Multivariate Statistical Methods Lab 3

Karo Ziomek, Joshua Hudson, Carles Sans Fuentes

12 de diciembre de 2017

R Markdown

Question 1: Principal components, including interpretation of them

Here I write the code to process preliminary the data

```
link <- "C:/Users/Carles/Desktop/MasterStatistics-MachineLearning/Master_subjects/Multivariate_Statisti
# link='~/LIU/Semester3/P2/MVS/L1/T1-9.dat'
data <- t(read.table(link))
colnames(data) <- c(data[1, ])
data <- data[2:nrow(data), ]
## Preparing data
mydata <- apply(data, 2, as.numeric)
mydata <- t(mydata)
colnames(mydata) <- c("hundred", "twohundred", "fourhundred",
    "eighthundred", "1500", "3000", "marathon")
## preview of mydata
mydata <- as.data.frame(mydata)
mydata$hundred <- mydata$hundred/60
mydata$twohundred <- mydata$twohundred/60
mydata$fourhundred <- mydata$fourhundred/60
```

(a) Obtain the sample correlation matrix R for these data, and determine its eigenvalues and eigenvectors.

```
corMat <- cor(mydata)  #correlation Matrix
eigenvalues <- eigen(corMat)$values
eigenvectors <- eigen(corMat)$vectors
```

(b) Determine the first two principal components for the standardized variables. Prepare a table showing the correlations of the standardized variables with the components, and the cumulative percentage of the total (standardized) sample explained by the two components.

```
# Standardizing
standardized <- apply(mydata, 2, FUN = function(x) {
    x - mean(x)/sd(x)
})
corMat <- cor(standardized)  #correlation Matrix
eigenvalues <- eigen(corMat)$values
eigenvectors <- eigen(corMat)$vectors
percExpl <- eigenvalues/sum(eigenvalues)  #Explanation from the total variance of each Principal compon
Cumulative <- sum(percExpl[1:2])
PC1 <- eigenvectors[, 1]
PC2 <- eigenvectors[, 2]

corrXY <- data.frame(matrix(nrow = 2, ncol = 7))
for (i in 1:2) {
    for (k in 1:7) {
```

```
        corrXY[i, k] = eigenvectors[i, k] * sqrt(eigenvalues[i])
    }
}
rownames(corrXY) <- c("PC1", "PC2")
colnames(corrXY) <- colnames(corMat)
corrXY$CumVarProp <- c(eigenvalues[1]/sum(eigenvalues), (eigenvalues[1] +
    eigenvalues[2])/sum(eigenvalues))

library(knitr)
kable(corrXY, caption = "Correlations between the first 2 PCs and the variables, as well as the Cumulati
```

Table 1: Correlations between the first 2 PCs and the variables, as well as the Cumulative Proportion of the sample variance

	hundred	twohundred	fourhundred	eighthundred	1500	3000	marathon	CumVarProp
PC1	-0.9103780	-0.9812531	-0.3387846	1.4147638	-0.4026196	-1.3006173	0.2143350	0.8296606
PC2	-0.3038482	-0.3279673	-0.0799112	0.1538824	0.0741365	0.5906575	-0.2106396	0.9194740

(c) Interpret the two principal components obtained in Part b. (Note that the first component is essentially a normalized unit vector and might measure the athletic excellence of a given nation. The second component might measure the relative strength of a nation at the various running distances.)

The first principal component is -0.3777657, -0.3832103, -0.3680361, -0.394781, -0.389261, -0.3760945, -0.3552031. This means we are taking the negative sum of all our competitions that maximize the variance.

The second principal component is -0.4071756, -0.4136291, -0.4593531, 0.1612459, 0.3090877, 0.4231899, 0.3892153. This means we are taking the first 3 races (the ones with less distance) against the ones with more distances (from 1500 on) on similar weights but in different sign, having the one in the middle (800 middle) as well negative accounting though less than the other variables.

These first 2 PCs account for around 92% of the variance, so it seems reasonable to keep only these 2.

(d) Rank the nations based on their score on the first principal component. Does this ranking correspond with your inituitive notion of athletic excellence for the various countries?

```
Countryxeigen <- apply(mydata, 1, FUN = function(x) {
    x * PC1
})
dim(mydata)
```

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

```
rankCountry <- apply(Countryxeigen, 2, sum)
rankCountry[order(rankCountry, decreasing = TRUE)]
```

```
##       GBR       KEN       CHN       JPN       RUS       USA       NOR
## -54.01460 -55.19828 -55.32885 -55.75782 -56.11461 -56.11626 -56.21781
##       GER       IRL       ROM       POR       BEL       NED       ITA
## -56.24218 -56.53970 -56.55653 -56.95274 -57.02416 -57.04265 -57.04388
##       AUS       POL       MEX       CZE       SUI      KORN       ESP
## -57.07767 -57.25667 -57.45993 -57.72726 -57.77936 -57.99595 -58.08203
##       NZL      KORS       BRA       FIN       FRA       CAN       HUN
## -58.22683 -58.32000 -58.68648 -58.74843 -58.75398 -58.76428 -58.84872
##       DEN       LUX       SWE       ARG       TUR       CHI       GRE
```

```
## -59.26873 -59.55226 -59.64315 -59.85548 -59.85779 -60.54232 -60.76270
##       AUT       INA       COL       SIN       ISR       IND       MYA
## -60.99577 -61.46642 -61.66480 -61.75264 -62.05545 -62.49989 -62.67208
##       TPE       THA       CRC       PHI       MRI       DOM       MAS
## -63.35167 -64.52970 -65.20881 -65.57805 -65.88576 -65.99017 -66.72007
##       GUA       BER       SAM       COK       PNG
## -67.67430 -68.59219 -76.58396 -82.99372 -85.64362
```

We would have expected Russia and USA to be top, and Norway, Japan not so high as they appear here; however we have very limited knowledge on track races.

Question 2: Factor analysis

Solve Exercise 9.28 of Johnson, Wichern, the same data as above. Try both PC and ML as estimation methods. Notice that R's factanal() only does ML estimation. For the PC method you can use the principal() function of the psych package. What does it mean that the parameter rotation of factanal() is set to "varimax" by default (equivalently rotate of principal())? Do not forget to check the adequacy of your model Tip: Read section "A Large Sample Test for the Number of Common Factors".

Activity question: Use the sample covariance matrix S and interpret the factors. Compute factor scores, and check for outliers in the data. Repeat the analysis with the sample correlation matrix R

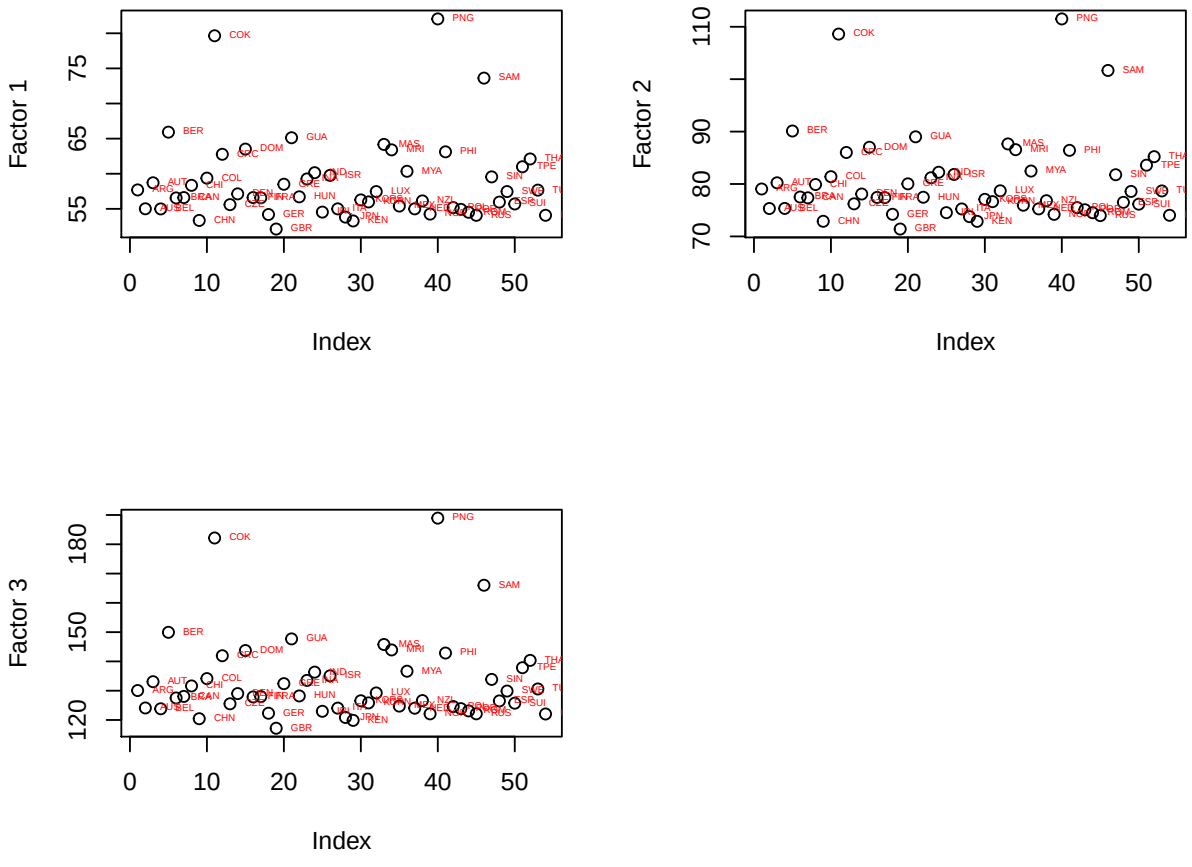
Covariance matrix, Principal Component estimation

```
## Use the sample covariance matrix S and interpret the
## factors. Compute factor scores, and check for outliers in
## the data. Repeat the analysis with the sample correlation
## matrix R.
library(psych)


######################### Evaluation


Evaluation <- function(factmod, mydata, m = 3) {
    p = dim(mydata)[2]
    n = dim(mydata)[1]
    L <- factmod$loadings[, 1:m]
    num <- tcrossprod(L) + diag(CovPrinc$uniquenesses)
    T <- (n - 1 - (2 * p + 4 * m + 5)/6) * log(det(num))/det((n -
        1)/n * covMat)
    T0 <- qchisq(0.95, df = (((p - m)^2 - p - m)/2))

    if (T > T0) {
        print(" Bad, we have to reject H0. m is not good enough")
    } else {
        print("Good, we cannot reject H0. m is good")
    }
    # so given that is False, we cannot reject H0, and therefore
    # our choice is good print results Maximum Likelihood Factor
    # Analysis entering raw data and extracting 2 factors, with
```

```
    # varimax rotation

}
##### Covariance
covMat <- cov(mydata)
CovPrinc <- principal(covMat, nfactors = 3, rotate = "varimax")
Evaluation(CovPrinc, mydata, 3)
```

```
## [1] "Good, we cannot reject H0. m is good"
```

```
Covpload <- CovPrinc$loadings[, 1:3]
scores1 <- as.matrix(mydata) %*% Covpload
par(mfrow = c(2, 2))
for (i in 1:3) {
    plot(scores1[, i], ylab = paste0("Factor ", i))
    text(scores1[, i], rownames(scores1), cex = 0.4, pos = 4,
        col = "red")
}
```



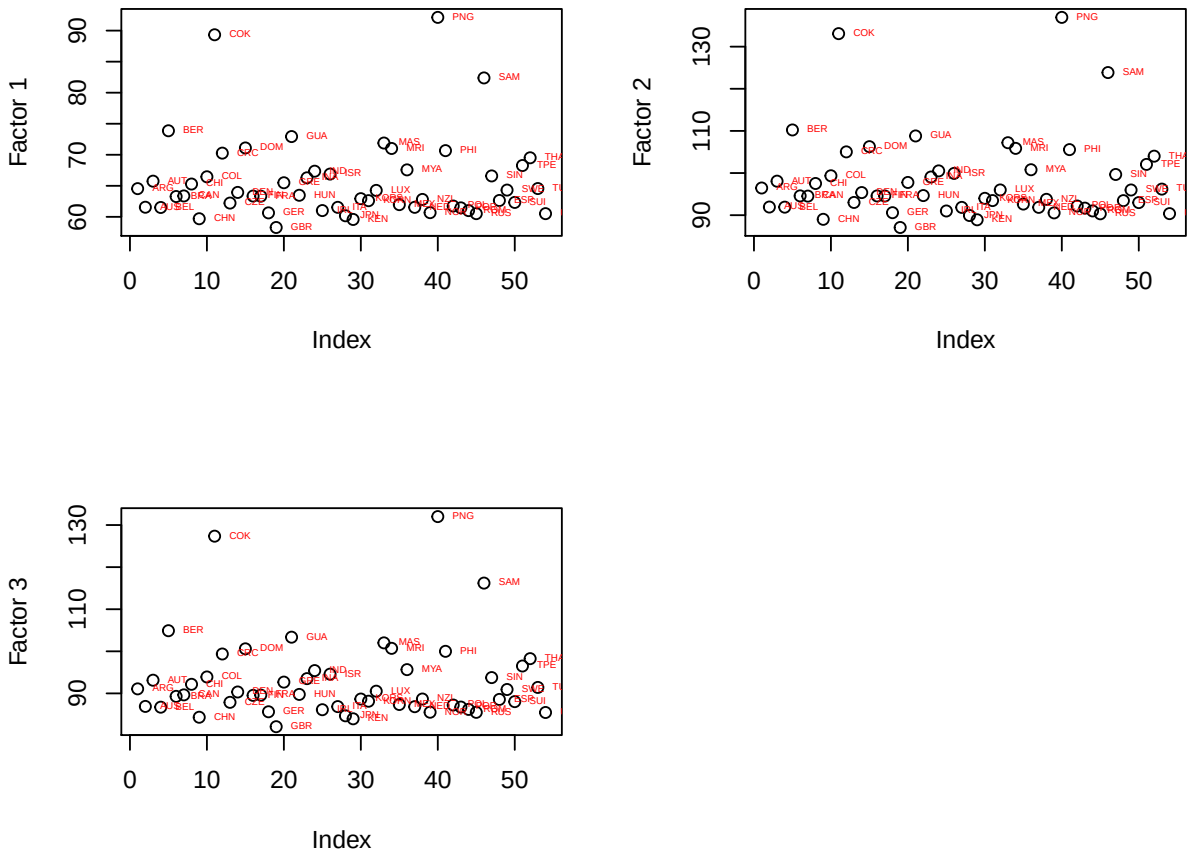
From the factor score plots, we can see that COK, PNG and SAM are clear outliers.

Covariance matrix, Maximum Likelihood estimation

```
CovFact <- factanal(covmat = covMat, factors = 3, rotation = "varimax")
Evaluation(CovFact, mydata, 3)
```

```
## [1] "Good, we cannot reject H0. m is good"
```

```
# plot factor 1 by factor 2
Covfload <- CovFact$loadings[, 1:3]
scores2 <- as.matrix(mydata) %*% Covfload
par(mfrow = c(2, 2))
for (i in 1:3) {
    plot(scores2[, i], ylab = paste0("Factor ", i))
    text(scores2[, i], rownames(scores2), cex = 0.4, pos = 4,
        col = "red")
}
```



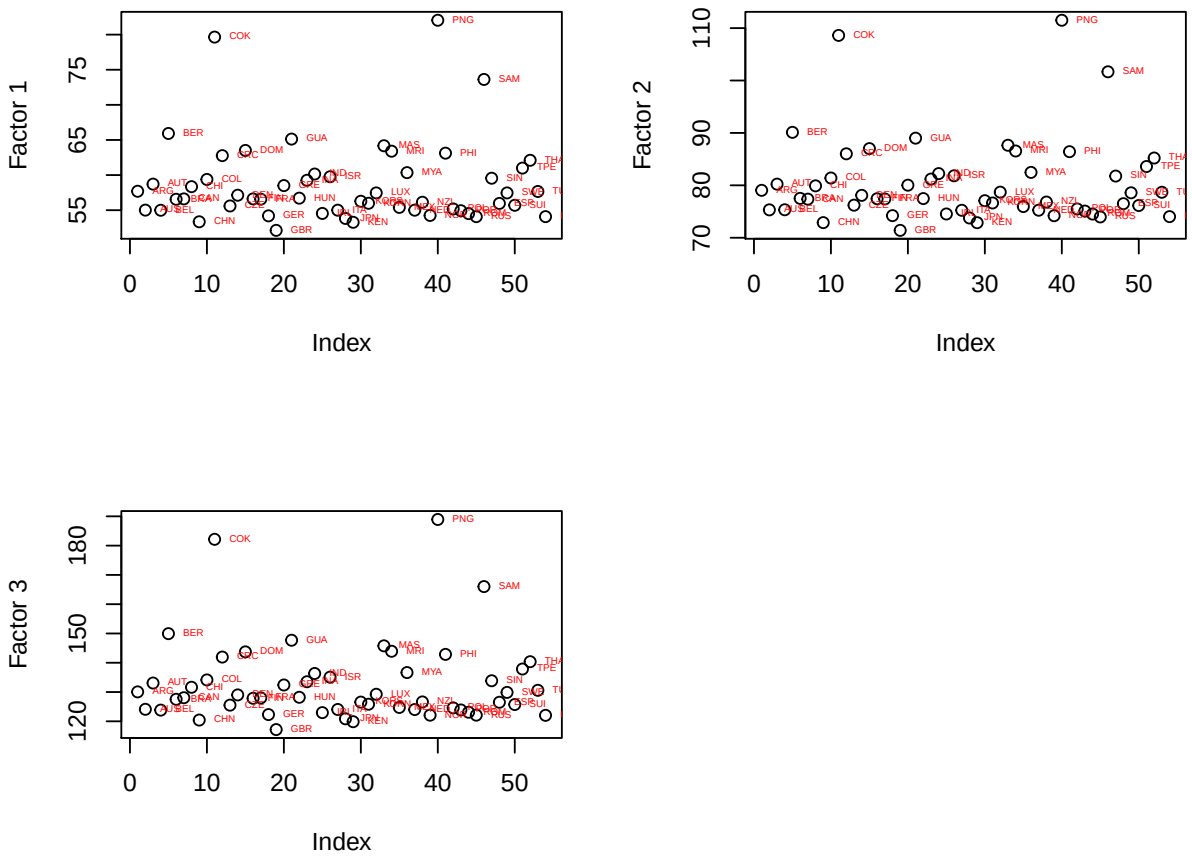
Again, we can see that COK, PNG and SAM are clear outliers.

Correlation matrix, Principal Component estimation

```
##### Correlation
corMat <- cor(mydata)
CorPrinc <- principal(corMat, nfactors = 3, rotate = "varimax")
Evaluation(CorPrinc, mydata, 3)
```

```
## [1] "Good, we cannot reject H0. m is good"
```

```
Corpload <- CorPrinc$loadings[, 1:3]
scores3 <- as.matrix(mydata) %*% Corpload
par(mfrow = c(2, 2))
for (i in 1:3) {
    plot(scores3[, i], ylab = paste0("Factor ", i))
    text(scores3[, i], rownames(scores3), cex = 0.4, pos = 4,
        col = "red")
}
```



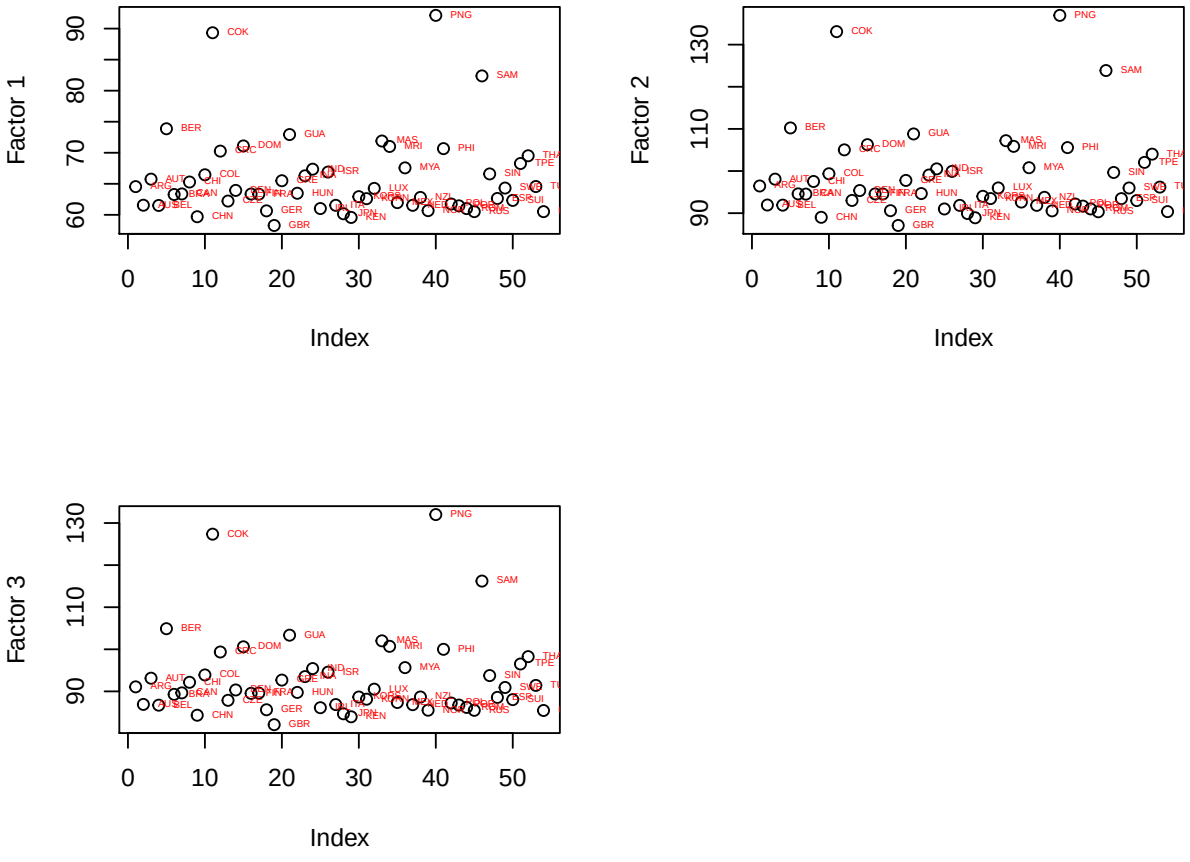
Once again, we can see that COK, PNG and SAM are clear outliers.

Correlation matrix, Maximum Likelihood estimation

```
CorFact <- factanal(covmat = corMat, factors = 3, rotation = "varimax")
Evaluation(CorFact, mydata, 3)
```

```
## [1] "Good, we cannot reject H0. m is good"
```

```
Corfload <- CorFact$loadings[, 1:3]
scores4 <- as.matrix(mydata) %*% Corfload
par(mfrow = c(2, 2))
for (i in 1:3) {
    plot(scores4[, i], ylab = paste0("Factor ", i))
    text(scores4[, i], rownames(scores4), cex = 0.4, pos = 4,
        col = "red")
}


## Seeing differences-- Left TO DO
```



Yet again, we can see that COK, PNG and SAM are clear outliers.

** What does it mean that the parameter rotation of factanal() is set to “varimax” by default (equivalently

rotate of principal())?**

The varimax rotation is a method used to simplify the expression of a particular sub-space in terms of just a few major items each one without changing the orthogonal basis but being rotated to align with respect to maximize variance. Varimax is so called because it maximizes the sum of the variances of the squared loadings (squared correlations between variables and factors). Preserving orthogonality requires that it is a rotation that leaves the sub-space invariant. This is achieved if: - Any given variable has a high loading on a single factor but near-zero loadings on the remaining factors
- Any given factor is constituted by only a few variables with very high loadings on this factor while the remaining variables have near-zero loadings on this factor.

Does it make a difference if R, rather than S, is factored?

It does make a difference, and it depends on the data one has. The covariance matrix is used when the variable scales are similar whereas the correlation matrix is used when variables are on different scales. The argument against R is that it is quite a drastic way of standardising your data. The problem with automatically using the covariance matrix is that the variables with the highest variance will dominate the first principal component.

In summary, use the correlation matrix R when within-variable range and scale widely differs, and use the covariance matrix S to preserve variance if the range and scale of variables is similar or in the same units of measure.