732A97 Multivariate Statistical Methods

Assignment 1 Examining multivariate data

Alexander Karlsson(aleka769),Raymond Sseguya(y),Mariano Mariani(marma330), Ruben Munoz(rubmu773)

2019-11-23

Contents

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.

1.1 A)

Instructions

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

Answer

Table 1: Analysis by numbers

	100m	200m	400m	800m	1500m	3000m	marathon
mean	11.3577778	23.1185185	51.989074	2.022407	4.189444	9.0807407	153.6193
median	11.3250000	22.9800000	51.645000	2.005000	4.100000	8.8450000	148.4300
sd	0.3941012	0.9290255	2.597202	0.086873	0.272365	0.8153269	16.4399
min	10.4900000	21.3400000	47.600000	1.890000	3.840000	8.1000000	135.2500
max	12.5200000	25.9100000	61.650000	2.290000	5.420000	13.1200000	221.1400
top.5.	10.7830000	21.9350000	48.497000	1.920000	3.913000	8.3665000	139.4030
top.95.	12.0195000	24.8270000	56.126000	2.181500	4.568000	10.1190000	180.2700

The 100m ,200m and 400m races are in different scale than them 800m, 1500m, 3000m and marathons. As expected, we see the variance increasing, as the races get longer. The longer the race the more factors affecting the final result: 100m races are won by less than a second differences, whereas in the marathons it's won in the scale of minutes.

1.2 B)

Instructions

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.

Answer

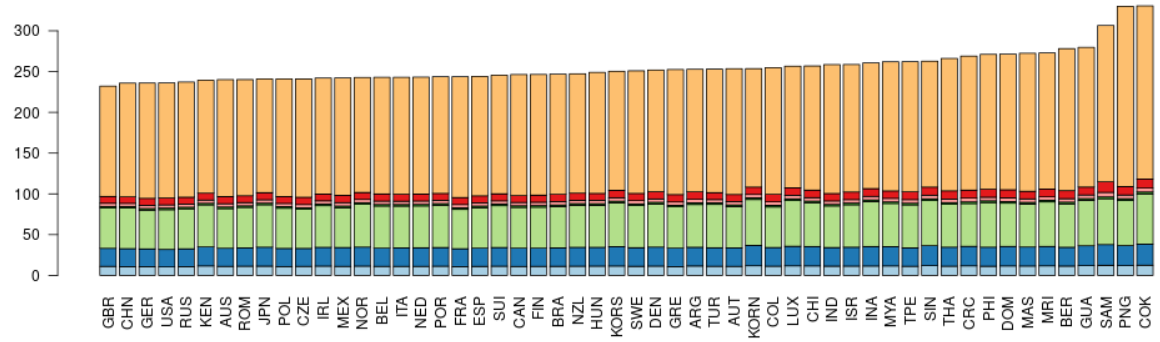

Figure 1: *Stacked barplot, it gives a general idea of what countries are overall faster. This plot goes from faster to slowest overall times.*

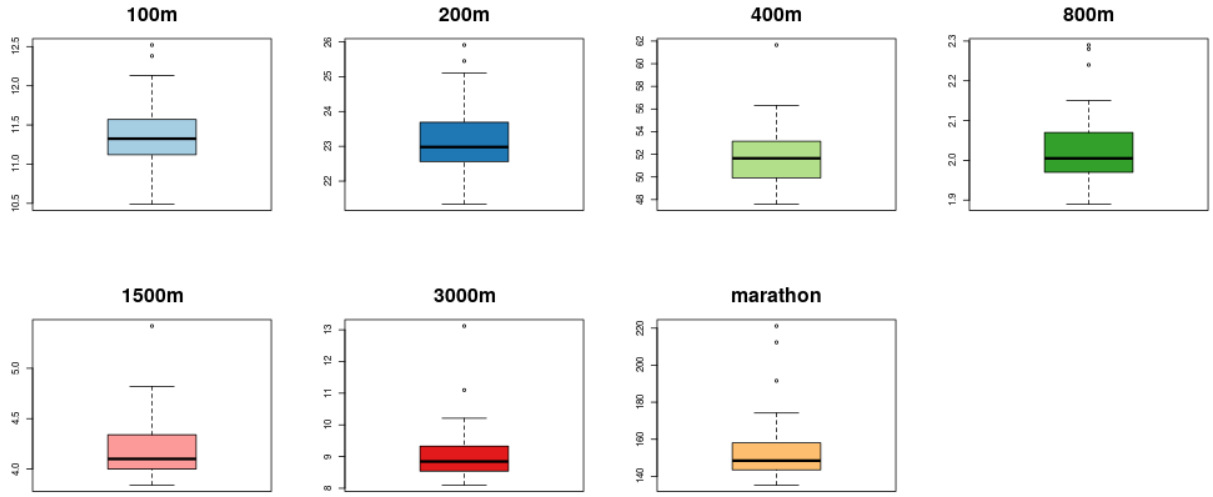

Figure 2: *A boxplot, it gives a general view of where the datapoints are more consentrated and easier to spot the outliers. This set of boxplots show all the countries in their respective cathegories.*

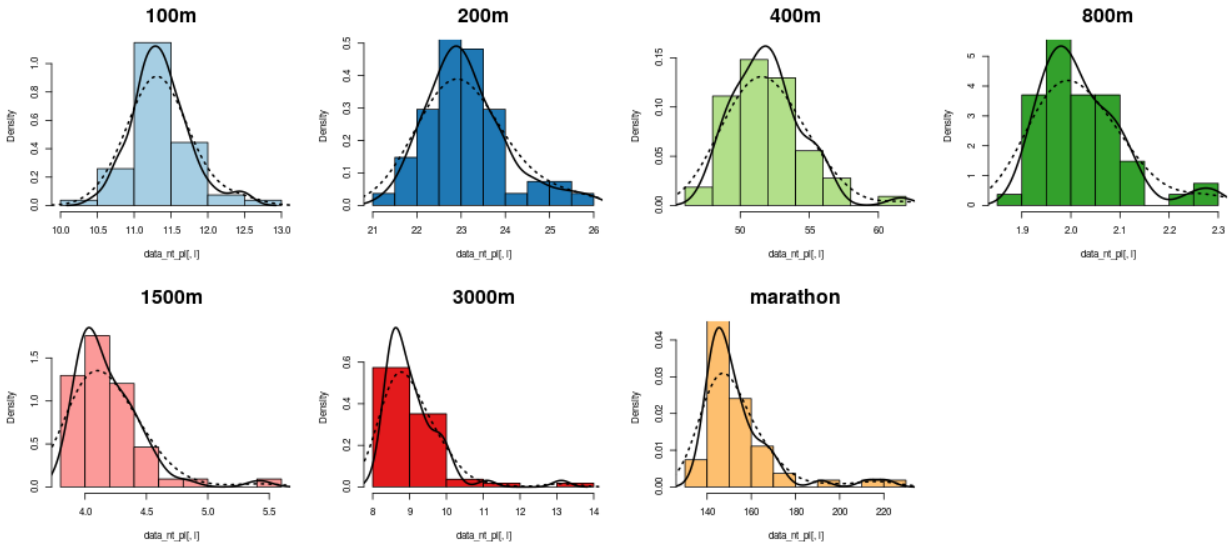

Figure 3: *Histogram with a density curve. Looking for variables that are morenormaly distributed.*

As the race gets longer, we see more variance and more extreme values. As it can be seen in Figure 3, the shorter the race the more it looks like a normal. Starting the 1500m race, it starts looking more like a log normal.

2 Relationships between the variables

2.1 A)

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

Answer

Table 2: Covariance matrix/table

	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

Table 3: Correlation matrix/table

	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

We can see that each type of race is highly correlated with similar races, if a country is good for a 100m race, its highly likely that will be too for 200m.

This could be due to certain countries training more for marathons or 100m, being better and that and worse in different races.

2.2 B)

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

Answer

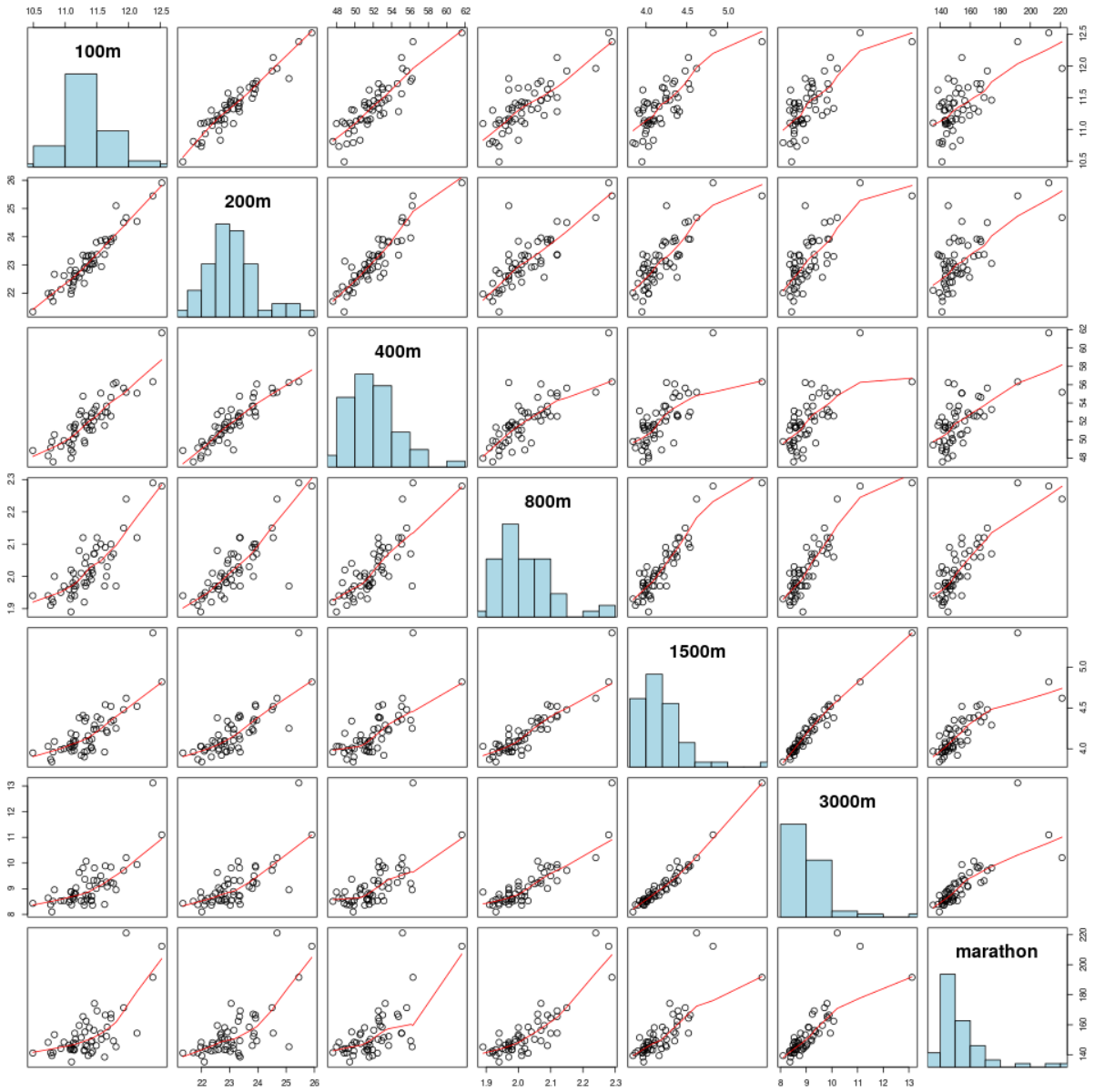

Figure 4: *Scatterplot of everybody against everybody.*

As pointed before, we see some extreme values in the marathon times. The scatterplot is quite useful to see how correlation decreases when the races length differ. Furthermore we see positive correlation in all cases, as expected, longer distances take longer to complete.

2.3 C)

Instructions 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.

Answer

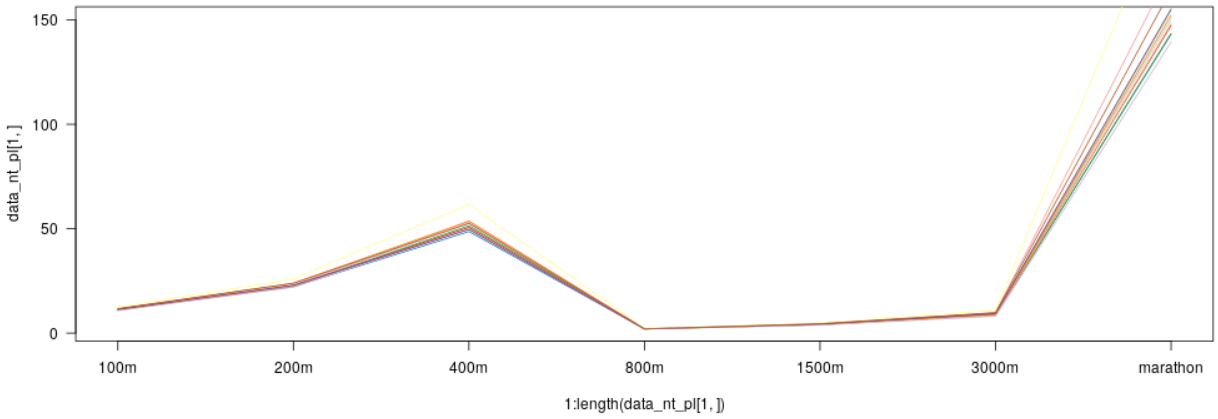

Figure 5: *Parallel coordinates plot*

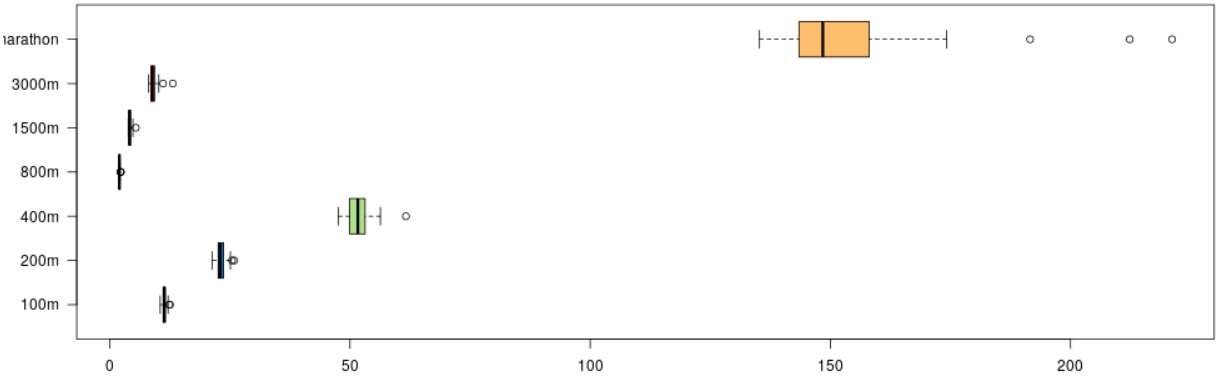

Figure 6: *Multivariate boxplot*

In figure 5: parallel coordinates plot, we can see each line as a different country. The x axis is each different race, and the y axis is the time it took. This plot is quite useful in the sense, that theoretically it would always increase, and we see it decreasing. This reminds us that we are using different scales for the races... Even though it might appear that the runners complete the 800m faster than the 400m, this is due that the 800m results are in minutes, as opposed to the 400m that are measured in seconds.

Figure 6: Multivariate boxplot is quite useful to detect outliers of all different races lengths in one single plot. We can see how the results for 100m,200m,400m,800m are tighter in the cluster, whereas the marathon presents several outliers.

3 Examining for extreme values

3.1 A)

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

One approach to measuring "extremism" is to look at the distance (needs to be defined!) between an observation and the sample mean vector, i.e. we look how far one is from the average. Such a distance can be called an multivariate residual for the given observation.

Answer

3.2 B)

Instructions The most common residual is the Euclidean distance between the observation and sample mean vector, i.e.

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

This 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 observation (here p = 7).

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 − 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.

Answer

3.3 C)

Instructions The different variables have different scales so it is possible that the distances can be dominated by some few variables. To avoid this we can use the squared distance

$$d^2_{\mathbf{V}}(\vec{x}, \bar{x}) = (\vec{x} - \bar{x})^T \mathbf{V}^{-1} (\vec{x} - \bar{x})$$

where V is a diagonal matrix with variances of the appropriate variables on the diagonal. The effect, is that for each variable the squared distance is divided by its variance and we have a scaled independent distance. It is simple to compute this measure by standardizing the raw data with both means (centring) and standard deviations (scaling), and then compute the Euclidean distance for the normalized data. Carry out these computations and conclude which countries are the most extreme ones. How do your conclusions compare with the unnormalized ones?

Answer

3.4 D)

Instructions The most common statistical distance is the Mahalanobis distance

$$d_{\mathrm{M}}^2(\vec{x}, \bar{x}) = (\vec{x} - \bar{x})^T \mathbf{C}^{-1} (\vec{x} - \bar{x})$$

where C is the sample covariance matrix calculated from the data. With this measure we also use the relationships (covariances) between the variables (and not only the marginal variances as d V (· , ·) does). Compute the Mahalanobis distance, which countries are most extreme now?

Answer

3.5 E)

Instructions 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.

Answer

4 Appendix

4.1 Code

4.1.1 Code used for Describing individual variables

```
# required libraries
library(dplyr)
library(kableExtra)
library(RColorBrewer)
my_r_color <- brewer.pal(n=12, name="Paired") # colorblind friendly
# load data
data_nt <- read.table(file="T1-9.dat")
# add feature names
colnames(data_nt) <- c("Country", "100m", "200m", "400m","800m", "1500m", "3000m",
↪  "marathon")

# get the analicys of each feature
data_nt_anal <- data.frame("mean"=apply(data_nt[,-1], 2, mean),
            "median"=apply(data_nt[,-1], 2, median),
            "sd"=apply(data_nt[,-1], 2, sd),
            "min"=apply(data_nt[,-1], 2, min),
            "max"=apply(data_nt[,-1], 2, max),
            "top 5%"=apply(data_nt[,-1], 2, FUN=function(x){quantile(x,0.05)}),
            "top 95%"=apply(data_nt[,-1], 2, FUN=function(x){quantile(x,0.95)}))
# create a fancy table
data_nt_anal %>% t() %>% kable("latex", booktabs=TRUE, caption = "Analysis by numbers")
↪  %>% kable_styling(latex_options =c("striped", "HOLD_position"))

## random nice plots
## stacked barchart
data_nt_pl <- data_nt[,-1]
rownames(data_nt_pl) <- data_nt[,1]
# sort from lowest to higher time
totsum <- apply(data_nt_pl, 1, sum)
totsum <- sort(totsum); index_x <- c()
for(i in 1:length(totsum)){
  index_x <- c(index_x,
               which(rownames(data_nt_pl) == names(totsum)[i]))
}
## png(filename="images/plot01.png", width = 2*500, height = 1*350)
## data_nt_pl[index_x,] %>% t() %>% barplot(col=my_r_color, las=2)
## dev.off()
##
## ## find extreme values
## colnames_cat <- colnames(data_nt_pl)
## png(filename="images/plot02.png", width = 2*500, height = 1*450)
## par(mfrow = c(2,4))
## for(i in 1:length(data_nt_pl)){
##   s=2
##   boxplot(data_nt_pl[,i], col=my_r_color[i], main= colnames_cat[i],
##           cex.main=s) #cex.lab=s, cex.axis=s, cex.sub=s)
## }
```

```
## par(mfrow = c(1,1))
## dev.off()
##
## # find if variables are normaly distributed/ hustogram with density curve
## png(filename="images/plot03.png", width = 2*500, height = 1*450)
## par(mfrow = c(2,4))
## for(i in 1:length(data_nt_pl)){
##   # avoid desity line being cutted
##   dessitylocal=density(data_nt_pl[,i])
##   hist(data_nt_pl[,i], col=my_r_color[i], main= colnames_cat[i],
##           ylim = c(0, max(dessitylocal$y)), cex.main=2, freq=FALSE)
##   lines(dessitylocal, lwd=2)
##   lines(density(data_nt_pl[,i], adjust=2),
##         lwd=2, lty="dotted")
## }
## par(mfrow = c(1,1))
## dev.off()
```

4.1.2 Code used for Relationships between the variables

```
df_covariance <- cov(data_nt_pl) # covariance matrix
# print it fancy
df_covariance %>% t() %>% kable("latex", booktabs=TRUE, caption = "Covariance
↪  matrix/table") %>% kable_styling(latex_options =c("striped", "HOLD_position"))

df_correlation <- cor(data_nt_pl) # correlation matrix
# print it fancy
df_correlation %>% t() %>% kable("latex", booktabs=TRUE, caption = "Correlation
↪  matrix/table") %>% kable_styling(latex_options =c("striped", "HOLD_position"))

## png(filename="images/plot04.png", width = 2*500, height = 2*500)
## # plot(data_nt_pl)
## ## put histograms on the diagonal
## panel.hist <- function(x, ...)
## {
##     usr <- par("usr"); on.exit(par(usr))
##     par(usr = c(usr[1:2], 0, 1.5) )
##     h <- hist(x, plot = FALSE)
##     breaks <- h$breaks; nB <- length(breaks)
##     y <- h$counts; y <- y/max(y)
##     rect(breaks[-nB], 0, breaks[-1], y, col = "light blue", ...)
## }
## pairs(data_nt_pl, panel = panel.smooth,
##       cex = 1.5, pch = 1, bg = "light blue", horOdd=TRUE,
##       diag.panel = panel.hist, cex.labels = 2, font.labels = 2)
## dev.off()


## # Parrallel coordinates plot
## png(filename="images/plot05.png", width = 2*500, height = 1*400)
## plot(x=1:length(data_nt_pl[1,]), y=data_nt_pl[1,], type="l",
##      col=my_r_color, las=1, xaxt = "n")
```

```
## axis(1, at=1:length(data_nt_pl[1,]), labels=colnames(data_nt_pl))
## for(i in 2:length(data_nt_pl[,1])){
##   lines(x=1:length(data_nt_pl[i,]), y=data_nt_pl[i,], col=my_r_color[i])
## }
## dev.off()
## # multivariate scatterplots
## png(filename="images/plot06.png", width = 2*500, height = 1*400)
## boxplot((data_nt_pl), col=my_r_color, horizontal=T, las=1)
## dev.off()
##
## mlab <- "Time it took to run 100m"
## plab <- "Time it took to run 200m"
## layout(matrix(c(2, 0, 1, 3), nrow = 2, byrow = TRUE),
##        widths = c(2, 1),
##        heights = c(1, 2),
##        respect = TRUE)
## png(filename="images/plot07.png", width = 2*500, height = 2*500)
```