Multivariate Statistical Methods - Assignment 1

Henrik Karlsson, Milda Poceviciute, Ugurcan Lacin and Ramon Laborda

11/15/2017

The following packages are used in order to build the graphs

```
library(tidyverse)
library(ggplot2)
library(GGally)
library(ggcorrplot)
library(gridExtra)
library(scatterplot3d)
library(knitr)
```

The data is the national track records for women in 54 different countries in 7 different distances, 100m, 200m, 400m, 800m, 1500m, 3000m and marathon. We have assumed that variables for 100m to 400m is stored in decimal representation in seconds and that 800m to marathon is stored in decimal representation in minutes. Data is imported into R and named according to R syntax rules.

```
df <- read.csv("./T1-9.dat",
               sep = "\t", header = FALSE)
names(df) <- c("country", "m100", "m200", "m400", "m800", "m1500", "m3000", "marathon")
```

Question 1

Part a

By using numerical values in data set, some measurements are calculated, such as, mean, median, max, e.t.c. They are presented to understand how data set is.

```
tdf <- df %>%
    gather(distance, value, -country) %>%
    mutate(sort_order = case_when(distance == "m100" ~1,
                                  distance == "m200" ~2,
                                  distance == "m400" ~3,
                                  distance == "m800" ~4,
                                  distance == "m1500" ~5,
                                  distance == "m3000" ~6,
                                  distance == "marathon" ~7))

df_summary <- tdf %>%
    group_by(distance) %>%
    summarise(min = min(value),
              max = max(value),
              avg = round(mean(value),6),
              median = round(median(value),6),
              sd = round(sd(value),6),
              variance = round(var(value),6)) %>%
    left_join(., select(tdf, country, value, sort_order), by = c("min" = "value")) %>%
    rename(best = country) %>%
    left_join(., select(tdf, country, value), by = c("max" = "value")) %>%
```

```
    rename(worst = country) %>%
    arrange(sort_order) %>%
    select(-sort_order)


kable(df_summary, format = "markdown", digits = 2)
```

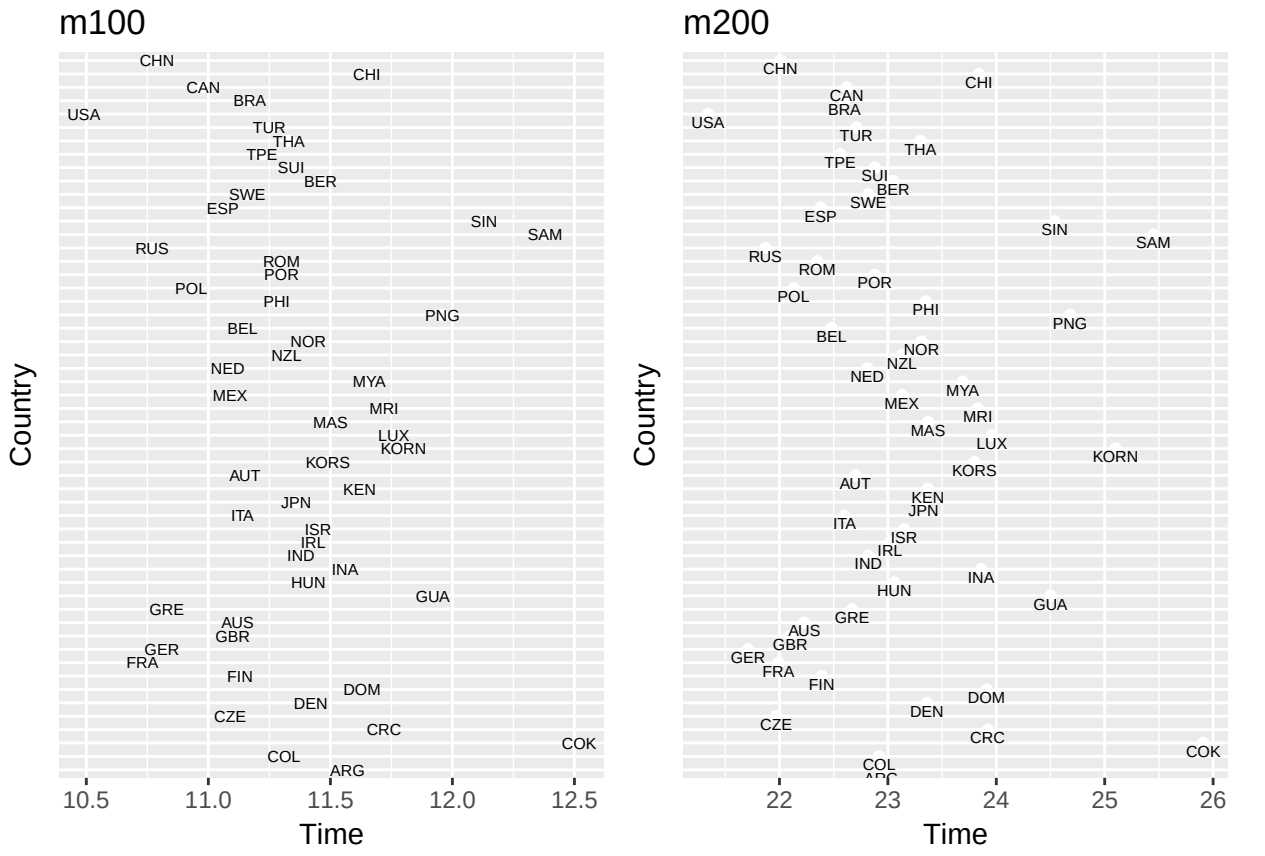
distance	min	max	avg	median	sd	variance	best	worst
m100	10.49	12.52	11.36	11.32	0.39	0.16	USA	COK
m200	21.34	25.91	23.12	22.98	0.93	0.86	USA	COK
m400	47.60	61.65	51.99	51.65	2.60	6.75	GER	COK
m800	1.89	2.29	2.02	2.00	0.09	0.01	CZE	SAM
m1500	3.84	5.42	4.19	4.10	0.27	0.07	CHN	SAM
m3000	8.10	13.12	9.08	8.85	0.82	0.66	CHN	SAM
marathon	135.25	221.14	153.62	148.43	16.44	270.27	GBR	PNG

Part b

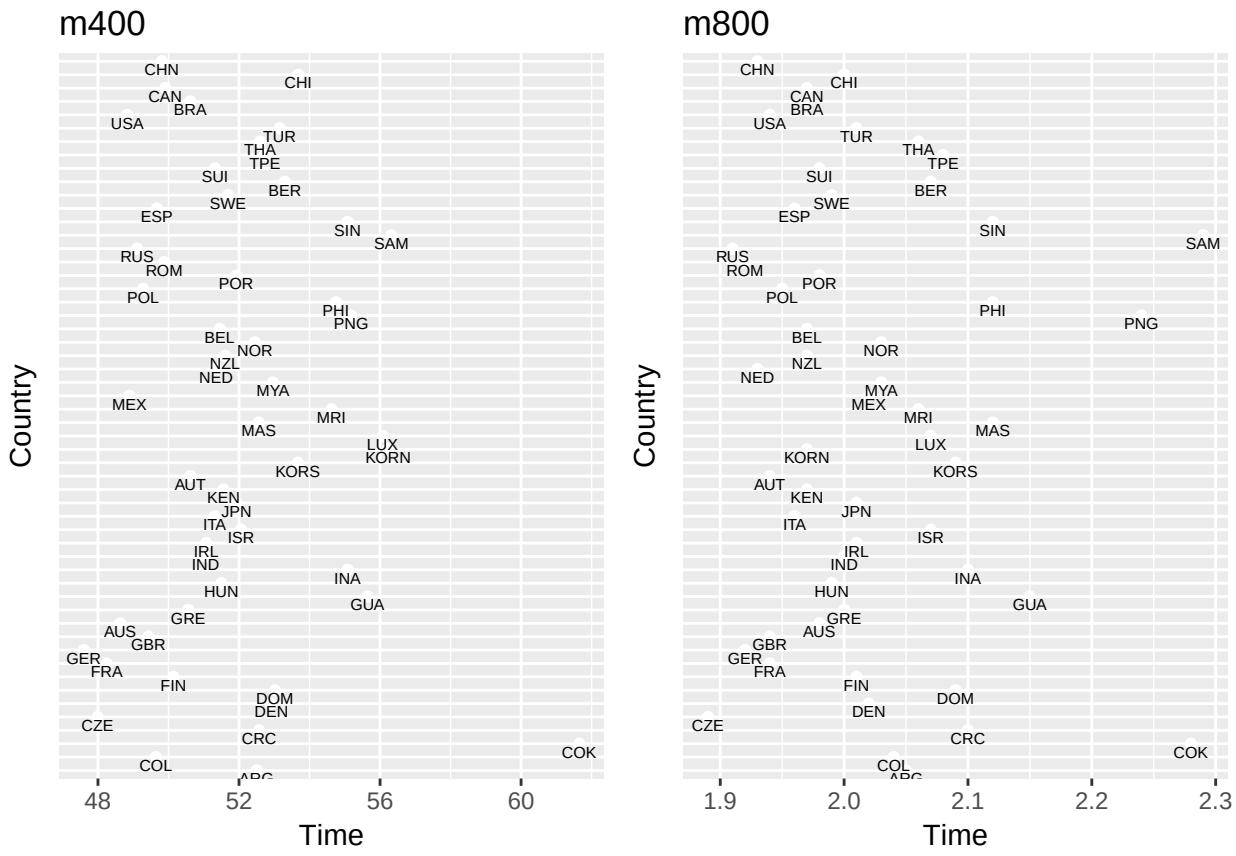
Using ggplot2 we represent the observation of every country with each time, to observe if there are outliers or extreme values.

```
plot1 <- ggplot(df, aes(x=df[, 2], y=row.names(df))) + geom_point(col="white")+
  labs(list(title =colnames(df)[2] , x = "Time")) +
  geom_text(aes(label=df[, 1]), size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot2 <- ggplot(df, aes(x=df[, 3], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[3] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot3 <- ggplot(df, aes(x=df[, 4], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[4] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot4 <- ggplot(df, aes(x=df[, 5], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[5] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot5 <- ggplot(df, aes(x=df[, 6], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[6] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
```

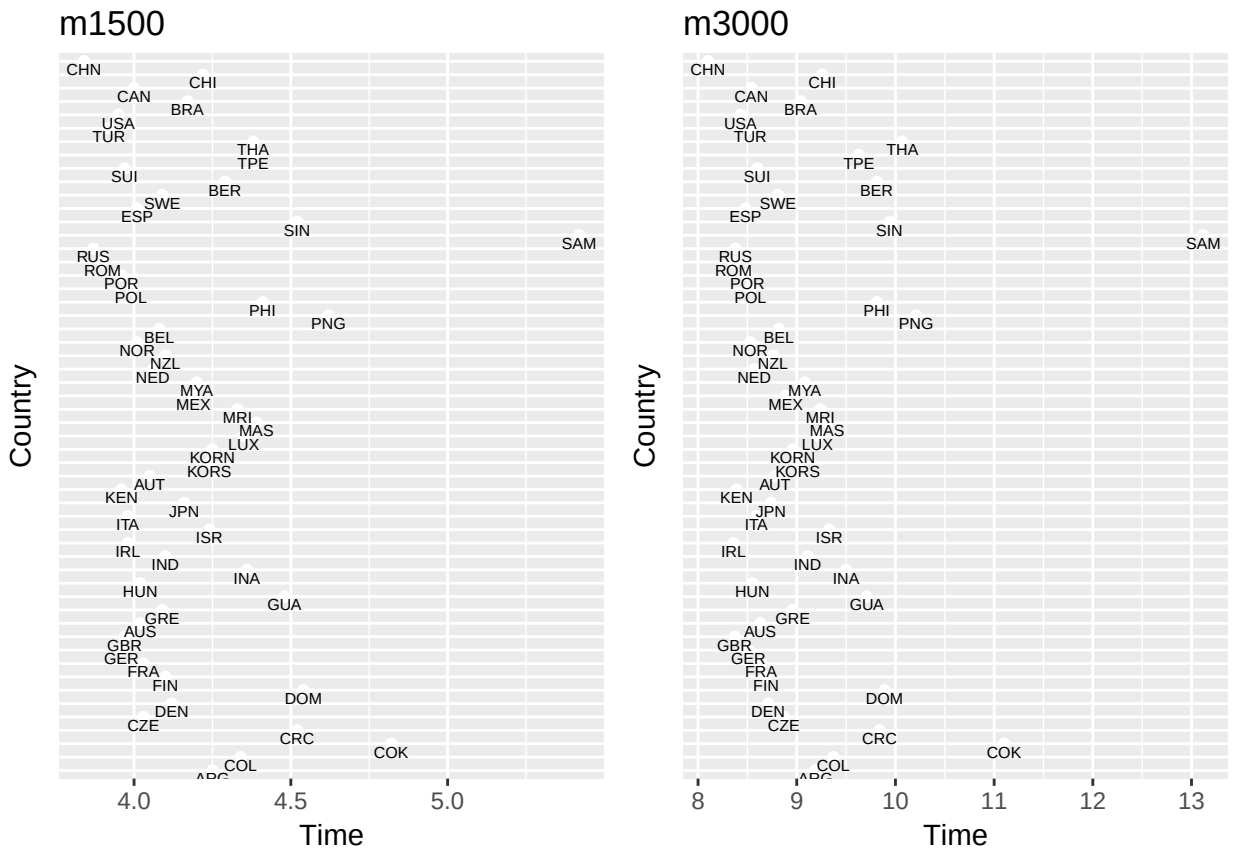
```
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot6 <- ggplot(df, aes(x=df[, 7], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[7] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot7 <- ggplot(df, aes(x=df[, 8], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[8] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

#Plots
grid.arrange(plot1, plot2, ncol=2)
```



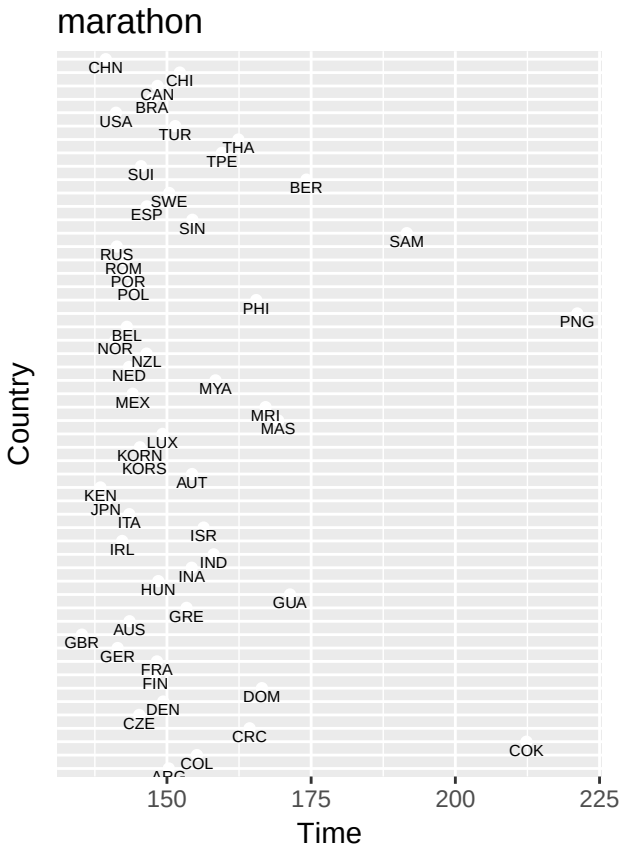

```
grid.arrange(plot3,plot4, ncol=2)
```




```
grid.arrange(plot5, plot6, ncol=2)
```



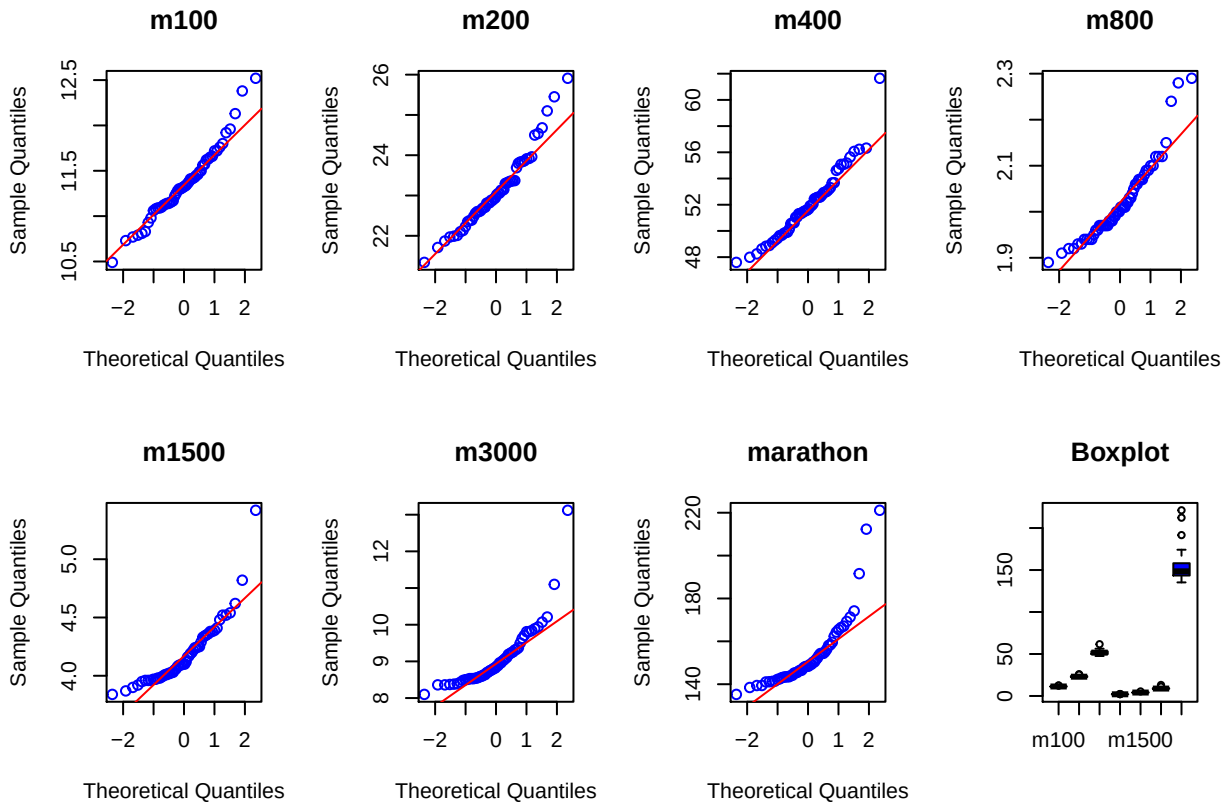

```
grid.arrange(plot7, ncol=2)
```



As seen in the graphs, COK(Cook Island) and SAM(Samoa) are the countries that have extreme values in most running distances.

Are the variables Normally distributed?

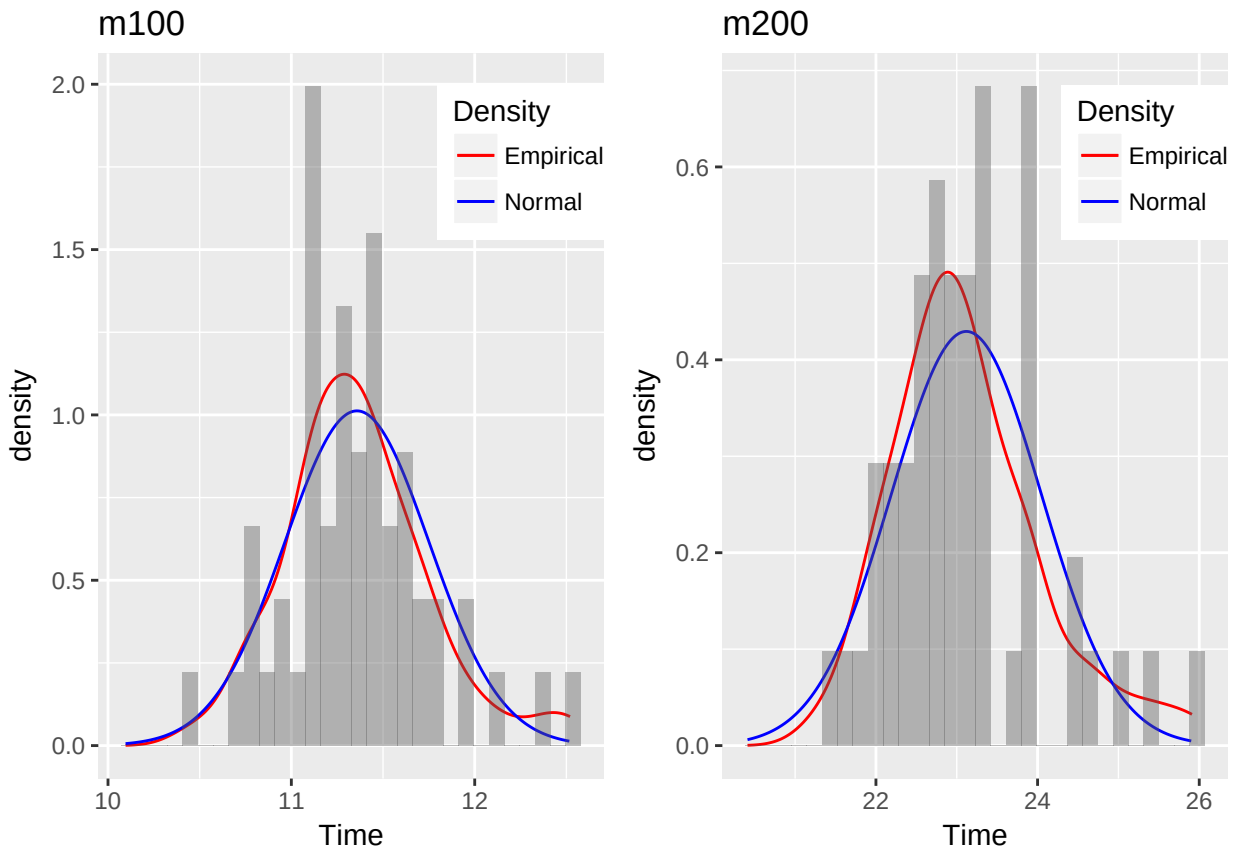
```
par(mfrow=c(2, 4))
for(i in 2:8){
  qqnorm(df[,i],main=colnames(df[i]), col="blue")
  qqline(df[,i], col="red")
}
boxplot_df <- boxplot(df[,2:8],col="blue",main="Boxplot")
```



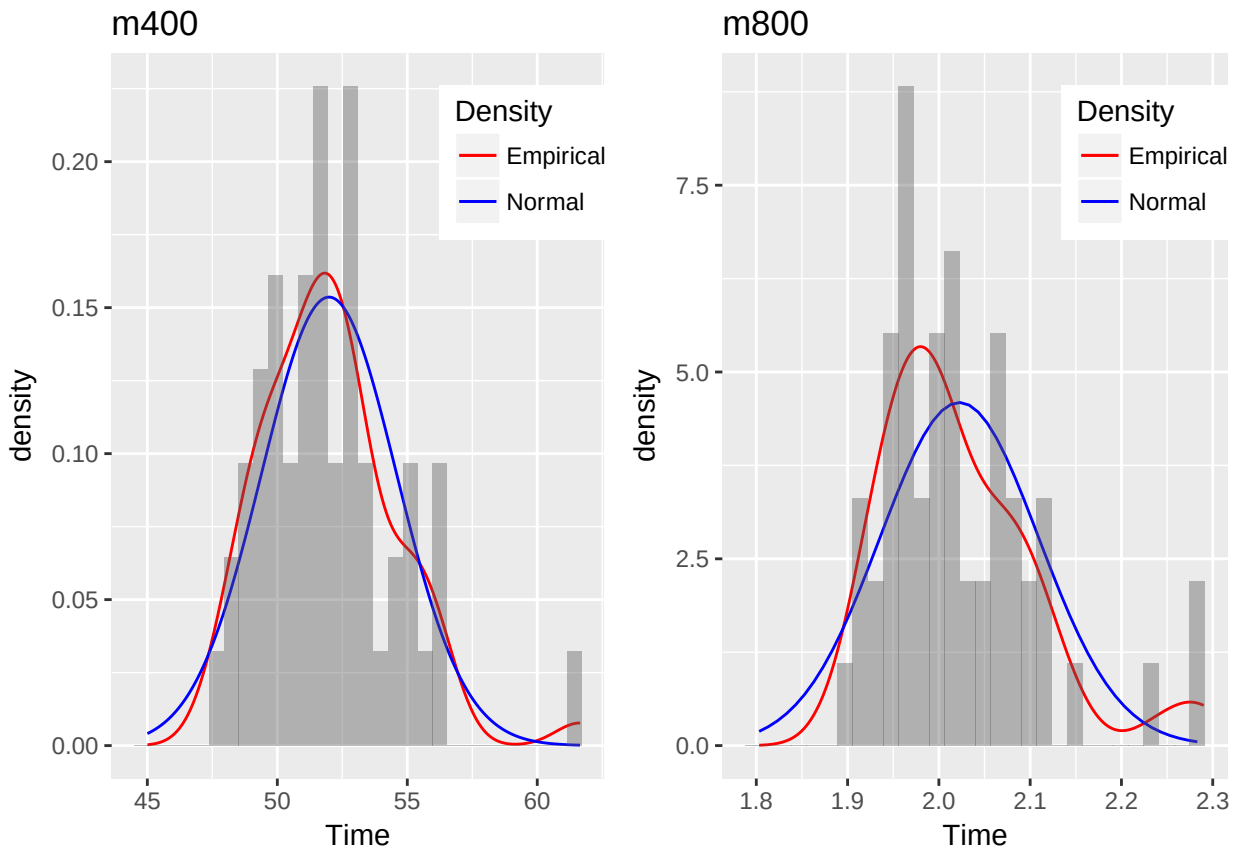

```
par(mfrow=c(1, 1))
```

The residuals of all the variables are close to the red line of normality, so we asume all of them have a Normal distribution. In the boxplot we can observe that the values differ more in a short distance race like 100m than marathon.

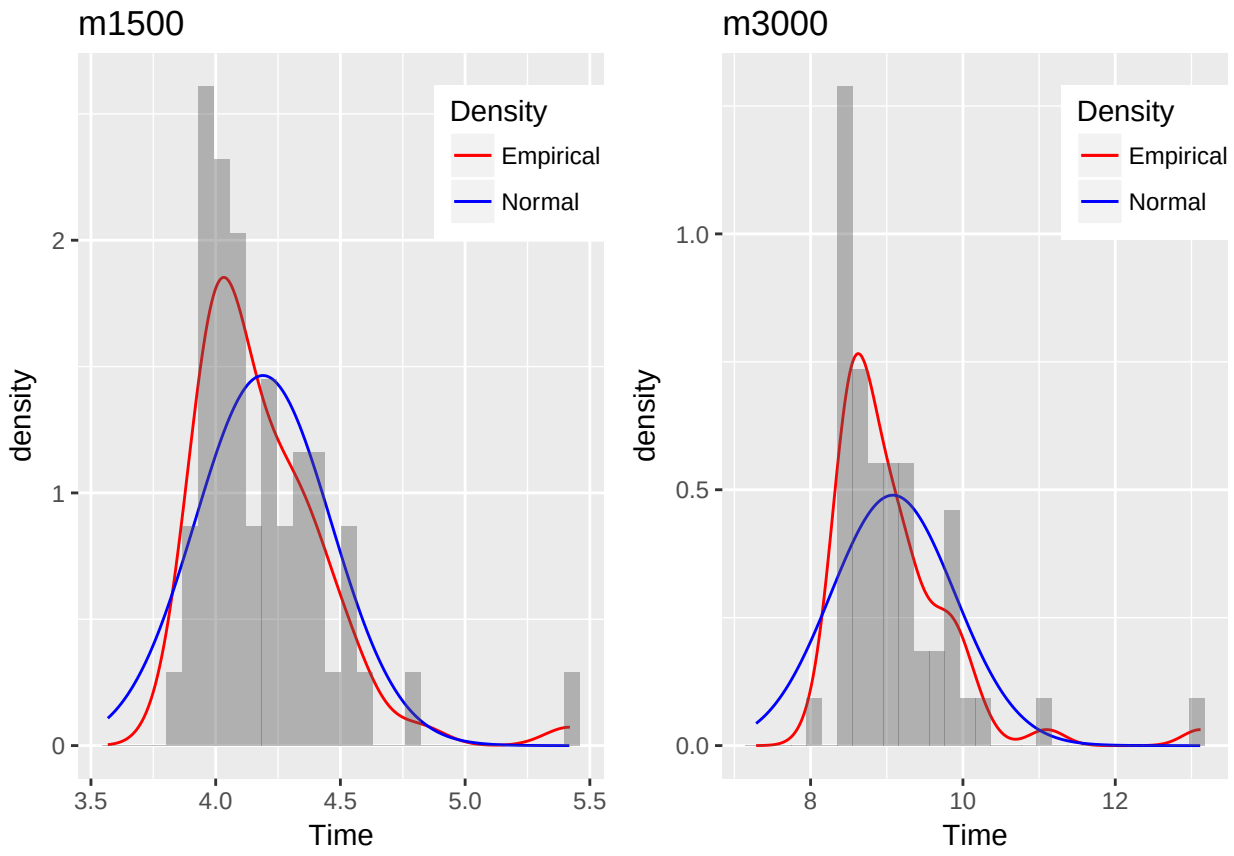
```
graphs <- list()
for(i in 1:nrow(df_summary)){
    range <- eval(substitute(seq(as.numeric(df_summary[i, "min"]) - as.numeric(df_summary[i, "sd"]), as
    ynorm <- eval(substitute(dnorm(x = range, mean = as.numeric(df_summary[i, "avg"]), sd = as.numeric(
    graphs[[i]] <- eval(substitute(qplot(df[,i+1], geom = "blank") +
        geom_line(aes(y = ..density.., colour = "Empirical"), stat = 'density') +
        geom_line(aes(x = range, y = ynorm, color = "Normal")) +
        geom_histogram(aes(y = ..density..), alpha = 0.4) +
        scale_colour_manual(name = 'Density', values = c('red', 'blue')) +
        theme(legend.position = c(0.85, 0.85)) +
        xlab("Time") +
        ggtitle(names(df[i+1])),list(i=i,range=range,ynorm=ynorm)))
}


grid.arrange(graphs[[1]], graphs[[2]], ncol=2)
```



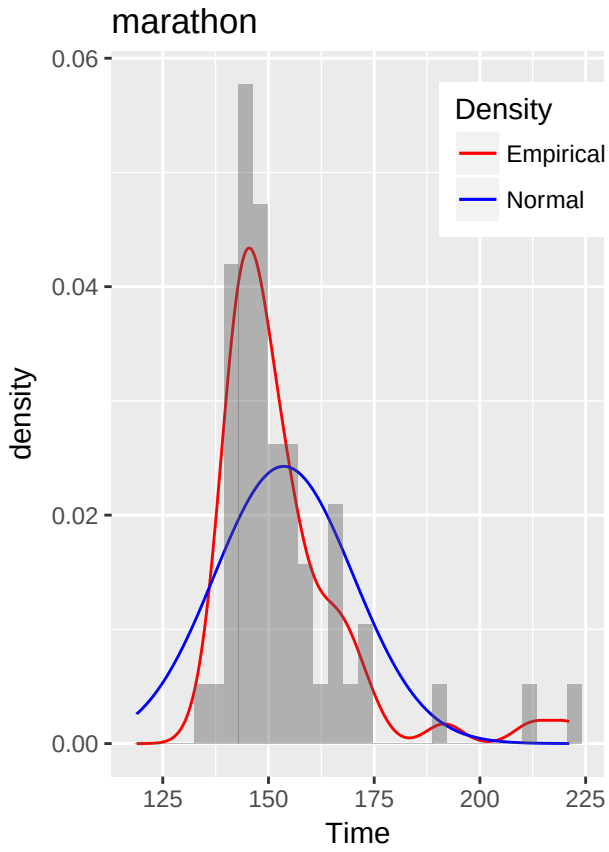

```
grid.arrange(graphs[[3]],graphs[[4]], ncol=2)
```



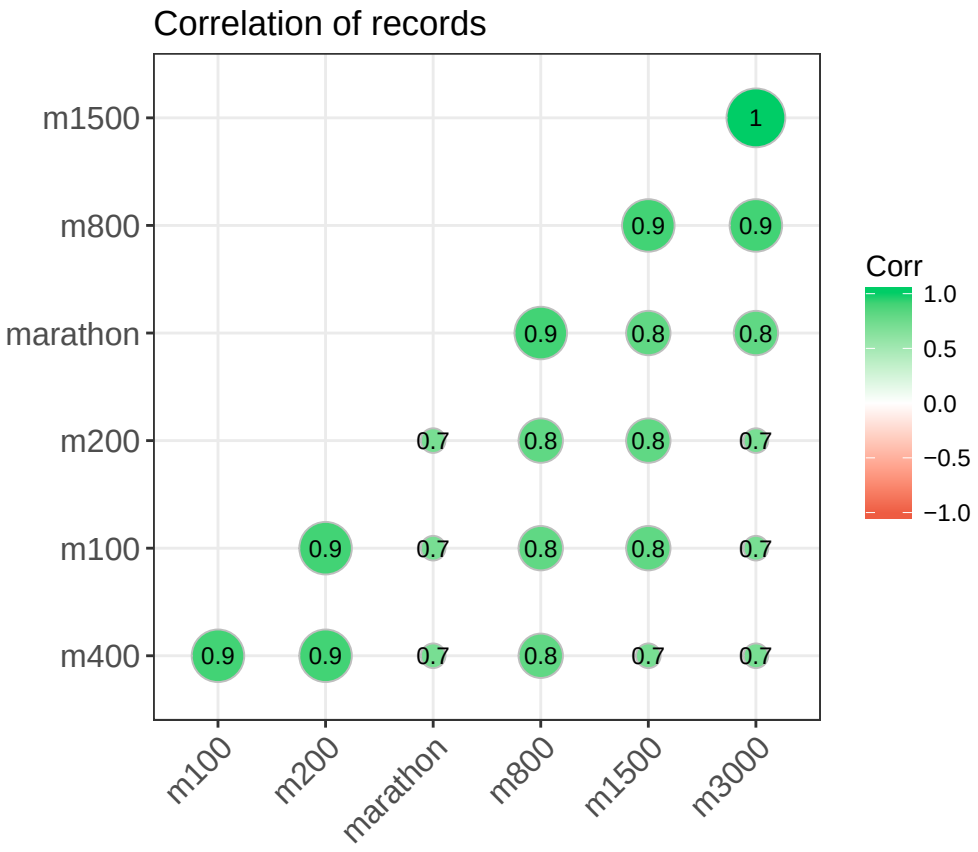

```
grid.arrange(graphs[[5]], graphs[[6]], ncol=2)
```




```
grid.arrange(graphs[[7]], ncol=2)
```

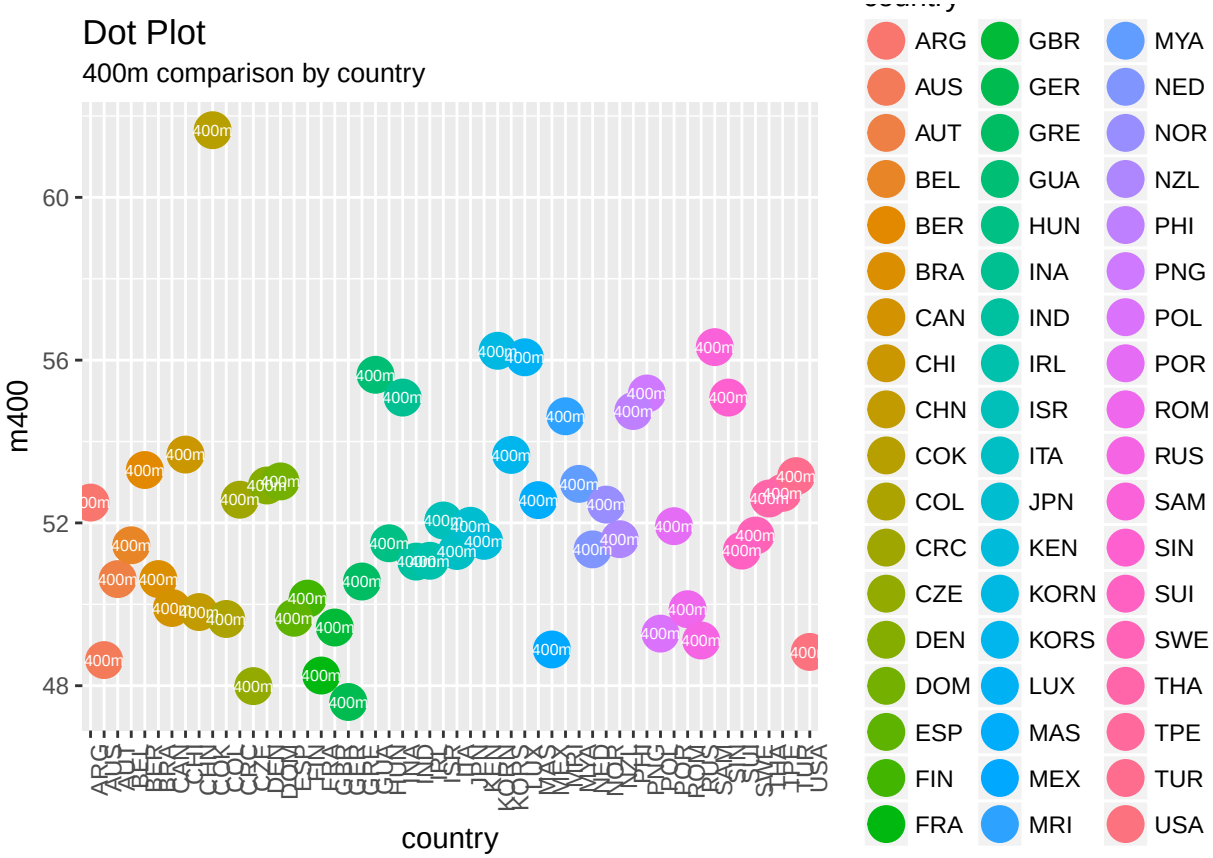



```
options(scipen=999)  # turn-off scientific notation like 1e+48
corr <- round(cor(df[2:8]), 1)

# Correlation Plot
ggcorrplot(corr, hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           method="circle",
           colors = c("tomato2", "white", "springgreen3"),
           title="Correlation of records",
           ggtheme=theme_bw)
```



In addition, correlation plot is another approach to present correleation between variables. According to this graph, the inference is possible that 3000m and 1500m are highly correlated each other which is 1 as a value. If 0.9 correlation value is considered, then 200m is also highly correlated with 400m and 100m. So, interpretation can be made easily.

```
marathonByCountries <- df[,c("country","marathon")]
sample <- head(marathonByCountries,10)

m400ByCountries <- df[,c("country","m400")]
sample <- head(m400ByCountries,20)
# Plot
ggplot(m400ByCountries, aes(x=country, y=m400, label="400m")) +
  geom_point(stat='identity', aes(col=country), size=6)  +
  geom_text(color="white", size=2) +
  labs(title="Dot Plot",
       subtitle="400m comparison by country") +
  theme(axis.text.x=element_text(angle=90,hjust=1))
```




```
# 400m COK has extreme value
# it can be produced for every type to find extreme on graph
```

Question 2

Part a

```
cor <- cor(df[,2:8])
print("Correlation Matrix:")
```

```
## [1] "Correlation Matrix:"
```

```
print(cor)
```

```
##                m100      m200      m400      m800     m1500     m3000
## m100      1.0000000 0.9410886 0.8707802 0.8091758 0.7815510 0.7278784
## m200      0.9410886 1.0000000 0.9088096 0.8198258 0.8013282 0.7318546
## m400      0.8707802 0.9088096 1.0000000 0.8057904 0.7197996 0.6737991
## m800      0.8091758 0.8198258 0.8057904 1.0000000 0.9050509 0.8665732
## m1500     0.7815510 0.8013282 0.7197996 0.9050509 1.0000000 0.9733801
## m3000     0.7278784 0.7318546 0.6737991 0.8665732 0.9733801 1.0000000
## marathon 0.6689597 0.6799537 0.6769384 0.8539900 0.7905565 0.7987302
##           marathon
## m100      0.6689597
## m200      0.6799537
```

```
## m400     0.6769384
## m800     0.8539900
## m1500    0.7905565
## m3000    0.7987302
## marathon 1.0000000
```

Looking at the correlation matrix above, it can be seen that the correlation is high between the short distance track records (100m to 400m) and medium distance track records (800m - 3000m). This is intuitive, because it's common for a short distance sprinter to compete in several distances and if a country have had a good sprinter, this person have most likely accomplished fast track records on several distances. The same logic applies to medium distance track records. The marathon have a weaker correlation compared to the other distances, which is intuitive, since it's the only long distance track record in the data set.

```
cov <- cov(df[,2:8])
print("Covariance Matrix:")
```

```
## [1] "Covariance Matrix:"
```

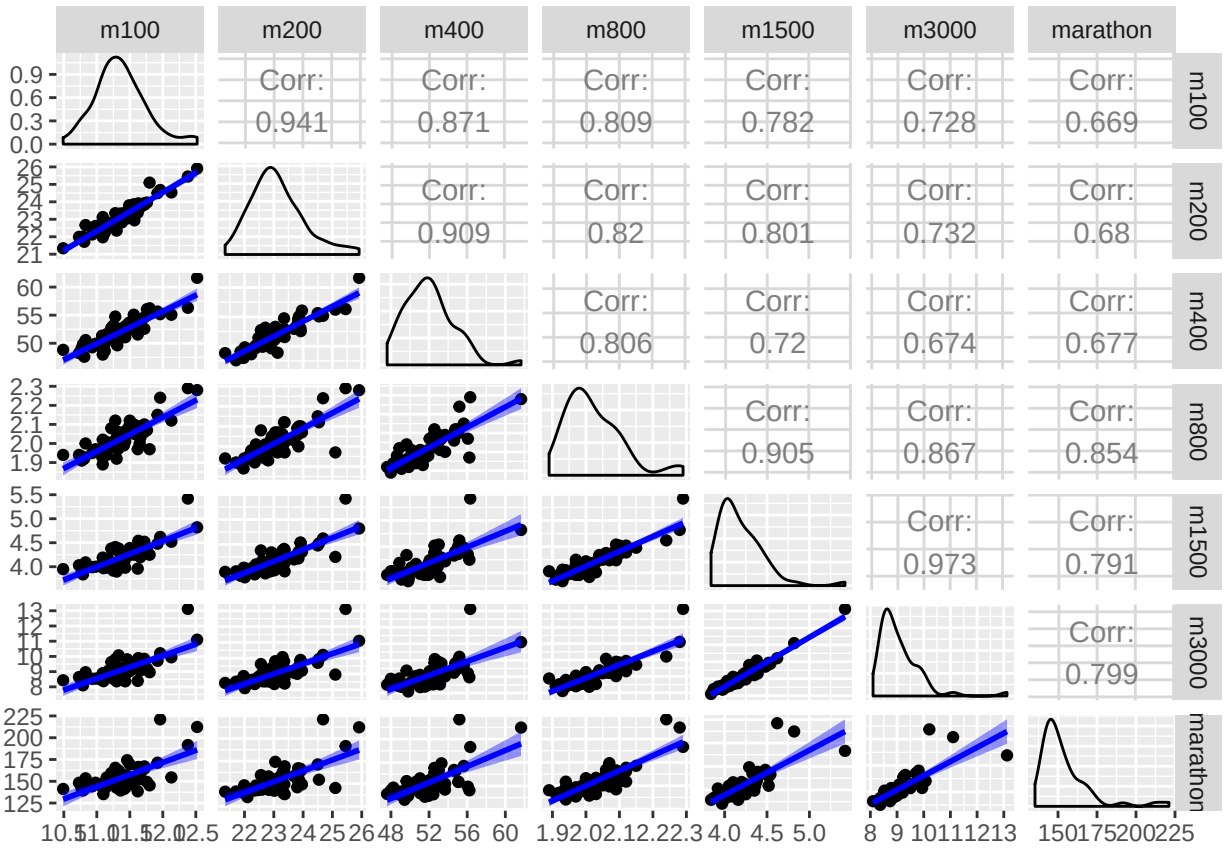
```
print(cov)
```

```
##                m100       m200       m400        m800      m1500
## m100     0.15531572  0.3445608  0.8912960 0.027703564 0.08389119
## m200     0.34456080  0.8630883  2.1928363 0.066165898 0.20276331
## m400     0.89129602  2.1928363  6.7454576 0.181807932 0.50917683
## m800     0.02770356  0.0661659  0.1818079 0.007546925 0.02141457
## m1500    0.08389119  0.2027633  0.5091768 0.021414570 0.07418270
## m3000    0.23388281  0.5543502  1.4268158 0.061379315 0.21615514
## marathon 4.33417757 10.3849876 28.9037314 1.219654647 3.53983732
##                m3000   marathon
## m100      0.23388281   4.334178
## m200      0.55435017  10.384988
## m400      1.42681579  28.903731
## m800      0.06137932   1.219655
## m1500     0.21615514   3.539837
## m3000     0.66475793  10.706091
## marathon 10.70609113 270.270150
```

The covariance matrix is a bit harder to interpret than the correlation matrix, due to the fact that the numbers in the covariance matrix isn't standardized as they are in the correlation matrix. However, the variance for each distance can be interpretted here. The longer the track distance, the higher variance. However this is not entirely true, but this is due to the fact that the data is not in the same scale for each distances.

Part b

```
my_fn <- function(data, mapping, ...){
  p <- ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method=lm, fill="blue", color="blue", ...)
  p
}
ggpairs(df[,2:8],columns = 1:7, lower = list(continuous = my_fn))
```



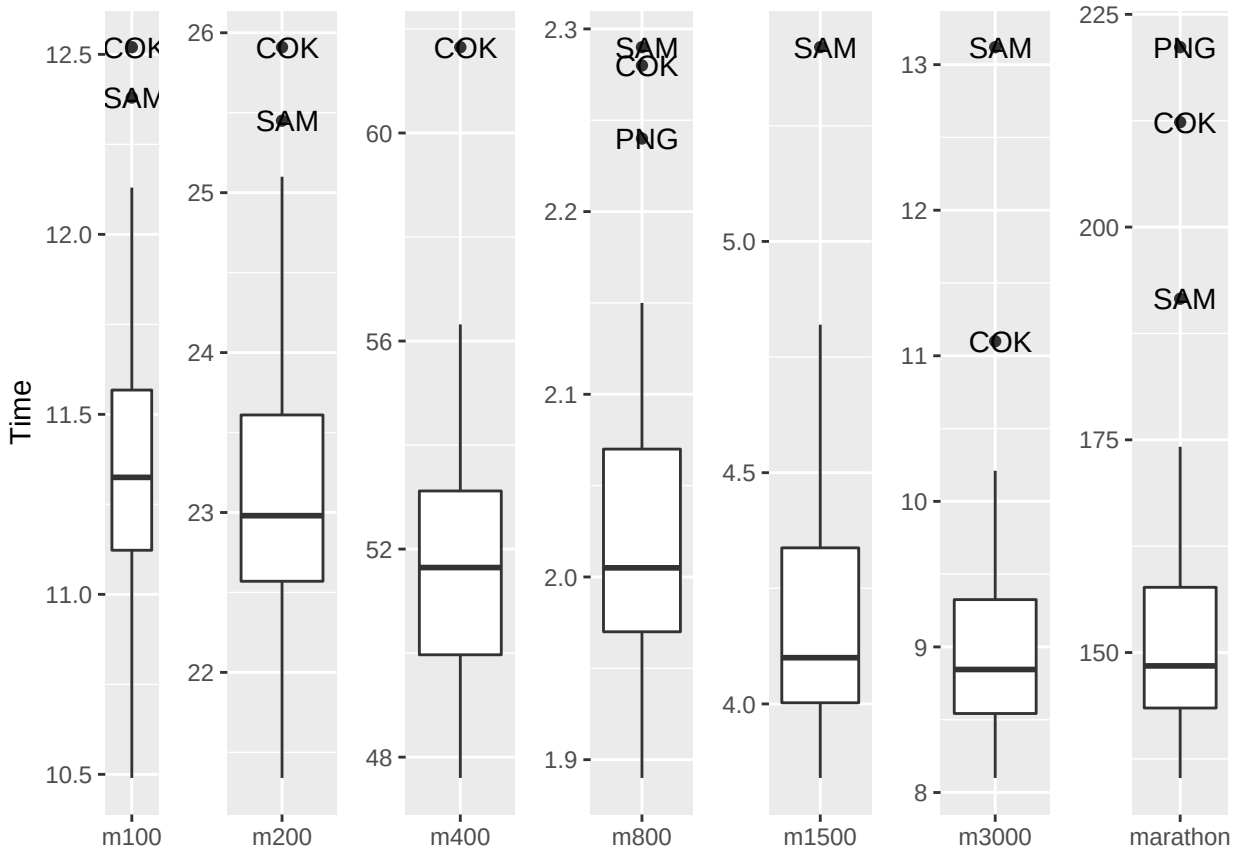
From the graph above, it can be seen from the scatterplots in the lower triangle, that there are some countries that stand out from the rest. The diagonal shows the density for each distance and the upper triangle shows the corrleation between each track distance.

A linear regression line with confidence interval have been added within each scatterplot.

Part c

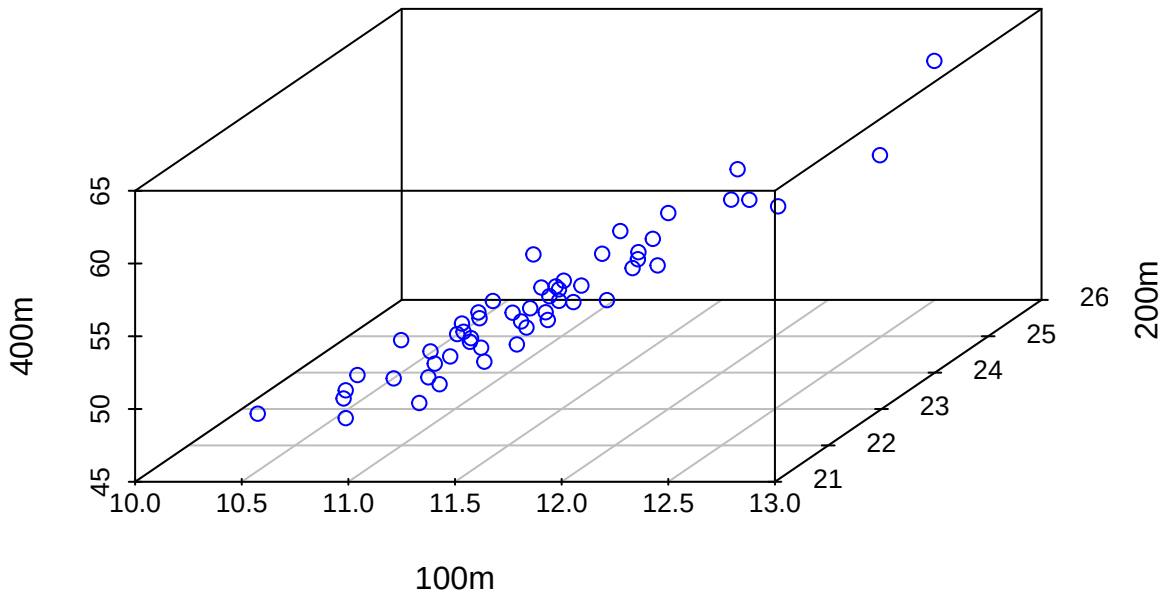
```
is_outlier <- function(x) {
  return(x < quantile(x, 0.25) - 1.5 * IQR(x) | x > quantile(x, 0.75) + 1.5 * IQR(x))
}

box_data <- tdf %>%
    group_by(distance) %>%
    mutate(outlier = is_outlier(value)) %>%
    mutate(outlier = ifelse(outlier == TRUE, as.character(.$country), ""))

dists <- c("m100", "m200", "m400", "m800", "m1500", "m3000", "marathon")
bx <- list()
bx[[1]] <- ggplot(filter(box_data, distance == eval(dists[1]))) +
    geom_boxplot(aes(x = distance, y = value)) +
    geom_text(aes(x = distance , y = value ,label=outlier), na.rm=TRUE, nudge_x=.05) +
    ylab("Time") +
    theme(axis.title.x = element_blank())

for(i in 2:length(dists)){
```

```
box <- ggplot(filter(box_data, distance == eval(dists[i]))) +
    geom_boxplot(aes(x = distance, y = value)) +
    geom_text(aes(x = distance , y = value ,label=outlier), na.rm=TRUE, nudge_x=.05) +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_blank())
bx[[i]] <- box
}

grid.arrange(bx[[1]],
             bx[[2]],
             bx[[3]],
             bx[[4]],
             bx[[5]],
             bx[[6]],
             bx[[7]],
             nrow = 1)
```



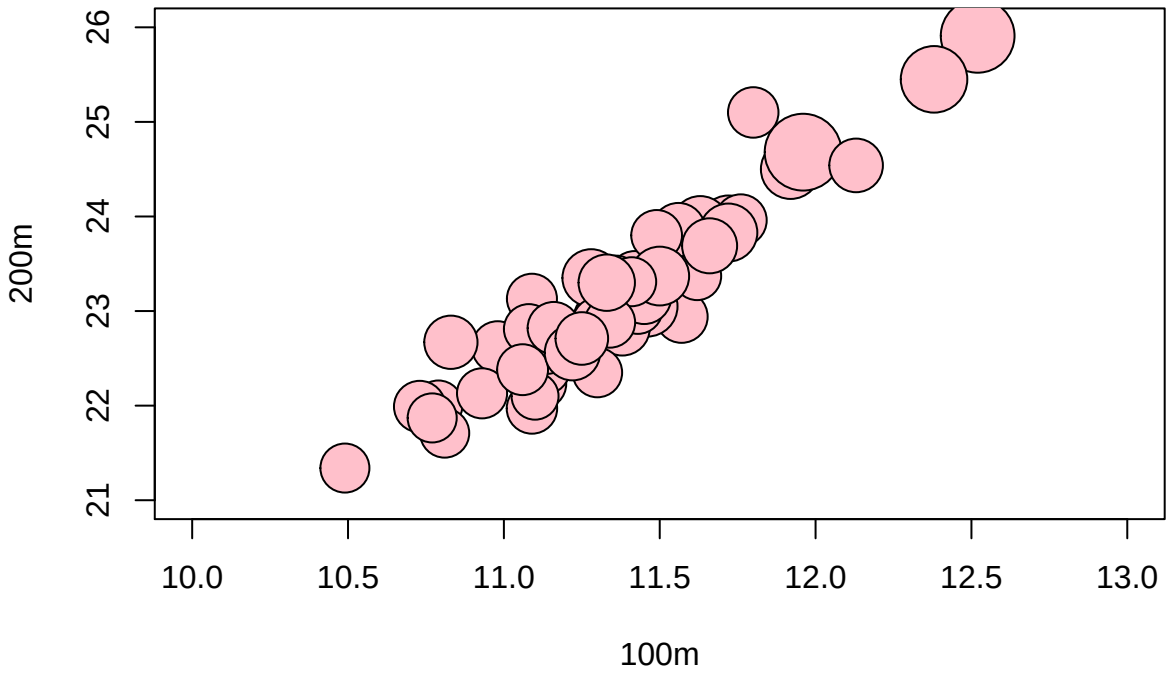
The boxplots above are useful to since one can easily see the spread of the data and if there are any outliers in the data. Here, every graph is computed by itself, due to the fact that the range of the Y-axis differ a lot between difference track distances. If the boxplot would have been combined in one call with the same Y-axis, one would not be able to visually see the spread within each track distance. Be careful when interpreting this graph, the purpose of this graph is not to compare different track distances instead use this graph to see the spread *within* each track distance.

```
#3D plot
scatterplot3d(df$m100,df$m200,df$m400,color="blue",angle=45, xlab = "100m", ylab = "200m", zlab = "400m
```



Below is an example of a 3D graph, were you can compare how the countries differ in a three dimensional space for short track distances (100m, 200m and 400m).

```
#other type of 3D, bubble scatterplot
plot(df$m100,df$m200,pch=1,lwd=2,ylim=c(21,26),xlim=c(10,13), xlab = "100m", ylab = "200m")
symbols(df$m100,df$m200,circles=df$marathon,inches=0.2,add=TRUE,bg="pink",fg="black")
```



Another way of visualizing a third dimension in a 2D graph is to outline the third dimension size or color. Below is an example how a 2D scatterplot between 100m and 200m for each country have got a third dimension, the marathon speed which is represented by the size of the dot, the bigger dot the slower time.

Question 3

Part a

As we can see in our previous analysis (scatterplots, boxplots etc), is that Samoa (SAM), Cook Island (COK), Papua New Guinea (PNG) and Bermuda (BER) are the countries that have extreme values in some distances. This means that their times seem to be further away from the main cluster of other countries times.

Part b

```
# Euclidean distance

X <- as.matrix(df[,2:8])
# Distance
n <- nrow(X)
one_v <- t(rep.int(1,54))
sample_means <-  1/n * (one_v %*% X)
one_mat <- matrix(rep.int(1,54),nrow = 1, ncol =54)
means_mat <- t(t(sample_means) %*% one_mat)
centered_data <- X - means_mat

M <- centered_data %*% t(centered_data)
distances <- sqrt(diag(M))

ind_5 <- which(distances %in% sort(distances, decreasing = TRUE)[1:5])
countries <- df[ind_5,1]

result_b <- data.frame(distances = sort(distances, decreasing = TRUE)[1:5],
                       #country_pos = which(distances %in% sort(distances, decreasing = TRUE)[1:5]),
                       country = countries)
```

Using Euclidean distance we get that these 5 countries have the most extreme distances:

```
##   distances country
## 1  67.62796     BER
## 2  59.61517     COK
## 3  38.52476     GBR
## 4  20.61606     PNG
## 5  18.59146     SAM
```

Part c

```
# Squared distance
# Covariance matrix
a <- diag(n) - (1/n)*t(one_mat)%*%one_mat
cov_mat <- (1/(n-1))* t(X) %*% a %*% X
var_v <- as.vector(diag(cov_mat))
# standard deviantion daigonal matrix
V <- diag(sqrt(var_v))
scaled <- centered_data %*% solve(V)
# Euclidean distance of scaled data
d2 <- scaled %*% t(scaled)
```

```r
distances2 <- sqrt(diag(d2))

ind2_5 <- which(distances2 %in% sort(distances2, decreasing = TRUE)[1:5])
countries2 <- df[ind2_5,1]

result_c <- data.frame(distances = sort(distances2, decreasing = TRUE)[1:5],
                       country = countries2)
```

Using squared distance we get that these 5 countries have the most extreme distances:

```
##   distances country
## 1  8.693837     COK
## 2  8.037485     PNG
## 3  5.850548     SAM
## 4  3.588439     SIN
## 5  3.383026     USA
```

We see that three countries are in the top 5 highest distance tables from parts b and c - PNG, SAM and COK. However, the ranking of them in top 5 has changed. Also the top 2 countries have 2 times higher distance then the bottom two countries from the list, indicating that they have much more extreme times.

Part d

```r
# Mahalanobis distance

S_inv <- solve(cov_mat)

d3 <- centered_data %*% S_inv %*% t(centered_data)

distances3 <- sqrt(diag(d3))
ind3_5 <- which(distances3 %in% sort(distances3, decreasing = TRUE)[1:5])
countries3 <- df[ind3_5,1]

result_d <- data.frame(distances = sort(distances3, decreasing = TRUE)[1:5],
                       country = countries3)
```

Using Mahalanobis distance we get that these 5 countries have the most extreme distances:

```
##   distances country
## 1  5.917268     COK
## 2  5.523337    KORN
## 3  5.115383     MEX
## 4  4.453538     PNG
## 5  3.772391     SAM
```

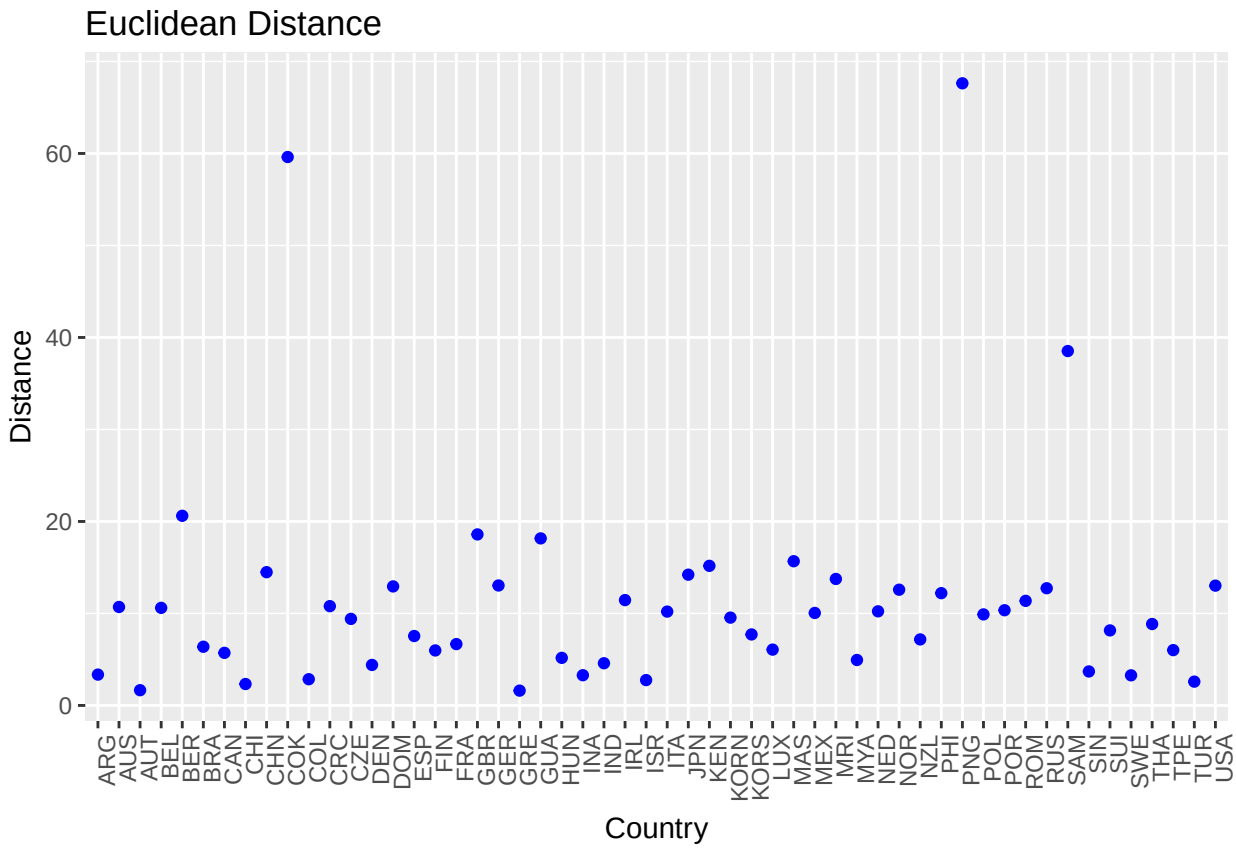
The same three countries are in all three extreme distance tables determined using Euclidean, Squared and Mahalanobis distances - PNG, SAM and COK.

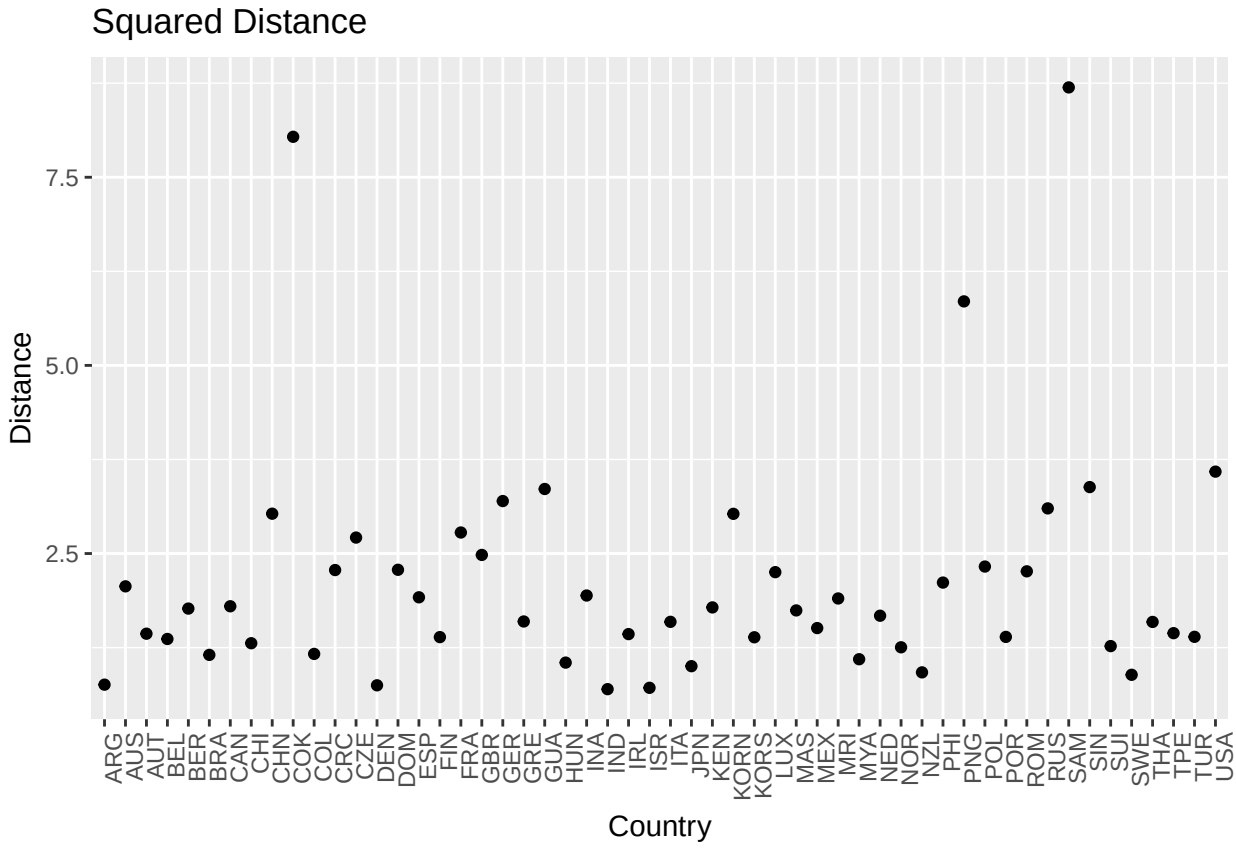
Part e

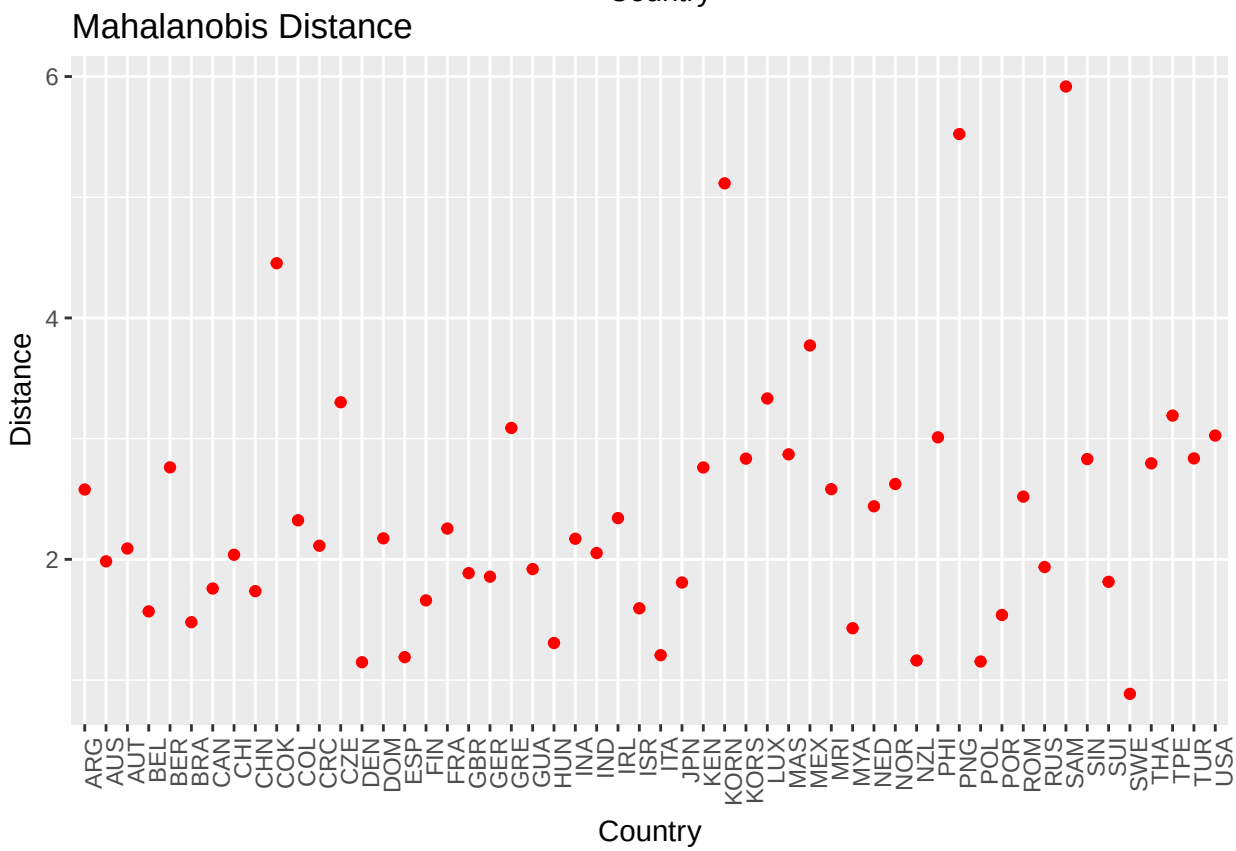
The Euclidean distance centers data, but does not account for the variance. Squared distance scales the centered data by standard deviations. The Mahalanobis distance also includes coveriance into data processing before calculating distance.

```r
ind_full <- which(distances %in% sort(distances, decreasing = TRUE))

countries <- df[ind_full,1]

data <- df
full_result_b <- data.frame(distances = sort(distances, decreasing = TRUE),
                            country_pos = 1:54,
                            country = data[order(distances, decreasing = TRUE),1])

full_result_c <- data.frame(distances = sort(distances2, decreasing = TRUE),
                            country_pos = 1:54,
                            country = data[order(distances2, decreasing = TRUE),1])

full_result_d <- data.frame(distances = sort(distances3, decreasing = TRUE),
                            country_pos = 1:54,
                            country = data[order(distances3, decreasing = TRUE),1])

b_swe_pos <- subset(full_result_b, country == "SWE")
c_swe_pos <- subset(full_result_c, country == "SWE")
d_swe_pos <- subset(full_result_d, country == "SWE")
```

We can observe that the most extreme countries are: Samoa, Cook Island and Papua New Guinea, as they are in top 5 distances for all three distance measurements. From the plots below, other countries distances (excluding the extreme value countries) seem to be clustered closer to each other in Euclidean and Squared distance plots with more distant extreme values. In Mahalanobis distance plot the difference between distances seem to be smaller and the extreme values are less big in comparison to others.







In all the distances, sweden is above de 47 rank. (48 with Euclidean, 50 with Square and 54 with mahalanobis):

```
##    distances country_pos country
## 48  3.275672          48     SWE
```

```
##    distances country_pos country
## 50 0.8886386          50     SWE
```

```
##    distances country_pos country
## 54   0.88578          54     SWE
```

Appendix

```
library(tidyverse)
library(ggplot2)
library(GGally)
library(ggcorrplot)
library(gridExtra)
library(scatterplot3d)
library(knitr)
df <- read.csv("./T1-9.dat",
               sep = "\t", header = FALSE)
names(df) <- c("country", "m100", "m200", "m400", "m800", "m1500", "m3000", "marathon")
tdf <- df %>%
    gather(distance, value, -country) %>%
    mutate(sort_order = case_when(distance == "m100" ~1,
                                  distance == "m200" ~2,
                                  distance == "m400" ~3,
                                  distance == "m800" ~4,
                                  distance == "m1500" ~5,
                                  distance == "m3000" ~6,
                                  distance == "marathon" ~7))

df_summary <- tdf %>%
    group_by(distance) %>%
    summarise(min = min(value),
              max = max(value),
              avg = round(mean(value),6),
              median = round(median(value),6),
              sd = round(sd(value),6),
              variance = round(var(value),6)) %>%
    left_join(., select(tdf, country, value, sort_order), by = c("min" = "value")) %>%
    rename(best = country) %>%
    left_join(., select(tdf, country, value), by = c("max" = "value")) %>%
    rename(worst = country) %>%
    arrange(sort_order) %>%
    select(-sort_order)


kable(df_summary, format = "markdown", digits = 2)
plot1 <- ggplot(df, aes(x=df[, 2], y=row.names(df))) + geom_point(col="white")+
  labs(list(title =colnames(df)[2] , x = "Time")) +
  geom_text(aes(label=df[, 1]), size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
```

```
          axis.ticks.y = element_blank())

plot2 <- ggplot(df, aes(x=df[, 3], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[3] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot3 <- ggplot(df, aes(x=df[, 4], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[4] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot4 <- ggplot(df, aes(x=df[, 5], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[5] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot5 <- ggplot(df, aes(x=df[, 6], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[6] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot6 <- ggplot(df, aes(x=df[, 7], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[7] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

plot7 <- ggplot(df, aes(x=df[, 8], y=row.names(df))) +  geom_point(col="white")+
  labs(list(title =colnames(df)[8] , x = "Time")) +
  geom_text(aes(label=df[, 1]), vjust=1.2, hjust=0.5, size=2) +
    ylab("Country") +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank())

#Plots
grid.arrange(plot1, plot2, ncol=2)
grid.arrange(plot3,plot4, ncol=2)
grid.arrange(plot5, plot6, ncol=2)
grid.arrange(plot7, ncol=2)
par(mfrow=c(2, 4))
for(i in 2:8){
  qqnorm(df[,i],main=colnames(df[i]), col="blue")
  qqline(df[,i], col="red")
```

```
}
boxplot_df <- boxplot(df[,2:8],col="blue",main="Boxplot")
par(mfrow=c(1, 1))
graphs <- list()
for(i in 1:nrow(df_summary)){
    range <- eval(substitute(seq(as.numeric(df_summary[i, "min"]) - as.numeric(df_summary[i, "sd"]), as
    ynorm <- eval(substitute(dnorm(x = range, mean = as.numeric(df_summary[i, "avg"]), sd = as.numeric(
    graphs[[i]] <- eval(substitute(qplot(df[,i+1], geom = "blank") +
        geom_line(aes(y = ..density.., colour = "Empirical"), stat = 'density') +
        geom_line(aes(x = range, y = ynorm, color = "Normal")) +
        geom_histogram(aes(y = ..density..), alpha = 0.4) +
        scale_colour_manual(name = 'Density', values = c('red', 'blue')) +
        theme(legend.position = c(0.85, 0.85)) +
        xlab("Time") +
        ggtitle(names(df[i+1])),list(i=i,range=range,ynorm=ynorm)))
}


grid.arrange(graphs[[1]], graphs[[2]], ncol=2)
grid.arrange(graphs[[3]],graphs[[4]], ncol=2)
grid.arrange(graphs[[5]], graphs[[6]], ncol=2)
grid.arrange(graphs[[7]], ncol=2)

options(scipen=999)  # turn-off scientific notation like 1e+48
corr <- round(cor(df[2:8]), 1)

# Correlation Plot
ggcorrplot(corr, hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           method="circle",
           colors = c("tomato2", "white", "springgreen3"),
           title="Correlation of records",
           ggtheme=theme_bw)
marathonByCountries <- df[,c("country","marathon")]
sample <- head(marathonByCountries,10)

m400ByCountries <- df[,c("country","m400")]
sample <- head(m400ByCountries,20)
# Plot
ggplot(m400ByCountries, aes(x=country, y=m400, label="400m")) +
  geom_point(stat='identity', aes(col=country), size=6)  +
  geom_text(color="white", size=2) +
  labs(title="Dot Plot",
       subtitle="400m comparison by country") +
  theme(axis.text.x=element_text(angle=90,hjust=1))
# 400m COK has extreme value
# it can be produced for every type to find extreme on graph
cor <- cor(df[,2:8])
print("Correlation Matrix:")
print(cor)
cov <- cov(df[,2:8])
```

```
print("Covariance Matrix:")
print(cov)
my_fn <- function(data, mapping, ...){
  p <- ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method=lm, fill="blue", color="blue", ...)
  p
}
ggpairs(df[,2:8],columns = 1:7, lower = list(continuous = my_fn))


is_outlier <- function(x) {
  return(x < quantile(x, 0.25) - 1.5 * IQR(x) | x > quantile(x, 0.75) + 1.5 * IQR(x))
}

box_data <- tdf %>%
    group_by(distance) %>%
    mutate(outlier = is_outlier(value)) %>%
    mutate(outlier = ifelse(outlier == TRUE, as.character(.$country), ""))

dists <- c("m100", "m200", "m400", "m800", "m1500", "m3000", "marathon")
bx <- list()
bx[[1]] <- ggplot(filter(box_data, distance == eval(dists[1]))) +
    geom_boxplot(aes(x = distance, y = value)) +
    geom_text(aes(x = distance , y = value ,label=outlier), na.rm=TRUE, nudge_x=.05) +
    ylab("Time") +
    theme(axis.title.x = element_blank())

for(i in 2:length(dists)){
box <- ggplot(filter(box_data, distance == eval(dists[i]))) +
    geom_boxplot(aes(x = distance, y = value)) +
    geom_text(aes(x = distance , y = value ,label=outlier), na.rm=TRUE, nudge_x=.05) +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_blank())
bx[[i]] <- box
}

grid.arrange(bx[[1]],
             bx[[2]],
             bx[[3]],
             bx[[4]],
             bx[[5]],
             bx[[6]],
             bx[[7]],
             nrow = 1)

#3D plot
scatterplot3d(df$m100,df$m200,df$m400,color="blue",angle=45, xlab = "100m", ylab = "200m", zlab = "400m
#other type of 3D, bubble scatterplot
plot(df$m100,df$m200,pch=1,lwd=2,ylim=c(21,26),xlim=c(10,13), xlab = "100m", ylab = "200m")
symbols(df$m100,df$m200,circles=df$marathon,inches=0.2,add=TRUE,bg="pink",fg="black")
# Euclidean distance
```

```r
X <- as.matrix(df[,2:8])
# Distance
n <- nrow(X)
one_v <- t(rep.int(1,54))
sample_means <-  1/n * (one_v %*% X)
one_mat <- matrix(rep.int(1,54),nrow = 1, ncol =54)
means_mat <- t(t(sample_means) %*% one_mat)
centered_data <- X - means_mat

M <- centered_data %*% t(centered_data)
distances <- sqrt(diag(M))

ind_5 <- which(distances %in% sort(distances, decreasing = TRUE)[1:5])
countries <- df[ind_5,1]

result_b <- data.frame(distances = sort(distances, decreasing = TRUE)[1:5],
                       #country_pos = which(distances %in% sort(distances, decreasing = TRUE)[1:5]),
                       country = countries)

result_b
# Squared distance
# Covariance matrix
a <- diag(n) - (1/n)*t(one_mat)%*%one_mat
cov_mat <- (1/(n-1))* t(X) %*% a %*% X
var_v <- as.vector(diag(cov_mat))
# standard deviantion daigonal matrix
V <- diag(sqrt(var_v))
scaled <- centered_data %*% solve(V)
# Euclidean distance of scaled data
d2 <- scaled %*% t(scaled)
distances2 <- sqrt(diag(d2))

ind2_5 <- which(distances2 %in% sort(distances2, decreasing = TRUE)[1:5])
countries2 <- df[ind2_5,1]

result_c <- data.frame(distances = sort(distances2, decreasing = TRUE)[1:5],
                       country = countries2)

result_c
# Mahalanobis distance

S_inv <- solve(cov_mat)

d3 <- centered_data %*% S_inv %*% t(centered_data)

distances3 <- sqrt(diag(d3))
ind3_5 <- which(distances3 %in% sort(distances3, decreasing = TRUE)[1:5])
countries3 <- df[ind3_5,1]

result_d <- data.frame(distances = sort(distances3, decreasing = TRUE)[1:5],
                       country = countries3)

result_d
```

```r
ind_full <- which(distances %in% sort(distances, decreasing = TRUE))

countries <- df[ind_full,1]

data <- df
full_result_b <- data.frame(distances = sort(distances, decreasing = TRUE),
                            country_pos = 1:54,
                            country = data[order(distances, decreasing = TRUE),1])

full_result_c <- data.frame(distances = sort(distances2, decreasing = TRUE),
                            country_pos = 1:54,
                            country = data[order(distances2, decreasing = TRUE),1])

full_result_d <- data.frame(distances = sort(distances3, decreasing = TRUE),
                            country_pos = 1:54,
                            country = data[order(distances3, decreasing = TRUE),1])

b_swe_pos <- subset(full_result_b, country == "SWE")
c_swe_pos <- subset(full_result_c, country == "SWE")
d_swe_pos <- subset(full_result_d, country == "SWE")


# Euclidean distance plot
 ggplot(full_result_b, aes(x=full_result_b[, 3], y=full_result_b[,1])) +
    geom_point(col="blue")+
    labs(list(title ="Euclidean Distance" , x = "Country", y = "Distance")) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Squared distance plot
 ggplot(full_result_c, aes(x=full_result_c[, 3], y=full_result_c[,1])) +
    geom_point(col="black")+
    labs(list(title ="Squared Distance" , x = "Country", y = "Distance")) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))

 # Mahalanobis distance plot
 ggplot(full_result_c, aes(x=full_result_d[, 3], y=full_result_d[,1])) +
    geom_point(col="red")+
    labs(list(title ="Mahalanobis Distance" , x = "Country", y = "Distance")) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))

b_swe_pos
c_swe_pos
d_swe_pos
```