Factor analysis exercises

Christopher Wilson

Introduction

Before we begin, install and load the following packages:

- tidyverse
- psych
- GPArotation

For this session, we will use the BFI dataset from the psych package. It is made up of 25 self-report personality items from the International Personality Item Pool, gender, education level and age for 2800 subjects and used in the Synthetic Aperture Personality Assessment.

The personality items are split into 5 categories: Agreeableness (A), Conscientiousness (C), Extraversion(E), Neuroticism(N), Openness(O).

Each item was answered on a six point scale: 1 Very Inaccurate, 2 Moderately Inaccurate, 3 Slightly Inaccurate, 4 Slightly Accurate, 5 Moderately Accurate, 6 Very Accurate.

To load the data file, use the following code:

```
data("bfi")
```

Note: columns 1 to 25 are the data we are interested in. To include only these columns in analysis, we use the code below to make a subset of the data

We are also using the na.omit() command to remove any rows that have empty values

```
bfidata <- bfi[1:25] %>% na.omit()

# We are also using the na.omit() command to remove any rows that have empty
values
```

1. Use the describe() command to review a summary of the data

```
describe(bfidata)

##    vars    n mean   sd median trimmed  mad min max range  skew kurtosis
## A1    1 2436 2.41 1.41      2    2.22 1.48   1   6     5  0.84    -0.28
## A2    2 2436 4.80 1.18      5    4.97 1.48   1   6     5 -1.12     1.01
## A3    3 2436 4.60 1.31      5    4.79 1.48   1   6     5 -1.01     0.46
## A4    4 2436 4.69 1.49      5    4.92 1.48   1   6     5 -1.02     0.02
## A5    5 2436 4.54 1.27      5    4.70 1.48   1   6     5 -0.84     0.13
## C1    6 2436 4.53 1.24      5    4.66 1.48   1   6     5 -0.87     0.32
## C2    7 2436 4.37 1.32      5    4.50 1.48   1   6     5 -0.74    -0.15
## C3    8 2436 4.30 1.29      5    4.41 1.48   1   6     5 -0.68    -0.15
## C4    9 2436 2.55 1.38      2    2.41 1.48   1   6     5  0.61    -0.59
## C5   10 2436 3.31 1.63      3    3.26 1.48   1   6     5  0.06    -1.23
## E1   11 2436 2.98 1.63      3    2.86 1.48   1   6     5  0.37    -1.09
## E2   12 2436 3.15 1.61      3    3.07 1.48   1   6     5  0.23    -1.15
## E3   13 2436 3.98 1.35      4    4.05 1.48   1   6     5 -0.47    -0.46
## E4   14 2436 4.41 1.47      5    4.58 1.48   1   6     5 -0.82    -0.32
## E5   15 2436 4.39 1.34      5    4.53 1.48   1   6     5 -0.78    -0.11
## N1   16 2436 2.94 1.58      3    2.84 1.48   1   6     5  0.37    -1.02
## N2   17 2436 3.52 1.53      4    3.52 1.48   1   6     5 -0.08    -1.06
## N3   18 2436 3.22 1.59      3    3.17 1.48   1   6     5  0.14    -1.18
## N4   19 2436 3.20 1.57      3    3.14 1.48   1   6     5  0.20    -1.09
## N5   20 2436 2.97 1.62      3    2.85 1.48   1   6     5  0.38    -1.07
## O1   21 2436 4.81 1.13      5    4.96 1.48   1   6     5 -0.90     0.46
## O2   22 2436 2.68 1.55      2    2.53 1.48   1   6     5  0.62    -0.76
## O3   23 2436 4.45 1.21      5    4.57 1.48   1   6     5 -0.77     0.32
## O4   24 2436 4.93 1.19      5    5.13 1.48   1   6     5 -1.24     1.18
## O5   25 2436 2.47 1.32      2    2.32 1.48   1   6     5  0.76    -0.18
##       se
## A1 0.03
## A2 0.02
## A3 0.03
## A4 0.03
## A5 0.03
## C1 0.03
## C2 0.03
## C3 0.03
## C4 0.03
## C5 0.03
## E1 0.03
## E2 0.03
## E3 0.03
## E4 0.03
## E5 0.03
## N1 0.03
## N2 0.03
## N3 0.03
## N4 0.03
## N5 0.03
## O1 0.02
## O2 0.03
## O3 0.02
## O4 0.02
## O5 0.03
```

1. Check the number of complete responses using the commands:

```
bfidata %>% complete.cases() %>% sum()
```

```
## [1] 2436
```

1. Run Bartletts test of sphericity

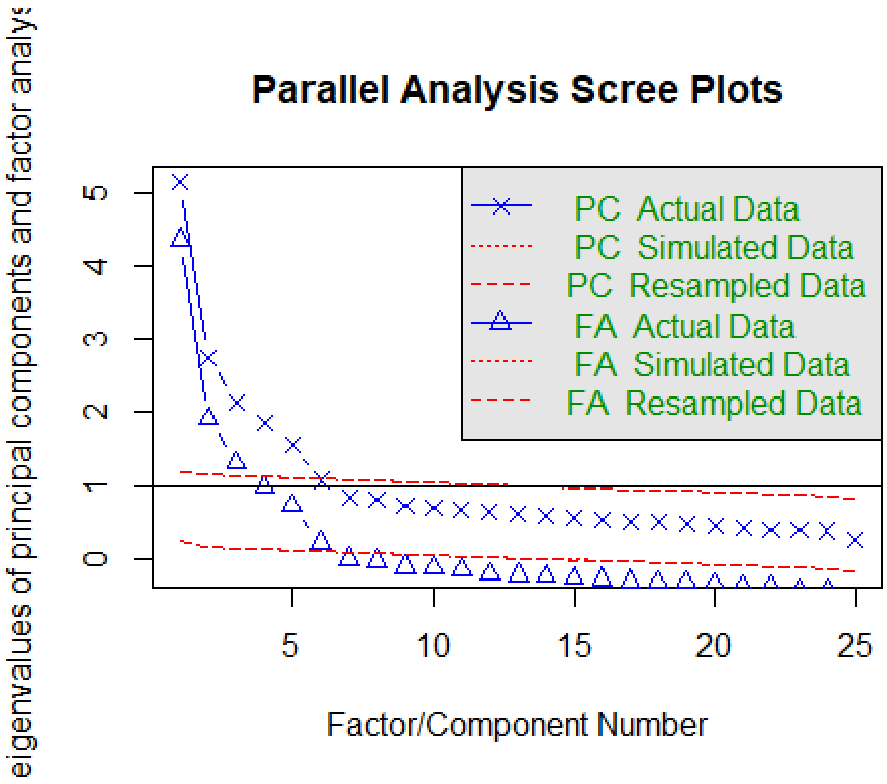
```
# make a correlation matrix first

bfi.maxtrix <- cor(bfidata)

# run bartlettt's test
cortest.bartlett(bfi.maxtrix, n= 2436)

## $chisq
## [1] 18146.07
##
## $p.value
## [1] 0
##
## $df
## [1] 300
```

1. Run the Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy

```
KMO(bfidata) # using the data

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = bfidata)
## Overall MSA =  0.85
## MSA for each item =
##   A1   A2   A3   A4   A5   C1   C2   C3   C4   C5   E1   E2   E3   E4   E5
## 0.75 0.84 0.87 0.88 0.90 0.84 0.80 0.85 0.83 0.86 0.84 0.88 0.90 0.88 0.89
##   N1   N2   N3   N4   N5   O1   O2   O3   O4   O5
## 0.78 0.78 0.86 0.89 0.86 0.86 0.78 0.84 0.77 0.76

#OR

KMO(bfi.maxtrix) # using the correlation matrix

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = bfi.maxtrix)
## Overall MSA =  0.85
## MSA for each item =
##   A1   A2   A3   A4   A5   C1   C2   C3   C4   C5   E1   E2   E3   E4   E5
## 0.75 0.84 0.87 0.88 0.90 0.84 0.80 0.85 0.83 0.86 0.84 0.88 0.90 0.88 0.89
##   N1   N2   N3   N4   N5   O1   O2   O3   O4   O5
## 0.78 0.78 0.86 0.89 0.86 0.86 0.78 0.84 0.77 0.76
```

1. Run a parallel analysis to determine the number of factors

```
fa.parallel(bfidata)
```



```
## Parallel analysis suggests that the number of factors =  6  and the number
of components =  5
```

1. Run a factor analysis based on the suggested number of factors (no rotation)

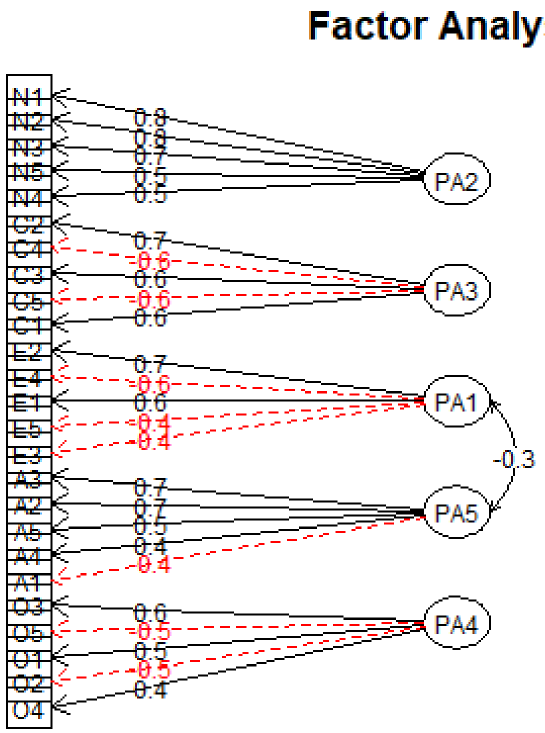
```
factoranalysis1 <- fa(bfidata, nfactors = 6, fm="pa", max.iter = 100, rotate
= "none")
```

1. Interpret the output and determine an optimal number of factors based on: scree plot, variance levels, eigenvalues

Based on a combination of eigenvalues, scree plot, variance explained and SS loadings, there appear to be 5 factors in the data

1. Re-run factor analysis with optimal number of factors and use “oblimin” rotation

```
factoranalysis2 <- fa(bfidata, nfactors = 5, fm="pa", max.iter = 100, rotate
= "oblimin")

fa.diagram(factoranalysis2) ## can use fa.diagram to see factors and the
questions that Load onto them
```



1. Run cronbach’s alpha on the subscales using the loadings from the pattern matrix

To do this, We need to make subsets of data based on the questions that load onto each factor. Seperate analysis is then run for each question set. For this example, I will analyse the factor labelled as pa1 in the diagram, which appears to contain all of the extraversion questions.

```
pa1 <- bfidata %>% select(E1,E2,E3,E4,E5)

 psych::alpha(pa1, check.keys=TRUE)

## Warning in psych::alpha(pa1, check.keys = TRUE): Some items were
negatively correlated with total scale and were automatically reversed.
##  This is indicated by a negative sign for the variable name.

## 
## Reliability analysis   
## Call: psych::alpha(x = pa1, check.keys = TRUE)
## 
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean  sd median_r
##       0.77      0.77    0.73      0.39 3.3 0.0074  4.1 1.1     0.39
## 
##  lower alpha upper     95% confidence boundaries
## 0.75 0.77 0.78 
## 
##  Reliability if an item is dropped:
##     raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
## E1-      0.73      0.73    0.68      0.40 2.7   0.0088 0.0046  0.39
## E2-      0.69      0.70    0.64      0.36 2.3   0.0101 0.0026  0.36
## E3       0.73      0.73    0.68      0.40 2.7   0.0087 0.0072  0.40
## E4       0.71      0.71    0.65      0.38 2.4   0.0096 0.0033  0.38
## E5       0.75      0.75    0.69      0.42 2.9   0.0083 0.0049  0.42
##
##  Item statistics
##        n raw.r std.r r.cor r.drop mean  sd
## E1- 2436  0.72  0.70  0.59   0.52  4.0 1.6
## E2- 2436  0.78  0.77  0.70   0.61  3.8 1.6
## E3  2436  0.68  0.70  0.58   0.50  4.0 1.4
## E4  2436  0.75  0.75  0.67   0.58  4.4 1.5
## E5  2436  0.65  0.67  0.53   0.46  4.4 1.3
##
## Non missing response frequency for each item
##       1    2    3    4    5    6 miss
## E1 0.24 0.24 0.14 0.16 0.13 0.09    0
## E2 0.19 0.24 0.13 0.21 0.14 0.10    0
## E3 0.06 0.11 0.15 0.30 0.26 0.12    0
## E4 0.05 0.09 0.09 0.16 0.34 0.26    0
## E5 0.04 0.08 0.10 0.22 0.34 0.21    0

 ## Why does the code say psych::alpha() instead of just alpha() ?
 ## Because there are other r functions that are called alpha, for example in
the ggplot package. If you load packages that contain functions with the same
name, you can make sure the correct functon is being run by specifying which
package it should come from.
```