Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

CHAPTER 5

Basic Use of Principal Components

OBJECTIVE

After learning how to compute principal components in chapter 3, and how to visualize key PCA results in chapter 4, this chapter will show you some basic use of principal components. More applications of principal components are discussed in chapter 6. In this chapter, you will learn to determine the minimum number of principal components that will define the few dimensions in which your data will be analyzed. You will also learn the important technique data reconstruction and some of its applications.

Contents

https://agreestat.com/books/pca4r/ https://agreestat.com/books/

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

5.1 Introduction

This chapter reviews various applications of principal components. First, you will see how to determine the correct number of principal components to retain for your analysis. I will show you how to perform PCA on a very large dataset, a problem that is common in this era of big data. An analysis performed on PC scores must still be interpreted with respect the units in which the original data is expressed. You will also learn in this chapter, how to reconstruct original data using a small number of principal components.

After you compute the principal components, you need to decide how many of them you need to retain in your analysis. A trade-off must be made between 2 conflicting goals. On one hand, you need to reduce the number of synthetic variables (i.e. the number of dimensions) to simplify your analysis. On the other hand, you want to explain most of the variation in your dataset, which requires adding more synthetic variables to your analysis. This is a case of simplicity versus accuracy. I will address this issue in section 5.2 with a discussion of Kaiser's rule, Jolliffe's rule as well as the Elbow method.

In section 5.3, you will learn how to predict PC scores using a PCA output obtained from another and smaller dataset. This technique could prove useful when analyzing very large datasets, which might consume substantial computing resources. You will split the large dataset into a smaller training dataset and a larger testing dataset. PCA will be performed on the training dataset, and the results used to predict scores in the testing dataset. Section 5.4 on the other hand, will address the important problem of reconstructing the original dataset using a small number of selected synthetic variables. This technique has a useful application in anomaly detection that will also be discussed.

5.2 Optimal Number of Principal Components

Consider the "Motor Trend Car Road Tests" datasets, which comes with the R software under the name `mtcars`. It comprises fuel consumption and 10 design and performance characteristics of 32 automobiles. Two of the variables in this dataset named `vs` (Engine: 0 = "V-shaped", 1 = "straight"), and `am` (Transmission: 0 = "automatic", 1 = "manual") are categorical and will be excluded from the Principal Component Analysis (PCA)[1].

[1]Remember that PCA exclusively deals with quantitative measurements. Therefore, all categorical variables must first be removed from your dataset before it can be analyzed.

https://agreestat.com/books/pca4r/ https://agreestat.com/books/

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

Table 5.1 shows an output extract of the PCA analysis that I performed on the `mtcars` dataset by running the R script of Program 5.1. You may want to execute this program yourself to see the output in its entirety. The standard deviations of the 9 synthetic variables are contained in the `$sdev` list element, whereas the 9 series of PC coefficients `PC1` $\cdots$ `PC9` are contained in the `$rotation` list element in the form of a 9×9 matrix. Remember that principal components are vectors that constitute a new coordinate system into which the original data points will be mapped. The numbers you see in the `$rotation` list component are the coordinates of the PCs in the Cartesian coordinate system.

Table 5.1: Summary of the PCA performed on the `mtcars` dataset

```
$sdev
[1] 2.378 1.443 0.710 0.515 0.428 0.352 0.324 0.242 0.149

$rotation
        PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9
mpg  -0.393  0.028 -0.221 -0.006 -0.321  0.720 -0.381 -0.125  0.115
cyl   0.403  0.016 -0.252  0.041  0.117  0.224 -0.159  0.810  0.163
disp  0.397 -0.089 -0.078  0.339 -0.487 -0.020 -0.182 -0.064 -0.662
hp    0.367  0.269 -0.017  0.068 -0.295  0.354  0.696 -0.166  0.252
drat -0.312  0.342  0.150  0.846  0.162 -0.015  0.048  0.135  0.038
wt    0.373 -0.172  0.454  0.191 -0.187 -0.084 -0.428 -0.198  0.569
qsec -0.224 -0.484  0.628 -0.030 -0.148  0.258  0.276  0.356 -0.169
gear -0.209  0.551  0.207 -0.282 -0.562 -0.323 -0.086  0.316  0.047
carb  0.245  0.484  0.464 -0.214  0.400  0.357 -0.206 -0.108 -0.320
```

Program 5.1. Performing the principal component analysis on the R built-in `mtcars` dataset

```

#--  PCA of mtcars dataset --

pacman::p_load(factoextra,tidyverse,xlsx)
outdir <- "C:/Users/.../Books/PCA-Using-R/chap5/"
mtcars.df <- select(mtcars,-c(vs,am))

```

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

```
#-- Performing the PCA
mtcars.pca <- prcomp(mtcars.df, center=TRUE, scale=TRUE)
mtcars.out <- lapply(mtcars.pca,round,3)
print(mtcars.out)
```
End of Script

Once the PCA output is obtained, you need to decide how many principal components you are going to retain in your analysis. The number of principal components is expected to be small as it represents the smaller number of dimensions along which your analysis will be conducted.

You learned in chapter 3 that although the number of principal components equals the number of original variables, only a small number of these components are retained for analysis. A key question to be asked is "what is the right number of principal components that you should retain in your analysis?" The smaller that number, the better as the primary goal of PCA is dimensionality reduction.

Most methods proposed in the literature for determining the optimal number of principal components are criticized for being heuristic and subjective, although a few of them have a solid statistical base. It is widely acknowledged that the analyst's judgement is still necessary to make the final determination. Ferré (1995) for example, admitted that there was no ideal solution that would meet all research goals, while Jolliffe (2002) indicated that statistically-based procedures do not appear to offer clear advantages over simpler procedures. Nevertheless, readers interested in an elaborate comparative analysis of various methods can find interesting material in Zwick and Velicer (1986) or in Jackson (1993). In this section, I am going to review the following 3 of the most-widely used methods in practice:

- **Kaiser's rule.** The Kaiser's rule discussed in Kaiser (1960) stipulates that you should retain a principal component only if the variance of the associated synthetic variable exceeds 1. The rationale behind this rule is that synthetic variables with a variance smaller than 1 contains less information than the original standardized variables. So, why keep it?

- **Jolliffe's Rule.** A rule closely related to that of Kaiser, and due to Jolliffe (2002) stipulates that the appropriate number of principal components to retain corresponds to the number of eigenvalues that are larger than 0.7. An eigenvalue equals the variance of the associated synthetic variable.

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

According to Jolliffe (2002), the threshold of 0.7 is motivated by the need to account for the effect of sample variance. That is, an eigenvalue may well exceed 1 even though its sample-based estimated value is as low as 0.7 due to sampling errors.

- **Elbow Method.** Another PC selection criterion proposed by Cattell (1966), also known as the "Elbow Method" is based on a visual examination of the Scree[2] graph shown of Figure 5.1. Cattell (1966) recommends to look for the point on the graph beyond which the scree graph becomes a more-or-less straight line, not necessarily horizontal. All PCs up to that turning point should be retained, and the remaining can be neglected.

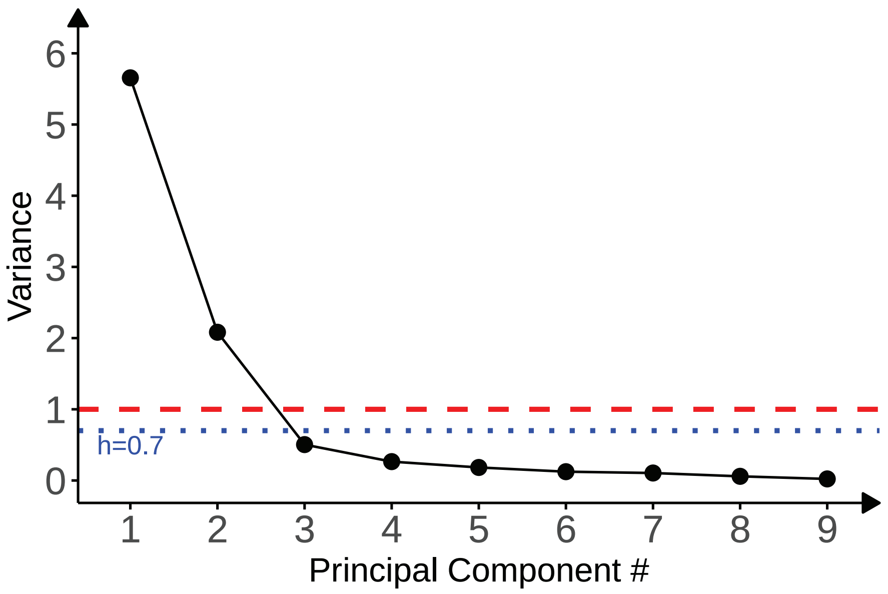

Figure 5.1: Scree plot showing the 9 synthetic variables of the `MTcars` dataset and their variances

Table 5.2 shows all 9 principal components, their standard errors (`SE`), variance (`Var`), as well as the associated percent of total variance explained individually and cumulatively. The variance of the first PC is 5.656 and accounts for 62.8%

[2]Jolliffe (2002) indicates that the name scree graph stems "...from the similarity of its typical shape to that of the accumulation of loose rubble, or scree, at the foot of a mountain slope."

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

of total variance. The second PC has a variance of 2.082, which accounts for 23.1% of total variance. Together, the 2 PCs account for 86% of total variance cumulatively.

The scree plot of Figure 5.1 was created with the script file of Program 5.2. This script is similar to the many scripts discussed in chapter 4 and will not be further discussed here. However, you can download this script using the link `https://bit.ly/3uoZROM`.

Table 5.2: Percent of total variance explained by each principal component, for the `Motor Trend Car Road Tests`[a] dataset

PC#	SE	Var	%Tot.var	Cumulative%
1	2.3782	5.656	62.8	62.8
2	1.4429	2.082	23.1	86.0
3	0.7101	0.504	5.6	91.6
4	0.5148	0.265	2.9	94.5
5	0.4280	0.183	2.0	96.6
6	0.3518	0.124	1.4	97.9
7	0.3241	0.105	1.2	99.1
8	0.2419	0.059	0.7	99.8
9	0.1490	0.022	0.2	100.0
Total		9	100.0	-

[a]This dataset comes with R, and is named `mtcars`.

Program 5.2. R script for creating the scree plot of Figure 5.1

```

#--  PCA of mtcars dataset --

pacman::p_load(factoextra,tidyverse,xlsx)
outdir <- "C:/Users/.../Books/PCA-Using-R/chap5/"
mtcars.df <- select(mtcars,-c(vs,am))

#-- Performing the PCA

mtcars.pca <- prcomp(mtcars.df, center=TRUE, scale=TRUE)
```

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

```
summary(mtcars.pca)

#-- Line Screeplot based on variance absolute values

splot.line <- fviz_eig(mtcars.pca,geom="line", main = "",
                        choice = "eigenvalue",
                        ggtheme = theme_classic(),
                        xlab = "Principal Component #") +
   scale_y_continuous(limits = c(0,6.3), name="Variance",
                      breaks = seq(0,6,1)) +
   geom_point(shape=16,size=3) +
   theme(text=element_text(size=14),
         axis.text = element_text(size=16),
         axis.line = element_line(
           arrow=arrow(type="closed",length=unit(8,'pt')))) +
   geom_hline(yintercept=1, linetype="dashed",
              color = "red", size=1) +
   geom_hline(yintercept=0.7, linetype="dotted",
              color = "blue", size=1) +
   annotate("text", x=1.0, y=0.5, label="h=0.7",color="blue")
 ggsave(filename = paste0(outdir,"screeplot-mtcars.pdf"),
       width=5.2,height=3.8,units="in")
```

End of Script

It follows from Figure 5.1 that only the first 2 principal components have each a variance that exceeds 1. Therefore, according to Kaiser's rule, subsequent analyses should be based on the first 2 synthetic variables. This recommendation will narrow the original multidimensional data analysis problem down to 2 dimensions, which capture 86% of the total data variation as seen in the rightmost column of Table 5.2.

Jolliffe's rule also recommends retaining only the first 2 principal components in subsequent analyses. In fact, it follows from the third column of Table 5.2 that only `PC1` and `PC2` have variances that exceed Jolliffe's cutoff point of 0.7.

As for the Elbow Method, Jolliffe (2002) stated "The first point on the straight line is then taken to be the last factor/component to be retained." Therefore, the PC number associated with the graph 'elbow', should be the number of components retained in subsequent analyses. When applied to Figure 5.1, this rule recommend to retain the first 3 PCs. After the third PC, the scree

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

graph becomes more or less flat.

In my views, there is no need to have an objective general-purpose method for determining the optimal number of principal components. In reality, the number of principal components to retain ultimately depends on what you want to do with those selected. Some researchers will focus solely on the first principal component because they are looking for a single composite score that summarizes the entire dataset. Others may be interested in detecting outlying observations, in which case using 2 principal components could be more appropriate (more on this in chapter 6). If PC scores are to be used as independent variables in a regression model then you may want to use a sufficiently large number of principal components to ensure all initial variables are adequately represented. Some of these topics are further discussed in chapter 6.

5.3 Predicting Scores for New Data

Performing Principal Component Analysis (PCA) is computationally intensive. Therefore, if your dataset contains several hundreds of thousands of records, it would be unwise to run the PCA on the entire file, as the execution could take a long time to its completion and will consume a lot of resources. The solution to this problem is to split the initial large dataset into 2 data subsets. The first (and often smaller) data-subset known as the training dataset is used as input to the PCA. The outcome of this first analysis is then used to "predict" the synthetic variables for the second and often larger data-subset known as the "Testing dataset."

Using a concrete dataset, I am going to show you how the testing dataset PC scores can be approximated without any need to run PCA again. Consider one more time the R built-in `iris` dataset that I initially discussed in section 4.5 of chapter 4. This dataset contains 150 records of flowers of different types. I selected a random sample of 97 records (65% of the 150 records) to create my training dataset. The remaining 53 records represent the test dataset. That is, the PCA produces synthetic variables for all 97 records in the training dataset. Then I will "predict" the values of these synthetic variables for the remaining 53 records without performing PCA again on the testing dataset.

Program 5.3 shows an R script that takes the `iris` dataset, renames its columns to shorten and lowercase their names (c.f. lines `#07, #08, and #09`) for easy reference[3]. In line `#10`, I remove the categorical `species` variable from

[3]The setnames() function is available in the `data.table` package and is used to name the

Get the entire ebook for $9.99 using the link: https://sites.fastspring.com/agreestat/instant/pca4rstudio/

the dataset, since PCA only takes quantitative measurements. Finally, in lines #14-#17, I split the `iris` dataset, by assigning 65% randomly-selected records (i.e. 97 records) to the training dataset named `iris.train` and the remaining 53 records to the testing dataset `iris.test`.

My objective is to perform the PCA on `iris.train`, and use the results to predict PC scores for the 53 records in the testing dataset `iris.test`.

Program 5.3. Creating a training and a testing datasets from `iris`

```
01  #
02  #-- Predicting PC scores for a testing data subset of iris --
03  #
04  pacman::p_load(data.table,tidyverse,xlsx)
05  outdir <- "C:/Users/.../Documents/Books/PCA-Using-R/chap5/"
06    #- Rename iris columns with shorter lower-cased names
07  col.names <- c("sepal.l","sepal.w","petal.l",
08                 "petal.w","species")
09  iris.df <- setnames(iris,old=colnames(iris), new=col.names)
10  iris.num.df <- select(iris.df,-species) #remove 'species' col
11
12  #-- split iris into training (65%) and testing (35%) datasets
13
14  set.seed(1)
15  samp <- sample(nrow(iris.num.df), nrow(iris.num.df)*0.65)
16  iris.train <- iris.num.df[samp,]
17  iris.test <- iris.num.df[-samp,]
```

End of Script

After running PCA on the `iris.train` dataset as shown in line #21 of Program 5.4, I obtained the summary results shown in Table 5.4. The standard deviations of the 4 synthetic variables are given by: `1.711, 0.961, 0.355, 0.151.` The sub-table entitled "Rotation (n x k) = (4 x 4):" is a 4×4 matrix of principal components. `PC1-PC4` are the 4 vectors that form the new coordinate system of principal components, to which original data are mapped, to obtain PC scores. That is, each column of this matrix contains Principal Component coefficients, which are the coordinates of vectors `PC1-PC4` in the original Cartesian coordinate system.

columns in a data frame.