The basic idea of PCA is to find a set of linear transformations of the original variables such that the new set of variables could describe *most* of the variance in a relatively fewer number of variables. The new set of variables is presented, and actually derived, in a decreasing order of contribution. In addition, the first new variable, which we call the first *principal component,* contains the largest proportion of the variance of the original variable set, and the second principal component contains less, and so on.

The usual procedure, then, is to keep only the first few of the principal components, which contain, say, 95% or 99% of the variance of the original set of variables. PCA is particularly useful when

- There are too many (independent) variables
- The independent variables show a high correlation between them

As we will see later in this chapter, there are two main methods for performing PCA of a set of data. The first one involves working with the variance–covariance matrix. However, the values included in this matrix depend on the units and magnitude of each variable. Therefore, a variable representing a customer's balance will be in the range of, say, \$0 to \$100,000.00, and the age field will be in the range of 0 to 100. To normalize all the variables, the matrix representing the correlations, R, is sometimes used instead to calculate the principal components. There is a live and active debate in the literature about the advantages and disadvantages of these two approaches. In this chapter, we present the implementation of both methods.

The final output of PCA is the new set of variables, principal components, which represents the original dataset. The user then would normally use only the first few of these new variables because they contain most of the information of the original dataset.

In the following sections, we first present the theoretical background of PCA. However, if you are not interested in the theory, skip Section 15.2 and proceed directly to Section 15.3, which provides the macro for SAS implementation of PCA with an example. Finally, we present a modified macro that selects the most contributing variables containing the required percentage of the variance.

The final section in this chapter discusses some of the issues that frequently arise while using PCA.

15.2 Mathematical Formulations

Let us start with a set of p variables, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p$, and assume that we have n observations of these variables. The mean vector μ is the vector whose p components are defined as

$$\mu_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij}, \qquad i = 1, \cdots, p. \tag{15.1}$$

The unbiased $p \times p$ variance–covariance matrix (simply the covariance matrix) of this sample is defined as

$$S = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - \mu)(x_j - \mu)'. \tag{15.2}$$

Finally, the $p \times p$ correlation matrix R of this sample is defined as

$$R = D^{-1/2} S D^{1/2}, \tag{15.3}$$

where the matrix $D^{1/2}$ is the *sample standard deviation matrix,* which is calculated from the covariance S as the square root of its diagonal elements or

$$D^{1/2} = \begin{bmatrix} \sqrt{s_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{s_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{s_{pp}} \end{bmatrix}, \tag{15.4}$$

while the matrix $D^{-1/2}$ is the inverse of $D^{1/2}$, or

$$D^{-1/2} = \begin{bmatrix} \frac{1}{\sqrt{s_{11}}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{s_{22}}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sqrt{s_{pp}}} \end{bmatrix}. \tag{15.5}$$

Let the p-pairs $(\lambda_1, v_1), (\lambda_2, v_2), \cdots, (\lambda_p, v_p)$, be the eigenvalue–eigenvectors of the covariance matrix S, with the eigenvalues arranged in descending order, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.

It can be demonstrated that the ith sample principal component is given by

$$y_i = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \end{bmatrix} v_i, \qquad i = 1, \cdots, p. \tag{15.6}$$

The variance of the variable y_i is equal to its corresponding eigenvalue λ_i. Therefore, the total sample variance is the sum of all the eigenvalues:

$$Tr(S) = \sum_{i=1}^{p} S_{ii} = \sum_{i=1}^{p} \lambda_i. \tag{15.7}$$

To obtain the principal components using the correlation matrix R, which is also called the *standardized observations covariance matrix,* we replace S with R in the preceding equations and denote the resulting principal components z_i, $i = 1, \cdots, p$.

In this case, the eigenvalue–eigenvector pairs of the matrix R are $(\theta_1, w_1), \cdots, (\theta_p, w_p)$, with $\theta_1 \geq \cdots \geq \theta_p \geq 0$ and the principal components of the standardized variables are given by

$$z_i = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \end{bmatrix} w_i, \qquad i = 1, \cdots, p. \tag{15.8}$$

Similarly, the variance of the variable z_i is equal to its corresponding eigenvalue θ_i, and the total sample variance of the standardized variables is the sum of all the eigenvalues of R:

$$Tr(R) = \sum_{i=1}^{p} R_{ii} = \sum_{i=1}^{p} \theta_i. \tag{15.9}$$

15.3 Implementing and Using PCA

We may summarize the mathematical formulations in the previous section by stating that PCA works by finding the eigenvalues of the covariance matrix and using the eigenvectors as the linear transformations to obtain the principal components. The importance of each principal component is determined by the relative magnitude

of its eigenvalue. Therefore, we call the principal component corresponding to the highest eigenvalue *the first principal component*; the second highest, the second principal component, and so on.

In addition to the relative ordering of the principal components, an important property of PCA is that the sum of the total variance in the original variables is equal to the sum of the eigenvalues of the covariance matrix.

Before we proceed, let us demonstrate these principles with an example.

Table 15.1 shows 20 records of credit card customers of a bank. The variables represent the average monthly balance, the average transaction value, the average monthly interest paid, and the average balance on the customer's checking account. All the variables are related to the credit card business of the bank's clients (note that

Table 15.1 Data of 20 credit card customers.

Customer ID	*Average credit card balance*	*Average transaction value*	*Average interest paid*	*Checking account balance*
1	338.55	102.66	17.9	180.00
2	149.39	30.55	8.9	210.92
3	135.47	39.33	7.4	232.76
4	26.78	7.13	1.5	200.00
5	184.91	44.21	9.9	461.13
6	333.97	106.35	19.3	263.83
7	464.49	77.14	24.0	501.01
8	26.88	6.60	1.5	439.64
9	458.13	72.39	25.6	449.92
10	395.32	108.18	22.6	188.54
11	257.60	38.24	15.0	496.47
12	98.34	15.26	5.2	463.50
13	244.86	41.45	12.8	441.58
14	388.85	55.93	20.2	429.51
15	401.28	117.87	23.3	538.55
16	426.62	65.65	24.5	250.42
17	420.27	113.09	22.1	348.48
18	247.72	38.04	13.2	469.68
19	392.29	72.17	23.2	474.02
20	210.17	49.81	11.3	381.06

the balance of the checking account is relevant because it can be assumed that the bank's clients use their checking accounts to pay credit card bills). Therefore, we may attempt to find a transformation to reduce these four variables into a smaller set while keeping all or most of the information in the data.

PROC PRINCOMP of SAS/STAT performs principal component analysis. The following listing shows its use with the data in Table 15.1.

```
PROC PRINCOMP DATA=CC COV;
  VAR  AvgBalance   AvgTransValue   AvgInt  CheckBalance;;
RUN;
```

Invoking PROC PRINCOMP will result in the calculation of the covariance matrix, the eigenvalues, and the eigenvectors. The following is the listing of the SAS output of the preceding code.

```
                The PRINCOMP Procedure

                    Observations          20
                    Variables              4

                     Simple Statistics

           AvgBalance  AvgTransValue     AvgInt   CheckBalance

Mean       280.0945000   60.10250000  15.47000000  371.0510000
StD        141.7152098   35.31563779   7.88309984  123.0054403

                       Covariance Matrix

               AvgBalance AvgTransValue   AvgInt CheckBalance

AvgBalance    20083.20068  4108.87486  1110.41977   2616.63039
AvgTransValue  4108.87486  1247.19427   230.09403   -462.56824
AvgInt         1110.41977   230.09403    62.14326    132.19482
CheckBalance   2616.63039  -462.56824   132.19482  15130.33835

            Total Variance    36522.876561

         Eigenvalues of the Covariance Matrix

     Eigenvalue    Difference    Proportion    Cumulative

1    21908.4417     7617.4826        0.5999        0.5999
2    14290.9591    13968.2050        0.3913        0.9911
3      322.7540      322.0323        0.0088        1.0000
4        0.7218                      0.0000        1.0000
```

```
                    Eigenvectors

                  Prin1      Prin2     Prin3      Prin4

AvgBalance      0.920774   -.323359  -.211416   -.054024
AvgTransValue   0.175967   -.135396  0.975021   -.006076
AvgInt          0.050739   -.018725  -.005535   0.998521
CheckBalance    0.344438   0.936353  0.067867   0.000433
```

The results show that the first two eigenvalues represent 99.11% of the variance of the original four variables. Therefore, by using the first and second principal components, it is possible to keep 99% of the information contained in the four variables.

Furthermore, the parameters of the equations defining the new variables, denoted `Prin1` and `Prin2`, are given by components of the eigenvectors for the first and second principal components. Thus, we can define the new variables `Prin1` and `Prin2` as

$$\begin{aligned} \texttt{Prin1} &= 0.920774\,(\texttt{AvgBalance}) + 0.175967\,(\texttt{AvgTransValue}) \\ &\quad + 0.050739\,(\texttt{AvgInt}) + 0.344438\,(\texttt{CheckBalance}), \\ \texttt{Prin2} &= -.323359\,(\texttt{AvgBalance}) - .135396\,(\texttt{AvgTransValue}) \\ &\quad - .018725\,(\texttt{AvgInt}) + 0.936353\,(\texttt{CheckBalance}). \end{aligned} \tag{15.10}$$

`PROC SCORE` of SAS/STAT allows the automatic substitution in the last two equations, provided that we store the values of the eigenvectors in a dataset and instruct `PROC PRINCOMP` to compute only the first two eigenvalues and eigenvectors. However, before performing a full analysis of the covariance matrix of the dataset, we would not have known that we needed only two principal components to keep more than 99% of the data variance. Therefore, in our SAS macro implementation of PCA, we adopt a two-step approach.

First, we invoke `PROC PRINCOMP` to analyze the full covariance matrix and obtain all the eigenvalues. Then we calculate the number of princpial components needed to preserve a certain percentage of the variance. The second step follows by using only the identified number of principal components to generate the new variables. The new variables will be denoted `Prin1, Prin2,` and so on. The following macro implements this strategy (see Table 15.2).

Step 1

First run `PRINCOMP` to calculate all the eigenvalues.

```
proc princomp data=&DSin &Method outstat=&DSEigen noprint;
var &VarList;
run;
```

Step 2

Select the top $P\%$ of the summation of the eigenvalues.

Table 15.2 Parameters of macro `PrinComp2()`.

Header	`PrinComp2(DSin, VarList, Method, P, DSEigen, DSout);`
Parameter	*Description*
`DSin`	Input dataset
`VarList`	List of variables
`Method`	PCA method (COV or empty)
`P`	Percentage of total variance to keep
`DSEigen`	Output dataset to store the eigenvectors
`DSout`	Output dataset with the principal components added

```
data Tempcov1;
 set &DSEigen;
 if _Type_ ne 'EIGENVAL' then delete;
 drop _NAME_;
 run;
proc transpose data=Tempcov1 out=TempCovT;
run;

data TempCov2;
 set TempCovT;
 retain SumEigen 0;
 SumEigen=SumEigen+COl1;
run;

proc sql noprint;
select max(SumEigen) into :SEigen from TempCov2;
quit;

data TempCov3;
 set TempCov2;
IEigen=_N_;
PEigen = SumEigen/&SEigen;
run;


/* We now count the number of eigenvalues needed to
 reach P_Percent */

proc sql noprint;
 select count(*) into :Nh from Tempcov3 where PEigen >= &P;
 select count(*) into :NN from TempCov3;
%let N=%eval(&NN-&Nh+1);
quit;
```

Step 3
Keep only the selected set of eigenvalues and their equivalent eigenvectors. Use this reduced set for generation of new variables.

```
/* Delete from the DSEigen all the rows above
 the needed N eigenvectors */
data TempCov4;
 set &DSEigen;
run;
proc sql noprint;
%do i=%eval(&N+1) %to &NN;
 delete from TempCov4 where _NAME_ = "Prin&i";
%end;
quit;


/* And score */
proc score data=&Dsin Score=TempCov4 Out=&DSout;
Var &VarList;
run;
```

Step 4
Finally, clean the workspace and finish the macro.

```
proc datasets library=work nodetails;
delete Tempcov1 Tempcov2 Tempcov3 Tempcov4 Tempcovt;
run;
quit;

%mend;
```

15.4 Comments on Using PCA

15.4.1 Number of Principal Components

In the previous section, we presented the macro `PrinComp2()`, which allows the automatic selection of the top principal components by specifying the percentage of the variance that needs to be preserved. In most practical applications, it is sufficient to keep between 80% and 95% of the variance.

15.4.2 Success of PCA

Sometimes PCA may not be a suitable tool for the reduction of the given dataset or it shows that the variables cannot be reduced, at least not by using PCA. A characteristic

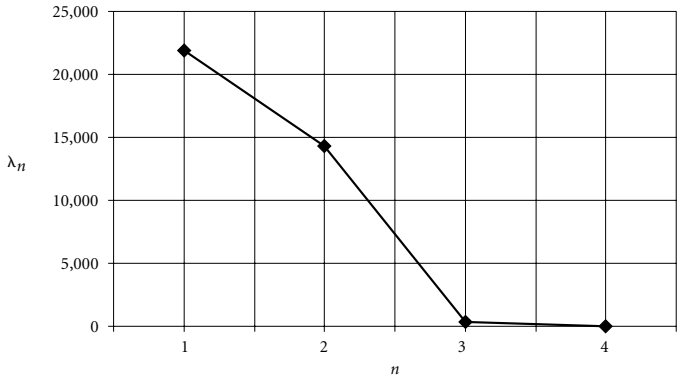

Figure 15.1 Scree plot of the credit card data.

feature of these cases is that the decline in the magnitude of the eigenvalues is very slow. This leads to taking a larger number of principal components and, therefore, achieving insignificant reduction of the number of variables.

Statisticians use a plot called *scree plot,* which displays the magnitude of the eigenvalues on a line chart, to show the rate of decay of the eigenvalues. The scree plot of the eigenvalues of the last example is given in Figure 15.1. The figure shows that the slope representing the change in the magnitude of the eigenvalues changes from very steep in the first two eigenvalues to very shallow in the third. It also shows that the fourth eigenvalue is almost zero.

Using the scree plot, it is easy to see that we need only two eigenvalues to preserve most of the variance. However, Figure 15.2 is the scree plot for a different dataset, where the magnitude of the eigenvalues is also decreasing, but very slowly. In this

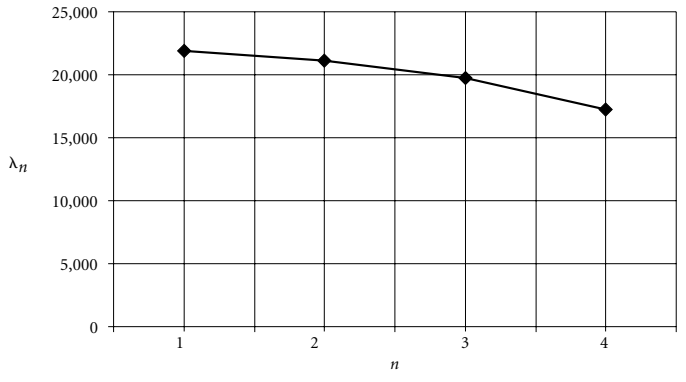

Figure 15.2 Scree plot of uncorrelated data.

case, we conclude that PCA did not find a simple reduced representation of the data and that the variables are not significantly correlated.

15.4.3 Nominal Variables

As discussed in Section 9.4, nominal variables can be mapped into indicator (dummy) variables. These indicator variables can be used as ordinary variables in PCA. Although it is possible to perform PCA using this approach, it is not recommended because of the scant information available about the behavior of indicator variables in PCA. Only a limited number of studies have explored this approach. Therefore, we recommend using other techniques specific to the analysis of categorical variables.

15.4.4 Dataset Size and Performance

PCA is based on finding the eigenvalues of the covariance matrix. Therefore, the required computational resources depend on the number of variables more than the number of records. Therefore, performing PCA with large datasets should not pose a significant problem using SAS. The only real issue arises when the number of variables included in the analysis becomes large, say more than 50. In this case, the numerical algorithms for finding the eigenvalues and eigenvectors may themselves be pushed to the limit and provide unreliable results. Therefore, we recommend that PCA be used only with a reasonable number of variables (< 20 or so), which are believed to be correlated and could be reduced into a smaller core set. This should be based on the understanding of the data at hand.