Multivariate Statistical Methods 732A97

Lab 3

Albin Västerlund
albva223

Eric Herwin
erihe068

Balaji Ramkumar
balra340

Guilherme Barros
guiba484

Contents

Question 1: Principal components, including interpretation of them

In this task we will do the task 8.19 in the book.

a)

For this task we will compute the sample correlation matrig and determine it's eigenvalues and eigenvectors.

The sample correlation matrix:

	m100	m200	m400	m800	m1500	m3000	Marathon
m100	1.0000000	0.9410886	0.8707802	0.8091758	0.7815510	0.7278784	0.6689597
m200	0.9410886	1.0000000	0.9088096	0.8198258	0.8013282	0.7318546	0.6799537
m400	0.8707802	0.9088096	1.0000000	0.8057904	0.7197996	0.6737991	0.6769384
m800	0.8091758	0.8198258	0.8057904	1.0000000	0.9050509	0.8665732	0.8539900
m1500	0.7815510	0.8013282	0.7197996	0.9050509	1.0000000	0.9733801	0.7905565
m3000	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

The eigen decomposition:

```
## eigen() decomposition
## $values
## [1] 5.80762446 0.62869342 0.27933457 0.12455472 0.09097174 0.05451882
## [7] 0.01430226
##
## $vectors
##                  Z1         Z2         Z3          Z4          Z5
## m100     -0.3777657 -0.4071756 -0.1405803  0.58706293 -0.16706891
## m200     -0.3832103 -0.4136291 -0.1007833  0.19407501  0.09350016
## m400     -0.3680361 -0.4593531  0.2370255 -0.64543118  0.32727328
## m800     -0.3947810  0.1612459  0.1475424 -0.29520804 -0.81905467
## m1500    -0.3892610  0.3090877 -0.4219855 -0.06669044  0.02613100
## m3000    -0.3760945  0.4231899 -0.4060627 -0.08015699  0.35169796
## Marathon -0.3552031  0.3892153  0.7410610  0.32107640  0.24700821
##                  Z6          Z7
## m100      0.53969730  0.08893934
## m200     -0.74493139 -0.26565662
## m400      0.24009405  0.12660435
## m800     -0.01650651 -0.19521315
## m1500    -0.18898771  0.73076817
## m3000     0.24049968 -0.57150644
## Marathon -0.04826992  0.08208401
```

The outout shows 7 eigenvalues and 7 prncipal compnents.

b)

In this task we will standardize the vaiables and determine the first two principal compoents.

	m100	m200	m400	m800	m1500	m3000	Marathon
Z1	-0.3777657	-0.3832103	-0.3680361	-0.3947810	-0.3892610	-0.3760945	-0.3552031

	m100	m200	m400	m800	m1500	m3000	Marathon
Z2	-0.4071756	-0.4136291	-0.4593531	0.1612459	0.3090877	0.4231899	0.3892153

In the output we can see the two first components.

The table with correlations:

	Z1
m100	-0.9103780
m200	-0.9234990
m400	-0.8869307
m800	-0.9513832
m1500	-0.9380805
m3000	-0.9063506
Marathon	-0.8560043

	Z1
m100	-0.3228503
m200	-0.3279673
m400	-0.3642220
m800	0.1278522
m1500	0.2450762
m3000	0.3355481
Marathon	0.3086096

The output shows the correlations of the components.

```
#total standardized variance for the two compnents
sum(eig$values[1:2]/ sum(eig$values))
```

```
## [1] 0.919474
```

In the output we can see that almost 92% of the variance is exmplained by the two first components.

c)

From the principal compnents we can see that the first pricipal compnent produces scores on how good the countries are in running. So if a country is bad (i.e. have high time) in running they will have a high negative score.

The second principal component produces scores that give higher negative scores to the countries that perform bad on low distances, but good on higher distances. The countries that performs bad on longer distances gets a more positive scores.

d)

In this task we will rank the nations based on their score for the first principal component.

This is the 6 nations with the highest negative scores:

	country	score
11	COK	-119.71928
40	PNG	-119.35616
46	SAM	-111.15524
21	GUA	-101.47063
5	BER	-100.82440
34	MRI	-98.98619

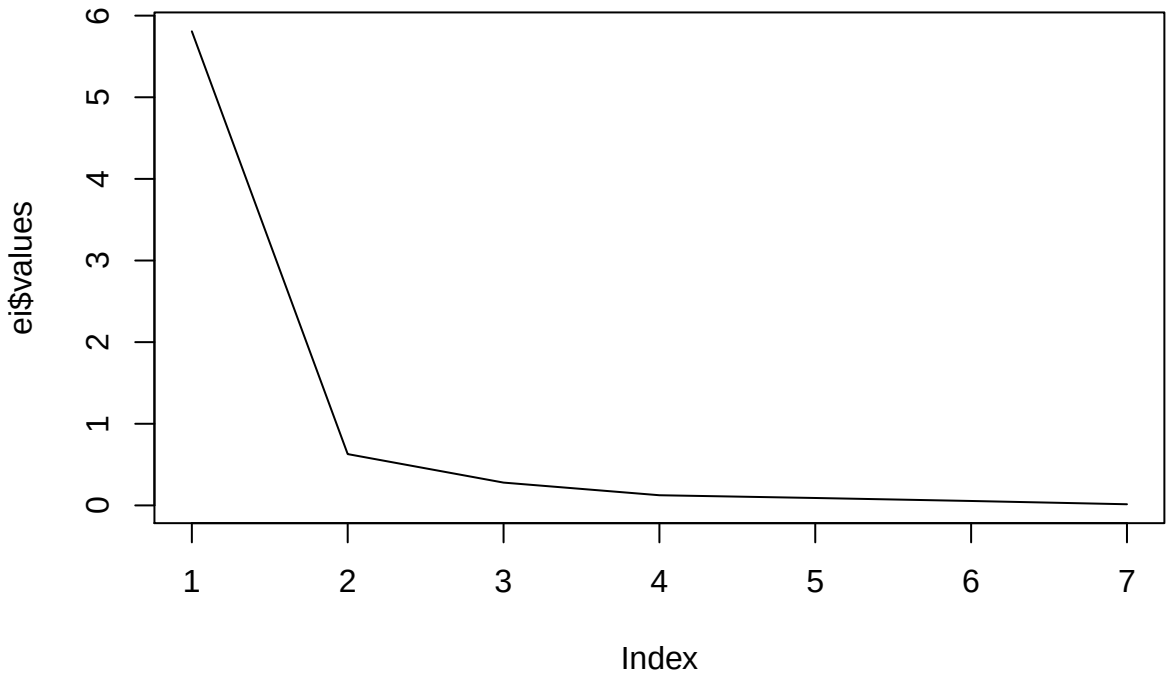
This is the 6 nations with the lowest negative scores:

	country	score
29	KEN	-86.98080
45	RUS	-86.12949
54	USA	-85.72607
18	GER	-85.66515
9	CHN	-85.65724
19	GBR	-84.35454

We can see from the pricipal compnents that is based on the data we have, that COK are the "worst" country in terms of running. From the result we can also say that GBR is the "best" nation in terms of running.

Question 2: Factor analysis

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

## [1] 0.919474
```



As we can see, the first two eigen values are much bigger than the rest and contain 92% of the variance. We will do the rest of the analysis using 2 factors for the PCA.

We now do PCA for two factors without varimax rotation and with it:

```
#From the plot, we interpret the number of factors as 2.

#Using principal component analysis using r.

princi1 <- principal(my_data[2:8],nfactors = 2,rotate = "none") #rotation is none

princi1$loadings # factors
```

```
##
## Loadings:
##          PC1    PC2
## 100m      0.910 -0.323
## 200m      0.923 -0.328
## 400m      0.887 -0.364
## 800m      0.951  0.128
## 1500m     0.938  0.245
## 3000m     0.906  0.336
## Marathon  0.856  0.309
##
##                  PC1   PC2
## SS loadings    5.808 0.629
## Proportion Var 0.830 0.090
## Cumulative Var 0.830 0.919
```

```
princi2 <- principal(my_data[2:8],nfactors = 2,rotate = "varimax")#rotation is varimax

princi2$loadings #factors
```

```
##
```

```
## Loadings:
##          RC1   RC2
## 100m     0.431 0.865
## 200m     0.437 0.877
## 400m     0.385 0.878
## 800m     0.773 0.569
## 1500m    0.845 0.475
## 3000m    0.885 0.388
## Marathon 0.830 0.373
##
##                  RC1   RC2
## SS loadings    3.309 3.128
## Proportion Var 0.473 0.447
## Cumulative Var 0.473 0.919
```

We can see that with the varimax rotation the proportional variance between two the two components is much closer.

For the unrotated components, it is possible to interpret the first factor as the general performance of the country, while the second one is contrasting the performance between short distance and long distance events.

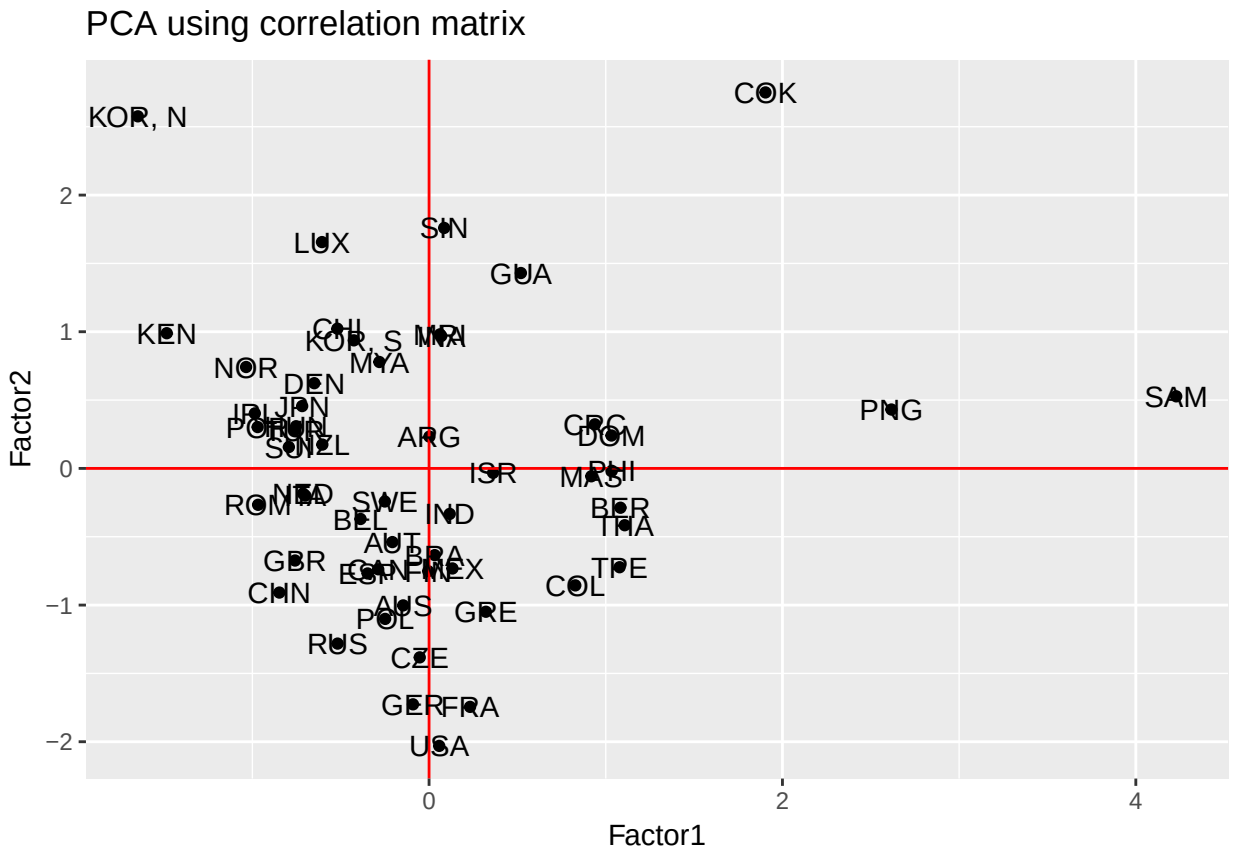
For the rotated components, the first component gives much more weight to long distance events while the second component gives more weight to short distance events.

We will now calculate the factor score coefficients:

```
#For getting factor coefficients

princi2_mat <- matrix(princi2$loadings,ncol = 2)

principal_factor_coeff <-  t(princi2_mat)%*%solve(r)

b <- t(principal_factor_coeff)

colnames(b) <- c("Factor1", "Factor2")

b #factor coeff
```

```
##             Factor1     Factor2
## 100m     -0.2439476  0.47829981
## 200m     -0.2479760  0.48572410
## 400m     -0.2925628  0.52283367
## 800m      0.2591053 -0.03249896
## 1500m     0.3869765 -0.16822033
## 3000m     0.4830071 -0.27553526
## Marathon  0.4470070 -0.25073338
```

```
PCA_Scores <- data.frame(princi2$scores, country=my_data[,1])


ggplot(PCA_Scores, aes(x= princi2$scores[,1], y= princi2$scores[,2])) + geom_point()  +
  geom_vline(xintercept=0, col="red") +
  geom_hline(yintercept=0, col="red") +
  geom_text(aes(label=country))+
  xlab("Factor1")+
  ylab("Factor2")+
  ggtitle("PCA using correlation matrix")
```




```
#from the plot we FIND 4 outliers, KORN,COK,PNG,SAM
```

And we can see that KORN,COK,PNG,SAM can be considered outliers.

Now, we do the same analysis using S:

```
#Using principal component analysis using s.

princi1_s <- principal(my_data[2:8],nfactors = 2,rotate = "none",covar = TRUE) #rotation is none

princi1_s$loadings #Factors
```

```
##
## Loadings:
##          PC1    PC2
## 100m      0.267  0.230
## 200m      0.640  0.582
## 400m      1.785  1.881
## 800m
## 1500m     0.217
## 3000m     0.654  0.158
## Marathon 16.438 -0.238
##
##                   PC1    PC2
## SS loadings    274.363  4.017
## Proportion Var  39.195  0.574
## Cumulative Var  39.195 39.768
```

```r
princi2_s <- principal(my_data[2:8],nfactors = 2,rotate = "varimax",covar = TRUE) #rotation is varimax


princi2_s$loadings #Factors
```

```
##
## Loadings:
##          RC1    RC2
## 100m      0.173  0.307
## 200m      0.404  0.765
## 400m      1.038  2.376
## 800m
## 1500m     0.179  0.142
## 3000m     0.561  0.371
## Marathon 15.537  5.375
##
##                    RC1    RC2
## SS loadings    243.005 35.375
## Proportion Var  34.715  5.054
## Cumulative Var  34.715 39.768
```

We can see that by using S instead of R, the scaling makes a lot of difference for the components. The two most noticiable things happening are how marathon gets a huge weight and how 800m does not matter much. This is also a result of how the data has different scales for the events, some in minutes and others in seconds.

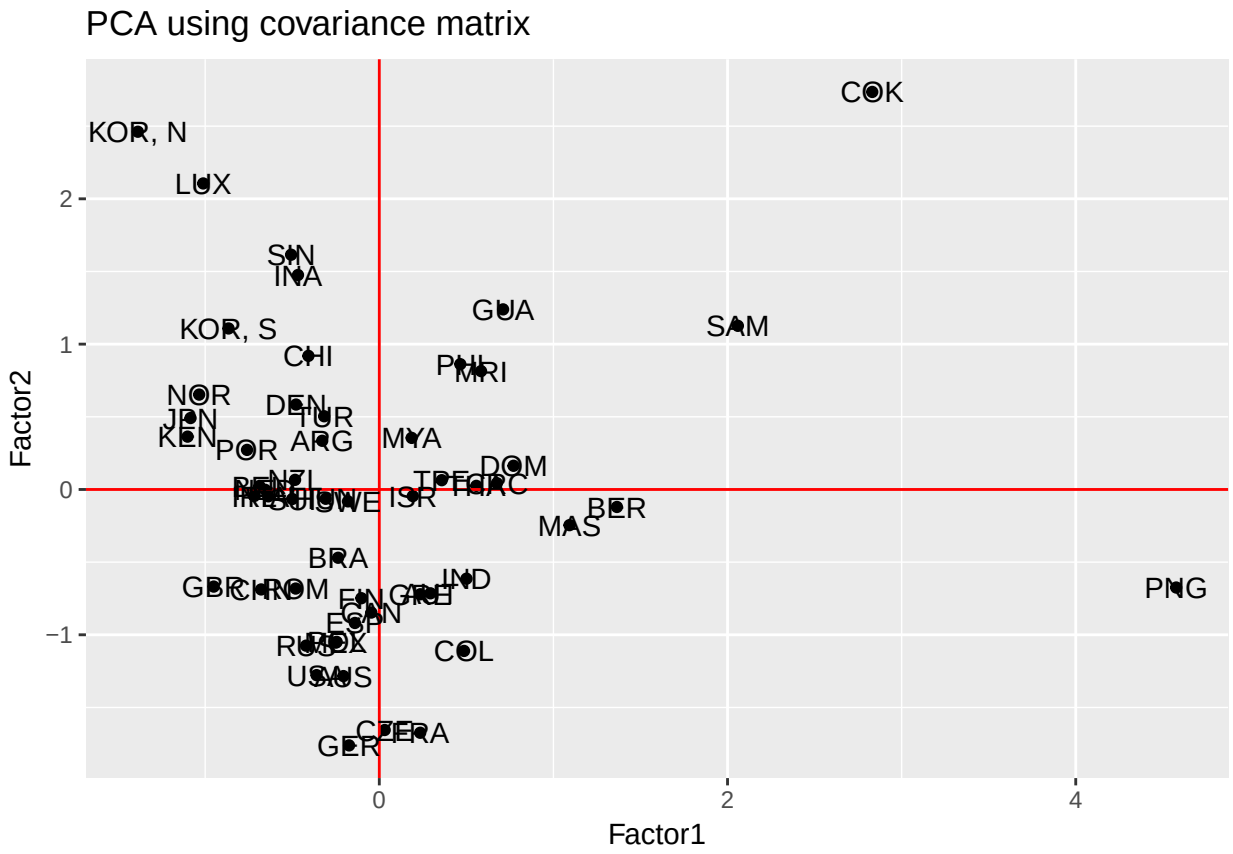
We will now calculate the factor score coefficients:

```r
princi2_s_mat <- matrix(princi2_s$loadings,ncol = 2)


principal_factor_coeff_s <-  t(princi2_s_mat)%*%solve(r)

c <- t(principal_factor_coeff)

colnames(c) <- c("Factor1","Factor2")

c #factor coeff
```

```
##             Factor1     Factor2
## 100m     -0.2439476  0.47829981
## 200m     -0.2479760  0.48572410
## 400m     -0.2925628  0.52283367
## 800m      0.2591053 -0.03249896
## 1500m     0.3869765 -0.16822033
## 3000m     0.4830071 -0.27553526
## Marathon  0.4470070 -0.25073338
```

```r
PCA_s_Scores <- data.frame(princi2$scores, country=my_data[,1])


ggplot(PCA_s_Scores, aes(x= princi2_s$scores[,1], y= princi2_s$scores[,2])) + geom_point()  +
  geom_vline(xintercept=0, col="red") +
  geom_hline(yintercept=0, col="red") +
  geom_text(aes(label=country))+
  xlab("Factor1")+
  ylab("Factor2")+
```

```
  ggtitle("PCA using covariance matrix")
```




```
#We find there are 5 outliers. COK,SAM,PNG,KORN,LUX
```

It is interesting to notice that the final factor analysis for S is the same as for R, which implies that the scaling doesn't matter.

We can see that in this case COK,SAM,PNG,KORN,LUX could be considered outliers.

At last, we now do the factor analysis using the ML method:

```
#### Using factanal[ML method]

#rotation as none. using r
fact1<- factanal(my_data[2:8], factors=2, rotation="none" )

fact1$loadings #factors
```

```
##
## Loadings:
##          Factor1 Factor2
## 100m      0.876   0.372
## 200m      0.898   0.411
## 400m      0.826   0.406
## 800m      0.925
## 1500m     0.974  -0.186
## 3000m     0.945  -0.281
## Marathon  0.809
##
```

```
##                Factor1 Factor2
## SS loadings      5.609   0.594
## Proportion Var   0.801   0.085
## Cumulative Var   0.801   0.886
```

```
#rotation as varimax using r

fact2 <- factanal(my_data[2:8], factors=2, rotation="varimax",scores = "Bartlett" )

fact2$loadings #factors
```

```
##
## Loadings:
##          Factor1 Factor2
## 100m     0.461   0.833
## 200m     0.455   0.877
## 400m     0.401   0.829
## 800m     0.732   0.566
## 1500m    0.882   0.454
## 3000m    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
```

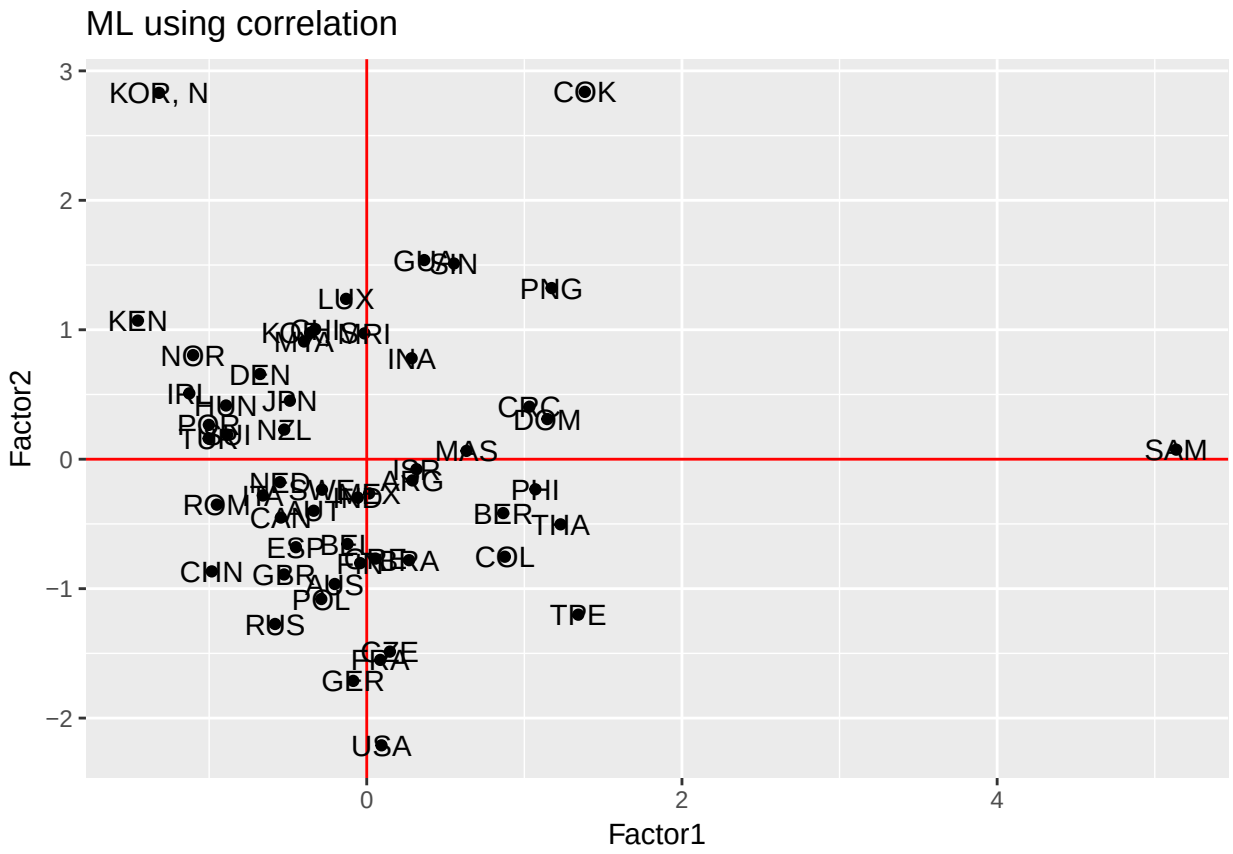
We can see that the results obtained by the ML method resembles the analysis of R done before where the first unrotated factor is the general performance and the second is the difference between long and short distance (although 800m and marathon were considered irrelevant in this case). The rotated factors also follow the same pattern as the analysis of R.

```
#For getting factor coefficients

f2_mat<-matrix(fact2$loadings,ncol=2)

fact2_coeff <- t(f2_mat)%*%solve(r)

a <- t(fact2_coeff)

colnames(a) <- c("Factor1", "Factor2")

a # factor coefficients
```

```
##              Factor1      Factor2
## 100m     -0.10616834  0.239562104
## 200m     -0.47286517  1.020099582
## 400m     -0.07662240  0.158683063
## 800m      0.03630460  0.025299002
## 1500m     0.77645256 -0.325117412
## 3000m     0.58664592 -0.370004275
## Marathon  0.02354883 -0.003408494
```

Is is interesting to notice that the ML method reached a result very similar to the ones found before. Although the final factors are not the same, the first factor has negative values for 100m, 200m and 400m and the last one has a value very close to 0 for 800m and negative values for 1500m and Marathon.

```
MLScores <- data.frame(fact2$scores, country=my_data[,1])


ggplot(MLScores, aes(x=fact2$scores[,1], y= fact2$scores[,2])) + geom_point()  +
  geom_vline(xintercept=0, col="red") +
  geom_hline(yintercept=0, col="red") +
  geom_text(aes(label=country))+
  xlab("Factor1")+
  ylab("Factor2")+
  ggtitle("ML using correlation")
```




```
#We found outliers are KORN,COK,SAM
```

We can see that in this case KORN,COK,SAM can be considered outliers.