Aalto University
School of Science
Degree programme in Engineering Physics and Mathematics

Robustness of factor analysis in analysis of data with discrete variables

Student Project
26.3.2012

Juha Törmänen

Contents

1 Introduction

Exploratory factor analysis (EFA) and confirmatory factor analysis (CFA) are useful and powerful tools for discovering underlying structure in data sets (Thompson 2004). These procedures are often used to formulate a small set of latent variables which is able to explain the results of a much larger questionnaire or a test, either by applying both methods in conjunction or by applying either of them separately (Hurley et al. 1997). One use for EFA and CFA methods is in discovering the latent factors of a set of self-response questionnaire answers. This study evaluates how two common features of self-response questionnaires, high error variance and discrete integer-valued date, affect these methods in relation to an experimental study with similar parameters.

EFA methods are used to discover latent variables that cause the observed variables to covary in the data set. In general, the methods extract a small set of factors from the data that can be used to explain the results of the observed variables well. There are several factor extraction methods, and they are often accompanied by a rotation of the resulting data matrix to simplify and clarify the resulting structure (Costello and Osborne 2005). The number of factors to extract is decided beforehand with the aid of methods such as Velicer's minimum average partial method (MAP) or Horn's parallel analysis (PA) (Hayton et al. 2004).

CFA methods are used to test already existing assumptions about the underlying latent variable structure of the data. Such assumptions can be either hypothesized by the testers or be the result of an exploratory procedure such as EFA. In CFA, these assumptions are formulated into a structural equation model (Bollen 1989) and the quality and fit of the model is used to evaluate how well the assumptions fit to the data.

In a self-response questionnaire, the inputs can be presented as Likert-scale multiple choice items (Likert 1932) where the participant may pick his or her answer from options such as "never"; "very seldom"; "seldom"; "sometimes"; "often"; "very often", and "always". The answers to such an item are discrete and have an ordinal scale — the difference difference between "never" and "very seldom" is unlikely to be exactly the same as the difference between "sometimes" and "often". When such answers are used in factor analysis, they are treated as continuous variables with the answers converted to integer values, such as 0–6. This can possibly cause a mismatch with the assumptions of the factor analytic methods. Most notably, CFA methods expect the data to be multivariate normal distributed (Bentler and Chou 1987).

Several solutions have been proposed for solving the problem of using data with an ordinal scale with factor analytic methods. Muthen and Kaplan (1985) have found that using some of the less common CFA estimators such as the asymptotically distribution-free estimator may help to deal with data with a categorical scale. As another solution, the data covariance matrix used in the factor analytic methods may be replaced by a polychoric correlation matrix, which attempts to model the correlations between the underlying continuous indicators of the variables (Bollen 1989, pp. 441-445). Yet another solution is to rely on a large number of observations and use the standard methods that are widely implemented in EFA and CFA software such as IBM SPSS Statistics and IBM SPSS Amos.

The experimental study that this paper relates to was conducted in the Systems Analysis Laboratory of Aalto University by Törmänen (2012). The study was based on a 76-item inventory with 7-point Likert scale answers and discovered a eight-factor latent variable structure underlying the items. The resulting structure was found to be fitting when considering face and content validity, but the used factorial validity indicators indicated a lack of fit of the data to the model. The RMSEA index of the resulting model was 0.061 and the CFI index 0.808. In a good model, the RMSEA index is expected to be under 0.05 and the CFI index over 0.95 (Schermelleh-Engel et al. 2003).

In this study, the objective is to simulate how a similar latent factor structure would perform if the underlying data was generated from an ideal multivariate normal distribution with the same item error variances, factor loadings and factor variances and covariances as the actual data. The data is generated from normal and multivariate normal distributions and converted to a 0–6 point interval scale, and a similar set of methods is used to analyze the data as was used in the experimental study. The effects of the conversion to integer scale are evaluated with regard to three questions:

1. **Number of factors to be extracted:** Are the methods used to estimate the number of factors to be extracted robust?
2. **Exploratory factoring:** Are the EFA methods able to reproduce the correct model structure?
3. **Confirmatory validation:** How are the model fit indices affected by the discrete data?

The results are meant to aid the interpretation of the original study, and to create a reference point to which similar studies can be compared to.

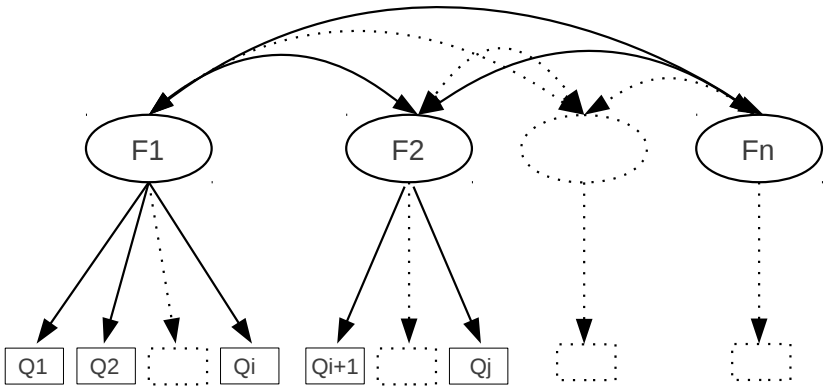

Figure 1: Factor model diagram for a simple factor structure. Dotted lines represent any number of items at the same level. Straight lines are loadings and curved lines are covariances.

2 Methods

2.1 Data generation

In an ideal structural equation model with a simple structure, the observed variables are related only to the single latent variable (factor) they load to and to their variable-specific error variance. The latent variables are assumed to be multivariate normal distributed with respect to each other, and the observed variable scores s_i are assumed to be generated with the formula

$$s_i = \lambda_i F_i + \epsilon(i) \tag{1}$$

where λ_i is the *factor loading* of item i and F_i is the *factor score* of the factor the item loads to. The error variances $\epsilon(i)$ are independent and normally distributed. This structure is visualized in Figure 1.

In this simulation, each latent variable is assumed to have the same amount of variance and all the covariances between the latent variables are assumed to be equal. All factor loadings to observed variables are equal in size and set to $\lambda_i = 1$. Thus, the data is generated with the following heuristic.

1. For each participant, generate eight factor scores from a multivariate normal distribution with a mean value of μ where the variance of each factor is σ_f^2 and the covariance between two scores is γ.
2. Generate an equal number of items for each factor, where the score s_i for item i is $s_i = \mu + e(i)$, where $e(i) \sim N(0, \sigma_e^2)$
3. Round the scores to integer values in the range 0–6.

The parameters are defined from the data described by Törmänen (2012). They are calculated from a structural equation model of the actual data where each factor loading was set to +1 or -1 with items phrased positively or negatively. Factor means, variances, covariances, and error variances are estimated from the arithmetic means of all available item statistics. These estimates are 0.417 for factor variances, 0.287 for factor covariances, 0.598 for error variances and 4.22 for factor means. As the factor and error variances are the only sources of variation in the data and their variances are not dependent on each other, these numbers can be interpreted to mean that $0.417/(0.417 + 0.598) \approx 41\%$ of item variance is explained by the factor variances and the rest 59% are explained by the item-specific error variances.

2.2 Estimation of the number of factors

The number of factors to be extracted from the data is estimated with two methods, Velicer's MAP (Velicer 1976) and Horn's PA (Horn 1965). Hayton et al. (2004) note that MAP often underestimates and PA slightly overestimates the number of factors to extract. If the methods estimate there to be less factors than there are in actuality, it will cause all further factor analytic procedures to produce erroneous results.

The robustness of the MAP and PA methods is simulated with data with 300 samples of answers with eight factors and nine items in each factors — that is, 72 items in total. The simulation is repeated one hundred times and the number of resulting factors for both MAP and PA is noted for each repetition.

2.3 Exploratory factor analysis

EFA is calculated with the principal axis method and oblimin rotation with the number of factors estimated by the PA method presented in the earlier section. These methods are the same that were used in the experimental comparison study. The results are further developed to a simple structure where each item loads onto only one factor. Items that load onto only one factor with a loading of at least 0.32 as recommended by Costello and Osborne (2005) are included in the model. If an item loads onto several factors with a stronger loading, or does not have any strong loadings at all, it is dropped from the model.

This EFA process is repeated one hundred times with a new simulated data set every time. The robustness of the method is evaluated by comparing the resulting factoring with the real factors. The results are expected to differ from the true structure in two ways:

1. **Number of dropped items.** Some items may have none or too many high loadings, which will cause it to be dropped from the model.
2. **Number of split factors.** In some cases, items belonging to the same latent variable may factor into different factors.

If either items are dropped or factors are split, or if the PA estimate of the number of factors is wrong, the resulting output factoring won't match the real latent variable structure perfectly.

2.4 Model fit indices

The effect of discrete integer data on the CFA model fit indices is evaluated by simulating a set of 900 participants with eight factors and seven items in each factor (56 items in total). The CFA procedure is done by fitting a structural equation model (Bollen 1989) similar to the one illustrated in Figure 1 to the data. The used model is the exact same that is used to generate the data; thus, the focus in this part of the study is only to study the effect of discrete variables on the model fit indices.

The model fit is evaluated in three different ways. The χ^2 test evaluates directly how well the observed covariance matrix fits the covariance matrix implied by the structural equation model (Schermelleh-Engel et al. 2003, pp. 31-32). The Root Mean Square Error of Approximation (RMSEA) measures a null hypothesis of a "close fit" in the population (Schermelleh-Engel

et al. 2003, p. 36). RMSEA is usually considered good if the index is below 0.05, and adequate if it is between 0.05 and 0.08. The Comparative Fit Index (CFI) measures the relative fit of the model compared to the independence model, where all variables would be assumed uncorrelated. CFI is considered to have an acceptable fit when its value is above 0.95 (Schermelleh-Engel et al. 2003, p. 41).

The results are compared to the recommended values for a good quality structural equation model and also to the same numbers in the Törmänen (2012) study.

Table 1: Comparison of the original parameters and the average statistics for the simulated parameters

parameter		original data	simulated data
factor score mean	μ	4.22	4.20
factor score variance	σ_f^2	0.417	0.411
factor score covariance	γ	0.287	0.282
item error variance	σ_e^2	0.598	0.629

3 Results

3.1 Descriptives

The data is generated with R 2.12.2 (R Development Core Team 2011) with the parameters defined in the Methods section of this study. The multivariate normal factor scores are generated with the version 7.3-11 of the MASS package of R (Venables and Ripley 2002). The source code used for generating the data is shown in Appendix A.

The quality of the simulation data is evaluated by generating an answer set of 1600 simulated participants with eight factors and nine items per factor and comparing the resulting descriptive statistics with those of the given parameters. The numbers are shown in Table 1. The numbers are very close to actual results, even though the multivariate normal answers have been rounded to the nearest integer. These statistics imply that the simulated data matches the original distribution closely.

A typical histogram of the answers to a generated item is shown in Figure 2. The generated answer distribution is bell curve shaped, with a part of the higher end of the bell curve being cut off by the maximum answer of 6. The items fail the Shapiro-Wilk test of normality (Shapiro and Wilk 1965), with the null hypothesis of normality rejected with p values between 10^{-11} and 10^{-13}.

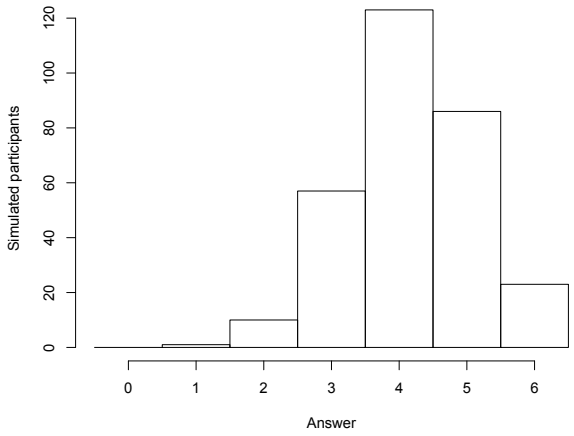

Figure 2: A sample histogram of simulated item answers for 300 participants

Table 2: Distribution of the estimations of the number of factors to extract

method	6 factors	7 factors	8 factors
Velicer's MAP	11	28	61
Horn's parallel analysis	2	9	89

100 simulation runs conducted.

3.2 Number of factors

Simulation: 100 repetitions, 300 samples, eight factors with nine items each

The R implementations of Horn's PA and Velicer's MAP (Revelle 2011) are used to estimate the robustness of the methods in estimating the correct number of factors from the simulated data. Figure 3 shows the scree plot of a simulation run. Based on the scree plot, the simulated data contains a single very strong component and after that, many weak components. There is a small but visible step in eigenvalues after the first eight components; thus, an experienced statistician might be able to estimate the correct number of factors based on this scree plot alone.

Table 2 shows how the estimates by MAP and PA are distributed. The MAP estimate is correct in 61 of the one hundred cases, while the PA estimate is correct in 89 of the cases. MAP estimates a smaller or equal amount of factors than PA for every repetition; this fits well with the observation by Hayton et al. (2004) that MAP tends to underfactor when it is inaccurate.

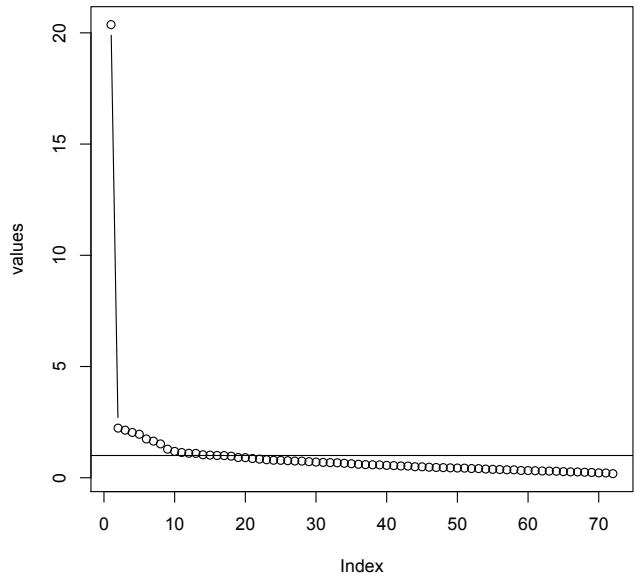

Figure 3: Scree plot of a simulation run with 300 participants, 72 items and 8 factors

3.3 Exploratory factor analysis

Simulation: 100 repetitions, 300 samples, eight factors with nine items each

The 100 simulations for estimating the quality of the entire EFA procedure were run separately from the simulations in the previous section. Within these simulations, thirteen of the simulation runs matched the original factor structure perfectly. In the rest 87 simulation runs, at least one item was dropped from the final model either due to not having any absolute loadings equal or greater than 0.32 or due to having several of them. The number of dropped items is described in a histogram in Figure 4. The mean number of dropped items in a simulation run is 2.92. The maximum number of dropped items, 22, was caused by a single run where only six factors were retained. In addition to rejecting items due to bad loadings, in two simulations an item was factored into a wrong factor.

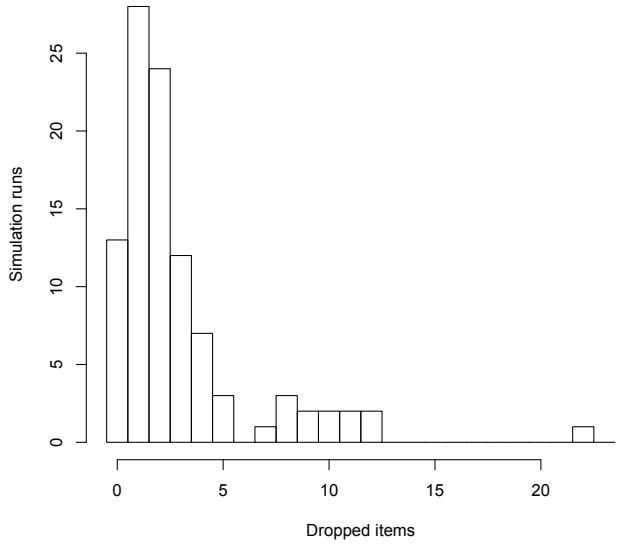

Figure 4: Histogram of the number of dropped items in the exploratory factor analysis simulation runs

3.4 Confirmatory factor analysis

Simulation: four repetitions, 900 samples, eight factors with seven items each

In the original study, only 50 items of the 76 were retained for the confirmatory factor analysis. In these simulations, on average 69 items would be retained in the factor structure. The CFA comparison is made more compatible with the original study by simulating a smaller number of items for it. The simulation is done with AMOS 20 (Arbuckle 2010; Byrne 2001). Due to the amount of manual work required to set up each simulation, the procedure was only repeated four times.

The output from AMOS 20 is shown in Table 3. The results can be considered excellent; the χ^2 hypothesis test isn't rejected for any of the four simulation runs. The RMSEA and CFI model fit indices are also far better than the recommended respective threshold values of 0.05 and 0.95.

Table 3: AMOS 20 output for the confirmatory factor analysis models

simulation	χ^2	χ^2 test p	RMSEA	CFI
1	1496.510	0.549	0.000	1.000
2	1489.818	0.598	0.000	1.000
3	1569.006	0.119	0.007	0.996
4	1545.395	0.224	0.006	0.997

The degrees of freedom (df) for each simulation run is 1504.

4 Discussion

The robustness studies done in this paper show that discrete Likert-scale variables with a large amount of error variance are able to produce results that range from good to excellent.

The exploratory investigation of the simulated data estimated there to be 6–8 factors when the data was simulated with eight multivariate normal factors. Horn's parallel analysis was found to be more robust than Velicer's minimum average partial method with respect to the simulated data. PA was able to estimate the correct number of factors in 89 of the 100 repetitions, while MAP estimated correctly 61 repetitions.

When the PA method was combined with an EFA procedure with the principal axis method and oblimin rotation, the structure of the simulated data was replicated perfectly in only thirteen of the one hundred repetitions. The mean number of items dropped from the model was 2.92. An item loaded onto a wrong factor in two repetitions. Thus, even though the EFA procedures have difficulty in producing the exact same structure the data contains, the differences were found to be usually small and would not have a large impact on the models.

The CFA procedure handled the simulated data very well when given the correct factor model. The χ^2 test null hypothesis of covariance matrix fit was retained in each of the four simulation repetitions — even though rejecting the null hypothesis with experimental data is so common that the χ^2 test isn't usually recommended to be used as a formal test statistic (Schermelleh-Engel et al. 2003). The RMSEA and CFI indices were also near perfect in each of the repetitions. Thus, the CFA methods seem to be able to deal with discrete data extremely well.

The experimental results with similar parameters studied by Törmänen (2012) gave very different results, with item rejections much more common — 26 of the 76 items were rejected based on the EFA procedure. In addition, the CFA model fit index values were significantly worse (χ^2 test $p < 0.000$, RMSEA 0.061, CFI 0.808). Based on the observations done in this study, the discrete nature of the data and the large amount of item specific variance are unlikely to be the causes for these features.

This study shows that discrete data and 59% of item variance coming from error variance don't affect the results of exploratory and confirmatory factor analyses much, and that both principal axis factoring and structural equation modeling seem to be robust when it comes to these kinds of error sources. An additional source of error further studies could assess would be the effect of ordinal data on the methods. This study generated data that was discrete but had an interval scale; assuming that the answers would be modified to not fit on an interval scale, the performance of the EFA and CFA methods could be assessed further.

Bibliography

J.L. Arbuckle. *IBM SPSS® Amos™ 19 User's Guide*. SPSS Inc. Chicago, IL, 2010.

P.M. Bentler and C.P. Chou. Practical issues in structural modeling. *Sociological Methods & Research*, 16(1):78, 1987.

K.A. Bollen. *Structural equations with latent variables*. Wiley New York, 1989.

B.M. Byrne. *Structural equation modeling with Amos: Basic concepts, applications, and programming*. Mahwah, NJ: Erlbaum, 2001.

A.B. Costello and J.W. Osborne. Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical Assessment, Research & Evaluation*, 10(7):1–9, 2005.

J.C. Hayton, D.G. Allen, and V. Scarpello. Factor retention decisions in exploratory factor analysis: A tutorial on parallel analysis. *Organizational Research Methods*, 7(2):191, 2004.

J.L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

A.E. Hurley, T.A. Scandura, C.A. Schriesheim, M.T. Brannick, A. Seers, R.J. Vandenberg, and L.J. Williams. Exploratory and confirmatory factor analysis: Guidelines, issues, and alternatives. *Journal of Organizational Behavior*, 18(6):667–683, 1997.

R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

B. Muthen and D. Kaplan. A comparison of some methodologies for the factor analysis of non-normal likert variables. *British Journal of Mathematical and Statistical Psychology*, 38(2):171–189, 1985.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org/. ISBN 3-900051-07-0.

William Revelle. *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois, 2011. URL http://personality-project.org/r/psych.manual.pdf. R package version 1.01.9.

K. Schermelleh-Engel, H. Moosbrugger, and H. Müller. Evaluating the fit of structural equation models: Tests of significance and descriptive goodness-of-fit measures. *Methods of psychological research online*, 8(2):23–74, 2003.

S.S. Shapiro and M.B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

B. Thompson. *Exploratory and confirmatory factor analysis: Understanding concepts and applications.* American Psychological Association, 2004.

J. Törmänen. Systems intelligence inventory. Master's thesis, Aalto University School of Science, 2012.

W.F. Velicer. Determining the number of components from the matrix of partial correlations. *Psychometrika*, 41(3):321–327, 1976.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL http://www.stats.ox.ac.uk/pub/MASS4. ISBN 0-387-95457-0.

A Appendix: R 2.12.2. source code

```
generate_participant <- function(fac_mean=4.22, fac_var=0.417,
    fac_cov=0.287, error_var=0.598, factors=8, items=6, output_
    factors=TRUE) {
        # Generate the answers for a simulated test participant
            with the given parameters
        # Outputs a vector with the factor scores first and the
            rounded answers after

        require(MASS) # for the multivatirate normal
            distribution function mvrnorm

        data <- rep(NA, factors+factors*items) # collect data
            here

        # Generate covariance matrix:
        co <- mat.or.vec(nr=factors, nc=factors)
        co[] <- fac_cov
        for (i in 1:factors)
                co[i,i] <- fac_var

        # Generate factors from a multivariate normal
            distribution
        data[1:factors] <- mvrnorm(1, rep(fac_mean, factors), co
            )

        # Generate scores by adding error variance
        for (i in 1:factors) {
                for (j in 1:items) {
                        value <- data[i] + rnorm(1, 0, sqrt(
                            error_var))
                        data[factors+(i-1)*items+j] <- max(0,
                            min(6, round(value)))
                }
        }

        if (output_factors)
                return(data)
        else
                return(data[(factors+1):(factors+factors*items)
                    ]) # only return answers
}
```

```
generate_data <- function(participants=100, ...) {
        # Simulate an number of parcicipants using the generate_
            participant function

        # Generate one test participant to set up data matrix
        p <- generate_participant(...)

        # Gather data here
        data <- mat.or.vec(nr=participants, nc=length(p))

        for (i in 1:participants) {
                data[i,] <- generate_participant(...)
        }

        return(data)
}
```

```
efa_factor_simulation <- function(reps=100, n=300, factors=8,
    items=9) {
        require(psych) # For EFA methods

        # Gather results of Velicer's MAP and Horn's parallel
            analysis here
        out <- as.data.frame(mat.or.vec(nr=reps, nc=2))
        colnames(out) <- c("MAP", "Horn")

        # Simulation repetitions
        for (i in 1:reps) {
                # Generate data
                data <- generate_data(participants=n, factors=
                    factors, items=items, output_factors=F)

                # Estimate amount of factors with Velicer's MAP
                out[i, 1] <- which.min(VSS(data)$map)

                # Estimate amount of factors with Horn's
                    parallel analysis
                out[i, 2] <- fa.parallel(data, fm="pa")$nfact
        }

        return(out)
}
```

```
efa_simulation_run <- function(n=300, factors=8, items=9) {
        # Run a full EFA simulation run and output a vector of
            the resulting factoring

        require(psych) # For EFA methods

        # Generate data
        data <- generate_data(participants=n, factors=factors,
            items=items, output_factors=F)

        # Estimate number of factors with Horn's parallel
            analysis
        fac <- fa.parallel(data, fm="pa")$nfact

        # Factor analysis
        f <- fa(data, fm="pa", rotate="oblimin", nfactors=fac)

        # Loadings high enough
        l <- abs(f$loadings) >= 0.32

        # Get output factoring
        out <- rep(NA, nrow(l))
        for (i in 1:nrow(l)) {
                if (sum(l[i,]) == 1)
                        out[i] <- which.max(f$loadings[i,])
        }

        return(out)
}
```

```
efa_simulation <- function(reps=20, n=300, factors=8, items=9) {
        # Run an EFA simulation with repetitions

        result <- mat.or.vec(nr=reps, nc=(factors*items))

        for (i in 1:reps)
                result[i,] <- efa_simulation_run(n, factors,
                    items)

        return(result)
}
```