

Factorial analysis - Practice 2

José Luis Romero Béjar

2023-11-19

© This material is licensed under a **Creative Commons CC BY-NC-ND** attribution which allows *works to be downloaded and shared with others, as long as they are referenced, but may not be modified in any way or used commercially.*

In this practice, from among 25 items of a personality test, the variables that correspond to each of the five aspects of the personality of an individual will be identified. The five characteristics that define the personality of an individual are: A - Agreeableness or friendliness; C - Consciousness or responsibility; E - Extraversion; N - Neuroticism and - Openness to experiences. To do this, the *bfi* data set from the *psych* library is considered. This data set contains 2800 observations with 28 variables, of which 25 correspond to the different items of a personality test (MBI: Maslach Burnout Inventory).

To carry out this practice you must download the following files available on the PRADO platform of the course:

- *FA_2_eng.Rmd*

This brief guide is intended to familiarize the reader with the following:

- Perform a prior exploratory analysis of the data to identify possible missing data and extreme values.
- Make decisions and deal with missing data and extreme values.
- Check the assumptions and perform a factorial analysis (FA).
- Choosing the optimal number of factors.
- Interpretation of different graphical outputs of interest for this method.
- R language: functions debugging.

Loading packages and data set

Loading and installing R packages

The following source code module is responsible for loading, if they are already installed, all the packages that will be used in this R session. While an R package can be loaded at any time when it is to be used, it is advisable to optimize its calls with this code chunk at the beginning.

Loading a package into an R session **requires it to be already installed**. If it is not, the first step is to run the sentence:

```
install.packages("name_of_the_library")
```

```
#####  
# Loading necessary packages and reason #  
#####  
  
# This is an example of the first installation of a package  
# Only runs once if the package is not installed  
# Once it is installed this sentence has to be commented (not to run again)  
# install.packages("summarytools")
```

```

# Package required to call 'factanal' function
library(stats)

# Package required to call 'freq' function
library(summarytools)

# Package required to call 'cortest.bartlett' function
library(psych)

# Package required to call 'hetcor' function
library(polycor)

# Package required to call 'ggcorrplot' function
library(ggcorrplot)

# Package required to call 'corrplot' function
library(corrplot)

# Package required to call 'rplot' function
library(corr)

```

Description of the data set *bfi*

The following code chunk shows how to load the dataset *bfi* included in the *psych* package.

bfi data set

```

# bfi is a data.frame in the package psych
# Remember that package 'psych' is required
# For this example, only the 25 first variables (related to personality) are considered
bfi_s <- bfi[,1:25]

```

The data.frame *bfi_s* contains the variables *A1* to *A5*, *C1* to *C5*, *E1* to *E5*, *N1* to *N5* and *O1* to *O5*, which are the responses of each individual to the 25 items of the MBI questionnaire.

Basic descriptive statistics

In this section, a preliminary exploratory data analysis of the data set is performed. For this purpose, if the variable is **quantitative**, the basic **numerical descriptive statistics** and a representation of its **histogram**, **density** and **boxplot** are shown. On the other hand, for the **categorical** variables their **frequency table** and a **sector and bar diagram** are provided.

In this occasion this is a voluntary assignment to the reader, who can replicate the source code of *Practice_PCA_1.1* or *Practice_PCA_1.2* devoted to this task.

Exploring the data set

```

# This line loads the variable names from this data.frame
# So that we can access by their name with no refer to the data.frame identifier
attach(bfi_s)

# Retrieving the name of all variables
colnames(bfi_s)

```

```
## [1] "A1" "A2" "A3" "A4" "A5" "C1" "C2" "C3" "C4" "C5" "E1" "E2" "E3" "E4" "E5"
## [16] "N1" "N2" "N3" "N4" "N5" "O1" "O2" "O3" "O4" "O5"
```

```
# Displaying a few records
```

```
head(bfi_s, n=5)
```

```
##      A1 A2 A3 A4 A5 C1 C2 C3 C4 C5 E1 E2 E3 E4 E5 N1 N2 N3 N4 N5 O1 O2 O3 O4
## 61617 2 4 3 4 4 2 3 3 4 4 3 3 3 4 4 3 4 2 2 3 3 6 3 4
## 61618 2 4 5 2 5 5 4 4 3 4 1 1 6 4 3 3 3 3 5 5 4 2 4 3
## 61620 5 4 5 4 4 4 5 4 2 5 2 4 4 4 5 4 5 4 2 3 4 2 5 5
## 61621 4 4 6 5 5 4 4 3 5 5 5 3 4 4 4 2 5 2 4 1 3 3 4 3
## 61622 2 3 3 4 5 4 4 5 3 2 2 2 5 4 5 2 3 4 4 3 3 3 4 3
##      O5
## 61617 3
## 61618 3
## 61620 2
## 61621 5
## 61622 3
```

```
# Displaying frequencies table for all the variables A1 to A5
```

```
freq(A1)
```

```
## Frequencies
## A1
## Type: Integer
##
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	922	33.12	33.12	32.93	32.93
2	818	29.38	62.50	29.21	62.14
3	402	14.44	76.94	14.36	76.50
4	337	12.10	89.04	12.04	88.54
5	223	8.01	97.05	7.96	96.50
6	82	2.95	100.00	2.93	99.43
<NA>	16			0.57	100.00
Total	2800	100.00	100.00	100.00	100.00

```
freq(A2)
```

```
## Frequencies
## A2
## Type: Integer
##
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	47	1.69	1.69	1.68	1.68
2	126	4.54	6.24	4.50	6.18
3	151	5.45	11.68	5.39	11.57
4	553	19.94	31.63	19.75	31.32
5	1023	36.89	68.52	36.54	67.86
6	873	31.48	100.00	31.18	99.04
<NA>	27			0.96	100.00
Total	2800	100.00	100.00	100.00	100.00

```
freq(A3)
```

```
## Frequencies
```

```
## A3
## Type: Integer
##
##          Freq  % Valid  % Valid Cum.  % Total  % Total Cum.
## -----
##          1    90     3.24      3.24      3.21      3.21
##          2   172     6.20      9.44      6.14      9.36
##          3   207     7.46     16.91     7.39     16.75
##          4   564    20.33     37.24    20.14     36.89
##          5   986    35.54     72.78    35.21     72.11
##          6   755    27.22    100.00    26.96     99.07
##         <NA>    26          0.93     100.00
##         Total 2800   100.00   100.00   100.00   100.00
```

freq(A4)

```
## Frequencies
## A4
## Type: Integer
##
##          Freq  % Valid  % Valid Cum.  % Total  % Total Cum.
## -----
##          1   129     4.64      4.64      4.61      4.61
##          2   215     7.73     12.37     7.68     12.29
##          3   185     6.65     19.02     6.61     18.89
##          4   451    16.22     35.24    16.11     35.00
##          5   654    23.52     58.76    23.36     58.36
##          6  1147    41.24    100.00    40.96     99.32
##         <NA>    19          0.68     100.00
##         Total 2800   100.00   100.00   100.00   100.00
```

freq(A5)

```
## Frequencies
## A5
## Type: Integer
##
##          Freq  % Valid  % Valid Cum.  % Total  % Total Cum.
## -----
##          1    59     2.12      2.12      2.11      2.11
##          2   186     6.68      8.80      6.64      8.75
##          3   254     9.12     17.92     9.07     17.82
##          4   617    22.16     40.09    22.04     39.86
##          5   973    34.95     75.04    34.75     74.61
##          6   695    24.96    100.00    24.82     99.43
##         <NA>    16          0.57     100.00
##         Total 2800   100.00   100.00   100.00   100.00
```

Descriptive analysis (numerical and graphical)

Variables A1 to A5

Assignment to the reader.

Variables C1 to C5

Assignment to the reader.

Variables E1 to E5

Assignment to the reader.

Variables N1 to N5

Assignment to the reader.

Variables O1 to O5

Assignment to the reader.

Not available data (NA)

Identification and treatment

The decision for not available data is to replace them by the median of their variable. This decision has been made assuming that the behavior of the *NA* is totally random (this would have to be analyzed in depth to confirm this decision made). Perhaps it is not the best option, it depends on the problem under analysis and the data recorded, but it is a way to introduce the reader to **how to define functions in R language**.

The following source code defines the function *not_available* whose utility is to deal with not available data.

```
# Construction of the function that handles missing values.
```

```
not_available<-function(data,na.rm=F){  
  data[is.na(data)]<-median(data,na.rm=T)  
  data  
}  
bfi_s<-as.data.frame(apply(bfi_s,2,not_available))
```

```
# Viewing the data again
```

```
head(bfi_s,n=3)
```

```
##      A1 A2 A3 A4 A5 C1 C2 C3 C4 C5 E1 E2 E3 E4 E5 N1 N2 N3 N4 N5 O1 O2 O3 O4  
## 61617 2 4 3 4 4 2 3 3 4 4 3 3 3 4 4 3 4 2 2 3 3 6 3 4  
## 61618 2 4 5 2 5 5 4 4 3 4 1 1 6 4 3 3 3 3 5 5 4 2 4 3  
## 61620 5 4 5 4 4 4 5 4 2 5 2 4 4 4 5 4 5 4 2 3 4 2 5 5  
##      05  
## 61617 3  
## 61618 3  
## 61620 2
```

Factorial analysis

Requirements

Correlated variables

According to the numerical results below, it is observed that the data **are correlated** both **at the sample level** (see correlation matrix) and **at the populacion level** (Bartlett's sphericity test is significant).

```
#####
# Correlation at sample level #
#####

# Are the variables correlated at sample level?
correlation_matrix<-cor(bfi_s)
# Since this matrix is a 25x25 matrix it is only displayed de first 6x6 places
correlation_matrix[1:6,1:6]
```

```
##           A1           A2           A3           A4           A5           C1
## A1  1.0000000 -0.33739524 -0.26307391 -0.1460951 -0.1799651  0.02745450
## A2 -0.3373952  1.00000000  0.48289584  0.3342597  0.3875393  0.09141116
## A3 -0.2630739  0.48289584  1.00000000  0.3601527  0.5024392  0.09736768
## A4 -0.1460951  0.33425971  0.36015273  1.0000000  0.3074039  0.08761990
## A5 -0.1799651  0.38753934  0.50243919  0.3074039  1.0000000  0.11900536
## C1  0.0274545  0.09141116  0.09736768  0.0876199  0.1190054  1.00000000
```

```
det(correlation_matrix)
```

```
## [1] 0.0008029071
```

```
#####
# Correlation at population level #
#####

# Bartlett's sphericity test:
# This test checks whether the correlations are significantly different from 0
# The null hypothesis is  $H_0$ ;  $\det(R)=1$  means the variables are uncorrelated
# R denotes the correlation matrix
# cortest.bartlett function in the package psych performs this test
# This function works with standardized data.

# Standardization
bfi_scale<-scale(bfi_s)

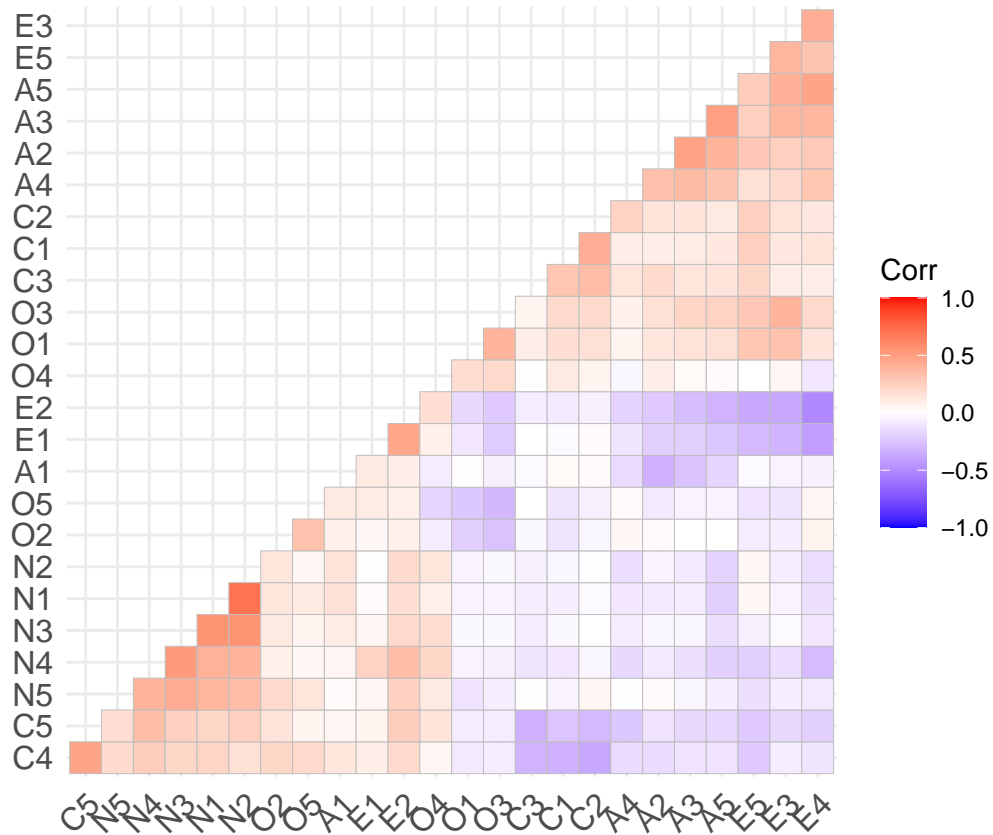
# Bartlett's sphericity test
cortest.bartlett(cor(bfi_scale))
```

```
## $chisq
## [1] 640.2666
##
## $p.value
## [1] 8.937414e-27
##
## $df
## [1] 300
```

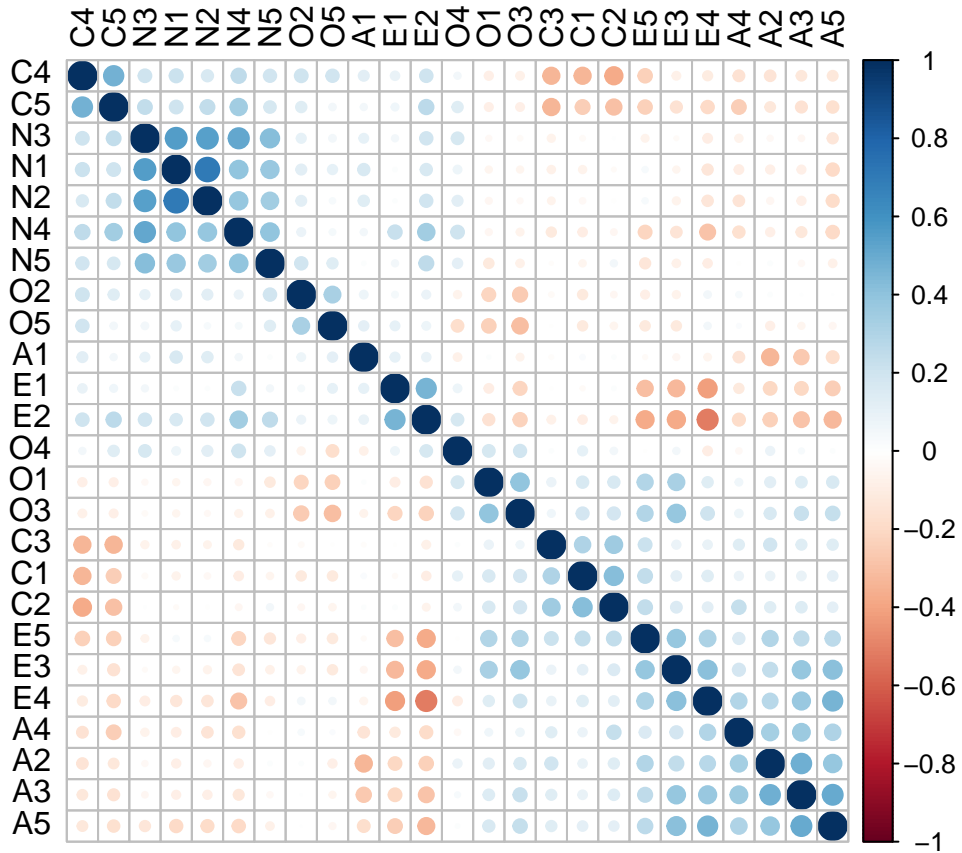
Different graphical outputs are illustrated below that provide an intuitive idea of the correlation between the variables. A *trained eye* could anticipate the appropriate number of factors with this visual information.

```
# Polychoric correlation matrix
# Remember that package 'polycor' is required for 'hetcor' function
poly_cor<-hetcor(bfi_s)$correlations

# Remember that package 'ggcorrplot' is required for 'ggcorrplot' function
ggcorrplot(poly_cor, type="lower",hc.order=T)
```



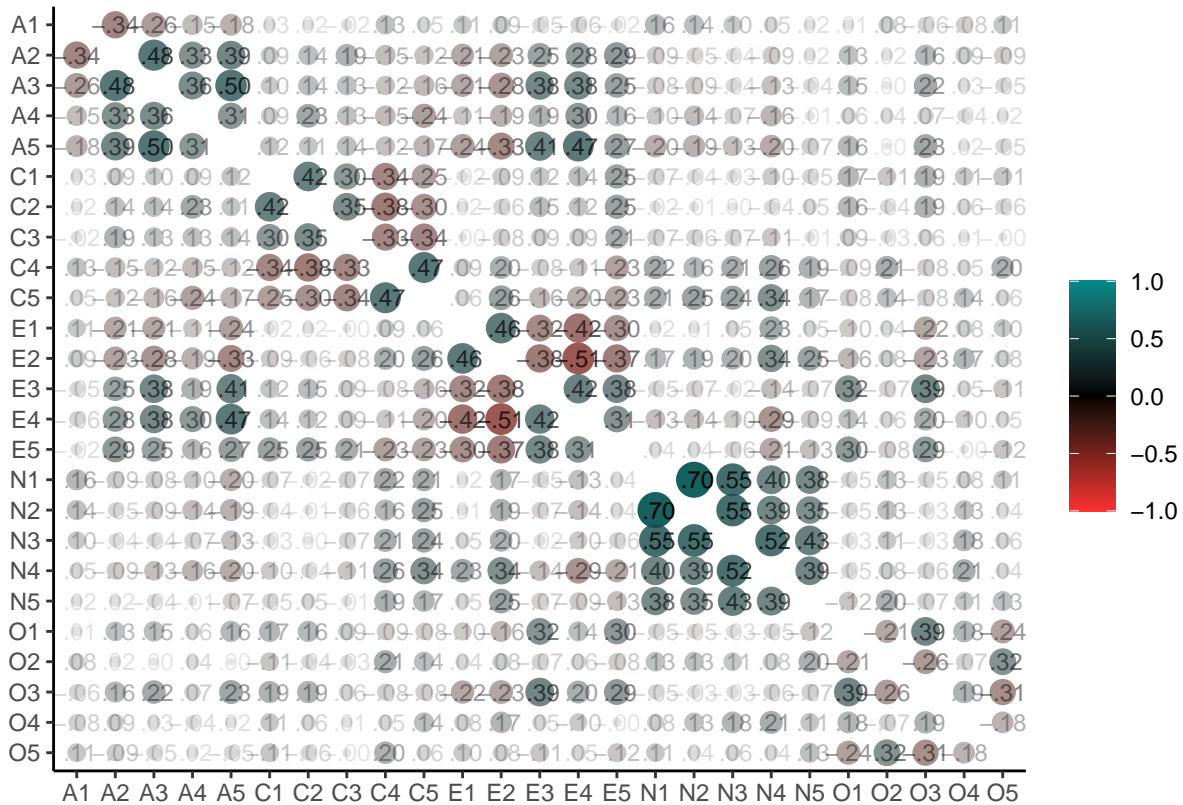
```
# Another interesting visual representation is the following
# Remember that package 'corrplot' is required for 'corrplot' function
corrplot(cor(bfi_s), order = "hclust", tl.col='black', tl.cex=1)
```



```

# Or this other option is also very visual
# Remember that package 'corrr' is required for 'rplot' function
bfi_s_correlaciones <- correlate(bfi_s) #C?lculo de un objeto de correlaciones
rplot(bfi_s_correlaciones, legend = TRUE, colours = c("firebrick1", "black", "darkcyan"),
      print_cor = TRUE)

```

Factorial analysis

Obtaining

You must choose a method to extract the factors: main factor, likelihood, etc. The `fa()` function implements up to 6 different methods. This example compares the outputs of two methods.

First, two models with three factors are performed. The factorial matrix for three latent factors for the two models is displayed.

```
### Test of two models with three factors
modelo1<-fa(poly_cor,
            nfactors = 3,
            rotate = "none",
            fm="mle") # likelihood method

modelo2<-fa(poly_cor,
            nfactors = 3,
            rotate = "none",
            fm="minres") # minimal residual model

# Outputs of these models: factorial matrices, etc.

print("Model 1: mle")

## [1] "Model 1: mle"
```

```
modelo1$loadings
```

```
##  
## Loadings:  
##   ML1   ML2   ML3  
## A1  0.219  
## A2 -0.373  0.339  
## A3 -0.438  0.379 -0.179  
## A4 -0.360  0.207  
## A5 -0.526  0.302 -0.225  
## C1 -0.279  0.201  0.456  
## C2 -0.268  0.272  0.522  
## C3 -0.279  0.156  0.416  
## C4  0.450         -0.492  
## C5  0.493         -0.348  
## E1  0.343 -0.304  0.276  
## E2  0.561 -0.229  0.219  
## E3 -0.439  0.429 -0.179  
## E4 -0.528  0.346 -0.280  
## E5 -0.407  0.426  
## N1  0.589  0.531  
## N2  0.580  0.529  
## N3  0.541  0.486  
## N4  0.584  0.212  
## N5  0.417  0.300  
## O1 -0.259  0.235  
## O2  0.188         -0.175  
## O3 -0.309  0.309  
## O4  0.116  0.166  
## O5  0.171         -0.134  
##  
##               ML1   ML2   ML3  
## SS loadings   4.255 2.280 1.430  
## Proportion Var 0.170 0.091 0.057  
## Cumulative Var 0.170 0.261 0.319
```

```
print("Model 2: minres")
```

```
## [1] "Model 2: minres"
```

```
modelo2$loadings
```

```
##  
## Loadings:  
##   MR1   MR2   MR3  
## A1 -0.218  
## A2  0.444  0.272 -0.116  
## A3  0.516  0.293 -0.202  
## A4  0.389  0.121  
## A5  0.572  0.199 -0.239  
## C1  0.324  0.122  0.461  
## C2  0.328  0.196  0.499  
## C3  0.308         0.370  
## C4 -0.461  0.110 -0.455  
## C5 -0.487  0.138 -0.295
```

```

## E1 -0.399 -0.188  0.274
## E2 -0.597         0.240
## E3  0.525  0.330 -0.151
## E4  0.580  0.201 -0.313
## E5  0.505  0.286
## N1 -0.433  0.598
## N2 -0.423  0.599
## N3 -0.416  0.622
## N4 -0.522  0.392
## N5 -0.341  0.411
## O1  0.317  0.193  0.138
## O2 -0.190         -0.219
## O3  0.384  0.257
## O4         0.239  0.154
## O5 -0.199         -0.185
##
##                MR1   MR2   MR3
## SS loadings    4.384 2.167 1.426
## Proportion Var 0.175 0.087 0.057
## Cumulative Var 0.175 0.262 0.319

```

Finally, a comparison of the communalities of these two methods is illustrated. It appears that communalities of the likelihood model (first column) are greater than those of the minimum residual model (second column).

```

# Comparing communalities
sort(modelo1$communality,decreasing = T)->c1
sort(modelo2$communality,decreasing = T)->c2
head(cbind(c1,c2))

```

```

##          c1          c2
## N1 0.6293538 0.5630216
## N2 0.6214922 0.5465918
## N3 0.5291443 0.5444486
## E4 0.4763169 0.4739129
## C4 0.4461309 0.4334983
## C2 0.4186651 0.4314669

```

```

# Comparison of uniqueness, that is, the proportion of variance that has not
# been explained by the factor (1-communality)
sort(modelo1$uniquenesses,decreasing = T)->u1
sort(modelo2$uniquenesses,decreasing = T)->u2
head(cbind(u1,u2),n=10)

```

```

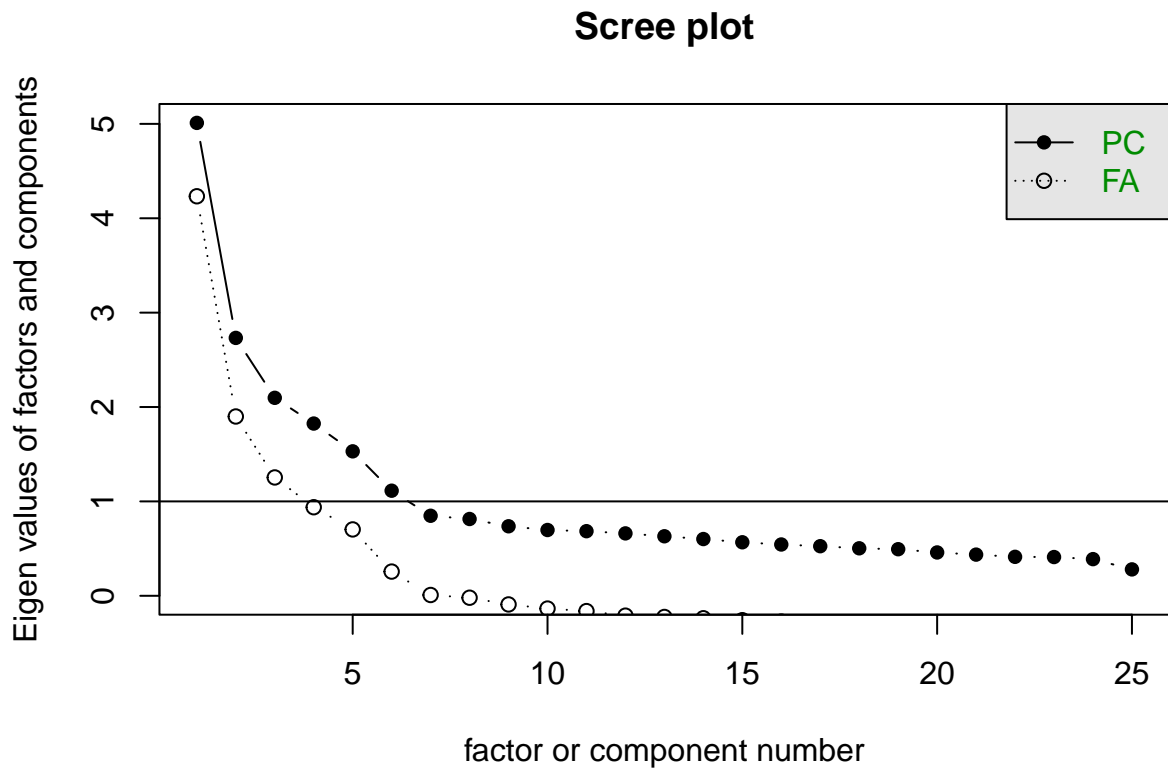
##          u1          u2
## O5 0.9501722 0.9430729
## O4 0.9497621 0.9253802
## A1 0.9437979 0.9153293
## O2 0.9314935 0.9083777
## O1 0.8725032 0.8434360
## A4 0.8276622 0.8308149
## O3 0.8090390 0.7789369
## A2 0.7375881 0.7638136
## N5 0.7363367 0.7303181
## C3 0.7249751 0.7158668

```

Appropriate number of latent factors

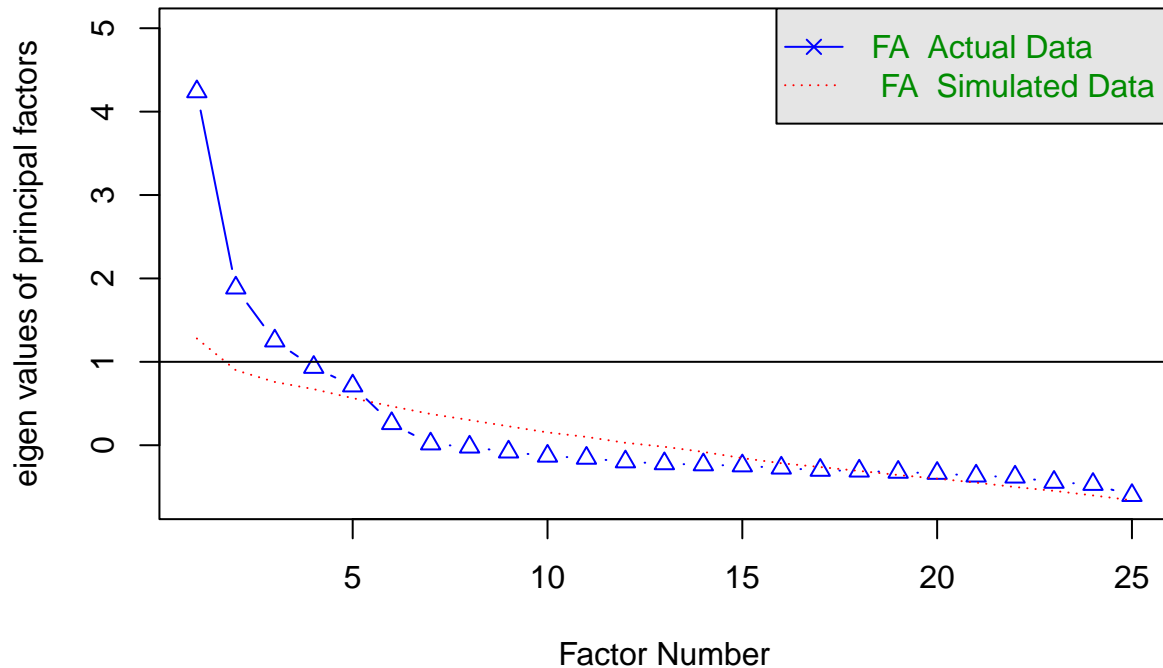
There are different criteria, among which the **Scree plot** (Cattel 1966) and **parallel analysis** (Horn 1965) stand out. According to the following graphical outputs, **5** is considered to be the **optimal number of factors** (parallel analysis), despite **3** are the appropriate according to the first graphical scree plot.

```
# Scree plot  
scree(poly_cor)
```



```
#Parallel analysis  
fa.parallel(poly_cor,n.obs=100,fa="fa",fm="ml")
```

Parallel Analysis Scree Plots



Parallel analysis suggests that the number of factors = 5 and the number of components = NA
 The factorial model with 5 factors implementing a varimax rotation to seek a simpler interpretation is performed.

```
modelo_varimax<-fa(poly_cor,nfactors = 5,rotate = "varimax",
  fa="mle")
```

```
# The rotated factorial matrix is shown
print(modelo_varimax$loadings,cut=0)
```

```
##
## Loadings:
##   MR2  MR1  MR3  MR5  MR4
## A1  0.121  0.036  0.024 -0.406 -0.082
## A2  0.029  0.204  0.143  0.613  0.068
## A3  0.006  0.315  0.107  0.637  0.059
## A4 -0.059  0.180  0.230  0.425 -0.105
## A5 -0.111  0.389  0.089  0.536  0.062
## C1  0.005  0.061  0.531  0.024  0.192
## C2  0.088  0.028  0.647  0.110  0.105
## C3 -0.021  0.021  0.547  0.119 -0.004
## C4  0.252 -0.063 -0.605 -0.039 -0.112
## C5  0.295 -0.170 -0.553 -0.056  0.033
## E1  0.042 -0.573  0.040 -0.100 -0.069
## E2  0.246 -0.678 -0.091 -0.104 -0.037
## E3  0.021  0.535  0.079  0.266  0.267
```

```

## E4 -0.099  0.642  0.100  0.300 -0.084
## E5  0.032  0.498  0.316  0.091  0.208
## N1  0.767  0.075 -0.039 -0.216 -0.085
## N2  0.746  0.033 -0.030 -0.193 -0.019
## N3  0.732 -0.057 -0.066 -0.031  0.001
## N4  0.582 -0.333 -0.172 -0.004  0.066
## N5  0.538 -0.154 -0.035  0.102 -0.145
## O1  0.005  0.217  0.119  0.066  0.495
## O2  0.189  0.004 -0.097  0.087 -0.452
## O3  0.025  0.299  0.084  0.132  0.584
## O4  0.229 -0.182 -0.016  0.153  0.372
## O5  0.096 -0.007 -0.057 -0.016 -0.532
##
##
##          MR2  MR1  MR3  MR5  MR4
## SS loadings  2.670 2.414 1.962 1.783 1.464
## Proportion Var 0.107 0.097 0.078 0.071 0.059
## Cumulative Var 0.107 0.203 0.282 0.353 0.412

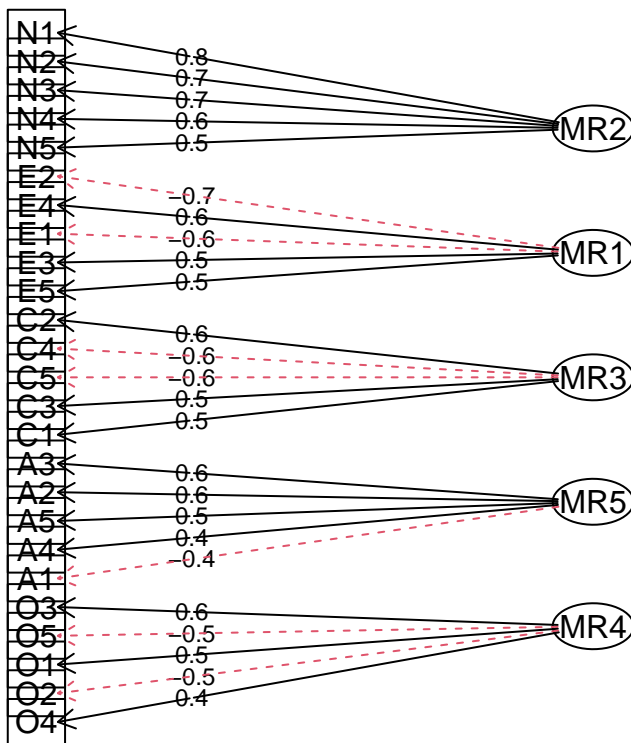
```

Visually we could make the effort to see what variables each correlates with one of the factors, but it is very tedious. So we use the following representation in diagram mode.

In this diagram, among other things, it is seen that the first factor is associated with the items E1, E2, E3, E4 and E5, which are the items of the questionnaire that try to identify the quality of extraversion.

```
fa.diagram(modelo_varimax)
```

Factor Analysis



Another way to do the previous analysis.

```

# Remember that package 'stats' is required for 'factanal' function
# This function only performs the mle method
FA<-factanal(bfi_s,factors=5, rotation="varimax")
FA$loadings

```

```

##
## Loadings:
##   Factor1 Factor2 Factor3 Factor4 Factor5
## A1  0.113          -0.360
## A2          0.177  0.150  0.582
## A3          0.271  0.108  0.647
## A4          0.152  0.233  0.442 -0.102
## A5 -0.112  0.337          0.587
## C1          0.518          0.204
## C2          0.620  0.137  0.130
## C3          0.544  0.131
## C4  0.229        -0.627
## C5  0.275 -0.185 -0.564
## E1          -0.586        -0.123
## E2  0.230 -0.677        -0.150
## E3          0.479          0.336  0.302
## E4 -0.104  0.599          0.372
## E5          0.483  0.313  0.127  0.229
## N1  0.798          -0.216
## N2  0.780          -0.202
## N3  0.714
## N4  0.555 -0.355 -0.185
## N5  0.520 -0.182          0.106 -0.134
## O1          0.177  0.105          0.519
## O2  0.174          -0.113  0.112 -0.434
## O3          0.256          0.165  0.605
## O4  0.214 -0.219          0.142  0.373
## O5          -0.508
##
##           Factor1 Factor2 Factor3 Factor4 Factor5
## SS loadings    2.648  2.230  1.955  1.944  1.502
## Proportion Var  0.106  0.089  0.078  0.078  0.060
## Cumulative Var  0.106  0.195  0.273  0.351  0.411

```