PCA Reducing Dimentionality - Iris

Joseph Shu

2/24/2020

One of the most famous data sets in statistics is Fisher's iris data. The data set (avail- able in file iris.csv) contains measurements of 50 specimens from each of three different species of iris — Iris setosa, Iris versicolor, and Iris virginica — on the following di- mensions (measurements are in millimeters):

- X1 species (1 = Iris setosa, 2 = Iris versicolor, 3 = Iris virginica)
- X2 sepal length
- X3 sepal width
- X4 petal length
- X5 petal width

(a) Analyze the iris data (variables X2–X5) using principal components analysis. How many components do you need to adequately describe the data? How would you interpret them?

(b) Plot the average principal component scores for each of the three different types of iris for the first two principal components. Describe your findings.

Install and load necessary packages

```
library(psych)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(data.table)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::%+%()     masks psych::%+%()
## x ggplot2::alpha()   masks psych::alpha()
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
iris <- read.csv("iris.csv")
```

Explore the Data and Summary Statistic

```
psych::describe(iris)
```

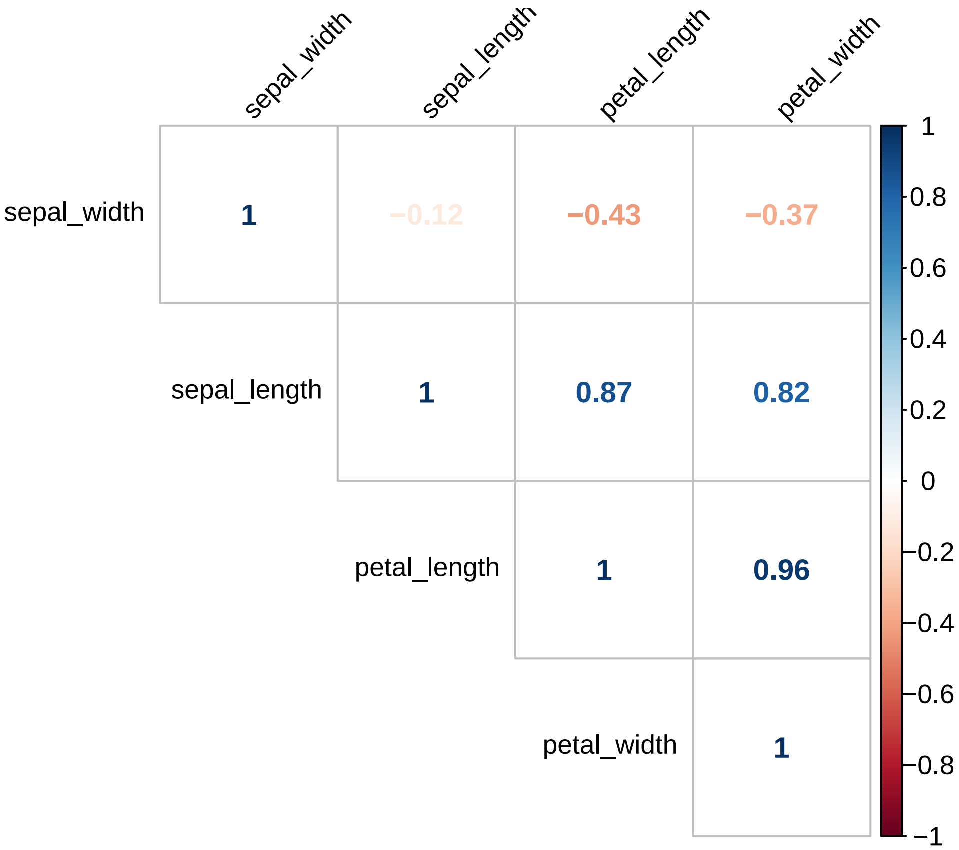
```
##              vars   n mean   sd median trimmed  mad min max range  skew
## species         1 150 2.00 0.82   2.00    2.00 1.48 1.0 3.0   2.0  0.00
## sepal_length    2 150 5.84 0.83   5.80    5.81 1.04 4.3 7.9   3.6  0.31
## sepal_width     3 150 3.06 0.44   3.00    3.04 0.44 2.0 4.4   2.4  0.31
## petal_length    4 150 3.76 1.77   4.35    3.76 1.85 1.0 6.9   5.9 -0.27
## petal_width     5 150 1.20 0.76   1.30    1.18 1.04 0.1 2.5   2.4 -0.10
##              kurtosis   se
## species         -1.52 0.07
## sepal_length    -0.61 0.07
## sepal_width      0.14 0.04
## petal_length    -1.42 0.14
## petal_width     -1.36 0.06
```

```
vars <- scale(iris[,-1])
cor <- cor(vars)
upper<-round(cor,3) # we round the results to the 3d digit after comma
upper[upper.tri(cor)]<-""
upper<-as.data.frame(upper)
upper
```

```
##              sepal_length sepal_width petal_length petal_width
## sepal_length            1                                     
## sepal_width        -0.118           1                         
## petal_length        0.872      -0.428            1            
## petal_width         0.818      -0.366        0.963           1
```

Correlation Matrix

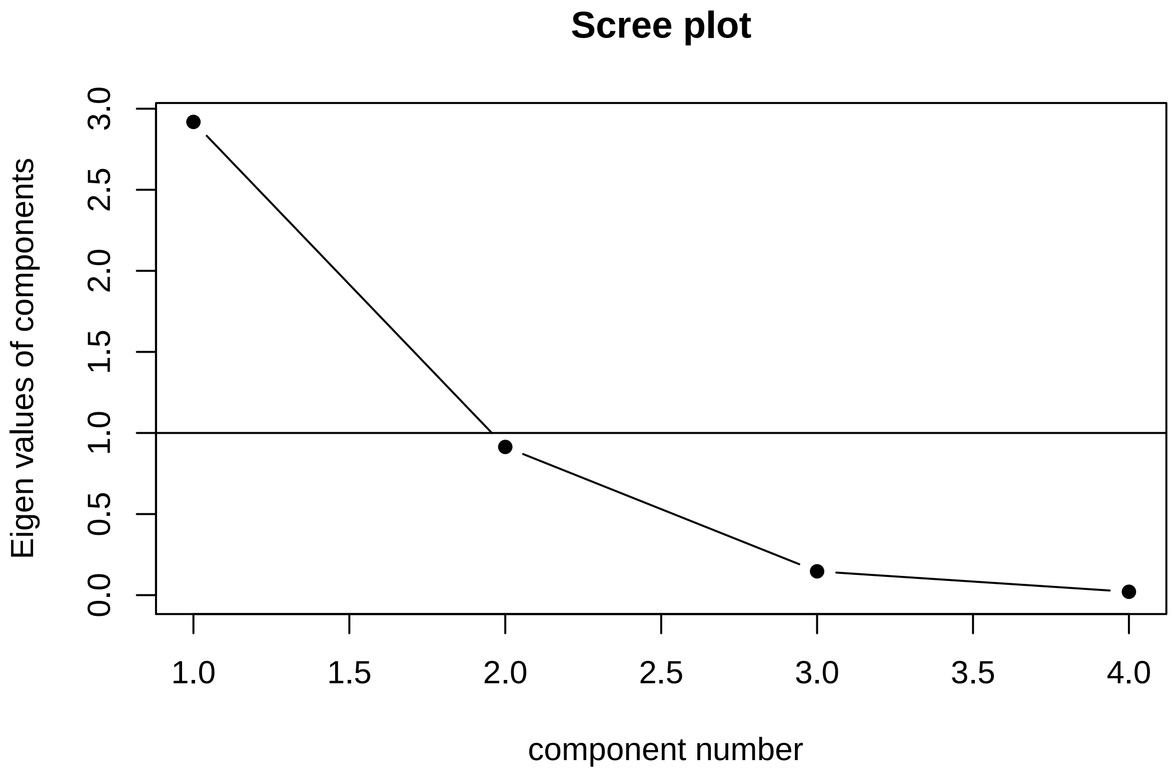
```
library(corrplot)
corrplot(cor,
         method = "number",
         type = "upper",
         order = "hclust", # reorder by the size of the correlation coefficients
         tl.cex = 0.8, # font size of the variable labels
         tl.col = "black", # color of the variable labels
         tl.srt = 45, # rotation angle for the variable labels
         number.cex = 0.9 # font size of the coefficients
)
```



	sepal_width	sepal_length	petal_length	petal_width
sepal_width	1	−0.12	−0.43	−0.37
sepal_length		1	0.87	0.82
petal_length			1	0.96
petal_width				1

We can see that sepal_length is highly correlated with petal_length and petal_width. Also, petal_length is highly correlated with petal_width.

```r
library(psych)
scree(cor, pc = TRUE, factors = FALSE)
```



A prelimaray thought of the scree plot is that, it might already be enough to use just one component to explain the variance, since the second component has a EV lower than 1.0

Finding the Eigen Value

```
EV = eigen(cor)$values
EV
```

```
## [1] 2.91849782 0.91403047 0.14675688 0.02071484
```

```
EV/length(EV)
```

```
## [1] 0.729624454 0.228507618 0.036689219 0.005178709
```

Here we can see in the first ouput that the EV value of the first component is 2.92 and the second component is 0.91. Futhermore, in second output, the first component can explain already about 73% of the variance.

```
cumsum(EV/length(EV))
```

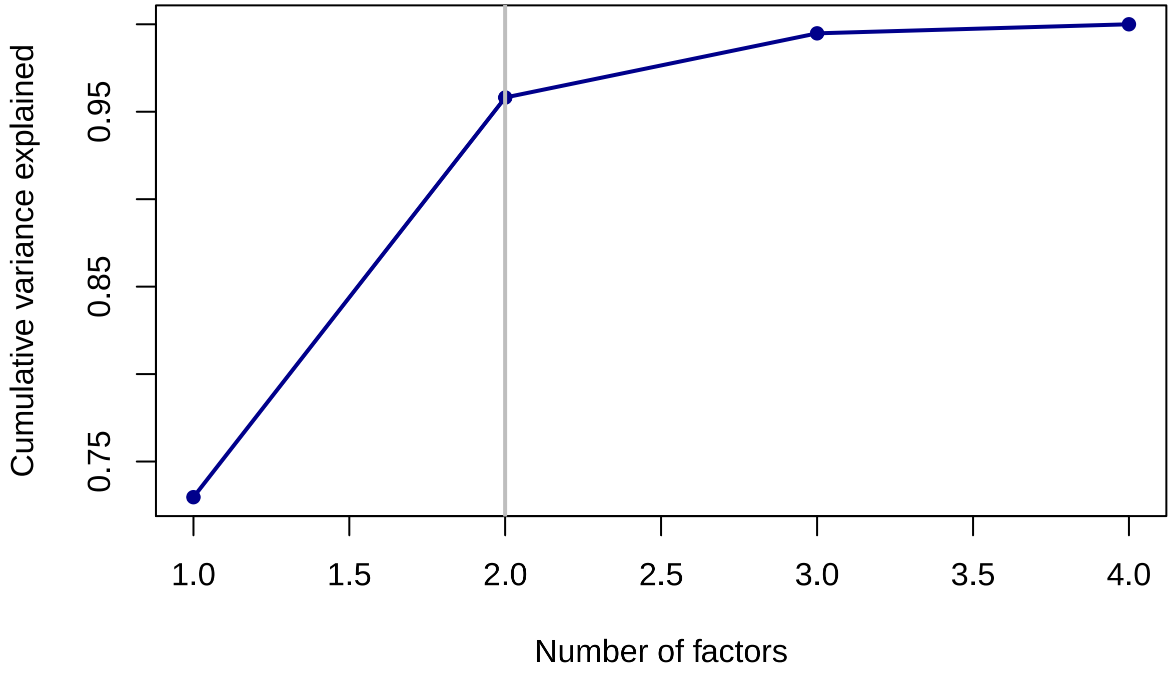
```
## [1] 0.7296245 0.9581321 0.9948213 1.0000000
```

A cumulative EV can be seen here, that if we take two factors, they can explain up to about 96% of the variance.

(a) By the exploratory analysis above, even though taking 2 factors we can explain a higher amount of variance, the second component has an EV value less than one. As a result, we just take one component that could explain about 73% of the variance for parsimony's sake.

What if we take 2 components?

```r
# Shares for the cumulative variance explained
plot(cumsum(EV/length(EV)),
     type = "o", # type of plot: "o" for points and lines 'overplotted'
     col = "darkblue",
     pch = 16, # plot symbol: 16 = filled circle
     cex = 1, # size of plot symbols
     xlab = "Number of factors", # a title for the x axis
     ylab = "Cumulative variance explained", # a title for the y axis
     lwd = 2) # line width
abline(v = 2, lwd = 2, col = "grey") # draw a vertical line at v = 2
```



Unrotated PCA

```r
PCA <- principal(r = cor,
                 nfactors = 2,
                 rotate="none",
                  scores = TRUE)
print(PCA,
      digits = 3, # to round numbers to the third digit
      cut = 0.35, # to show only values > 0.35
      sort = TRUE) # to sort rows by loading size0
```

```
## Principal Components Analysis
## Call: principal(r = cor, nfactors = 2, rotate = "none", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##              item    PC1    PC2    h2      u2  com
## petal_length    3  0.992        0.984 0.01627 1.00
## petal_width     4  0.965        0.935 0.06472 1.01
## sepal_length    1  0.890  0.361 0.923 0.07740 1.32
## sepal_width     2 -0.460  0.883 0.991 0.00908 1.51
##
##                         PC1   PC2
## SS loadings           2.918 0.914
## Proportion Var        0.730 0.229
## Cumulative Var        0.730 0.958
## Proportion Explained  0.762 0.238
## Cumulative Proportion 0.762 1.000
##
## Mean item complexity =  1.2
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.031
##
## Fit based upon off diagonal values = 0.998
```

We can see that with two pricipal components together, we can already explain the variance up to about 96% cumulatively, but as mentioned above, PC2 has a loadings of 0.91.

```
PCA$communality
```

```
## sepal_length  sepal_width petal_length  petal_width
##    0.9225986    0.9909193    0.9837300    0.9352804
```

If we take two components, they can capture most of the information in all dimentions. The communalities are all over 90%, some even up to 99%.

Unrotated factor loadings

```
PCA$loadings
```

```
##
## Loadings:
##              PC1    PC2
## sepal_length  0.890  0.361
## sepal_width  -0.460  0.883
## petal_length  0.992
## petal_width   0.965
##
##                  PC1   PC2
## SS loadings    2.918 0.914
## Proportion Var 0.730 0.229
## Cumulative Var 0.730 0.958
```

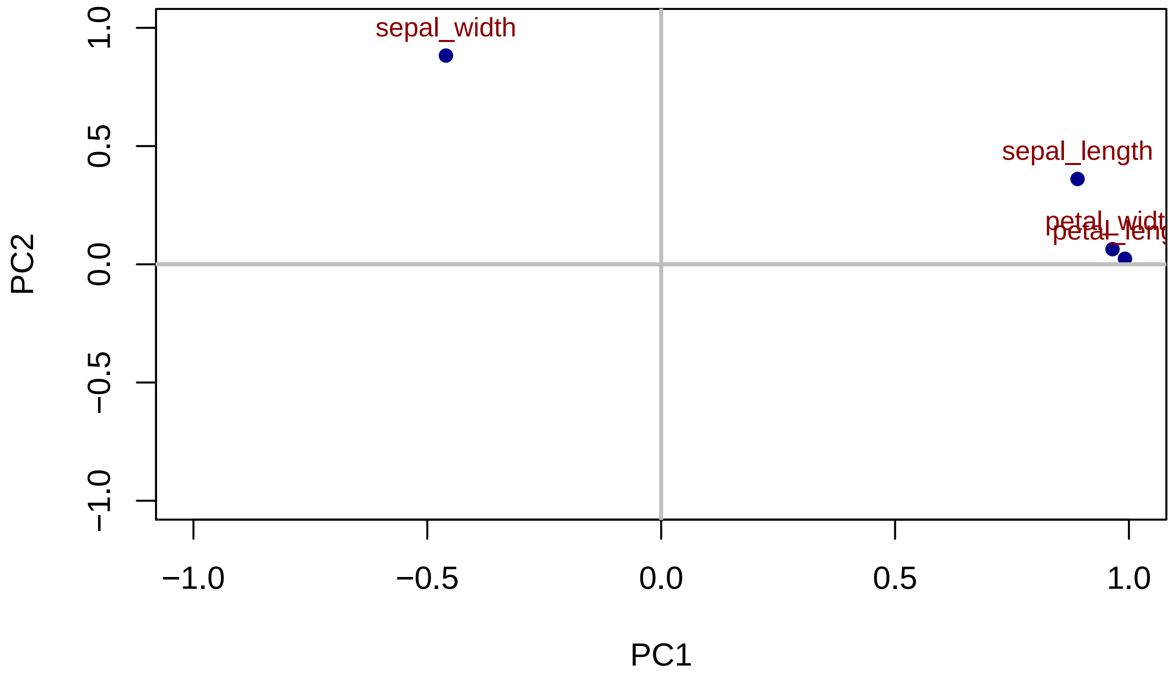
Without rotation, it is unclear that sepal_width belongs to which component. We can visualize it with a better interpretation.

```
L <- as.data.table(unclass(PCA$loadings), keep.rownames = T)
```

```
plot(x = L$PC1, y = L$PC2,
     col ="darkblue",
     pch = 16,        # plot symbol: 16 = filled circle
     cex = 1,         # size of plot symbols
     xlab = "PC1",    # a title for the x axis
     ylab = "PC2",    # a title for the y axis
     xlim = c(-1,1),  # x axis values from -1 to 1
     ylim = c(-1,1))  # y axis values from -1 to 1

# add point labels
text(L$PC1, L$PC2,
     labels = L$rn,
     pos = 3,
     cex = 0.8,
     col = "darkred")

# add vertical and horizontal lines
abline(h = 0, lwd = 2, col = "grey") # draw a horizontal line at h = 0
abline(v = 0, lwd = 2, col = "grey") # draw a vertical line at v = 0
```



Without rotation, we can see that PC1 captures more information than PC2. So let's see how it look like later with rotation.

We could also plot each observation of three different species in two dimentions to see we can remark the patterns somehow.

```
# extract un-rotated scores of principal components
PCA.scores = factor.scores(vars, unclass(PCA$loadings))$scores
head(PCA.scores)
```

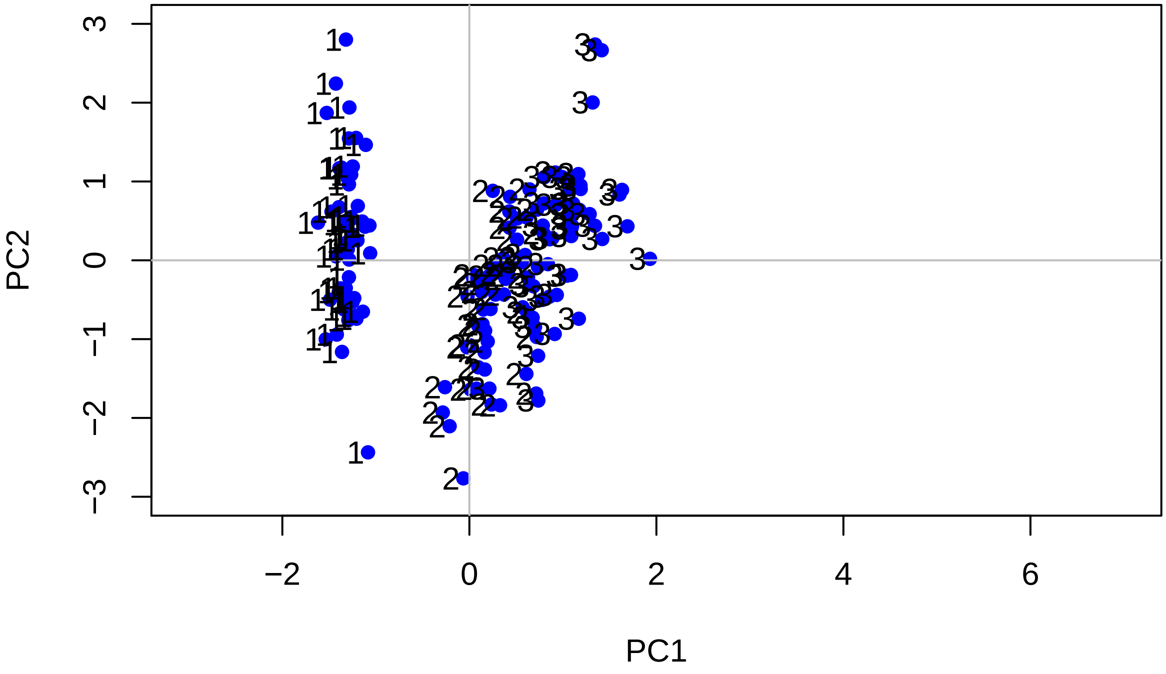
```
##            PC1        PC2
## [1,] -1.321232  0.5004175
## [2,] -1.214037 -0.7027698
## [3,] -1.379296 -0.3564318
## [4,] -1.341465 -0.6227710
## [5,] -1.394238  0.6743121
## [6,] -1.210927  1.5524358
```

```
iris.scores <- cbind(iris, PCA.scores)
```

```
plot(x = iris.scores$PC1,
     y = iris.scores$PC2,
     xlab = "PC1", ylab = "PC2",
     xlim = c(-3, 7), ylim = c(-3, 3),
     pch = 16, cex = 1, col = "blue")
```

```
abline(h = 0, col = "grey")
abline(v = 0, col = "grey")

# add point labels
text(x = iris.scores$PC1,
     y = iris.scores$PC2,
     labels = iris.scores$species,
     cex = 1,
     adj = 1.2,
     col = "black")
```



We can see that PC1 already captures most of the patterns for three different species. Each species obviously scores similarly based on PC1. However, species two scores mostly around 0, which implies that PC1 may not capture much of species two.

Rotated PCA solution

```
rotatedPCA <- principal(r = cor,
           nfactors = 2,
           rotate = "varimax",
           scores = TRUE)

print(rotatedPCA,
      digits = 3, # round numbers to the 3d digits
      cut = 0.5, # to show only values > 0.5
      sort = TRUE # sort rows by factor
      )
```

```
## Principal Components Analysis
## Call: principal(r = cor, nfactors = 2, rotate = "varimax", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##              item    RC1    RC2    h2      u2  com
## sepal_length    1  0.959        0.923 0.07740 1.01
## petal_length    3  0.944        0.984 0.01627 1.20
## petal_width     4  0.932        0.935 0.06472 1.15
## sepal_width     2         0.985 0.991 0.00908 1.04
##
##                         RC1   RC2
## SS loadings           2.702 1.130
## Proportion Var        0.676 0.283
## Cumulative Var        0.676 0.958
## Proportion Explained  0.705 0.295
## Cumulative Proportion 0.705 1.000
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.031
##
## Fit based upon off diagonal values = 0.998
```

After rotation, it is explicit that sepal_length, petal_length and petal_width can be explained in the first component. Sepal_width can be explained in the second component.

Rotated factor loadings

```
rotatedPCA$loadings
```

```
##
## Loadings:
##              RC1    RC2
## sepal_length  0.959
## sepal_width  -0.145  0.985
## petal_length  0.944 -0.304
## petal_width   0.932 -0.257
##
##                  RC1   RC2
## SS loadings    2.702 1.130
## Proportion Var 0.676 0.283
## Cumulative Var 0.676 0.958
```

The un-cut loadings for each attribute can be seen here.

We can the plot the average principle component scores for each of the three different species in two dimentions.

```r
# extract rotated factor scores
rotatedPCA.scores = factor.scores(vars, unclass(rotatedPCA$loadings))$scores
head(rotatedPCA.scores)
```

```
##              RC1        RC2
## [1,] -1.0834754  0.9067262
## [2,] -1.3775358 -0.2648876
## [3,] -1.4198321  0.1165198
## [4,] -1.4716069 -0.1474634
## [5,] -1.0952963  1.0949536
## [6,] -0.6336551  1.8641036
```

```r
iris.scores <- cbind(iris, rotatedPCA.scores)
head(iris.scores)
```

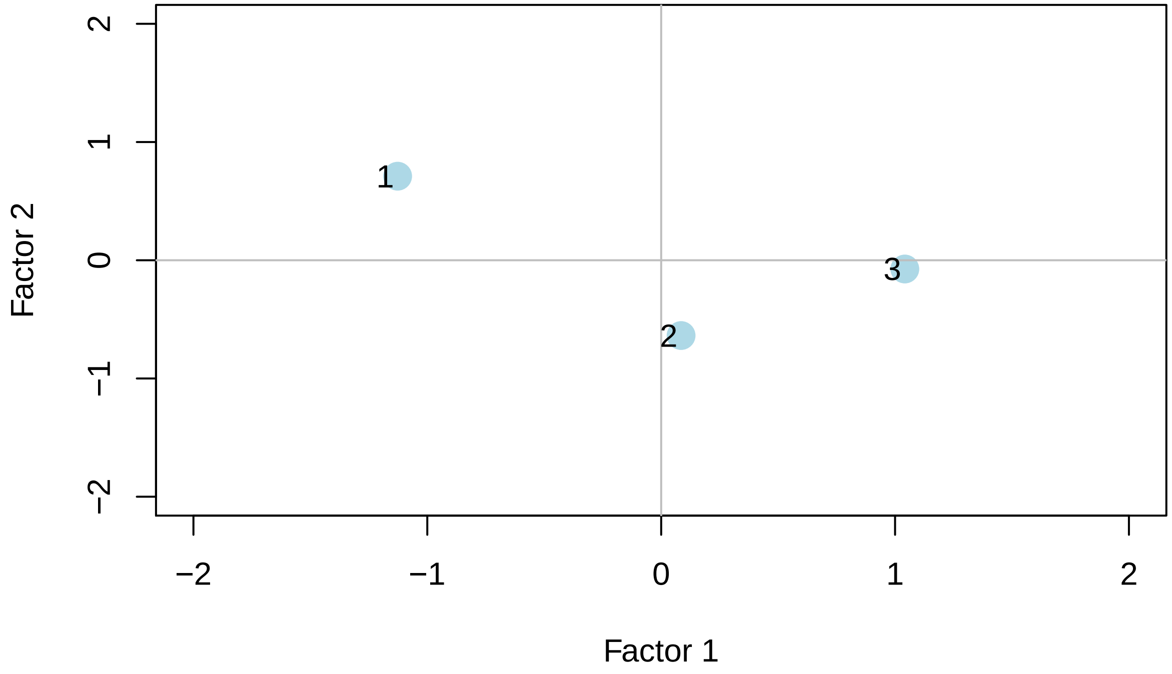
```
##   species sepal_length sepal_width petal_length petal_width        RC1
## 1       1          5.1         3.5          1.4         0.2 -1.0834754
## 2       1          4.9         3.0          1.4         0.2 -1.3775358
## 3       1          4.7         3.2          1.3         0.2 -1.4198321
## 4       1          4.6         3.1          1.5         0.2 -1.4716069
## 5       1          5.0         3.6          1.4         0.2 -1.0952963
## 6       1          5.4         3.9          1.7         0.4 -0.6336551
##          RC2
## 1  0.9067262
## 2 -0.2648876
## 3  0.1165198
## 4 -0.1474634
## 5  1.0949536
## 6  1.8641036
```

```r
# Average values for factor scores for each species
mean.scores = aggregate(iris.scores[, c("RC1", "RC2")],
          by = list(Species = iris.scores$species),
          FUN = mean)
mean.scores
```

```
##   Species         RC1         RC2
## 1       1 -1.12691479  0.71091113
## 2       2  0.08511478 -0.63685911
## 3       3  1.04180001 -0.07405202
```

```r
plot(x = mean.scores$RC1,
     y = mean.scores$RC2,
     xlab = "Factor 1", ylab = "Factor 2",
     xlim = c(-2, 2), ylim = c(-2, 2),
     pch = 16, cex = 2, col = "lightblue")

abline(h = 0, col = "grey")
abline(v = 0, col = "grey")
# add point labels
text(x = mean.scores$RC1,
     y = mean.scores$RC2,
     labels = mean.scores$Species,
     cex = 1,
     adj = 1.2,
     col = "black")
```



Species three (iris virginica) focuses more on factor 1 than factor 2 ; species two (iris versicolor), as we mentioned above, cannot be captured from factor 1 and scores lower than average in factor 2. Species one iris setosa, on the other hand, focuses more on factor 2.