Lab3

Naveen (navga709), Sridhar(sriad858), Juan(juado206), Samia(sambu064)

December 8, 2019

Contents

Question 1 : Principal components, including interpretation of them **2**

Read the data . . . 2

A . . . 2

B . . . 3

C . . . 3

D . . . 3

Question 2 **6**

Selecting number of components required . . . 6

PC estimation method . . . 7

Using covariance matrix . . . 7

Using correlation matrix . . . 8

ML estimation method . . . 10

Using covariance matrix . . . 10

Using correlation matrix . . . 12

What does it mean that the parameter rotation of factanal() is set to “varimax” by default? . . . 14

```r
library(psych)
library(psy)
```

Question 1 : Principal components, including interpretation of them

Read the data

```r
dataset = read.table("T1-9.dat", row.names = 1)
head(dataset)
```

```
       V2    V3    V4   V5   V6   V7     V8
ARG 11.57 22.94 52.50 2.05 4.25 9.19 150.32
AUS 11.12 22.23 48.63 1.98 4.02 8.63 143.51
AUT 11.15 22.70 50.62 1.94 4.05 8.78 154.35
BEL 11.14 22.48 51.45 1.97 4.08 8.82 143.05
BER 11.46 23.05 53.30 2.07 4.29 9.81 174.18
BRA 11.17 22.60 50.62 1.97 4.17 9.04 147.41
```

A

```r
##a)

R <- cor(dataset)
eigen <- eigen(R, symmetric = TRUE, only.values = FALSE)
round(eigen$values, 4)
```

```
[1] 5.8076 0.6287 0.2793 0.1246 0.0910 0.0545 0.0143
```

```r
round(eigen$vectors, 3)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
[1,] -0.378 -0.407 -0.141  0.587 -0.167  0.540  0.089
[2,] -0.383 -0.414 -0.101  0.194  0.094 -0.745 -0.266
[3,] -0.368 -0.459  0.237 -0.645  0.327  0.240  0.127
[4,] -0.395  0.161  0.148 -0.295 -0.819 -0.017 -0.195
[5,] -0.389  0.309 -0.422 -0.067  0.026 -0.189  0.731
[6,] -0.376  0.423 -0.406 -0.080  0.352  0.240 -0.572
[7,] -0.355  0.389  0.741  0.321  0.247 -0.048  0.082
```

In this first section, we'll use the eigen command to find the eigenvalues and vectors from the sample correlation matrix. eigen*valuesandeigen*vectors will store the info.

B

The product of the square root of a given eigenvalue with its corresponding eigenvector will return the correlation of the variables with the components. r_table shows the correlations of the standarized variables with the first two components.

```
##b)

r_table <- t(round(matrix(c(sqrt(eigen$values[1])*eigen$vectors[, 1], sqrt(eigen$values[2])*eigen$vecto
rownames(r_table) <- c("r y1, z1", "r y2, z2 ")
r_table
```

```
             [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
r y1, z1  -0.910 -0.923 -0.887 -0.951 -0.938 -0.906 -0.856
r y2, z2  -0.323 -0.328 -0.364  0.128  0.245  0.336  0.309
```

```
values_table <- t(round(matrix(c(eigen$values, integer(length(eigen$values)), integer(length(eigen$valu
rownames(values_table) <- c("Eigenvalues", "Percentage", "Cumulative")
colnames(values_table) <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7")
values_table
```

```
               x1    x2    x3    x4    x5    x6    x7
Eigenvalues 5.808 0.629 0.279 0.125 0.091 0.055 0.014
Percentage  0.000 0.000 0.000 0.000 0.000 0.000 0.000
Cumulative  0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

```
for (i in 1:dim(values_table)[2]) {
  values_table[2, i] <- values_table[1, i] / sum(values_table[1, ])
  values_table[3, i] <- (values_table[1, i] / sum(values_table[1, ])) + sum(values_table[3, i-1])
}
round(values_table, 3)
```

```
               x1    x2    x3    x4    x5    x6    x7
Eigenvalues 5.808 0.629 0.279 0.125 0.091 0.055 0.014
Percentage  0.830 0.090 0.040 0.018 0.013 0.008 0.002
Cumulative  0.830 0.919 0.959 0.977 0.990 0.998 1.000
```

Likewise, this values_table is created to store the requested cumulative percentage associated with each component. Can be seen how the first two already explain 0.919 of the variance, filled in by the for loop.

C

First component measures the overall excellence of a given country while the second one can be used to compare the times in shorter distance with the ones in longer distance.

D

With this final part of the code, a new matrix called score will be created, filled in by the the score of each country ordered in descending order. The results match pretty well with the ranking that a person with basic notions on athletism could do.

```r
##d)

score <- cbind(row.names(dataset), integer(dim(dataset)[1]))

for (i in 1:dim(dataset)[1]) {
  score[i, 2] <- r_table[1, ] %*% t(dataset[i, 1:7])

  j <- i
  while(j>1) {
    if(score[j, 2] > score[j-1, 2]) {
      extra <- score[j-1, ]
      score [j-1, ] <- score[j, ]
      score[j,] <- extra
    }
    j <- j - 1
  }
}
score
```

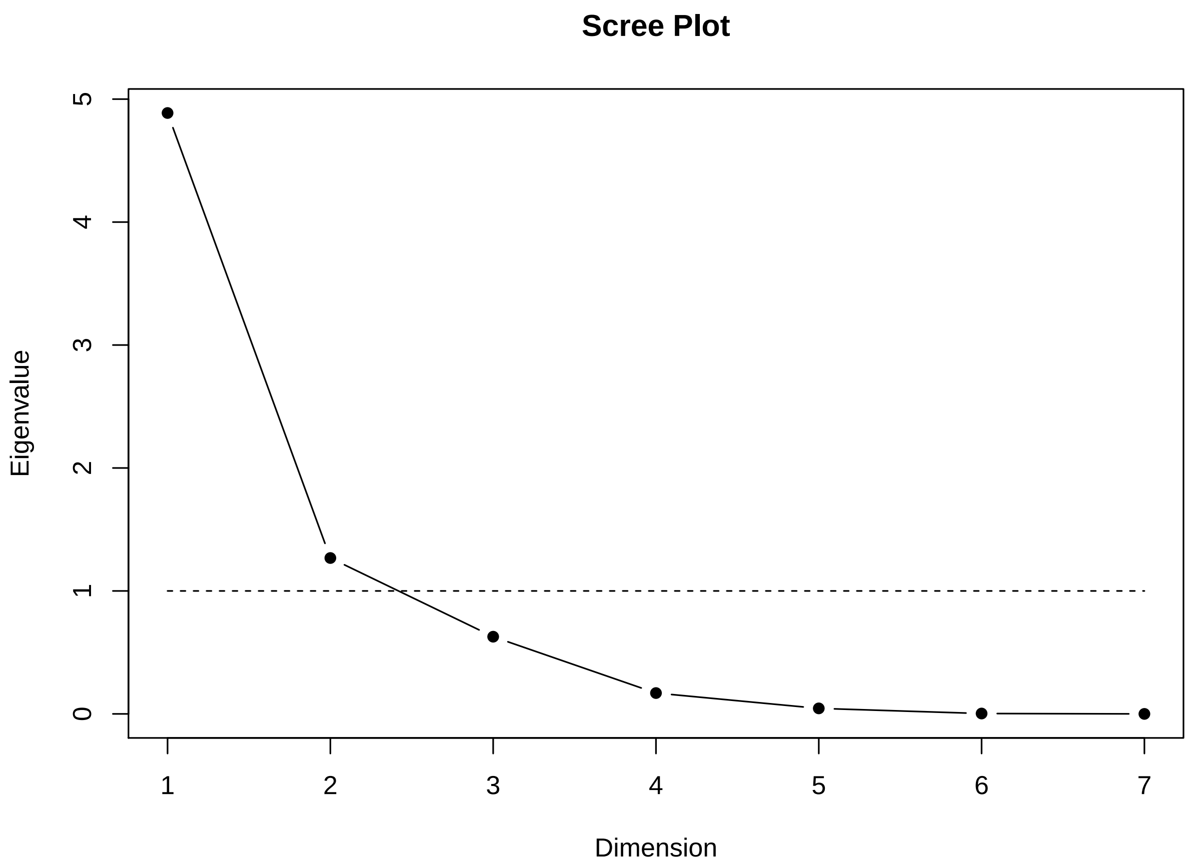
```
      [,1]   [,2]
 [1,] "COK"  "-288.4922"
 [2,] "PNG"  "-287.6178"
 [3,] "SAM"  "-267.85294"
 [4,] "GUA"  "-244.51601"
 [5,] "BER"  "-242.95938"
 [6,] "MRI"  "-238.52931"
 [7,] "MAS"  "-238.12871"
 [8,] "DOM"  "-237.37718"
 [9,] "PHI"  "-237.07154"
[10,] "CRC"  "-235.19133"
[11,] "THA"  "-232.66916"
[12,] "SIN"  "-229.98116"
[13,] "TPE"  "-229.18244"
[14,] "MYA"  "-229.15612"
[15,] "INA"  "-228.16436"
[16,] "ISR"  "-226.207"
[17,] "IND"  "-226.03507"
[18,] "CHI"  "-224.77878"
[19,] "LUX"  "-224.68478"
[20,] "COL"  "-222.82076"
[21,] "KORN" "-222.1444"
[22,] "AUT"  "-221.72066"
[23,] "TUR"  "-221.28361"
[24,] "ARG"  "-221.20593"
[25,] "GRE"  "-220.79301"
[26,] "DEN"  "-220.4054"
[27,] "SWE"  "-219.5121"
[28,] "KORS" "-219.23508"
[29,] "HUN"  "-217.87353"
[30,] "NZL"  "-216.44498"
[31,] "BRA"  "-216.08257"
[32,] "FIN"  "-215.5869"
[33,] "CAN"  "-215.4991"
```

```
[34,] "SUI"  "-214.91348"
[35,] "ESP"  "-213.49941"
[36,] "POR"  "-213.40898"
[37,] "FRA"  "-213.23086"
[38,] "NED"  "-212.86809"
[39,] "ITA"  "-212.69923"
[40,] "BEL"  "-212.66482"
[41,] "NOR"  "-212.58883"
[42,] "MEX"  "-212.02726"
[43,] "IRL"  "-211.91564"
[44,] "JPN"  "-211.00009"
[45,] "CZE"  "-210.83373"
[46,] "POL"  "-210.82698"
[47,] "ROM"  "-210.19389"
[48,] "AUS"  "-210.08938"
[49,] "KEN"  "-209.59804"
[50,] "RUS"  "-207.54739"
[51,] "USA"  "-206.57551"
[52,] "GER"  "-206.42829"
[53,] "CHN"  "-206.40939"
[54,] "GBR"  "-203.26973"
```

Question 2

Selecting number of components required

```
dataset = read.table("T1-9.dat", row.names = 1)
fit = factanal(dataset, 3, rotation="varimax")
scree.plot(fit$correlation)
```




```
fit
```

```
Call:
factanal(x = dataset, factors = 3, rotation = "varimax")

Uniquenesses:
   V2    V3    V4    V5    V6    V7    V8
0.106 0.005 0.133 0.047 0.005 0.041 0.225

Loadings:
   Factor1 Factor2 Factor3
V2 0.815   0.413   0.245
```

```
V3 0.886   0.410   0.203
V4 0.797   0.311   0.367
V5 0.512   0.617   0.556
V6 0.449   0.849   0.270
V7 0.361   0.866   0.280
V8 0.380   0.553   0.571

               Factor1 Factor2 Factor3
SS loadings      2.824   2.593   1.022
Proportion Var   0.403   0.370   0.146
Cumulative Var   0.403   0.774   0.920

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 9.73 on 3 degrees of freedom.
The p-value is 0.021
```

According to this plot, two factors would be good for this data. But the p-value for three factors is the highest. So, we carry out the rest of the exersise with 3 factors.

PC estimation method

Using covariance matrix

```r
R = cor(dataset)
S = cov(dataset)
nFactors = 3

fit.PC1 <- principal(dataset, nfactors=nFactors, rotate="varimax", covar = S, n.obs = dim(dataset)[1])
# print results
fit.PC1$loadings
```

```
Loadings:
   RC1    RC3    RC2
V2  0.105  0.193  0.290
V3  0.245  0.450  0.726
V4  0.840  0.640  2.371
V5
V6  0.103  0.216  0.119
V7  0.330  0.663  0.300
V8 13.988  6.730  5.415

                   RC1    RC3    RC2
SS loadings    196.554 46.427 35.666
Proportion Var  28.079  6.632  5.095
Cumulative Var  28.079 34.712 39.807
```

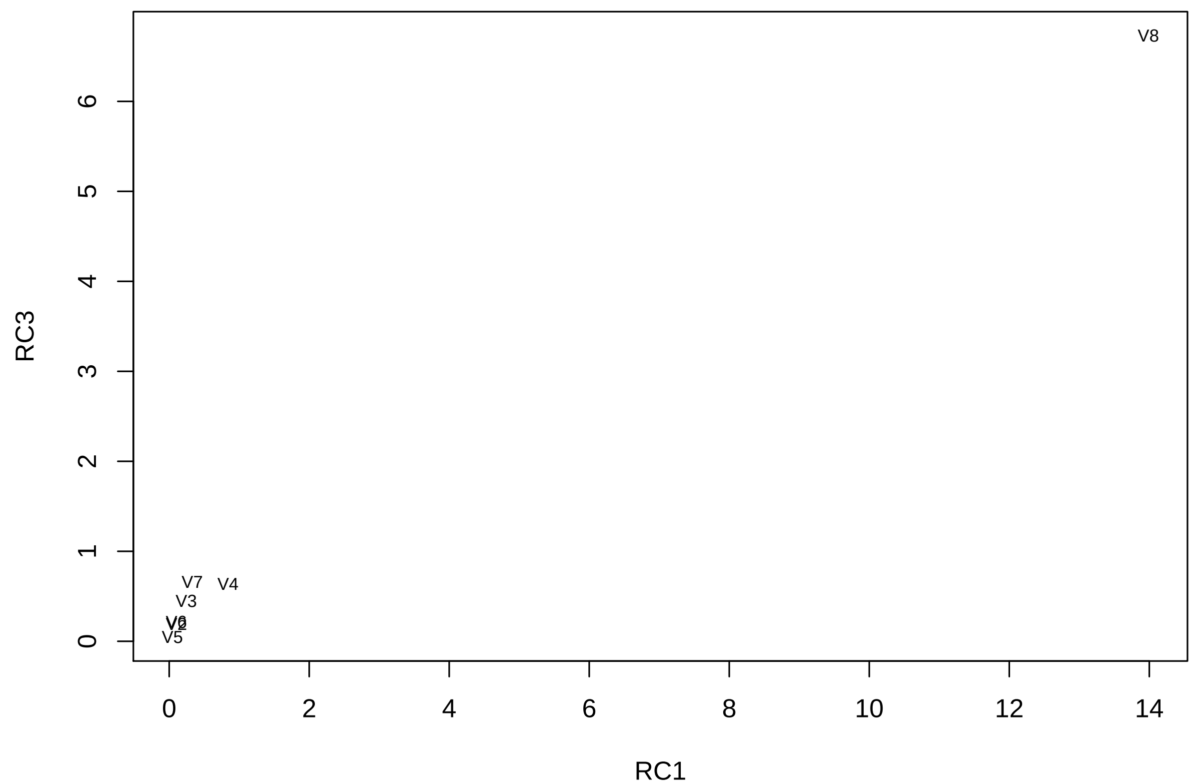
```r
cat("\nUniqueness:\n")
```

```
Uniqueness:
```

```r
fit.PC1$uniquenesses
```

```
          V2           V3           V4           V5           V6 
2.293009e-02 7.423673e-02 6.494369e-03 9.098223e-04 2.712644e-03 
          V7           V8 
2.558714e-02 1.110157e-05 
```

```r
# plot pc 1 by pc 2
load <- fit.PC1$loadings[,1:2]
plot(load,type="n") # set up plot
text(load,labels=names(dataset),cex=.7) # add variable names
```



Using correlation matrix

```r
fit.PC2 <- principal(dataset, nfactors=nFactors, rotate="varimax", covar = R, n.obs = dim(dataset)[1])
# print results
fit.PC2$loadings
```

```
Loadings:
```

```
   RC1    RC3    RC2
V2  0.105  0.193  0.290
V3  0.245  0.450  0.726
V4  0.840  0.640  2.371
V5
V6  0.103  0.216  0.119
V7  0.330  0.663  0.300
V8 13.988  6.730  5.415

                   RC1    RC3    RC2
SS loadings    196.554 46.427 35.666
Proportion Var  28.079  6.632  5.095
Cumulative Var  28.079 34.712 39.807
```

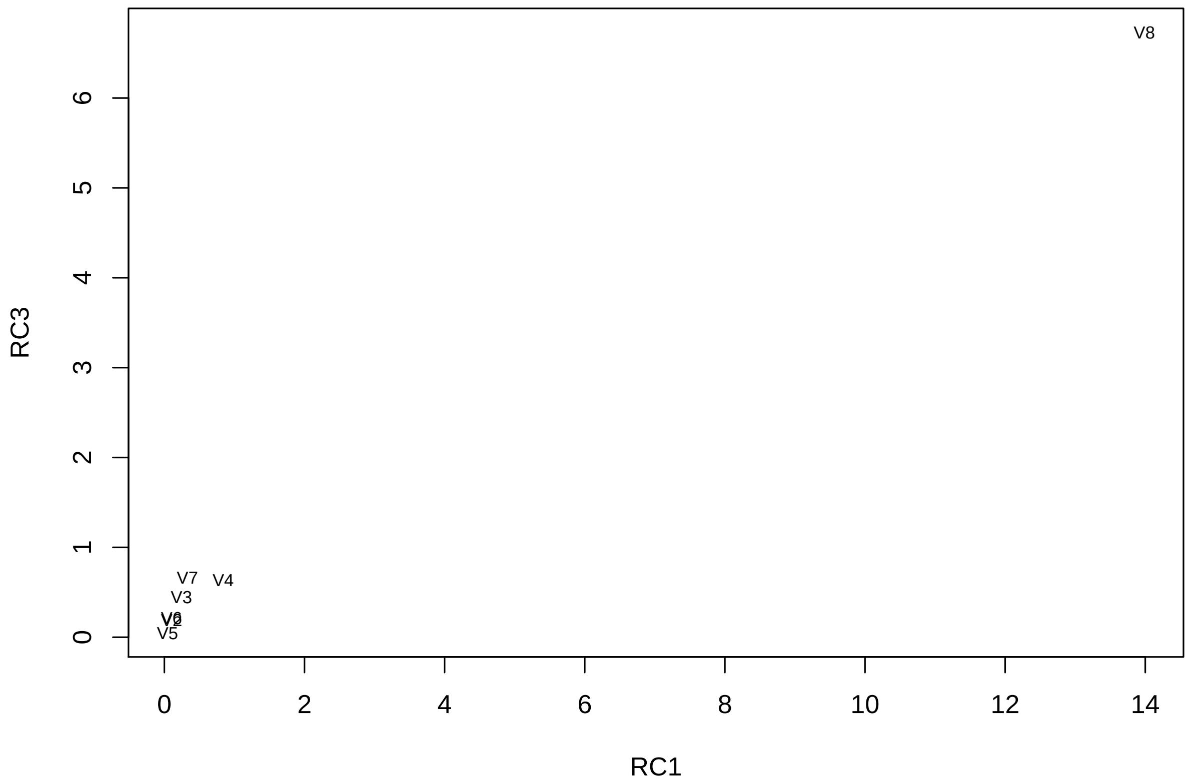
```
cat("\nUniqueness:\n")
```

```
Uniqueness:
```

```
fit.PC2$uniquenesses
```

```
          V2           V3           V4           V5           V6
2.293009e-02 7.423673e-02 6.494369e-03 9.098223e-04 2.712644e-03
          V7           V8
2.558714e-02 1.110157e-05
```

```
# plot pc 1 by pc 2
load <- fit.PC2$loadings[,1:2]
plot(load,type="n") # set up plot
text(load,labels=names(dataset),cex=.7) # add variable names
```



The results dont change if we use the correlation matrix or the covariance matrix to calculate the principal components.

ML estimation method

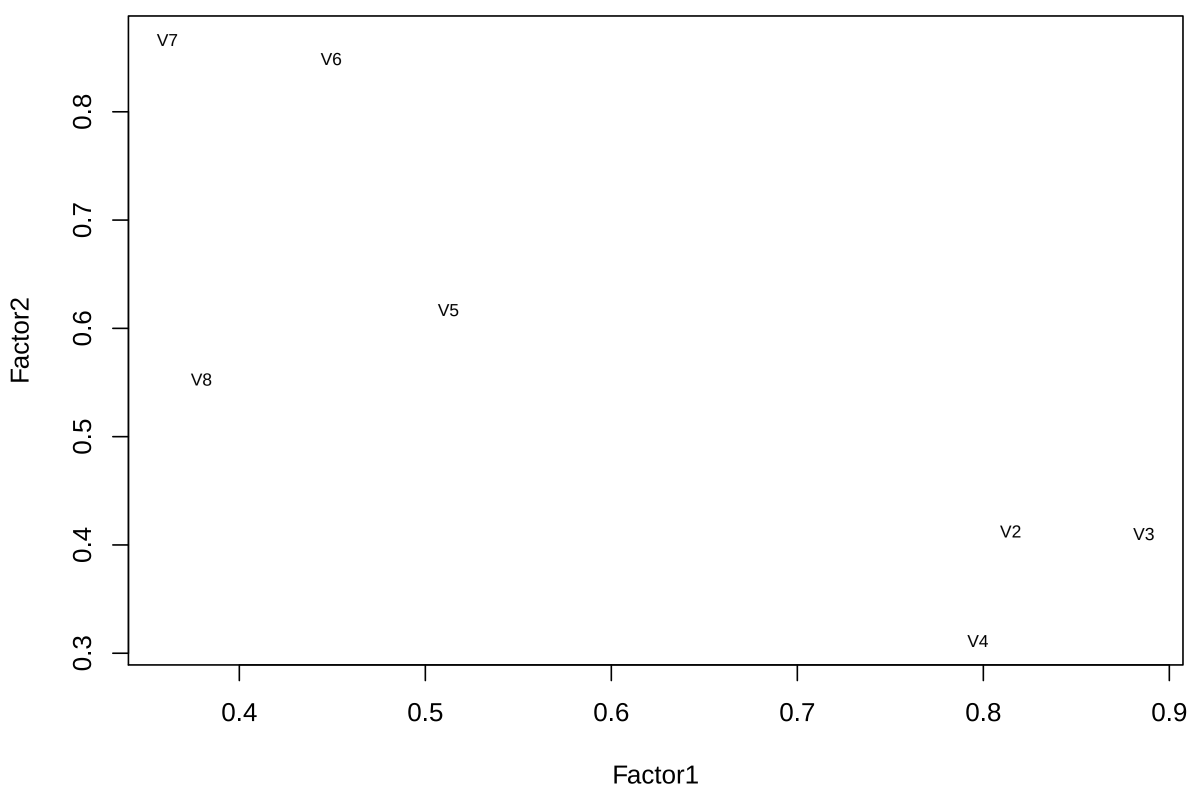
Using covariance matrix

```
fit1 <- factanal(dataset, nFactors, covmat = S, n.obs = dim(dataset)[1],  rotation="varimax")
fit1 # print results
```

```
Call:
factanal(x = dataset, factors = nFactors, covmat = S, n.obs = dim(dataset)[1],     rotation = "varimax")

Uniquenesses:
   V2    V3    V4    V5    V6    V7    V8 
0.106 0.005 0.133 0.047 0.005 0.041 0.225 

Loadings:
   Factor1 Factor2 Factor3
V2 0.815   0.413   0.245  
V3 0.886   0.410   0.203  
```

```
V4 0.797   0.311   0.367
V5 0.512   0.617   0.556
V6 0.449   0.849   0.270
V7 0.361   0.866   0.280
V8 0.380   0.553   0.571

               Factor1 Factor2 Factor3
SS loadings      2.824   2.593   1.022
Proportion Var   0.403   0.370   0.146
Cumulative Var   0.403   0.774   0.920

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 9.73 on 3 degrees of freedom.
The p-value is 0.021
```

```
# plot factor 1 by factor 2
load <- fit1$loadings[,1:2]
plot(load,type="n") # set up plot
text(load,labels=names(dataset),cex=.7) # add variable names
```




```
#calculate factor scores
fit1.scores = cor(t(dataset), fit1$loadings)
cat("\nFitst six rows of the Factor Scores:\n")
```

Fitst six rows of the Factor Scores:

```r
head(fit1.scores)
```

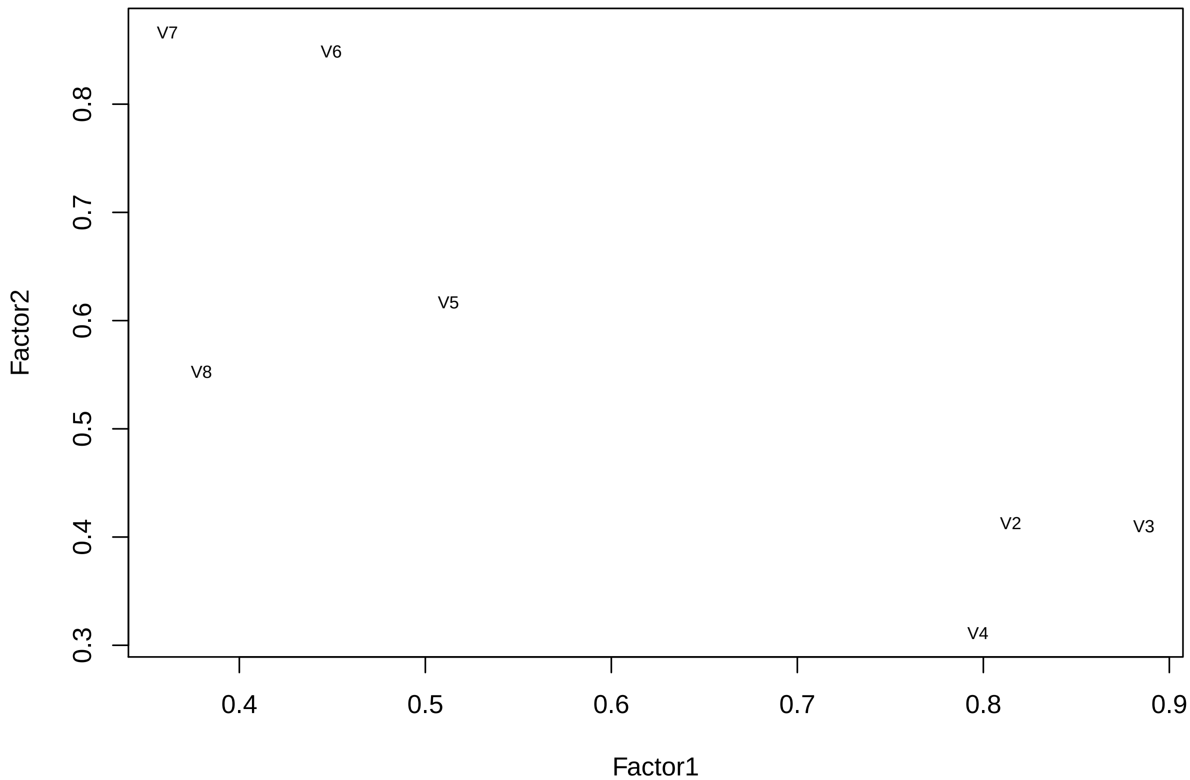
```
       Factor1    Factor2   Factor3
ARG -0.2338515 -0.2686512 0.5715722
AUS -0.2363472 -0.2649257 0.5712166
AUT -0.2449554 -0.2567551 0.5756043
BEL -0.2266491 -0.2758291 0.5689964
BER -0.2644377 -0.2378668 0.5834183
BRA -0.2365797 -0.2652189 0.5715730
```

Using correlation matrix

```r
fit2 <- factanal(dataset, nFactors, covmat = R, n.obs = dim(dataset)[1], rotation="varimax")
fit2 # print results
```

```
Call:
factanal(x = dataset, factors = nFactors, covmat = R, n.obs = dim(dataset)[1],     rotation = "varimax")

Uniquenesses:
   V2    V3    V4    V5    V6    V7    V8
0.106 0.005 0.133 0.047 0.005 0.041 0.225

Loadings:
   Factor1 Factor2 Factor3
V2 0.815   0.413   0.245
V3 0.886   0.410   0.203
V4 0.797   0.311   0.367
V5 0.512   0.617   0.556
V6 0.449   0.849   0.270
V7 0.361   0.866   0.280
V8 0.380   0.553   0.571

               Factor1 Factor2 Factor3
SS loadings      2.824   2.593   1.022
Proportion Var   0.403   0.370   0.146
Cumulative Var   0.403   0.774   0.920

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 9.73 on 3 degrees of freedom.
The p-value is 0.021
```

```r
# plot factor 1 by factor 2
load <- fit2$loadings[,1:2]
plot(load,type="n") # set up plot
text(load,labels=names(dataset),cex=.7) # add variable names
```




```r
#calculate factor scores
fit2.scores = cor(t(dataset), fit2$loadings)
cat("\nFitst six rows of the Factor Scores:\n")
```

```
Fitst six rows of the Factor Scores:
```

```r
head(fit2.scores)
```

```
       Factor1    Factor2   Factor3
ARG -0.2338515 -0.2686512 0.5715722
AUS -0.2363472 -0.2649257 0.5712166
AUT -0.2449554 -0.2567551 0.5756043
BEL -0.2266491 -0.2758291 0.5689964
BER -0.2644377 -0.2378668 0.5834183
BRA -0.2365797 -0.2652189 0.5715730
```

It does not make a difference if a correlation matrix is used instead of the covariance matrix to create the factors. We get the same results in both the cases.

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

Rotation serves to make the output more understandable. Varimax rotation is an orthogonal rotation of the factor axes to maximize the variance of the squared loadings of a factor (column) on all the variables (rows) in a factor matrix, which has the effect of differentiating the original variables by extracted factor. Each factor will tend to have either large or small loadings of any particular variable. A varimax solution yields results which make it as easy as possible to identify each variable with a single factor. This is the most common rotation option, this is the reason it is set to that rotation by default