Multivariate Statistics Lab 1

Carles Sans fuentes, Joshua Hudson, Karolina Ziomek

21 de noviembre de 2017

R Markdown

Question 1: Describing individual variables

Consider the data set in the T1-9.data file, National track records for women. for 54 different countries we have the national records for 7 variables (100, 200, 400, 800, 1500, 3000m and marathon). Use R to do the following analyses.

Here I write the code to process preliminary the data

```
link <- "C:/Users/Carles/Desktop/MasterStatistics-MachineLearning/Master_subjects/Multivariate_Statisti
data <- t(read.table(link))
colnames(data) <- c(data[1, ])
data <- data[2:nrow(data), ]
## Preparing data
mydata <- apply(data, 2, as.numeric)
mydata <- t(mydata)
colnames(mydata) <- c("hundred", "twohundred", "fourhundred",
    "eighthundred", "1500", "3000", "marathon")
## preview of mydata
mydata <- as.data.frame(mydata)
mydata$hundred <- mydata$hundred/60
mydata$twohundred <- mydata$twohundred/60
mydata$fourhundred <- mydata$fourhundred/60
```

a)Describe the 7 variables with mean values, standard deviations e.t.c.

The mean and the standard deviations of the variance are the following:

```
mymean <- apply(mydata, 2, mean)
mysd <- apply(mydata, 2, sd)
df_table <- data.frame(mean = mymean, sd = mysd)
library(knitr)
kable(df_table, caption = " Means and Standard deviations for all 7 variables")
```

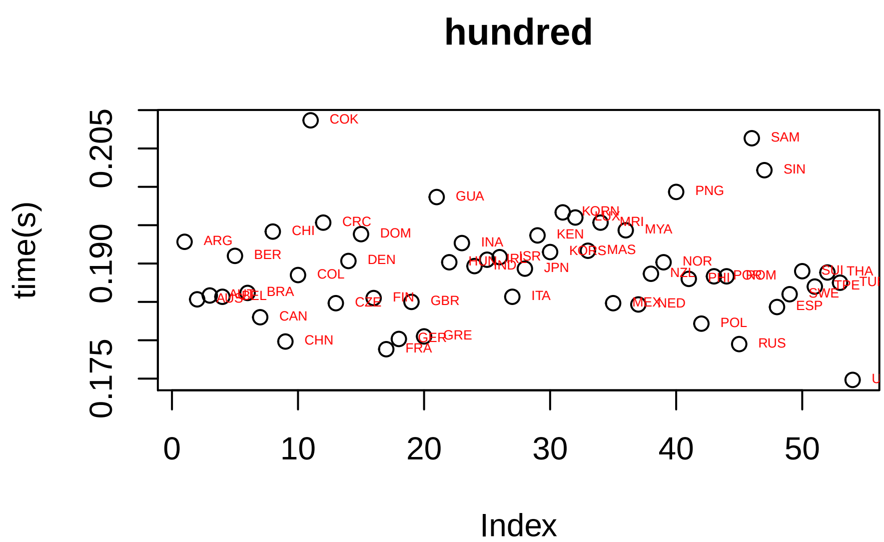
Table 1: Means and Standard deviations for all 7 variables

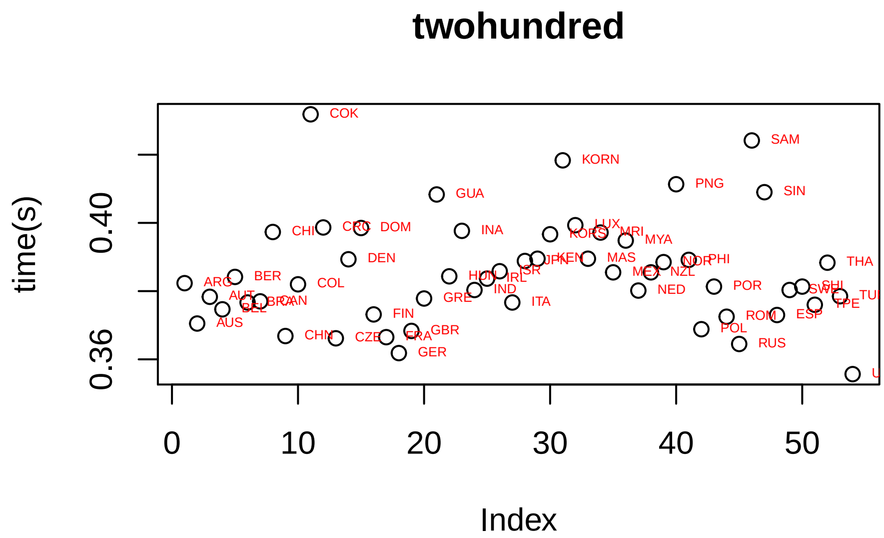
	mean	sd
hundred	0.1892963	0.0065684
twohundred	0.3853086	0.0154838
fourhundred	0.8664846	0.0432867
eighthundred	2.0224074	0.0868730
1500	4.1894444	0.2723650
3000	9.0807407	0.8153269
marathon	153.6192593	16.4398951

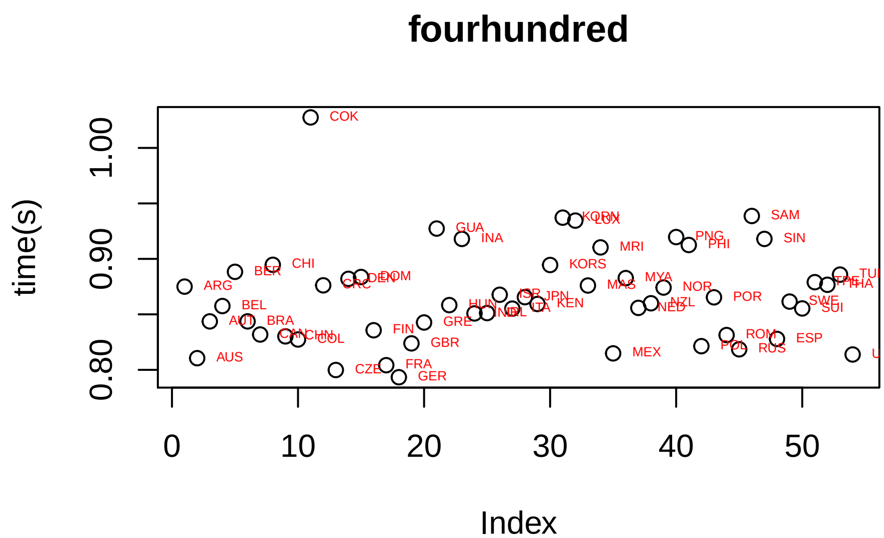
b) Illustrate the variables with different graphs (explore what plotting possibilities R has). Make sure that the graphs look attractive (it is absolutely necessary to look at the labels, font

sizes, point types). Are there any apparent extreme values? Do the variables seem normally distributed? Plot the best fitting (match the mean and standard deviation, i.e. method of moments) Gaussian density curve on the data's histogram. for the last part you may be interested in the hist() and density() functions.

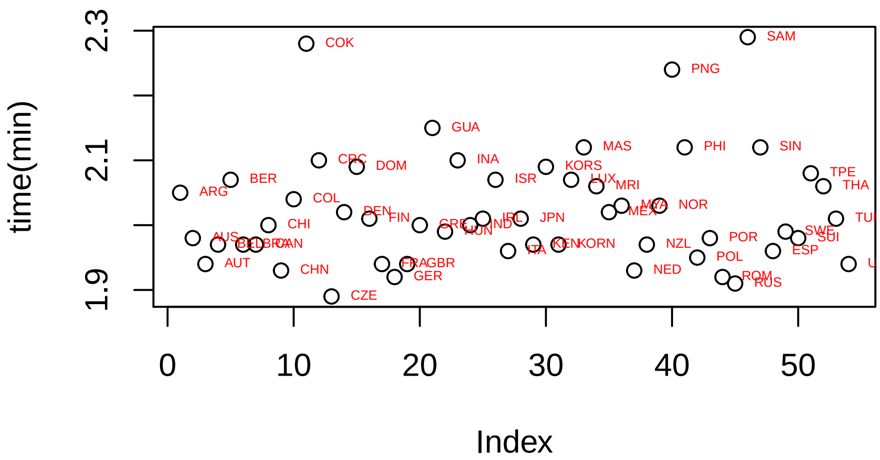
```
track <- mydata
par(mfrow = c(1, 1))
for (i in 1:dim(track)[2]) {
    if (i <= 3) {
        time = "time(s)"
    } else {
        time = "time(min)"
    }
    plot(track[, i], main = colnames(track)[i], ylab = time)
    text(track[, i], rownames(track), cex = 0.4, pos = 4, col = "red")
}
```



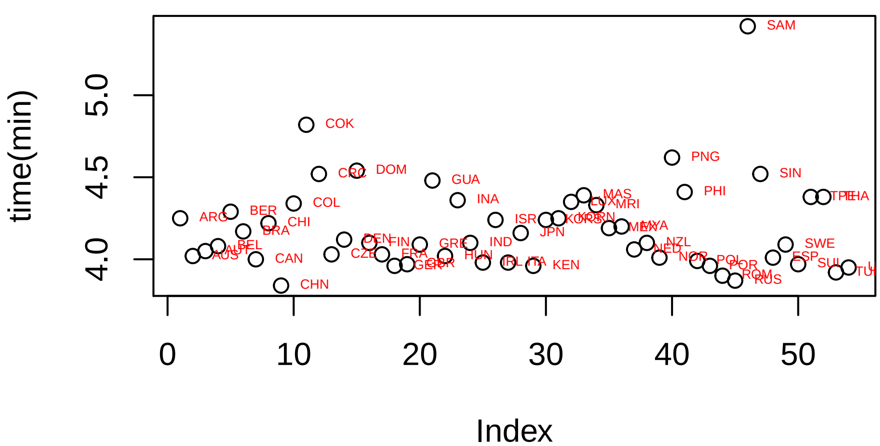

twohundred
time(s)
0.36
0.40
Index
0
10
20
30
40
50

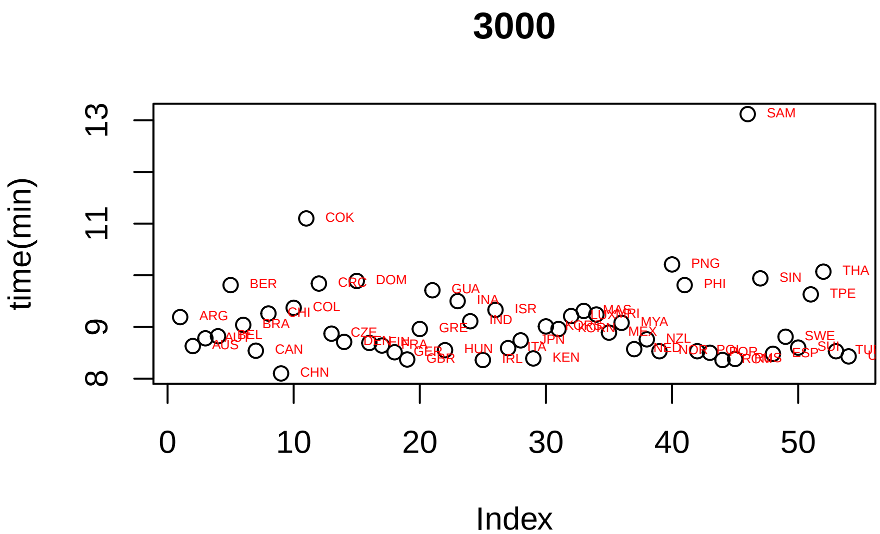

fourhundred
time(s)
0.80
0.90
1.00
Index
0
10
20
30
40
50

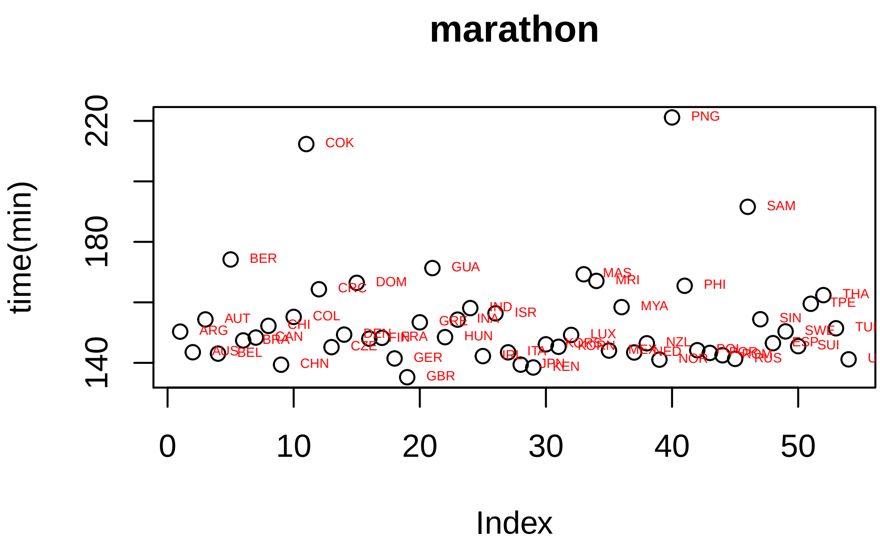
eighthundred



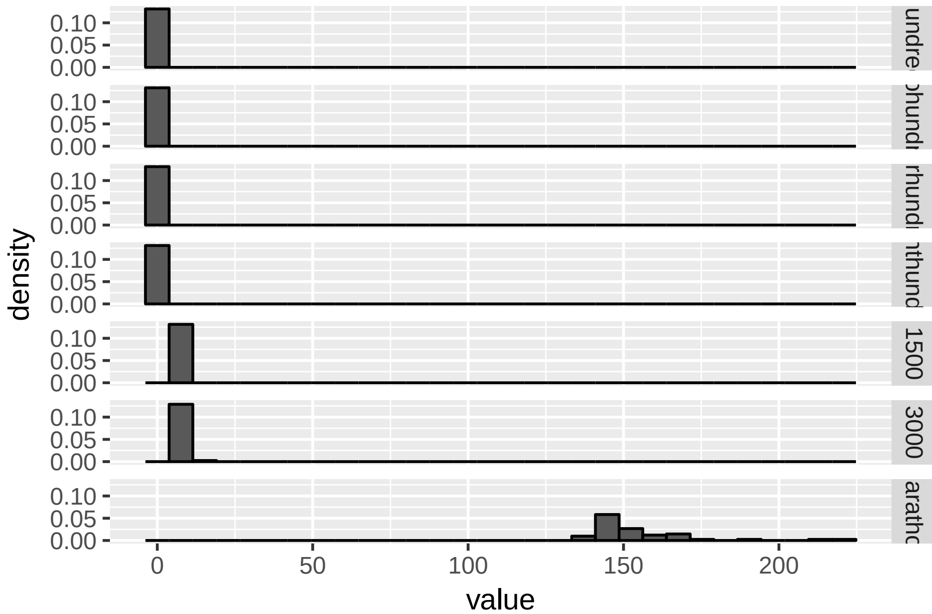
1500





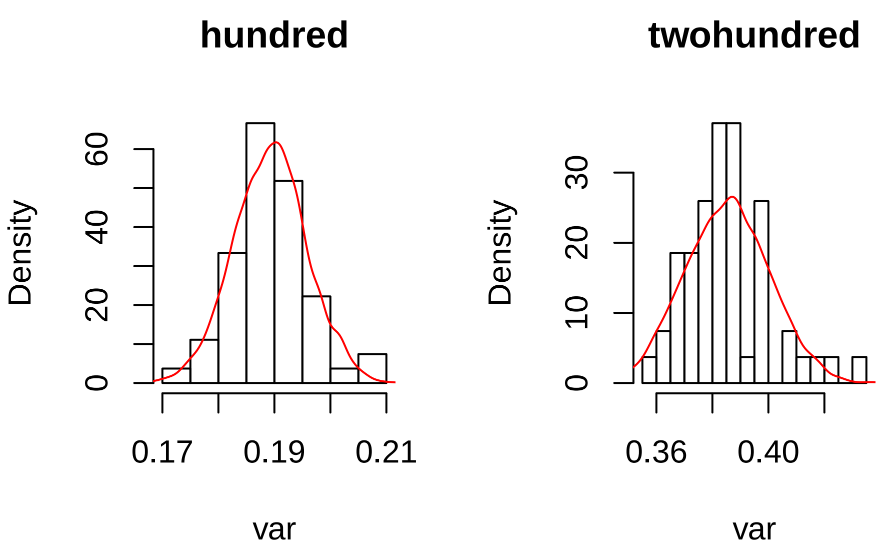


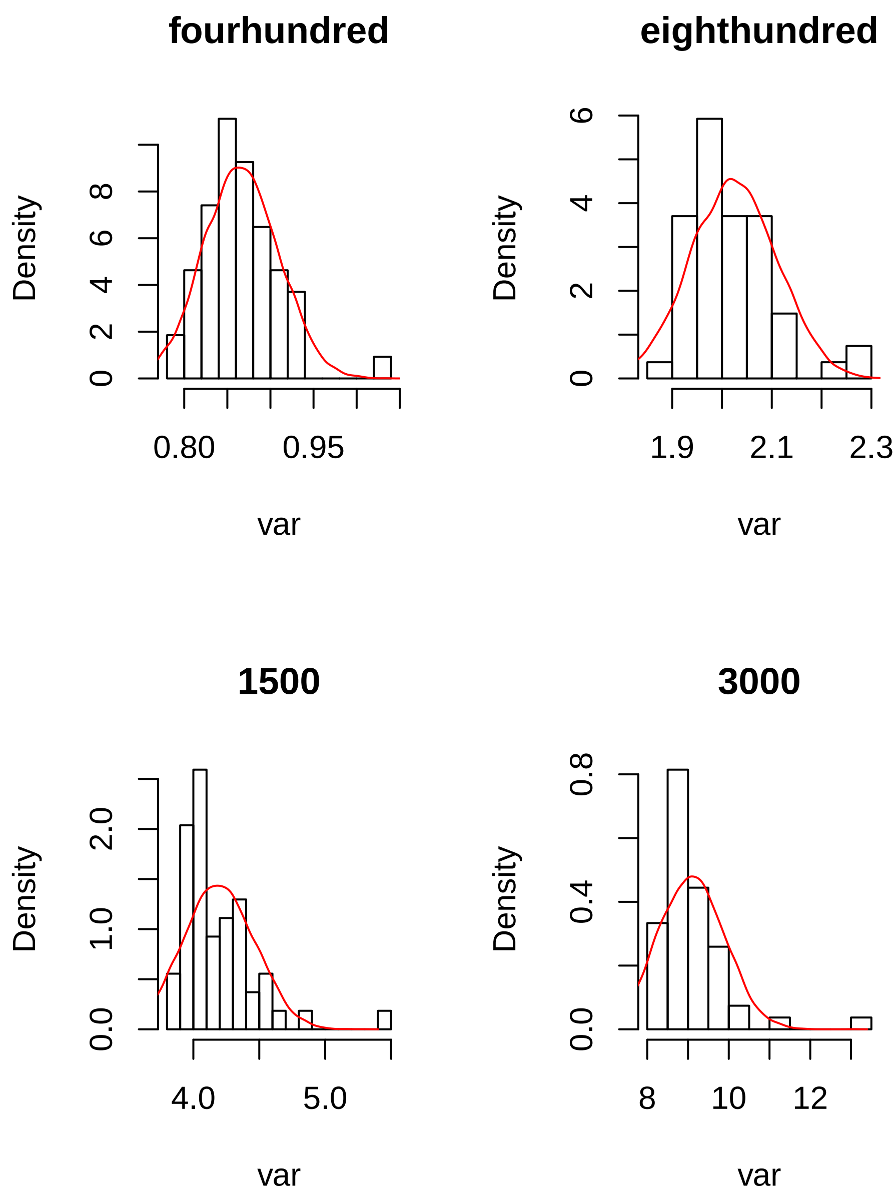

```r
par(mfrow = c(1, 1))
library(reshape2)
data <- mydata
mydata$country = rownames(mydata)
xymelt <- melt(mydata)

library(plotly)

sp <- ggplot(xymelt, aes(x = value)) + geom_histogram(aes(y = ..density..),
    col = "black")

# Divide by levels of 'sex', in the vertical direction
sp + facet_grid(variable ~ ., scales = "free_x")
```

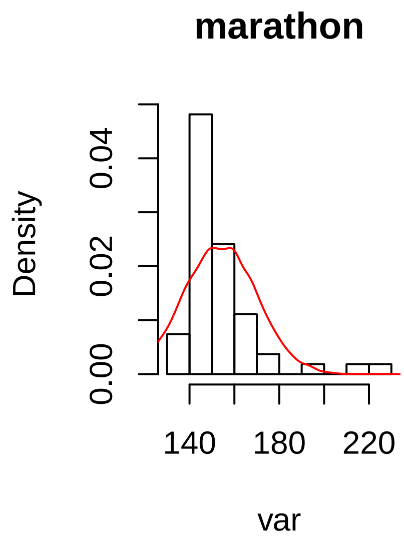


It looks like there exists extreme values in each race distribution.

```
par(mfrow = c(1, 2))
for (i in 1:dim(track)[2]) {
    var <- track[, i]
    hist(var, breaks = 12, freq = FALSE, main = colnames(track)[i])
    lines(density(rnorm(10000, mean(var), sd(var))), col = "red")
}
```




fourhundred
Density
0
2
4
6
8
0.80
0.95
var
eighthundred
Density
0
2
4
6
1.9
2.1
2.3
var
1500
Density
0.0
1.0
2.0
4.0
5.0
var
3000
Density
0.0
0.4
0.8
8
10
12
var



Answer: Given the graphics we have plotted as well as the distributions of each variable separately, we could say that the first races (100, 200 even 400) could be normally distributed but the following races are quite skewed.

Question 2: Relationships between the variables

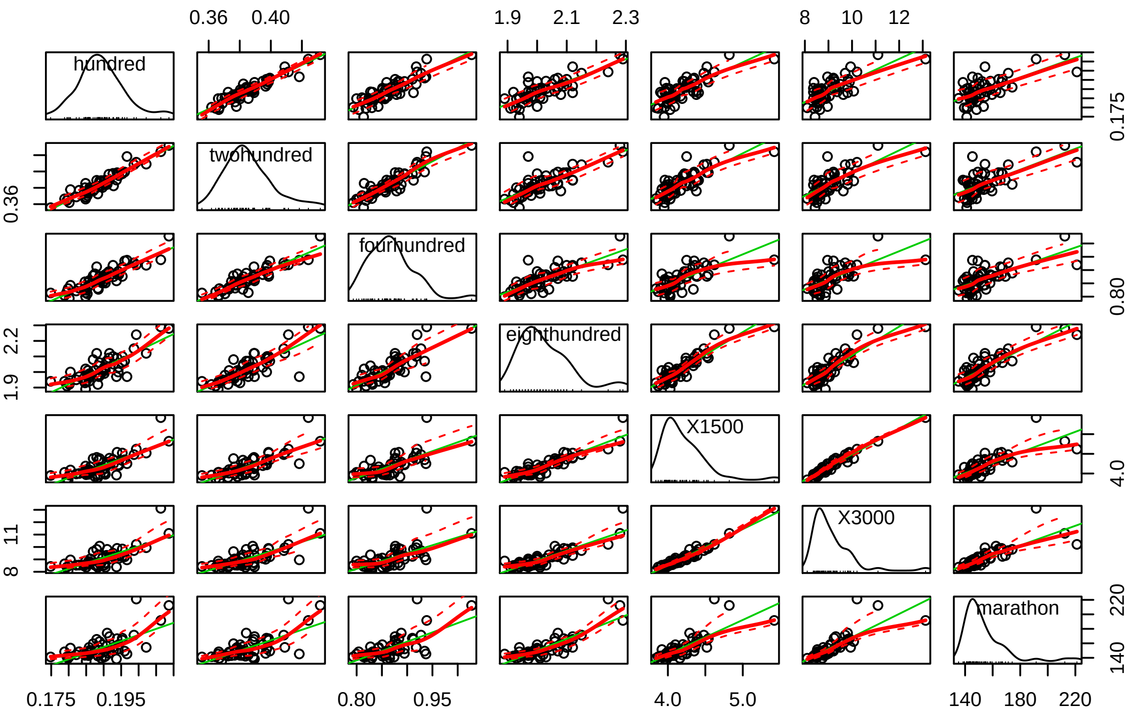
a) Compute the covariance and correlation matrices for the 7 variables. Is there any apparent structure in them? Save these matrices for future use.

```
CovMat <- cov(data)
CorMat <- cor(data)
```

It looks that all of them have quite a high and positive correlation. This makes sense since there exists a relationship between meters and time. It can also be seen that those races where the meters differ the least the correlation is higher, whereas the more distance there is, the correlation is lower but still strong.

b) Generate and study the scatterplots between each pair of variables. Any extreme values?

```
par(mar = c(1, 1, 1, 1))
library(car)
scatterplotMatrix(data)
```

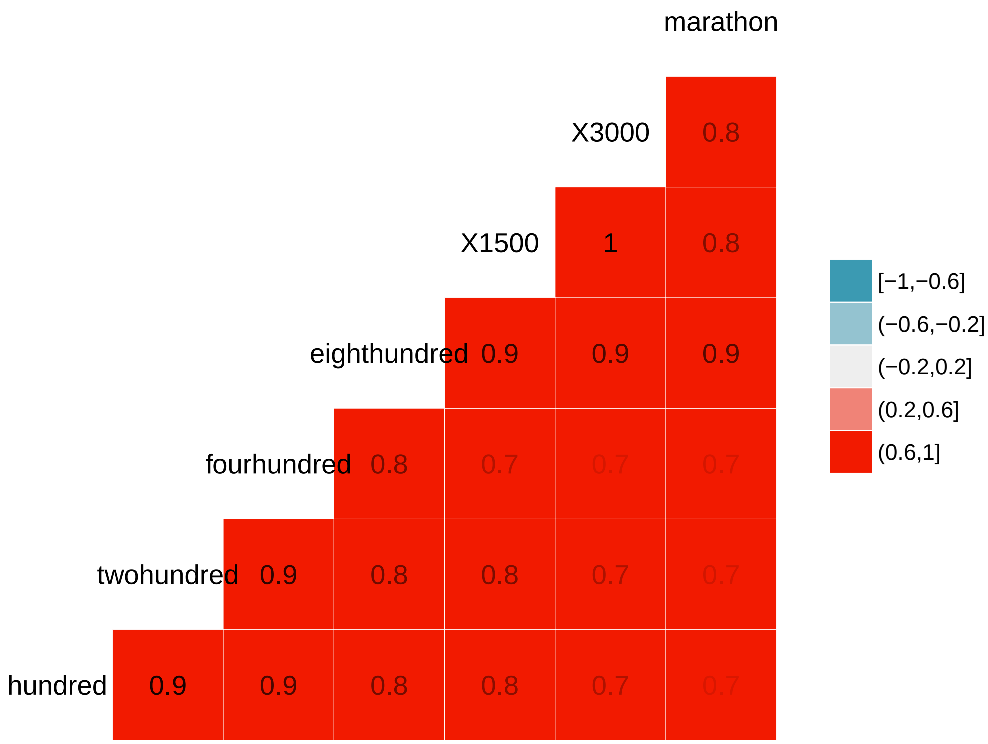


Answer:

Yes, there are extreme values particularly for the longer races and the differences between lenght of races.

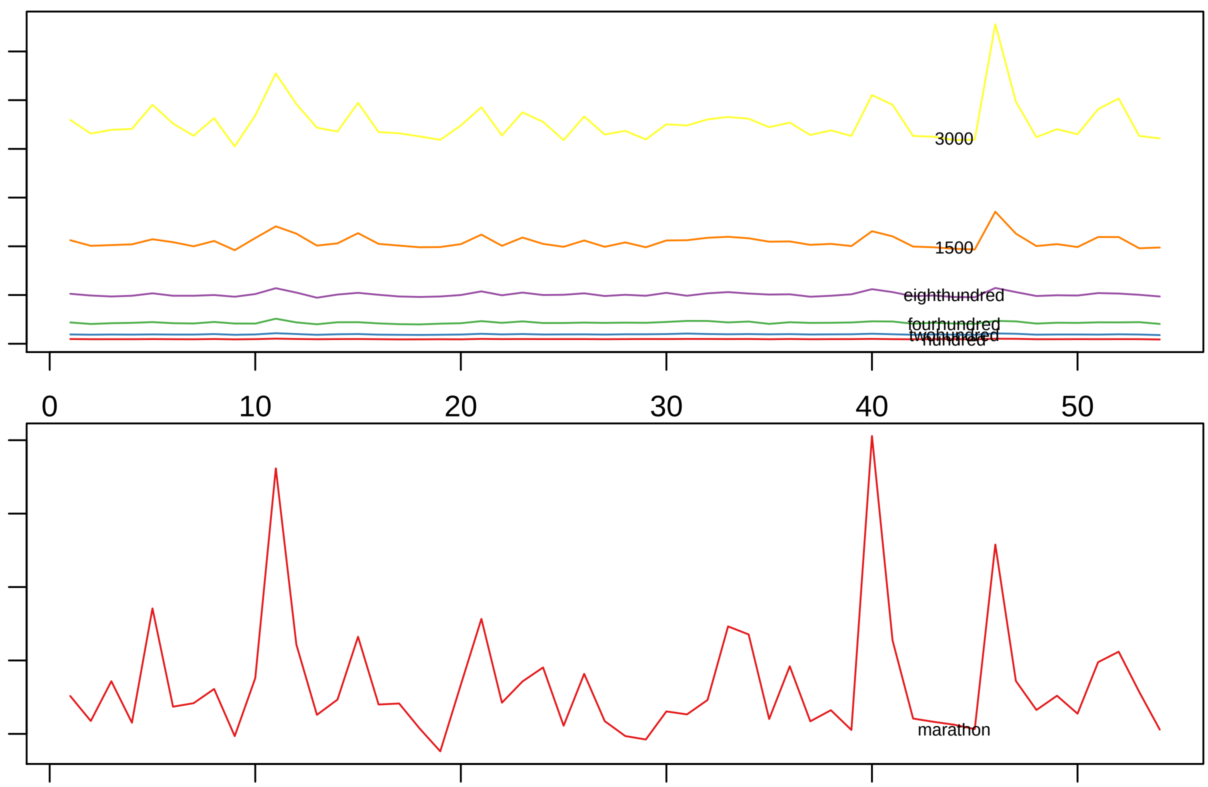
c) Explore what other plotting possibilities R offers for multivariate data. Present other (at least two) graphs that you find interesting with respect to this data set.

```
library(car)
library("GGally")
ggcorr(data, nbreaks = 5, limits = c(0.6, 1), label = TRUE, label_alpha = TRUE)  ##Correlation of data
```

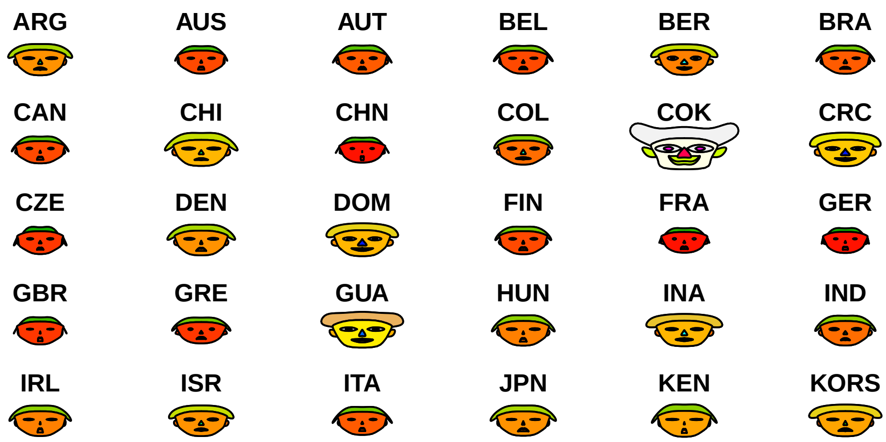



```
par(mar = c(1, 1, 1, 1))
library(ggplot2)
library(scales)
library(RColorBrewer)
par(mfrow = c(1, 1))
library(RColorBrewer)
par(mfrow = c(1, 1))
makeProfilePlot <- function(mylist, names) {
    require(RColorBrewer)
    # find out how many variables we want to include
    numvariables <- length(mylist)
    # choose 'numvariables' random colours
    colours <- brewer.pal(numvariables, "Set1")
    # find out the minimum and maximum values of the variables:
    mymin <- 1e+20
    mymax <- 1e-20
    for (i in 1:numvariables) {
        vectori <- mylist[[i]]
        mini <- min(vectori)
        maxi <- max(vectori)
        if (mini < mymin) {
            mymin <- mini
        }
        if (maxi > mymax) {
            mymax <- maxi
        }
    }
```

```
    # plot the variables
    for (i in 1:numvariables) {
        vectori <- mylist[[i]]
        namei <- names[i]
        colouri <- colours[i]
        if (i == 1) {
            plot(vectori, col = colouri, type = "l", ylim = c(mymin,
                mymax), xlab = "index of the countries", ylab = "minutes of each proof")
        } else {
            points(vectori, col = colouri, type = "l")
        }
        lastxval <- length(vectori)
        lastyval <- vectori[length(vectori)]
        text((lastxval - 10), (lastyval), namei, col = "black",
            cex = 0.6)

    }
}
names <- colnames(data)
mylist <- list(mydata$hundred, mydata$twohundred, mydata$fourhundred,
    mydata$eighthundred, mydata$`1500`, mydata$`3000`, mydata$marathon)
par(mfrow = c(2, 1))
makeProfilePlot(mylist[1:(length(mylist) - 1)], names[1:(length(mylist) -
    1)])
makeProfilePlot(mylist[length(mylist)], names[length(mylist)])
```

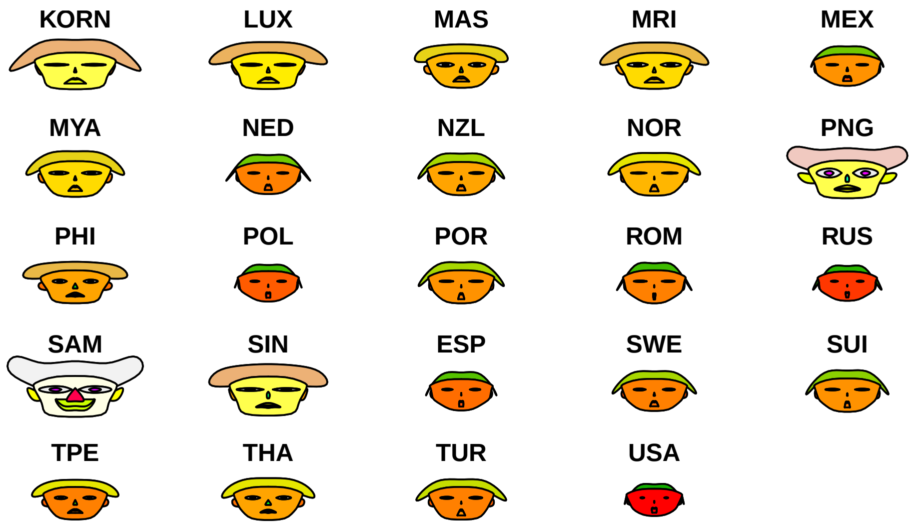


```r
############## Chernoff faces
library(aplpack)
faces(track[1:30, ])
```




```
## effect of variables:
##  modified item       Var
##  "height of face   " "hundred"
##  "width of face    " "twohundred"
##  "structure of face" "fourhundred"
##  "height of mouth  " "eighthundred"
##  "width of mouth   " "1500"
##  "smiling          " "3000"
##  "height of eyes   " "marathon"
##  "width of eyes    " "hundred"
##  "height of hair   " "twohundred"
##  "width of hair   "  "fourhundred"
##  "style of hair   "  "eighthundred"
##  "height of nose  "  "1500"
##  "width of nose   "  "3000"
##  "width of ear    "  "marathon"
##  "height of ear   "  "hundred"
```

```r
faces(track[31:54, ])
```




```
## effect of variables:
##  modified item       Var
##  "height of face   " "hundred"
##  "width of face    " "twohundred"
##  "structure of face" "fourhundred"
##  "height of mouth  " "eighthundred"
##  "width of mouth   " "1500"
##  "smiling          " "3000"
##  "height of eyes   " "marathon"
##  "width of eyes    " "hundred"
##  "height of hair   " "twohundred"
##  "width of hair   "  "fourhundred"
##  "style of hair   "  "eighthundred"
##  "height of nose  "  "1500"
##  "width of nose   "  "3000"
##  "width of ear    "  "marathon"
##  "height of ear   "  "hundred"
```

Question 3: Examining for extreme values

a) Look at the plots (esp. scatterplots) generated in the previous question. Which 3-4 countries appear most extreme? Why do you consider them extreme? Answer:

The most extremes seemed to be SAM, COK and PNG because visibly they are far away from all the other countries.

One approach to measuring "extremism" is to look at the distance (needs to be defined!) between an observation and the sample mean vector, i.e. we look how far one is from the

average. Such a distance can be called an multivariate residual for the given observation.

```r
S <- apply(data, 2, FUN = function(x) {
    x - mean(x)
})
```

b) The most common residual is the Euclidean distance between the observation and sample mean vector, i.e.

$$d(\vec{x}, \hat{x}) = \sqrt{(\vec{x} - \hat{x})^T(\vec{x} - \hat{x})}$$

This distance can be immediately generalized to the Lr, r > 0 distance as

$$d_L(\vec{x}, \hat{x}) = (\sum_{i=1}^{p} |\vec{x} - \hat{x}|^r)^{1/r}$$

where p is the dimension of the observation (here p = 7). Compute the squared Euclidean distance (i.e. r = 2) of the observation from the sample mean for all 54 countries using R's matrix operations. first center the raw data by the means to get

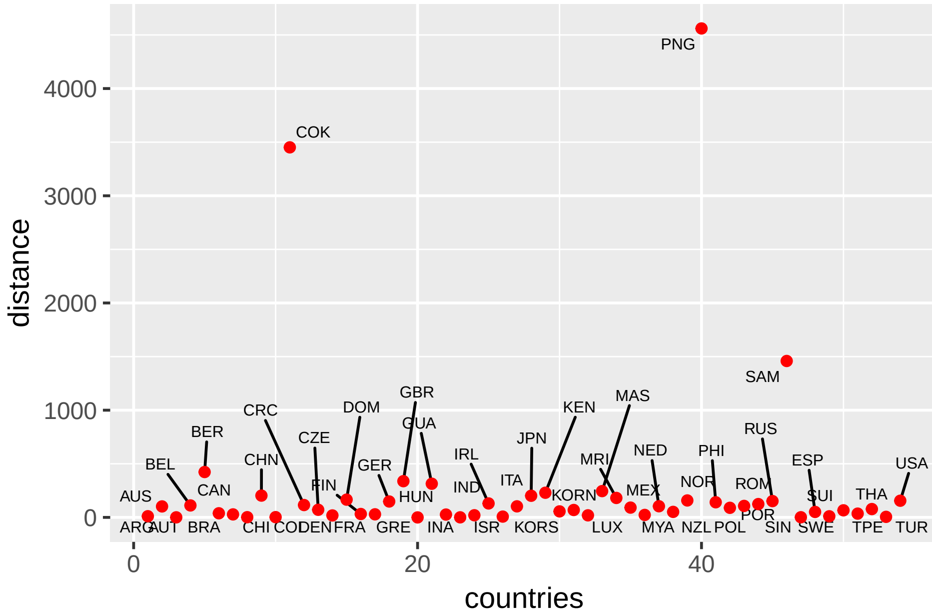
$$\vec{x} - \hat{x}$$

for each country. Then do a calculation with matrices that will result in a matrix that has on its diagonal the requested squared distance for each country. Copy this diagonal to a vector and report on the most extreme countries. In this questions you MAY NOT use any loops.

```r
library("ggrepel")
countrDist <- tcrossprod(S)

countries <- diag(countrDist)

sort(countries, decreasing = TRUE)[1:10]
```

```
##       PNG       COK       SAM       BER       GBR       GUA       MAS
## 4560.5620 3451.5209 1458.9263  423.2887  337.9919  314.1713  245.3612
##       KEN       CHN       JPN
##  230.0327  203.5661  202.0202
```

```r
ggplot(data = as.data.frame(countries), aes(y = countries, x = 1:length(countries))) +
    xlab("countries") + ylab("distance") + geom_text_repel(label = colnames(countrDist),
    size = 2) + geom_point(color = "red")
```



The top 3 are the ones we visibly noticed before.

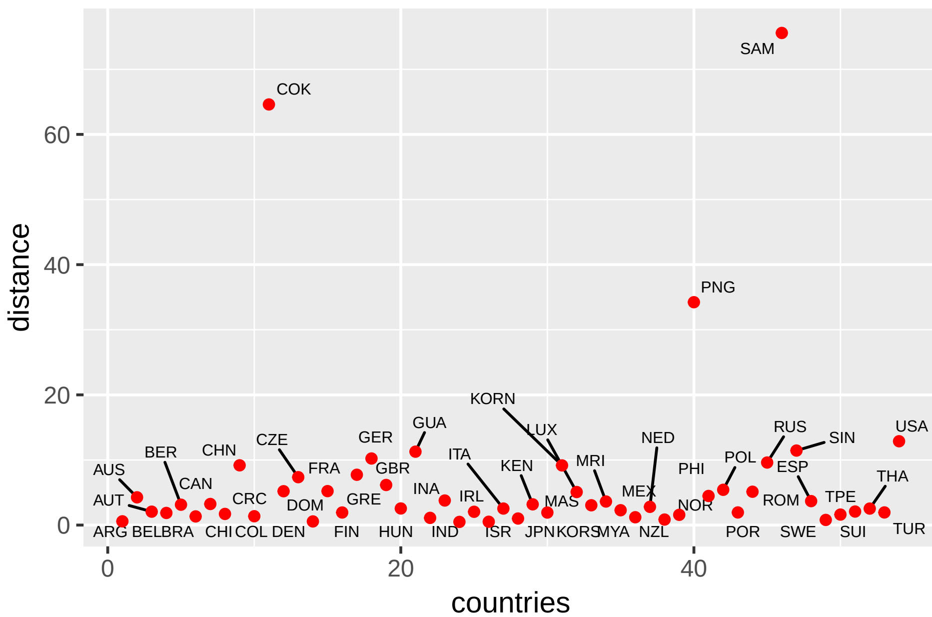
c) The different variables have different scales so it is possible that the distances can be dominated by some few variables. To avoid this we can use the squared distance

$$d_V^2(\vec{x}, \hat{x}) = (\vec{x} - \hat{x})^T V^{-1} (\vec{x} - \hat{x})$$

```
newmat <- apply(data, 2, FUN = function(x) {
    (abs(x - mean(x)))/sd(x)
})

countrDist2 <- (tcrossprod(newmat))

countries2 <- diag(countrDist2)
sort(countries2, decreasing = TRUE)[1:10]
```

```
##       SAM       COK       PNG       USA       SIN       GUA       GER
## 75.582802 64.601160 34.228907 12.876894 11.444864 11.273864 10.223646
##       RUS       CHN      KORN
##  9.608420  9.176893  9.165266
```

```
ggplot(data = as.data.frame(countries2), aes(y = countries2,
    x = 1:length(countries2))) + xlab("countries") + ylab("distance") +
    geom_text_repel(label = colnames(countrDist2), size = 2) +
    geom_point(color = "red")
```



where V is a diagonal matrix with variances of the appropriate variables on the diagonal. The effect, is that for each variable the squared distance is divided by its variance and we have a scaled independent distance. It is simple to compute this measure by standardizing the raw data with both means (centring) and standard deviations (scaling), and then compute the Euclidean distance for the normalized data. Carry out these computations and conclude which countries are the most extreme ones. How do your conclusions compare with the unnormalized ones?

The conclusions are that the top 3 is still the same changing the two next ones. The normalized distance is a better measure since everything is compared on the same scale. For that, we could also say that the second result is more accurate.

d) The most common statistical distance is the Mahalanobis distance

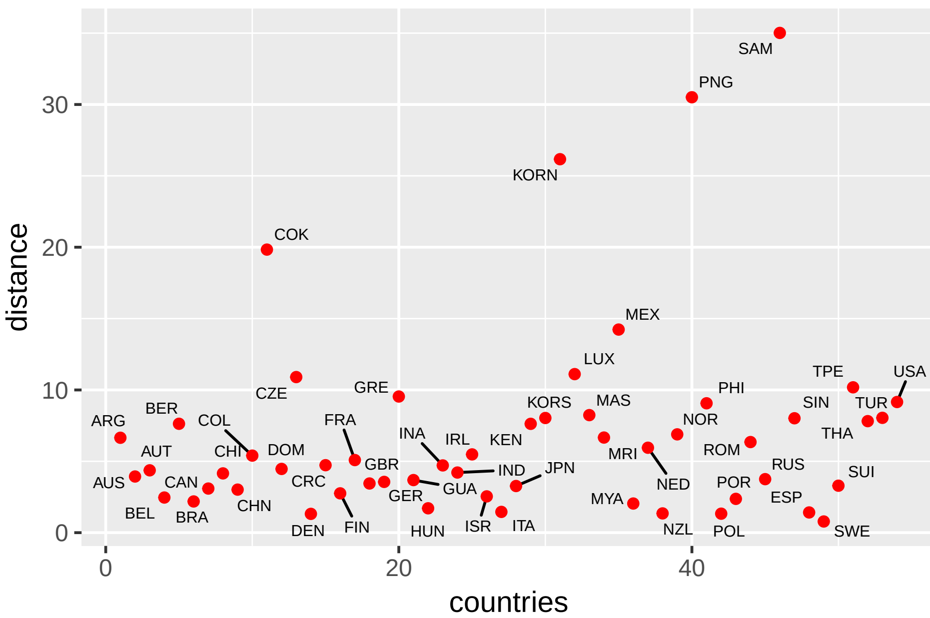
$$d_M^2(\vec{x}, \hat{x}) = (\vec{x} - \hat{x})^T C^{-1} (\vec{x} - \hat{x})$$

where C is the sample covariance matrix calculated from the data. With this measure we also use the relationships (covariances) between the variables (and not only the marginal variances as dV(· , ·) does). Compute the Mahalanobis distance, which countries are most extreme now?

```
dmahal <- S %*% solve(as.matrix(CovMat)) %*% t(S)


countries3 <- diag(dmahal)
sort(countries3, decreasing = TRUE)[1:10]
```

```
##       SAM       PNG      KORN       COK       MEX       LUX       CZE
## 35.014063 30.507248 26.167141 19.834001 14.230932 11.108846 10.901456
##       TPE       GRE       USA
## 10.183996  9.540322  9.155697
```

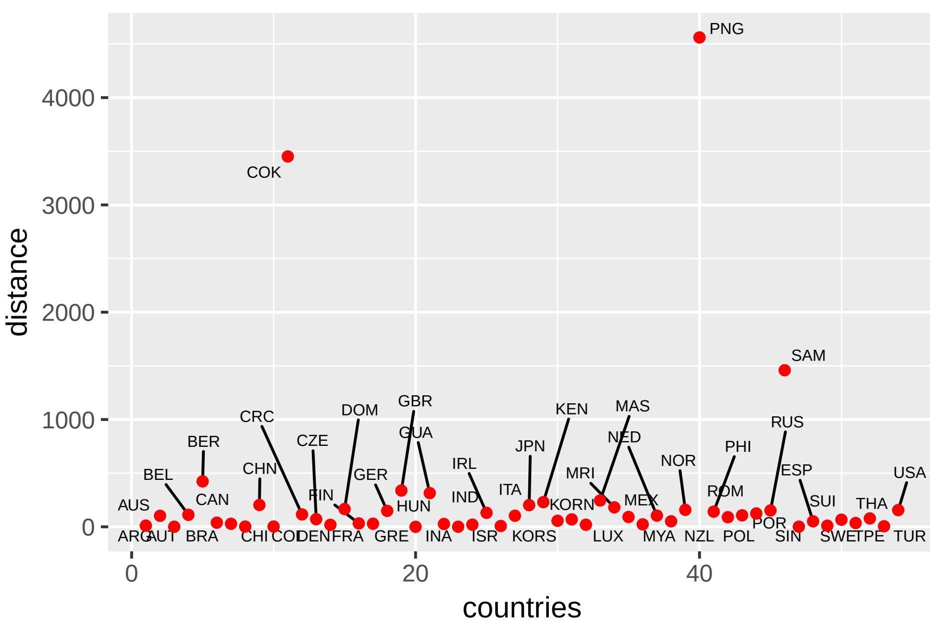
```
ggplot(data = as.data.frame(countries3), aes(y = countries3,
    x = 1:length(countries2))) + xlab("countries") + ylab("distance") +
    geom_text_repel(label = colnames(countrDist2), size = 2) +
    geom_point(color = "red")
```



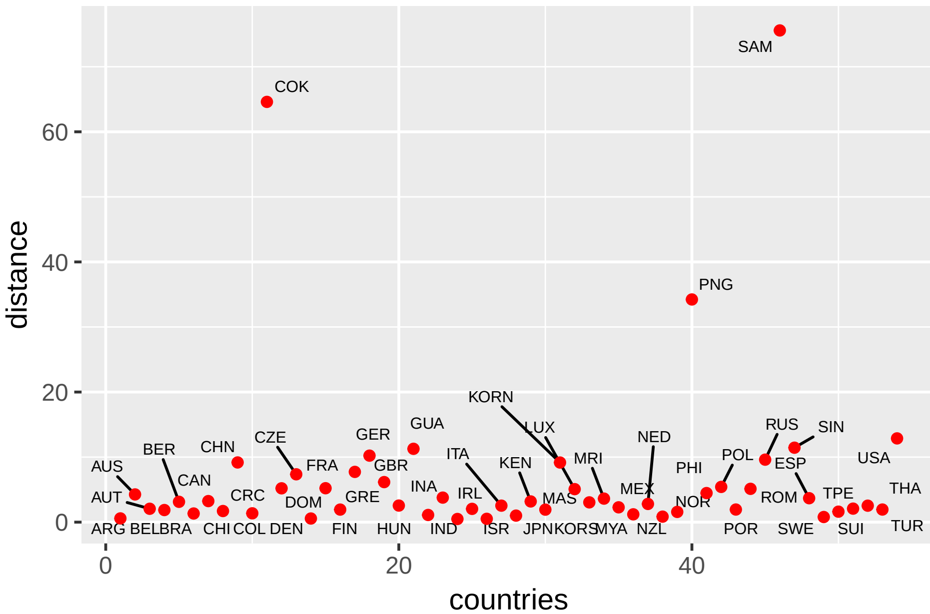
The top 10 countries can be seen above.

e) Compare the results in b)-d). Some of the countries are in the upper end with all the measures and perhaps they can be classified as extreme. Discuss this. But also notice the different measures give rather different results (how does Sweden behave?). Summarize this graphically.

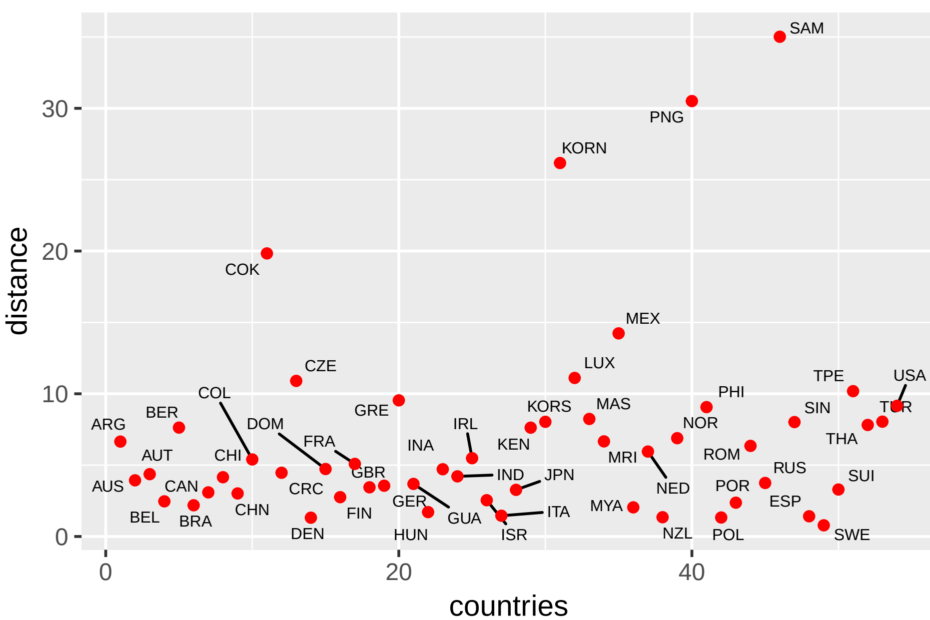
```
library("ggrepel")
par(mfrow = c(1, 3))
ggplot(data = as.data.frame(countries), aes(y = countries, x = 1:length(countries))) +
    xlab("countries") + ylab("distance") + geom_text_repel(label = colnames(countrDist),
    size = 2) + geom_point(color = "red")
```



```r
ggplot(data = as.data.frame(countries2), aes(y = countries2,
    x = 1:length(countries))) + xlab("countries") + ylab("distance") +
    geom_text_repel(label = colnames(countrDist), size = 2) +
    geom_point(color = "red")
```




```r
ggplot(data = as.data.frame(countries3), aes(y = countries3,
    x = 1:length(countries))) + xlab("countries") + ylab("distance") +
    geom_text_repel(label = colnames(countrDist), size = 2) +
    geom_point(color = "red")
```



Answer: We can notice that the top5 of all distances includes SAM COK and PNG, which were the ones we notices in the a part if this question, but the other countries appeared in different places depending on the distance measure. For that, different measures should be evaluated and then choose the ones that best apply for each case.