Assignment 3: Principle component and factor analysis

Milda Poceviciute, Henrik Karlsson, Ugurcan Lacin and Ramon Laborda

12 December 2017

Question 1: Principal components, including interpretation of them

a)

```
# Data
mydata <- read.table("T1-9.dat", sep="\t")
X <- mydata[,-1]
# Correlation matrix
R <- cor(X)
# Eigen decomposition
ed <- eigen(R)
```

The sample correlation matrix is:

```
##           V2        V3        V4        V5        V6        V7        V8
## V2 1.0000000 0.9410886 0.8707802 0.8091758 0.7815510 0.7278784 0.6689597
## V3 0.9410886 1.0000000 0.9088096 0.8198258 0.8013282 0.7318546 0.6799537
## V4 0.8707802 0.9088096 1.0000000 0.8057904 0.7197996 0.6737991 0.6769384
## V5 0.8091758 0.8198258 0.8057904 1.0000000 0.9050509 0.8665732 0.8539900
## V6 0.7815510 0.8013282 0.7197996 0.9050509 1.0000000 0.9733801 0.7905565
## V7 0.7278784 0.7318546 0.6737991 0.8665732 0.9733801 1.0000000 0.7987302
## V8 0.6689597 0.6799537 0.6769384 0.8539900 0.7905565 0.7987302 1.0000000
```

The coresponding eigendecomposition is:

```
## Eigen values are:
```

```
## 5.807624 0.6286934 0.2793346 0.1245547 0.09097174 0.05451882 0.01430226
```

```
## Eigen vectors are:
```

```
##
## Eigenvector  1
##
## -0.3777657 -0.4071756 -0.1405803 0.5870629 -0.1670689 0.5396973 0.08893934
##
## Eigenvector  2
##
## -0.3832103 -0.4136291 -0.1007833 0.194075 0.09350016 -0.7449314 -0.2656566
##
## Eigenvector  3
##
## -0.3680361 -0.4593531 0.2370255 -0.6454312 0.3272733 0.240094 0.1266044
##
## Eigenvector  4
##
## -0.394781 0.1612459 0.1475424 -0.295208 -0.8190547 -0.01650651 -0.1952131
##
## Eigenvector  5
```

```
##
## -0.389261 0.3090877 -0.4219855 -0.06669044 0.026131 -0.1889877 0.7307682
##
## Eigenvector  6
##
## -0.3760945 0.4231899 -0.4060627 -0.08015699 0.351698 0.2404997 -0.5715064
##
## Eigenvector  7
##
## -0.3552031 0.3892153 0.741061 0.3210764 0.2470082 -0.04826992 0.08208401
```

b)

The principle components of R are given by:

$Y_i = \vec{e_i}^T * \vec{X}$

Hence, the first two PCAs are:

$Y_1 = -0.3777657 * Z_2 - 0.4071756 * Z_3 - 0.1405803 * Z_4 + 0.5870629 * Z_5 - 0.1670689 * Z_6 + 0.5396973 * Z_7 + 0.08893934 * Z_8$

$Y_2 = -0.4071756 * Z_2 - 0.4136291 * Z_3 - 0.4593531 * Z_4 + 0.1612459 Z_5 + 0.3090877 * Z_6 + 0.4231899 * Z_7 + 0.3892153 * Z_8$

```r
# Normalise data:
Z <- scale(X)
R1 <- cor(Z)
# Eigen decomposition
edZ <- eigen(R1)
lambda1 <- which.max(edZ$values)
lambda2 <- which.max(edZ$values[-1])+1
total_lambda <- sum(edZ$values)
# Find first two PCAs
eiv1 <- edZ$vectors[,lambda1]
eiv2 <- edZ$vectors[,lambda2]
pca1 <- Z %*% eiv1
pca2 <- Z %*% eiv2
```

The first and Second Principal Components are:

```
##            PCA1          PCA2
## 1  -0.393240234 -0.1316106539
## 2   1.931642887  0.4910673439
## 3   1.262520373  0.1931483517
## 4   1.291730279 -0.0024053163
## 5  -1.396108552  0.7607805514
## 6   1.006778878  0.3795169129
## 7   1.734340591  0.2625382896
## 8  -0.811838204 -0.8689689997
## 9   2.989466907  0.0515565410
## 10 -0.001927672  0.9440511396
## 11 -7.906227224 -0.5205487107
## 12 -2.166811506  0.3329829275
## 13  2.406030321  0.7596584086
## 14  0.082495533 -0.7134670147
```

```
## 15 -2.192409809  0.4313474208
## 16  1.266731340  0.4263465242
## 17  2.518345696  1.1230568367
## 18  3.047516603  0.9345292649
## 19  2.442706280 -0.0333740439
## 20  1.197800425  0.7754294368
## 21 -3.294123799 -0.5291973432
## 22  0.788251063 -0.5905189337
## 23 -1.741942057 -0.5146702995
## 24  0.354256642  0.2542124561
## 25  1.035907216 -0.7726532308
## 26 -0.574161730  0.2181299839
## 27  1.547452839 -0.2725521643
## 28  0.481657610 -0.6557135033
## 29  0.917735409 -1.3818382037
## 30 -0.830794629 -0.7687520619
## 31 -1.455347346 -2.3771213453
## 32 -1.721467731 -1.2782741127
## 33 -1.495210140  0.5386190883
## 34 -1.749727754 -0.5254636441
## 35  0.995766285  0.4905095362
## 36 -0.815981458 -0.5990664129
## 37  1.544760622 -0.2873591443
## 38  0.755235487 -0.4320195250
## 39  0.553003461 -0.9934747091
## 40 -5.257449747  1.1953938028
## 41 -1.763533682  0.5797417480
## 42  2.273765780  0.4911613673
## 43  1.175249957 -0.7069615582
## 44  2.123005711 -0.3810120022
## 45  3.042948214  0.4460682284
## 46 -8.213415123  2.0282582323
## 47 -3.093919517 -0.9564211276
## 48  1.889462264  0.2470324869
## 49  0.839149567  0.0001607055
## 50  1.113545239 -0.5263585776
## 51 -0.659093139  1.0063775050
## 52 -1.223805050  0.8469872902
## 53  0.850127798 -0.5785810419
## 54  3.299148823  1.1897213000
```

The total variance explained by the first two PCA are:

```
# Total Sample Variance explained:
# By PCA1:
var_expl1 <- edZ$values[1]/total_lambda
# By PCA2:
var_expl2 <- edZ$values[2]/total_lambda

cat(paste("First PCA explains ", round(var_expl1*100,2), "% of total varaince"))
```

```
## First PCA explains  82.97 % of total varaince
```

```
cat("\n")
```

```r
cat(paste("Second PCA explains ", round(var_expl2*100,2), "% of total varaince"))
```

```
## Second PCA explains  8.98 % of total varaince
```

We can see that the PCA1 captures the majority of the variance in the data, and the sum of the first two PCAs is above 90%. This is sufficient to capture the most available information of the underlying data.

Now the correlation between all PCAs and the correlation between PCAs and the normalised data is calculated.

```r
pca_mat <- matrix()
temp <- matrix()

for (i in 1:nrow(edZ$vectors)){
    temp <- Z %*% edZ$vectors[i,]
    if (i == 1){
      pca_mat <-   temp
    }
    else{
    pca_mat <-  cbind(pca_mat,temp)
    }
}

cor_pca1 <- matrix()
cor_pca2 <- matrix()

for (k in 1:ncol(Z)){
    # Correlation between PCA1 and standardised variables
    cor_pca1[k] <- eiv1[k]*sqrt(edZ$values[1])/sd(Z[,k])
    # Correlation between PCA2 and standardised variables
    cor_pca2[k] <- eiv2[k]*sqrt(edZ$values[2])/sd(Z[,k])
}
```

The table below shows the correlation between the first two PCAs and the 7 variables of the standardised data:

```r
result <- data.frame(PCA1 = cor_pca1, PCA2 = cor_pca2)
rownames(result) <- colnames(Z)
result
```

```
##          PCA1       PCA2
## V2 -0.9103780 -0.3228503
## V3 -0.9234990 -0.3279673
## V4 -0.8869307 -0.3642220
## V5 -0.9513832  0.1278522
## V6 -0.9380805  0.2450762
## V7 -0.9063506  0.3355481
## V8 -0.8560043  0.3086096
```

c)

The PCA2 takes the difference between the short running distance and long distances, hence it could measure the relative strengh of each nations athletes. The PCA1 is more difficult to interpret, but it could measure the athletic excellence of the nations.

We can see that PCA1 has very high absolute correlation with all standardised variables and the PCA2 has much lower absolute correlation with the standardised variables. This is consistent with the fact that PCA1

captures the most underlying variance in the data, while PCA2 explains much less of the total variance in the data.

d)

```
pca_countries <- data.frame(Country = mydata[order(pca1, decreasing = TRUE),1], PCA1 = pca1[order(pca1,
```

The list of the countries ranked by PCA1:

```
##    Country         PCA1
## 1      USA  3.299148823
## 2      GER  3.047516603
## 3      RUS  3.042948214
## 4      CHN  2.989466907
## 5      FRA  2.518345696
## 6      GBR  2.442706280
## 7      CZE  2.406030321
## 8      POL  2.273765780
## 9      ROM  2.123005711
## 10     AUS  1.931642887
## 11     ESP  1.889462264
## 12     CAN  1.734340591
## 13     ITA  1.547452839
## 14     NED  1.544760622
## 15     BEL  1.291730279
## 16     FIN  1.266731340
## 17     AUT  1.262520373
## 18     GRE  1.197800425
## 19     POR  1.175249957
## 20     SUI  1.113545239
## 21     IRL  1.035907216
## 22     BRA  1.006778878
## 23     MEX  0.995766285
## 24     KEN  0.917735409
## 25     TUR  0.850127798
## 26     SWE  0.839149567
## 27     HUN  0.788251063
## 28     NZL  0.755235487
## 29     NOR  0.553003461
## 30     JPN  0.481657610
## 31     IND  0.354256642
## 32     DEN  0.082495533
## 33     COL -0.001927672
## 34     ARG -0.393240234
## 35     ISR -0.574161730
## 36     TPE -0.659093139
## 37     CHI -0.811838204
## 38     MYA -0.815981458
## 39  KOR, S -0.830794629
## 40     THA -1.223805050
## 41     BER -1.396108552
## 42  KOR, N -1.455347346
## 43     MAS -1.495210140
```

```
## 44     LUX -1.721467731
## 45     INA -1.741942057
## 46     MRI -1.749727754
## 47     PHI -1.763533682
## 48     CRC -2.166811506
## 49     DOM -2.192409809
## 50     SIN -3.093919517
## 51     GUA -3.294123799
## 52     PNG -5.257449747
## 53     COK -7.906227224
## 54     SAM -8.213415123
```

The top 5 countries are consistent with our view on which countries' athletes are the best performing at various running distances. Hence, we conlcude that assumption that PCA1 is showing the overall athletic excellence of the country is reasonable.

Question 2 - Factor analysis

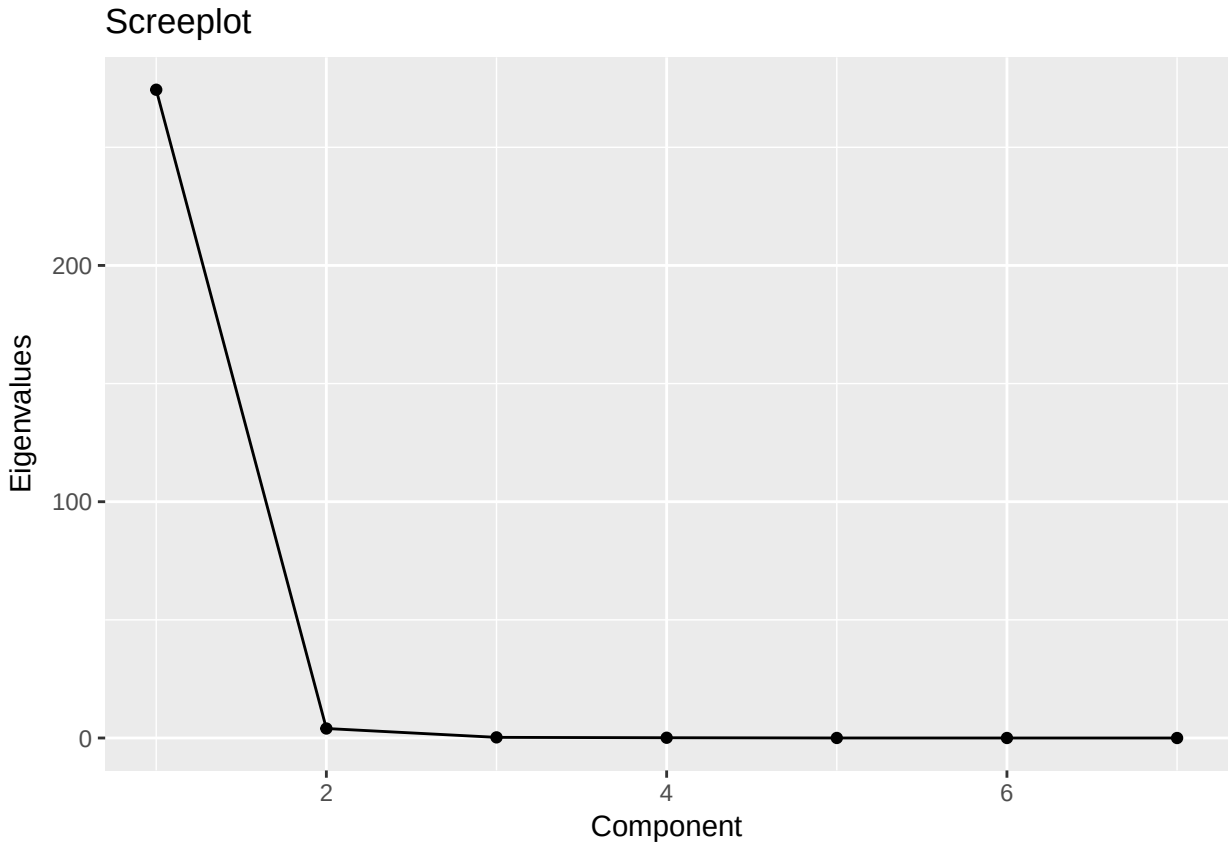
9.28

Perform a factor analysis on the national track records for women. Use the sample covariance matrix S and interpret factors. Compute the factor scores, and check for outliers in the data. Repeat the analysis with the sample correlation matrix R. Does it make a difference if R, rather than S is factored? Explain

```
library(ggplot2)
library(psych)
df <- read.table("T1-9.dat", sep = "\t")
names(df) <- c("country", "m100", "m200", "m400", "m800", "m1500", "m3000", "marathon")

X <- df[,2:8]

S <- cov(X)
R <- cor(X)

ed <- eigen(S)

# Test cummulative sum of variance explained by each variable
cum_var <- unlist(lapply(1:7, function(x){
    sum(ed$values[x])/sum(ed$values)
}))
cum_var
```

```
## [1] 9.841530e-01 1.440805e-02 9.622529e-04 4.108832e-04 5.425166e-05
## [6] 9.306974e-06 2.212396e-06
```

```
qplot(y = ed$values, x = 1:7, geom = "line") +
    geom_point() +
    labs(title = "Screeplot", y = "Eigenvalues", x = "Component")
```



The vector above shows how much of variance that is explained by each variable. The third variable adds less than 0.1% to the model, so the factor analysis will be performed with 2 factors.

Also, by looking at the screeplot, there is an "elbow" at to components, that also suggests that our factor analysis should be perfomed with a 2 factor solution.

Next off, we test if there is outliers in the data. We do so by calculating the Malanobis distance and perform a chi-square test.

```
# Chech for outliers in data, malahanobis vs chisq
X_avg <- colMeans(X)
resX_avg <- t(t(X)-X_avg)

# Compute Mahalanobis distance per row so there is a number per country
dist_maha <- resX_avg %*% solve(cov(X)) %*% t(resX_avg)
dist <- diag(dist_maha)


# Chi square value with
c2 <- qchisq(.95, df=ncol(X))
result_df <- data.frame(countries = df[,1], test_diff = dist-c2)
outliers <- subset(result_df, result_df[,2]>0)
outliers
```

```
##    countries  test_diff
## 11       COK  5.7668602
## 31    KOR, N 12.1000010
## 35       MEX  0.1637918
## 40       PNG 16.4401072
```

```
## 46       SAM 20.9469226
```

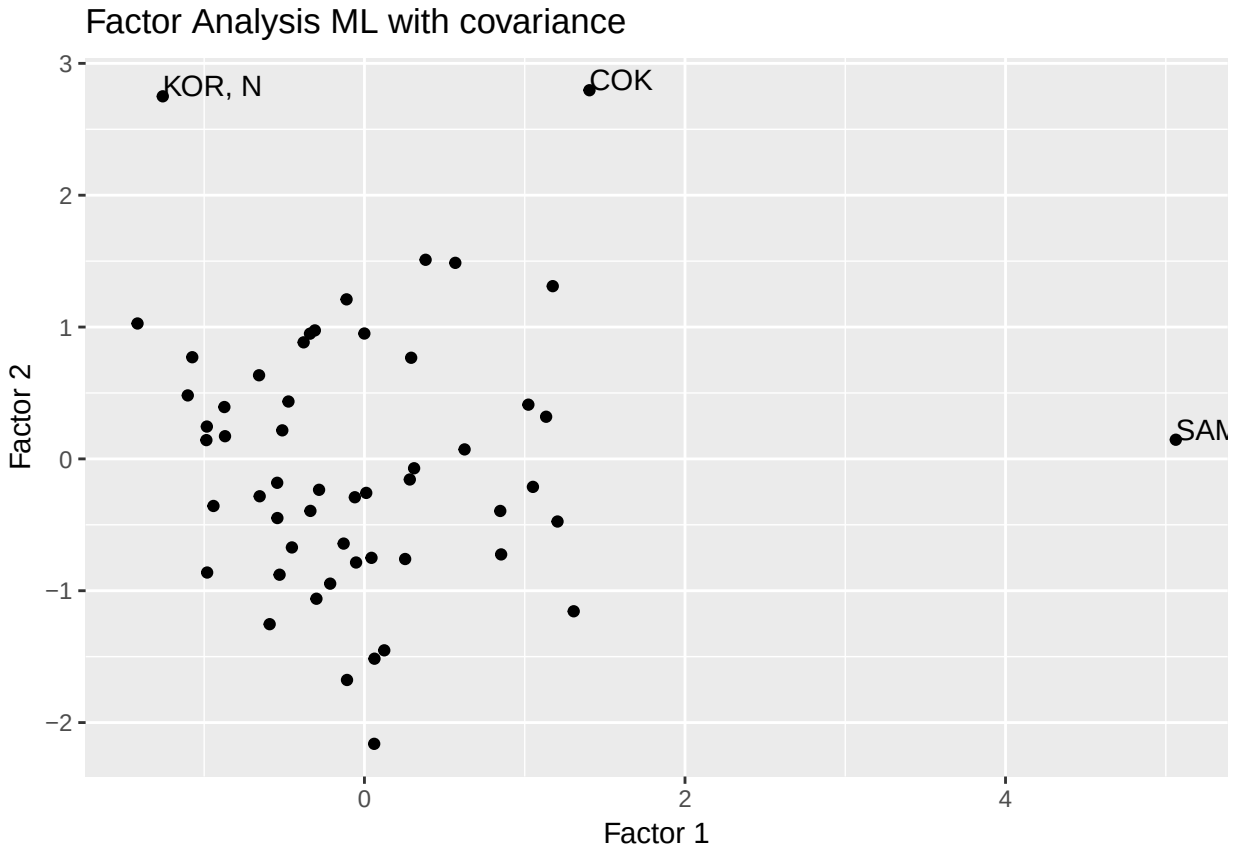
5 countries are considered being outliers in the data set, they are listed above.

Factor analysis with maximum likelihood

```
# ML factor analysis ------------------------------------------------------

fac_ml_S <- factanal(x = X, factors = 2, covmat = S, n.obs = 54, rotation = "varimax")
fac_ml_S
```

```
##
## Call:
## factanal(x = X, factors = 2, covmat = S, n.obs = 54, rotation = "varimax")
##
## Uniquenesses:
##     m100     m200     m400     m800    m1500    m3000 marathon
##    0.094    0.024    0.152    0.144    0.016    0.028    0.338
##
## Loadings:
##          Factor1 Factor2
## m100     0.461   0.833
## m200     0.455   0.877
## m400     0.401   0.829
## m800     0.732   0.566
## m1500    0.882   0.454
## m3000    0.918   0.361
## marathon 0.693   0.427
##
##                Factor1 Factor2
## SS loadings      3.216   2.987
## Proportion Var   0.459   0.427
## Cumulative Var   0.459   0.886
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 31.43 on 8 degrees of freedom.
## The p-value is 0.000118
```

```
fs_ml_S <- factor.scores(X, fac_ml_S)
gg_fs_ml_S <- data.frame(fac1 = fs_ml_S$scores[,1],
                         fac2 = fs_ml_S$scores[,2],
                         country = df$country)
ggplot(gg_fs_ml_S, aes(x = fac1, y = fac2)) +
    geom_point() +
    geom_text(aes(label=ifelse(fac1 > 2 | fac2 > 2, as.character(country),"")), hjust=0,vjust=0) +
    labs(title = "Factor Analysis ML with covariance", x = "Factor 1", y = "Factor 2")
```



The results for the maximum likelihood factor analysis based on the covariance matrix with two factors is presented above. When we analyse the loadings the first factor, it can be seen that it's is stronger with the longer running distances and sprinting distances have weaker loadings. The situation is the opposite in the second factor. The first factor can be interpreted as an endurance factor and the second factor as speed/strengh. The distance 800m have relativly strong loadings in both factors, but a bit higher in the first factor, so we consider it to belong to the first factor. It is intuitive that 800m meter is having high loadings in both factors, since 800m is being considered as one of the toughest running distances there is, which require both speed/strenght and endurance.

The first factor explains 45.9% of the variance and the second explains 42.7%. In total does the factor analysis explain 88.6% of the variance.

The graph shows the factor scores for each country. We are trying to see if there are country that are an outlier by visually look at the data. In the plot, 3 countries that seem to be outliers, KORN, COK and SAM.

```
fac_ml_R <- factanal(x = X, factors = 2, covmat = R, n.obs = 54, rotation = "varimax")
fac_ml_R
```

```
##
## Call:
## factanal(x = X, factors = 2, covmat = R, n.obs = 54, rotation = "varimax")
##
## Uniquenesses:
##     m100     m200     m400     m800    m1500    m3000 marathon
##    0.094    0.024    0.152    0.144    0.016    0.028    0.338
##
## Loadings:
##          Factor1 Factor2
```

```
## m100     0.461   0.833
## m200     0.455   0.877
## m400     0.401   0.829
## m800     0.732   0.566
## m1500    0.882   0.454
## m3000    0.918   0.361
## marathon 0.693   0.427
##
##                Factor1 Factor2
## SS loadings      3.216   2.987
## Proportion Var   0.459   0.427
## Cumulative Var   0.459   0.886
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 31.43 on 8 degrees of freedom.
## The p-value is 0.000118
```

```
# fs_ml_R <- factor.scores(X, fac_ml_R)
# gg_fs_ml_R <- data.frame(fac1 = fs_ml_R$scores[,1],
#                          fac2 = fs_ml_R$scores[,2],
#                          country = df$country)
# ggplot(gg_fs_ml_R, aes(x = fac1, y = fac2)) +
#     geom_point() +
#     geom_text(aes(label=ifelse(fac1 > 2 | fac2 > 2, as.character(country),"")), hjust=0,vjust=0) +
#     labs(title = "Factor Analysis ML with correlation", x = "Factor 1", y = "Factor 2")
```

When the maximum likelihood factor analysis is computed by the correlation matrix instead of the covariance matrix, we can see that the results are identical.

Our best guess why this happens, is because we believe that the maximum likelihood estimation somehow standardize the covariance matrix and/ or the correlation matrix before the factor analysis is calculated, so that the output of the two options becomes identical. The correlation matrix is a standardization of the covariance matrix, so both matrices contain the same type of information.

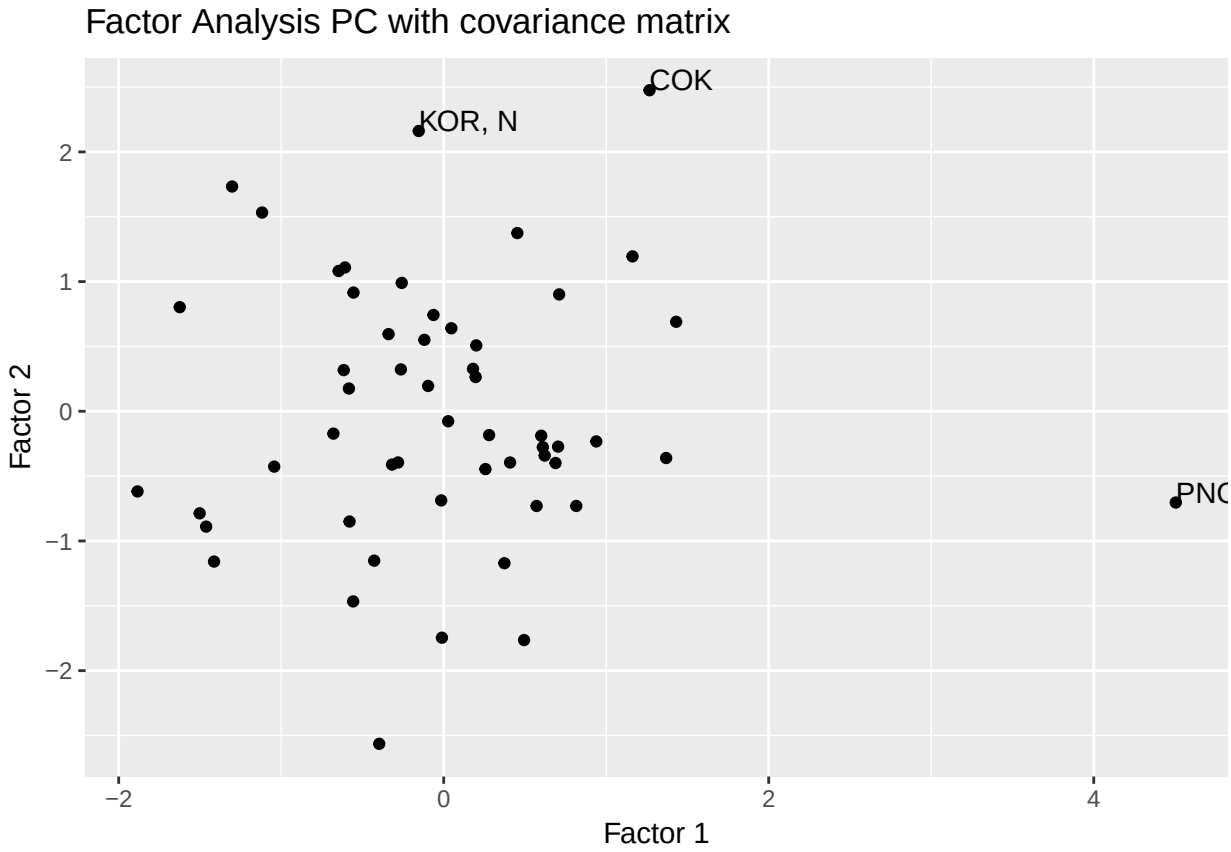
Factor analysis with Principal Components

```
library(psych)
fac_pc_S <- principal(r = S, nfactors = 2, rotate = "varimax", covar = TRUE)
fac_pc_S
```

```
## Principal Components Analysis
## Call: principal(r = S, nfactors = 2, rotate = "varimax", covar = TRUE)
## Unstandardized loadings (pattern matrix) based upon covariance matrix
##            RC1  RC2      h2      u2   H2      U2
## m100      0.17 0.31 1.2e-01 0.03100 0.80 2.0e-01
## m200      0.40 0.77 7.5e-01 0.11435 0.87 1.3e-01
## m400      1.04 2.38 6.7e+00 0.02014 1.00 3.0e-03
## m800      0.06 0.05 6.3e-03 0.00126 0.83 1.7e-01
## m1500     0.18 0.14 5.2e-02 0.02200 0.70 3.0e-01
## m3000     0.56 0.37 4.5e-01 0.21213 0.68 3.2e-01
## marathon 15.54 5.37 2.7e+02 0.00026 1.00 9.5e-07
##
##                          RC1   RC2
## SS loadings           243.00 35.37
```

```
## Proportion Var          0.87  0.13
## Cumulative Var          0.87  1.00
## Proportion Explained    0.87  0.13
## Cumulative Proportion   0.87  1.00
##
##  Standardized loadings (pattern matrix)
##          item  RC1  RC2   h2      u2
## m100        1 0.44 0.78 0.80 2.0e-01
## m200        2 0.43 0.82 0.87 1.3e-01
## m400        3 0.40 0.92 1.00 3.0e-03
## m800        4 0.70 0.58 0.83 1.7e-01
## m1500       5 0.66 0.52 0.70 3.0e-01
## m3000       6 0.69 0.46 0.68 3.2e-01
## marathon    7 0.95 0.33 1.00 9.5e-07
##
##                  RC1  RC2
## SS loadings     2.83 3.05
## Proportion Var  0.40 0.44
## Cumulative Var  0.40 0.84
## Cum. factor Var 0.48 1.00
##
## Mean item complexity =  1.6
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.02
##
## Fit based upon off diagonal values = 1
```

```r
fs_pc_S <- factor.scores(X, fac_pc_S)
gg_fs_pc_S <- data.frame(fac1 = fs_pc_S$scores[,1],
                         fac2 = fs_pc_S$scores[,2],
                         country = df$country)
ggplot(gg_fs_pc_S, aes(x = fac1, y = fac2)) +
    geom_point() +
    geom_text(aes(label=ifelse(fac1 > 2 | fac2 > 2, as.character(country),"")), hjust=0,vjust=0) +
    labs(title = "Factor Analysis PC with covariance matrix", x = "Factor 1", y = "Factor 2")
```



When factor analysis with Principal Components with the covariance matrix is performed, we can see that the unstandardized loadings is very effected by the covariance matrix, were marathon with a high variance have much stronger loading than the other distances. In the unstandardized case, we can see that the first factor explains 87% of the variance and the result doesn't provide us with any useful information.

However, if we analyse the standardized result, we can see that the pattern is similar as the factor analysis with maximum likelihood estimation and the same interpretation of the factors can be drawn. Here, the 1500m have relativly high loadings in both factors.

The first factor explains 40% of the variance and the second explains 44%.

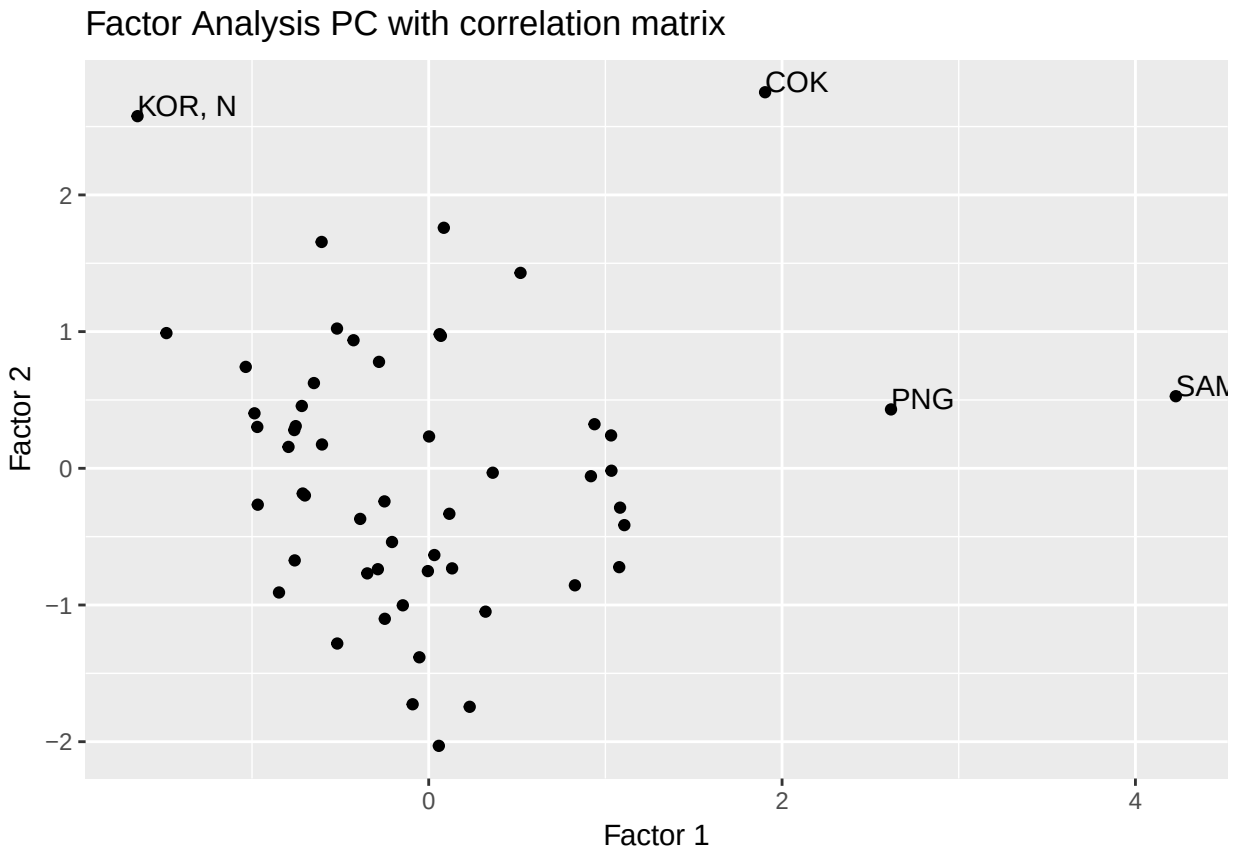
When looking at the plot, one can see that there still is 3 countries that is further away from the center, still KORN, COK and PNG. But in this case, all countries are more scattered and it's harder to distinguish what country that could be considered an outlier, compared to the maximum likelihood case.

```
fac_pc_R <- principal(r = R, nfactors = 2, rotate = "varimax", covar = FALSE)
fac_pc_R
```

```
## Principal Components Analysis
## Call: principal(r = R, nfactors = 2, rotate = "varimax", covar = FALSE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##           RC1  RC2   h2    u2 com
## m100     0.43 0.86 0.93 0.067 1.5
## m200     0.44 0.88 0.96 0.040 1.5
## m400     0.39 0.88 0.92 0.081 1.4
## m800     0.77 0.57 0.92 0.079 1.8
## m1500    0.85 0.48 0.94 0.060 1.6
## m3000    0.89 0.39 0.93 0.066 1.4
## marathon 0.83 0.37 0.83 0.172 1.4
```

```
##
##                         RC1  RC2
## SS loadings            3.31 3.13
## Proportion Var         0.47 0.45
## Cumulative Var         0.47 0.92
## Proportion Explained   0.51 0.49
## Cumulative Proportion  0.51 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.03
##
## Fit based upon off diagonal values = 1
```

```
fs_pc_R <- factor.scores(X, fac_pc_R)
gg_fs_pc_R <- data.frame(fac1 = fs_pc_R$scores[,1],
                         fac2 = fs_pc_R$scores[,2],
                         country = df$country)
ggplot(gg_fs_pc_R, aes(x = fac1, y = fac2)) +
    geom_point() +
    geom_text(aes(label=ifelse(fac1 > 2 | fac2 > 2, as.character(country),"")), hjust=0,vjust=0) +
    labs(title = "Factor Analysis PC with correlation matrix", x = "Factor 1", y = "Factor 2")
```



When performing the factor analysis with principal components based on the correlation matrix, the results are similar to both the maximum likelihood method and principal component with covariance matrix. We do not receive an unstandardized factor loading here, because the corrlation matrix is a standardization of the covariance matrix.

The results here still fit with our earlier description of the two factors, and I'd say, it's the easiest case to interpret the two factors.

The first factor explains 47% of the variance and the second explains 45%.

When looking at the scatterplot, most countries seem less scattered compared to PC with covariance matrix, however, now 4 countries seem to be outliers, KORN, COK, PNG and SAM.

What does it mean that the rotation is set to "varimax" by default?

Rotation is being used to easier differentiate between different factors. The rotation does not change the position of variables relative to eachother when the rotation is performed, which imples that correlation between factors are persevered. Instead, the loadings change in a rotation.

Varimax rotation is a orthogonal rotation of the loading matrix. An orthogonal rotation changes the factor variance but factors remain uncorrelated and variable communalities is perserved. What the Varimax rotation does, is that it tries to maximize variance among the squared loadings of each factor.