Multivariate Statistical Methods 732A97

Lab 1

Albin Västerlund
albva223

Eric Herwin
Erihe068

Balaji Ramkumar
Balra340

Guilherme Barros
Guiba484

Contents

Question 1: Describing individual variables

1.a)

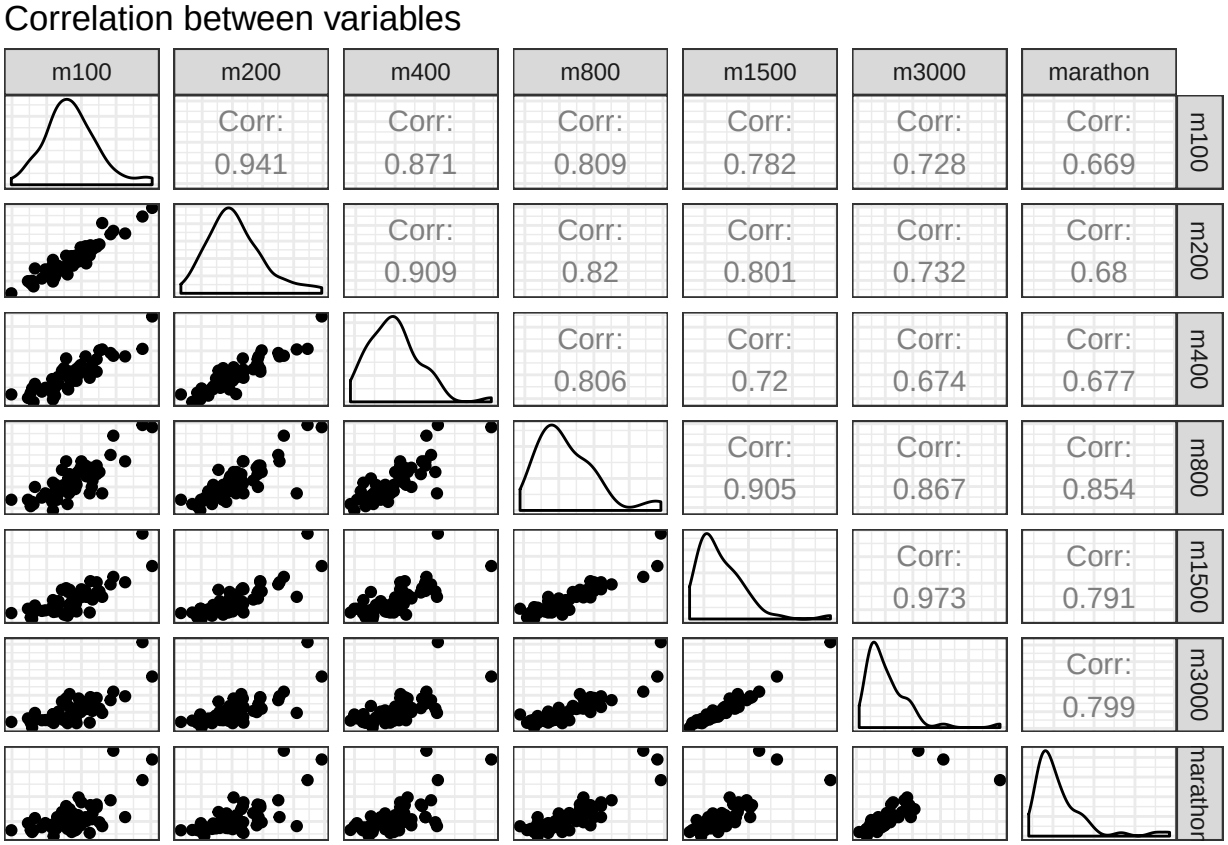
In this task we will describe the 7 variables with desctriptive statistics. The data for 800m - marathon are expressed in minutes.

variable	n	var	min	quantile_25	median	mean	quantile_75	max
m100	54	0.1553157	10.49	11.1225	11.325	11.357778	11.5675	12.52
m200	54	0.8630883	21.34	22.5700	22.980	23.118519	23.6100	25.91
m400	54	6.7454576	47.60	49.9675	51.645	51.989074	53.1175	61.65
m800	54	0.0075469	1.89	1.9700	2.005	2.022407	2.0700	2.29
m1500	54	0.0741827	3.84	4.0025	4.100	4.189444	4.3375	5.42
m3000	54	0.6647579	8.10	8.5425	8.845	9.080741	9.3250	13.12
marathon	54	270.2701504	135.25	143.4800	148.430	153.619259	157.6650	221.14

From the table we can conclude that running a marathon takes the longest time.

1.b)

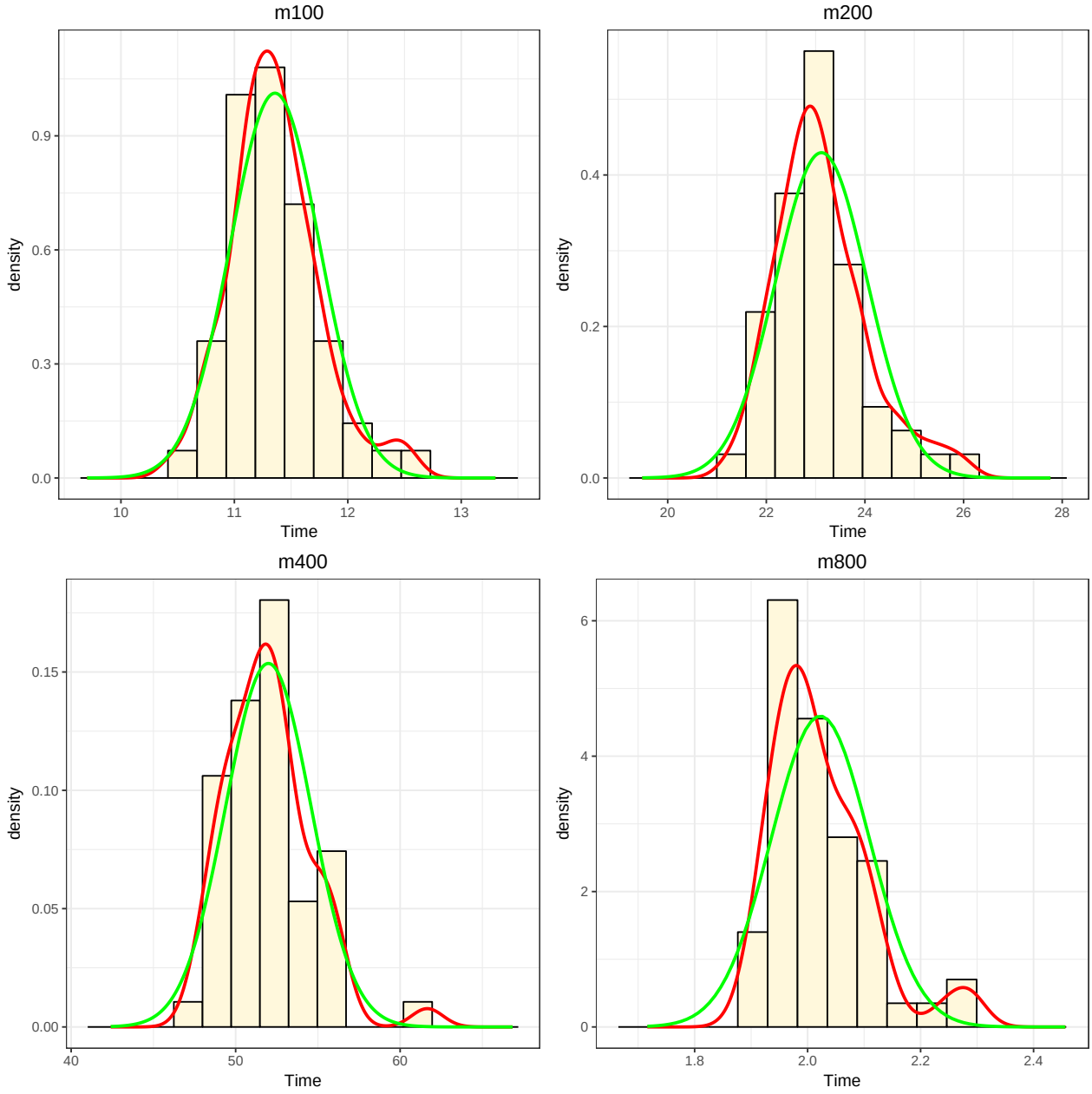
In this task we will illustrate the variables with different graphs.

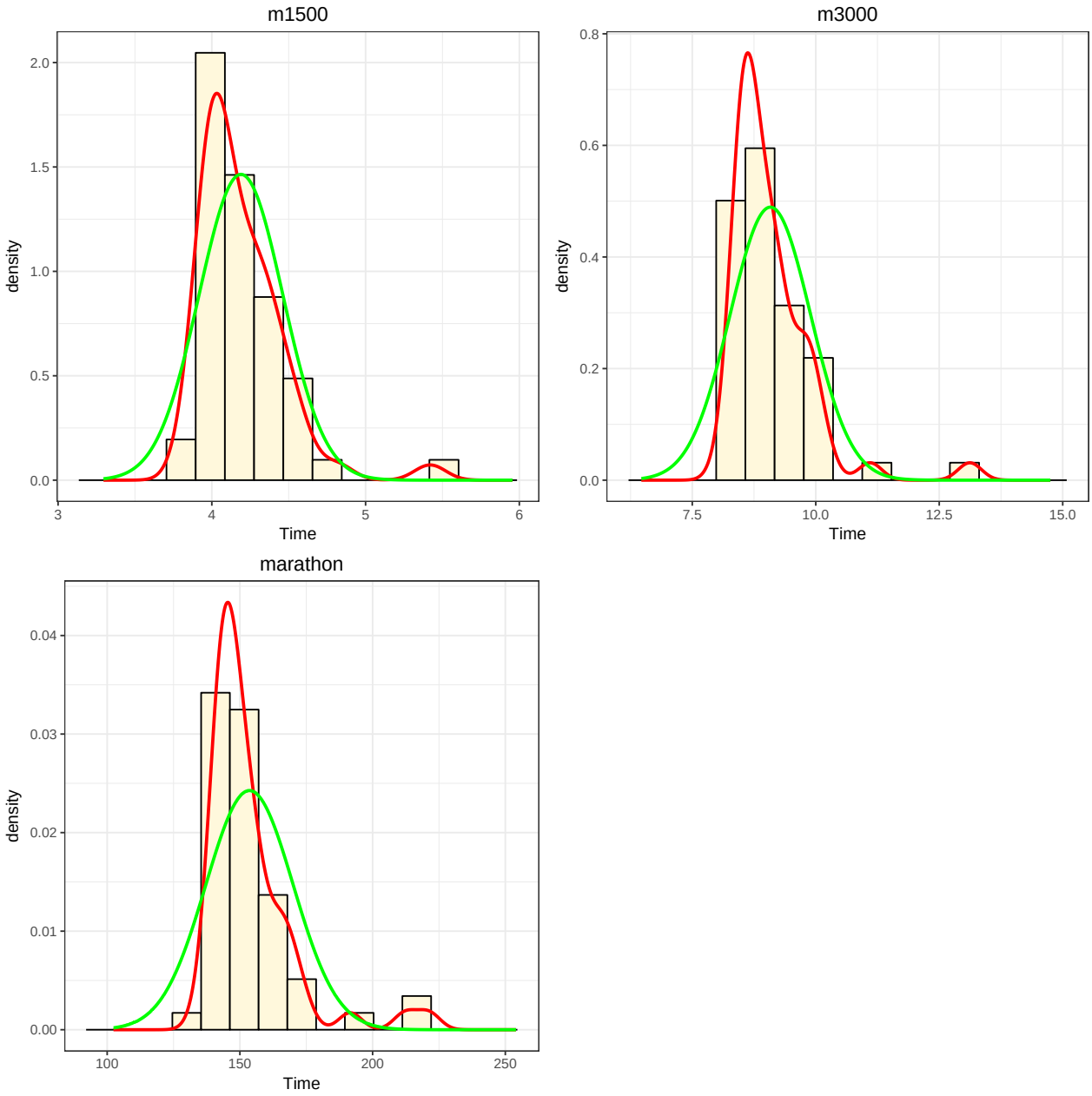


We can see that for some combinations of variables, there are some points which seem to be ouliers.

To investigate more on how the variables are distributed, we plot all the variables in histograms. The green

line is the density curve for a normal disturbution while the red line is the estimated density for the data with method kenal.





We can see that the times for the 100m and 200m are close to a normal distribution while the distribution for the time for the marathon does not seem to be a normal distribution.

Question 2: Relationships between the variables

2.a)

In this task we will compute the correlation and the covariance matrices for the 7 variables.

```
# Correlation matrix
kable(round(cor(my_data[2:8]),2))
```

	m100	m200	m400	m800	m1500	m3000	marathon
m100	1.00	0.94	0.87	0.81	0.78	0.73	0.67
m200	0.94	1.00	0.91	0.82	0.80	0.73	0.68
m400	0.87	0.91	1.00	0.81	0.72	0.67	0.68
m800	0.81	0.82	0.81	1.00	0.91	0.87	0.85
m1500	0.78	0.80	0.72	0.91	1.00	0.97	0.79
m3000	0.73	0.73	0.67	0.87	0.97	1.00	0.80
marathon	0.67	0.68	0.68	0.85	0.79	0.80	1.00

From the output we can see that all the variables are highly correlated with each other. The closer the distance is the higher is the correlation bettwen the variables.

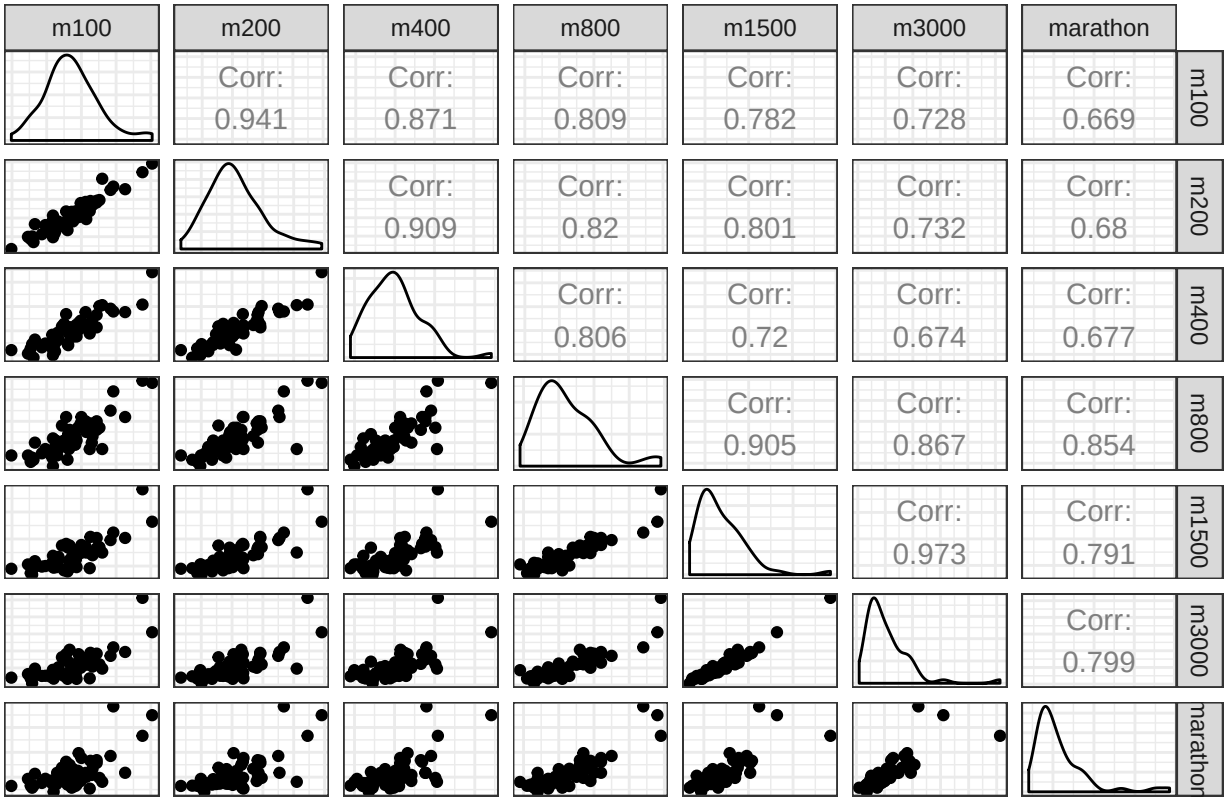
```
# Covariance matrix
kable(round(cov(my_data[2:8]),2))
```

	m100	m200	m400	m800	m1500	m3000	marathon
m100	0.16	0.34	0.89	0.03	0.08	0.23	4.33
m200	0.34	0.86	2.19	0.07	0.20	0.55	10.38
m400	0.89	2.19	6.75	0.18	0.51	1.43	28.90
m800	0.03	0.07	0.18	0.01	0.02	0.06	1.22
m1500	0.08	0.20	0.51	0.02	0.07	0.22	3.54
m3000	0.23	0.55	1.43	0.06	0.22	0.66	10.71
marathon	4.33	10.38	28.90	1.22	3.54	10.71	270.27

In this table we can also see that the the variance gets higher as the distance becomes greater. However, as the data for 800m and marathon variables are expressed in minutes, the variance is smaller.

2.b)

For this step we will study the variables with a scatterplot towards eachother. This plot is the same as the plot in 1.b).



We can see, as previously, that it seems some points can be considered as outliers/extreme values for different variables.

2.c)

In the task we will explore two other plotting possibilities for multivariate data.

We start exploring how the contries that run "fast" in 100m seems to differ from the countries that are "slow" on 100m and in other distance.

It seems like the countries that are "slow" in 100m seems to have better results in longer distance races.

Humans are good to recognize faces, so we also do a face plot to see which countries seem to have more equal values on the 7 variables.

Face plot


```
## effect of variables:
##  modified item       Var
##  "height of face   " "m100"
##  "width of face    " "m200"
##  "structure of face" "m400"
##  "height of mouth  " "m800"
##  "width of mouth   " "m1500"
##  "smiling          " "m3000"
##  "height of eyes   " "marathon"
##  "width of eyes    " "m100"
##  "height of hair   " "m200"
##  "width of hair    "  "m400"
##  "style of hair    "  "m800"
##  "height of nose   "  "m1500"
##  "width of nose    "  "m3000"
##  "width of ear     "  "marathon"
```

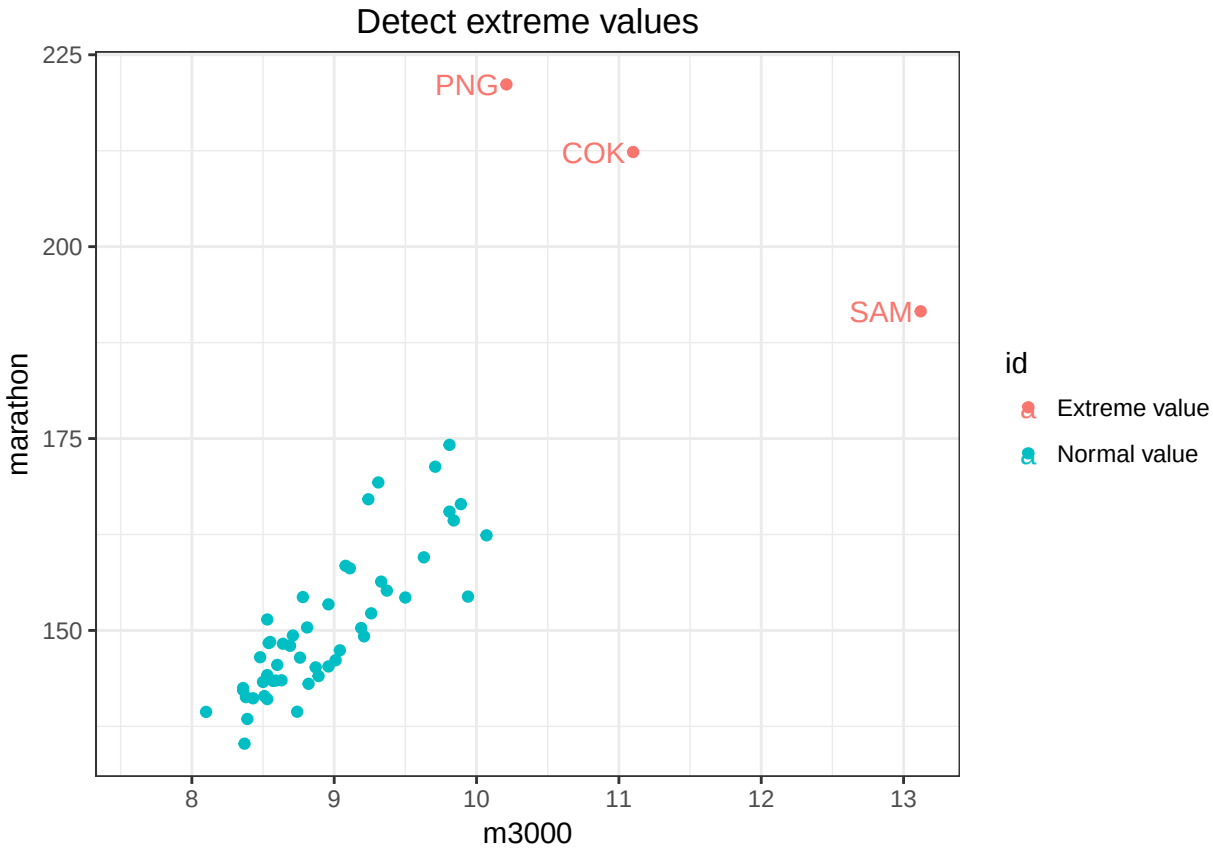
```
##  "height of ear    "  "m100"
```

It seems that the countries COK, GUA, PNG and SAM differ from the other countries.

Question 3: Examining for extreme values

3.a)

In this task we will see which countries are the extreme values. We have seen that in the previous scatterplot-matrices that some points can be seen as outliers for different variables.



If we look at `marathon` against `m300` we can see that there are three possible outliers. Theese extreme values are the countries PNG, COK and SAM. They are considered as extreme values because they seperate themselves from the rest of the data clearly.

To better understand and summarize this information, we will calculate the distances between countries and the mean.

3.b)

We first start by computing the Euclidian Distance between countries and the mean:

```
my_data3b<-as.matrix(my_data[2:8])
my_data3b<-scale(my_data3b,scale=FALSE)

# Same answer
```

```r
my_data$euclidean_centring<-sqrt(diag(my_data3b%*%t(my_data3b)))

# Bigest euclidean distance
big_5<-order(my_data$euclidean_centring,decreasing = TRUE)[1:5]


min_data_frame<-data.frame(euclidean_distance_centring=paste(my_data$country[big_5],"has distance",
                                                             round(my_data$euclidean_centring[big_5],2)))
kable(min_data_frame,col.names = "Centred data")
```

Centred data
PNG has distance 67.63
COK has distance 59.62
SAM has distance 38.52
BER has distance 20.62
GBR has distance 18.59

We have the output where the five countries with the largest values. We can see that the countries from the exercise 1.a) are the top three in this list.

3.c)

In this task we will compute the distances, but with a matrix $\mathbf{V}$ and where this matrix have the variances on the diagonal.

```r
my_data3c<-as.matrix(my_data[2:8])
my_data3c<-scale(my_data3c)

# Same answer
my_data$euclidean_standardizing<-sqrt(diag(my_data3c%*%t(my_data3c)))

# Bigest euclidean distance
big_5<-order(my_data$euclidean_standardizing,decreasing = TRUE)[1:5]


min_data_frame$euclidean_distance_standardizing<-
  paste(my_data$country[big_5],"has distance",
        round(my_data$euclidean_standardizing[big_5],2))

kable(min_data_frame,col.names = c("Euclidean distance","Euclidean distance (standardized data)") )
```

Euclidean distance	Euclidean distance (standardized data)
PNG has distance 67.63	SAM has distance 8.69
COK has distance 59.62	COK has distance 8.04
SAM has distance 38.52	PNG has distance 5.85
BER has distance 20.62	USA has distance 3.59
GBR has distance 18.59	SIN has distance 3.38

For this standardized Eucledian distances we can still conclude that the SAM, COK and PNG are the biggest outliers.

3.d)

For this task we will compute the Mahalanobis distance that takes the covariance of the original matrix into consideration.

```
my_data3d<-as.matrix(my_data[2:8])
my_data3d<-scale(my_data3d,scale=FALSE)

C<-cov(my_data3d)
C_inv<-solve(C)
my_data$mahalonobis_dist<- sqrt(diag(my_data3d%*%C_inv%*%t(my_data3d)))

# Bigest euclidean distance
big_5<-order(my_data$mahalonobis_dist,decreasing = TRUE)[1:5]

# Data frame
min_data_frame$mahalonobis_distance<-
  paste(my_data$country[big_5],"has distance",
        round(my_data$mahalonobis_dist[big_5],2))


kable(min_data_frame,col.names = c("Euclidean distance","Euclidean distance (normalized data)","Mahalanc
```

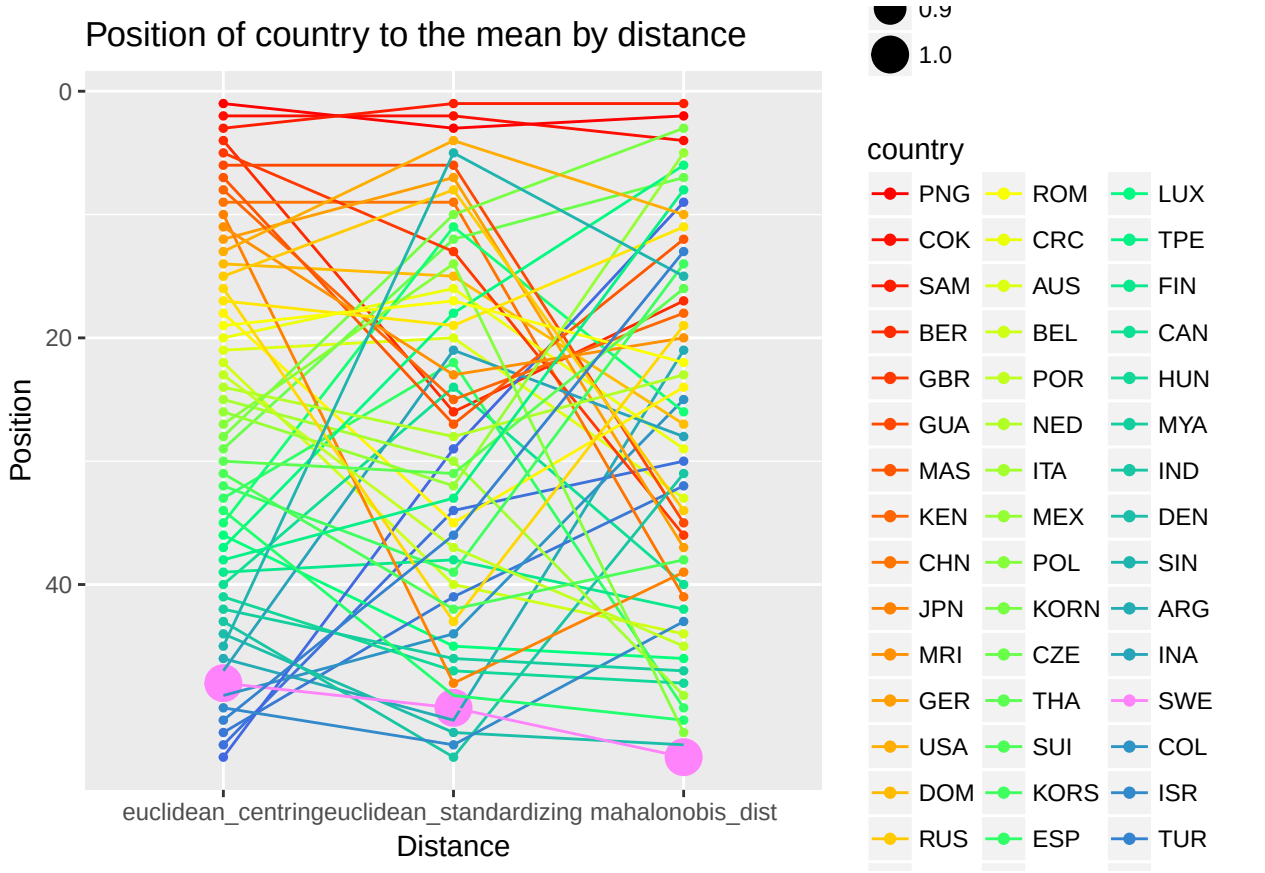
Euclidean distance	Euclidean distance (normalized data)	Mahalanobis distance
PNG has distance 67.63	SAM has distance 8.69	SAM has distance 5.92
COK has distance 59.62	COK has distance 8.04	PNG has distance 5.52
SAM has distance 38.52	PNG has distance 5.85	KORN has distance 5.12
BER has distance 20.62	USA has distance 3.59	COK has distance 4.45
GBR has distance 18.59	SIN has distance 3.38	MEX has distance 3.77

We can still see that the SAM, PNG and COK are in the still in the list of most extreme values, but KORN has climbed up to the top5 when we consider mahalanobis distance

3.e)

In this task we will compare the results from b)–d) and see how different distances affect Sweden.

```
ggplot(country_positions_df_2
       ,aes(x=variable, y=value, group=country,colour=country)) +
  geom_point(mapping = aes(size=size_2)) +
  stat_summary(fun.y=sum, geom="line")+
  scale_colour_manual(values=colors,
                      limits=country_list)+
  ggtitle("Position of country to the mean by distance") +
  xlab("Distance") + ylab("Position") + scale_y_reverse()
```




```
#print(my_data)
```

And we can see except for a few cases, most countries tend to keep their position (first country is the most distant to the mean). Also, we can see that Sweden (the observation marked with large dots) is one of the countries consistently close to the mean, which makes it a very lagom country in running sports.