Pincipal Component Analysis

Johnny Lo

Feb 2023

Contents

1 Principal Component Analysis

PCA is an unsupervised technique that takes high dimensional data and represent them in lower dimension by exploiting the relationships between the variables, without too much loss of information. PCA is the most popular dimension reduction technique. It also serves as an intermediate step to other analyses such as factor analysis and principal component regression.

To start, you need to install and load the relevant packages for this workshop.

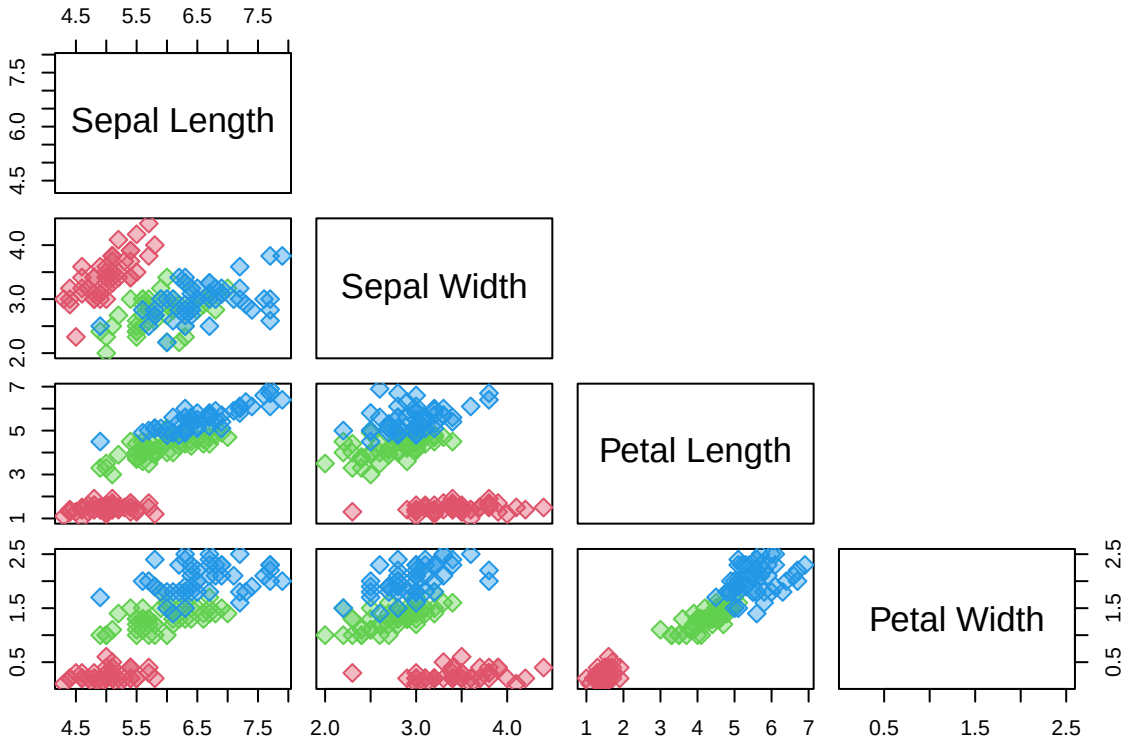
```
#De-comment to install the packages below
# install.packages(c("vegan","tidyverse","timeDate","scales","scatterplot3d",
#                    "mlbench","ggpubr","factoextra"))
library(vegan) #PCoA and distance/similarity matrices
library(scatterplot3d) #3-D plot
library(mlbench)  #Cancer dataset
library(scales) #For more graphic scales options
library(tidyverse)
library(ggpubr)
library(factoextra)
```

To illustrate the PCA process in ***R***, we will use the iris data.

```
data(iris)  #Load the dataset
#Rename the Species
iris$Species <- factor(iris$Species,
                       labels=c("Setosa","Versicolor","Virginica"))
```

First we create a scatter plot matrix showing the bivariate relationship between the four features of the iris flowers.

```
#Scatter plot matrix
pairs(iris[,1:4],
      pch=23, #Shape of the points
      #Colour of the outline of the points
      col=as.numeric(iris$Species)+1,
      #Fill colour of the points with reduced colour intensity
      bg=alpha(as.numeric(iris$Species)+1,0.4),
      cex=1.5,  #Size of the points
      upper.panel=NULL,  #Do not display the the upper panel
      #Substitution the full stop in the feature names with a space
      labels=gsub("[[:punct:]]"," ",colnames(iris[,1:4])))
```



Exercise: Based on the scatter plot matrix, how do each species compare to each other in terms of these features?

Enter text here

Now, we will perform PCA on the iris dataset using the function *prcomp(.)*. This function performs PCA by single value decomposition(SVD). Note that the raw/unstandardised data is used by default. To perform PCA with standardised data instead, set the argument **scale=TRUE**, which is what we will do here. Note that the PCA of the iris dataset in the lecture slides are based on the unstandardised data.

```
#PCA
pca.iris <- prcomp(iris[,1:4], #iris dataset excluding the Species column
                   scale=TRUE) #Standardised the data
```

```
#Show the individual and cumulative proportion of variance explained.
summary(pca.iris)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Exercise: What proportion of variance is explained by the first **two** components?

Enter text here

We can also output the principal component coefficients (often referred to as loadings).

```
pca.iris$rotation
```

```
##                     PC1         PC2        PC3        PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

Note that because data were standardised in the PCA process, we can interpret the coefficient directly without having to compute the correlations between each of the principal component with each of the features.

Exercise: Are PC1 and PC2 interpretable? If so, what would say about them?

Enter text here

Scree plot is a useful way to determine how many principal components one should retain. In other words, the lowest number of dimensions we can have without too much loss of information.

```
plot(pca.iris, type="l", main="Scree plot - Iris")
```

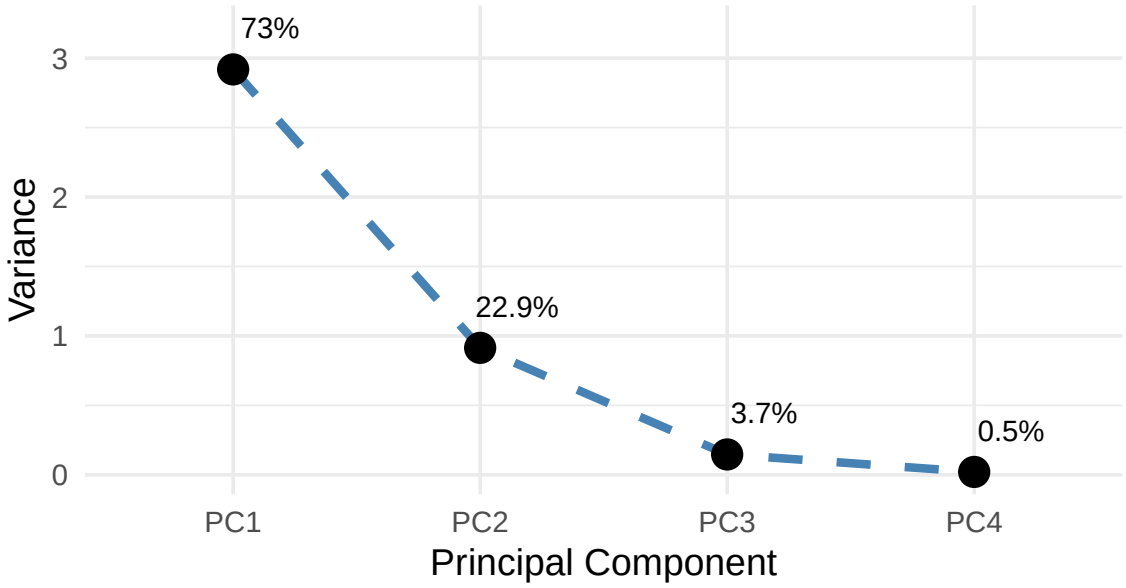
You can also create your own scree plot with *ggplot(.)*.

```
#Extract the eigenvalues and proportions explained
varexp.iris <- summary(pca.iris)$importance; varexp.iris
```

```
##                             PC1       PC2       PC3       PC4
## Standard deviation     1.708361 0.9560494 0.3830886 0.1439265
## Proportion of Variance 0.729620 0.2285100 0.0366900 0.0051800
## Cumulative Proportion  0.729620 0.9581300 0.9948200 1.0000000
```

```
#Create the data frame for plotting
df <- data.frame(Variance=varexp.iris[1,]^2,PC=1:length(pca.iris$sdev));

ggplot(df,aes(PC,Variance))+
  geom_line(colour="steelblue",size=1.5,linetype=2)+
  geom_point(size=5)+
  theme_minimal(base_size=14)+
  xlab("Principal Component")+
  ylab("Variance")+
  scale_x_discrete(limits=paste("PC",1:length(pca.iris$sdev),sep=""))+
  annotate("text",x=c(1:4)+0.15,y=varexp.iris[1,]^2+0.3,
           label=paste(round(varexp.iris[2,]*100,1),"%",sep=""))
```



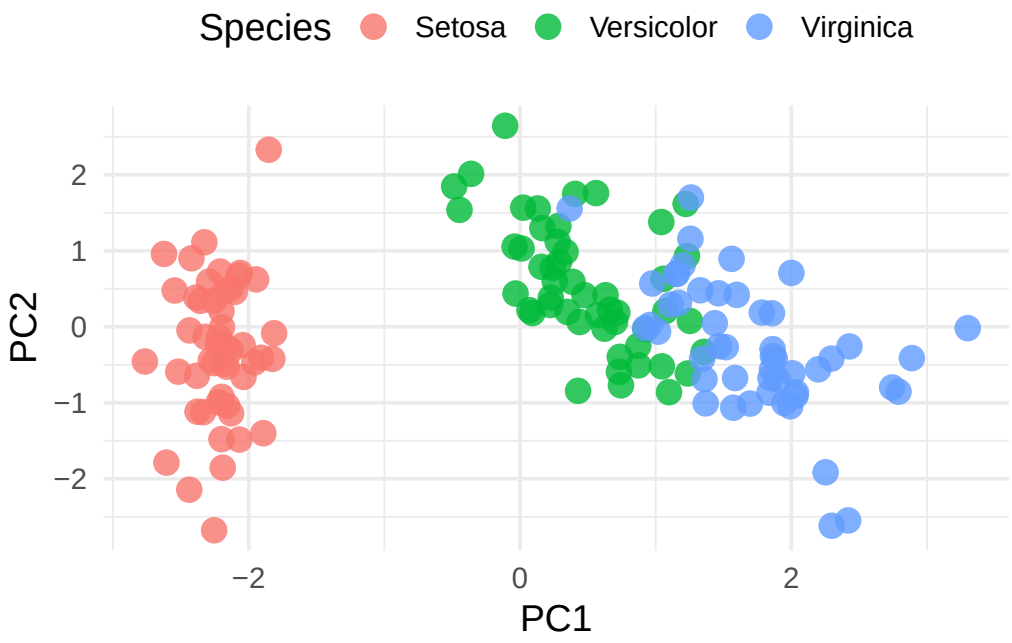
Exercise: How many components should we retain in this case?

Enter text here

Now, we can visualise our data in the first two dimensions, i.e. PC1 and PC2.

```
df <- data.frame(pca.iris$x,  #PCA scores
                 Species=iris$Species);

ggplot(df,aes(x=PC1,y=PC2))+
       geom_point(aes(colour=Species),alpha=0.8,size=4)+
       theme_minimal(base_size=14)+
       theme(legend.position = "top")+
       xlab("PC1")+
       ylab("PC2");
```



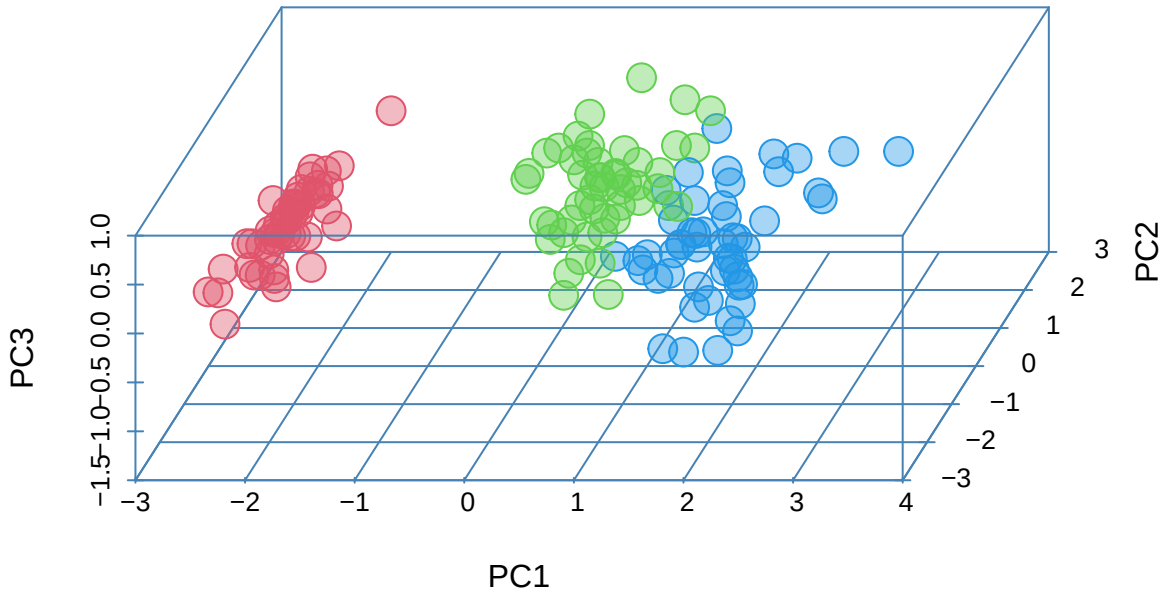
Try visualising the data with PC1 vs PC3, i.e. set the argument **y** in the *aes(.)* function to PC3.

Exercise: Can PC1 or PC2 be used to classify the species? If so, how?

Exercise: Suppose you found an iris with sepal length = 6.35 cm, sepal width = 2.65, petal length = 4.05 cm and petal width = 1.46 cm. Which species does it likely belong to?
Hint: Compute the corresponding scores for PC1 and PC2 and see where it lies in the PC plot.

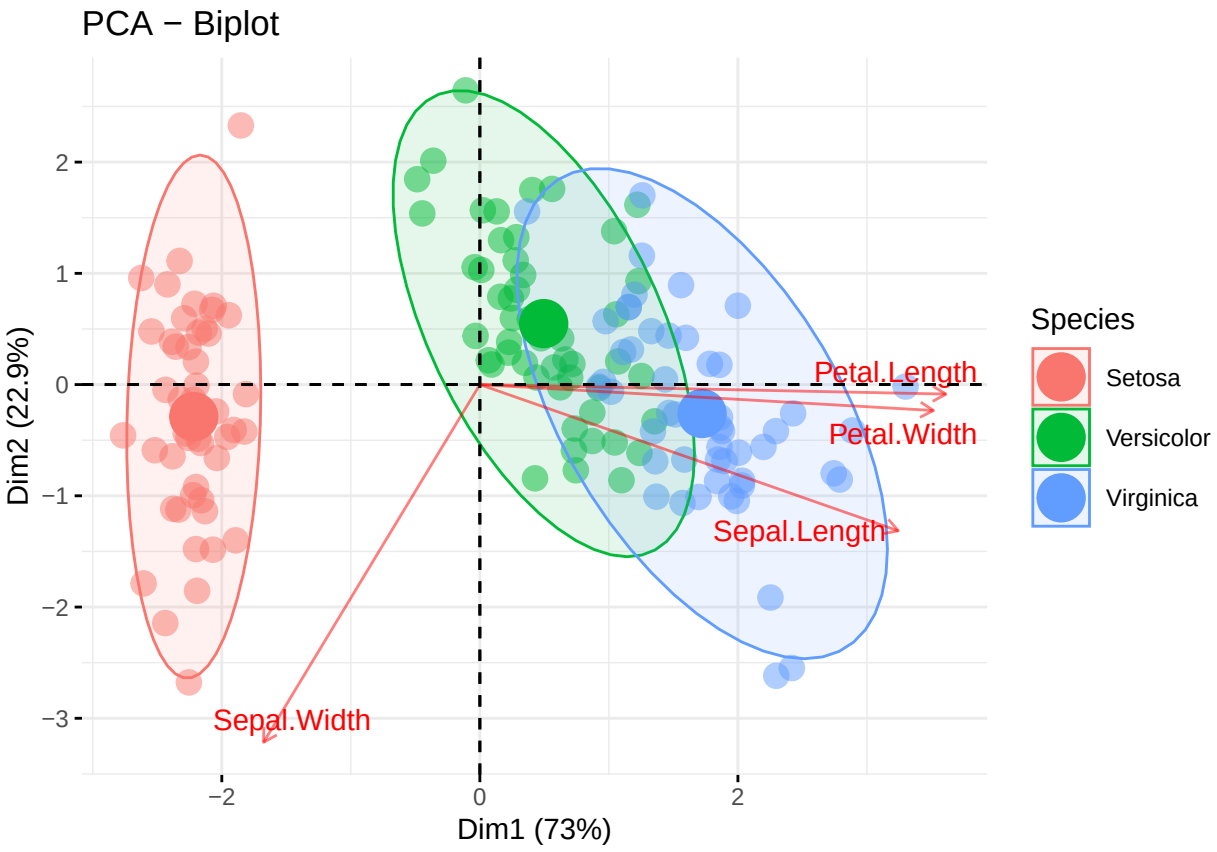
We can also visualise the data by plotting first three PCs in a 3-D plot. We can also change the rotation angle for a better look.

```
#3D plot
scatterplot3d(pca.iris$x[,1:3],pch=21,
              color=as.numeric(iris$Species)+1,
              bg=alpha(as.numeric(iris$Species)+1,0.4),
              cex.symbols=2,
              col.grid="steelblue",
              col.axis="steelblue",
              angle=70);
```



We can also use the biplot function from the **factoextra** package, which overlays the PCA plot with the loadings plot.

```
fviz_pca_biplot(pca.iris,
                axes = c(1,2),  #Specifying the PCs to be plotted.
                #Parameters for samples
                col.ind=iris$Species,  #Outline colour of the shape
                fill.ind=iris$Species,  #fill colour of the shape
                alpha=0.5,  #transparency of the fill colour
                pointsize=4,  #Size of the shape
                pointshape=21,  #Type of Shape
                #Parameter for variables
                col.var="red",  #Colour of the variable labels
                label="var",  #Show the labels for the variables only
                repel=TRUE,  #Avoid label overplotting
                addEllipses=TRUE,  #Add ellipses to the plot
                legend.title=list(colour="Species",fill="Species",alpha="Species"))
```



Exercise: Interpret the above biplot.

Enter text here

Let's consider the breast cancer dataset from the **mlbench** package. To find more information about the dataset, run the command **?BreastCancer**.

First, we need to remove the missing data, and then convert all the predictors to numeric.

```
data(BreastCancer)

#Remove all data with missing values and the ID (1st) column
bc <- na.omit(BreastCancer[,2:ncol(BreastCancer)])
n.feat <- ncol(bc)-1;  #number of features

for (I in 1:n.feat)
{
  bc[,I] <- as.numeric(as.character(bc[,I]))
}

str(bc)  #Examine the data structure
```

```
## 'data.frame':    683 obs. of  10 variables:
##  $ Cl.thickness   : num  5 5 3 6 4 8 1 2 2 4 ...
##  $ Cell.size      : num  1 4 1 8 1 10 1 1 1 2 ...
##  $ Cell.shape     : num  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marg.adhesion  : num  1 5 1 1 3 8 1 1 1 1 ...
##  $ Epith.c.size   : num  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.nuclei    : num  1 10 2 4 1 10 10 1 1 1 ...
##  $ Bl.cromatin    : num  3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.nucleoli: num  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : num  1 1 1 1 1 1 1 1 5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:16] 24 41 140 146 159 165 236 250 276 293 ...
##   ..- attr(*, "names")= chr [1:16] "24" "41" "140" "146" ...
```

Exercise: Perform PCA on the *cleaned* breast cancer dataset and answer the following questions. You will also need to consider whether the data need to be standardised in the PCA process.

1. What are the individual and cumulative proportions of variance explained by the first **5** principal components?

2. How many PCs (i.e. dimensions) would you need to explain at least 80% of the original variation?

3. Generate the biplot with PC1 and PC2. What can you say about patients with benign and malignant cancer in relation to the features that were measured?
4. Can PC1 or PC2 be used as a classifier?

Enter text here