Assignment 1: Examining multivariate data

Group 12

Dávid Hrabovszki-davhr856, Laura Julia Melis-lauju103, Spyridon Dimitriadis-spydi472, Vasileia Kampouraki-vaska979

22/11/2019

Question 1: Describing individual variables

1.a

All 7 variables are numeric continuous. We notice that 100m, 200m and 400m are in seconds, while 800m, 1500m, 3000 and marathon are in minutes.

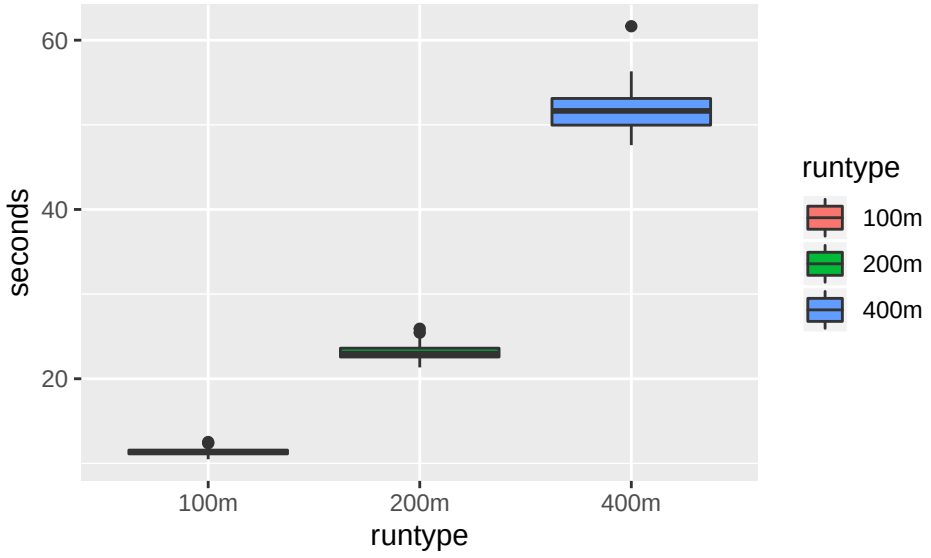
```
##             100m   200m   400m  800m 1500m  3000m marathon
## mean      11.358 23.119 51.989 2.022 4.189  9.081  153.619
## median    11.325 22.980 51.645 2.005 4.100  8.845  148.430
## mode      11.140 22.600 50.620 1.970 4.100  8.530  150.320
## min       10.490 21.340 47.600 1.890 3.840  8.100  135.250
## max       12.520 25.910 61.650 2.290 5.420 13.120  221.140
## range      2.030  4.570 14.050 0.400 1.580  5.020   85.890
## sd         0.394  0.929  2.597 0.087 0.272  0.815   16.440
## skewness   0.590  0.809  0.943 1.182 1.943  2.537    2.312
## kurtosis   0.719  0.661  1.766 1.418 5.775  9.198    6.039
```

1.b

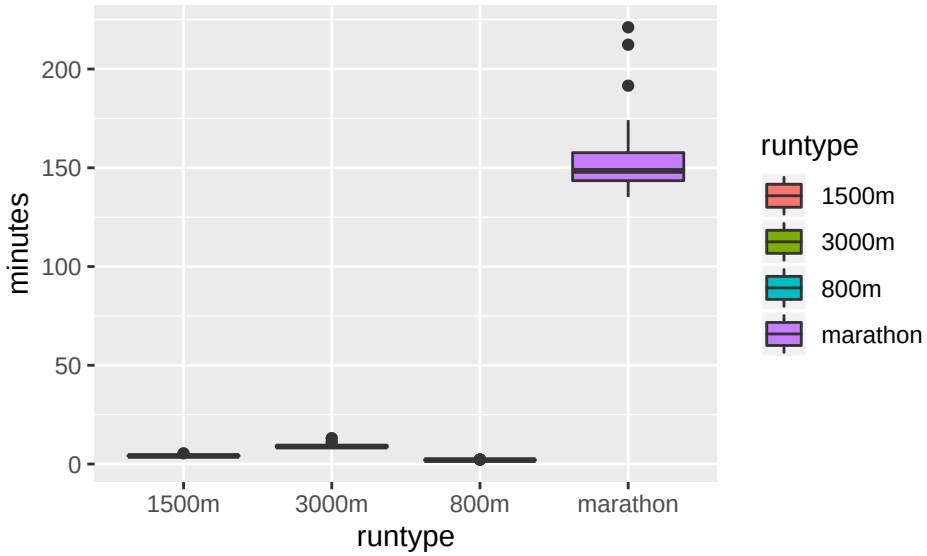
Extreme values

Box plot

Boxplot is a visualization of the median,quartiles,maximum and minimum of a data set. The individual points represent the outliers. Our criteria to classify the outliers are the dots that are bellow the first quartile and above the fourth quartile.



We obviously see one outlier for the 400m and probably 1 or more outliers for the 100m and 200m but it's not that clear in those cases.

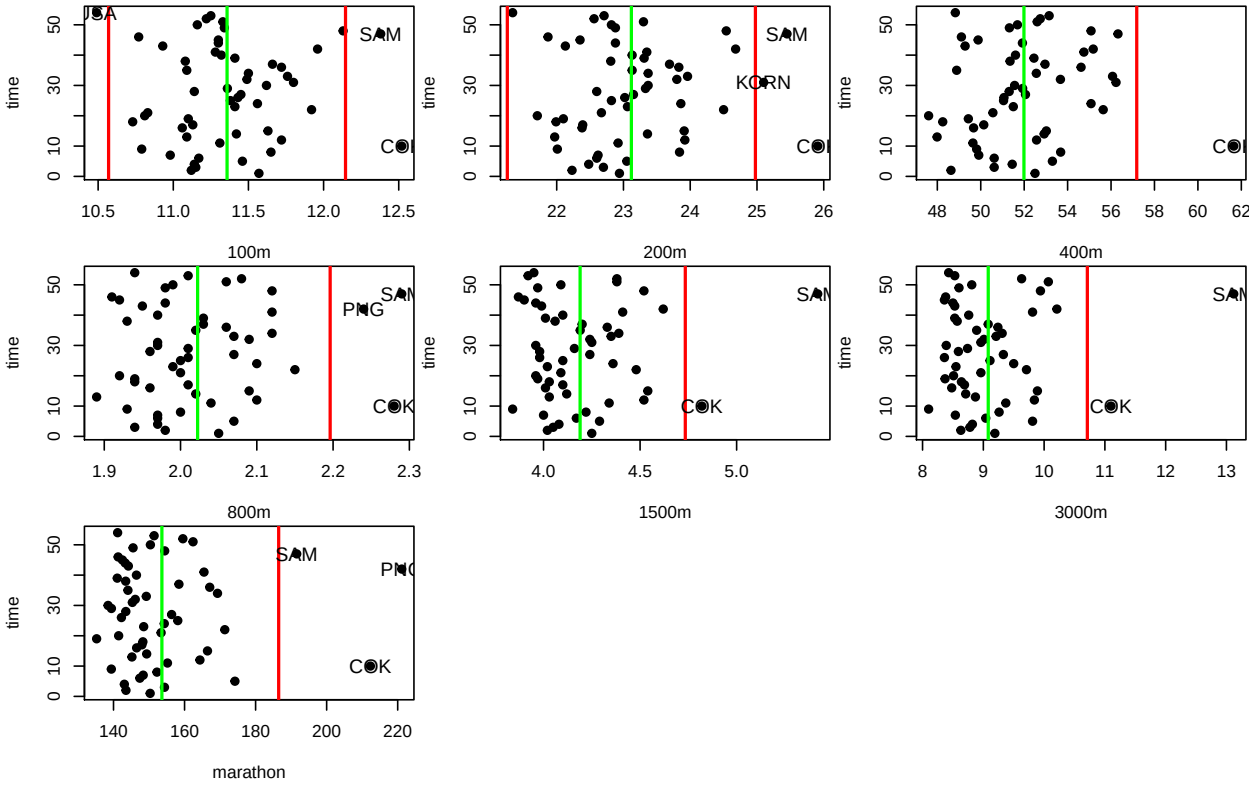


In terms of boxplots (i.e. out of the forth quatile) marathon has 3 outliers and is spread. We can see that there are outliers also for the 1500m,3000m and 800m but the image of their boxplot is not so clear for more commenting, so we will investigate them further in the next step.

Scatterplots with labeled extreme points

In the following plots, green lines represent the mean and red lines represent the lower and upper limit. These limits have been calculated as follows:

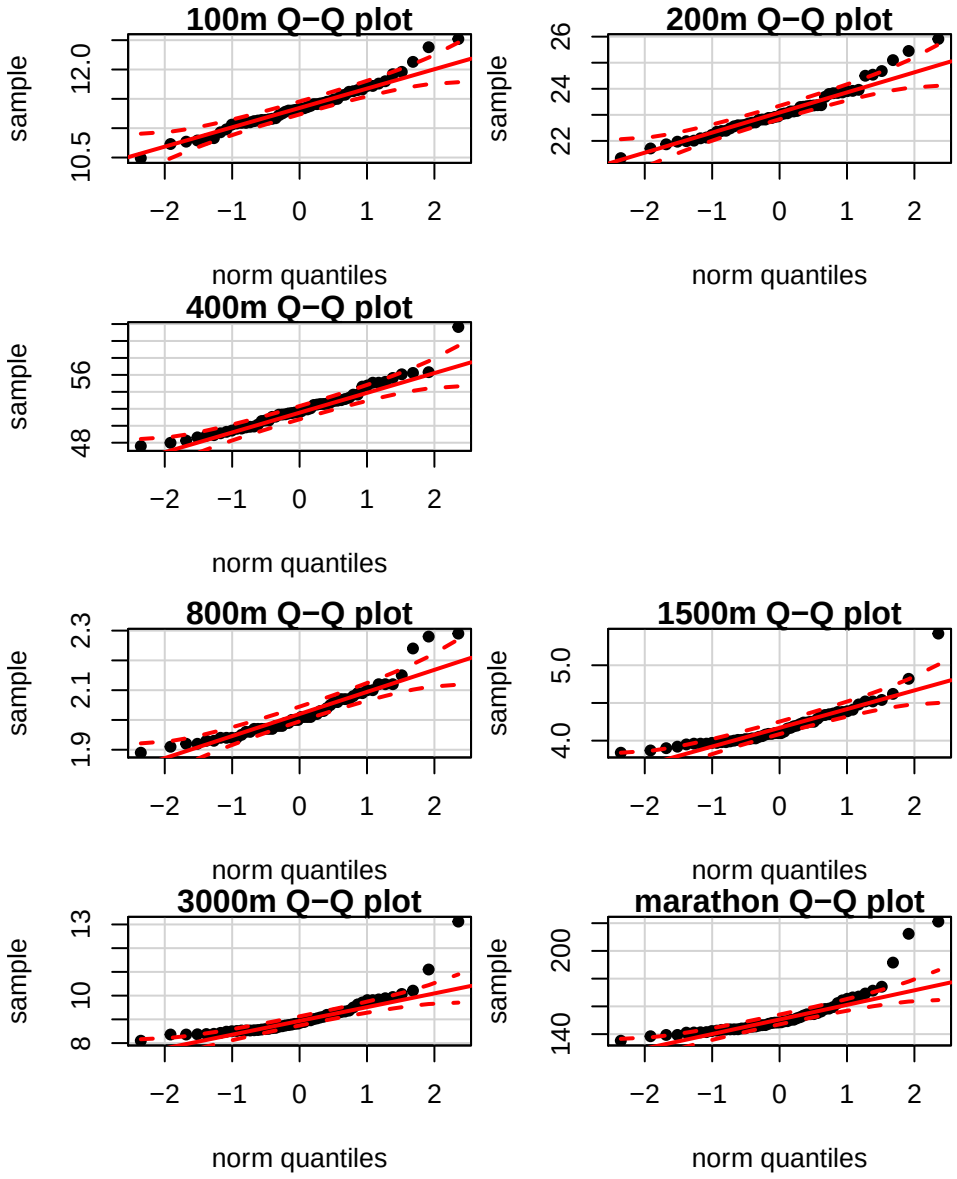
$$\bar{x} \pm 2s \text{ because } \Pr\{\mu - 2\sigma \leq X \leq \mu + 2\sigma\} = 0.9545$$



Looking at the scatterplots we can observe a more clear image of the outliers and also which countries are the outliers. For the case of 100m race we can see one lower outlier (USA) and two above the upper quartile (SAM and COK) and for the case of the marathon we can confirm the three outliers of the upper quartile (SAM, PNG and COK).

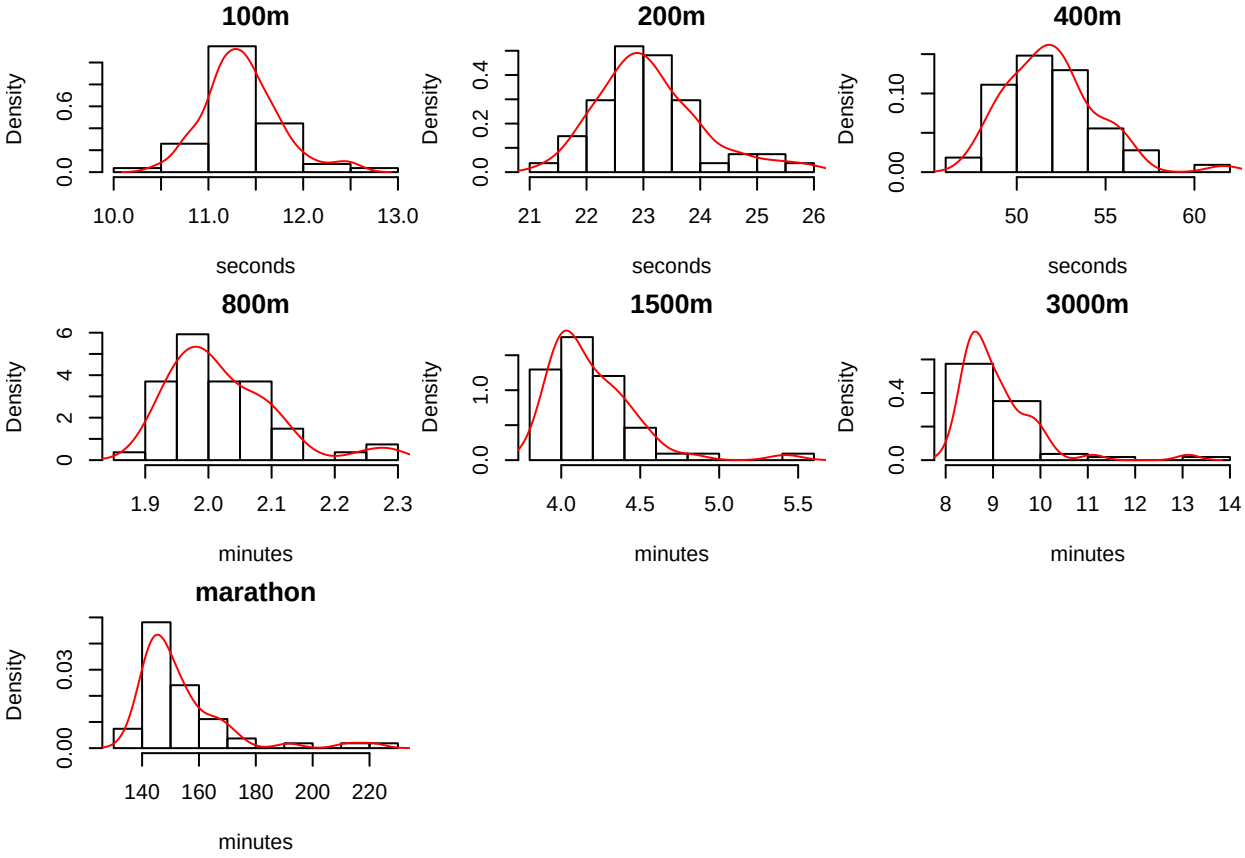
Normally distributed

Q-Q plots



The Normal QQ plot is used to determine if the data are close to being normally distributed. We cannot be sure that the data are normally distributed but we can say in some cases that they aren't. The closer all points lie to the line, the closer the distribution of the sample comes to the normal distribution. Furthermore, the red line should be almost y=x (i.e. 45 degrees angle from the horizontal line). With 95% confidence level from Quantile-Comparison Plot, we can confirm the outliers as we said from box plots before (we could even specify which observations are outliers by putting the argument id=T in qqPlot). Moreover, exept from the outliers all variables could be normal but a bit skewed. The variables that look more normal are 100m, 200m and 400m.

Gaussian density curve on the data's histogram.



From histograms and densities we can confirm some conclusions from before. The variables '100m', '200m' and '400m' look normal. The variables are '1500m', '3000m' and 'marathon' look skewed and from the table of 1.a we can confirm that their skewed is high; 1.943142, 2.5372744, 2.311548 respectively.

Question 2: Relationships between the variables

2.a

Covariance matrix

```
##           100m   200m   400m  800m 1500m  3000m marathon
## 100m     0.155  0.345  0.891 0.028 0.084  0.234    4.334
## 200m     0.345  0.863  2.193 0.066 0.203  0.554   10.385
## 400m     0.891  2.193  6.745 0.182 0.509  1.427   28.904
## 800m     0.028  0.066  0.182 0.008 0.021  0.061    1.220
## 1500m    0.084  0.203  0.509 0.021 0.074  0.216    3.540
## 3000m    0.234  0.554  1.427 0.061 0.216  0.665   10.706
## marathon 4.334 10.385 28.904 1.220 3.540 10.706  270.270
```

All the covariances between variables are positive, meaning that all pair of variables are positively related. Also, we can observe that the covariances are smaller between those variables representing short races and larger between variables that represent linger races (such as "Marathon"). The highest covariance coefficient is $\text{Cov}(400\text{m}, \text{Marathon}) = 28.9$ and the lowest, $\text{Cov}(800\text{m}, 1500\text{m}) = 0.0214$.

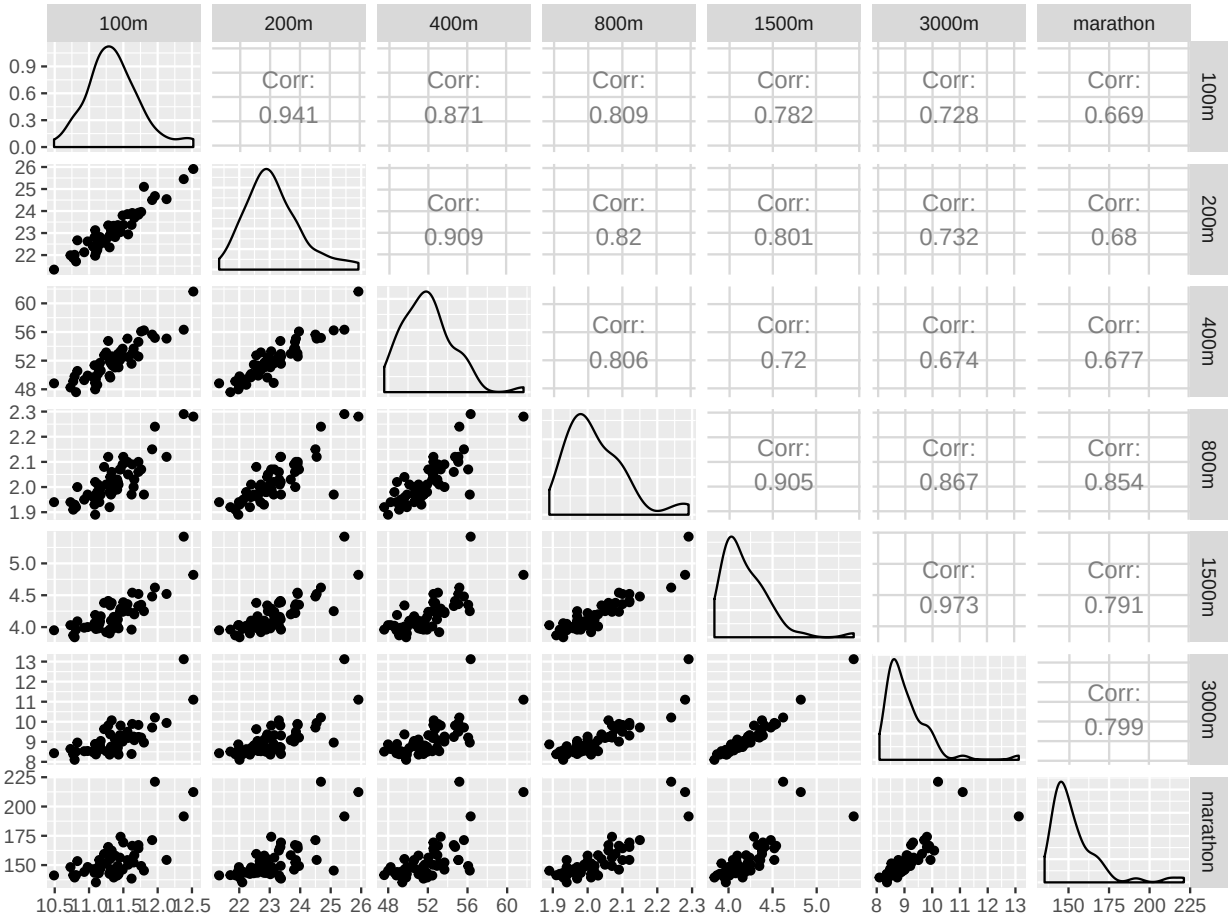
Correlation matrix

```
##           100m  200m  400m  800m 1500m 3000m marathon
## 100m     1.000 0.941 0.871 0.809 0.782 0.728    0.669
## 200m     0.941 1.000 0.909 0.820 0.801 0.732    0.680
## 400m     0.871 0.909 1.000 0.806 0.720 0.674    0.677
## 800m     0.809 0.820 0.806 1.000 0.905 0.867    0.854
## 1500m    0.782 0.801 0.720 0.905 1.000 0.973    0.791
## 3000m    0.728 0.732 0.674 0.867 0.973 1.000    0.799
## marathon 0.669 0.680 0.677 0.854 0.791 0.799    1.000
```

The correlation coefficient is a measure that calculates the strength and direction of the linear relationship between two variables ($r \in \{-1, +1\}$). We observe that all the coefficients are positive (and greater than 0.65) so all pairs of variables have a positive linear relationship meaning that, as the value of one variable increases, the value of the other variable increases too. The highest coefficient is $\text{Cor}(1500\text{m}, 3000\text{m}) = 0.9734$ and the lowest, $\text{Cor}(100\text{m}, \text{Marathon}) = 0.6690$.

2.b

In this plot the diagonal represents the densities, the lower left part of the matrix contains the scatterplots and the upper right contains the Correlations between the variables.

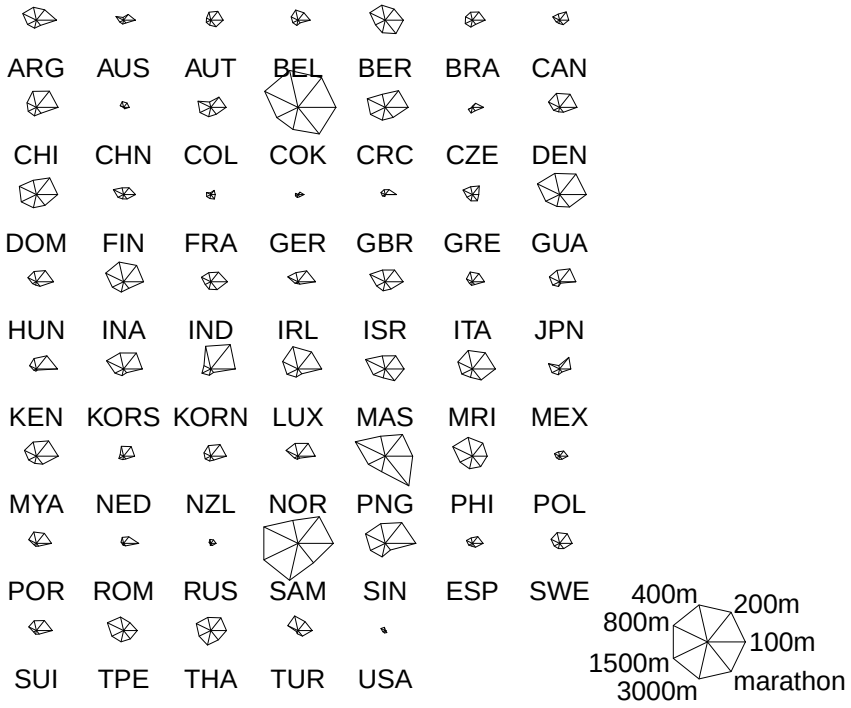


From pair scatterplot, we could see the positive linear relation between the variables and we can spot,for example, strong positive correlation between the 1500m and 3000m variables. Also, we can observe outliers at almost every pair (e.g. '200m' vs '800m','400m' vs '800m', '400m' vs '1500m'), but it is highly seen on every valiable vs 'marathon'.

2.c

Star Plot

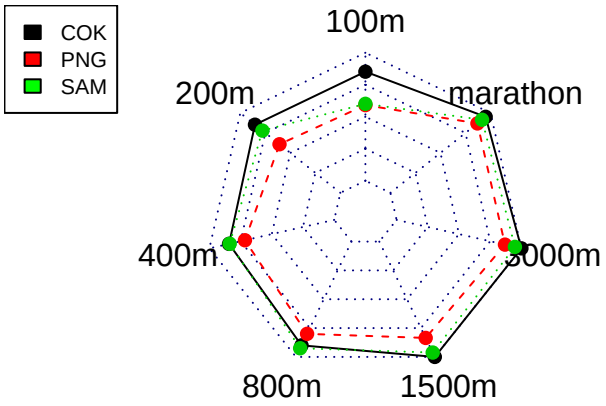
Every country in star plot is represented by an heptagon and each variable is the line that connects the center with each vertex. The length of each line represents how big the value of the variable is. Thus, the biggest the heptagon the worse the performance for a country.



Based on the above analysis, COK, SAM and PNG are the three countries with the worst performances. On the opposite hand, CHN, GER, USA and RUS seem to be the countries with the best performances.

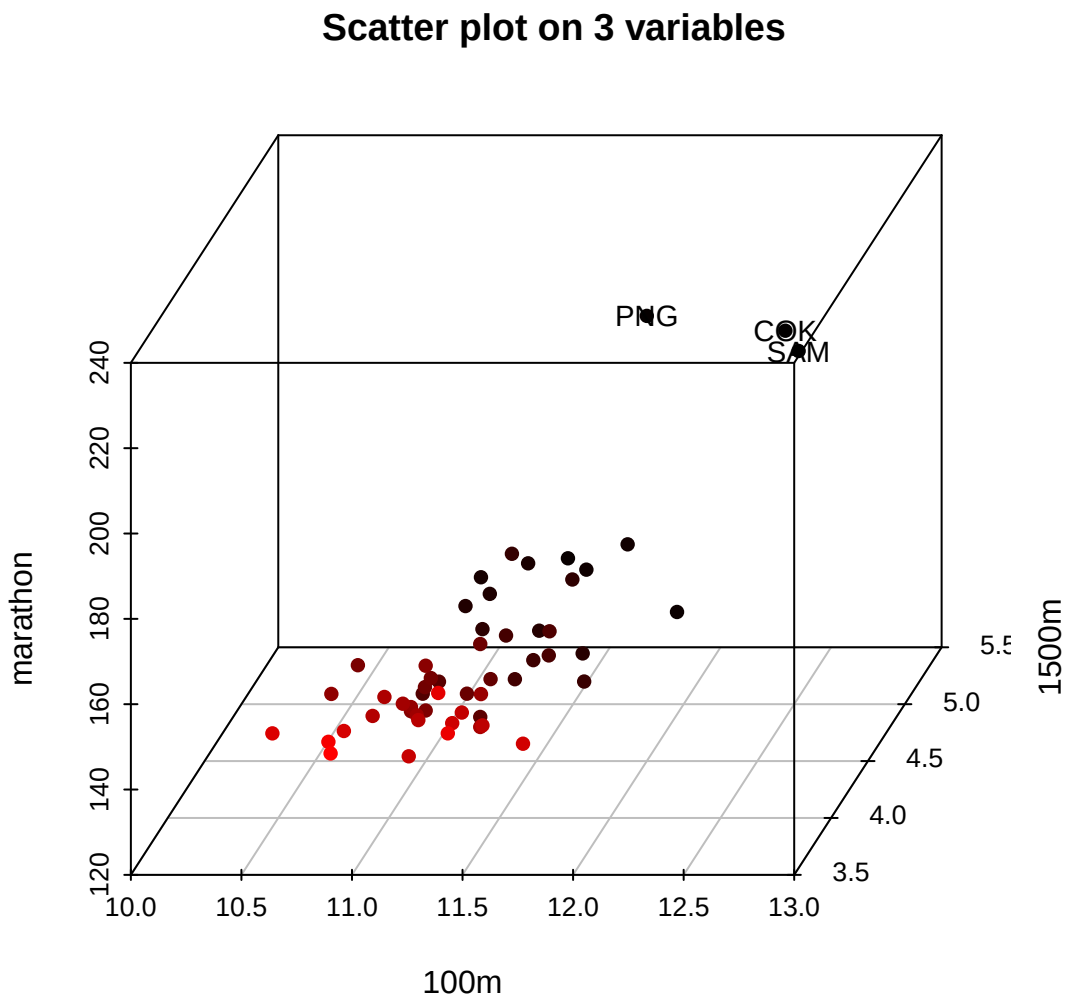
Radar Chart

In Radar Chart, the biggest the heptagon the biggest the values for each variable are. As we saw from the Star plot the three most extreme countries were COK,PNG and SAM, so we investigate them further using the Radar Chart.



3D scatterplot

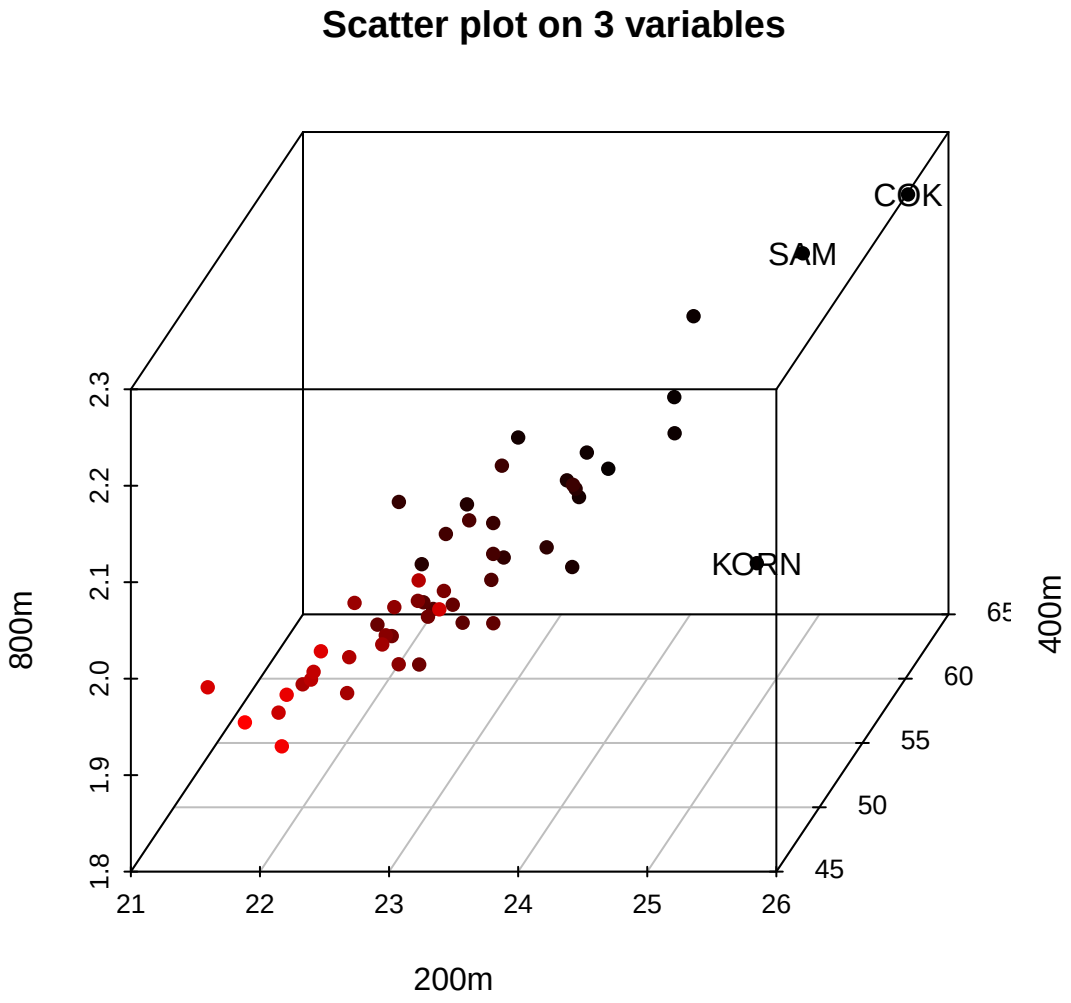
We will choose the 3 most uncorrelated variables, in order to have the most information, to plot in 3D scatterplot: “100m” vs “1500m” vs “marathon”.



Note: points are drawn in different colors related to y coordinates.

We can see that PNG, COK and SAM seem to be outliers.

In the next 3D scatterplot we consider: 200m, 400m and 800m because we saw from 2b) that they have outliers.



The outliers that we can observe are for SAM,KORN and COK.

Question 3: Examining for extreme values

3.a

Which 3-4 countries appear most extreme?

The only extreme country in terms of good performance is United States (USA) just for 100m variable and that is seen from its scatter plot.

As we can see from star plots above, in terms of bad performance the most extreme countries are Samoa (SAM), Cook Islands (COK), Papua New Guinea (PNG) and Guatemala (GUA).

We consider them extreme because the values of the times for the runtypes are either too small compared to the mean value (good performance) or too big (bad performance).

3.b The Euclidean distance

```
##      PNG      COK      SAM      BER      GBR
## 67.62796 59.61517 38.52476 20.61606 18.59146
```

The five most extreme countries are Papua New Guinea (PNG), Cook Islands (COK), Samoa (SAM), Bermuda (BER) and United Kingdom (GBR).

3.c The Euclidean distance for the normalized data

```
##      SAM      COK      PNG      USA      SIN
## 75.10146 64.13381 33.88423 12.83796 11.38478
```

Using the Euclidean distance for the normalized data, the five most extreme countries are Samoa (SAM), Cook Islands (COK), Papua New Guinea (PNG), United States (USA) and Singapore (SIN).

We observe that compared with the unnormalized variables, three out of the five countries (SAM,COK,PNG) are considered extreme in both cases, however they differ in 2 countries and that was expected because in the above Euclidean distance the variables are not standardized so we couldn't expect the same results.

3.d The Mahalanobis distance

```
##      SAM      PNG     KORN      COK      MEX
## 34.97640 30.40781 22.41218 20.00230 13.87395
```

Using the Mahalanobis distance, the five most extreme countries are Samoa (SAM), Papua New Guinea (PNG), North Korea (KORN), Cook Islands (COK) and Mexico (MEX).

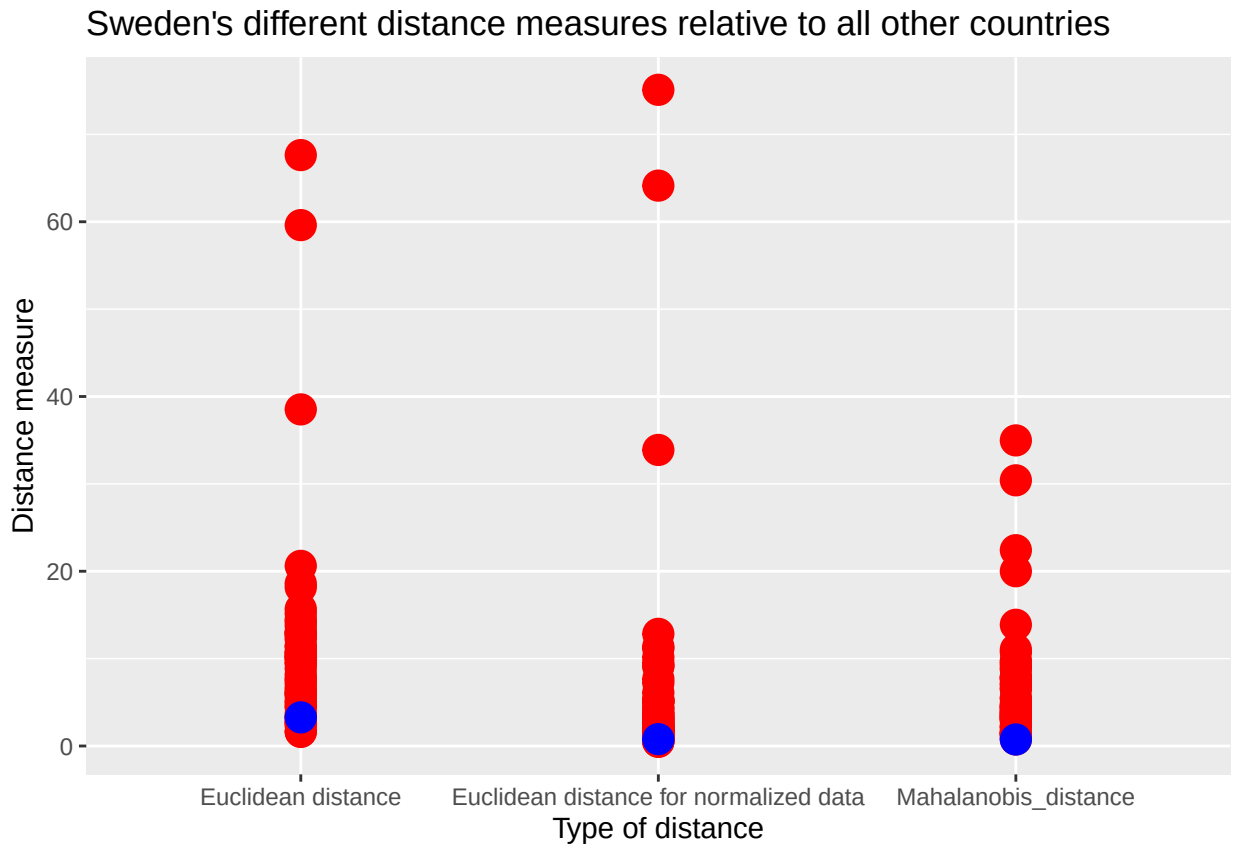
3.e

Euclidean distance is the most well known form of distance calculation. However, it is not the best way to calculate distances in a dataset and that's because it is sensitive to the scales of the variables involved, and it actually doesn't take into account this difference between the variables. Another drawback of the Euclidean distance is that it does not take into account the correlation between the variables and thus, if a variable is correlated to another variable then euclidean distance will weight the correlated variable more heavily in its calculations than the other variables leading to not so accurate results.

To avoid those biased measurments, we can standardize the data so as to elliminate units and weigh equally the variables.

The Mahalanobis distance takes into account the covariance among the variables in calculating distances.So, in contrast to the Euclidean distance, Mahalanobis distance does not get affected in terms of results from the units and the correlation between the variables.

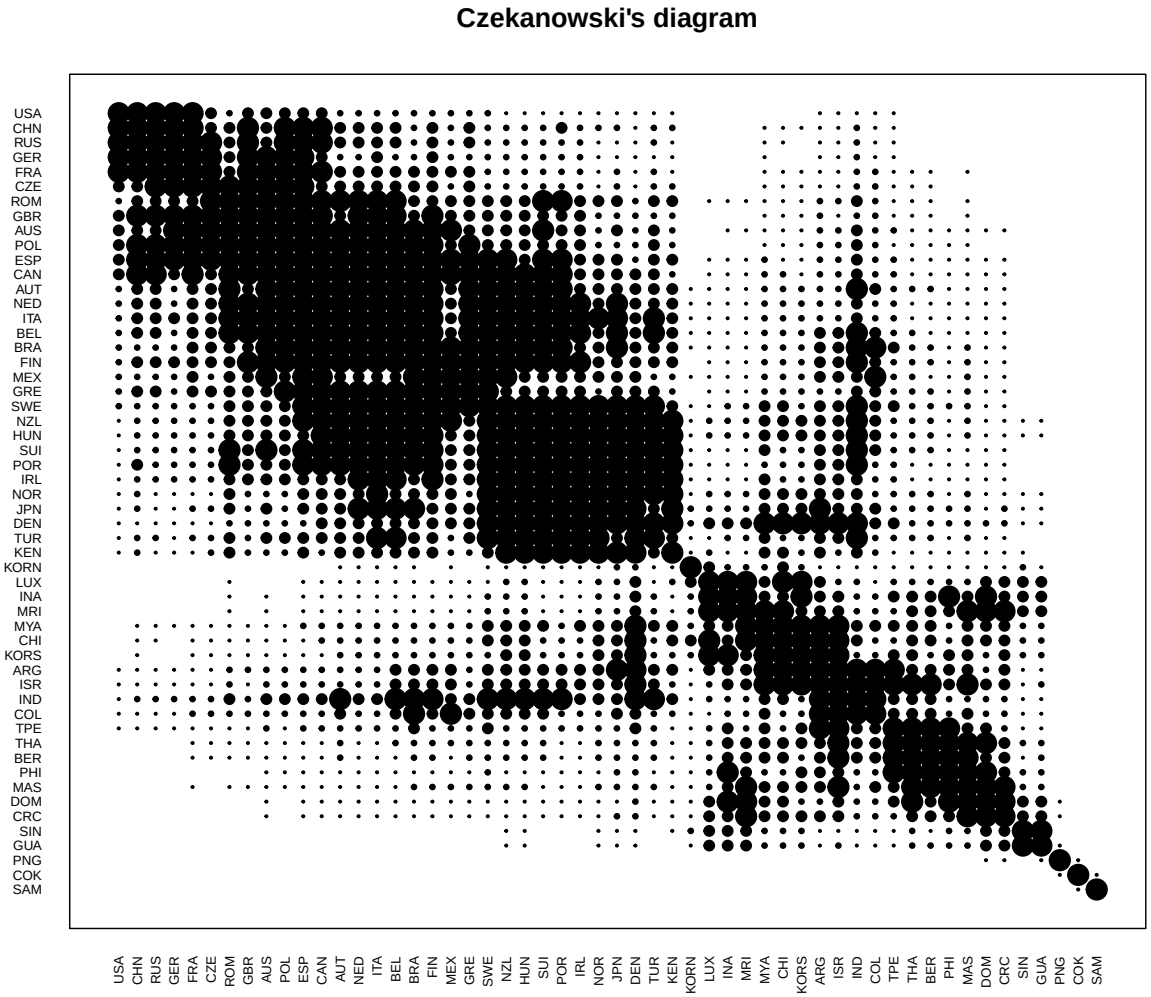
In our case the variables represent the run type and that's why all the countries try to have low values and the extreme values represent the worst performances, which could be caused by some injuries, for instance in marathon.



In this plot, each point represents a country (blue points represent Sweden), x-axis shows the three different distance measures that we have used and y-axis indicates the distance value.

We observe that Sweden is in the three cases quite close to the mean (distance is close to 0), so it has an average performance.

Czekanowski's diagram



Czekanowski's diagram represents with dots the distances between all possible pairs of countries. The bigger the dot is, the smaller the distance is. The diagonal represents the distance of each country with itself, that's why the dots in the diagonal are the biggest and bold. We can observe from the diagram the clusters that have been created out of the observations that have similar values. We could say that in our case there are three clusters. In the top left part of the diagonal we find the cluster with the countries with the best performances, in the middle we can see the countries with an average performance and in the down right part there is the cluster with the countries with the worst performances. The extreme values are in the end of the diagonal and are not included in any of the clusters. USA, for instance, which had a very good performance is in the first cluster, Sweden which had an average performance is in the second cluster and Singapur is in the last cluster with not so good performance. We can confirm those by comparing to the Star plot used above.

Appendix

```
knitr::opts_chunk$set(echo = TRUE, message = F, warning = F, error = F)
# NEEDED LIBRARIES
library(ggplot2)
library(tidyr)
library(gridExtra)
library(car)
library(scatterplot3d)
library(fmsb)
library(GGally)
library(RMaCzek)
##################--QUESTION 1--##################

#library(ggplot2)
#library(tidyr)
df = read.table("T1-9.dat")
colnames(df) = c("country","100m", "200m", "400m", "800m", "1500m", "3000m" , "marathon")
df1 = df[,-1]  # without the column of the countries
colnames(df1) = c("100m", "200m", "400m", "800m", "1500m", "3000m" , "marathon")
# dim(df) #54x8
# colSums(is.na(df)) #how many NANs per column
# head(df)
# the value that appears most often
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

statsdf = data.frame(cbind(mean=sapply(df1, mean),
                        median=sapply(df1, median),
                        mode=sapply(df1, getmode),
                        min=sapply(df1, min),
                        max=sapply(df1, max),
                        range=sapply(df1, function(x)max(x)-min(x)),
                        sd=sapply(df1, sd),
                        skewness=sapply(df1, timeDate::skewness),
                        kurtosis=sapply(df1, timeDate::kurtosis)
                        ))

as.data.frame(t(round(statsdf,3)))
#statsdf
# tidyr::gather
# a table with columns the type of the run and the seconds
df2 = gather(df1[,c(1,2,3)], "runtype", "seconds")
df22 = gather(df1[,-c(1,2,3)], "runtype", "minutes")
#df222 = gather(df1[,-c(3,7)], "runtype", "seconds")  #without: 400m, marathon
#df2222 = gather(df1[,-c(2,3,6 ,7)], "runtype", "seconds") #without: 200m, 400m, marathon
# https://www.r-graph-gallery.com/89-box-and-scatter-plot-with-ggplot2.html

#library(gridExtra)
ggplot(df2, aes(x=runtype, y=seconds, fill=runtype)) + geom_boxplot()
#ggplot(df22, aes(x=runtype, y=minutes, fill=runtype)) + geom_boxplot()
```

```r
#ggplot(df222, aes(x=runtype, y=seconds, fill=runtype)) + geom_boxplot()
#ggplot(df2222, aes(x=runtype, y=seconds, fill=runtype)) + geom_boxplot()

#grid.arrange(plot1, plot2, ncol=2)
ggplot(df22, aes(x=runtype, y=minutes, fill=runtype)) + geom_boxplot()
par(mfrow=c(3,3), oma = c(0,0,0,0) , mar = c(4,4,0,0) + 0.1)
for(i in 2:8){
 low_dist <- mean(df[,i])-2*sd(df[,i])  # 95.45% of the values are inside this interval:
 upp_dist <- mean(df[,i])+2*sd(df[,i])
 extreme <- which(df[,i] > upp_dist | df[,i] < low_dist)

plot(df[,i], df[,1], xlab=names(df[i]), ylab="time",pch = 19)+
  abline(v=c(low_dist,upp_dist, mean(df[,i])), col=c("red","red","green"),lwd = 2) +
  text(df[extreme,i], df[extreme,1],as.vector(df[extreme,1]),cex = 1.1 )
}
#library("car")

#par(mfrow=c(3,3),mar=c(4, 3.8, 1, 0.5))  #(bottom, left, top, right)
par(mfrow=c(2,2),mar=c(4, 3.8, 1, 0.5))  #(bottom, left, top, right)
qqPlot(df$'100m',ylab="sample",main = "100m Q-Q plot",envelope=.95,
       col.lines="red",pch=19,id=F)
qqPlot(df$'200m',ylab="sample",main = "200m Q-Q plot",envelope=.95,
       col.lines="red",pch=19,id=F)
qqPlot(df$'400m',ylab="sample",main = "400m Q-Q plot",envelope=.95,
       col.lines="red",pch=19,id=F)
#pch = 19: solid circle
#envelope=.95: 95% confidence level  #id=TRUE: show the id of the outliers
par(mfrow=c(2,2),mar=c(4, 3.8, 1, 0.5))  #(bottom, left, top, right)
qqPlot(df$'800m',ylab="sample",main = "800m Q-Q plot",envelope=.95,
       col.lines="red",pch=19,id=F)
qqPlot(df$'1500m',ylab="sample",main = "1500m Q-Q plot",envelope=.95,
       col.lines="red",pch=19,id=F)
qqPlot(df$'3000m',ylab="sample",main = "3000m Q-Q plot",envelope=.95,
       col.lines="red",pch=19,id=F)
qqPlot(df$marathon,ylab="sample",main = "marathon Q-Q plot",envelope=.95,
       col.lines="red",pch=19,id=F)
# https://www.r-graph-gallery.com/74-margin-and-oma-cheatsheet.html
par(mfrow=c(3,3),mar=c(4, 3.8, 2, 0.5))  #(bottom, left, top, right)   ##oma=c(,,,)
dens = density(df1$`100m`)
hist(df1$`100m`, freq=F, xlab="seconds",main="100m") #probability densities
lines(dens,col="red")

dens = density(df1$`200m`)
hist(df1$`200m`, freq=F, xlab="seconds",main="200m") #probability densities
lines(dens,col="red")

dens = density(df1$`400m`)
hist(df1$`400m`, freq=F, xlab="seconds",main="400m", ylim=c(0,0.16))
lines(dens,col="red") #prob density

dens = density(df1$`800m`)
hist(df1$`800m`, freq=F, xlab="minutes",main="800m")
lines(dens,col="red")
```

```r
dens = density(df1$`1500m`)
hist(df1$`1500m`, freq=F, xlab="minutes",main="1500m", ylim=c(0,1.8))
lines(dens,col="red")

dens = density(df1$`3000m`)
hist(df1$`3000m`, freq=F, xlab="minutes",main="3000m", ylim=c(0,0.75))
lines(dens,col="red")

dens = density(df1$`marathon`)
hist(df1$`marathon`, freq=F, xlab="minutes",main="marathon")
lines(dens,col="red")

##################--QUESTION 2--##################

# sample covariance matrix
C = round(cov(df1),3)
C
R = round(cor(df1),3)
R
#library(GGally)
ggpairs(df1,progress=F)
d = df[,-1]
rownames(d) = df$country
stars(d, full = T, key.loc = c(20, 2))
#library(fmsb)
#radar chart
min_row <- sapply(df1, min)
max_row <- sapply(df1, max)
data_radar <- rbind(min_row, max_row, df1)
data_radar[,1:7] <- lapply(data_radar[,1:7], function(x) as.numeric(as.character(x)))

# countries c(1,2,13,49,55)
radarchart(data_radar[c(1,2,11,40,46),1:7])   # "COK"=11,"PNG"=40,"SAM"=46
legend("topleft", c("COK","PNG","SAM"),fill=c("black","red","green"),cex = 0.70)

#library(scatterplot3d)
s3d = scatterplot3d(df1[,c(1,5,7)],angle =60)
s3d1 = scatterplot3d(df1[,c(2,3,4)],angle =60)
#http://www.sthda.com/english/wiki/scatterplot3d-3d-graphics
       #-r-software-and-data-visualization
scatterplot3d(df1[,c(1,5,7)], pch = 16, highlight.3d = T,angle =60,
              grid=T,main="Scatter plot on 3 variables") #color="steelblue", type = "h"
text(s3d$xyz.convert(df1[c(40,11,46),c(1,5,7)]),
     labels=as.vector(df[c(40,11,46),1]),cex= 0.9)

scatterplot3d(df1[,c(2,3,4)], pch = 16, highlight.3d = T, angle =60,
              grid=T,main="Scatter plot on 3 variables")
text(s3d1$xyz.convert(df1[c(31,11,46),c(2,3,4)]),
     labels=as.vector(df[c(31,11,46),1]),cex = 1  )
##################--QUESTION 3--##################

#center the raw data by the means
resid = sapply(df[,-1],function(x)x-mean(x))
```

```r
#Euclidean distance between the observation and sample mean vector
euclidist = sqrt(abs(resid%*%t(resid)))

dists = diag(euclidist)   #diag is squared distance for each country
names(dists) = df$country
head(sort(dists, decreasing=TRUE),5)  #sort with decreasing order and take the first five
V = diag(diag(C)) #first diag to take the diagonal
                  #then diag again to make the vector a diagonal matrix
sqdist = resid%*%solve(V)%*%t(resid)

sqdist = diag(sqdist)
names(sqdist) = df$country
#sort with decreasing order and take the first five
head(sort(sqdist, decreasing=TRUE),5)
mahdist = resid%*%solve(C)%*%t(resid)

mahdist = diag(mahdist)
names(mahdist) = df$country
#sort with decreasing order and take the first five
head(sort(mahdist, decreasing=TRUE),5)
#stripchart showing how Sweden behaves through the 3 different distance measures

#converting data structure to mine(David)
data_distance_vector<-as.data.frame(dists)
data_distance_vector<-cbind(rownames(data_distance_vector),
                            data_distance_vector, stringsAsFactors = FALSE)
colnames(data_distance_vector) <- c("Country", "distance")


data_distance_sd_vector <- as.data.frame(sqdist)
data_distance_sd_vector <- cbind(rownames(data_distance_sd_vector),
                                 data_distance_sd_vector, stringsAsFactors = FALSE)
colnames(data_distance_sd_vector) <- c("Country", "distance")


data_distance_mahalanobis <- as.data.frame(mahdist)
data_distance_mahalanobis <- cbind(rownames(data_distance_mahalanobis),
                                   data_distance_mahalanobis, stringsAsFactors = FALSE)
colnames(data_distance_mahalanobis) <- c("Country", "distance")


euclidean_squared<-cbind(data_distance_vector,type = "Euclidean distance",
                         stringsAsFactors = FALSE)
euclidean_normalized<-cbind(data_distance_sd_vector,
                            type = "Euclidean distance for normalized data",
                            stringsAsFactors = FALSE)
mahal<-cbind(data_distance_mahalanobis,type = "Mahalanobis_distance",
             stringsAsFactors = FALSE)



#merged df for graph
distance_merged <- rbind(euclidean_squared, euclidean_normalized, mahal)
```

```r
swe_eucl_squared <- distance_merged[which(distance_merged$Country=="SWE"),]$distance[1]
swe_eucl_normd <- distance_merged[which(distance_merged$Country=="SWE"),]$distance[2]
swe_mahal <- distance_merged[which(distance_merged$Country=="SWE"),]$distance[3]

data_summary <- function(x) {
  res<-0
  if(swe_eucl_squared %in% x) {
    res <- swe_eucl_squared
  } else if(swe_eucl_normd %in% x) {
    res <- swe_eucl_normd
  } else {
    res <- swe_mahal
  }
  return(y = res)
}

#unscaled plot
p<-ggplot(distance_merged, aes(x = type, y =distance)) +
  geom_jitter(position = position_jitter(0), size = 5, col = "red") +
  labs(title = "Sweden's different distance measures relative to all other countries",
       x = "Type of distance", y = "Distance measure")
p + stat_summary(fun.y = data_summary, geom = "point", shape = 19,
                 size = 5, color = "blue")

#library(RMaCzek)
df3d = df1
rownames(df3d) = df$country
m = czek_matrix(df3d)
plot.czek_matrix(m)
```