

# 732A97 Multivariate Statistics Lab 1

Raymond Sseguya

2019-11-23

## Question 1: Describing individual variables

Consider the data set in the T1-9.dat file, National track records for women. For 55 different countries we have the national records for 7 variables (100, 200, 400, 800, 1500, 3000m and marathon ). Use R to do the following analyses.

input

```
trackrcs <- read.table("T1-9.dat",
  col.names = c("countries", "x100m", "x200m", "x400m", "x800m", "x1500m", "x3000m", "marathon"))

trackrcs2 <- (trackrcs)[,-1]
rownames(trackrcs2) <- trackrcs[,1]
```

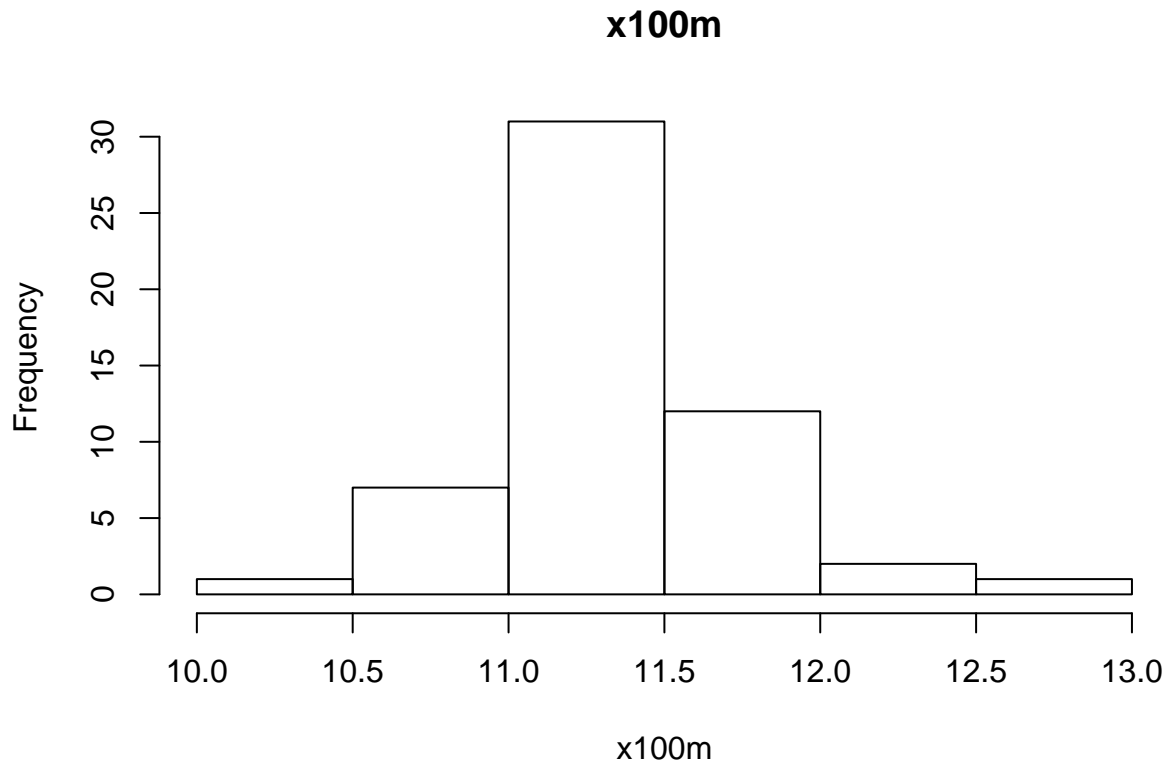
a) Describe the 7 variables with mean values, standard deviations e.t.c

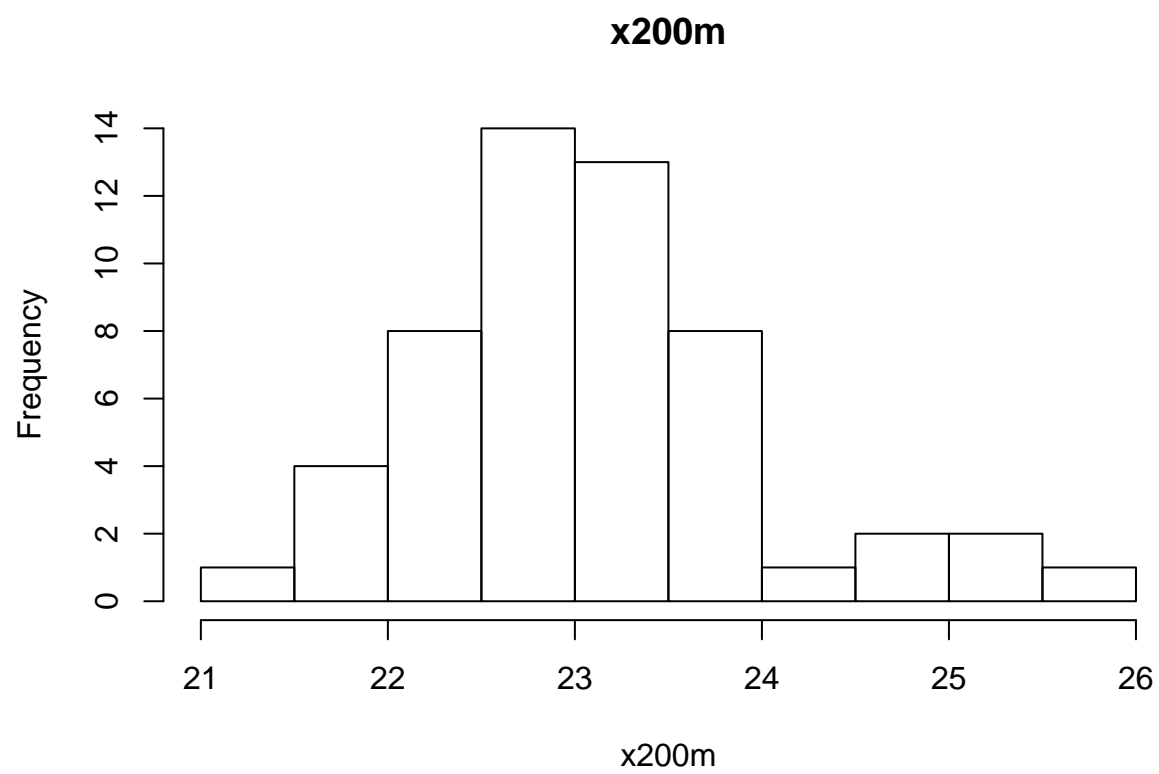
```
apply(trackrcs[, -1], 2, FUN = function(cl){
  c("mean"=mean(cl),
    "median"=median(cl),
    "sd"=sd(cl),
    "95% quantile"=quantile(cl, probs = 0.975, names = FALSE),
    "5% quantile"=quantile(cl, probs = 0.025, names = FALSE),
    "max"=max(cl),
    "min"=min(cl))
}) %>% t()
```

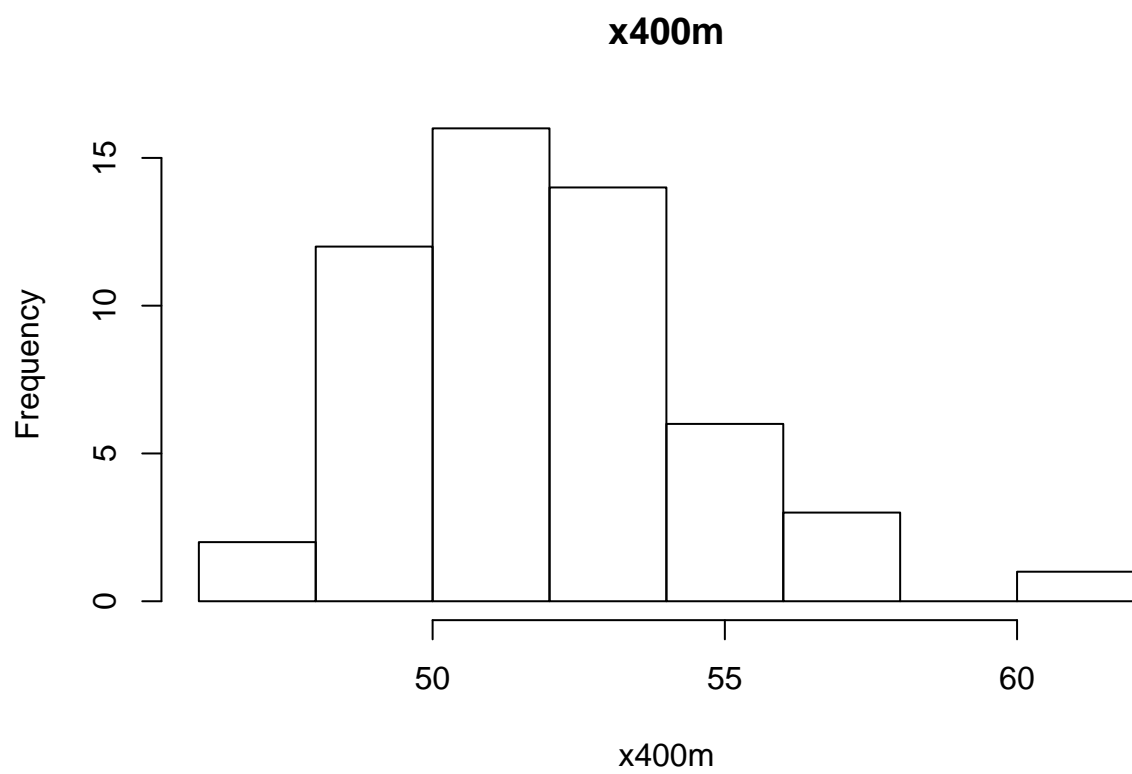
##	mean	median	sd	95% quantile	5% quantile	max
## x100m	11.357778	11.325	0.39410116	12.29875	10.74300	12.52
## x200m	23.118519	22.980	0.92902547	25.33625	21.76200	25.91
## x400m	51.989074	51.645	2.59720188	56.29075	48.07450	61.65
## x800m	2.022407	2.005	0.08687304	2.26700	1.91325	2.29
## x1500m	4.189444	4.100	0.27236502	4.75500	3.87975	5.42
## x3000m	9.080741	8.845	0.81532689	10.81075	8.36000	13.12
## marathon	153.619259	148.430	16.43989508	205.58625	138.76900	221.14
##	min					
## x100m	10.49					
## x200m	21.34					
## x400m	47.60					
## x800m	1.89					
## x1500m	3.84					
## x3000m	8.10					
## marathon	135.25					

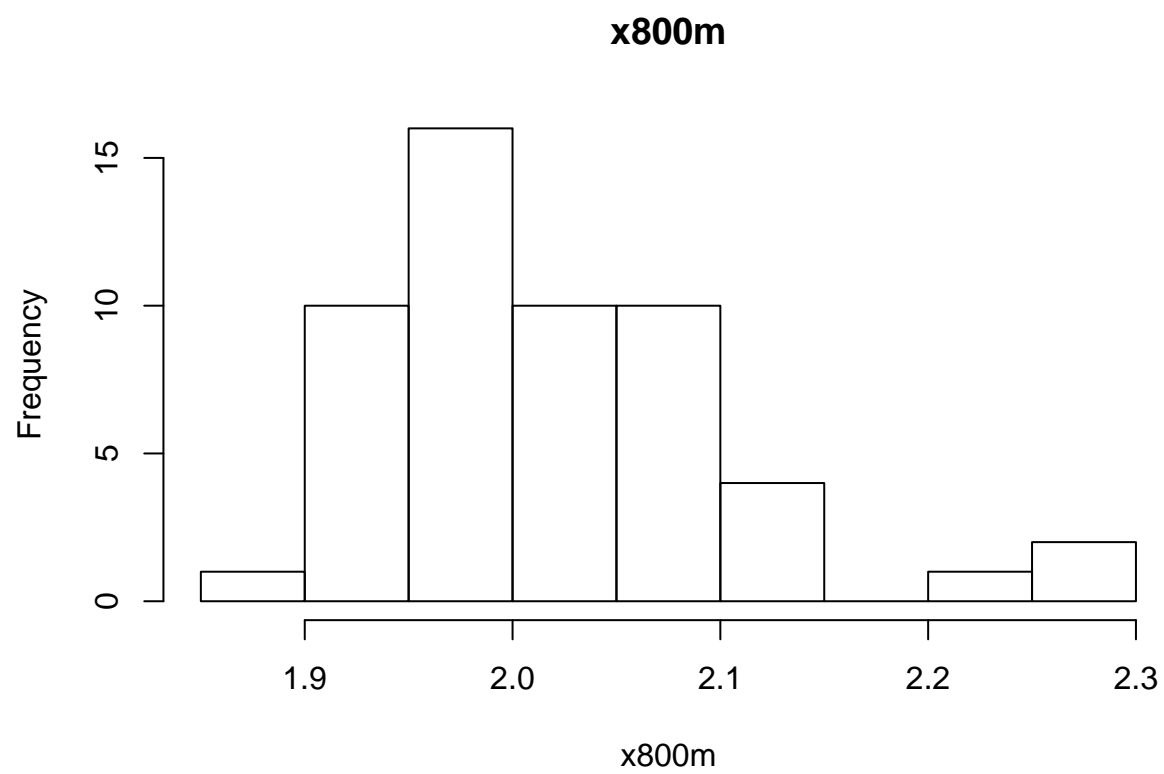
b) Illustrate the variables with different graphs (explore what plotting possibilities R has). Make sure that the graphs look attractive (it is absolutely necessary to look at the labels, font sizes, point types). Are there any apparent extreme values? Do the variables seem normally distributed? Plot the best fitting (match the mean and standard deviation, i.e. method of moments) Gaussian density curve on the data's histogram. For the last part you may be interested in the `hist()` and `density()` functions.

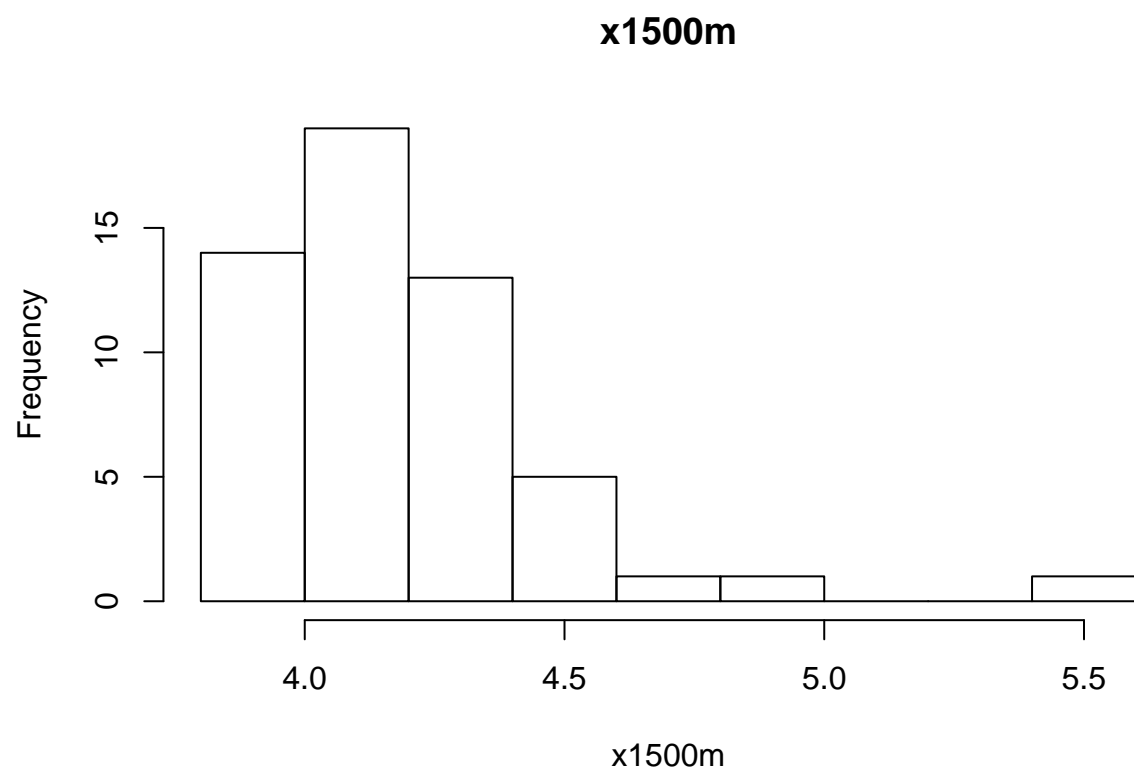
```
j=0; lapply(trackrcs2, FUN = function(i){  
  j <- j+1  
  hist(i, main = colnames(trackrcs2)[j], xlab=colnames(trackrcs2)[j])  
})
```

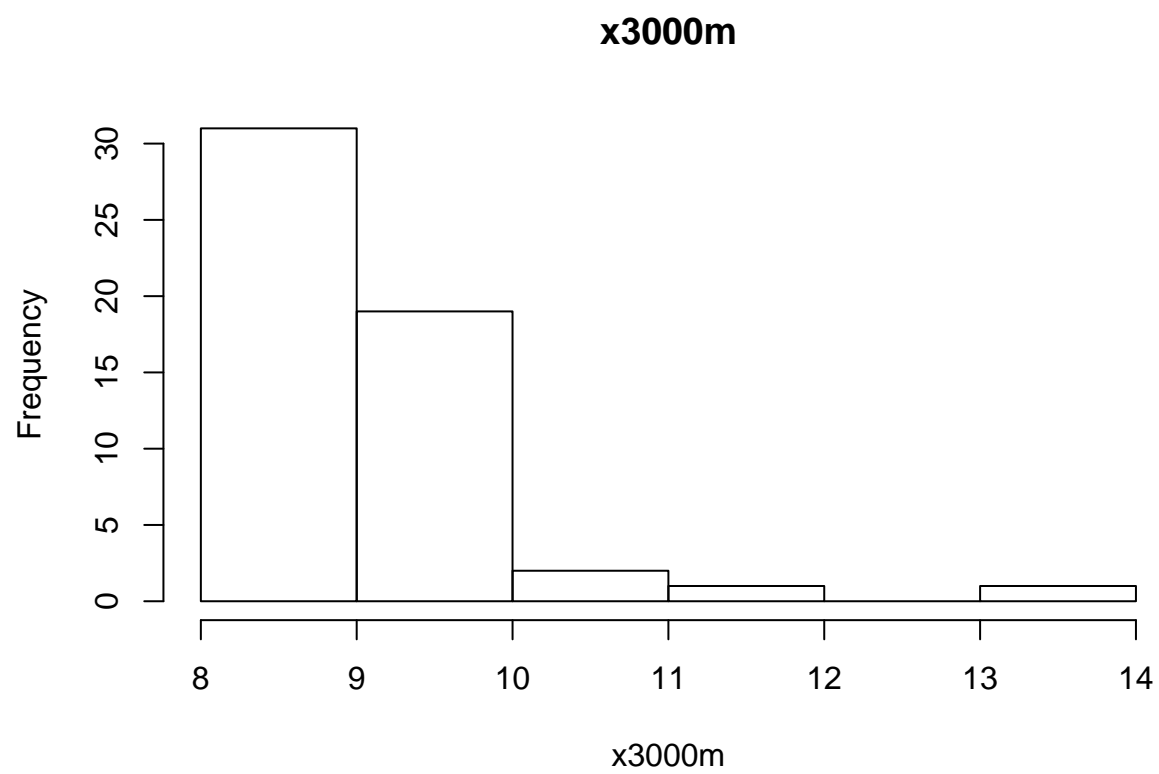




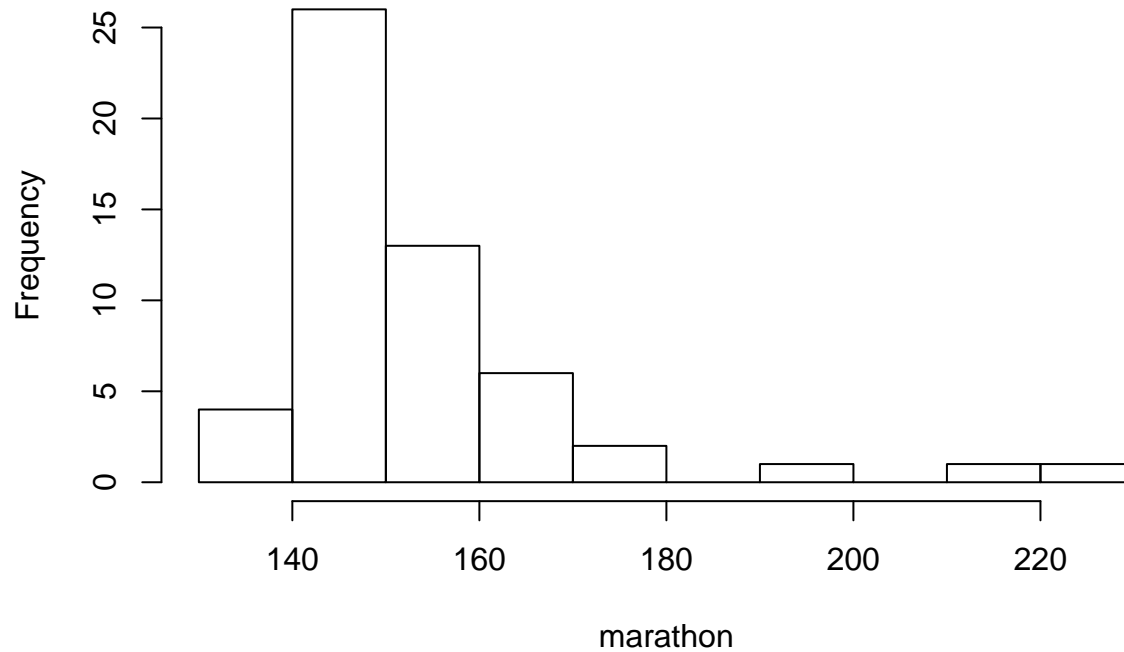








## marathon



```
## $x100m
## $breaks
## [1] 10.0 10.5 11.0 11.5 12.0 12.5 13.0
##
## $counts
## [1] 1 7 31 12 2 1
##
## $density
## [1] 0.03703704 0.25925926 1.14814815 0.44444444 0.07407407 0.03703704
##
## $mids
## [1] 10.25 10.75 11.25 11.75 12.25 12.75
##
## $xname
## [1] "i"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $x200m
## $breaks
## [1] 21.0 21.5 22.0 22.5 23.0 23.5 24.0 24.5 25.0 25.5 26.0
##
```



```

## $counts
## [1] 1 4 8 14 13 8 1 2 2 1
##
## $density
## [1] 0.03703704 0.14814815 0.29629630 0.51851852 0.48148148 0.29629630
## [7] 0.03703704 0.07407407 0.07407407 0.03703704
##
## $mids
## [1] 21.25 21.75 22.25 22.75 23.25 23.75 24.25 24.75 25.25 25.75
##
## $xname
## [1] "i"
##
## $equidist
## [1] TRUE
##
## attr("class")
## [1] "histogram"
##
## $x400m
## $breaks
## [1] 46 48 50 52 54 56 58 60 62
##
## $counts
## [1] 2 12 16 14 6 3 0 1
##
## $density
## [1] 0.018518519 0.111111111 0.148148148 0.129629630 0.055555556 0.027777778
## [7] 0.000000000 0.009259259
##
## $mids
## [1] 47 49 51 53 55 57 59 61
##
## $xname
## [1] "i"
##
## $equidist
## [1] TRUE
##
## attr("class")
## [1] "histogram"
##
## $x800m
## $breaks
## [1] 1.85 1.90 1.95 2.00 2.05 2.10 2.15 2.20 2.25 2.30
##
## $counts
## [1] 1 10 16 10 10 4 0 1 2
##
## $density
## [1] 0.3703704 3.7037037 5.9259259 3.7037037 3.7037037 1.4814815 0.0000000
## [8] 0.3703704 0.7407407
##
## $mids

```

```

## [1] 1.875 1.925 1.975 2.025 2.075 2.125 2.175 2.225 2.275
##
## $xname
## [1] "i"
##
## $equidist
## [1] TRUE
##
## attr("class")
## [1] "histogram"
##
## $x1500m
## $breaks
## [1] 3.8 4.0 4.2 4.4 4.6 4.8 5.0 5.2 5.4 5.6
##
## $counts
## [1] 14 19 13 5 1 1 0 0 1
##
## $density
## [1] 1.29629630 1.75925926 1.20370370 0.46296296 0.09259259 0.09259259
## [7] 0.00000000 0.00000000 0.09259259
##
## $mids
## [1] 3.9 4.1 4.3 4.5 4.7 4.9 5.1 5.3 5.5
##
## $xname
## [1] "i"
##
## $equidist
## [1] TRUE
##
## attr("class")
## [1] "histogram"
##
## $x3000m
## $breaks
## [1] 8 9 10 11 12 13 14
##
## $counts
## [1] 31 19 2 1 0 1
##
## $density
## [1] 0.57407407 0.35185185 0.03703704 0.01851852 0.00000000 0.01851852
##
## $mids
## [1] 8.5 9.5 10.5 11.5 12.5 13.5
##
## $xname
## [1] "i"
##
## $equidist
## [1] TRUE
##
## attr("class")

```

```

## [1] "histogram"
##
## $marathon
## $breaks
## [1] 130 140 150 160 170 180 190 200 210 220 230
##
## $counts
## [1] 4 26 13 6 2 0 1 0 1 1
##
## $density
## [1] 0.007407407 0.048148148 0.024074074 0.011111111 0.003703704
## [6] 0.000000000 0.001851852 0.000000000 0.001851852 0.001851852
##
## $mids
## [1] 135 145 155 165 175 185 195 205 215 225
##
## $xname
## [1] "i"
##
## $equidist
## [1] TRUE
##
## attr("class")
## [1] "histogram"

```

## Question 2: Relationships between the variables

a) Compute the covariance and correlation matrices for the 7 variables. Is there any apparent structure in them? Save these matrices for future use.

```
C <- cov((trackrcs)[,-1])
corr_m <- cor((trackrcs)[,-1])
```

C

```
##           x100m      x200m      x400m      x800m      x1500m
## x100m      0.15531572  0.3445608  0.8912960  0.027703564  0.08389119
## x200m      0.34456080  0.8630883  2.1928363  0.066165898  0.20276331
## x400m      0.89129602  2.1928363  6.7454576  0.181807932  0.50917683
## x800m      0.02770356  0.0661659  0.1818079  0.007546925  0.02141457
## x1500m     0.08389119  0.2027633  0.5091768  0.021414570  0.07418270
## x3000m     0.23388281  0.5543502  1.4268158  0.061379315  0.21615514
## marathon  4.33417757 10.3849876 28.9037314 1.219654647 3.53983732
##           x3000m      marathon
## x100m      0.23388281  4.334178
## x200m      0.55435017 10.384988
## x400m      1.42681579 28.903731
## x800m      0.06137932  1.219655
## x1500m     0.21615514  3.539837
## x3000m     0.66475793 10.706091
## marathon 10.70609113 270.270150
```

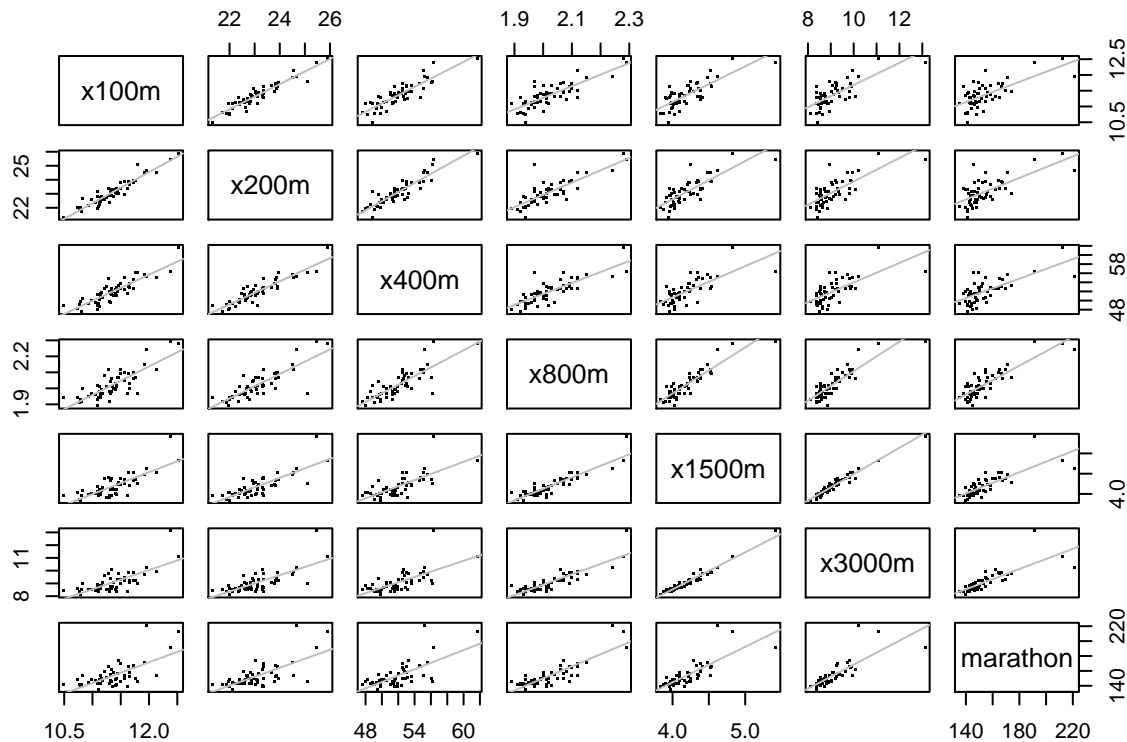
corr\_m

```
##           x100m      x200m      x400m      x800m      x1500m      x3000m
## x100m      1.0000000  0.9410886  0.8707802  0.8091758  0.7815510  0.7278784
## x200m      0.9410886  1.0000000  0.9088096  0.8198258  0.8013282  0.7318546
## x400m      0.8707802  0.9088096  1.0000000  0.8057904  0.7197996  0.6737991
## x800m      0.8091758  0.8198258  0.8057904  1.0000000  0.9050509  0.8665732
## x1500m     0.7815510  0.8013282  0.7197996  0.9050509  1.0000000  0.9733801
## x3000m     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
## x100m      0.6689597
## x200m      0.6799537
## x400m      0.6769384
## x800m      0.8539900
## x1500m     0.7905565
## x3000m     0.7987302
## marathon  1.0000000
```

Both matrices are symmetric. The correlation matrix has ones on the main diagonal.

b) Generate and study the scatterplots between each pair of variables. Any extreme values?

```
pairs(trackrcs[,-1], pch = ".", cex = 1.5, panel = function(x, y, ...){
  points(x, y, ...)
  abline(lm(y ~ x), col = "grey") })
```



The scatterplot matrix tells us that “marathon” is quite an outlier.

c) Explore what other plotting possibilities R offers for multivariate data. Present other (at least two) graphs that you find interesting with respect to this data set.

chernoff face

```
library(aplpack)
faces(trackrcs2)
```

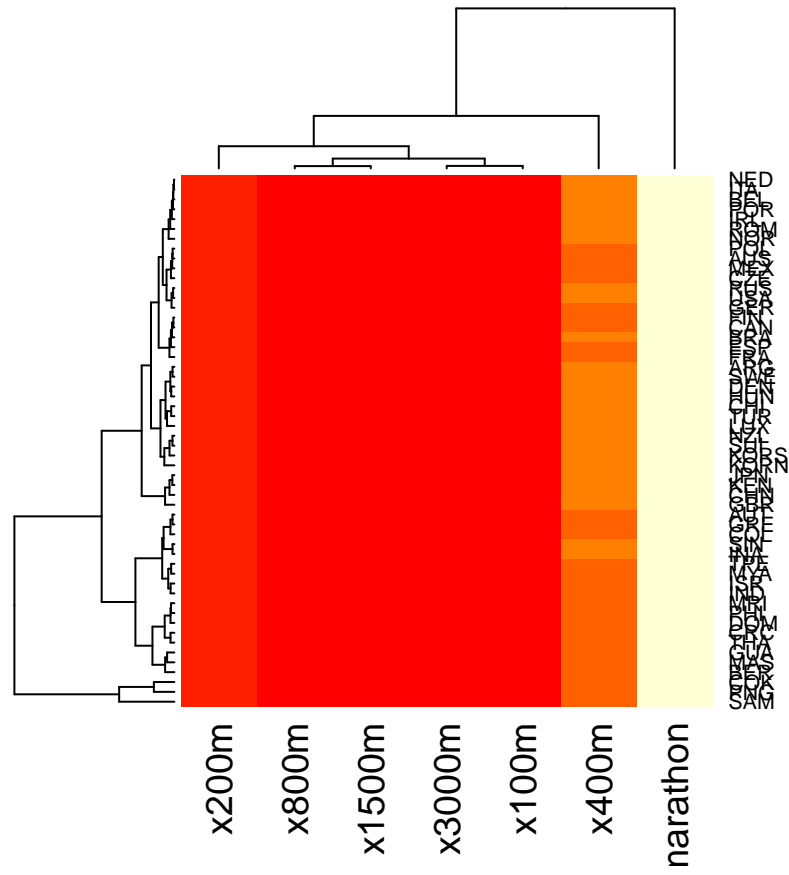
ARG	AUS	AUT	BEL	BER	BRA	CAN	CHI
CHN	COL	COK	CRC	CZE	DEN	DOM	FIN
FRA	GER	GBR	GRE	GUA	HUN	INA	IND
IRL	ISR	ITA	JPN	KEN	KORSKORN	LUX	
MAS	MRI	MEX	MYA	NED	NZL	NOR	PNG
PHI	POL	POR	ROM	RUS	SAM	SIN	ESP
SWE	SUI	TPE	THA	TUR	USA		

```
## effect of variables:
## modified item      Var
## "height of face   " "x100m"
## "width of face    " "x200m"
## "structure of face" "x400m"
## "height of mouth  " "x800m"
## "width of mouth   " "x1500m"
## "smiling          " "x3000m"
## "height of eyes   " "marathon"
## "width of eyes    " "x100m"
## "height of hair    " "x200m"
## "width of hair     " "x400m"
## "style of hair     " "x800m"
## "height of nose    " "x1500m"
## "width of nose     " "x3000m"
## "width of ear      " "marathon"
## "height of ear     " "x100m"
```

A stars plot is not very useful. We can tell from literature that it is also outdated.

heatmaps

```
heatmap(x=as.matrix(trackrcs2))
```



### Question 3: Examining for extreme values

a) Look at the plots (esp. scatterplots) generated in the previous question. Which 3-4 countries appear most extreme? Why do you consider them extreme?

The 3 countries that seem most extreme are Brazil, Russia and the United States.

b) Compute the squared Euclidean distance (i.e.  $r = 2$ ) of the observation from the sample mean for all 55 countries using R's matrix operations. First center the raw data by the means to get  $(x - \bar{x})$  for each country. Then do a calculation with matrices that will result in a matrix that has on its diagonal the requested squared distance for each country. Copy this diagonal to a vector and report on the five most extreme countries. In this questions you MAY NOT use any loops.

```
x_bar = apply(trackrcs2,1,mean)
d0 = as.matrix(trackrcs2-x_bar); dim(d0)
```

```
## [1] 54 7
```

```
deviation = sqrt( d0%*%t(d0) ); dim(deviation)
```

```
## [1] 54 54
```

```
diagonal_vector <- diag(deviation)
deviation_countries <-
  cbind.data.frame(countries = as.vector(trackrcs[,1]),diagonal_vector)
deviation_countries_ordered <-
  deviation_countries[order(-deviation_countries$diagonal_vector), ]
deviation_countries_ordered[1:5,]
```

```
##      countries diagonal_vector
## PNG      PNG      193.0557
## COK      COK      185.0669
## SAM      SAM      165.7015
## BER      BER      151.3534
## GUA      GUA      148.7706
```

The five most extreme countries are PNG, COK, SAM, BER, GUA.



c)

```
V <- diag(apply(trackrcs2,2,var))
d_sq_v <- d0%%solve(V)%%t(d0)
diagonal_vector2 <- diag(d_sq_v)
deviation_countries2 <-
  cbind.data.frame(countries = as.vector(trackrcs[,1]),diagonal_vector2)
deviation_countries_ordered2 <-
  deviation_countries[order(-deviation_countries2$diagonal_vector), ]

deviation_countries_ordered2[1:5,]
```

```
##      countries diagonal_vector
## COK      COK      185.0669
## PNG      PNG      193.0557
## SAM      SAM      165.7015
## GUA      GUA      148.7706
## BER      BER      151.3534
```

Except Great Britain, still the top five most extreme countries are the same and they are COK, PNG, SAM, GUA, BER

d) Compute the Mahalanobis distance, which countries are most extreme now?

```
d_sq_m <- d0%%solve(C)%%t(d0)
diagonal_vector3 <- diag(d_sq_m)
deviation_countries3 <-
  cbind.data.frame(countries = as.vector(trackrcs[,1]),diagonal_vector2)
deviation_countries_ordered3 <-
  deviation_countries[order(-deviation_countries2$diagonal_vector), ]

deviation_countries_ordered3[1:5,]
```

```
##      countries diagonal_vector
## COK      COK      185.0669
## PNG      PNG      193.0557
## SAM      SAM      165.7015
## GUA      GUA      148.7706
## BER      BER      151.3534
```

Still the top five most extreme countries are the same and they are COK, PNG, SAM, GUA, BER

e) Compare the results in b){d). Some of the countries are in the upper end with all the measures and perhaps they can be classified as extreme. Discuss this. But also notice the different measures give rather different results (how does Sweden behave?). Summarize this graphically. Produce Czekanowski's diagram using e.g. the RMaCzek package. In case of problems please describe them.

Sweden

Czekanowski's diagram

```
library(RMaCzek)
x<-czek_matrix(trackrcs2, n_classes = 7)
plot(x)
plot.czek_matrix(x)
```

Czekanowski's diagram

