Lab1 Examining multivariate data

Andreas C Charitos [andch552]

1/22/2021

Contents

Problem 1

Data Overview

Table 1: First 3 rows of the data

country	100m	200m	400m	800m	1500m	3000m	marathon
ARG	11.57	22.94	52.50	2.05	4.25	9.19	150.32
AUS	11.12	22.23	48.63	1.98	4.02	8.63	143.51
AUT	11.15	22.70	50.62	1.94	4.05	8.78	154.35

a)

Compute the means, the variances and the standard deviations for all variables.

Table of column means

Table 2: Column means

	x
100m	11.357778
200m	23.118519
400m	51.989074
800m	2.022407
1500m	4.189444
3000m	9.080741
marathon	153.619259

Table of column variances

Table 3: Column variances

	x
100m	0.1553157
200m	0.8630883
400m	6.7454576
800m	0.0075469
1500m	0.0741827
3000m	0.6647579
marathon	270.2701504

Table of column standard deviations

Table 4: Column standard deviations

	x
100m	0.3941012

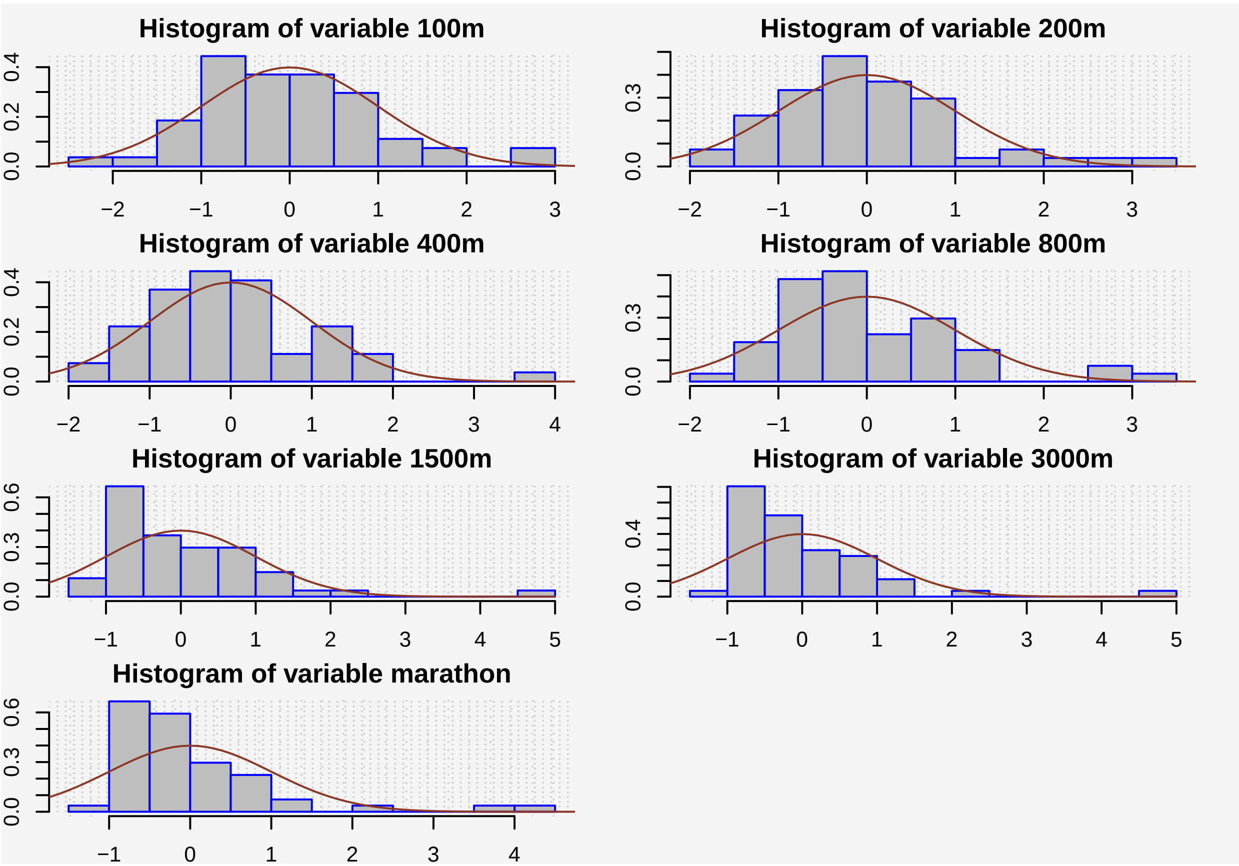
	x
200m	0.9290255
400m	2.5972019
800m	0.0868730
1500m	0.2723650
3000m	0.8153269
marathon	16.4398951

The tables above provide an overview of the column mean, variance and standard deviations for all the variables.

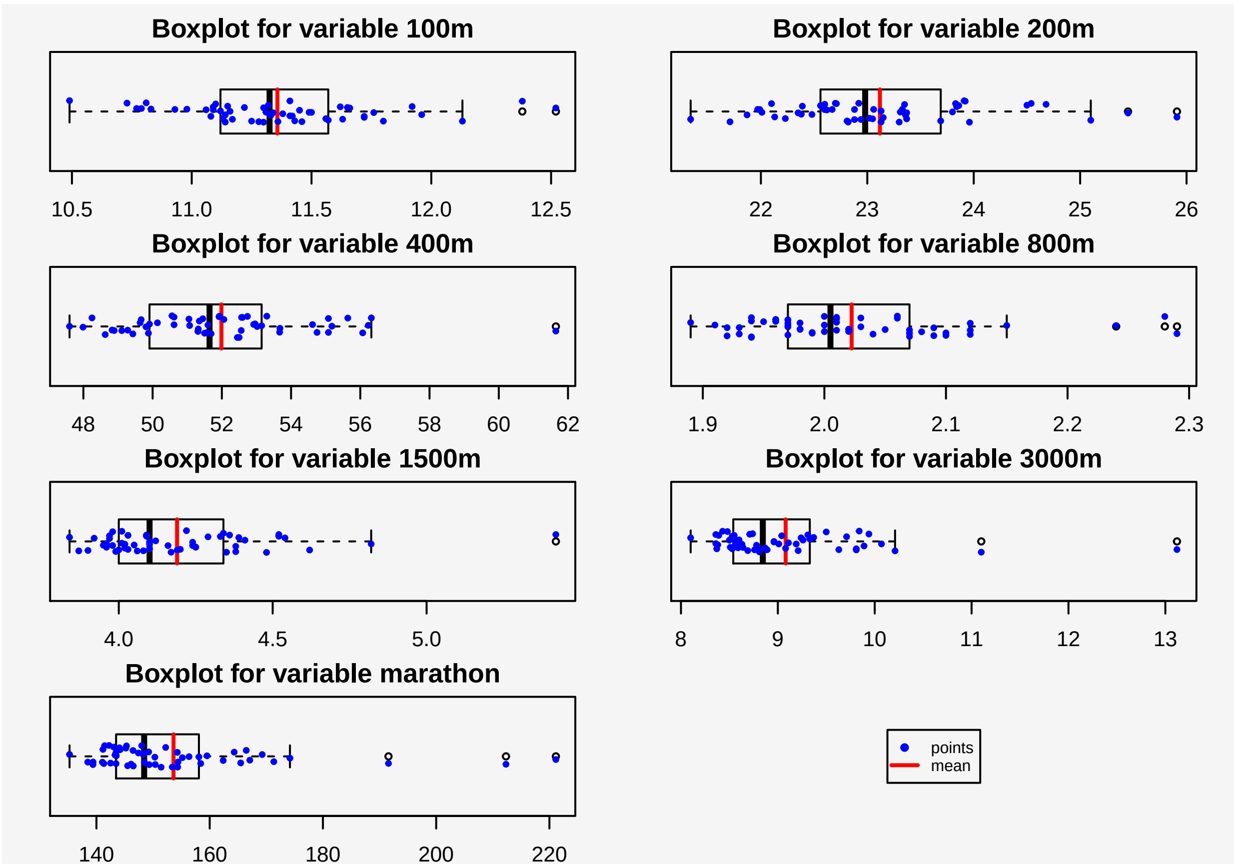
b)

Illustrate the variables using box-plots and histograms. Do the variables look normally distributed? Justify your answer.

Histograms



Boxplots



For all of the variables there seems to be outliers towards their upper quantiles. The closest variable to a normal distribution is the variable 100. Another issue with the data is that is truncated and thus by construction doesn't have the same domain of a normal distribution over the random variable.

Problem 2

a)

Compute the covariance and correlation matrices for the 7 variables.

Correlation matrix

Table 5: Correlation matrix

	100m	200m	400m	800m	1500m	3000m	marathon
100m	1.0000000	0.9410886	0.8707802	0.8091758	0.7815510	0.7278784	0.6689597
200m	0.9410886	1.0000000	0.9088096	0.8198258	0.8013282	0.7318546	0.6799537
400m	0.8707802	0.9088096	1.0000000	0.8057904	0.7197996	0.6737991	0.6769384
800m	0.8091758	0.8198258	0.8057904	1.0000000	0.9050509	0.8665732	0.8539900
1500m	0.7815510	0.8013282	0.7197996	0.9050509	1.0000000	0.9733801	0.7905565
3000m	0.7278784	0.7318546	0.6737991	0.8665732	0.9733801	1.0000000	0.7987302

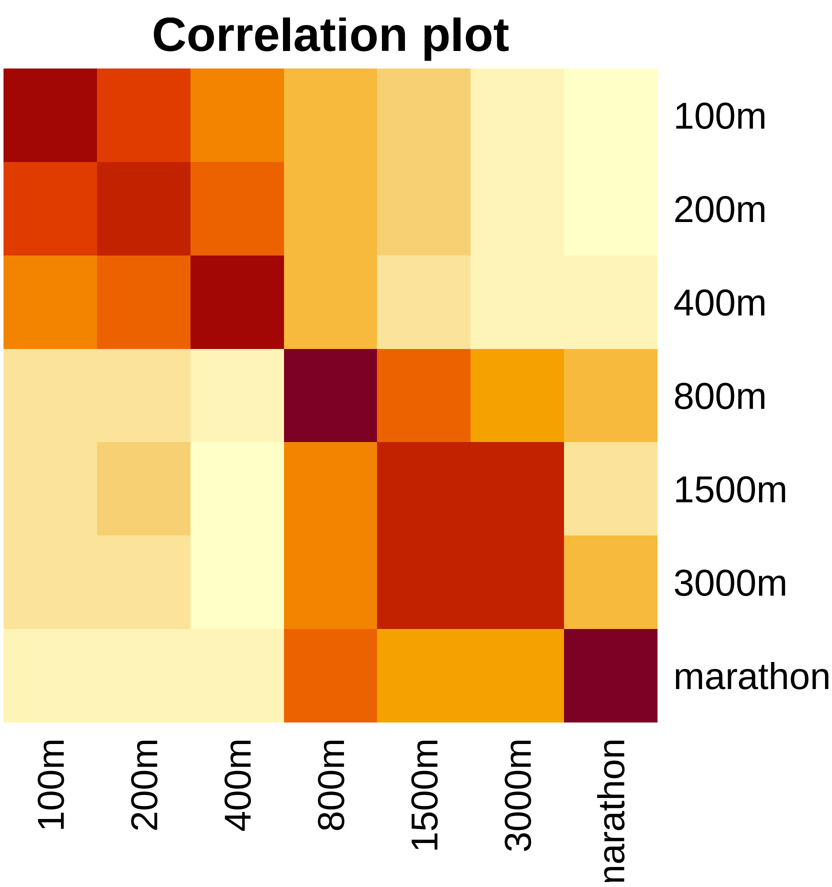
	100m	200m	400m	800m	1500m	3000m	marathon
marathon	0.6689597	0.6799537	0.6769384	0.8539900	0.7905565	0.7987302	1.0000000

Covariance matrix

Table 6: Covariance matrix

	100m	200m	400m	800m	1500m	3000m	marathon
100m	0.1553157	0.3445608	0.8912960	0.0277036	0.0838912	0.2338828	4.334178
200m	0.3445608	0.8630883	2.1928363	0.0661659	0.2027633	0.5543502	10.384988
400m	0.8912960	2.1928363	6.7454576	0.1818079	0.5091768	1.4268158	28.903731
800m	0.0277036	0.0661659	0.1818079	0.0075469	0.0214146	0.0613793	1.219655
1500m	0.0838912	0.2027633	0.5091768	0.0214146	0.0741827	0.2161551	3.539837
3000m	0.2338828	0.5543502	1.4268158	0.0613793	0.2161551	0.6647579	10.706091
marathon	4.3341776	10.3849876	28.9037314	1.2196546	3.5398373	10.7060911	270.270150

Heatmap

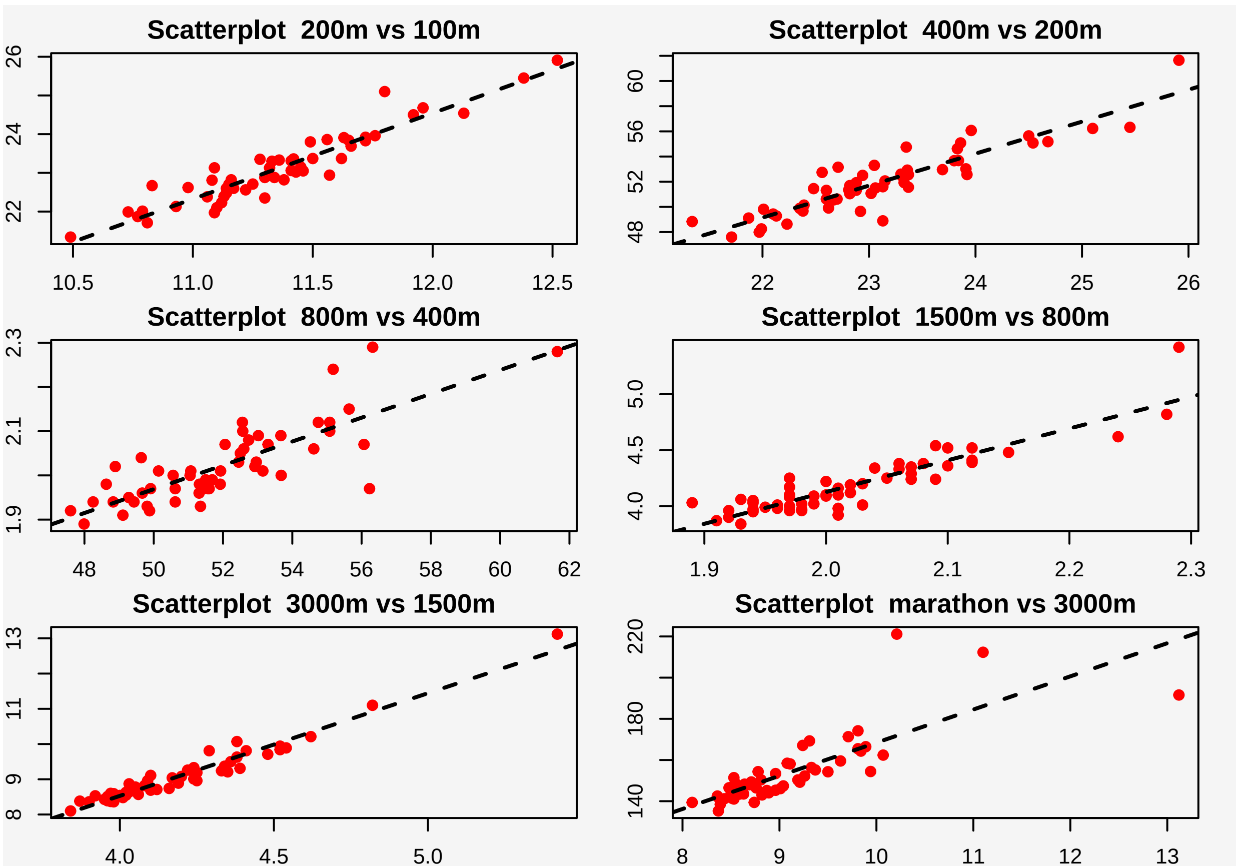


From the heatmap plot above we can conclude that we have two groups of tracks that are cmore correlated than the others. One group is formed by the: {100m, 200m and 400} ond the other group from the {800m, 1500m, 3000m and the marathon} variables.

b)

Illustrate the relations between variables for the 6 pairs: $(x_1; x_2), (x_2; x_3), (x_3; x_4), (x_4; x_5), (x_5; x_6) and (x_6; x_7)$, using scatterplots. Do you observe some extreme values?

Scatterplots

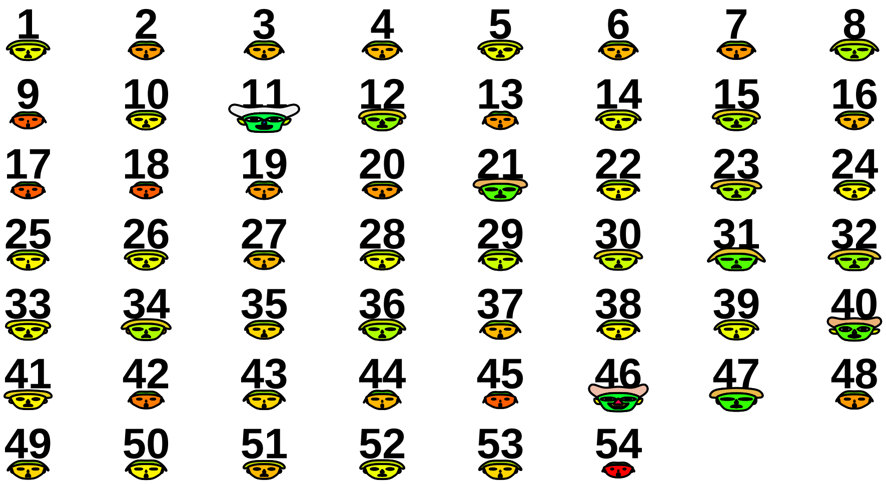


As we can see from the above scatterplots the extreme values are present always on the upper quantiles of each variables. Meaning that there are some countries that excels on each category.

c)

Which other plotting possibilities for multivariate data you know? Present at least one of them for the given data set. Why did you choose this graph?

Chernoff faces plot




```
## effect of variables:
##  modified item       Var
##  "height of face   " "100m"
##  "width of face    " "200m"
##  "structure of face" "400m"
##  "height of mouth  " "800m"
##  "width of mouth   " "1500m"
##  "smiling          " "3000m"
##  "height of eyes   " "marathon"
##  "width of eyes    " "100m"
##  "height of hair   " "200m"
##  "width of hair   "  "400m"
##  "style of hair   "  "800m"
##  "height of nose  "  "1500m"
##  "width of nose   "  "3000m"
##  "width of ear    "  "marathon"
##  "height of ear   "  "100m"
```

Problem 3

In problem 2, b) you observed some extreme values. Which countries look the most extreme? One of the possibilities to answer this question is to compute a distance between an observation and the sample mean vector (to look how far an observation is from the average). Compute the Euclidean distances of observations from the sample mean for all countries. Which 3 countries are the most extreme?

Top 3 most extreme countries

The euclidean distance is defined as :

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

The distance can be immediately generalized to the $L^r, r > 0$ distance as

$$d_{L^r}(\vec{x}, \bar{x}) = \left(\sum_{i=1}^{p} |\vec{x}_i - \bar{x}_i|^r \right)^{1/r}$$

where p is the dimension of the osbervation (here $p = 7$).

Table of extreme countries

countries	most_extreme
PNG	1
COK	2
SAM	3
BER	4
GBR	5

Apprendix

```r
## ----message=FALSE,echo=FALSE-------------------------------------------------
# Import libraries -----------------------------------------------------------
library(ggplot2)
library(GGally)
library(reshape)
# library(kableExtra)
library(knitr)
library(dplyr)
library(plotly)
library(RColorBrewer)


## ---- echo=FALSE--------------------------------------------------------------
dt = read.delim("T1-9.dat", header=FALSE)

colnames(dt) = c('country', '100m', '200m', '400m', '800m', '1500m', '3000m','marathon')

kable(dt[1:3,],
      caption = "First 3 rows of the data")



## ----echo=F-------------------------------------------------------------------
col_means = sapply(dt[, -1], mean)
kable(col_means,
       caption = "Column means")



## ----echo=F-------------------------------------------------------------------
col_sd = sapply(dt[, -1], sd)
kable(col_sd,
      caption = "Column standard deviations")



## ----echo=F-------------------------------------------------------------------
# Histograms
# Values for the normal distribution.

x = seq(-5, 5, 0.1)
y = dnorm(x)
par(mar=rep(2,4))
par(mfrow=c(4,2), bg='whitesmoke')
for (i in 2:8){
  hist(scale(dt[, i]),
       freq=FALSE,
       breaks=10,
       main=paste('Histogram of variable', colnames(dt)[i]),
       col='gray',
       border='blue', panel.first = grid(25,25))
```

```r
  lines(x, y, col='tomato4')
}


## ----echo=F--------------------------------------------------------------
# Boxplots
par(mar=rep(2,4))
par(mfrow=c(4,2), bg='whitesmoke')
for(i in 2:9){
  if(i!=9){
  boxplot(dt[, i], horizontal = TRUE,
          main = paste('Boxplot for variable', colnames(dt)[i]))
  # Add mean line
  segments(x0 = mean(dt[, i]), y0 = 0.8,
           x1 = mean(dt[, i]), y1 = 1.2,
           col = "red", lwd = 2)
  # Add mean point
  # points(mean(dt[, i]), 1, col = 3, pch = 19, cex=2)
  stripchart(dt[, i], method = "jitter",
             pch = 19, add = TRUE,
             col = "blue", cex =0.5)}else{
    par(mai=c(0,0,0,0))
    plot.new()
    legend('center',legend=c('points','mean'),
           col=c('blue', 'red'), pch=c(19, NA),
           lwd=c(NA, 2), cex=0.7)
  }

}



## ----echo=FALSE----------------------------------------------------------

# a) ---------------------------------------------
# calculate matrices
corr_mat=cor(dt[, 2:8]) ; cov_mat=cov(dt[, 2:8])
# print correlation mat
# print(corr_mat)
kable(corr_mat,
      caption = "Correlation matrix")
# print covariance mat
# print(cov_mat)
kable(cov_mat,
      caption = "Covariance matrix")



## ----echo=FALSE----------------------------------------------------------

# b) ---------------------------------------------
par(mfrow=c(3,2), bg='whitesmoke')
for(i in 2:7){
```

```r
  name1=colnames(dt)[i+1]
  name0=colnames(dt)[i]
  title=paste0(name1," vs ",name0)
  # print(title)
  plot(dt[, i], dt[, i+1],
       xlab=colnames(dt)[i], ylab=colnames(dt)[i+1],
       col='red', pch =19,
       main=paste("Scatterplot ", title))
  lm_model=lm(dt[,i+1]~dt[,i], data=dt)
  abline(lm_model,lty=2, lwd=2)
  }



## ----echo=FALSE-------------------------------------------------------------

# c) ---------------------------------------------
my_cols= colorRampPalette(brewer.pal(8, "PiYG"))(25)
heatmap(as.matrix(dt[, 2:8]), labRow=dt$country, scale='column', col = my_cols)




## ----echo=FALSE-------------------------------------------------------------

euclidean_dist=function(X){
  X_centered=sweep(X, 2, colMeans(X))
  X_dist=sqrt(diag(X_centered %*% t(X_centered)))
return(X_dist)
}

distances_ed = euclidean_dist(as.matrix(dt[, 2:8]));
idxs = sort(distances_ed, decreasing=TRUE, index.return=TRUE)$ix;
countries = dt$country[idxs[1:5]]
kable(as.data.frame(countries))



## ----code=readLines(knitr::purl("/home/quartermaine/Desktop/multivariate_statistical_methods-732A97/l
## NA
```