Repertory Grid Study for Expert Competencies

Nele Albers

07 May, 2024

Contents

This file is meant to reproduce our analysis for the repertory grid study conducted for obtaining the views of experts on the competencies for quitting smoking built by preparatory activities. The code here is partially based on the one provided here: https://towardsdatascience.com/exploratory-factor-analysis-in-r-e31b0015f224.

These files are required: "Data/all_expert_competency_ratings_transposed.csv"

And these files are created:

- "comp_3factors_df_loadings_experts.csv,"
- "comp_3factors_df_ratings_experts.csv," and
- labels and explanations mapped onto all three factors in the folder "Labels_and_Explanations_for_Factors."

Setup

Let's import the packages we need.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(nFactors)
```

```
## Loading required package: lattice

##
## Attaching package: 'nFactors'
```

```
## The following object is masked from 'package:lattice':
##
##     parallel
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

And we load our data.

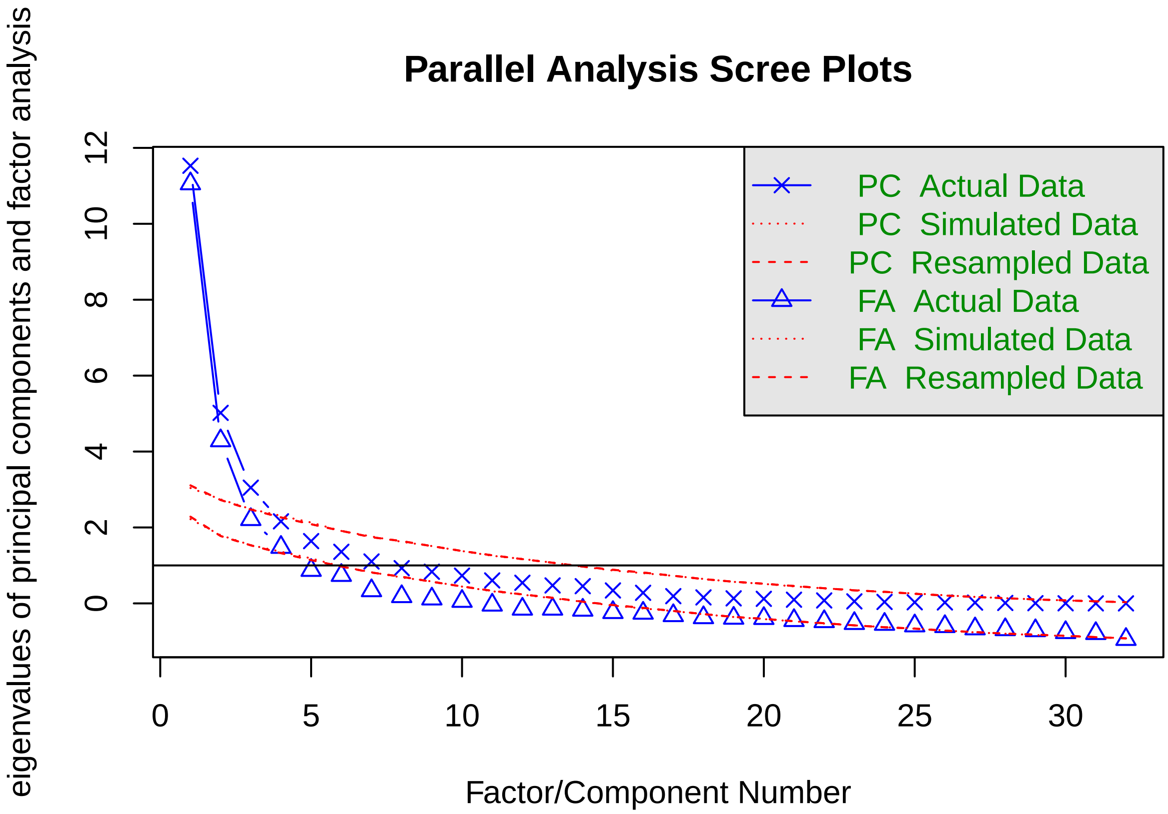
```
df_text = read.csv(file = "Data/all_expert_competency_ratings_transposed.csv")
# Remove last two rows since they contain the label and explanation
df <- head(df_text, - 2)
# Make character columns numeric
df <- df %>% mutate_if(is.character, as.numeric)
```

Determine possible numbers of factors

Let's first use parallel analysis. This suggests 4 factors.

```
parallel <- fa.parallel(df)
```

```
## Warning in log(det(m.inv.r)): NaNs produced

## In factor.stats, the correlation matrix is singular, an approximation is used

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## In factor.stats, the correlation matrix is singular, and we could not calculate
## the beta weights for factor score estimates

## In factor.stats: The factor scoring weights matrix is probably singular -- Factor score estimate resu
##  Try a different factor score estimation method

## Warning in max(R2, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## In factor.scores, the correlation matrix is singular, the pseudo inverse is  used

## I was unable to calculate the factor score weights, factor loadings used instead
```



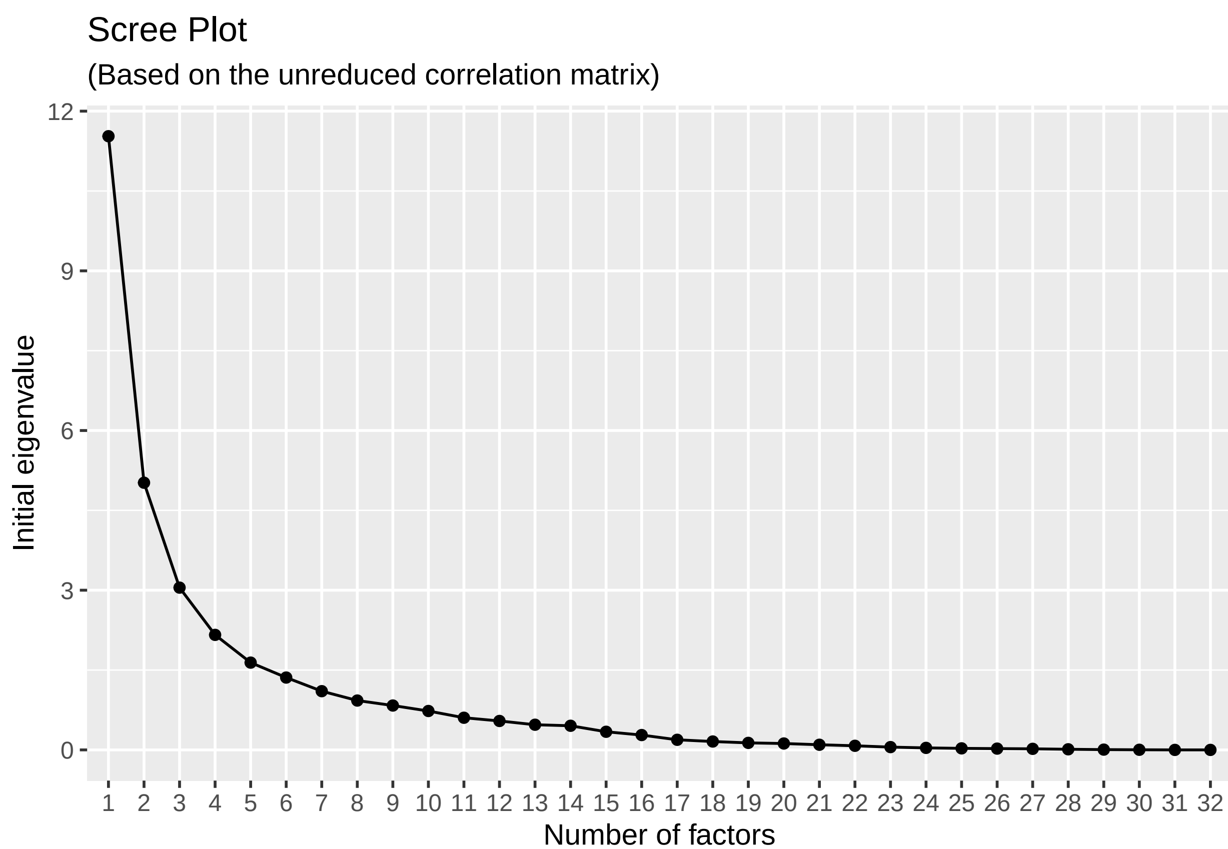

```
## Parallel analysis suggests that the number of factors =  3  and the number of components =  3
```

Next, we create a scree plot. Here the idea is to take as many factors as there are eigenvalues when there is a "big" drop in eigenvalues. Researchers may disagree where this big drop is. In our case, 2, 3, 4, and 5 seem to be possible factor numbers.

```
fafitfree <- fa(df, nfactors = ncol(df), rotate = "none")
```

```
## Warning in log(det(m.inv.r)): NaNs produced

## In factor.stats, the correlation matrix is singular, an approximation is used

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## In factor.stats, the correlation matrix is singular, and we could not calculate
## the beta weights for factor score estimates

## In factor.stats: The factor scoring weights matrix is probably singular -- Factor score estimate resu
##  Try a different factor score estimation method

## In factor.scores, the correlation matrix is singular, the pseudo inverse is  used

## I was unable to calculate the factor score weights, factor loadings used instead
```

```
n_factors <- length(fafitfree$e.values)
scree <- data.frame(Factor_n = as.factor(1:n_factors),
    Eigenvalue = fafitfree$e.values)
ggplot(scree, aes(x = Factor_n, y = Eigenvalue, group = 1)) +
    geom_point() + geom_line() + xlab("Number of factors") +
    ylab("Initial eigenvalue") + labs(title = "Scree Plot",
    subtitle = "(Based on the unreduced correlation matrix)")
```



Factor analysis

Now we run the factor analysis with the fa-method for 3 factors, which is the number of factors that we chose in the end.

```
num_factors = 3
fa.none <- fa(r=df,
 nfactors = num_factors,
 fm="minres", # type of factor analysis we want to use
 max.iter=100, # (50 is the default, but we have changed it to 100
 rotate="oblimin")
```

```
## Loading required namespace: GPArotation

## In factor.stats, the correlation matrix is singular, an approximation is used

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## In factor.stats, the correlation matrix is singular, and we could not calculate
## the beta weights for factor score estimates

## In factor.scores, the correlation matrix is singular, the pseudo inverse is  used

## I was unable to calculate the factor score weights, factor loadings used instead
```

```
print(fa.none)
```

```
## Factor Analysis using method =  minres
## Call: fa(r = df, nfactors = num_factors, rotate = "oblimin", max.iter = 100,
##     fm = "minres")
```

```
## Standardized loadings (pattern matrix) based upon correlation matrix
##       MR1   MR2   MR3   h2   u2 com
## X0   0.21 -0.89  0.07 0.89 0.11 1.1
## X1   0.74  0.36  0.25 0.59 0.41 1.7
## X2  -0.63  0.45  0.18 0.84 0.16 2.0
## X3   0.77  0.40  0.23 0.64 0.36 1.7
## X4  -0.63  0.45  0.18 0.84 0.16 2.0
## X5   0.78 -0.16 -0.06 0.71 0.29 1.1
## X6  -0.63  0.45  0.18 0.84 0.16 2.0
## X7   0.80 -0.15 -0.06 0.74 0.26 1.1
## X8   0.22  0.25 -0.83 0.78 0.22 1.3
## X9   0.41  0.81  0.09 0.71 0.29 1.5
## X10  0.15  0.75  0.24 0.64 0.36 1.3
## X11  0.58  0.17 -0.29 0.46 0.54 1.7
## X12 -0.11  0.64  0.32 0.65 0.35 1.5
## X13  0.10 -0.94  0.22 0.89 0.11 1.1
## X14 -0.55  0.20  0.34 0.61 0.39 2.0
## X15  0.64  0.27 -0.38 0.61 0.39 2.0
## X16 -0.03  0.43  0.39 0.41 0.59 2.0
## X17 -0.64  0.26  0.10 0.59 0.41 1.4
## X18 -0.10  0.18  0.27 0.15 0.85 2.0
## X19  0.56  0.17 -0.22 0.39 0.61 1.5
## X20  0.11  0.24 -0.49 0.27 0.73 1.6
## X21 -0.23  0.50 -0.18 0.33 0.67 1.7
## X22 -0.31  0.23 -0.29 0.20 0.80 2.8
## X23  0.18 -0.84  0.15 0.77 0.23 1.1
## X24 -0.22  0.45 -0.10 0.27 0.73 1.6
## X25  0.78 -0.17 -0.04 0.70 0.30 1.1
## X26 -0.04  0.18 -0.43 0.18 0.82 1.4
## X27 -0.26  0.56  0.26 0.59 0.41 1.9
## X28  0.12  0.37  0.63 0.59 0.41 1.7
## X29  0.85 -0.15  0.05 0.78 0.22 1.1
## X30 -0.20  0.55  0.13 0.44 0.56 1.4
## X31 -0.11  0.02 -0.58 0.32 0.68 1.1
##
##                        MR1  MR2  MR3
## SS loadings           7.79 7.23 3.42
## Proportion Var        0.24 0.23 0.11
## Cumulative Var        0.24 0.47 0.58
## Proportion Explained  0.42 0.39 0.19
## Cumulative Proportion 0.42 0.81 1.00
##
##  With factor correlations of
##       MR1   MR2   MR3
## MR1  1.00 -0.21 -0.21
## MR2 -0.21  1.00  0.20
## MR3 -0.21  0.20  1.00
##
## Mean item complexity =  1.6
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  496  and the objective function was  185.33 with Chi S
## The degrees of freedom for the model are 403  and the objective function was  134.5
##
```

```
## The root mean square of the residuals (RMSR) is  0.09
## The df corrected root mean square of the residuals is  0.1
##
## The harmonic number of observations is  44 with the empirical chi square  322.91  with prob <  1
## The total number of observations was  44  with Likelihood Chi Square =  3967.89  with prob <  0
##
## Tucker Lewis Index of factoring reliability =  0.117
## RMSEA index =  0.448  and the 90 % confidence intervals are  0.441 0.466
## BIC =  2442.86
## Fit based upon off diagonal values = 0.95
## Measures of factor score adequacy
##                                                    MR1   MR2   MR3
## Correlation of (regression) scores with factors    0.62  0.50  0.63
## Multiple R square of scores with factors           0.38  0.25  0.40
## Minimum correlation of possible factor scores     -0.24 -0.49 -0.20
```

Let's get the labels and explanations for the codes mapped to each factor. These labels and explanations were provided by the experts. We use a cutoff value of 0.4. We used these labels and explanations to examine the theoretical and practical plausibility of the factors resulting from using a certain number of factors as well as to find names for the resulting competencies.

```
cutoff = 0.4
weights = fa.none$weights
factor_labels_exp = list(list(c()), list(c()), list(c()),
    list(c()), list(c()), list(c()), list(c()), list(c()),
    list(c()), list(c()), list(c()))
for (code in 1:length(df)) {
    for (factor in 1:num_factors) {
        if (abs(weights[code, factor]) >= 0.4) {
            factor_labels_exp[[factor]] = append(factor_labels_exp[[factor]],
                paste(round(weights[code, factor],
                  2), df_text[45, code], "->", df_text[46,
                  code]))
        }
    }
}

# Print factors to text-files
for (factor in 1:num_factors) {

    # define file name
    sink(paste("Labels_and_Explanations_for_Factors/expert_comp_",
        num_factors, "factors", factor, ".txt"))

    # print my_list to file
    print(factor_labels_exp[[factor]])

    # close external connection to file
    sink()

}
```

Now we save the weights from the factor analysis and the ratings for the activities. We later need this to compute the contributions of the preparatory activities to the six expert competencies.

```r
df_ratings = t(t(df))  # Turn rating x activity into matrix

write.csv(fa.none$weights, file = paste0("comp_",
    num_factors, "factors_df_loadings_experts.csv"))
write.csv(df_ratings, file = paste0("comp_", num_factors,
    "factors_df_ratings_experts.csv"))
```