Behavioral Software Engineering - Example of psychometric evaluation with R

Daniel Graziotin, Per Lenberg, Robert Feldt, Stefan Wagner

2020-05-18

Contents

Introduction

The present document provides an executable hands-on introductory psychometric validation example written for a behavioral software engineering audience.

The present document is part of a paper:

> Daniel Graziotin, Per Lenberg, Robert Feldt, Stefan Wagner (2020): Behavioral Sofware Engineering: Methodological Introduction to Psychometrics. Under Review

Even though it is as self-contained as possible, we recommend reading it after reading the paper. The present document is also a R Markdown file. Its text version is interactive and can be executed directly in R Studio, making it completely reproducible.

Context

The present document contains an example, with simulated data, of the psychometric validation phases of a norm-based measurement instrument for assessing an overall construct with five sub-constructs.

Our fictitious construct is the "individual perception styles of source code" that, through a literature review (or, perhaps, after a grounded theory study), we believe is mainly composed of, or highly related, to the following five constructs: code curiosity, programming paradigm flexibility, learning disposition, collaboration propensity, and comfort in novelty.

We develop a measurement instrument with 31 items, all represented by likert items from 1 to 5, which is self-assessed by the individual software engineer. The likert items would ideally form five likert scales, that would actually emerge after an exploratory factors analysis. However, when we develop scales we have a rough idea of the related constructs anyway. We represent here what we expect the exploratory factor analysis to show, with all items grouped by what we might see as potential factors.

- F1, "Code curiosity"

  - 7 items, F1_1 to F1_7
- F2 “Programming paradigm flexibility”
  - 4 items, F2_1 to F1_4
- F3 “Learning disposition”
  - 9 items, F3_1 to F3_9
- F4 “Collaboration propensity”
  - 4 items, F4_1 to F4_4
- F5 “Comfort in novelty”
  - 5 items, F5_1 to F4_7

What the items actually are is not interesting for the purposes of the present document. Our team proceeds to psychometrically evaluate the measurement instrument, in particular, to reduce the number of items. Then, to validate the items belonging to the factors and, possibly, reducing the number of items agin. Finally, to offer a reasoning on statistical properties, reliability, and validity.

Requirements and options

The following are requirements that are typically found in a basic psychometric validation, with the exclusion of `fabricatr`, which is for fabricating data and we use for developing this example only.

```
if (!require("fabricatr")){
  install.packages("fabricatr", dep = T)
}
```

```
## Loading required package: fabricatr
```

```
require("fabricatr")

if (!require("psych")){
  install.packages("psych", dep = T)
}
```

```
## Loading required package: psych
```

```
require("psych")

if (!require("knitr")){
  install.packages("knitr", dep = T)
}
```

```
## Loading required package: knitr
```

```
require("knitr")

knitr::opts_chunk$set(echo = TRUE, cache = FALSE, width = 50)
```

We offer a dataset with pre-populated data that allows repetition of the entire document. For discarding our provided data and simulating new data (which should conform, to a fair extent, to what we expect in the various sections), set the `safeguard` variable to FALSE or delete the `graziotin_et_al-bse_psychometrics_example.csv` file. The new dataset will behave very similarly to the provided one–after all, we used the very same code to generate it. This option will allow reproducibility only.

For full repeatability, but mostly for better clarity, we recommend to start with our provided dataset. The reason is that the rest of this tutorial will provide reasoning around certain values for items that will slightly change in case a new dataset is generated.

The following are options that are useful for repetition and replication of our example.

```r
# set to FALSE to regenerate the dataset
safeguard <- TRUE && file.exists('graziotin_et_al-bse_psychometrics_example.csv')
overall.sample.size = 142
pilot.test.sample.size = 23
reliability.sample.size = 48
```

Data simulation

This section explains how we constructed the simulated dataset `graziotin_et_al-bse_psychometrics_example.csv`.

This section can be safely skipped as it does not pertain to psychometric evaluation. We provide the data simulation part for full reproducibility.

```r
if (!safeguard) {
  # 31 items belonging to 5 clusters
  items.initialset <- c(rep(1, 7), # cluster 1
                        rep(2, 4), # cluster 2
                        rep(3, 9), # cluster 3
                        rep(4, 4), # cluster 4
                        rep(5, 7)) # cluster 5

  likert.brakes <- c(-Inf,-1.5,-0.5, 0.5, 1.5, Inf)
  likert.values <- c("1", "2", "3", "4", "5")

  counter <- 0
  var.names <-
    unlist(lapply(items.initialset, function(x) {
      paste0("F", x, "_", counter <<- counter + 1)
    }))
  normalized.response <-
    draw_normal_icc(clusters = items.initialset,
                    ICC = sample(seq(0.4, 0.7, by = 0.05), 1),
                    sd = sample(seq(0.5, 0.9, by = 0.1), 1))
  ordered.response <- draw_ordered(x = normalized.response,
                                   breaks = likert.brakes,
                                   break_labels = likert.values)
  full.df <- rbind(ordered.response)
  colnames(full.df) <- var.names

  for (i in seq(1, overall.sample.size - 1)) {
    normalized.response <-
      draw_normal_icc(clusters = items.initialset,
                      ICC = sample(seq(0.4, 0.7, by = 0.05), 1),
                      sd = sample(seq(0.5, 0.9, by = 0.1), 1))
    ordered.response <- draw_ordered(x = normalized.response,
                                     breaks = likert.brakes,
                                     break_labels = likert.values)
    full.df <- rbind(full.df, ordered.response)
  }
  # each row is a participant id, from 1 to overall.sample.size
  rownames(full.df) <- seq(1, overall.sample.size)
}
```

Up to this point we have generated a perfect dataset. All items cluster as we wish them to be. We now add noise to the data. For two items, `F1_4` and `F5_25`, we simulate oddly worded or uninteresting items that

most users rate at the maximum and minimum values, respectively. This will allow us later on to drop these items during item facility analysis.

```
if (!safeguard) {
  full.df[, 'F1_4'] <- draw_ordered(
    x = rnorm(n = overall.sample.size, mean = 2.7, sd = 0.8),
    breaks = likert.brakes,
    break_labels = likert.values
  )
  full.df[, 'F5_25'] <- draw_ordered(
    x = rnorm(n = overall.sample.size, mean = 1.2, sd = 0.8),
    breaks = likert.brakes,
    break_labels = likert.values
  )
}
```

For other three items, `F4_21` to `F4_23`, we get even more malicious. Factor 4 only had 4 candidates for a representative cluster. We generate values for these three items again such that they do not likely belong to the cluster 'F4' anymore, but they do not provide clear extreme values that would get caught during item analysis.

```
if (!safeguard) {
  full.df[, 'F4_21'] <- draw_ordered(
    x = rnorm(n = overall.sample.size, mean = 1.5, sd = 2.3),
    breaks = likert.brakes,
    break_labels = likert.values
  )
  full.df[, 'F4_22'] <- draw_ordered(
    x = rnorm(n = overall.sample.size, mean = 3, sd = 2.1),
    breaks = likert.brakes,
    break_labels = likert.values
  )
  full.df[, 'F4_23'] <- draw_ordered(
    x = rnorm(n = overall.sample.size, mean = 4.5, sd = 1.8),
    breaks = likert.brakes,
    break_labels = likert.values
  )

  write.csv(x = as.data.frame(full.df),
            file = 'graziotin_et_al-bse_psychometrics_example.csv',
            row.names = F)
}
```

Item Review and Item Analysis

For brevity, as item review is mostly a review and reasoning of items, we assume that the 31 items for five factors are already the result of item review. So we proceed to perform a pilot test for item analysis.

We simulate a pilot study with the first `pilot.test.sample.size` lines of the dataframe. This is `pilot.df`. We call psych function describe() and start our item analysis.

```
full.df <-
  read.csv(file = 'graziotin_et_al-bse_psychometrics_example.csv')
pilot.df <- head(full.df, n = pilot.test.sample.size)
desc <- describe(pilot.df)
desc
```

```
##       vars  n mean   sd median trimmed  mad min max range  skew kurtosis   se
## F1_1     1 23 2.91 1.08      3    2.89 1.48   1   5     4  0.16    -0.65 0.23
## F1_2     2 23 2.74 0.96      3    2.79 1.48   1   4     3 -0.37    -0.90 0.20
## F1_3     3 23 2.78 1.09      3    2.79 1.48   1   5     4  0.00    -0.88 0.23
## F1_4     4 23 4.96 0.21      5    5.00 0.00   4   5     1 -4.19    16.26 0.04
## F1_5     5 23 2.83 1.11      3    2.79 1.48   1   5     4  0.33    -0.79 0.23
## F1_6     6 23 2.87 1.18      3    2.84 1.48   1   5     4  0.08    -0.99 0.25
## F1_7     7 23 2.87 1.01      3    2.84 1.48   1   5     4  0.25    -1.03 0.21
## F2_8     8 23 3.04 0.98      3    3.00 1.48   1   5     4  0.20    -0.32 0.20
## F2_9     9 23 3.30 0.97      3    3.32 1.48   1   5     4 -0.32    -0.33 0.20
## F2_10   10 23 3.04 0.93      3    2.95 1.48   2   5     3  0.57    -0.60 0.19
## F2_11   11 23 3.17 0.65      3    3.16 0.00   2   5     3  0.79     1.08 0.14
## F3_12   12 23 2.96 1.15      3    2.95 1.48   1   5     4  0.25    -0.74 0.24
## F3_13   13 23 2.83 0.94      3    2.89 1.48   1   4     3 -0.31    -0.94 0.20
## F3_14   14 23 2.83 1.07      3    2.79 1.48   1   5     4  0.33    -0.51 0.22
## F3_15   15 23 2.91 1.04      3    2.95 1.48   1   5     4 -0.07    -0.84 0.22
## F3_16   16 23 2.65 1.07      3    2.63 1.48   1   5     4  0.26    -0.77 0.22
## F3_17   17 23 2.74 0.92      3    2.68 1.48   1   5     4  0.50    -0.18 0.19
## F3_18   18 23 2.83 1.15      3    2.79 1.48   1   5     4 -0.02    -0.64 0.24
## F3_19   19 23 2.78 0.95      3    2.84 1.48   1   4     3 -0.19    -1.09 0.20
## F3_20   20 23 3.04 0.71      3    3.00 0.00   2   5     3  0.69     0.84 0.15
## F4_21   21 23 3.70 1.43      4    3.84 1.48   1   5     4 -0.47    -1.29 0.30
## F4_22   22 23 4.57 1.04      5    4.84 0.00   1   5     4 -2.39     4.74 0.22
## F4_23   23 23 4.96 0.21      5    5.00 0.00   4   5     1 -4.19    16.26 0.04
## F4_24   24 23 2.78 0.85      3    2.84 0.00   1   4     3 -0.88     0.13 0.18
## F5_25   25 23 4.04 0.71      4    4.11 0.00   2   5     3 -0.80     1.21 0.15
## F5_26   26 23 2.91 1.04      3    2.84 1.48   1   5     4  0.39    -0.68 0.22
## F5_27   27 23 3.04 1.02      3    3.11 0.00   1   5     4 -0.57    -0.08 0.21
## F5_28   28 23 2.65 1.11      3    2.63 1.48   1   5     4  0.11    -0.88 0.23
## F5_29   29 23 2.91 1.08      3    2.89 1.48   1   5     4  0.16    -0.65 0.23
## F5_30   30 23 2.96 0.98      3    3.05 1.48   1   4     3 -0.48    -0.94 0.20
## F5_31   31 23 2.91 1.08      3    2.95 1.48   1   5     4 -0.25    -0.78 0.23
```

The function provides useful descriptive statistics for each item. Our first observation is that some items have mean and median that deviate notably from the (in our case) central value of 3. These are `F1_4`, `F4_22`, `F4_23`, with a mean value approaching 5, and `F5_25`, with a mean value of 4.04, which is high but not alarming yet. Items `F1_4` and `F4_23` are also those with the smallest standard deviation of 0.21, which results in a variance of 0.04. The other two items show standard deviations in line with the rest of the dataset. Furthermore, `F1_4` and `F4_23` have unusual values for skew and kurtosis as well, indicating that the data is strongly biased on a value (indeed, their range is from 4 to 5). We conclude that items `F1_4` and `F4_23` have a high item facility (item difficulty), so they are candidate for deletion.

As for item discrimination, being our example one for trait measurement, we calculate how the two items correlate with those that are supposedly part of their factors.

```r
cor(pilot.df[, grepl("F1_" , names(pilot.df))], method = c("kendall"))
```

```
##            F1_1        F1_2        F1_3        F1_4        F1_5        F1_6
## F1_1  1.00000000  0.69676640  0.40410397 -0.03061384  0.78975397  0.64500204
## F1_2  0.69676640  1.00000000  0.41905498 -0.04702449  0.60392630  0.62954379
## F1_3  0.40410397  0.41905498  1.00000000 -0.04556977  0.41221507  0.60006751
## F1_4 -0.03061384 -0.04702449 -0.04556977  1.00000000 -0.06091449 -0.02992751
## F1_5  0.78975397  0.60392630  0.41221507 -0.06091449  1.00000000  0.68180919
## F1_6  0.64500204  0.62954379  0.60006751 -0.02992751  0.68180919  1.00000000
## F1_7  0.67202944  0.56452429  0.48974943 -0.23386171  0.75738930  0.61590315
##            F1_7
```

```
## F1_1  0.6720294
## F1_2  0.5645243
## F1_3  0.4897494
## F1_4 -0.2338617
## F1_5  0.7573893
## F1_6  0.6159031
## F1_7  1.0000000
```

```
cor(pilot.df[, grepl("F4_" , names(pilot.df))], method = c("kendall"))
```

```
##             F4_21       F4_22      F4_23       F4_24
## F4_21  1.00000000 -0.06830148  0.1115476  0.09441808
## F4_22 -0.06830148  1.00000000 -0.1082363  0.04275379
## F4_23  0.11154759 -0.10823626  1.0000000 -0.35909479
## F4_24  0.09441808  0.04275379 -0.3590948  1.00000000
```

Item `F1_4` behaves differently to other items of its cluster as it shows near zero correlation to all items but a weak one to `F1_7`.

Item `F4_23` does not show particular differences in its cluster (we manipulated Factor 4 responses to result in a likely void factor).

We conclude that only item `F1_4` possesses a high discrimination and we therefore are confident in removing it from the measurement instrument. `F4_23` would show high discrimination as well, but not compared to the rest of its theorized cluster, yet it has high item facility. We decide to drop it as well. We decide to not drop `F4_22` yet, but the entire cluster for Factor 4 looks suspicious already.

```
pilot.afteritemanalysis.df <-
  pilot.df[,-which(names(pilot.df) %in% c("F1_4", "F4_23"))]
```

Factor Analysis

Our psychometric evaluation study foresees an exploratory factor analysis (EFA) with an idea of items belonging to some possible constructs. While EFA does not require known factors at all (it is, after all, an exploration), we can still use it to cut down on items rather than on discovering how many factors we require. EFA *will* show us suggestions for factors, anyway.

We gather data with a new round of participants, this time aiming at a larger sample.

```
# we drop the pilot participants
study.df <-
  tail(full.df, n = NROW(full.df) - pilot.test.sample.size)
# we drop those columns that we identified after item analysis
study.beforeefa.df <-
  study.df[,-which(names(study.df) %in% c("F1_4", "F4_23"))]
```

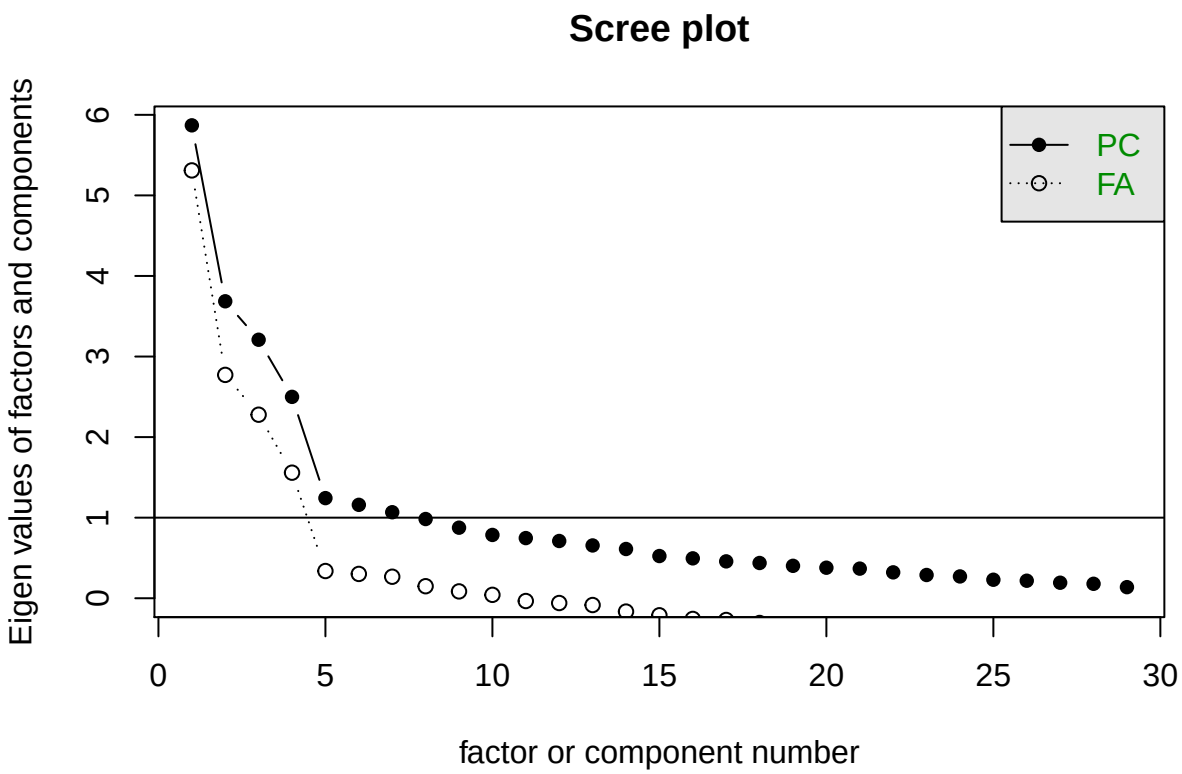
Following the structure of our paper, we show here two possibilities for factor extraction, namely the scree plot and the very simple structure, and then we provide an entire session of factor loading estimation, factor extraction, and factor rotation with principal axis factoring.

Factor extraction: scree plot

The scree test is based on eigenvalues. The `psych` package is nice enough to perform a principal component analysis (PCA) as well as of a principal axis factoring (PAF) for generating the plot.

We plot the scree test using our sample with 29 variables.

```
scree(study.beforeefa.df)
```



The breaking point starts with the fifth component for PCA and with the fifth factor for PAF, suggesting that 4 is the number of factors to extract.

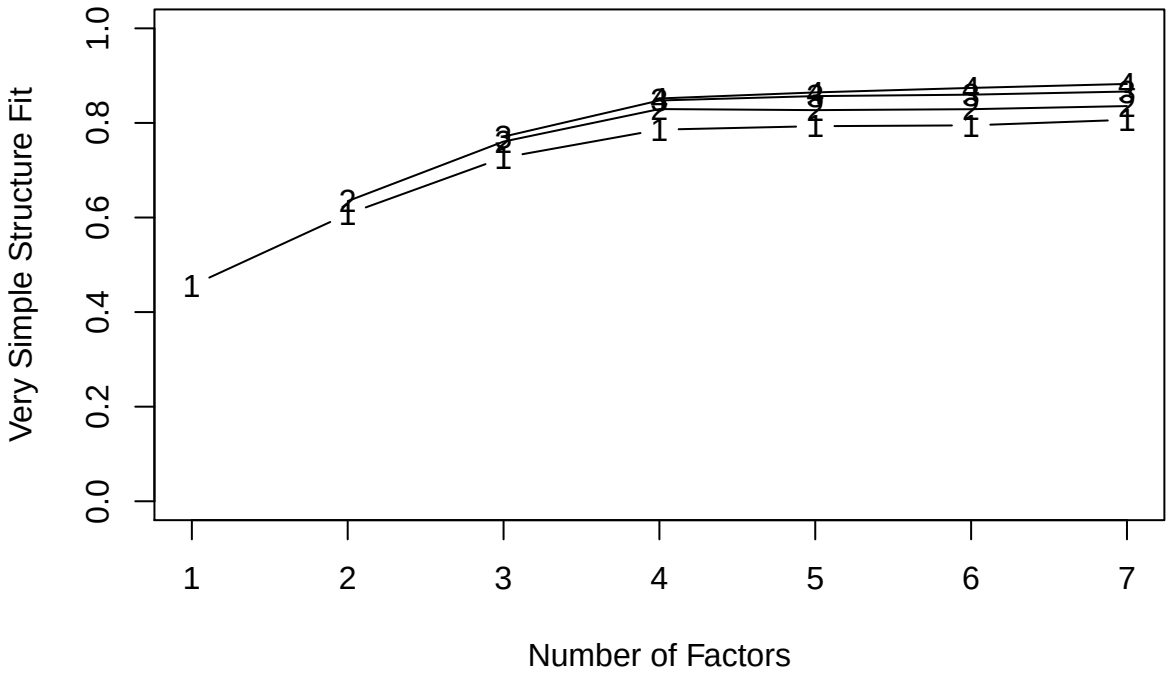
Factor extraction: Very Simple Structure

An alternative for factor extraction is the Very Simple Structure, or VSS, which can be used for extracting factors before their rotation.

We plot the structure using our initial five factor model with 29 items. As written in our paper, the VSS criterion requires a user specified number of factors, which should be above the target number of factors. We specify seven factors using the `n = 7` parameter.

```
vss.test.five <-
  vss(study.beforeefa.df,
      title = "Very Simple Structure of our fictitious five factor model", n = 7)
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.
```




```
vss.test.five
```

```
##
## Very Simple Structure of  Very Simple Structure of our fictitious five factor model
## Call: vss(x = study.beforeefa.df, n = 7, title = "Very Simple Structure of our fictitious five factor
## VSS complexity 1 achieves a maximimum of 0.81  with  7  factors
## VSS complexity 2 achieves a maximimum of 0.84  with  7  factors
##
## The Velicer MAP achieves a minimum of 0.02  with  4  factors
## BIC achieves a minimum of  -1071.41  with  4  factors
## Sample Size adjusted BIC achieves a minimum of  -135.64  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.46 0.00 0.036 377   995 3.3e-57    40.6 0.46 0.117  -807   385     1.0
## 2 0.61 0.63 0.027 349   714 3.0e-27    27.3 0.63 0.093  -954   149     1.2
## 3 0.73 0.76 0.018 322   487 7.6e-09    17.1 0.77 0.065 -1052   -34     1.2
## 4 0.79 0.83 0.015 296   343 3.1e-02    11.1 0.85 0.036 -1071  -136     1.2
## 5 0.79 0.83 0.018 271   310 5.1e-02    10.0 0.87 0.034  -985  -128     1.3
## 6 0.79 0.83 0.020 247   274 1.2e-01     9.0 0.88 0.029  -907  -126     1.3
## 7 0.81 0.84 0.023 224   247 1.4e-01     7.9 0.89 0.028  -823  -115     1.3
##   eChisq  SRMR eCRMS  eBIC
## 1   2359 0.156 0.162   558
## 2   1323 0.117 0.126  -345
## 3    575 0.077 0.087  -964
## 4    199 0.045 0.053 -1216
## 5    167 0.042 0.051 -1128
## 6    138 0.038 0.048 -1042
## 7    113 0.034 0.046  -958
```

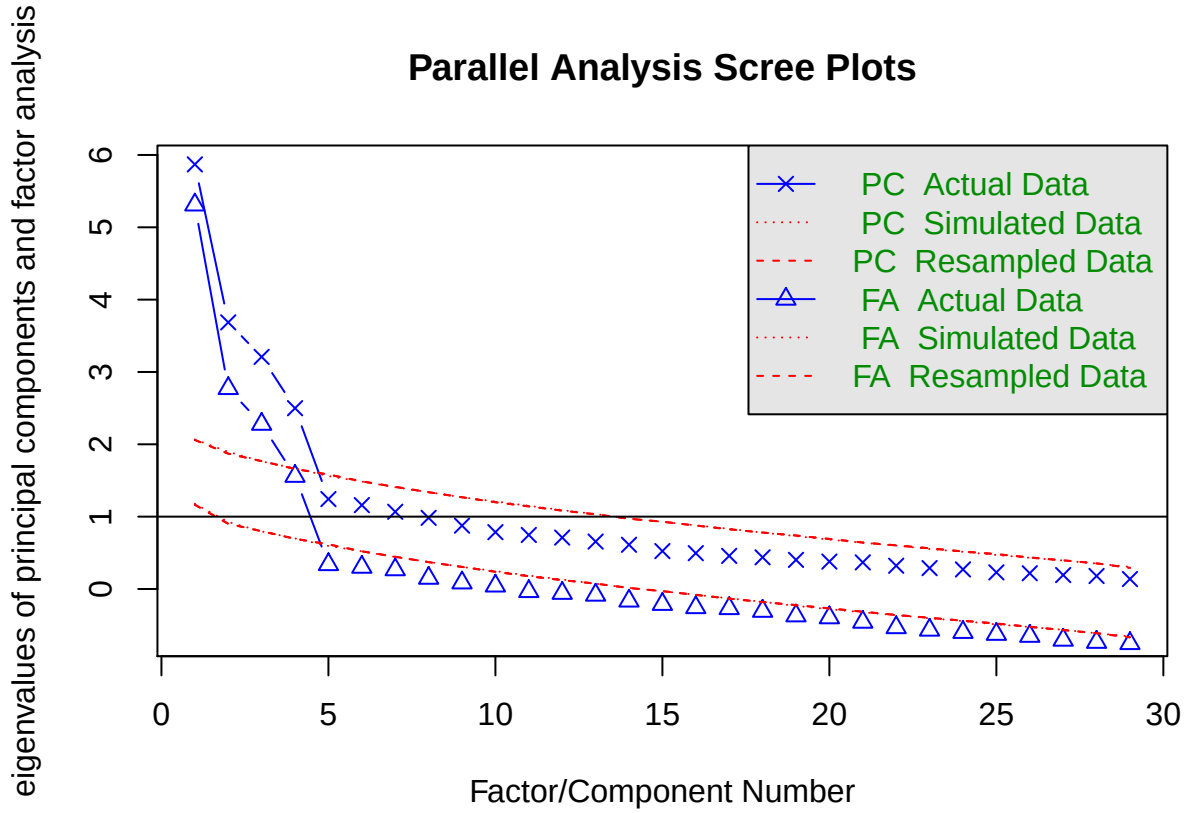
Besides the warning that the the estimated weights are probably incorrect, which would prompt us to use a different factor extraction method anyway, we can observe that the VSS criterion does achieve its maximum with 7 factors.

However, the increase in `vss1` and `vss2` score decreases rapidly after four factors. This can be noted graphically as well as in the console output. The `fit` behaves the same. The VSS still suggests that having four factors is optimal.

Factor extraction: Parallel Analysis

Finally, a robust method for factor extraction is Parallel Analysis (PA), which performs a constant comparison of our solution with randomly generated data.

```
pa.test.five <-
  fa.parallel(study.beforeefa.df)
```




```
## Parallel analysis suggests that the number of factors =  4  and the number of components =  4
```

As with the Scree Plot function, we see that psych `fa.parallel` function performs both a PCA and a PAF with the data. The textual output suggests already that four factors should be extracted. The graph shows our data and with calculated components (crosses) and factors (triangles). They are placed the same as with a Scree Plot. The red lines (upper one for PCA, lower one for PAF), on the other hand, represent the mean generated eingenvalues. Factors are retained as long as they are greater than the mean eigenvalue generated from the random matrices.

For both PCA and PAF, the generated data crosses the lines of the actual eigenvalue between 4 and 5 components and factors. This suggests that 4 factors should be extracted.

Factor loading, extraction, and rotation: principal axis factoring

The R psych package makes it easy to perform loading estimation, factor extraction, and factor rotation with a single function call. We will perform here a principal axis factoring (PAF) to fully go from five factors (the initial assumption) to four factors. We show in this section how a reasoning can be conducted by observing values as returned by functions for loading, extration, and rotation.

Besides our reasoning on pure numbers, we will add elements of confirmatory factory analysis (CFA) to guide our retention of items and factors, that is, measures of fit. First, the change in chi square from factor n to factor n - 1 should go from significant (there is is a bad fit) to insignificant. To the chi square test, we add other two measure of fit, namely the root mean square of the residuals (RMSR) and the root mean square error of approximation (RMSEA) index. RMSR should be as close to zero as possible while RMSEA should be below 0.05.

It should be kept in mind that measures of fit are mere indicators in EFA, being them part of a confirmatory factor analysis (CFA) strategy mostly, and should be taken into consideration together with a sensitivity analysis.

As suggested in our paper, we opt for Principal Axis Factoring (PAF) with a hybrid orthogonal first, oblique rotation then, namely a promax rotation. This means that we have a further criterion for factor reduction, that is reducing factors while the sum of squared loadings is below 1.00.

```
five.factors <-
  fa(
    study.beforeefa.df ,
    fm = "pa",
    nfactors = 5,
    rotate = "promax"
  )
```

```
## Loading required namespace: GPArotation
```

```
five.factors
```

```
## Factor Analysis using method =  pa
## Call: fa(r = study.beforeefa.df, nfactors = 5, rotate = "promax", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PA1   PA2   PA3   PA4   PA5    h2   u2 com
## F1_1  -0.06  0.71 -0.09  0.00 -0.04 0.550 0.45 1.1
## F1_2   0.00  0.74  0.04 -0.10 -0.08 0.558 0.44 1.1
## F1_3  -0.05  0.70  0.12  0.00 -0.04 0.486 0.51 1.1
## F1_5   0.00  0.70 -0.01  0.02  0.12 0.507 0.49 1.1
## F1_6  -0.08  0.76  0.05  0.05  0.19 0.607 0.39 1.2
## F1_7  -0.05  0.73  0.09 -0.06 -0.13 0.548 0.45 1.1
## F2_8  -0.02 -0.10  0.01  0.69 -0.15 0.484 0.52 1.1
## F2_9   0.04 -0.01  0.02  0.84  0.11 0.733 0.27 1.0
## F2_10  0.04 -0.04  0.15  0.64 -0.06 0.433 0.57 1.1
## F2_11  0.05  0.01  0.04  0.63  0.15 0.436 0.56 1.1
## F3_12  0.77  0.00  0.00 -0.03  0.08 0.585 0.42 1.0
## F3_13  0.64  0.04  0.00  0.04  0.07 0.419 0.58 1.0
## F3_14  0.72  0.00 -0.09  0.03 -0.07 0.515 0.48 1.1
## F3_15  0.82  0.05  0.01 -0.07  0.28 0.697 0.30 1.3
## F3_16  0.69  0.05  0.03  0.06  0.00 0.504 0.50 1.0
## F3_17  0.81 -0.02 -0.04  0.08 -0.11 0.700 0.30 1.1
## F3_18  0.65 -0.20  0.06  0.10  0.03 0.517 0.48 1.3
## F3_19  0.79  0.02 -0.08 -0.02  0.03 0.600 0.40 1.0
## F3_20  0.75 -0.10  0.01 -0.07 -0.18 0.639 0.36 1.2
## F4_21 -0.16 -0.01 -0.13  0.19  0.06 0.089 0.91 3.0
```

```
## F4_22  0.04  0.03  0.02  0.01  0.54 0.284 0.72 1.0
## F4_24 -0.06 -0.29  0.05  0.01 -0.05 0.094 0.91 1.2
## F5_25 -0.04 -0.09  0.02 -0.10  0.06 0.026 0.97 3.1
## F5_26  0.03 -0.04  0.72 -0.01  0.04 0.533 0.47 1.0
## F5_27 -0.13  0.02  0.77 -0.05 -0.06 0.579 0.42 1.1
## F5_28 -0.03 -0.03  0.68  0.16 -0.10 0.490 0.51 1.2
## F5_29 -0.06  0.01  0.66  0.07  0.03 0.424 0.58 1.0
## F5_30  0.05 -0.05  0.66  0.10  0.16 0.474 0.53 1.2
## F5_31  0.03  0.16  0.69 -0.07  0.00 0.488 0.51 1.1
##
##                         PA1  PA2  PA3  PA4  PA5
## SS loadings           5.02 3.31 2.95 2.12 0.60
## Proportion Var        0.17 0.11 0.10 0.07 0.02
## Cumulative Var        0.17 0.29 0.39 0.46 0.48
## Proportion Explained  0.36 0.24 0.21 0.15 0.04
## Cumulative Proportion 0.36 0.59 0.81 0.96 1.00
##
##  With factor correlations of
##       PA1   PA2   PA3   PA4   PA5
## PA1  1.00 -0.06  0.20  0.13 -0.09
## PA2 -0.06  1.00 -0.14  0.07 -0.05
## PA3  0.20 -0.14  1.00 -0.06 -0.07
## PA4  0.13  0.07 -0.06  1.00  0.05
## PA5 -0.09 -0.05 -0.07  0.05  1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 5 factors are sufficient.
##
## The degrees of freedom for the null model are  406  and the objective function was  14.23 with Chi S
## The degrees of freedom for the model are 271  and the objective function was  2.98
##
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.05
##
## The harmonic number of observations is  119 with the empirical chi square  167.43  with prob <  1
## The total number of observations was  119  with Likelihood Chi Square =  310.07  with prob <  0.051
##
## Tucker Lewis Index of factoring reliability =  0.946
## RMSEA index =  0.034  and the 90 % confidence intervals are  0 0.052
## BIC =  -985.07
## Fit based upon off diagonal values = 0.97
## Measures of factor score adequacy
##                                                 PA1  PA2  PA3  PA4  PA5
## Correlation of (regression) scores with factors   0.96 0.94 0.93 0.92 0.74
## Multiple R square of scores with factors          0.93 0.88 0.86 0.84 0.55
## Minimum correlation of possible factor scores     0.85 0.75 0.72 0.68 0.10
```

Let us now break down the output of this function.

The function represents the candidate factors with labels `PA1` to `PA5`. The factor numbering does not necessarily correspond to the factors we developed, that is, `PA1` is not automatically assigned to items `F1_x`. Factors with lower numbering are those that provide the highest sum of squared loadings (`SS loadings` in R output) and also the variance explained.

Let us first inspect the computed loadings (which could also be singled out with the `loadings(five.factors)`

function.), in particular, the summary table that provides the sum of squared loadings (SS loadings) and the variance explained for each factor. A first observation is that `PA5` provides a sum of squared loadings below 1, which is an indication for removal. The factor accounts also for only 0.04 variance, and the cumulative variance moves from 0.46 to 0.48 from four factors to five factors.

Goodness of fit with Likelihood Chi Square is already non-significant but approaching significance levels ($p = 0.052$). RMSR and RMSEA are within desirable levels.

If we then move to inspect the loadings of all our single items, we see a table of our F items crossed with the candidate factors PA. Furthermore, the table provides, for each item, the communalities (h2), the unique variance (u2), and the complexity of the component loadings (com).

We observe that the item possessing the highest load to `PA5` is `F4_22` with 0.536. All other items have loadings near 0. Furthermore, most of our items for a factor F have a corresponding factor PA (e.g., `F1` items load mostly on factor `PA2`) while `PA5` seems to meet all Fx items.

Another observation is on items for `F4`. As stated above, `F4_22` is the only one with a meaningful loading on a factor, that is `PA5`, while `F4_21` does not seem to have a meaningful loading on anything. The only remaining item, `F4_24`, provides a loading on `PA2`, which seems associated with `F1`items mostly. Given our previous suspicious on `F4`items, after an internal review, we decide to drop the F4 items entirely and test for a four factors model.

Factor reduction

We decide to drop the F4 items entirely and test for a four factors model.

```
# we drop those columns that we identified after item analysis
study.efa.df <-
  study.beforeefa.df[,-which(names(study.beforeefa.df) %in% c("F4_21", "F4_22", "F4_24"))]
four.factors <-
  fa(study.efa.df ,
     fm = "pa",
     nfactors = 4,
     rotate = "promax")
four.factors
```

```
## Factor Analysis using method =  pa
## Call: fa(r = study.efa.df, nfactors = 4, rotate = "promax", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PA1   PA2   PA3   PA4    h2   u2 com
## F1_1  -0.03  0.73 -0.11  0.00 0.568 0.43 1.0
## F1_2   0.04  0.75  0.03 -0.12 0.556 0.44 1.1
## F1_3  -0.02  0.68  0.11 -0.02 0.462 0.54 1.1
## F1_5   0.01  0.69 -0.04  0.03 0.481 0.52 1.0
## F1_6  -0.07  0.74  0.02  0.07 0.576 0.42 1.0
## F1_7  -0.01  0.73  0.07 -0.07 0.530 0.47 1.0
## F2_8  -0.06 -0.09 -0.01  0.67 0.432 0.57 1.1
## F2_9  -0.03 -0.02 -0.02  0.86 0.720 0.28 1.0
## F2_10 -0.01 -0.04  0.13  0.65 0.438 0.56 1.1
## F2_11 -0.01 -0.01  0.00  0.65 0.416 0.58 1.0
## F3_12  0.77  0.00  0.01 -0.04 0.582 0.42 1.0
## F3_13  0.63  0.04  0.00  0.04 0.414 0.59 1.0
## F3_14  0.72  0.01 -0.07  0.00 0.512 0.49 1.0
## F3_15  0.80  0.04  0.02 -0.07 0.615 0.38 1.0
## F3_16  0.70  0.05  0.04  0.04 0.506 0.49 1.0
## F3_17  0.82 -0.01 -0.02  0.04 0.683 0.32 1.0
## F3_18  0.64 -0.19  0.08  0.08 0.519 0.48 1.2
```

```
## F3_19  0.79  0.02 -0.07 -0.03 0.596 0.40 1.0
## F3_20  0.76 -0.09  0.04 -0.11 0.589 0.41 1.1
## F5_25 -0.04 -0.09  0.03 -0.11 0.024 0.98 2.4
## F5_26  0.05 -0.03  0.71 -0.03 0.528 0.47 1.0
## F5_27 -0.09  0.04  0.76 -0.08 0.567 0.43 1.1
## F5_28 -0.01 -0.01  0.67  0.12 0.473 0.53 1.1
## F5_29 -0.05  0.00  0.66  0.06 0.427 0.57 1.0
## F5_30  0.05 -0.06  0.64  0.09 0.451 0.55 1.1
## F5_31  0.07  0.17  0.69 -0.11 0.495 0.51 1.2
##
##                         PA1  PA2  PA3  PA4
## SS loadings            4.94 3.21 2.92 2.09
## Proportion Var         0.19 0.12 0.11 0.08
## Cumulative Var         0.19 0.31 0.43 0.51
## Proportion Explained   0.38 0.24 0.22 0.16
## Cumulative Proportion 0.38 0.62 0.84 1.00
##
##  With factor correlations of
##       PA1   PA2   PA3  PA4
## PA1  1.00 -0.08  0.16 0.24
## PA2 -0.08  1.00 -0.12 0.06
## PA3  0.16 -0.12  1.00 0.03
## PA4  0.24  0.06  0.03 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 4 factors are sufficient.
##
## The degrees of freedom for the null model are  325  and the objective function was  13.29 with Chi S
## The degrees of freedom for the model are 227  and the objective function was  2.51
##
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.05
##
## The harmonic number of observations is  119 with the empirical chi square  132.9  with prob <  1
## The total number of observations was  119  with Likelihood Chi Square =  265.54  with prob <  0.04
##
## Tucker Lewis Index of factoring reliability =  0.949
## RMSEA index =  0.037  and the 90 % confidence intervals are  0.009 0.056
## BIC =  -819.32
## Fit based upon off diagonal values = 0.97
## Measures of factor score adequacy
##                                                 PA1  PA2  PA3  PA4
## Correlation of (regression) scores with factors   0.96 0.93 0.92 0.92
## Multiple R square of scores with factors          0.92 0.87 0.85 0.84
## Minimum correlation of possible factor scores     0.84 0.75 0.71 0.68
```

The four factor model seems an improvement over the previous one. The variance explained has not decreased (it actually increases to 0.51), and each factor explains at minimum 16% of the variance. Where we did not improve was the Likelihood Chi Square test, which is now significant ($p = 0.04$). RMSR and RMSEA have not changed significantly are are still within desirable levels.

If we inspect the single items and their loadings on factors, however, the situation is improved significantly. All our developed items for a factor F load majorly on a single cluster PA found by the PAF analysis.

The only exception appears to be `F5_25`, which seems to not load on any factor whatsoever. After reviewing

the item, we decide to remove it and run the EFA again.

```
# we drop those columns that we identified after item analysis
study.efa.bis.df <-
  study.efa.df[,-which(names(study.efa.df) %in% c("F5_25"))]
four.factors.bis <-
  fa(study.efa.bis.df ,
     fm = "pa",
     nfactors = 4,
     rotate = "promax")
four.factors.bis
```

```
## Factor Analysis using method =  pa
## Call: fa(r = study.efa.bis.df, nfactors = 4, rotate = "promax", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PA1   PA2   PA3   PA4   h2   u2 com
## F1_1  -0.01  0.73 -0.12  0.02 0.57 0.43 1.1
## F1_2   0.05  0.74  0.02 -0.10 0.55 0.45 1.0
## F1_3  -0.01  0.68  0.10  0.00 0.46 0.54 1.0
## F1_5   0.03  0.69 -0.05  0.05 0.48 0.52 1.0
## F1_6  -0.05  0.75  0.00  0.10 0.58 0.42 1.0
## F1_7   0.01  0.73  0.06 -0.04 0.53 0.47 1.0
## F2_8  -0.04 -0.04 -0.03  0.66 0.43 0.57 1.0
## F2_9   0.00  0.04 -0.05  0.85 0.72 0.28 1.0
## F2_10  0.02  0.00  0.11  0.63 0.43 0.57 1.1
## F2_11  0.02  0.04 -0.02  0.65 0.43 0.57 1.0
## F3_12  0.77  0.00  0.01 -0.04 0.58 0.42 1.0
## F3_13  0.64  0.05 -0.01  0.05 0.41 0.59 1.0
## F3_14  0.73  0.02 -0.07  0.01 0.51 0.49 1.0
## F3_15  0.80  0.05  0.01 -0.06 0.62 0.38 1.0
## F3_16  0.70  0.07  0.03  0.05 0.51 0.49 1.0
## F3_17  0.82  0.00 -0.02  0.04 0.68 0.32 1.0
## F3_18  0.64 -0.18  0.07  0.08 0.52 0.48 1.2
## F3_19  0.79  0.03 -0.07 -0.02 0.60 0.40 1.0
## F3_20  0.76 -0.08  0.04 -0.10 0.59 0.41 1.1
## F5_26  0.04 -0.04  0.72 -0.05 0.53 0.47 1.0
## F5_27 -0.10  0.02  0.76 -0.09 0.57 0.43 1.1
## F5_28 -0.01 -0.01  0.67  0.12 0.47 0.53 1.1
## F5_29 -0.06 -0.01  0.66  0.05 0.43 0.57 1.0
## F5_30  0.04 -0.07  0.64  0.07 0.45 0.55 1.1
## F5_31  0.06  0.15  0.69 -0.11 0.49 0.51 1.2
##
##                        PA1  PA2  PA3  PA4
## SS loadings           4.94 3.20 2.92 2.07
## Proportion Var        0.20 0.13 0.12 0.08
## Cumulative Var        0.20 0.33 0.44 0.53
## Proportion Explained  0.38 0.24 0.22 0.16
## Cumulative Proportion 0.38 0.62 0.84 1.00
##
##  With factor correlations of
##       PA1   PA2   PA3   PA4
## PA1  1.00 -0.13  0.18  0.20
## PA2 -0.13  1.00 -0.09 -0.05
## PA3  0.18 -0.09  1.00  0.08
## PA4  0.20 -0.05  0.08  1.00
```

```
##
## Mean item complexity =  1
## Test of the hypothesis that 4 factors are sufficient.
##
## The degrees of freedom for the null model are  300  and the objective function was  12.97 with Chi S
## The degrees of freedom for the model are 206  and the objective function was  2.21
##
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.05
##
## The harmonic number of observations is  119 with the empirical chi square  108.99  with prob <  1
## The total number of observations was  119  with Likelihood Chi Square =  234.17  with prob <  0.087
##
## Tucker Lewis Index of factoring reliability =  0.962
## RMSEA index =  0.033  and the 90 % confidence intervals are  0 0.054
## BIC =  -750.33
## Fit based upon off diagonal values = 0.98
## Measures of factor score adequacy
##                                                 PA1  PA2  PA3  PA4
## Correlation of (regression) scores with factors   0.96 0.93 0.92 0.91
## Multiple R square of scores with factors          0.92 0.87 0.86 0.84
## Minimum correlation of possible factor scores     0.84 0.75 0.71 0.67
```

With the revised four factor model, the total variance explained rises to 0.53, which is another improvement.

It holds that all our developed items for a factor F load majorly on a single cluster PA found by the PAF analysis. In particular, our factors sorted for their importance (variance explained) are F3 (PA1), F1 (PA2), F5 (PA3), and F2 (PA4). We are satisfied with the resulting measurement instrument so far.

We improved on Likelihood Chi Square test, which is now back to non-significant ($p = 0.087$). RMSR and RMSEA have not changed significantly are are still within desirable levels. We decide to stop our EFA at this point and use the present form of our measurement instrument, with 4 factors and 26 items.

The resulting measurement instrument

We started with 31 items, which in our fictitious example were already a reduction, after item review, of an original measurement instrument with five factors. Thanks to item analysis and exploratory factor analysis, we were able to reduce the measurement instrument to 26 items (a 16% reduction) and four factors. All this with an improved confidence in what we can conclude from interpreting the results of employing the measurement instrument.

```
study.after.efa.df <- study.efa.bis.df
```

Item statistical properties

```
describe(study.after.efa.df)
```

```
##       vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
## F1_1     1 119 2.99 1.13      3    2.99 1.48   1   5     4  0.05    -0.68 0.10
## F1_2     2 119 2.87 0.97      3    2.89 1.48   1   5     4 -0.01    -0.37 0.09
## F1_3     3 119 2.87 1.11      3    2.87 1.48   1   5     4 -0.01    -0.60 0.10
## F1_5     4 119 2.97 1.05      3    2.96 1.48   1   5     4  0.09    -0.38 0.10
## F1_6     5 119 2.87 1.11      3    2.87 1.48   1   5     4  0.06    -0.70 0.10
## F1_7     6 119 2.94 1.06      3    2.93 1.48   1   5     4  0.12    -0.51 0.10
## F2_8     7 119 2.99 1.06      3    2.95 1.48   1   5     4  0.23    -0.60 0.10
## F2_9     8 119 3.06 1.01      3    3.03 1.48   1   5     4  0.08    -0.71 0.09
```

```
## F2_10    9 119 3.18 1.05      3    3.19 1.48   1   5     4 -0.14    -0.45 0.10
## F2_11   10 119 2.97 1.03      3    2.94 1.48   1   5     4  0.16    -0.81 0.09
## F3_12   11 119 3.13 1.06      3    3.15 1.48   1   5     4 -0.22    -0.38 0.10
## F3_13   12 119 3.20 1.07      3    3.21 1.48   1   5     4 -0.07    -0.57 0.10
## F3_14   13 119 3.13 1.05      3    3.11 1.48   1   5     4  0.01    -0.55 0.10
## F3_15   14 119 3.00 1.05      3    3.02 1.48   1   5     4 -0.09    -0.44 0.10
## F3_16   15 119 3.15 1.10      3    3.15 1.48   1   5     4  0.00    -0.64 0.10
## F3_17   16 119 3.13 1.17      3    3.15 1.48   1   5     4 -0.27    -0.76 0.11
## F3_18   17 119 3.18 1.10      3    3.18 1.48   1   5     4  0.03    -0.69 0.10
## F3_19   18 119 3.03 1.10      3    3.04 1.48   1   5     4 -0.07    -0.64 0.10
## F3_20   19 119 3.08 1.04      3    3.10 1.48   1   5     4 -0.19    -0.61 0.10
## F5_26   20 119 2.94 1.05      3    2.95 1.48   1   5     4  0.03    -0.63 0.10
## F5_27   21 119 3.01 1.04      3    3.02 1.48   1   5     4 -0.06    -0.60 0.10
## F5_28   22 119 3.05 1.10      3    3.05 1.48   1   5     4  0.09    -0.51 0.10
## F5_29   23 119 3.02 1.08      3    3.03 1.48   1   5     4 -0.07    -0.60 0.10
## F5_30   24 119 3.02 1.02      3    2.99 1.48   1   5     4  0.11    -0.51 0.09
## F5_31   25 119 2.96 0.96      3    2.93 1.48   1   5     4  0.20    -0.48 0.09
```

Possible minimum and maximum scores versus obtained minimum and maximum scores for F1, F2, F3, and F5, respectively.

```
f1.df <-
  study.after.efa.df[, grepl("F1_", names(study.after.efa.df))]
f2.df <-
  study.after.efa.df[, grepl("F2_", names(study.after.efa.df))]
f3.df <-
  study.after.efa.df[, grepl("F3_", names(study.after.efa.df))]
f5.df <-
  study.after.efa.df[, grepl("F5_", names(study.after.efa.df))]
print("F1")
```

```
## [1] "F1"
```

```
c(c(1, NCOL(f1.df) * 5), c(min(rowSums(f1.df)), max(rowSums(f1.df))))
```

```
## [1]  1 30  8 29
```

```
print("F2")
```

```
## [1] "F2"
```

```
c(c(1, NCOL(f2.df) * 5), c(min(rowSums(f2.df)), max(rowSums(f2.df))))
```

```
## [1]  1 20  5 20
```

```
print("F3")
```

```
## [1] "F3"
```

```
c(c(1, NCOL(f3.df) * 5), c(min(rowSums(f3.df)), max(rowSums(f3.df))))
```

```
## [1]  1 45  9 44
```

```
print("F5")
```

```
## [1] "F5"
```

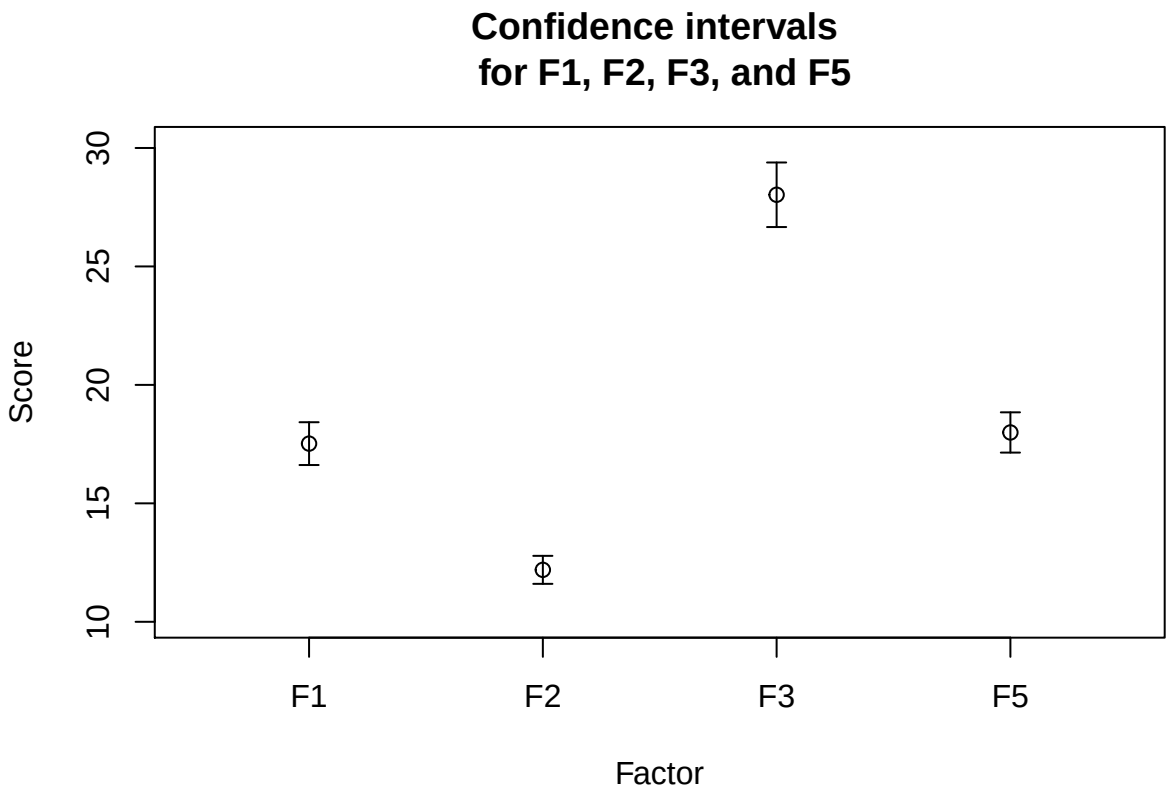
```
c(c(1, NCOL(f5.df) * 5), c(min(rowSums(f5.df)), max(rowSums(f5.df))))
```

```
## [1]  1 30  6 29
```

```r
factors.after.efa.df <-
  data.frame(
    F1 = rowSums(f1.df),
    F2 = rowSums(f2.df),
    F3 = rowSums(f3.df),
    F5 = rowSums(f5.df)
  )
describe(factors.after.efa.df)
```

```
##    vars   n  mean   sd median trimmed  mad min max range  skew kurtosis   se
## F1    1 119 17.52 4.97     17   17.43 4.45   8  29    21  0.20    -0.50 0.46
## F2    2 119 12.19 3.26     12   12.16 2.97   5  20    15  0.08    -0.34 0.30
## F3    3 119 28.03 7.51     29   28.14 7.41   9  44    35 -0.17    -0.49 0.69
## F5    4 119 17.99 4.68     18   17.97 4.45   6  29    23  0.01    -0.01 0.43
```

```r
error.bars(
  factors.after.efa.df,
  ylab = "Score",
  xlab = "Factor",
  main = "Confidence intervals \n for F1, F2, F3, and F5",
  eyes = F,
  sd = F
)
```




```r
print("Standardized scores available in factors.after.efa.standard.df but not shown here.")
```

```
## [1] "Standardized scores available in factors.after.efa.standard.df but not shown here."
```

```r
factors.after.efa.standard.df <-
  data.frame(
```

```
    F1 = scale(rowSums(f1.df)),
    F2 = scale(rowSums(f2.df)),
    F3 = scale(rowSums(f3.df)),
    F5 = scale(rowSums(f5.df))
  )
```

Reliability

Being our measurement instrument one for norm-referencing testing and not one for assessing skills, we perform a test-retest reliability assessment. Only a smaller number of participants (`reliability.sample.size`) accepted to take part to a second test.

```
reliablity.df.first <-
  tail(study.after.efa.df, reliability.sample.size)
reliablity.df.second <-
  tail(study.after.efa.df, reliability.sample.size)
f1.first <-
  rowSums(reliablity.df.first[, grepl("F1_" , names(reliablity.df.first))])
f2.first <-
  rowSums(reliablity.df.first[, grepl("F2_" , names(reliablity.df.first))])
f3.first <-
  rowSums(reliablity.df.first[, grepl("F3_" , names(reliablity.df.first))])
f5.first <-
  rowSums(reliablity.df.first[, grepl("F5_" , names(reliablity.df.first))])

f1.second <- round(jitter(unname(f1.first), amount = 3))
f2.second <- round(jitter(unname(f2.first), amount = 3))
f3.second <- round(jitter(unname(f3.first), amount = 3))
f5.second <- round(jitter(unname(f5.first), amount = 3))
```

Test-retest reliability for F1, F2, F3, and F5 are, respectively:

```
print("F1")
```

```
## [1] "F1"
```

```
cor(unname(f1.first), unname(f1.second))
```

```
## [1] 0.9462781
```

```
print("F2")
```

```
## [1] "F2"
```

```
cor(unname(f2.first), unname(f2.second))
```

```
## [1] 0.8248873
```

```
print("F3")
```

```
## [1] "F3"
```

```
cor(unname(f3.first), unname(f3.second))
```

```
## [1] 0.9751233
```

```
print("F5")
```

```
## [1] "F5"
```

```r
cor(unname(f5.first), unname(f5.second))
```

```
## [1] 0.9209
```

Validity

Our fictitious measurement instrument is novel and has no correspondence with existing measurement instruments. Therefore, any discussion on its validity depends on the instrument itself and the reasoning we can apply to it. Therefore, the present example can not elaborate on validity measurements.

Conclusion

The present document presented an introduction to psychometric evaluation with R for a behavioral software engineering audience. Through phases of item review and item analysis, exploratory factor analysis, item statistical properties, and reliability and validity evaluation, we show how we started with 31 items and 5 factors to represent our fictitious "individual perception styles of source code" to 26 items and four factors.

We could, therefore, offer the resulting measurement instrument in a paper describing its psychometric evaluation, for other researchers to use for further evaluation, and companies to use.