Unsupervised Learning Assignment 1

This notebook contains all the code for the unsupervised learning assignment 1 implementation of Principal Component Analysis(PCA) on the TrackRecords data set.

PCA enables the summarization and visualization of information in a dataset described by multiple intercorrelated quantitive variables. Through this process PCA acts to reduce the dimentionality of a dataset.

PCA's goal is to extract useful information from a multvariate data set and to express this useful information as a set of principle components. These principle components correspeond to a linear combination of the original variables.

Information in a dataset can be quantified by it's total variation. The goal of PCA is to identify the directions along which the variation in the data is maximum and therby extract the principle components.

As a result, PCA can be used to reduce the dimensionality of a multivarable data set. This enables easy visualization in two or three dimensions of a much higher dimentional dataset.

Enviroment Setup and data import

Begin by adding the libraries that are needed and reading in the data. Drop the first two rows as these are row names and indexes. The row names are added back later.

```
#clean workspace
rm(list=ls())

#install librarys
suppressMessages(library(factoextra))
suppressMessages(library("corrplot"))
suppressMessages(require(ggplot2))
suppressMessages(library(psych))

#import data
trackRecords <- read.csv("TrackRecords.csv")
X <- as.matrix(trackRecords[,-1:-2]) #remove the row number and row name
rownames(X) <- trackRecords[,2] #set the row name as the matrix row name
```

#Basic data exploration

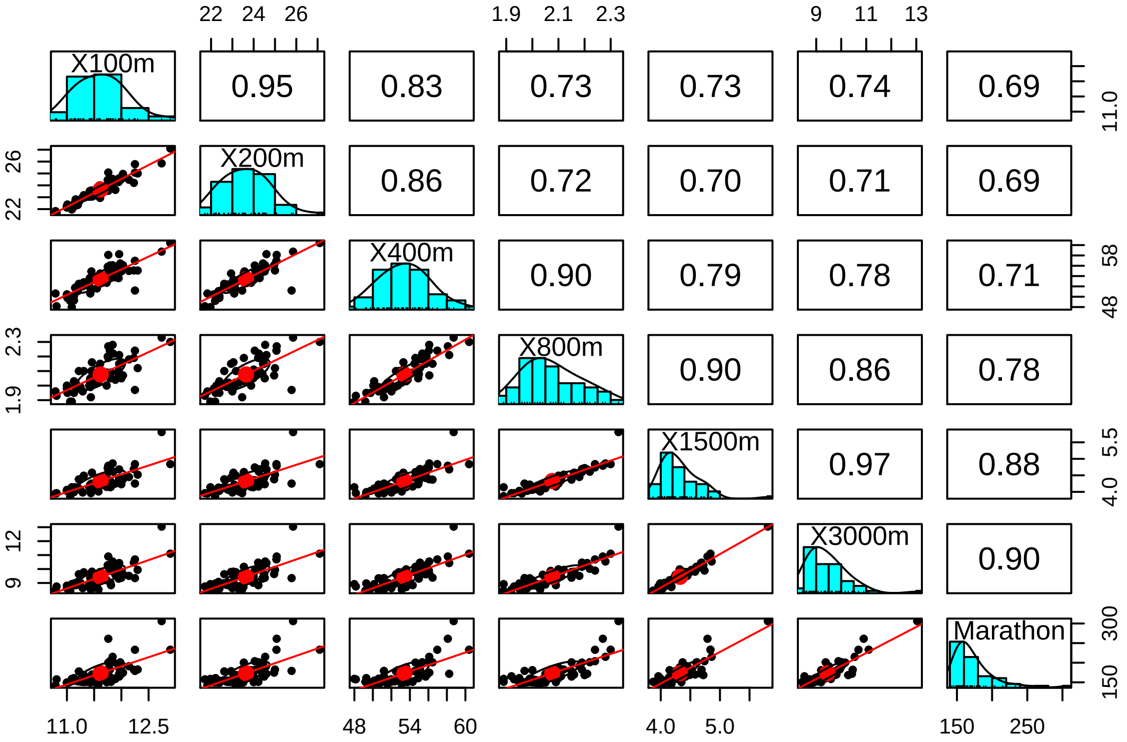
```
rownames(X)
```

```
##  [1] "argentin" "australi" "austria"  "belgium"  "bermuda"  "brazil"
##  [7] "burma"    "canada"   "chile"    "china"    "columbia" "cookis"
## [13] "costa"    "czech"    "denmark"  "domrep"   "finland"  "france"
## [19] "gdr"      "frg"      "gbni"     "greece"   "guatemal" "hungary"
## [25] "india"    "indonesi" "ireland"  "israel"   "italy"    "japan"
## [31] "kenya"    "korea"    "dprkorea" "luxembou" "malaysia" "mauritiu"
## [37] "mexico"   "netherla" "nz"       "norway"   "png"      "philippi"
## [43] "poland"   "portugal" "rumania"  "singapor" "spain"    "sweden"
## [49] "switzerl" "taipei"   "thailand" "turkey"   "usa"      "ussr"
## [55] "wsamoa"
```

```
sapply(trackRecords,class)
```

```
##        OBS    COUNTRY      X100m      X200m      X400m      X800m     X1500m
## "integer"  "factor" "numeric" "numeric" "numeric" "numeric" "numeric"
##    X3000m  Marathon
## "numeric" "numeric"
```

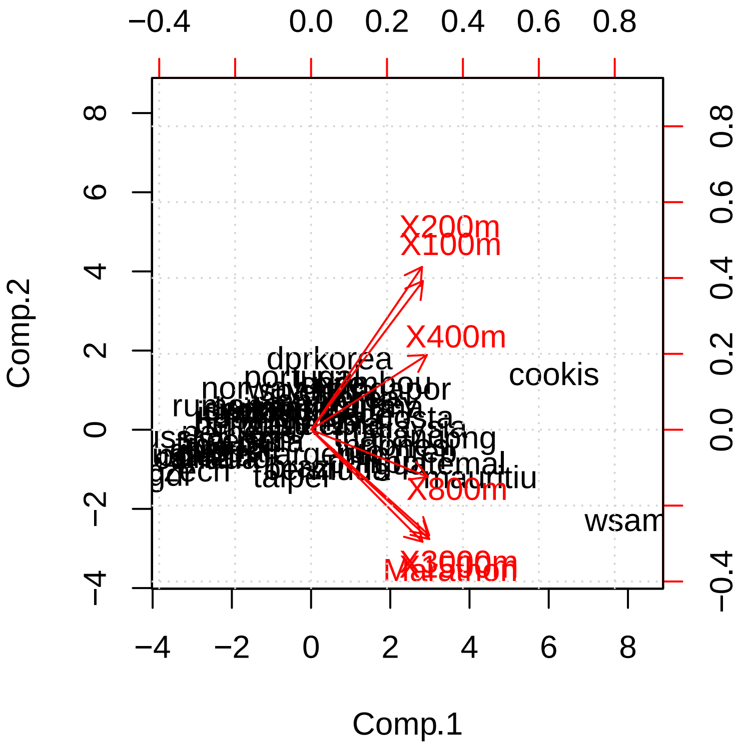
```
pairs.panels(trackRecords[,-1:-2],cex=1,lm=TRUE)
```



Generation of PCA

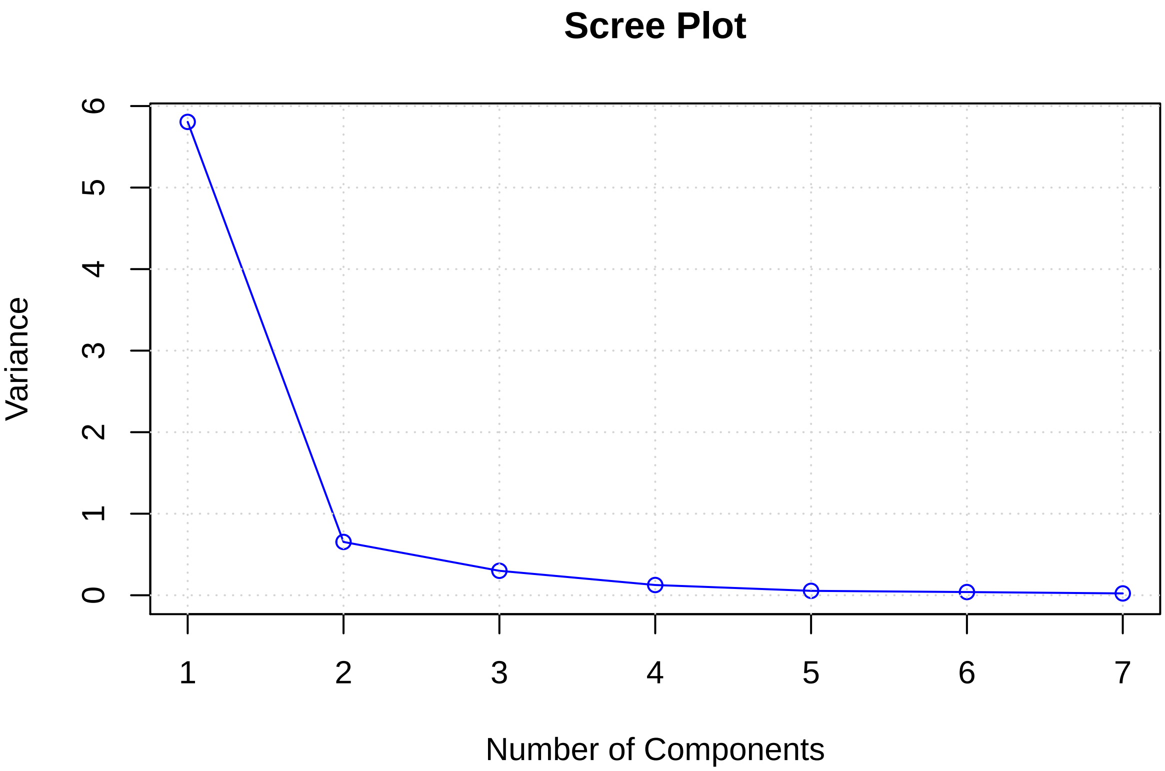
Next generate the basic PCA and create a simple biplot. This graph is refined later on.

```
#preform PCA
#Cor=TRUE: the data will be centered and scaled before the analysis
#scores=TRUE: the coordinates on each principal component are calculated
pca.out <- princomp(X, cor=TRUE, scores=TRUE)

#generate basic biplot
biplot(pca.out,scale=0)
grid()
```



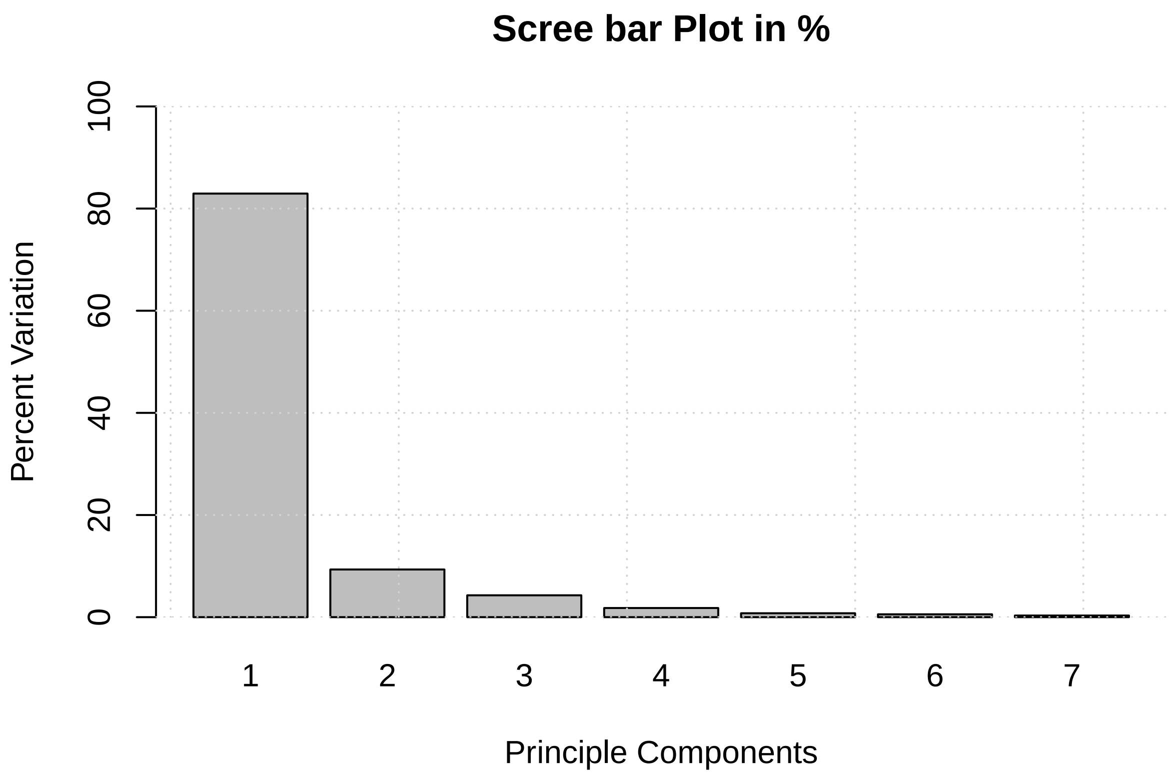
# Basic scree plot We can quantify the total variance expressed by the diffrent principle components using a scree plot. First, this is done using a line then a bar plot.

```
Variance<-(pca.out$sdev)^2 #calculate the standard deviation of this principle component
max_Var<-round(max(Variance),1)
Components<-c(1:length(Variance))
Components<-as.integer(Components)
plot(Components,
     Variance,
     main="Scree Plot",
     xlab="Number of Components",
     ylab="Variance",
     type="o",
     col="blue",
     ylim=c(0,max_Var))
grid()
```



# Bar plot-scree plot and variance explained Can express the total variance explain as the sum of all the variance expressed by each principle component. Can then plot this as a bar plot

```
totalVariance <- sum(Variance)
fractionOfVariance <- (Variance / totalVariance) * 100
barplot(fractionOfVariance,
        main = "Scree bar Plot in %",
        xlab = "Principle Components"
        , ylab="Percent Variation"
        , ylim = c(0,100)
        , names.arg = c(1:7))
grid()
```



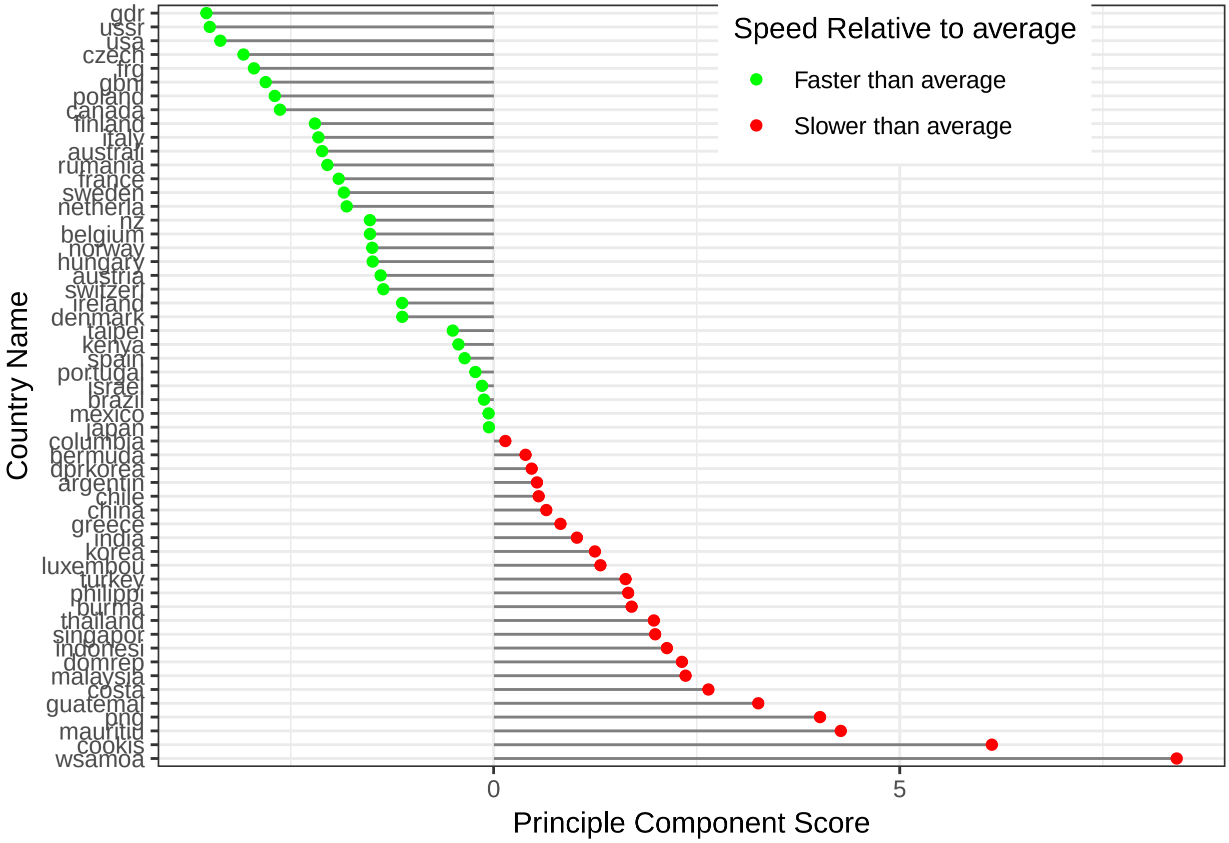
```
first2PC <- sum(fractionOfVariance[1:2])
first2PC
```

```
## [1] 92.27616
```

Order countreis based on first component

Next, the nations are ranked based on the score of the first principle component. This plot shows

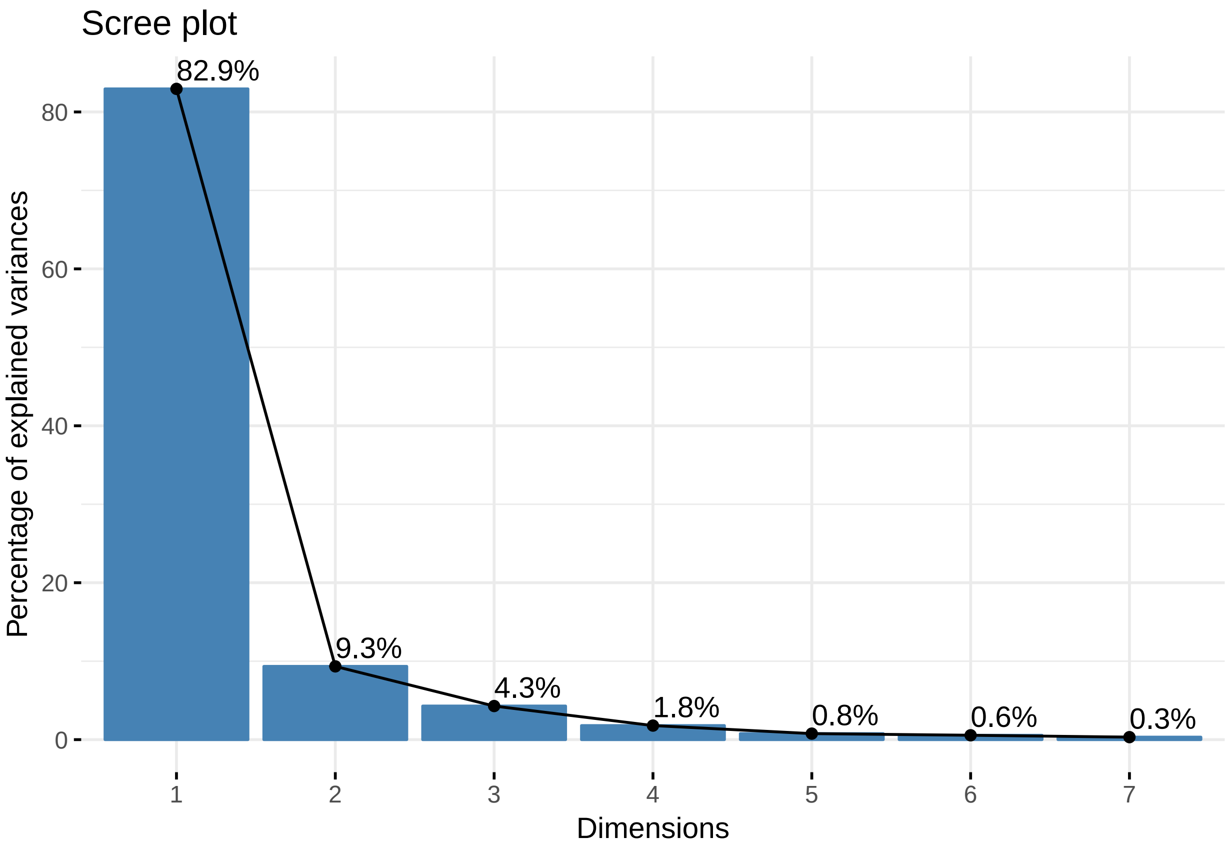
```
#order and process the data
pc1Ranked <- order(pca.out$scores[,1], decreasing = TRUE)
orderedCountries <- rownames(pca.out$scores)[pc1Ranked]
orderedValues <- pca.out$scores[pc1Ranked]
plotDataFrame <- cbind(orderedCountries,orderedValues)
plotDataFrame  <- as.data.frame(plotDataFrame)
plotDataFrame$orderedValues <- as.numeric(as.character(plotDataFrame$orderedValues))

#generate plot
plotDataFrame$orderedCountries <- factor(plotDataFrame$orderedCountries,
                                          levels = rownames(pca.out$scores)[pc1Ranked])
ggplot(plotDataFrame,
       aes(x=orderedValues,
           y=orderedCountries,
           color = orderedValues > 0)) +
  labs(x = "Principle Component Score",
       y = "Country Name",
       color = "Speed Relative to average") +
  scale_color_manual(labels = c("Faster than average",
                                "Slower than average"),
                     values = c("green", "red")) +
        geom_segment(aes(x = 0,
                         y = orderedCountries,
                         xend = orderedValues,
                         yend = orderedCountries),
                     color = "grey50") +
        geom_point() +
  theme_bw() +
  theme(legend.position=c(0.7,0.9))
```



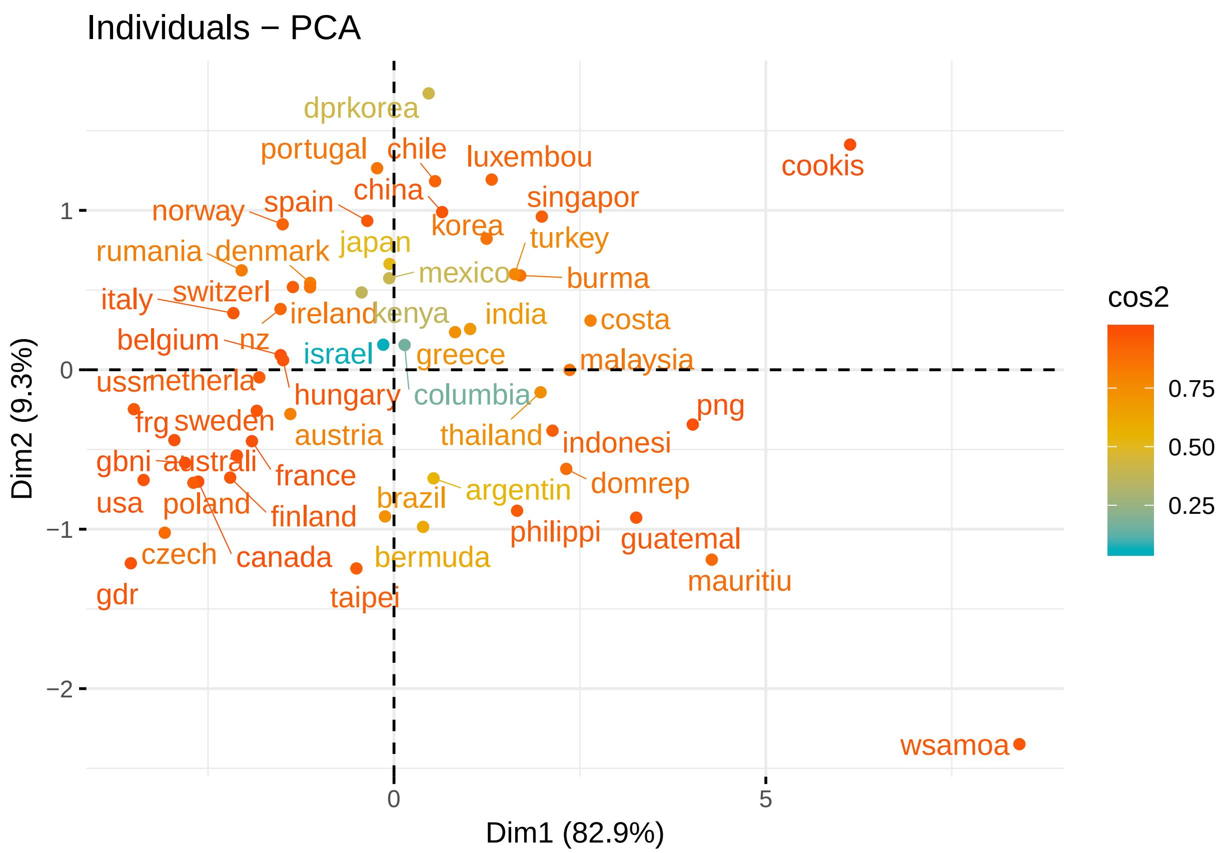
Better visualization of Scree and Cos2 valuation

Next, there are a number of visualizations that we can perform on the PCA.

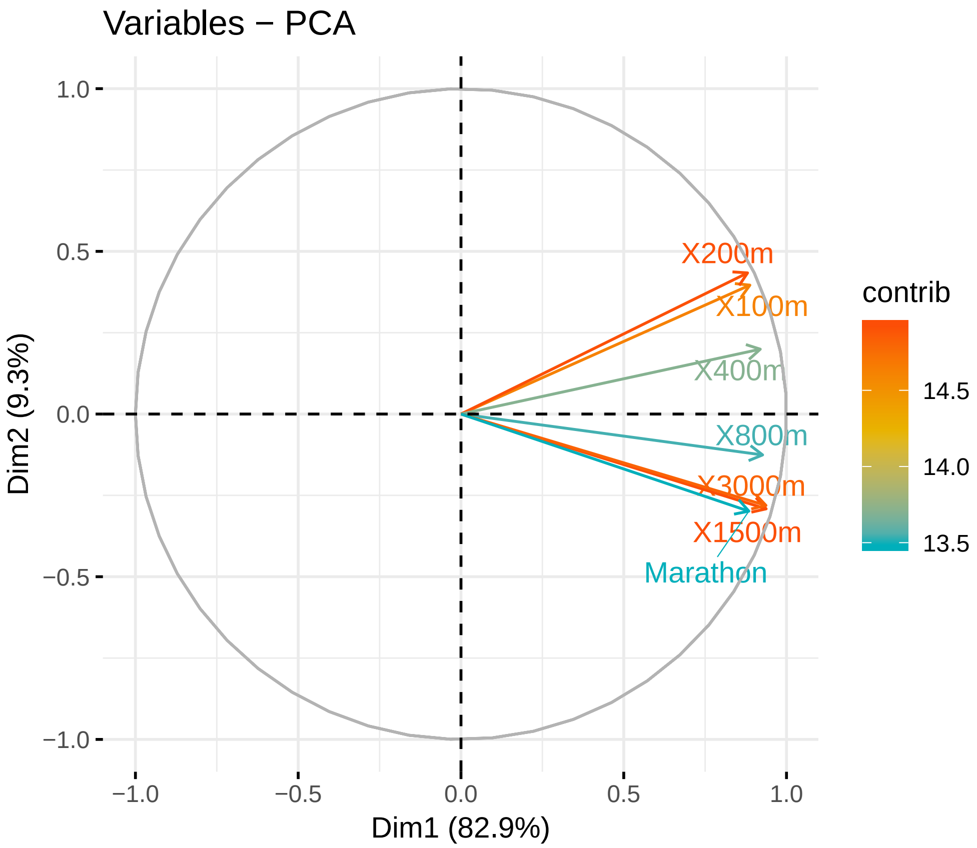
```
#Visualize eigenvalues (scree plot).
#Show the percentage of variances explained by each principal component.
fviz_eig(pca.out, addlabels = TRUE)
```



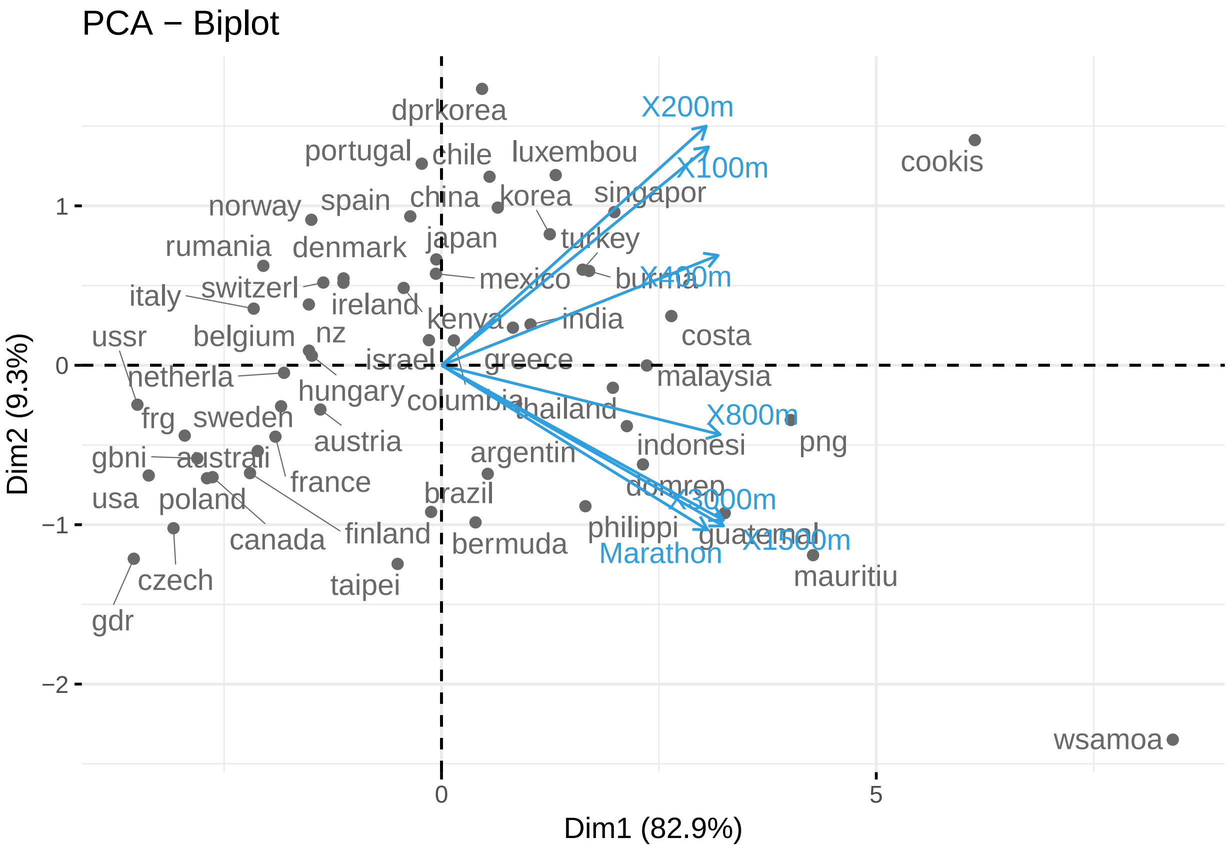
```
#Graph of individuals. Individuals with a similar profile are grouped together.
fviz_pca_ind(pca.out,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



```
#Graph of variables.
#Positive correlated variables point to the same side of the plot.
#Negative correlated variables point to opposite sides of the graph.
fviz_pca_var(pca.out,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



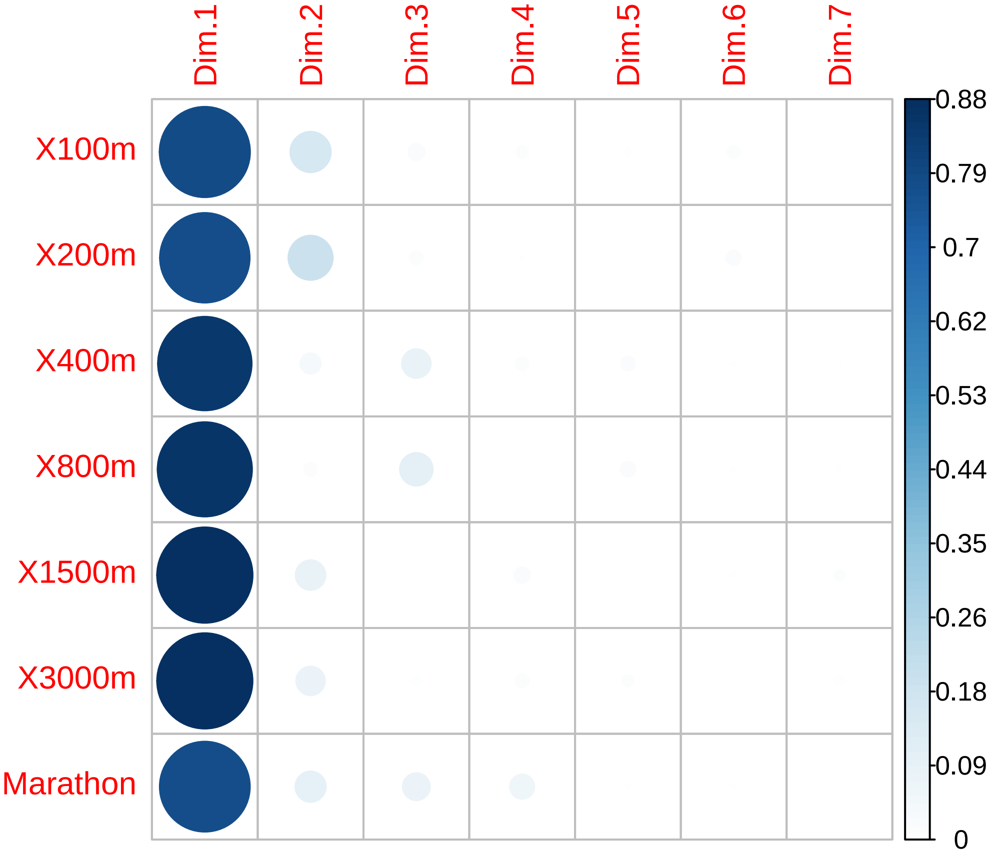
```r
#Biplot of individuals and variables
fviz_pca_biplot(pca.out, repel = TRUE,
                col.var = "#2E9FDF", # Variables color
                col.ind = "#696969"  # Individuals color
                )
```



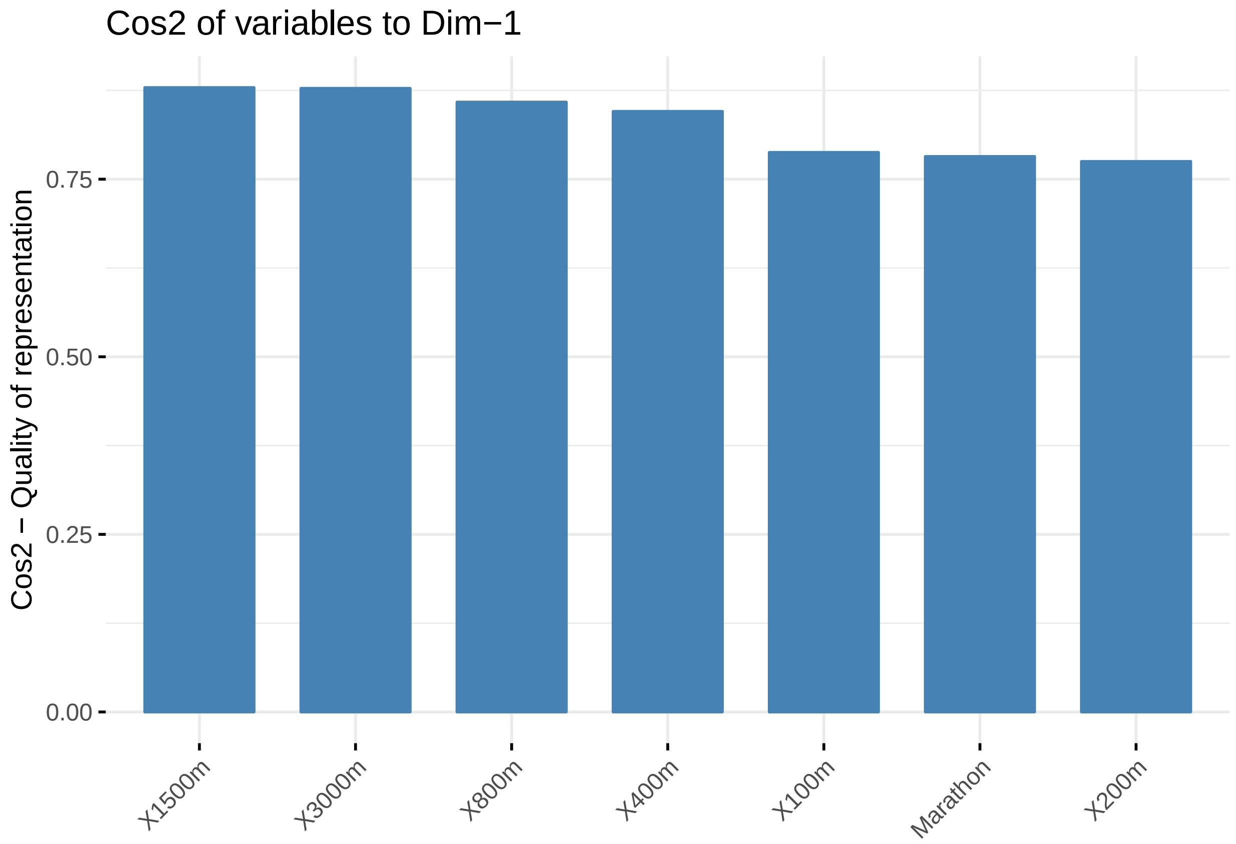
# Further Cos2 evalation A high cos2 indicates a good representation of the variable on the principal component. In this case the variable is positioned close to the circumference of the correlation circle.

A low cos2 indicates that the variable is not perfectly represented by the PCs. In this case the variable is close to the center of the circle.

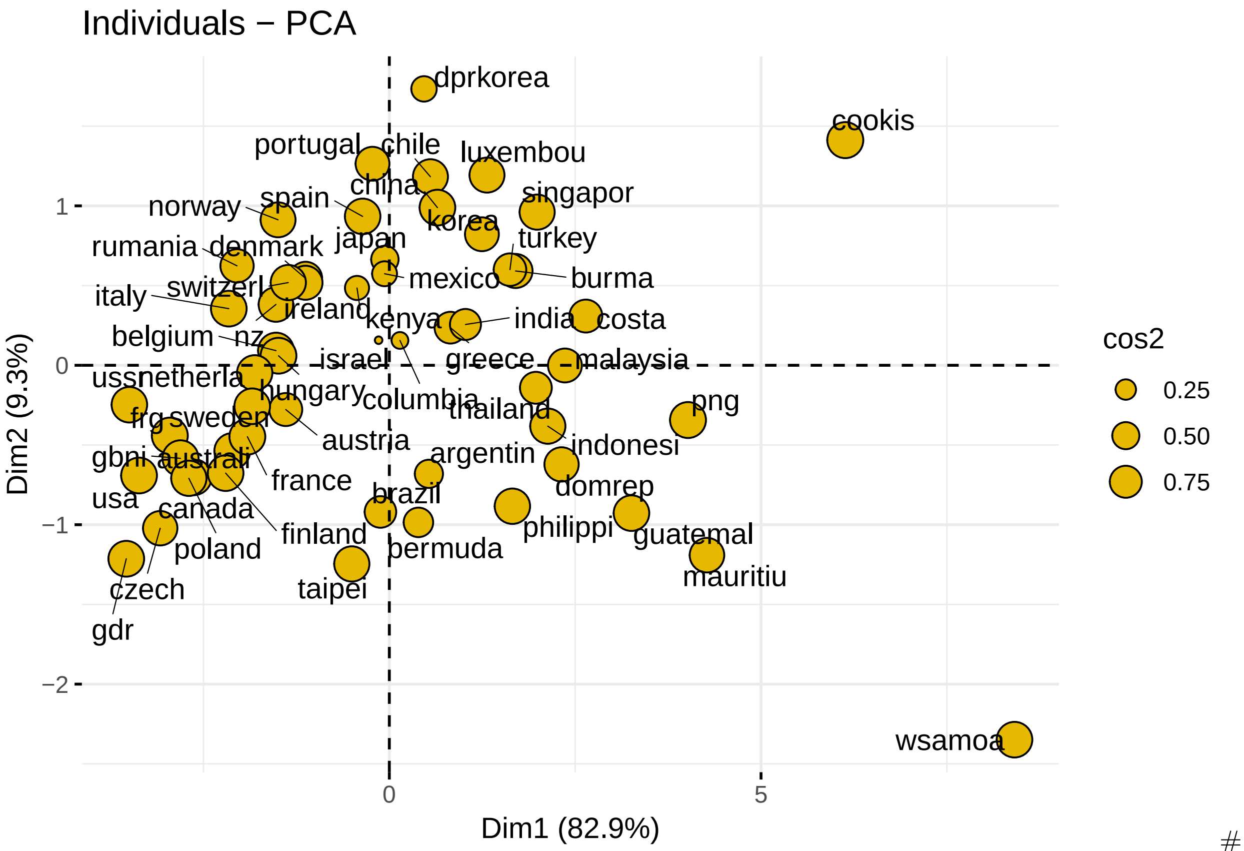
```
#We can look at the cos2 value.
#This represents the quality of representation for variables on the factor map.
# It’s calculated as the squared coordinates: var.cos2 = var.coord * var.coord.
var <- get_pca_var(pca.out)
corrplot(var$cos2, is.corr=FALSE)
```



```
#It's also possible to create a bar plot of variables cos2
fviz_cos2(pca.out, choice = "var", axes = 1)
```



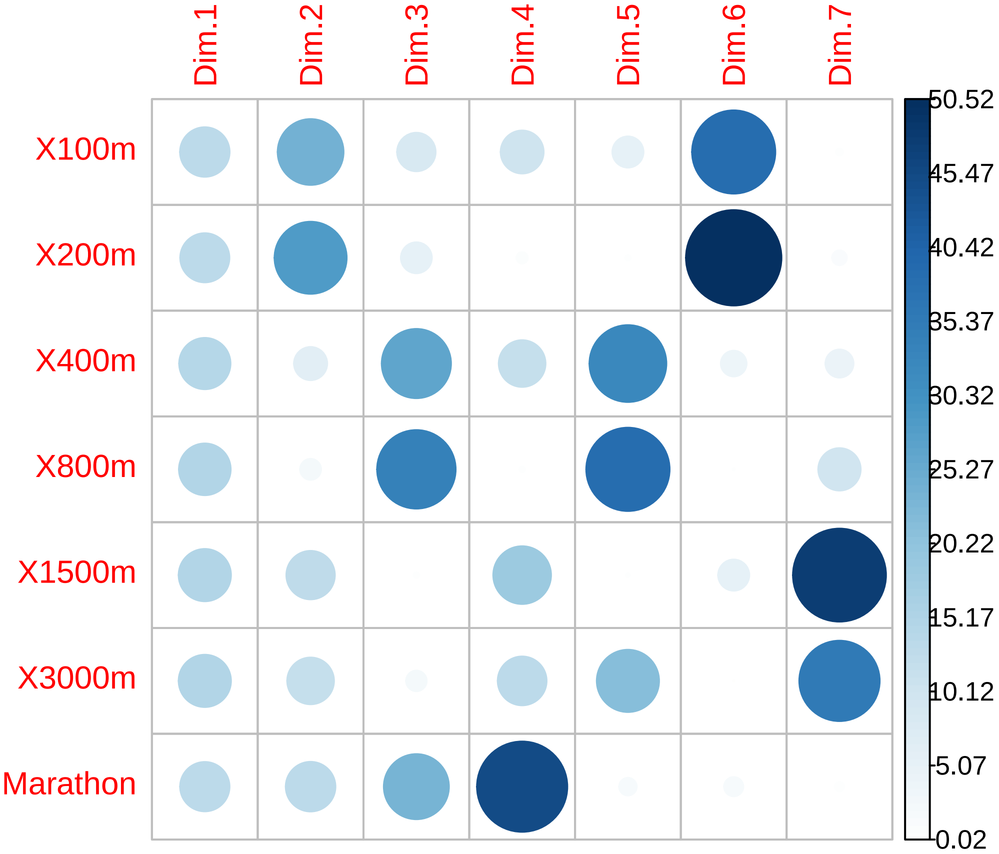
```
fviz_pca_ind(pca.out, pointsize = "cos2",
             pointshape = 21, fill = "#E7B800",
             repel = TRUE # Avoid text overlapping (slow if many points)
             )
```



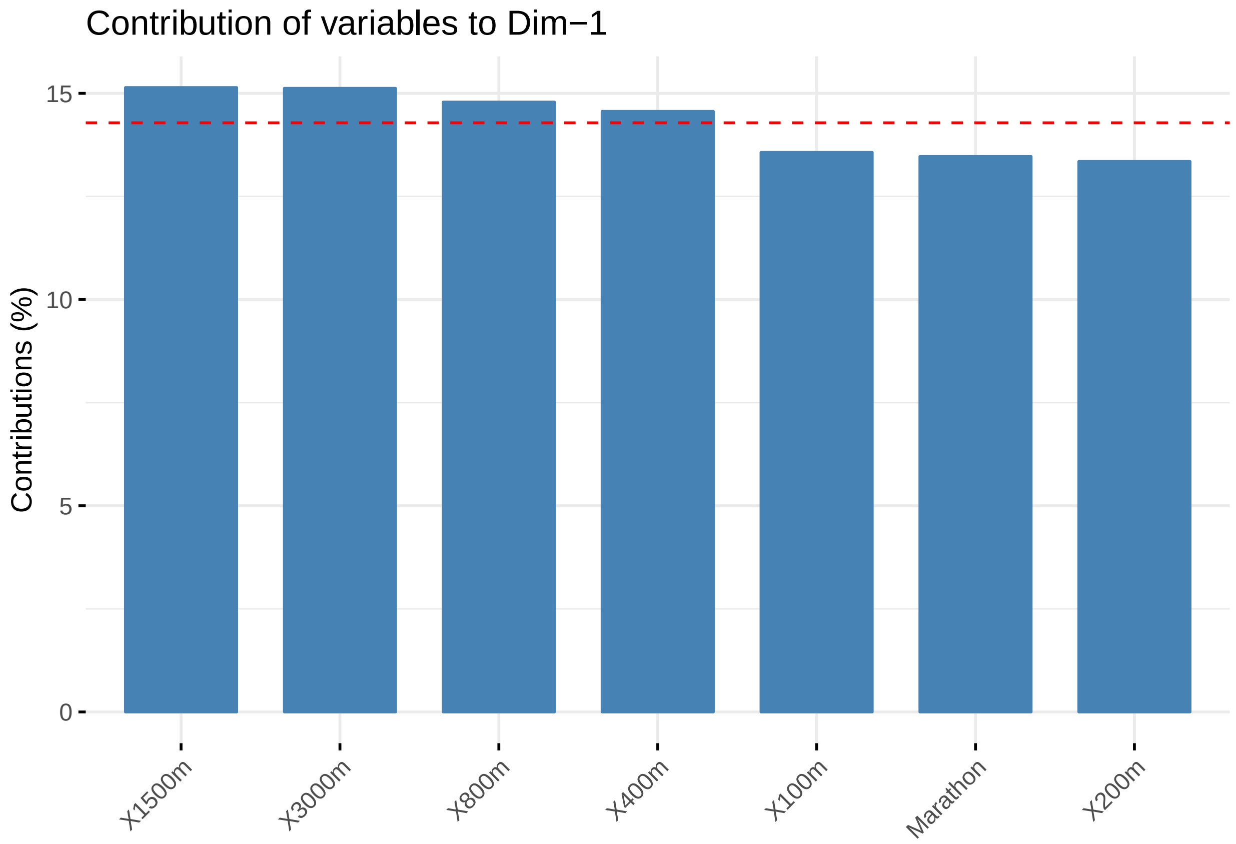
#

Dimension contribution Plots We can also look at the contribution to each principle component.

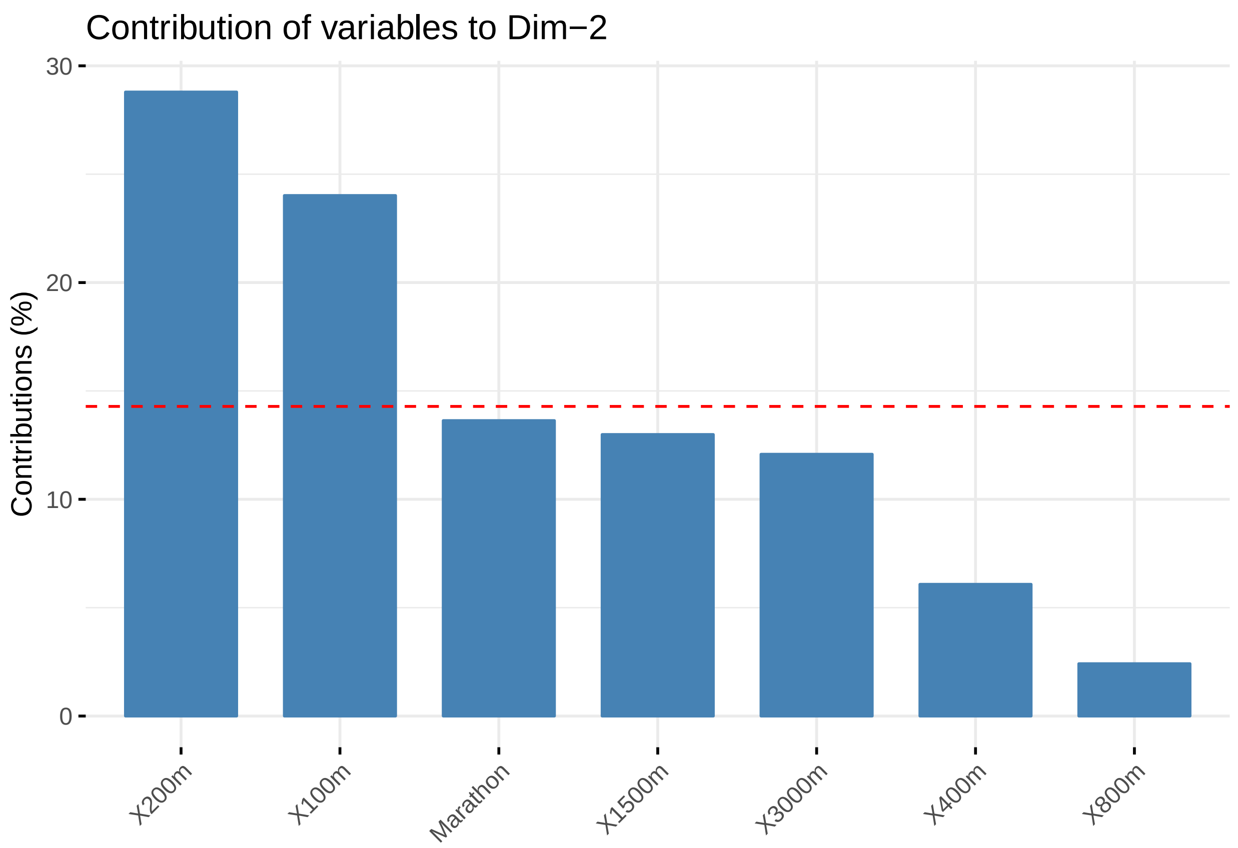
```
corrplot(var$contrib, is.corr=FALSE)
```



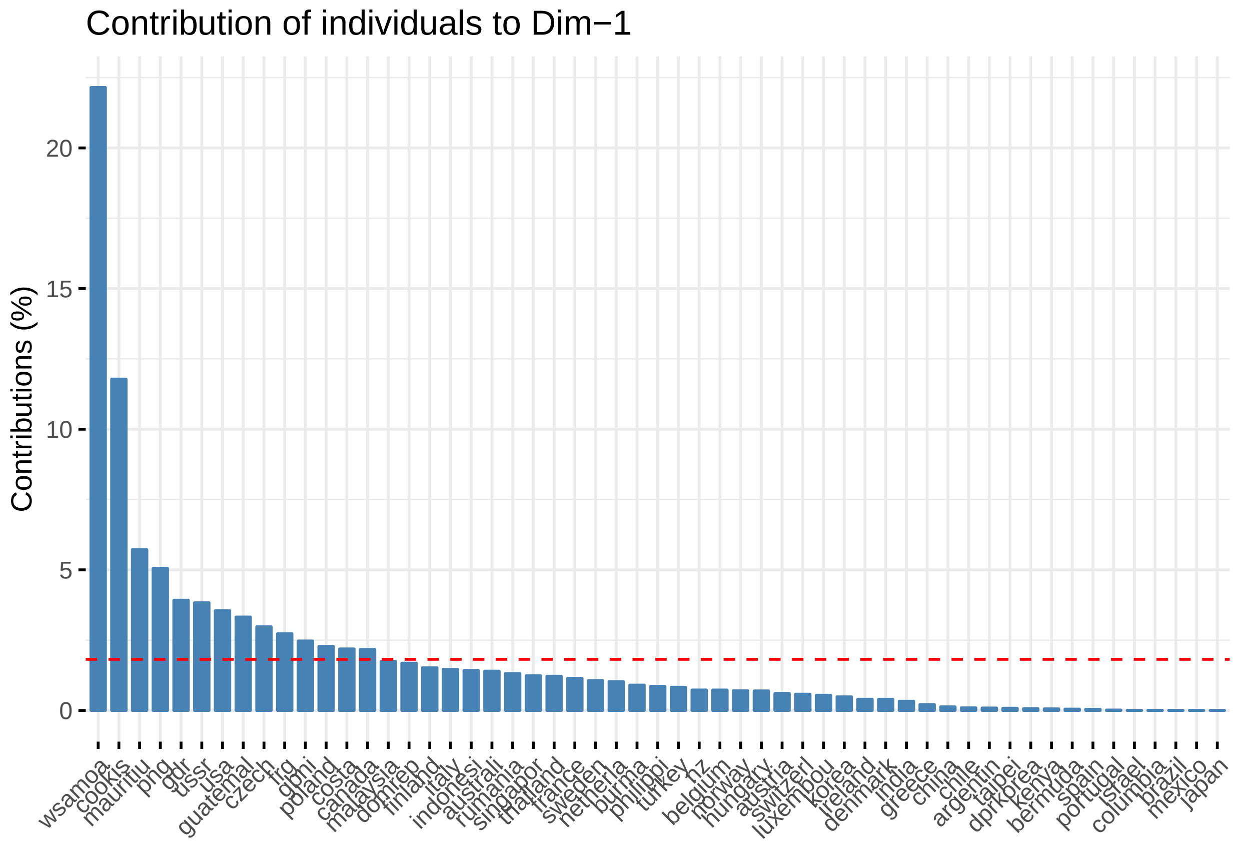
```
# Contributions of variables to PC1
fviz_contrib(pca.out, choice = "var", axes = 1, top = 7)
```



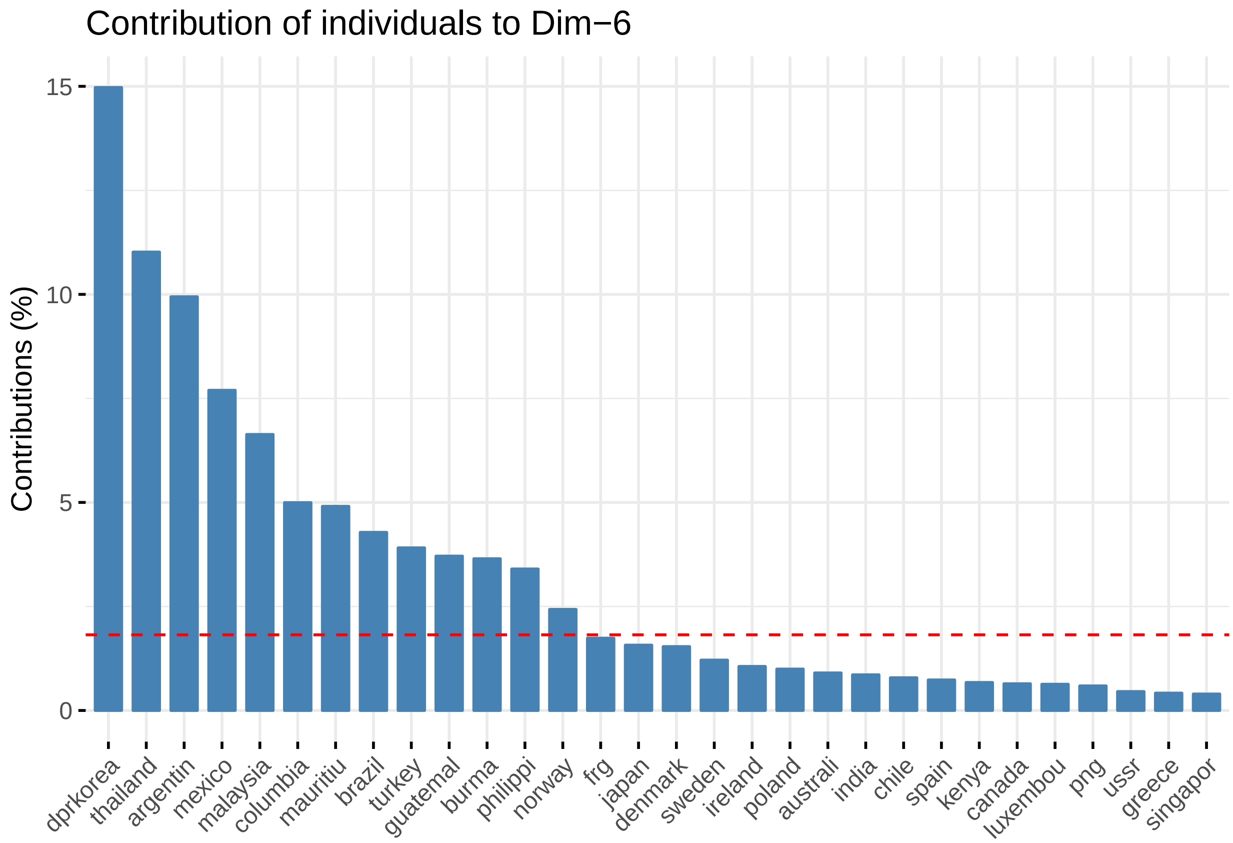
```
# Contributions of variables to PC2
fviz_contrib(pca.out, choice = "var", axes = 2, top = 7)
```



```
#contribution to first 2 principle components
fviz_contrib(pca.out, choice = "ind", axes = 1)
```

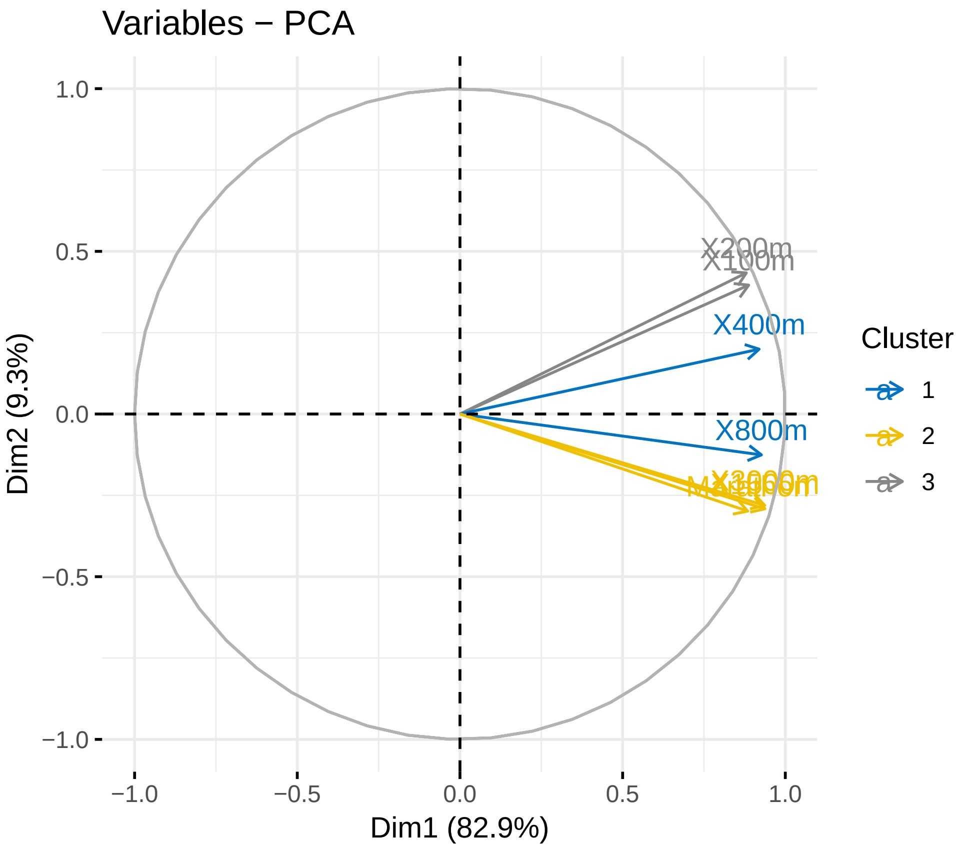


```
fviz_contrib(pca.out, choice = "ind", axes = 6, top = 30)
```



# Variable contribution plot We can also look at the contribution by each variable

```
# Create a grouping variable using kmeans
# Create 3 groups of variables (centers = 3)
set.seed(123)
res.km <- kmeans(var$coord, centers = 3, nstart = 25)
grp <- as.factor(res.km$cluster)
# Color variables by groups
fviz_pca_var(pca.out, col.var = grp,
             palette = c("#0073C2FF", "#EFC000FF", "#868686FF"),
             legend.title = "Cluster")
```


Variables − PCA
1.0
0.5
0.0
−0.5
−1.0
Dim2 (9.3%)
−1.0
−0.5
0.0
0.5
1.0
Dim1 (82.9%)
X200m
X100m
X400m
X800m
X3000m
X1500m
Marathon
Cluster
1
2
3