732A97 - Multivariate statistical methods

Lab 1 - "Examining multivariate data"

Alexander Karlsson (aleka769)

2019-11-23

Contents

Packages used can be seen below.

```
library(tidyverse)  # for tidy data manipulation
library(MVA)        # library attached to the book
library(lattice)    # for 3D plots
library(KernSmooth) # for scatterplot with contours
library(aplpack)    # for Chernoff faces
library(outliers)   # for outlier test on regression residuals
library(RMaCzek)    # for Czekanowski's diagram
```

The dataset used in this lab contains running times for different distances and different countries. It is aggregated in such a way that each country's data point for each distance is the national record for that specific distance.

```
trackData = read.table(file        = "T1-9.dat",
                       col.names   = c("Country", "100m", "200m", "400m",
                                       "800m", "1500m", "3000m", "42000m"),
                       check.names = FALSE)
```

Question 1 - Describing individual variables

a) Descriptive statistics of variables in data

The following statistics are considered relevant to describe data: *mean*, *standard deviation*, *median*, *5% quantile* and *95% quantile*. There are several other descriptive statistics that can be used (such as *mean/max* etc) but to remove cluttering in the report, the above ones were chosen. Note that variables are represented row-wise, as opposed to the original data.

```
sapply(trackData[2:8], function(d){
  c("mean" = mean(d), "sd" = sd(d), "median" = median(d),
    quantile(d, .05), quantile(d, .95))
}) %>% t()
```

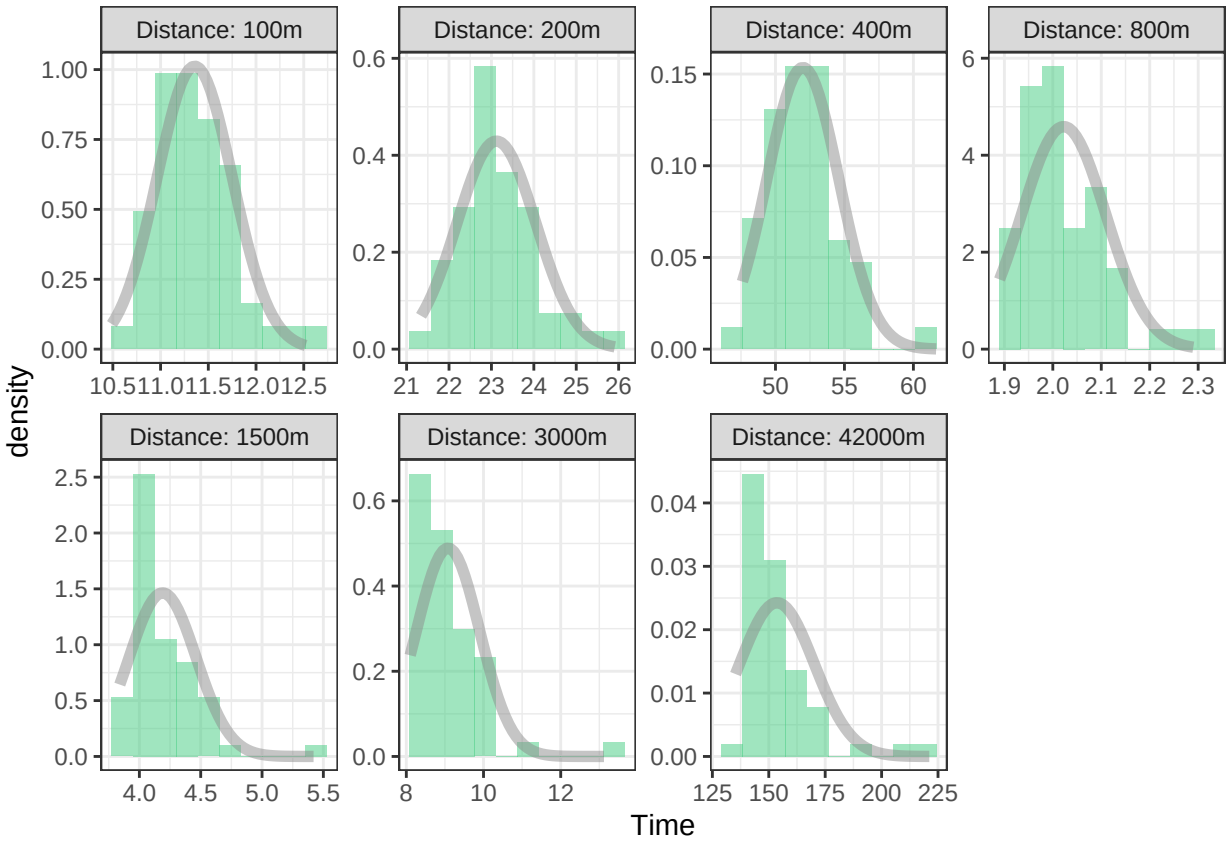
```
##              mean          sd  median       5%      95%
## 100m    11.357778  0.39410116  11.325  10.7830  12.0195
## 200m    23.118519  0.92902547  22.980  21.9350  24.8270
## 400m    51.989074  2.59720188  51.645  48.4970  56.1260
## 800m     2.022407  0.08687304   2.005   1.9200   2.1815
## 1500m    4.189444  0.27236502   4.100   3.9130   4.5680
## 3000m    9.080741  0.81532689   8.845   8.3665  10.1190
## 42000m 153.619259 16.43989508 148.430 139.4030 180.2700
```

Another important thing to consider is that the scales of data are not the same. The variables `100m`, `200m` and `400m` are emasured in seconds, whereas `800m`, `1500m`, `3000m` and `42000m` (marathon) are emasured in minutes. It is not reasonable to believe it takes three times as long to run a marathon than it takes to run 400 meters...

b) Visualise data

Below, the estimated sample histograms (green) for each distance are plotted as a filled density and compared to the theoretical normal distribution (gray) with mean and standard deviation of that specific variable as parameters.

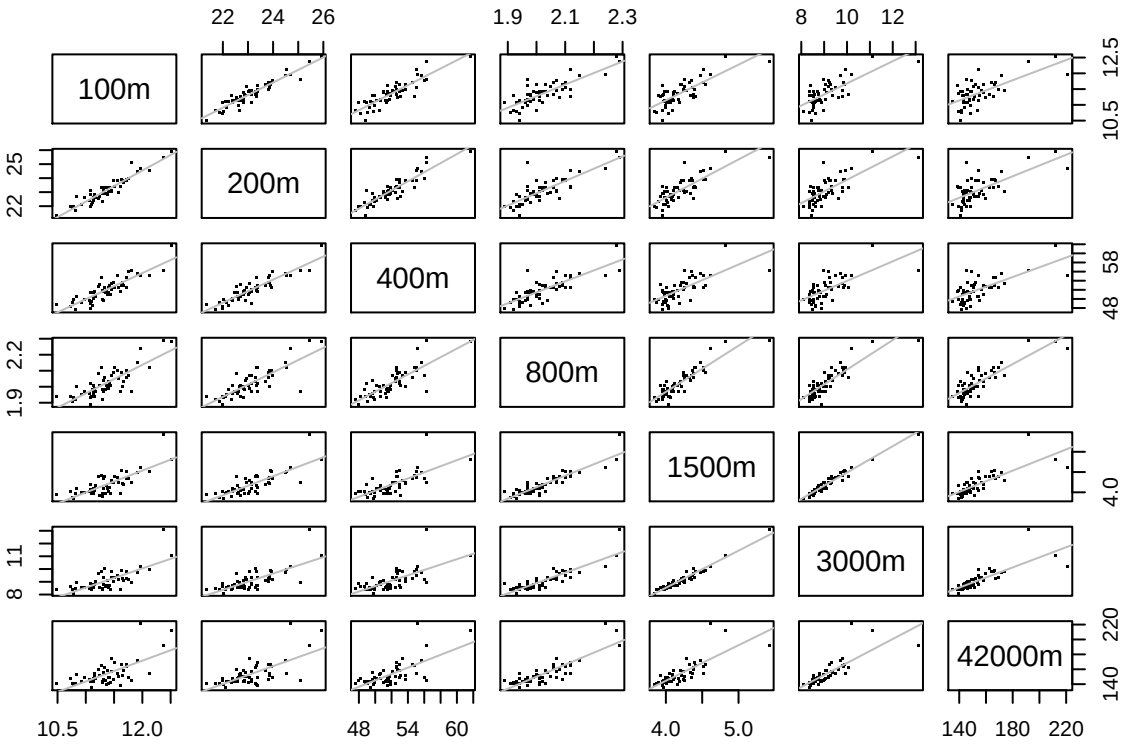
```
d1 = gather(trackData, key = "Distance", value = "Time", -Country) %>%
  mutate(., Distance = factor(Distance, levels = colnames(trackData[-1]))) %>%
  group_by(., Distance)

d2 = d1 %>% arrange(., -desc(Time)) %>% group_by(Distance) %>%
  mutate(., x = seq(min(Time), max(Time), length.out = n()),
         f = dnorm(x, mean(Time), sd(Time)))

ggplot(d1) +
   geom_histogram(aes(x = Time, y=..density..),
                  fill = "seagreen3", alpha = .5, bins = 10) +
  geom_line(mapping = aes(x, f), data = d2,
            color = "gray55", size = 2, alpha = .5) +
  facet_wrap(~Distance, ncol = 4, scales = "free", labeller = label_both) +
  theme_bw()
```



The tendency that longer distances are more skewed, with longer right tails (slower runners) is highlighted above.

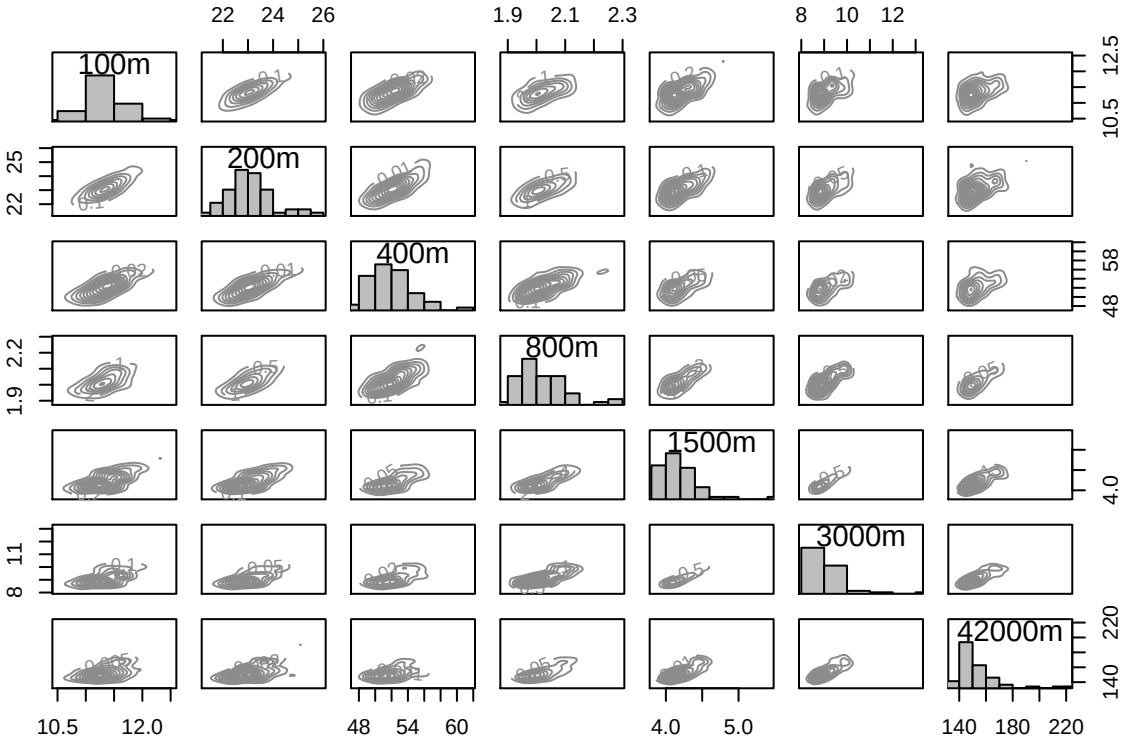
Below, pairwise scatterplots are created between all combinations of variables (distances). For all distances, there exists positive pairwise correlations in running times. Another interesting (but not so strange) phenomena that can be read out from these scatterplots is that similar distances have a higher correlation (tighter points around regression line) and higher slope.

```
pairs(trackData[2:8],
      panel = function (x, y, ...) {
          points(x, y, ...)
          abline(lm(y ~ x), col = "grey")
      }, pch = ".", cex = 1.5)
```



Below, histograms of each distance can be seen on the diagonal, whereas the off-diagonal shows contour plots for pairs of distances. Ellipsoid shapes of these contours indicate a bivariate normal relationship between variables. In case the contours form two or more "islands", the joint distribution is bimodal (or multimodal) in a multivariate plot.

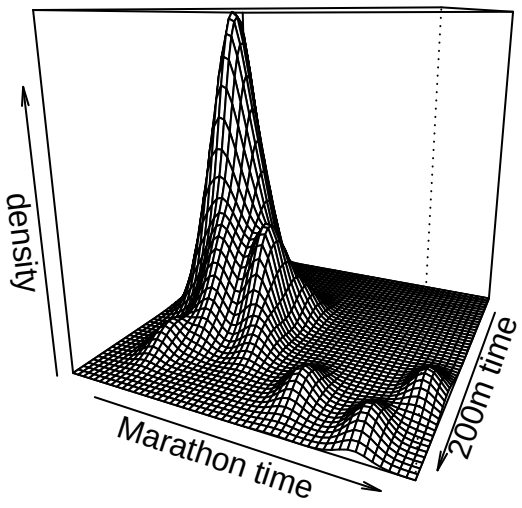
```
panel.hist <- function(x, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "grey", ...)
}
pairs(trackData[,2:8],
      diag.panel = panel.hist,
      panel = function (x,y) {
          data <- data.frame(cbind(x,y))
          par(new = TRUE)
          den <- bkde2D(data, bandwidth = sapply(data, dpik))
          contour(x = den$x1, y = den$x2,
                  z = den$fhat, axes = FALSE, col = "gray55")
      })
```



Although not too strong, such a phenomena can be seen between distances `42000m` and `200m` (highlighted below). The increase in density appears due to outliers, and is highlighted when a density function is used as estimator, simply because there are observations in those areas. Under the assumption that these observations are outliers, it is not reasonable to assume the the "true" joint density looks like the one below.

```
biModal = trackData[c("200m", "42000m")]
biModal = bkde2D(biModal, bandwidth = sapply(biModal, dpik))
persp(x = biModal$x1, y = biModal$x2, z = biModal$fhat,
      xlab = "200m time",
      ylab = "Marathon time",
      zlab = "density", theta = 115, box = TRUE,
      main = "Perspective plot for \"multimodal\" joint density")
```

Perspective plot for "multimodal" joint density



Question 2 - Relationships between the variables

Correlations, covariances and structure in data

```
trackCor = cor(trackData[2:8]) %>% round(3)
trackCor
```

```
##         100m  200m  400m  800m 1500m 3000m 42000m
## 100m   1.000 0.941 0.871 0.809 0.782 0.728  0.669
## 200m   0.941 1.000 0.909 0.820 0.801 0.732  0.680
## 400m   0.871 0.909 1.000 0.806 0.720 0.674  0.677
## 800m   0.809 0.820 0.806 1.000 0.905 0.867  0.854
## 1500m  0.782 0.801 0.720 0.905 1.000 0.973  0.791
## 3000m  0.728 0.732 0.674 0.867 0.973 1.000  0.799
## 42000m 0.669 0.680 0.677 0.854 0.791 0.799  1.000
```

As was concluded in `Question 1`, decreased similarity in distances leads to decreased correlations and vice versa. This is not true for exactly all correlations in the matrix above, but seems to be the general structure.

```
trackCov = cov((trackData[2:8])) %>% round(3)
trackCov
```

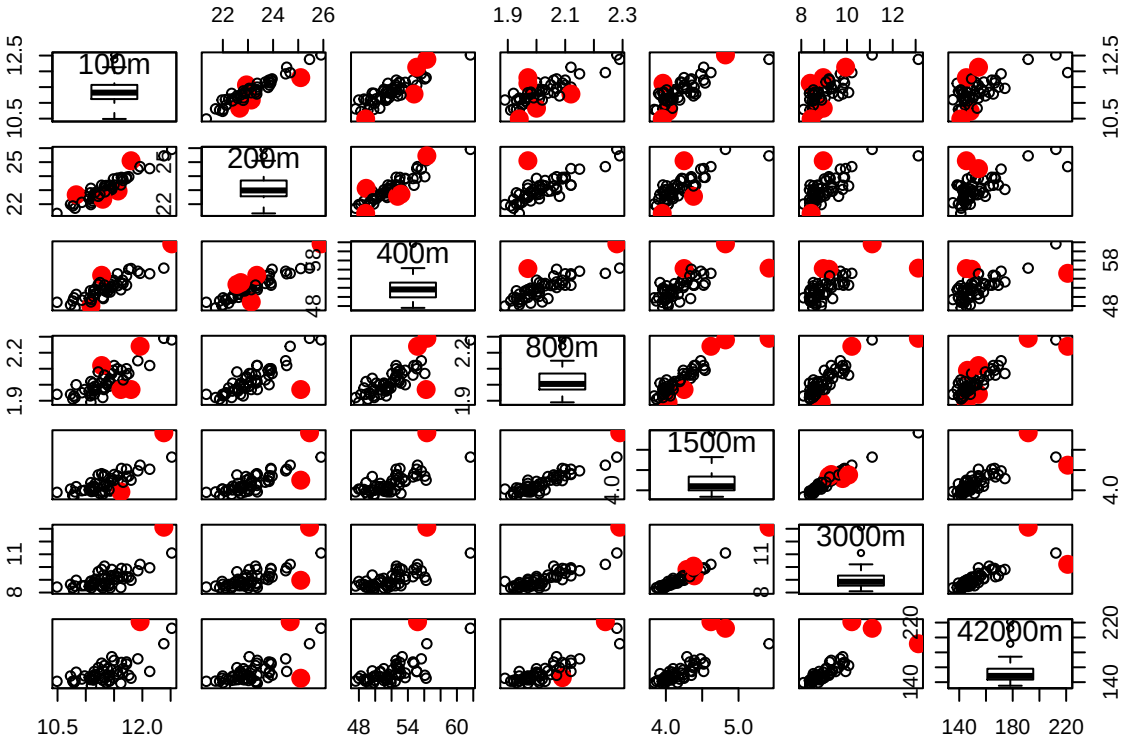
```
##         100m   200m   400m  800m 1500m  3000m  42000m
## 100m   0.155  0.345  0.891 0.028 0.084  0.234   4.334
## 200m   0.345  0.863  2.193 0.066 0.203  0.554  10.385
## 400m   0.891  2.193  6.745 0.182 0.509  1.427  28.904
## 800m   0.028  0.066  0.182 0.008 0.021  0.061   1.220
## 1500m  0.084  0.203  0.509 0.021 0.074  0.216   3.540
## 3000m  0.234  0.554  1.427 0.061 0.216  0.665  10.706
## 42000m 4.334 10.385 28.904 1.220 3.540 10.706 270.270
```

The covariances does not follow the same pattern as the correlations. Important to note is (again) that not all variables have the same scales. For a standardized dataset, pairwise covariances would likely decrease with increased difference in distance.

Scatterplots, extreme values

Below, the scatterplots between each pair of variables are plotted on the off-diagonal. The red dots indicate unusually large squared residuals from a linear model, those that exceed the 95th percentile of a standard normal (standardization is done internally of `scores()`). However, this method does not capture the exteme values on either the x- or y-axis!

```
pairs(x = trackData[2:8],
      panel = function (x, y, ...) {
        m_ = lm(y ~ x)
        s_ = scores(m_$residuals^2, prob = .95) + 1
        points(x, y, pch = c(1,16)[s_], cex = s_,
               col = s_)
      },
      diag.panel = function(x,y,...){
        par(new = TRUE)
        boxplot(x)
      })
```

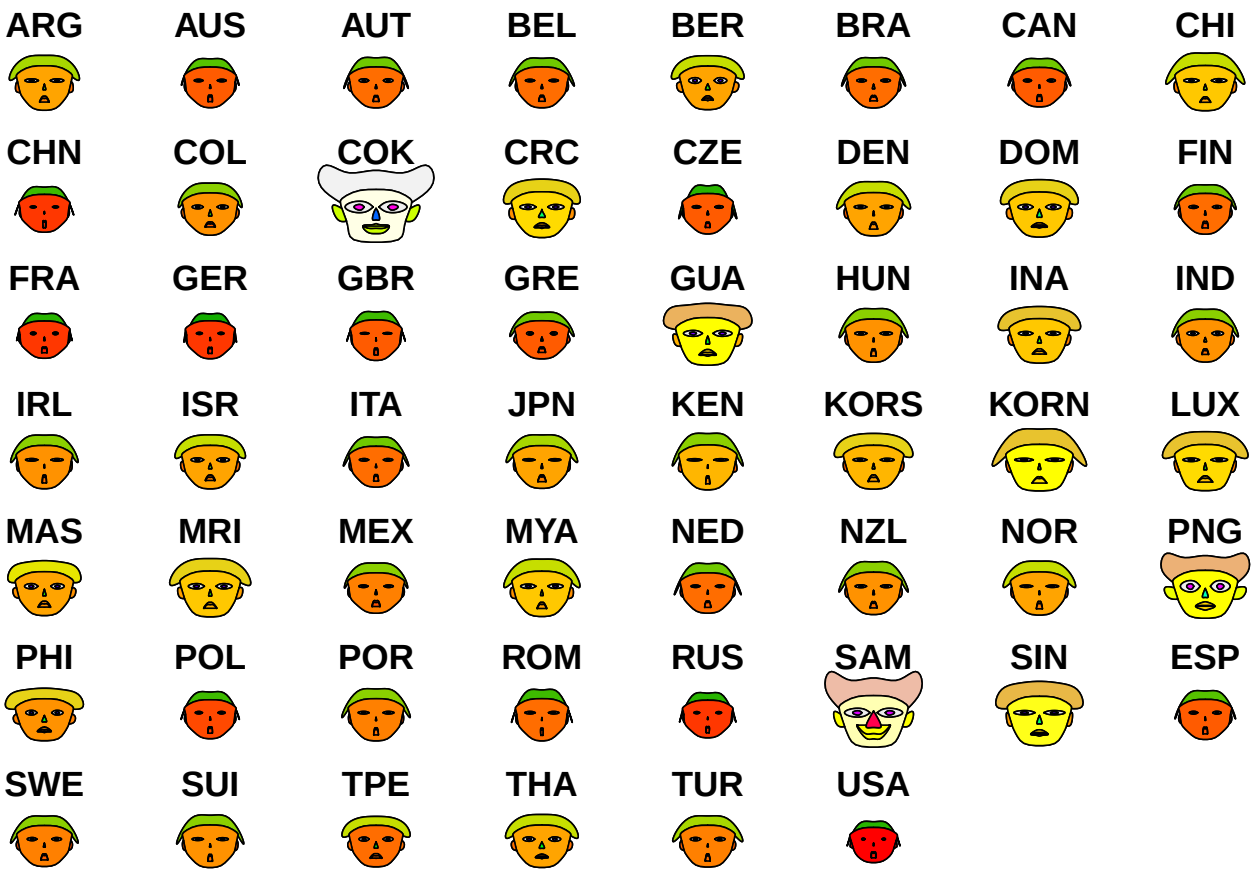


A more suitable measure to use would be some distance measure to the center of observations, such as Mahalanobis distance, but that's `Question 3`... For now, it's sufficient to say that there are outliers present in almost all scatterplots, especially for longer distances.

Other plotting options for multivariate data

Another way to represent multivariate data is to use Chernoff faces. Here, rows (countries) are represented by one face. Exactly how these faces are constructed is a bit tedious to understand from the source code, but the output from `faces()` gives a hint. Since humans are good at recognizing faces, this is a good way to represent many dimensions!

```
column_to_rownames(trackData, "Country") %>%
  faces(.,face.type=1, print.info = TRUE,
        main = "Chernoff faces for running data")
```



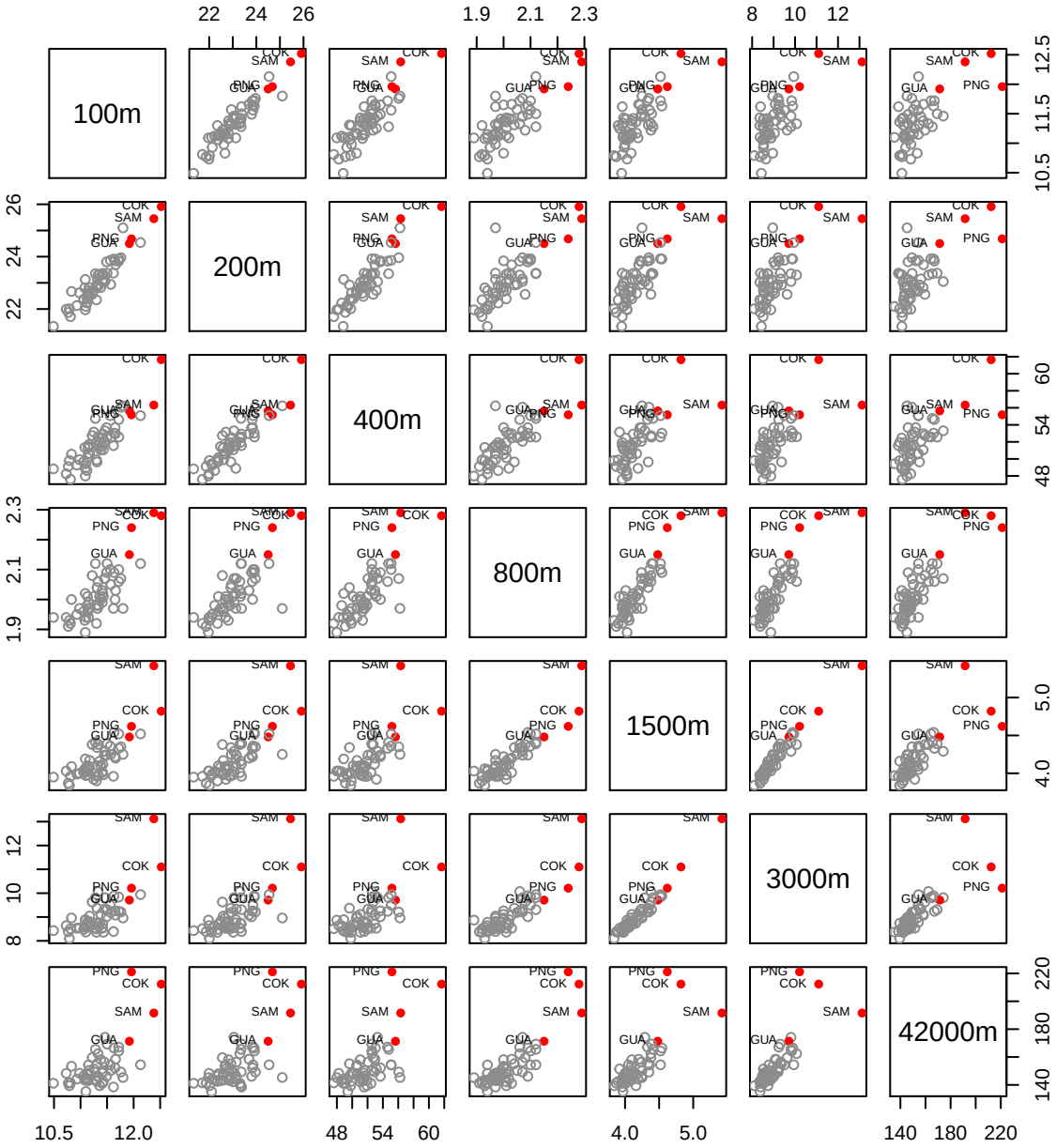

```
## effect of variables:
##  modified item       Var
##  "height of face   " "100m"
##  "width of face    " "200m"
##  "structure of face" "400m"
##  "height of mouth  " "800m"
##  "width of mouth   " "1500m"
##  "smiling          " "3000m"
##  "height of eyes   " "42000m"
##  "width of eyes    " "100m"
##  "height of hair   " "200m"
##  "width of hair    "  "400m"
##  "style of hair    "  "800m"
##  "height of nose   "  "1500m"
##  "width of nose    "  "3000m"
##  "width of ear     "  "42000m"
##  "height of ear    "  "100m"
```

By checking the output, it's possible to determine which countries are similar in specific variables. For example, faces with similar *height of eyes* and *width of ear* are similar in marathon distances. With that in mind, countries COK (Cook Islands), PNG (Papua New Guinea) and SAM (Samoa) have similar marathon running times.

The faces for these three countries, including GUA (Guatemala), seem to stand out from the rest in general. With that information, a scatter plot will be computed, where the four mentioned countries will have filled

points with red coloring and the rest will have non-filled circles with gray coloring.

```
w_ = trackData$Country %in% c("COK", "PNG", "SAM", "GUA") # Index vector
n_ = trackData$Country                                    # Name vector
c_ = ifelse(w_, "red", "gray55")                          # Color vector

pairs(x = trackData[2:8],
      panel = function (x, y, ...) {
        d = data.frame(n = n_, x, y)
        points(x, y, col = c_, pch = c(1,16)[w_+1])
        with(d[w_,], text(x, y, cex = 0.6, labels = n, pos = 2))
      })
```



It seems as the face recognition "method" was a good approach to find outliers. With the exception of GUA, the reamining three countries are outliers in almost all plots.

Question 3 - Examining for extreme values

Which countries seem most extreme?

SAM and COK are believed to be most extreme, followed by PNG, as these are far from the centers in all 2D-scatterplots.

Squared euclidean distance

The distances are computed, and output is sorted from largest distance to smallest distance.

```
X = as.matrix(trackData[2:8])
n = nrow(X)
p = ncol(X)

one = rep(1, n)
xm = (1/n) * (one %*% X)
Xm = one %*% xm
Xc = X - Xm

# Compute Euclidean distances:
Euc_d = diag(Xc %*% t(Xc))
names(Euc_d) = trackData$Country

# Report 5 highest:
Euc_d[order(Euc_d, decreasing = T)][1:5] %>% round(1)
```

```
##    PNG    COK    SAM    BER    GBR
## 4573.5 3554.0 1484.2  425.0  345.6
```

PNG has the largest squared distance. Looking at the scatterplot from 2c reveals that PNG has the highest time for marathon, which will result in higher distances due to the scales of data.

```
w_ = which(trackData$Country %in% c("PNG","COK","SAM","BER","GBR"))
trackData[w_,]
```

```
##    Country  100m  200m  400m 800m 1500m 3000m 42000m
## 5      BER 11.46 23.05 53.30 2.07  4.29  9.81 174.18
## 11     COK 12.52 25.91 61.65 2.28  4.82 11.10 212.33
## 19     GBR 11.10 22.10 49.43 1.94  3.97  8.37 135.25
## 40     PNG 11.96 24.68 55.18 2.24  4.62 10.21 221.14
## 46     SAM 12.38 25.45 56.32 2.29  5.42 13.12 191.58
```

Squared Euclidean distance with scaled data

The distance are calculated and sorted in decreasing order according to the new distance. As a comparison, the Euclidean distances from 3b are included.

```
# Scale centered data:
Xs = Xc / matrix(apply(X,2,sd), nrow = n, ncol = p, byrow = T)

# Compute new distances:
New_d = diag(Xs %*% t(Xs))
names(New_d) = trackData$Country
```

```
# This would give the same output as New_d:
# (Xc %*% (diag(apply(X, 2, var)^(-1)) %*% t(Xc))) %>% diag()

rbind("Euclidean" = Euc_d,
      "Scaled"    = New_d)[,order(New_d,decreasing = T)][,1:5] %>% round(1)
```

```
##              SAM    COK    PNG   USA  SIN
## Euclidean 1484.2 3554.0 4573.5 169.6 13.7
## Scaled      75.6   64.6   34.2  12.9 11.4
```

When looking at the normalized distances, the order of the three first countries are as expected; SAM and COK have (in general) more extreme values than PNG. Here, the as the scales of each variable has lost it's importance.

```
w_ = which(trackData$Country %in% c("PNG","COK","SAM","BER","GBR"))
trackData[w_,]
```

```
##    Country  100m  200m  400m 800m 1500m 3000m 42000m
## 5      BER 11.46 23.05 53.30 2.07  4.29  9.81 174.18
## 11     COK 12.52 25.91 61.65 2.28  4.82 11.10 212.33
## 19     GBR 11.10 22.10 49.43 1.94  3.97  8.37 135.25
## 40     PNG 11.96 24.68 55.18 2.24  4.62 10.21 221.14
## 46     SAM 12.38 25.45 56.32 2.29  5.42 13.12 191.58
```

Mahalanobis distance

Again, the previous results are kept. Output is sorted in decreasing order for the Mahalanobis distance.

```
C = cov(Xc)
Cinv = solve(C)

Mah_d = diag(Xc %*% Cinv %*% t(Xc))
names(Mah_d) = trackData$Country

rbind("Euclidean"   = Euc_d,
      "Other"       = New_d,
      "Mahalanobis" = Mah_d)[,order(Mah_d,decreasing = T)][,1:5] %>% round(1)
```

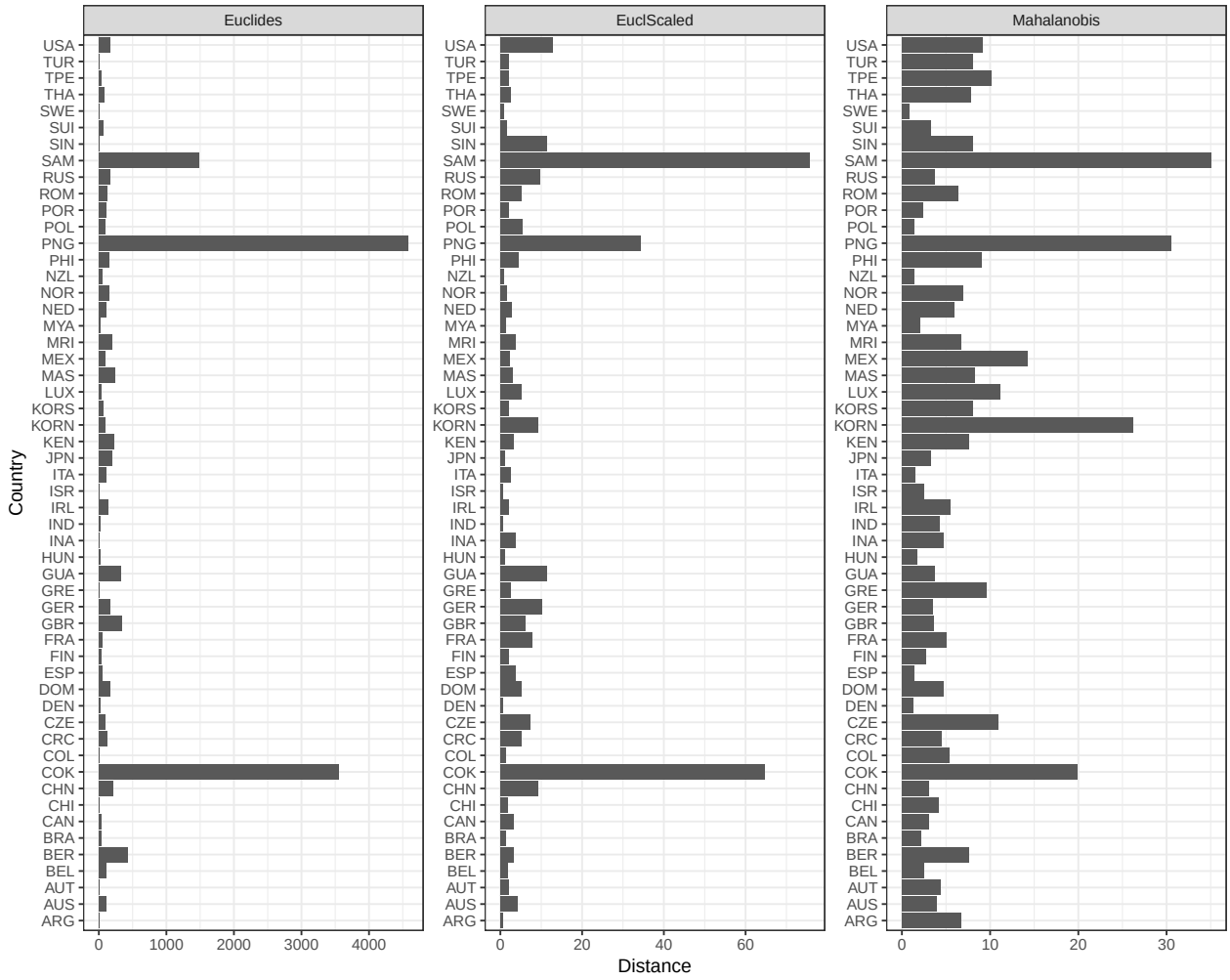
```
##                SAM    PNG KORN    COK   MEX
## Euclidean   1484.2 4573.5 91.2 3554.0 101.1
## Other         75.6   34.2  9.2   64.6   2.3
## Mahalanobis   35.0   30.5 26.2   19.8  14.2
```

SAM is again highest. Here, countries which deviate from the general pattern of others receive higher distances, in addition to just having extreme values. SAM behaves differently when looking at the scatterplots from Question 2, and receives the highest distance.

Czekanowski's diagram

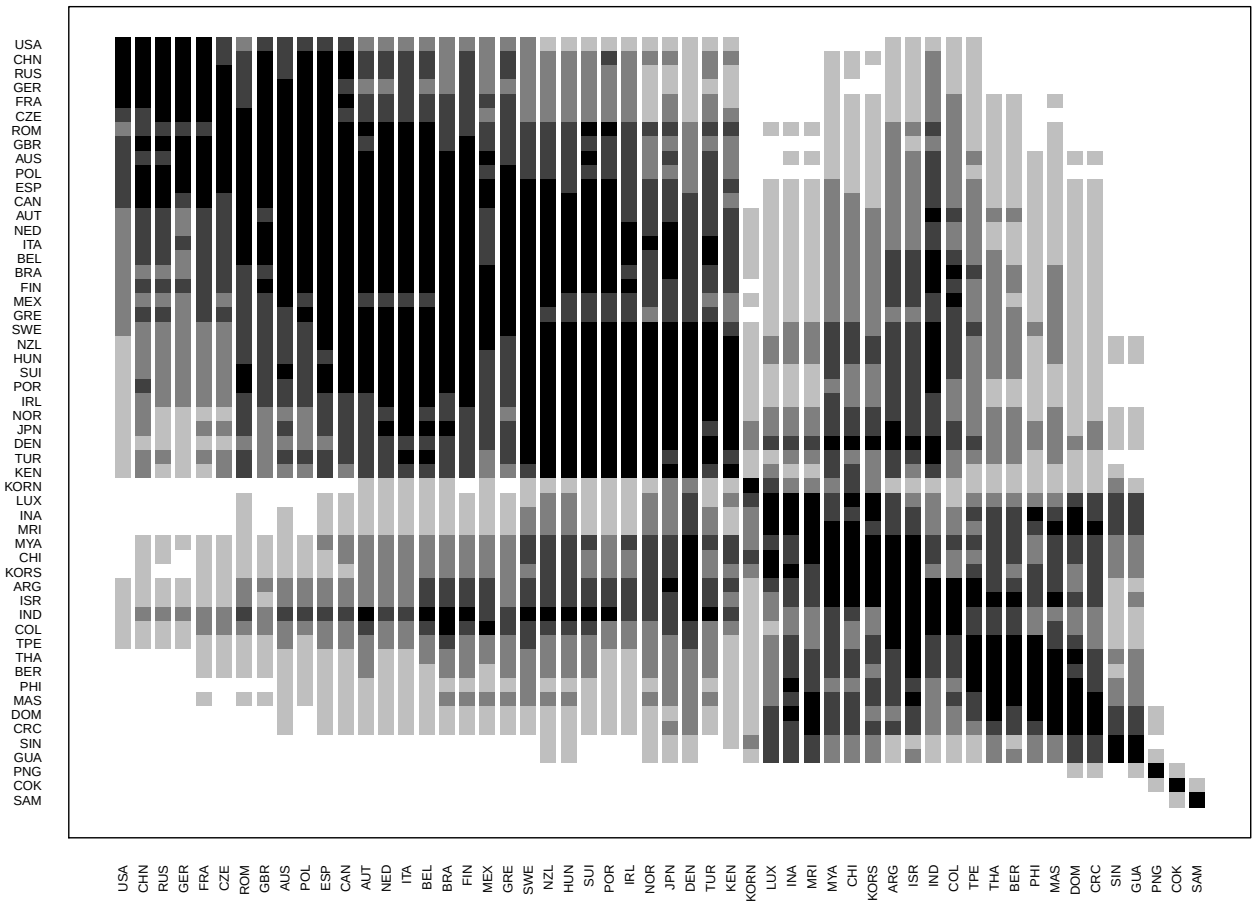
SAM, COK and PNG have been discussed earlier as outliers because of extreme values. However, with the covariance matrix incorporated into the calculations, countries which deviates from the "trend" are punished with higher weights.

```
data.frame("Distance" = c(Euc_d,New_d,Mah_d),
           "Country"  = trackData$Country,
           "Method"   = rep(c("Euclides", "EuclScaled", "Mahalanobis"), each = 54)) %>%
  ggplot(., aes(Country, Distance)) +
  geom_bar(stat = "identity") + theme_bw() +
  facet_wrap(~Method, ncol = 3, scales = "free") + coord_flip()
```



Sweden has very small distances, which means Swedes are close to average when it comes to running.

```
rownames(X) = trackData$Country
czek_matrix(X) %>% plot(., type = "col", ann = TRUE)
```



Sweden are grouped together with NZL, HUN, SUI, POR, IRL, NOR, JPN, DEN and TUR (mostly european countries).