Repertory Grid Study for User Competencies and Preparatory Activity Clustering

Nele Albers

07 May, 2024

This file is meant to reproduce our analysis for the repertory grid study conducted for obtaining the views of smokers on the competencies for quitting smoking built by preparatory activities. Further, we compute the clusters of preparatory activities that were used for the effort and transition predictions.

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_competency_ratings_transposed.csv."

And these files are created: "comp_ 5 _activity_clusters_5_ratings.csv," and the labels and explanations mapped onto the five 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(factoextra)  # To visualize clustering output
```

```
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

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

```
## Loading required package: lattice

##
## Attaching package: 'nFactors'

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

```r
library(psych)
```

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

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

And we load our data.

```r
df_text = read.csv(file = "Data/all_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)
```

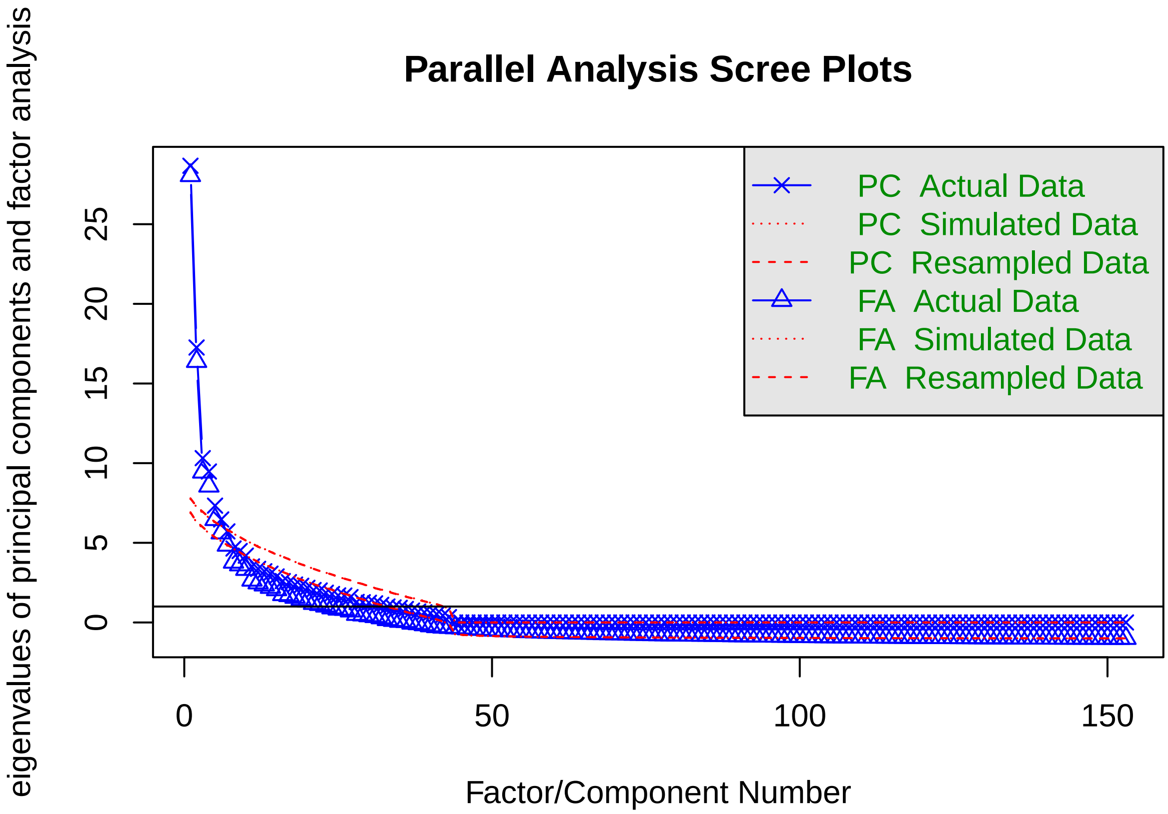
Determine possible numbers of factors

Let's first use parallel analysis. Parallel analysis proposes as many factors as there are eigenvalues based on the actual data that are strictly larger (i.e., not overlapping) with simulated data.

This suggests 6 factors.

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

```
## In smc, smcs < 0 were set to .0
## In smc, smcs < 0 were set to .0
## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done

## The determinant of the smoothed correlation was zero.
## This means the objective function is not defined.
## Chi square is based upon observed residuals.

## The determinant of the smoothed correlation was zero.
## This means the objective function is not defined for the null model either.
## The Chi square is thus based upon observed correlations.

## 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.

## 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 =  6  and the number of components =  6
```

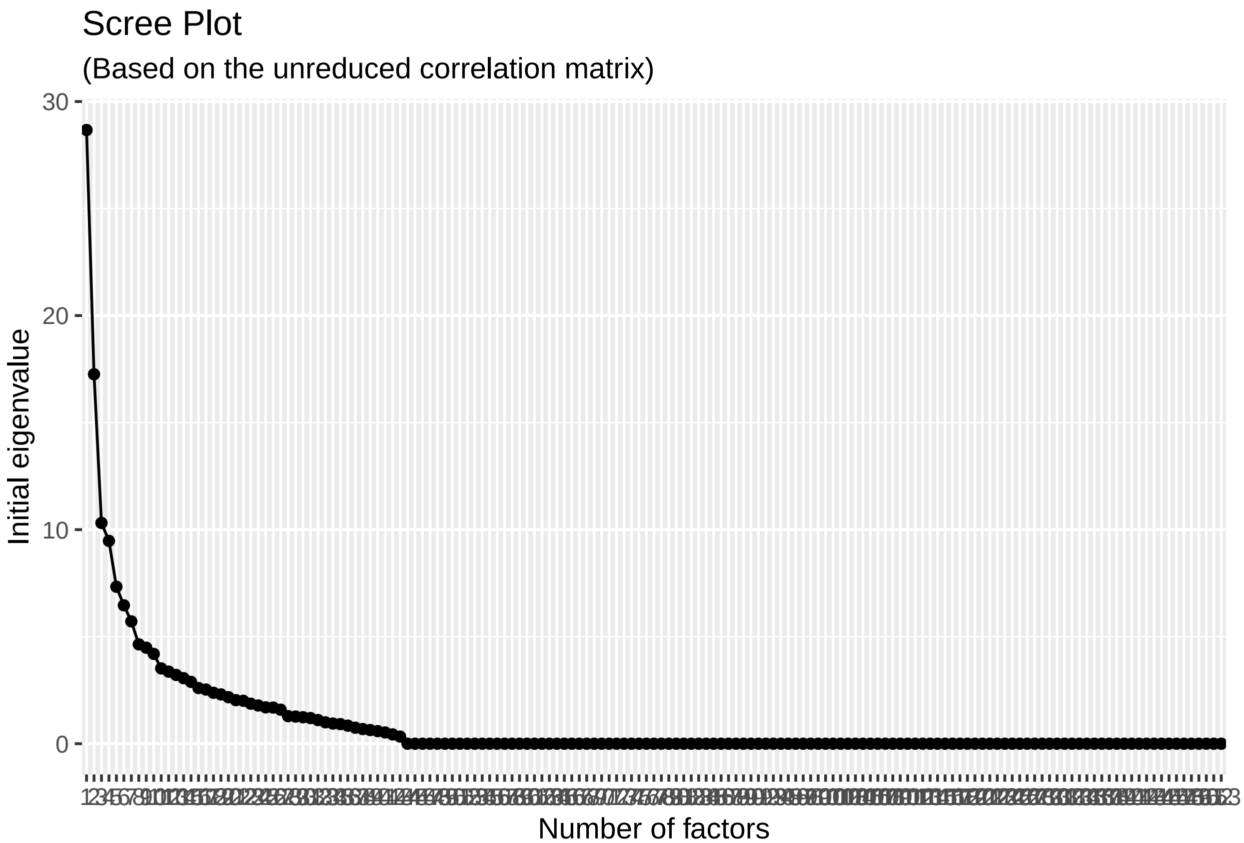
Next, we create a scree plot. Here the idea is to take as many factors as there are eigenvalues when there is a "big" drop in eigenvalues. Researchers may disagree where this big drop is. In our case, 3, 5, 8, or 11 factors seem to be possible.

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

```
## In smc, smcs < 0 were set to .0
## In smc, smcs < 0 were set to .0
## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done

## The determinant of the smoothed correlation was zero.
## This means the objective function is not defined.
## Chi square is based upon observed residuals.

## The determinant of the smoothed correlation was zero.
## This means the objective function is not defined for the null model either.
## The Chi square is thus based upon observed correlations.

## 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 5 factors, which is the number of factors that we chose in the end.

```
num_factors = 5
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")
```

```
## In smc, smcs < 0 were set to .0
## In smc, smcs < 0 were set to .0
## In smc, smcs < 0 were set to .0
```

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

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done
```

```
## The determinant of the smoothed correlation was zero.
## This means the objective function is not defined.
## Chi square is based upon observed residuals.
```

```
## The determinant of the smoothed correlation was zero.
## This means the objective function is not defined for the null model either.
## The Chi square is thus based upon observed correlations.
```

```
## 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.

## 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   MR4   MR5   h2    u2 com
## X0    0.39 -0.01  0.37 -0.01 -0.04 0.34 0.655 2.0
## X1   -0.34  0.23  0.11  0.27 -0.10 0.27 0.727 3.2
## X2    0.01 -0.23  0.60  0.04  0.25 0.48 0.524 1.7
## X3    0.23  0.22  0.22  0.01  0.14 0.22 0.779 3.7
## X4    0.24  0.51  0.47 -0.12 -0.20 0.63 0.366 2.9
## X5    0.12  0.22  0.28  0.11 -0.02 0.17 0.825 2.6
## X6    0.11 -0.04  0.18  0.79 -0.10 0.67 0.327 1.2
## X7   -0.08  0.06  0.59  0.12  0.14 0.40 0.604 1.3
## X8    0.16  0.77  0.09 -0.04  0.28 0.79 0.206 1.4
## X9    0.28  0.05  0.25  0.14  0.36 0.35 0.649 3.2
## X10  -0.54  0.11 -0.31  0.03 -0.05 0.47 0.530 1.7
## X11   0.51  0.32 -0.08  0.07  0.12 0.40 0.596 1.9
## X12   0.79  0.14  0.10  0.03 -0.03 0.70 0.298 1.1
## X13   0.21 -0.27  0.16  0.01 -0.03 0.14 0.856 2.6
## X14   0.77  0.00  0.16 -0.13  0.31 0.87 0.134 1.5
## X15  -0.05  0.25  0.02  0.40 -0.29 0.32 0.679 2.6
## X16   0.60  0.16  0.32  0.00  0.29 0.73 0.272 2.2
## X17  -0.08  0.38  0.14 -0.01  0.09 0.18 0.816 1.5
## X18  -0.05 -0.03  0.56  0.14  0.26 0.40 0.596 1.6
## X19  -0.24  0.20  0.31  0.19  0.11 0.23 0.774 3.8
## X20  -0.09 -0.02 -0.12  0.85  0.07 0.76 0.243 1.1
## X21   0.11  0.30  0.31  0.09 -0.51 0.43 0.571 2.6
## X22   0.31  0.40 -0.44 -0.11  0.08 0.41 0.592 3.0
## X23  -0.39  0.38  0.52  0.15 -0.19 0.55 0.447 3.3
## X24  -0.39  0.65 -0.11 -0.03 -0.25 0.62 0.382 2.1
## X25  -0.02  0.09 -0.06  0.78  0.03 0.62 0.376 1.0
## X26  -0.09  0.15  0.61 -0.07  0.01 0.40 0.601 1.2
## X27   0.12  0.15 -0.43 -0.22  0.45 0.44 0.558 2.8
## X28   0.78  0.04 -0.04  0.02  0.26 0.71 0.286 1.2
## X29   0.00  0.25  0.08  0.49  0.15 0.35 0.652 1.8
## X30  -0.05  0.53  0.03  0.09 -0.08 0.30 0.701 1.1
## X31   0.05  0.63 -0.13  0.44 -0.36 0.73 0.272 2.6
## X32   0.05  0.09  0.36  0.01  0.27 0.25 0.750 2.1
## X33   0.39  0.37  0.26 -0.12  0.23 0.55 0.445 3.7
## X34   0.30  0.09  0.15  0.67 -0.18 0.58 0.420 1.7
## X35  -0.31  0.60  0.38  0.01 -0.07 0.56 0.436 2.3
## X36   0.40  0.57  0.04  0.02  0.29 0.67 0.330 2.3
## X37   0.15  0.14  0.11  0.79 -0.10 0.69 0.313 1.2
## X38   0.23  0.52  0.08 -0.05  0.22 0.45 0.552 1.9
## X39   0.04  0.31  0.18  0.23  0.35 0.35 0.654 3.4
## X40   0.34 -0.34  0.01 -0.31  0.15 0.38 0.620 3.3
```

```
## X41   0.76  0.08  0.20 -0.07  0.17 0.78 0.216 1.3
## X42   0.55  0.12  0.08 -0.17 -0.11 0.38 0.618 1.4
## X43  -0.53  0.18  0.39 -0.14  0.39 0.54 0.456 3.2
## X44   0.02 -0.44  0.01  0.07 -0.50 0.49 0.510 2.0
## X45   0.35 -0.19  0.53 -0.24 -0.11 0.56 0.435 2.6
## X46   0.14  0.60  0.19  0.13  0.33 0.63 0.369 2.0
## X47   0.31  0.09 -0.04  0.09 -0.07 0.11 0.893 1.5
## X48  -0.14  0.20  0.26  0.05  0.47 0.37 0.627 2.2
## X49   0.58 -0.18 -0.32  0.19  0.03 0.41 0.587 2.0
## X50   0.06  0.18 -0.49  0.02  0.27 0.32 0.681 1.9
## X51   0.09 -0.16  0.21  0.18  0.23 0.16 0.843 4.0
## X52   0.34  0.06 -0.06  0.14 -0.20 0.16 0.842 2.2
## X53   0.22  0.63 -0.13 -0.04  0.33 0.62 0.378 1.9
## X54   0.77  0.17 -0.20  0.10  0.11 0.65 0.352 1.3
## X55   0.12 -0.04  0.27 -0.26 -0.01 0.18 0.822 2.4
## X56  -0.14 -0.05 -0.30  0.63  0.09 0.54 0.459 1.6
## X57  -0.75  0.09 -0.15  0.15 -0.29 0.81 0.186 1.5
## X58   0.00  0.12  0.11  0.40 -0.17 0.22 0.779 1.7
## X59   0.19  0.68  0.02 -0.11  0.11 0.57 0.435 1.3
## X60  -0.62  0.22  0.29 -0.13 -0.11 0.45 0.552 1.9
## X61   0.54  0.55  0.10 -0.28  0.05 0.77 0.233 2.6
## X62   0.45  0.37 -0.21 -0.17 -0.14 0.39 0.614 3.0
## X63   0.13  0.02 -0.09  0.19 -0.23 0.11 0.891 2.9
## X64  -0.09  0.24  0.55  0.16  0.25 0.49 0.510 2.1
## X65   0.45  0.08 -0.05  0.02  0.28 0.31 0.688 1.8
## X66   0.28  0.32 -0.06  0.16  0.18 0.27 0.733 3.2
## X67   0.25 -0.21 -0.08 -0.28 -0.24 0.25 0.748 4.0
## X68   0.18  0.27 -0.31 -0.01  0.30 0.28 0.722 3.6
## X69   0.13 -0.68  0.03 -0.13 -0.08 0.51 0.486 1.2
## X70   0.20 -0.13  0.59 -0.03  0.11 0.46 0.535 1.4
## X71   0.25 -0.86  0.00 -0.13 -0.03 0.82 0.176 1.2
## X72  -0.08  0.34 -0.05  0.06 -0.04 0.13 0.874 1.3
## X73   0.47  0.14 -0.36 -0.02  0.05 0.31 0.690 2.1
## X74   0.21  0.26  0.42 -0.02  0.19 0.42 0.581 2.7
## X75  -0.06  0.88 -0.20 -0.03 -0.14 0.79 0.208 1.2
## X76   0.92  0.01  0.14  0.19 -0.25 0.92 0.079 1.3
## X77  -0.08  0.51  0.25  0.21  0.08 0.41 0.587 2.0
## X78   0.17 -0.44  0.01 -0.12 -0.11 0.26 0.742 1.6
## X79   0.29 -0.09 -0.50 -0.29 -0.17 0.41 0.586 2.7
## X80   0.06 -0.09  0.28  0.10 -0.02 0.10 0.898 1.6
## X81   0.27  0.09 -0.10  0.02 -0.48 0.28 0.720 1.7
## X82   0.58  0.51 -0.02 -0.23  0.09 0.72 0.275 2.4
## X83   0.49  0.38 -0.21  0.14  0.02 0.42 0.582 2.5
## X84  -0.69  0.14  0.23 -0.01  0.15 0.49 0.510 1.4
## X85   0.40 -0.02  0.39 -0.02 -0.01 0.37 0.631 2.0
## X86  -0.28  0.01 -0.06  0.13 -0.71 0.67 0.326 1.4
## X87  -0.61  0.14 -0.21 -0.11 -0.20 0.54 0.465 1.7
## X88  -0.21  0.02  0.07 -0.43 -0.39 0.35 0.647 2.5
## X89   0.00  0.00  0.22  0.67 -0.20 0.53 0.467 1.4
## X90   0.39 -0.16  0.46  0.06  0.13 0.47 0.526 2.5
## X91   0.49 -0.11  0.52 -0.18  0.03 0.67 0.328 2.3
## X92  -0.42 -0.07 -0.21  0.04  0.42 0.38 0.619 2.5
## X93  -0.22  0.09  0.64  0.08  0.03 0.43 0.571 1.3
## X94  -0.73  0.11 -0.08 -0.10 -0.08 0.58 0.423 1.1
```

```
## X95   0.17  0.02  0.18 -0.03  0.32 0.21 0.792 2.2
## X96  -0.54  0.11 -0.31  0.03 -0.05 0.47 0.530 1.7
## X97   0.67  0.04  0.14 -0.14  0.03 0.57 0.433 1.2
## X98   0.56  0.30 -0.09  0.07  0.12 0.45 0.554 1.8
## X99   0.69  0.00  0.21 -0.10 -0.28 0.61 0.387 1.6
## X100  0.29 -0.30  0.17 -0.01  0.10 0.22 0.775 2.8
## X101 -0.75  0.09 -0.17  0.16 -0.28 0.82 0.177 1.5
## X102  0.17 -0.09 -0.08 -0.15  0.22 0.12 0.882 3.5
## X103 -0.63 -0.09  0.00  0.15 -0.11 0.48 0.520 1.2
## X104  0.15  0.16  0.09  0.39 -0.10 0.22 0.782 2.0
## X105  0.51  0.35  0.08 -0.14  0.14 0.51 0.491 2.2
## X106  0.05  0.68 -0.07  0.10  0.16 0.53 0.471 1.2
## X107 -0.31  0.82 -0.01  0.05 -0.05 0.75 0.253 1.3
## X108  0.57  0.05  0.27 -0.11  0.24 0.59 0.412 1.9
## X109  0.58 -0.05 -0.20  0.04 -0.17 0.35 0.651 1.5
## X110  0.03  0.02  0.14  0.20 -0.80 0.68 0.316 1.2
## X111 -0.40  0.11 -0.06 -0.39 -0.28 0.38 0.625 3.0
## X112 -0.66  0.22 -0.16 -0.20 -0.08 0.56 0.444 1.6
## X113 -0.26 -0.01  0.29 -0.01  0.01 0.12 0.880 2.0
## X114  0.78  0.05  0.09 -0.10 -0.11 0.66 0.344 1.1
## X115  0.43  0.07  0.40 -0.30 -0.02 0.53 0.465 2.9
## X116  0.12 -0.02  0.77  0.12 -0.20 0.67 0.333 1.2
## X117 -0.70  0.09  0.18  0.04  0.04 0.48 0.521 1.2
## X118 -0.26  0.12 -0.25 -0.08 -0.34 0.30 0.697 3.3
## X119  0.45  0.42  0.07 -0.01 -0.01 0.42 0.577 2.0
## X120  0.13  0.51  0.03  0.08  0.29 0.42 0.580 1.8
## X121  0.08 -0.26  0.27  0.68  0.02 0.57 0.430 1.6
## X122  0.08 -0.31 -0.16  0.09 -0.56 0.49 0.514 1.8
## X123  0.24  0.51  0.47 -0.12 -0.20 0.63 0.366 2.9
## X124 -0.23  0.06 -0.03  0.37  0.61 0.54 0.464 2.0
## X125 -0.34  0.21 -0.50 -0.07  0.36 0.54 0.456 3.1
## X126 -0.53  0.05  0.19 -0.41 -0.04 0.41 0.592 2.2
## X127  0.06  0.88  0.05 -0.15 -0.10 0.79 0.212 1.1
## X128 -0.06  0.36 -0.10 -0.21 -0.07 0.17 0.829 1.9
## X129 -0.86  0.09  0.01  0.07  0.07 0.74 0.260 1.0
## X130  0.46 -0.50 -0.02 -0.17  0.08 0.50 0.503 2.3
## X131 -0.84  0.01  0.22 -0.03  0.31 0.72 0.277 1.4
## X132  0.10  0.17  0.46  0.07 -0.19 0.29 0.709 1.8
## X133  0.02 -0.08  0.52 -0.44 -0.17 0.49 0.513 2.2
## X134  0.21  0.47  0.19 -0.02  0.15 0.39 0.614 2.0
## X135 -0.30 -0.35  0.11  0.39  0.18 0.38 0.623 3.5
## X136  0.13  0.13  0.00  0.15 -0.14 0.07 0.930 4.0
## X137 -0.12 -0.49 -0.01 -0.24 -0.53 0.66 0.336 2.5
## X138 -0.54  0.16  0.11 -0.07  0.22 0.33 0.670 1.7
## X139 -0.07 -0.26 -0.20  0.56  0.09 0.43 0.570 1.8
## X140  0.11 -0.33 -0.16  0.24 -0.36 0.36 0.645 3.4
## X141  0.32  0.77 -0.08  0.08 -0.06 0.71 0.285 1.4
## X142  0.19 -0.70  0.04  0.03 -0.20 0.58 0.422 1.3
## X143  0.20 -0.13  0.59 -0.03  0.11 0.46 0.535 1.4
## X144 -0.52  0.02  0.08  0.09  0.02 0.28 0.722 1.1
## X145  0.35  0.01  0.13 -0.01  0.06 0.17 0.831 1.4
## X146  0.85  0.09  0.12  0.01 -0.21 0.79 0.210 1.2
## X147 -0.12  0.04 -0.06  0.58  0.10 0.38 0.623 1.2
## X148  0.32 -0.47  0.46 -0.09  0.30 0.68 0.319 3.6
```

```
## X149  0.08 -0.10  0.32  0.08  0.02 0.13 0.873 1.5
## X150 -0.54 -0.06 -0.01 -0.36 -0.07 0.41 0.593 1.8
## X151  0.02 -0.12  0.28 -0.03  0.15 0.12 0.877 2.0
## X152  0.32  0.46  0.25  0.14  0.13 0.50 0.503 2.8
##
##                         MR1   MR2   MR3  MR4  MR5
## SS loadings           24.28 16.57 11.85 9.30 8.68
## Proportion Var         0.16  0.11  0.08 0.06 0.06
## Cumulative Var         0.16  0.27  0.34 0.41 0.46
## Proportion Explained   0.34  0.23  0.17 0.13 0.12
## Cumulative Proportion  0.34  0.58  0.75 0.88 1.00
##
##  With factor correlations of
##       MR1  MR2   MR3   MR4   MR5
## MR1  1.00 0.06  0.20 -0.10  0.11
## MR2  0.06 1.00  0.07  0.06  0.11
## MR3  0.20 0.07  1.00 -0.02  0.09
## MR4 -0.10 0.06 -0.02  1.00 -0.06
## MR5  0.11 0.11  0.09 -0.06  1.00
##
## Mean item complexity =  2.1
## Test of the hypothesis that 5 factors are sufficient.
##
## The degrees of freedom for the null model are  11628  and the objective function was  1620.34 with Ch
## The degrees of freedom for the model are 10873  and the objective function was  252.09
##
## 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  8887.52  with prob <  1
## The total number of observations was  44  with Likelihood Chi Square =  0  with prob <  1
##
## Tucker Lewis Index of factoring reliability =  0.524
## RMSEA index =  0  and the 90 % confidence intervals are  0 0
## BIC =  -44212.58
## Fit based upon off diagonal values = 0.86
```

Let's get the labels and explanations for the codes mapped to each factor. These labels and explanations were provided by smokers. 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_comp = 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_comp[[factor]] = append(factor_labels_comp[[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/comp_users_",
        num_factors, "factors", factor, ".txt"))

    # print to file
    print(factor_labels_comp[[factor]])

    # close external connection to file
    sink()

}
```

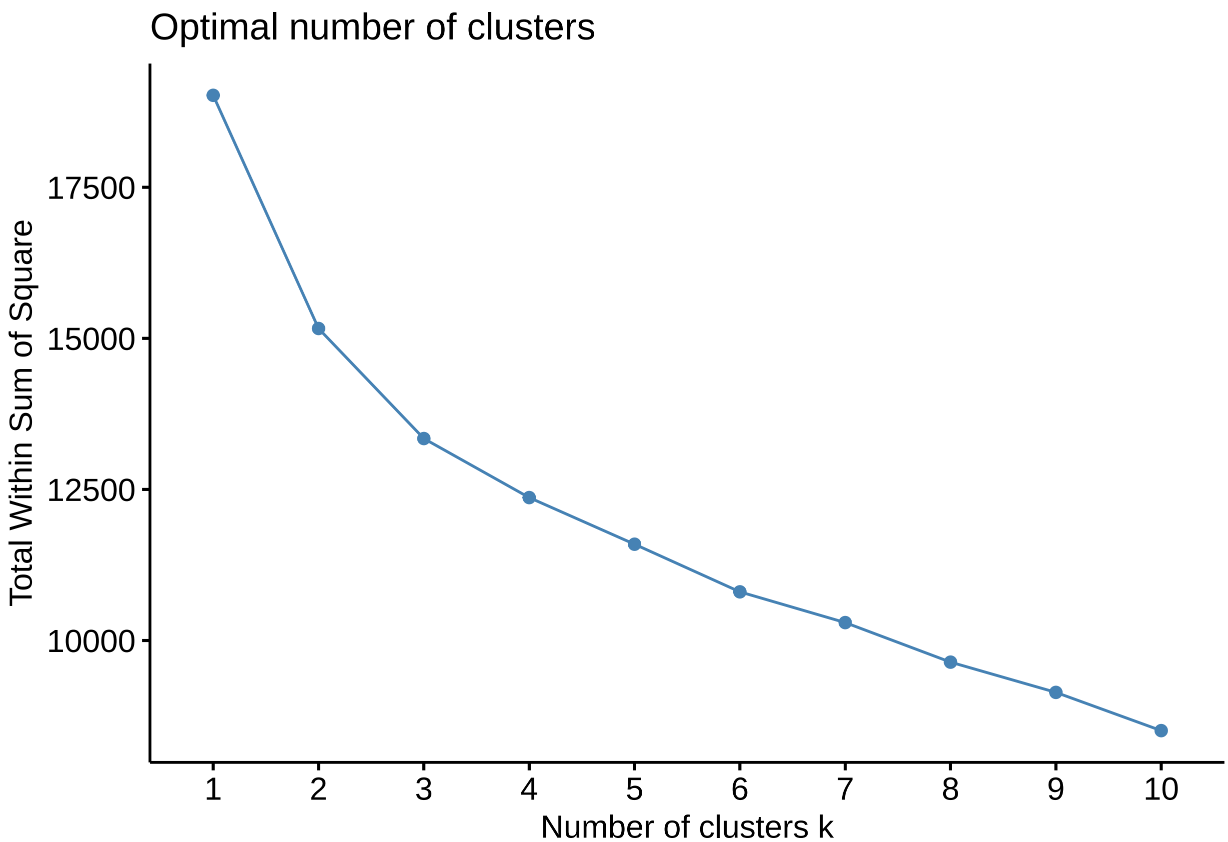
Cluster preparatory activities

Now we want to cluster the preparatory activities based on how they are perceived by users.

Determine possible numbers of clusters

First, we use the elbow method to determine the number of clusters for k-means. There appears to be an elbow at 3 clusters and again at 6 clusters.

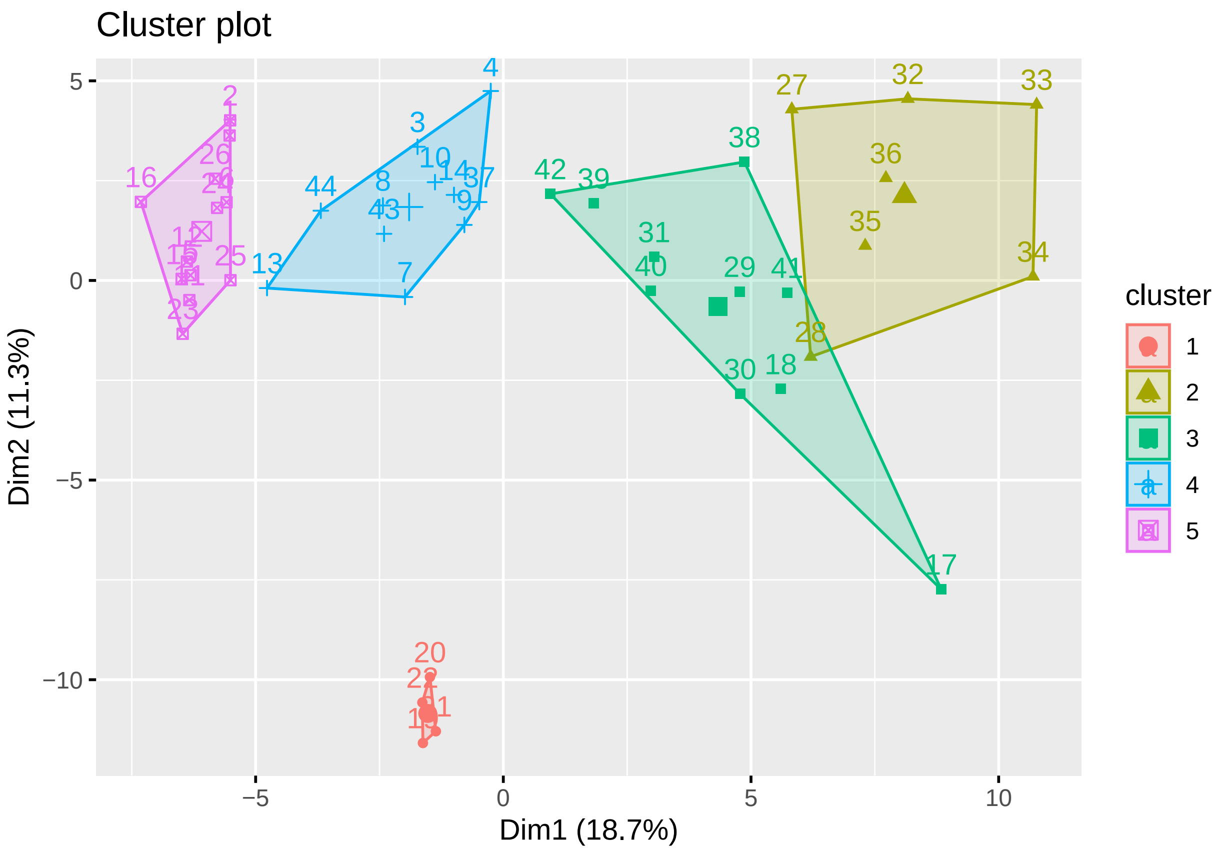
```
df_ratings = t(t(df))  # Turn rating x activity into matrix
fviz_nbclust(df_ratings, kmeans, method = "wss")
```



Clustering

Let's now perform k-means clustering on the activity ratings with 5 clusters. While the elbow-method points to only 3 clusters, we want to allow for at least as many clusters as we have factors for the user competencies. Using more than 5 clusters (e.g., 6, which is also a small elbow) leads to too few activities in some clusters (i.e., less than 4).

```
set.seed(18)  # For reproducibility
num_clusters = 5
km <- kmeans(df_ratings, centers = num_clusters,
    nstart = 100)

clusters <- data.frame(km$cluster)
fviz_cluster(km, data = df_ratings)
```




```
write.csv(clusters, paste("comp_", num_factors, "_activity_clusters_5_ratings.csv"))
```