Lab1

Andreas C Charitos[andch552]

1/22/2021

Contents

Problem 1

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)

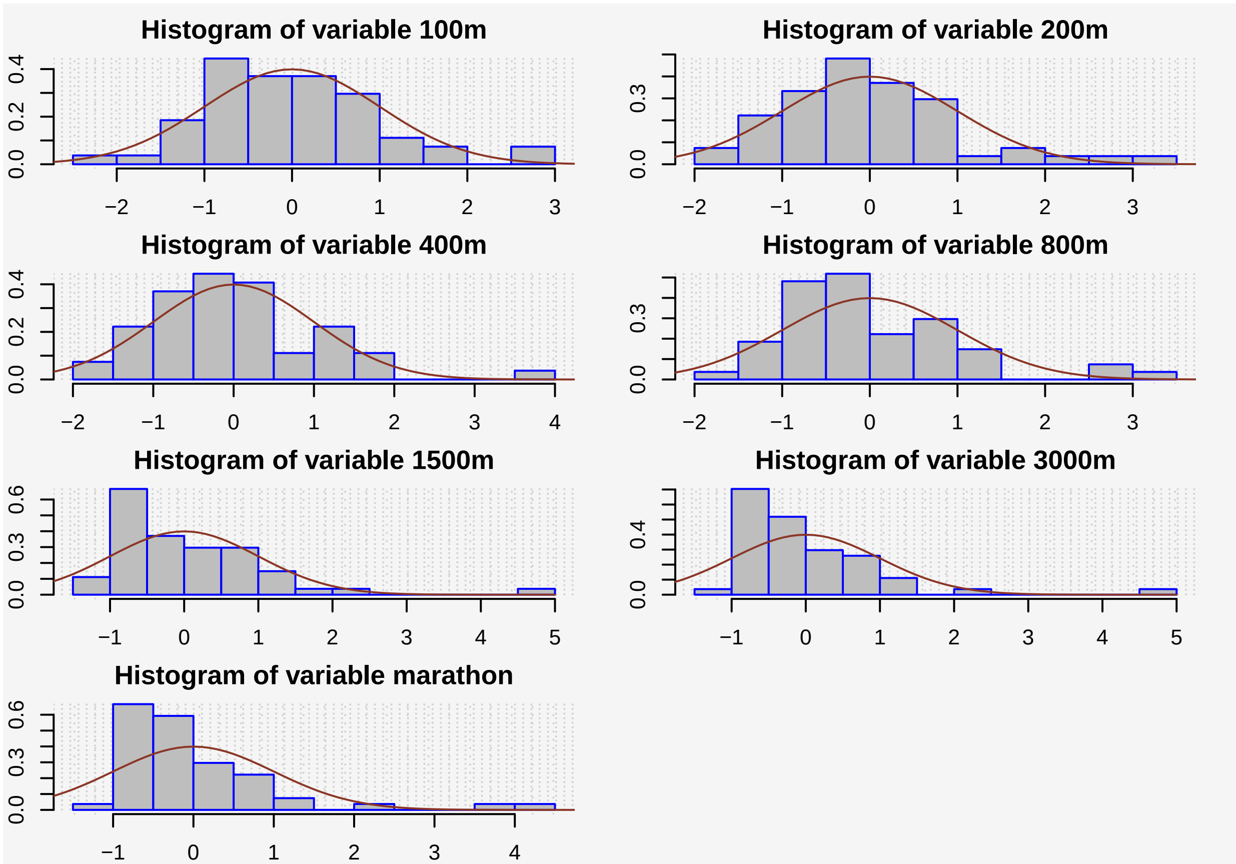
Table 2: Column means

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

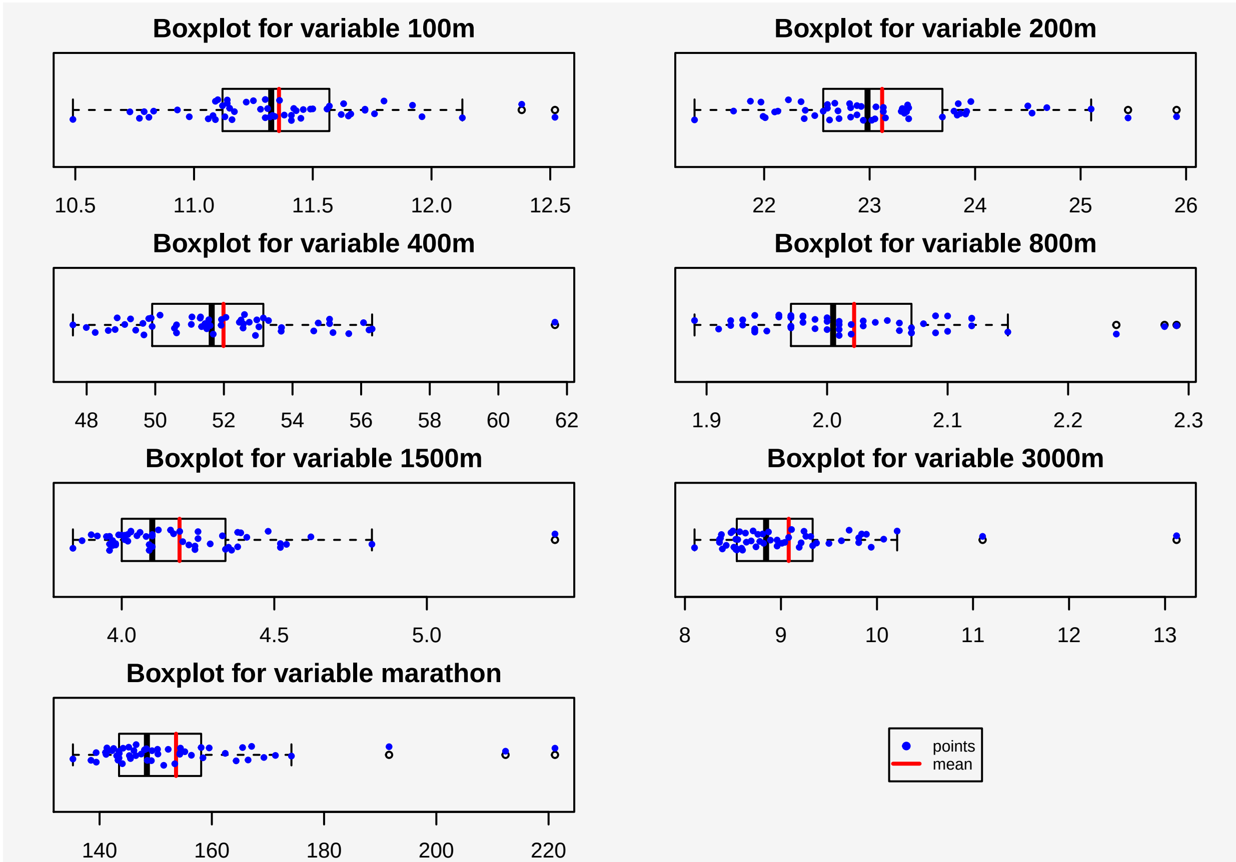
Table 3: Column standard deviations

	x
100m	0.3941012
200m	0.9290255
400m	2.5972019
800m	0.0868730
1500m	0.2723650
3000m	0.8153269
marathon	16.4398951

b)



Histogram of variable 100m
Histogram of variable 200m
Histogram of variable 400m
Histogram of variable 800m
Histogram of variable 1500m
Histogram of variable 3000m
Histogram of variable marathon



Problem 2

a)

Table 4: Correlation matrix

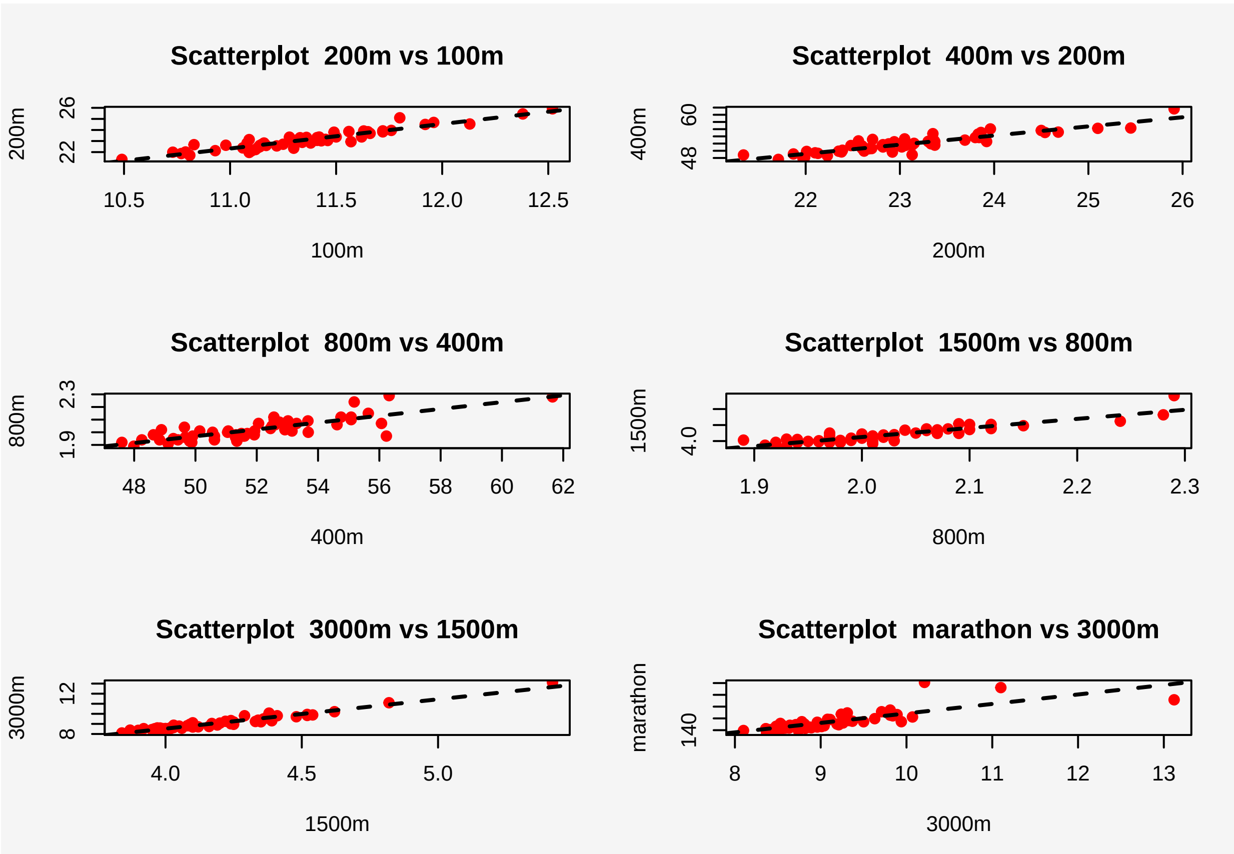
	100m	200m	400m	800m	1500m	3000m	marathon
100m	1.0000000	0.9410886	0.8707802	0.8091758	0.7815510	0.7278784	0.6689597
200m	0.9410886	1.0000000	0.9088096	0.8198258	0.8013282	0.7318546	0.6799537
400m	0.8707802	0.9088096	1.0000000	0.8057904	0.7197996	0.6737991	0.6769384
800m	0.8091758	0.8198258	0.8057904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

Table 5: Covariance matrix

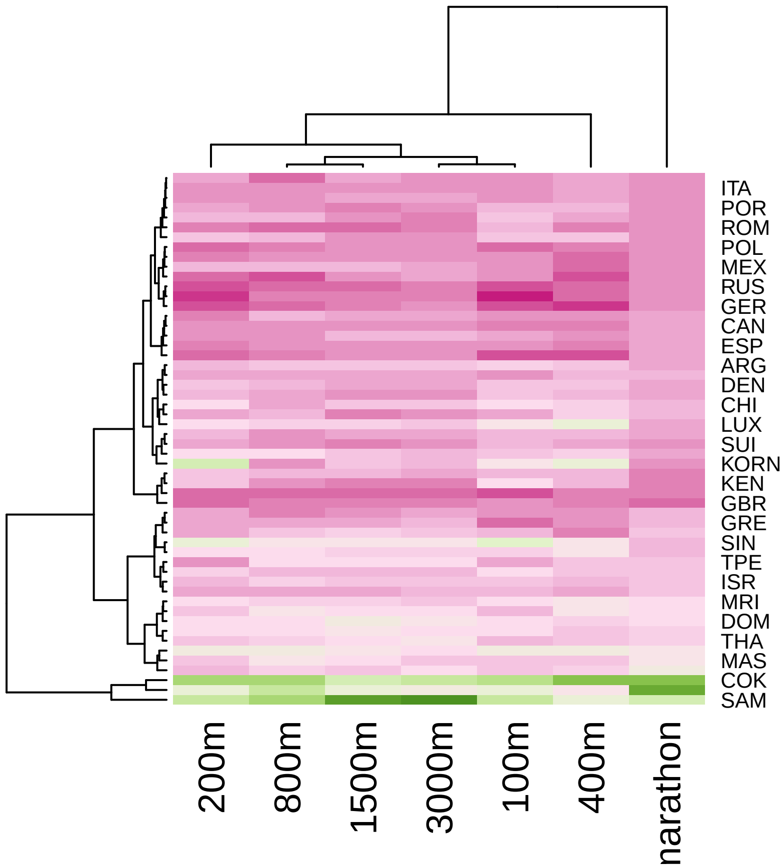
	100m	200m	400m	800m	1500m	3000m	marathon
100m	0.1553157	0.3445608	0.8912960	0.0277036	0.0838912	0.2338828	4.334178
200m	0.3445608	0.8630883	2.1928363	0.0661659	0.2027633	0.5543502	10.384988
400m	0.8912960	2.1928363	6.7454576	0.1818079	0.5091768	1.4268158	28.903731
800m	0.0277036	0.0661659	0.1818079	0.0075469	0.0214146	0.0613793	1.219655

	100m	200m	400m	800m	1500m	3000m	marathon
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

b)



c)



Problem 3

countries
PNG
COK
SAM
BER
GBR

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
```