

732A97 Multivariate Statistics Lab 1

Raymond Sseguya

2019-11-19

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")
colnames(trackrcs) <- 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

mean values

```
colMeans((trackrcs)[,-1])
```

```
##      x100m      x200m      x400m      x800m      x1500m      x3000m
## 11.357778 23.118519 51.989074  2.022407  4.189444  9.080741
##  marathon
## 153.619259
```

median

```
apply((trackrcs)[,-1], 2, median)
```

```
##      x100m      x200m      x400m      x800m      x1500m      x3000m  marathon
##   11.325    22.980    51.645     2.005     4.100     8.845   148.430
```

standard deviation

```
apply((trackrcs)[,-1], 2, sd)
```

```
##      x100m      x200m      x400m      x800m      x1500m      x3000m
## 0.39410116 0.92902547 2.59720188 0.08687304 0.27236502 0.81532689
##  marathon
## 16.43989508
```

maximum

```
apply((trackrcs)[,-1], 2, max)
```

```
##      x100m      x200m      x400m      x800m      x1500m      x3000m marathon
##      12.52      25.91      61.65       2.29       5.42      13.12      221.14
```

minimum

```
apply((trackrcs)[,-1], 2, min)
```

```
##      x100m      x200m      x400m      x800m      x1500m      x3000m marathon
##      10.49      21.34      47.60       1.89       3.84       8.10      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.

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.

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

```
cov_m
```

```
##           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
```

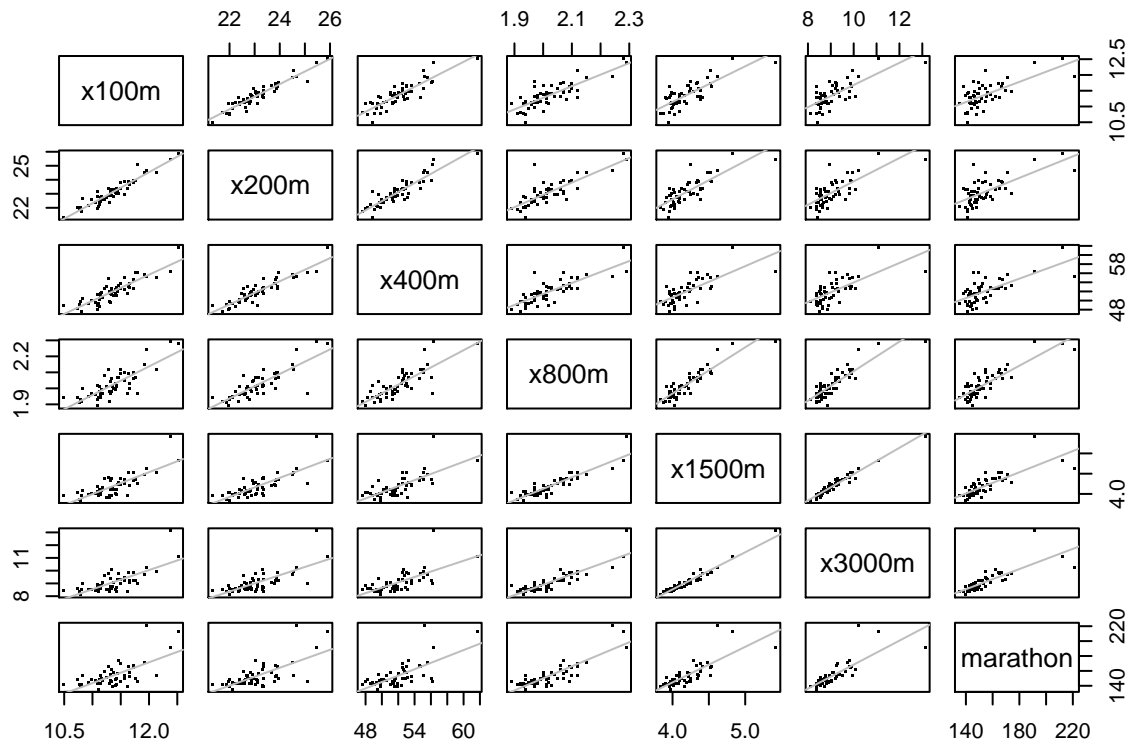
```
corr_m
```

```
##           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") })
```

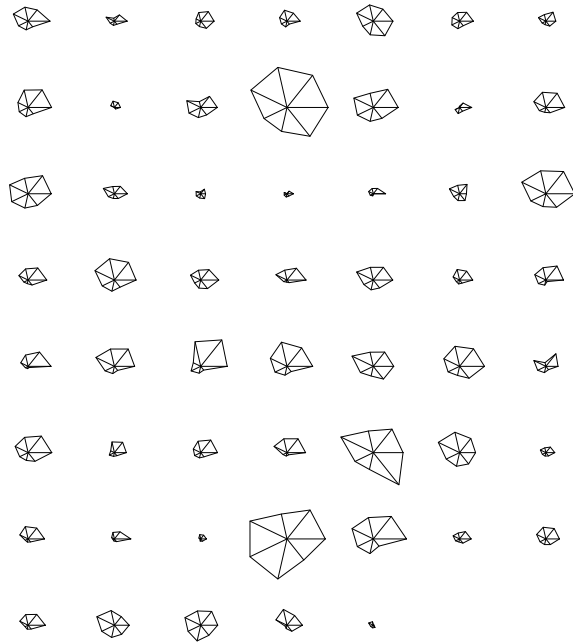


The scatterplot matrix tells us that “marathon” is quite an outlier.

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.

stars plot

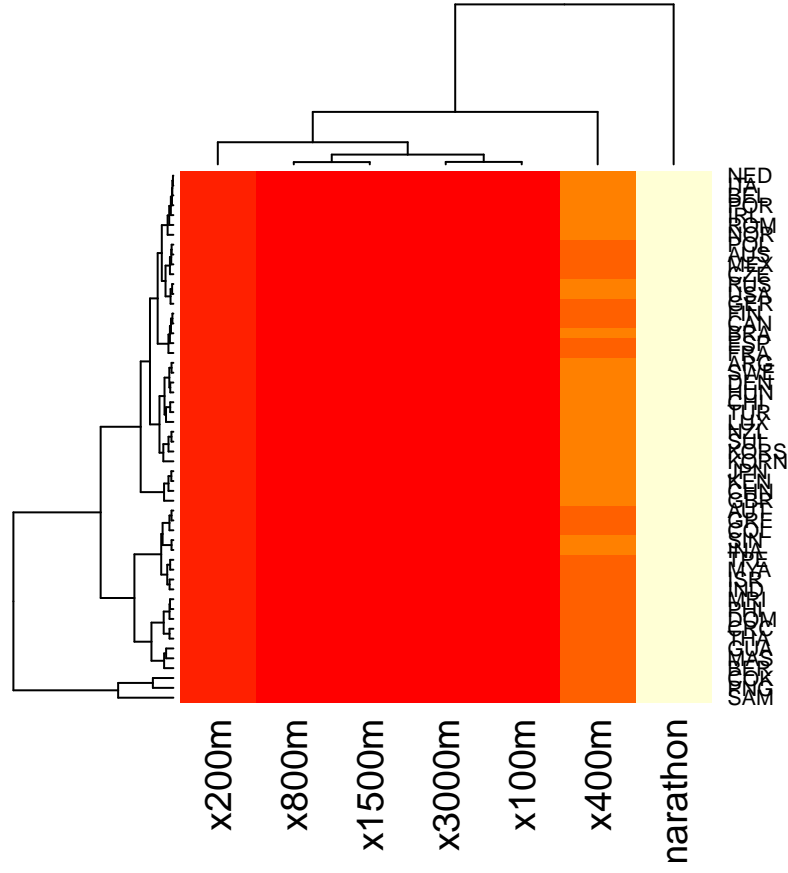
```
stars(trackrcs[,-1], cex = 0.55)
```



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?

```
C <- cov(trackrcs2)
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.

```
library(RMaCzek)
x<-czek_matrix(trackrcs2)
plot(x)
plot.czek_matrix(x)
```

Czekanowski's diagram

