Exercise 21 - Principal Component Analysis and Exploratory Factor Analysis

Zoltan Kekecs

27 May 2020

Table of Contents

Abstract

In this exercise we will deal with the "curse of dimensionality". This is a problem which we encunter if we have a lot of predictors/parameters in our regression model. If the number of paramters in the model is too high in comparison with the number of observations in the dataset, there is a large threat for overfitting, and the model coefficients will be less useful for prediction in new data from the target population. In the current exercise we will learn methods for dimensionality reduction, to reduce the number of predictors with as little information loss as possible. We will do this via Principal Component Analysis. Additionally, we will learn a related technique called Exploratory Factor Analysis which uses a similar logic to explore underlying/latent factors governing a set of observed variables.

This exercise is partially built on the Factor Analysis course in DataCamp and the UCLA Factor Analsysis guide: https://stats.idre.ucla.edu/spss/seminars/efa-spss/

Data management and descriptive statistics

Humor Style Questionnaire dataset

In this exercise we will work with a dataset containing data about people's style of humor. You can download the dataset from this GitHub link:

https://github.com/kekecsz/SIMM32/blob/master/2020/Lab_6/Humor%20Style%20Questionnaire.sav

(The dataset was originally downloaded from https://openpsychometrics.org/_rawdata/.) This data was collection with an interactive online version of the Humor Styles Questionnaire from Martin, R. A., Puhlik-Doris, P., Larsen, G., Gray, J., & Weir, K. (2003). Individual differences in uses of humor and their relation to psychological well-being: Development of the Humor Styles Questionnaire. Journal of Research in Personality, 37, 48-75.

The variables Q1 through Q32 were statements rated on a five point scale where 1=Never or very rarely true, 2=Rarely true, 3= Sometimes true, 4= Often true, 5=Very often or always true (-1=did not select an answer). The exact statements were:

Q1. I usually don't laugh or joke around much with other people.

Q2. If I am feeling depressed, I can usually cheer myself up with humor.

Q3. If someone makes a mistake, I will often tease them about it.

Q4. I let people laugh at me or make fun at my expense more than I should.

Q5. I don't have to work very hard at making other people laugh—I seem to be a naturally humorous person.

Q6. Even when I'm by myself, I'm often amused by the absurdities of life.

Q7. People are never offended or hurt by my sense of humor.

Q8. I will often get carried away in putting myself down if it makes my family or friends laugh.

Q9. I rarely make other people laugh by telling funny stories about myself.

Q10. If I am feeling upset or unhappy I usually try to think of something funny about the situation to make myself feel better.

Q11. When telling jokes or saying funny things, I am usually not very concerned about how other people are taking it.

Q12. I often try to make people like or accept me more by saying something funny about my own weaknesses, blunders, or faults.

Q13. I laugh and joke a lot with my closest friends.

Q14. My humorous outlook on life keeps me from getting overly upset or depressed about things.

Q15. I do not like it when people use humor as a way of criticizing or putting someone down.

Q16. I don't often say funny things to put myself down.

Q17. I usually don't like to tell jokes or amuse people.

Q18. If I'm by myself and I'm feeling unhappy, I make an effort to think of something funny to cheer myself up.

Q19. Sometimes I think of something that is so funny that I can't stop myself from saying it, even if it is not appropriate for the situation.

Q20. I often go overboard in putting myself down when I am making jokes or trying to be funny.

Q21. I enjoy making people laugh.

Q22. If I am feeling sad or upset, I usually lose my sense of humor.

Q23. I never participate in laughing at others even if all my friends are doing it.

Q24. When I am with friends or family, I often seem to be the one that other people make fun of or joke about.

Q25. I don't often joke around with my friends.

Q26. It is my experience that thinking about some amusing aspect of a situation is often a very effective way of coping with problems.

Q27. If I don't like someone, I often use humor or teasing to put them down.

Q28. If I am having problems or feeling unhappy, I often cover it up by joking around, so that even my closest friends don't know how I really feel.

Q29. I usually can't think of witty things to say when I'm with other people.

Q30. I don't need to be with other people to feel amused – I can usually find things to laugh about even when I'm by myself.

Q31. Even if something is really funny to me, I will not laugh or joke about it if someone will be offended.

Q32. Letting others laugh at me is my way of keeping my friends and family in good spirits.

On the next page test takers were prompted for three more variables:

age.

gender. chosen from drop down list (1=male, 2=female, 3=other)

accuracy. How accurate they thought their answers were on a scale from 0 to 100, answers were entered as text and parsed to an integer. They were instructed to enter a 0 if they did not want to be included in research.

life_stress. This is a new variable that was only added for the purpose of this exercise. This was not in the original study. This variable was simulated (not real). Let's imagine that participants had to "Rate on average how stressful do you think your life is (thinking about the bast year) on a scale of 0-9, where 0 means not at all stressful, and 9 means extremely stressful."

We can get basic descriptive statistics with **Analyse > Descriptive Statistics > Descriptives**, and we can get histograms and boxplots with **Analyse > Descriptive Statistics > Explore** as seen in the previous exercises.

```
DESCRIPTIVES VARIABLES=Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14
Q15 Q16 Q17 Q18 Q19 Q20 Q21

  Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 affiliative selfenhancing
agressive selfdefeating age

  gender accuracy life_stress

 /STATISTICS=MEAN STDDEV MIN MAX.


EXAMINE VARIABLES=Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15
Q16 Q17 Q18 Q19 Q20 Q21 Q22

  Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 affiliative selfenhancing
agressive selfdefeating age

  gender accuracy life_stress

 /PLOT BOXPLOT STEMLEAF HISTOGRAM

 /COMPARE GROUPS

 /STATISTICS DESCRIPTIVES

 /CINTERVAL 95

 /MISSING LISTWISE

 /NOTOTAL.
```

Let's say we would like to determine which are the most important features in humor style that could help us determine life_stress. One way for doing this would be to fit a linear regression model with life_stress as a dependent and Q1-Q32 of the questions as predictors.

If we run this model, in the Coefficients table we find that there are multiple variables that seem to have a significant added predictive power to the model. However, we also know that we have performed 32 statistical tests, which leads to an inflated risk for type I error. There is a large probability that at least one (or multiple) of these predictors are not really related to the outcome, the finding is just a "chance finding" due to sampling error (the luck of the draw in our sample). We have the same problem if we test the correlation of all of the predictors and the outcome variable.

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA COLLIN TOL
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT life_stress
  /METHOD=ENTER Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15
Q16 Q17 Q18 Q19 Q20 Q21 Q22 Q23
    Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32.


CORRELATIONS
  /VARIABLES=Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16
Q17 Q18 Q19 Q20 Q21 Q22 Q23 Q24
    Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 life_stress
  /PRINT=TWOTAIL NOSIG
  /MISSING=PAIRWISE.
```

In fact, if we run the correlation matrix (Anlalyse > Correlate > Bivariate) of all predictors, we see that almost every predictor variable is also correlated significantly with the other predictors. The collinearity diagnostics (VIF in the regression output) tells us that this multicollinearity is not problematic in our particular model (The VIFs are all below 3), but we can imagine that the correlation of the variables could lead to issues with multicollinearity as well in another similar study.

The curse of dimensionality

So entering 32 intercorrelated predictors into our model is not ideal, especially if we have a small sample size. This is sometimes referred to in the literature as the curse of dimensionality. Dimensionality refers to the fact that in statistical models such as regression, the more variables we have in the model, the higher dimensions are used in the math. For example, in simple regression, the regression line is truly a line that can be depicted in a two dimensional space: the dependent variable on the y axis and the predictor on the x axis. However, if we have two predictors (multiple regression), the regression line is actually a regression plane, that can only be depicted in a 3 dimensional space: the dependent variable on the y axis and the predictors on the x and z axes, and so on. In a

linear regression with 32 predictors, the regression plane is 33 dimensional. The more dimensions we have, the more flexible our model is, able to fit to the nooks and crannies of the sampling error, leading to higher and higher risk of overfitting.

This problem gets more and more problematic as the number of dimensions or parameters in the model is getting close to the number of observations. If we fit a simple regression line (using only 1 predictor) on data with only 2 observations, there is a line that fits the data perfectly, since there is always a straight line that can connect two points. Since the regression will find the line which passes closest to the observations, with 2 observations we will end up with a regression model that has 0 error. This might seem good at first, but this is actually bad. This means that our model can perfectly fit the error in our sample, and will have almost no resemblance of the actual pattern in the population. The same goes if we have 3 observations and a model with 2 predictors: there is a plane (sheet) that can perfectly fit 3 points in a 3 dimensional space, sothe model can again fit to the error and we don't know how much information we gain about the actual pattern in the whole population. So it is easy to see that the closer the number of dimensions is to the number of observations, the more flexible the model is, and the less informative the model coefficients would be for new data due to overfitting.

Principal Component Analysis

When facing the curse of dimensionality, one of the most straightforward options is to reduce the number of dimensions. We could do this by dropping some of the variables from the model, but based on what? If we did it by dropping the predictors with the lowest predictive power or correlation with the outcome we would be again contributing to overfitting, so this is not desirable. A better option would be to merge some of our more similar variables together, this way, decreasing the number of predictors overall, but still keeping information from all original predictors.

When we look at the correlation matrix of predictor variables we can notice that there are "clusters" of variables that are more closely related to each other than the other variables. For example variables Q1, Q5, Q13, Q17, and Q25 seem to form such a cluster. They are all correlated at around 0.4-0.5. This means that we could merge these variables into a single variable (for example by averaging them), and this way reducing the number of predictors by four. This is essentially what we do in a controlled systematic way in principal component analysis (PCA), with the distinction that one original variable can be a part of multiple "merged" variables.

How does PCA work?

In principal component analysis our goal is to reduce the number of dimensions. Nevertheless, in PCA we start by creating the same number of dimensions as we previously had. What we do is we re-define the orientation of the coordinate axes that we previously had defined by our variables. Basically what we do is shift our perspective of the observations, while the observations keep their original relations to each other. But we do this shift in the dimensions systematically with a purpose: we first find a dimension on

which the data has the most variance, this dimension captures the most variance from the data. We start from this initial dimension/axis, then find another dimension which captures most of the *remaining* variance. The variance that is not already captured by the first dimension, and so on, until we reach the number of dimensions we started out with. It turns out that we can maximize the amount of variance explained by each consecutive dimension if the dimension is perpendicular (orthogonal, at a 90 degree angle) to the previous dimensions. So we end up with a new coordinate system with the same number of dimensions as before, but with a different orientation.

Importantly, the new set of dimensions are systematically different from each other in the amount of variance captured by them. The amount of variance on a given dimension is called the **eigenvalue** of that dimension, and going from the first dimension we identified to the last, the eigenvalue (amount of variance in that particular dimension) decreases. This is the fact that allows us to reduce the number of dimensions in the end, because as the amount of variance on the dimensions decreases, the dimensions get less and less useful, less and less informative for us. A dimension (variable) on which almost all observatrions take the same or very similar values is less important to consider, so we might disregard some of the dimensions extracted this way and still be able to retain most of the information about the variability in our dataset. Imagine that we did a study among children in the first grade of primary school, and we record their age among ther variables. Imagine that it turns out that almost everyone in the sample is age 6 with just a few months between all the children. We might decide to ignore this variable in our study, since there is almost no variability in it in our sample. In PCA we artificially generate such variables, and variables that are extremely informative in comparison, in order to make it easier for us to decide which ones to retain and which ones to exclude from our analysis.

Building the PCA model in SPSS

We can run an initial PCA to find out what is the number of principal components that we can create from the data and still retain useful information from the entered original variables. **Go to Analyze > Dimension reduction > Factor** . In the main screen put the variables Q1-Q32 in the Variables box (leave the Selection variable box empty). In the **Descriptives menu** ask for **univariate descriptives** so that we can see the number of valid cases. Make sure that in the **Extraction menu** the **Method** is set to **Principal components** (this is the default), and in the same menu check the Scree plot checkbox. We don't need to change anything else right now, just make sure that

```
FACTOR
 /VARIABLES Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17
Q18 Q19 Q20 Q21 Q22 Q23 Q24
  Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32
 /MISSING LISTWISE
 /ANALYSIS Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17
Q18 Q19 Q20 Q21 Q22 Q23 Q24
  Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32
 /PRINT INITIAL EXTRACTION
 /PLOT EIGEN
 /CRITERIA MINEIGEN(1) ITERATE(25)
 /EXTRACTION PC
 /ROTATION NOROTATE
 /METHOD=CORRELATION.
```

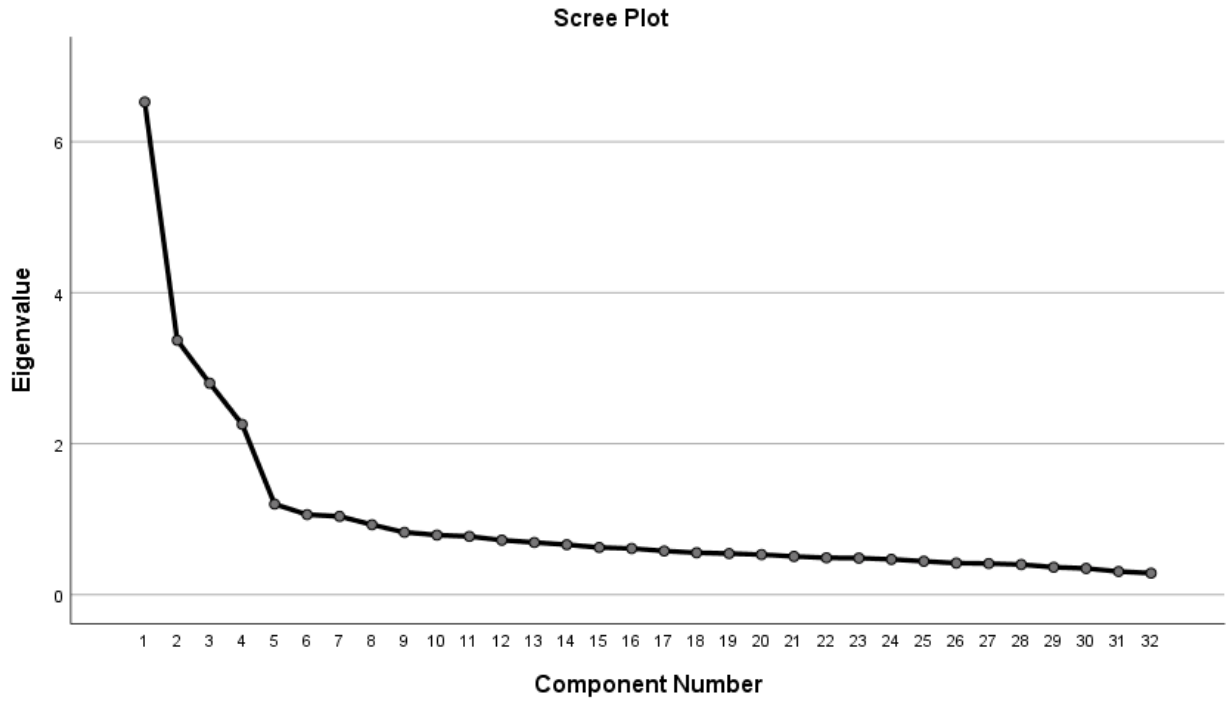
How many principal components to extract?

The Total variance explained table in the SPSS output describes the dimensions (principal components) extracted by the PCA. There are 32 of them of course, because that is the number of dimensions we entered into the PCA originally. The % Variance column shows the % of variance of tha data on the given dimension. As we would expect, this number decreases with with each dimension. The first principal component describes about 20% of the total variance of the data, while the 5th, 6th, and 7th dimension only capture about 3%. So how can we decide which of the principal components to retain?

There are multiple decision rules that we can use, and the literature is not consistent in which one is used. Some of the techniques used in the literature are mentioned below:

1. Scree test

A well accepted method for deciding about the number of dimensions to retain is the Scree test. We can use the Scree plot that we requested in the Extraction menu to perform this "test".



The scree plot shows the eigenvalue of each dimension connected by a line. The scree test is a visual analysis of the scree plot that starts with **finding the "elbow" of the graph**. This is the point where the eigenvalues seem to level off, where no more substantial break can be seen in the slope of the line. Components to **the left of this point** should be retained. This does not include the last break point, so everything *before* that point is retained, and every other dimension, including the one which represents the final breaking point, is excluded. This method is commonly used and is accepted in the literature, but it can be subjective and different researchers might come to different conclusions based on this.

This rule suggests that we should retain 4 component, since the last substantial break in the slope is at the 5^{th} factor.

2. The Kaiser-Guttman rule (Eigenvalue > 1)

One strategy that is objective and has the same result for each researcher is the Kaiser-Guttman rule. This is also a built-in default in SPSS. The method looks at the Eigenvalues of the dimensions and retain the ones that have Eigenvalue > 1. The eigenvalue of 1 has a bit of a special significance, because the average of the eigenvalues is always 1, whatever dimensions you use for describing the data. This also means that the eigenvalue for the original set of dimensions (defined by the original variables) also had an average eigenvalue of 1, and thus any dimension that has lower than 1 eigenvalue is explaining less variance than the average original variable. So dimensions with eigenvalues below 1 are considered less useful, since our initial goal was to combine the information from multiple variables into “super variables”, but if the new “super variable” explains less variance than the average single original variable, it is not seen as so “super” after all.

This decision rule for retaining the components is not commonly used, or is only used in combination with other rules.

This rule suggests that in our example we should retain 7 components, since even the 7th component has higher eigenvalue than 1.

3. Parallel analysis

The third, and currently most accepted, decision rule is using parallel analysis. The ide of parallel analysis is that we generate/simulate datasets which have similar characteristics to our original dataset (the same number of observations and the same number of variables), but we simulate that these variables come from a population where the varaibles do not correlate with each other. We repeate this many times. 1000-10000 is tipically used, the number depending on the level of precision we want to achieve. Generally the more the better, but the precision gained in the last 5000 iterations is not that great, so if you have a large dataset and a slow computer, you should first consider running a 1000 iterations, and only run the final with 5-10000 iterations once you have arrived at your final model.

We perform the same PCA on all of these random uncorrelated datasets separately, and take the average eigenvalue for each component, then, we create a scree plot of these average eigenvalues as well, and copare it to the scree plot of our original dataset. Thus, essentially we use the eigenvalues derived from the random datasets as a sort of "null-model".

We retain the components which have a higher eigenvalue than the average eigenvalue of the random data corresponding to the same component. In other words we can keep all components which are above the random data eigenvalue line on the scree plot.

SPSS does not contain an option for the parallel analysis, but we can reproduce this using the syntax. The code is rather sophisticated, but you don't have to write it yourself, Brian O'connor already did it for you. To get the syntax, you can wisit his website: https://people.ok.ubc.ca/brioconn/nfactors/nfactors.html and select rawpar.sps link, or click on this link directly: https://people.ok.ubc.ca/brioconn/nfactors/rawpar.sps. You can copy and paste this code into your syntax.

You need to adjust the following things in this syntax for it to run properly:

1. You need to delete this part (this generates a dataset on which you can try this syntax out, but we do not need that):

```
* Start of artificial data commands.

set length=none printback = off  width = 120.

input program.

loop #a=1 to 500.

compute com1 = normal (10).
```

```
compute com2 = normal (10).
compute com3 = normal (10).
compute var1 = normal (10) + com1.
compute var2 = normal (10) + com1.
compute var3 = normal (10) + com1.
compute var4 = normal (10) + com2.
compute var5 = normal (10) + com2.
compute var6 = normal (10) + com2.
compute var7 = normal (10) + com3.
compute var8 = normal (10) + com3.
compute var9 = normal (10) + com3.
end case.
end loop.
end file.
end input program.
factor var = var1 to var9.
* End of artificial data commands.
```

2. Change the “VAR =” to the range of variables in the dataset: in our case this is “Q1 to Q32”

3. Change the “compute ndatsets = ” to a 1000 at least.

4. Make sure that you have the “compute kind = ” part set to the right number. If you are doing PCA, this should be set to 1 (this is the default), in case of factor analysis this should be set to 2.

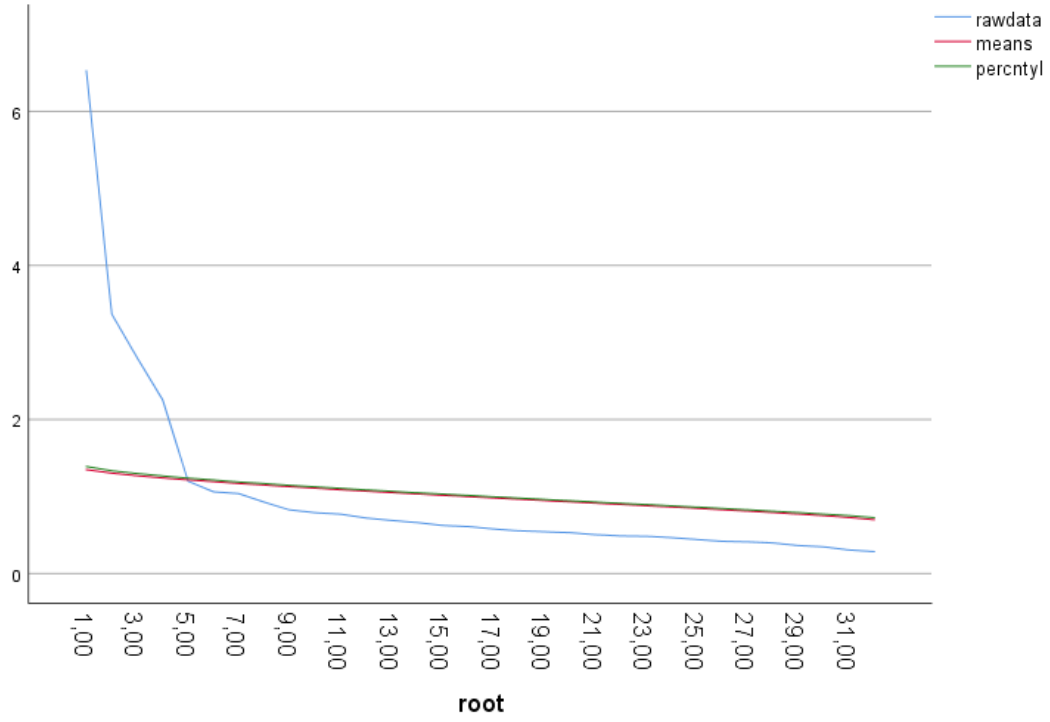
5. Set “compute randtype = ” to 2 if you have non-normally distributed data. This is necessary because the the original simulation generates data randomly from a normal distribution. If your data is not normally distributed, you need to use another method to generate the random data, and in order to match your data distribution the most, the permutation-based method is the best (which is reached by setting compute randtype = 2). This uses your own original data to generate random data with a method similar to bottstrapping.

6. The syntax saves a data file on your hard drive, and it will generate the scree plot from that. So you need to set the “outfile= 'screedata.sav'” to the actual complete path for the

folder and file name you want the program to use, and you need to set “GET file= 'screedata.sav'.” to the same path, so that the syntax can read this file from your hard drive.

Finally, you can select the whole syntax segment you have copied and modified from its start to its end and run it.

This may take a while, especially if you chose the permutation based method. (For me on a Intel i7 6600U processor it took about 2 seconds to run the code with compute ndatsets = 1000 and compute randtype = 1, while it took about 2 minutes to run the same with compute randtype = 2 in our example dataset where we have about 1000 valid observatiosn.)



The scree plot returned by the parallel analysis shows in our case that we should retain 4 factors, the eigenvalue of the 5th factor is 1.2 in the original dataset (it is called “raw data” in the table returned by the parallel analysis and on the scree plot), while the mean of the eigenvalus of the random data is 1.21 for the 5th factor in the random datasets. The scree plot also show the same conclusion, the red line runs below the eigenvalue of the 4th component in the “rawdata”, but now the 5th component.

Re-running the model with the final component number

Once we have decided on the final number of components to keep, we need to run the PCA again, but this time, with specifying the number of components to retain.

We can leave everything as set before, but in the **Extraction menu** instead of using the Eigenvalue as a cutoff, specify that you want to **extract a fixed number of factors.** In this example I will choose 4, since both the scree test and the parallel analysis indicated this number.

Interpreting the output

Descriptive statistics table. This table includes the basic descriptive statistics of the variables entered into the analysis. One important thing to pay attention to is the Analysis N, because this shows the number of valid cases (the total N minus the cases excluded because of missing values).

Communalities table. Communality indicates the "common variance" portion of the total variance of that particular variable. Common variance stands for the variance explained by the components, that is, the variance that is common with other variables. There are two columns here: Initial and Extration. The Initial column indicates the expected common variance within a variable. In PCA we assume that this is 1 (100%) for each variable. In other words, the total variance of the item can be explained by the underlying components. (This is one of the main differences compared to EFA). The Extraction column displays the amount of variance explained by the components that have been retained at the end of the PCA. Note that if we asked the PCA analysis in the Extraction menu to retain 32 factors, this number would be 1 for each variable, since the PCA will find a solution where the total variance is explained by the full number of components. In our case we retained 4 components, so in the Extraction column we can see the portion of the variance explained by these 4 components together of each variable.

Total variance explained table. This table contains information about the variance explained by each retained component, and the eigenvalues of each component separately. The Cumulative % column is also useful to look at to see what is the total % of variance explained by the components together. This represents the amount of information we retain from the variability of the original data if we choose to use the specified number of components we retained.

Component matrix. This table contains information about the correlation of each **original variable (these are also called items**) with each of the components. This correlation is commonly called "**loadings**" of the items. Our component matrix indicates that the correlation of Q1 is -0.569, 0.259, 0.067, and 0.350 with the four components respectively. We can interpret this as the first component holding the most information about Q1, since it "**has the highest loading**" (the absolute value of the loading) on this component, and this **loading is negative**. Note that the sum of the squared loadings for each item (each row) gives the communality listed in the communality table. Also, note that the sum of squared loadings for each component (each column) gives the eigenvalue for that component.

Computing and using factor scores

Now that we have arrived at the final components to extract, we can actually calculate the scores of each participant on these new dimensions based on their scores on the original variables.

We can do this in the Factor procedure where we set up the PCA model in the **Scores menu**. We need to scecify that we want the component scores (these are called factor scores usually in factor analysis) to be **saved as new variable**. The **regression** method is OK for saving these scores. When we run the analysis now four new variables are saved in our dataset corresponding to the factor scores in each component.

```
FACTOR
  /VARIABLES Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17
Q18 Q19 Q20 Q21 Q22 Q23 Q24
    Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32
  /MISSING LISTWISE
  /ANALYSIS Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17
Q18 Q19 Q20 Q21 Q22 Q23 Q24
    Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32
  /PRINT UNIVARIATE INITIAL EXTRACTION
  /PLOT EIGEN
  /CRITERIA FACTORS(4) ITERATE(25)
  /EXTRACTION PC
  /ROTATION NOROTATE
  /SAVE REG(ALL)
  /METHOD=CORRELATION.
```

We can use these new variables now to supplement the original set of variables in our regression analysis to overcome the issue presented by the curse of dimensionality.

```
REGRESSION
 /MISSING LISTWISE
 /STATISTICS COEFF OUTS R ANOVA COLLIN TOL
 /CRITERIA=PIN(.05) POUT(.10)
 /NOORIGIN
 /DEPENDENT life_stress
 /METHOD=ENTER FAC1_1 FAC2_1 FAC3_1 FAC4_1.
```

There are some interesting things to note regarding this analysis: For example, even though we only use 4 variables now instead of 32, the adjusted R^2 is not too much lower compared to the previous analysis with all the original predictors: for the previous model it was 0.32, while now it is 0.28. So we can still explain roughly the same amount of variance in the outcome. Second, the VIF is now 1 for all of the predictors. This is because the factors are completely uncorrelated with each other. So this is a great way of fixing multicollinearity as well.

Introduction to Exploratory Factor Analysis (EFA)

Exploratory Factor Analysis (EFA) is another dimension reduction technique that is similar to PCA.

In PCA we assumed that there is only "common variance", so we wanted to maximize the extent to which we can cover/capture the variability in the original variables with no particular regard for making the components have a real-life relevance or interpretability. By contrast, in EFA we assume that the observed variables are manifestations of a smaller set of latent factors that are not directly observable, but the observed variables hold information about them. In this framework we do not assume that there is only "common variance". Instead, in EFA we assume that any observed variable consists of three components: 1. Common variance (explained by the underlying factors), 2. Specific variance (variance that is specific to a particular item), 3. Error variance (for example due to measurement error). (2. and 3. together are also called uniqueness of an observer variable).

The most important steps in EFA:

- Assess factorability
- Factor extraction
- Selecting the ideal number of factors
- Factor rotation
- Interpretation of the factors

Factorability tests

When we test factorability we are interested in whether there is enough correlation between the observed variables that allow for carrying out EFA. We can test this using the Bartlett sphericity test and the Kaiser-Meyer-Olkin (KMO) test.

Bartlett sphericity test

The idea behind the Bartlett test is to copare the actual observed correlation matrix of the observed variables with a hypothetical null-correlation matrix, where every correlation is set to 0 (this is also called the identity matrix). We test the null hypothesis that the two correlation maricies don't differ from each other. So if the test is significant, we can say that the two matrices are significantly different from each other, that is, that the observed variables correlate with each other. This means that the observed variables are factorable.

However, there is a serious drawback to using the Bartlett's test: that with large enough samples this test almost always returns significant results. So even though people tend to report this test, they do not consider this as a definitive indicator of factorability. The only time we should take the result of this test seriously is when the ratio of the number of observatiosn and the number of observed variables is lower than 5. In our case this value is about 1071/32 = 33.5, so the Bartlett's test is not reliable.

Kaiser-Meyer-Olkin (KMO) teszt

The KMO test compares the partial correlation matrix with the regular correlation matrix. In the partial correlation matrix we calculate the correlation of every pair of observed variables if we take out the effect of every other observed variable from this correlation. The KMO value shows the difference between the partial and the regular correlation matrix. In cases where the variables hold a lot of common variance, (there is a large chance that they are governed by the same latent factors) partial correlations are low, so the KMO index will be large. In the KMO test, values close to 1 indicate good factorability. The KMO index needs to be at least 0.6 for the variables to have reasonable factorability.

We can ask for the factorability tests in the Descriptives menu within the Factor procedure by checking the KMO and Bartlett's test checkbox. The KMO index indicates good factorability.

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		,882
Bartlett's Test of Sphericity	Approx. Chi-Square	10676,882
	df	496
	Sig.	,000

Faktor extraction method

Factor extraction is done in the same way as we did with PCA. We choose the **Analyse > Dimension reduction > Factor** procedure. We specify everything the same way, except that in the Extraction Menu we select a different method than PCA. We usually choose Maximum Likelihood if the assumption of multivariate normality holds, while we choose Principal axis factoring if this assumption is violated.

Testing multivariate normality

SPSS does not provide a built-in option to assess multivariate normality, however this can be done by following this Guide:
https://eduimed.usm.my/EIMJ20150702/EIMJ20150702_10.pdf

The basic idea is to create a chi-square versus Mahalanobis distance plot.

1. First we need to get the Mahalanobis distances for all cases. This can be extracted by specifying a **linear regression with the observed variables** (Q1-Q32 in our case) **as predictors**, and **ANY variable as the outcome**. It does not matter what you use as an outcome, it does not affect the Mahalanobis distances, just the predictors. In the **Save menu** in the Regression prodcedure ask for saving **Mahalanobis distances**, then run the procedure (no other settings are necessary). The Mahalanobis distances will be saved in a new variable in the dataset.

2. Now you need to sort the dataset by this new variable containing the Mahalanobis distances in ascending order. This can be done in **Data > Sort cases**.

3. You need to compute p-values of chi-squared for the each case. This can be done in **Transform > Compute**. Specify a new variable name, for example pval, then, enter the following expression into the Numerical Expression box: ($CASENUM-0.5)/1071. Here, the $CASENUM is an artificial variable containing increasing number by rows, practically this is the row number for each case in the dataset. The number 1071 is the number of cases in the dataset. You need to change that to the actual number of cases in your dataset.

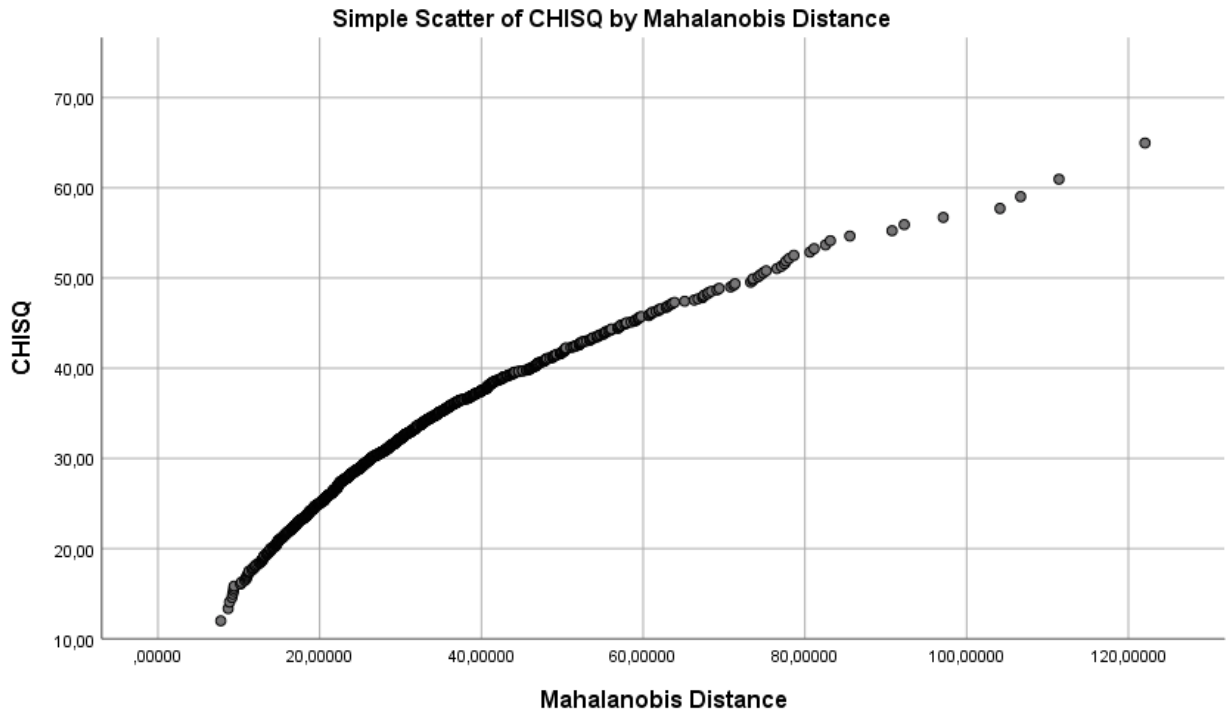
4. You need to compute the chi-squared values corresponding to the p-values. This is again done in **Transform > Compute** , specify a new variable name for the chi-squared values, for example CHISQ, then, enter the following expression into the Numerical Expression box: IDF.CHISQ(pval,32). Here, the pval refers to the variable containing the p-values we just created, and the 32 reflects the number of observed variables we want to test the multivariate normality of.

5. Finally, you need to go to the chart builder and create a scatterplot with the chi-squared values on the y axis and the Mahalanobis distances on the x axis.

This plot should show that the cases fall on a straight line (just like when evaluating a normal QQ plot).

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT age
  /METHOD=ENTER Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20 Q21
Q22 Q23
    Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32
  /SAVE MAHAL.
```

```
DATASET COPY  normcheck_dataset.

DATASET ACTIVATE  normcheck_dataset.

FILTER OFF.

USE ALL.

SELECT IF (NOT(SYSMIS(MAH_1))).

EXECUTE.


SORT CASES BY MAH_1(A).


COMPUTE pval=($CASENUM-0.5)/993.

EXECUTE.


COMPUTE CHISQ=IDF.CHISQ(pval,32).

EXECUTE.


* Chart Builder.

GGRAPH

  /GRAPHDATASET NAME="graphdataset" VARIABLES=MAH_1 CHISQ MISSING=LISTWISE
REPORTMISSING=NO

  /GRAPHSPEC SOURCE=INLINE

  /FITLINE TOTAL=NO.

BEGIN GPL

  SOURCE: s=userSource(id("graphdataset"))

  DATA: MAH_1=col(source(s), name("MAH_1"))

  DATA: CHISQ=col(source(s), name("CHISQ"))

  GUIDE: axis(dim(1), label("Mahalanobis Distance"))

  GUIDE: axis(dim(2), label("CHISQ"))

  GUIDE: text.title(label("Simple Scatter of CHISQ by Mahalanobis Distance"))

  ELEMENT: point(position(MAH_1*CHISQ))

END GPL.
```



The line of cases seems to be slightly deviating from a straight line at the lower end. This does not seem like a major deviation, but still it would be safer to use the Principal axis factoring as a method, since the assumption of multivariate normality seems to be violated to some extent. If the line looks more straight, we could use the maximum likelihood method for extraction.

How many factors to extract?

In the extraction menu we also need to specify how many factors to extract. At first, we can revert this to using the Eigenvalue > 1 criteria. We will need to assess the number of factors to extract again. For this we can use the same method as learned for PCA with some caveats.

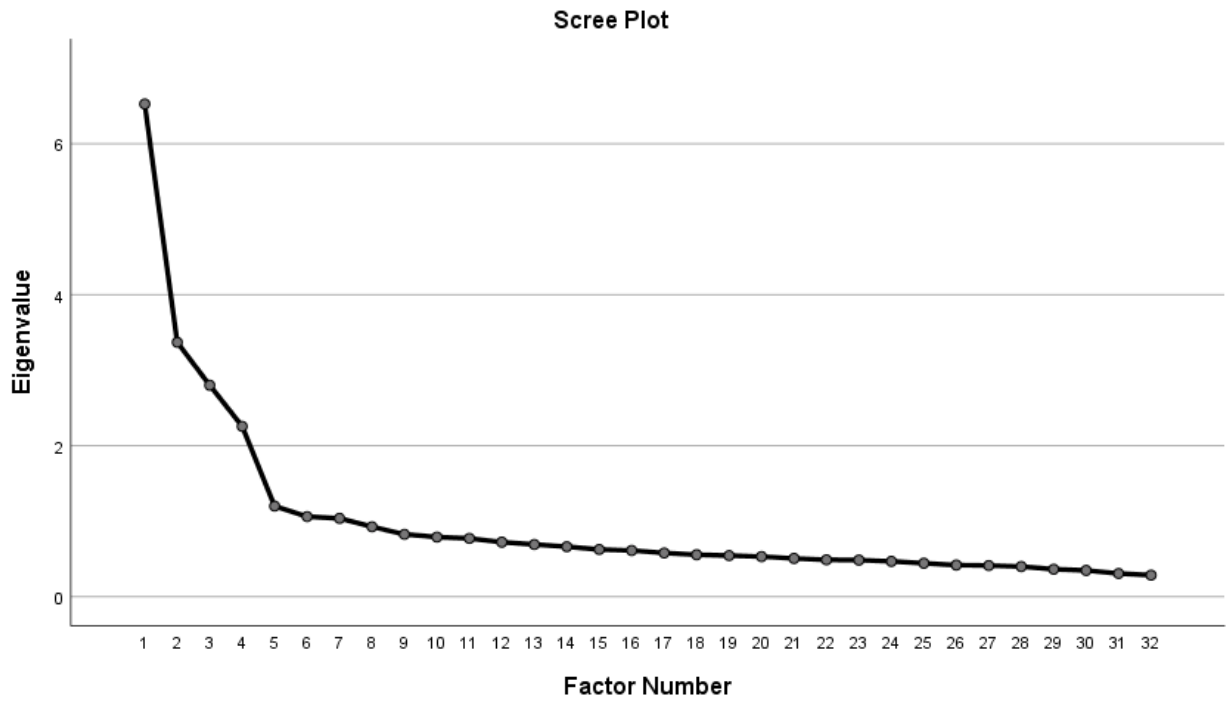
The scree plot remains a well accepted method. This indicates that we should extract 4 factors. However, this scree plot is produced based on the initial Eigenvalues and not the post-extraction eigenvalues. The post-extraction eigenvalues can be found in the parallel analysis scree plot, or you can create it based on the Extration Sum of Squared Loadings table in excel for example. The Eigenvalue > 1 guideline can also be used as a basic principle to further guide our decision.

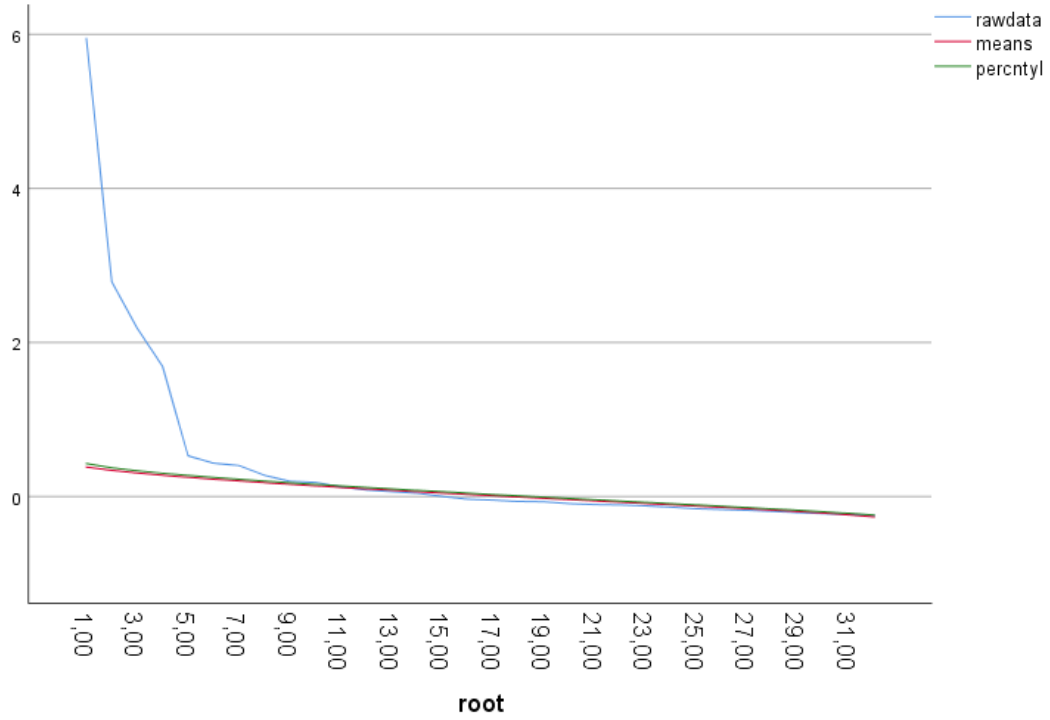
One issue with parallel analysis is highlighted in the output of this analysis if we choose compute kind = 2 (indicating regular EFA instead of PCA).

```
"Warning: Parallel analyses of adjusted correlation matrices
eg, with SMCs on the diagonal, tend to indicate more factors
than warranted (Buja, A., & Eyuboglu, N., 1992, Remarks on parallel
analysis. Multivariate Behavioral Research, 27, 509-540.).
The eigenvalues for trivial, negligible factors in the real
```

data commonly surpass corresponding random data eigenvalues

for the same roots. The eigenvalues from parallel analyses can be used to determine the real data eigenvalues that are beyond chance, but additional procedures should then be used to trim trivial factors."





So if we solely rely on using the regular EFA parallel analysis, we will get too many factors. Some use the PCA parallel analysis to decide how many factors to extract even in EFA. Others use multiple indicators and make a judgement that way. In factor analysis the number of factors extracted also relies on the interpretability of the factors. If one of the factors seems to be uninterpretable, it is often dropped.

In our case, let's use the scree plot as out main indicator, and extract 4 factors.

Interpreting the factors

The interpretation of the factors is an iterative process. IT often takes many runs and changing the settings of the factor procedure to get to the final model and until we have arrived at the final factor structure we cannot finalize our interpretation of the factors. Nevertheless, trying to interpret the meaning of the factors, that is, trying to figure out what latent constructs they could reflect, is important in fine-tuning the EFA.

This is an exploratory process so we shouldn't be afraid of exploring different options, parameter settings, extraction and rotation methods, item configurations. However, it is useful in the long run to **keep a record** of what were the different settings/options that were tried during this process until we get to the final factor structure. This could improve the transparency and reproducibility of our research, and it could also help us save time if we know what we have already tried before.

Which table to use for interpretation?

The interpretation of the factors is usually done by using one of the correlation matrixes produced by the SPSS output. If we did not use factor rotation this should be the Factor Matrix, if we used rotation this should be either the Rotated factor matrix (in the case of orthogonal rotations), or the Pattern Matrix (in the case of oblique rotations). When we use oblique rotations (like promax or direct oblimin) SPSS also produces table called the structure matrix. The difference between the pattern matrix and the structure matrix is that the Pattern matrix includes regression coefficients as loadings, while the Structure matrix includes correlation coefficients. The regression coefficients in the Pattern matrix is similar to what we would get if we built a linear regression model to predict the observed variable by the factor scores. If no rotation is used or in the case of orthogonal rotation the factors are uncorrelated, so there is no shared variance explained by them so this has the same meaning as a simple correlation coefficient in those cases. However, in oblique rotations, the factors are allowed to correlate, which results in some overlap in the portion of variance they explain of the variability of each observed variable. The pattern matrix accounts for this shared variance by using a regression coefficient instead of the correlation coefficient, this way, the meaning of the loadings in that table is the portion of the variance of an observed variable explained uniquely by the given factor, while taking into account the variance explained by all other factors. While the structure matrix contains simple correlation coefficients, which does not take into account the variance explained by the other factors.

The interpretation of the factors is usually done by looking at which items have high loadings on a given factor, and figuring out from this information what could that factor mean. To achieve the best interpretability it can help to select the option “sort by size” in the Options menu, so that the component/factor matix will be sorted based on which is the component on which the variable loads the highest.

For example the first component seems to hold information about generally how often and how readily does the person joke/use humor, while the second component seems to be mostly related to questions related to self-harm by humor, etc.

The sign of the loading is important here as well in the interpretation. For example Q17. I usually don’t like to tell jokes or amuse people. Has a negative loading on factor 1, while Q14. My humorous outlook on life keeps me from getting overly upset or depressed about things has a positive loading. This confirms that the first factor is about general attitude about joking and humor.

Factor Matrix[a]

	Factor			
	1	2	3	4
Q17. I usually don’t like to tell jokes or amuse people.	-,613	,256	,104	,363
Q14. My humorous outlook on life keeps me from getting overly upset or depressed about things.	,564	-,300	,139	,300
Q5. I don’t have to work very hard at making other people laugh—I seem to be a naturally humorous person.	,552	-,220	-,062	-,231

Q21. I enjoy making people laugh.	,550	-,282	,057	-,331
Q26. It is my experience that thinking about some amusing aspect of a situation is often a very effective way of coping with problems.	,547	-,212	,194	,335
Q1. I usually don't laugh or joke around much with other people.	-,546	,238	,069	,298
Q13. I laugh and joke a lot with my closest friends.	,539	-,256	-,089	-,237
Q25. I don't often joke around with my friends.	-,535	,315	,090	,295
Q10. If I am feeling upset or unhappy I usually try to think of something funny about the situation to make myself feel better.	,524	-,150	,248	,455
Q19. Sometimes I think of something that is so funny that I can't stop myself from saying it, even if it is not appropriate for the situation.	,480	,143	-,262	,065
Q2. If I am feeling depressed, I can usually cheer myself up with humor.	,477	-,209	,124	,317
Q6. Even when I'm by myself, I'm often amused by the absurdities of life.	,471	-,224	,027	,135
Q32. Letting others laugh at me is my way of keeping my friends and family in good spirits.	,457	,416	,305	-,121
Q9. I rarely make other people laugh by telling funny stories about myself.	-,451	,100	,020	,224
Q28. If I am having problems or feeling unhappy, I often cover it up by joking around, so that even my closest friends don't know how I really feel.	,429	,147	,016	,054
Q12. I often try to make people like or accept me more by saying something funny about my own weaknesses, blunders, or faults.	,416	,371	,316	-,164
Q3. If someone makes a mistake, I will often tease them about it.	,407	,263	-,336	,041
Q29. I usually can't think of witty things to say when I'm with other people.	-,396	,256	,267	,203
Q22. If I am feeling sad or upset, I usually lose my sense of humor.	-,348	,109	-,018	-,186
Q20. I often go overboard in putting myself down when I am making jokes or trying to be funny.	,383	,587	,269	-,074
Q8. I will often get carried away in putting myself down if it makes my family or friends laugh.	,435	,508	,331	-,110
Q4. I let people laugh at me or make fun at my expense more than I should.	,362	,440	,281	-,082
Q16. I don't often say funny things to put myself down.	-,383	-,399	-,142	,109
Q24. When I am with friends or family, I often seem to be the one that other people make fun of or joke about.	,147	,398	,265	,031
Q31. Even if something is really funny to me, I will not laugh or joke about it if someone will be offended.	-,289	-,252	,565	-,095
Q15. I do not like it when people use humor as a way of criticizing or putting someone down.	-,315	-,315	,497	-,099
Q7. People are never offended or hurt by my sense of humor.	-,168	-,337	,416	-,190

Q11. When telling jokes or saying funny things, I am usually not very concerned about how other people are taking it.	,191	,238	-,387	,223
Q27. If I don't like someone, I often use humor or teasing to put them down.	,231	,254	-,379	,128
Q23. I never participate in laughing at others even if all my friends are doing it.	-,285	-,151	,378	-,080
Q18. If I'm by myself and I'm feeling unhappy, I make an effort to think of something funny to cheer myself up.	,467	-,188	,234	,479
Q30. I don't need to be with other people to feel amused – I can usually find things to laugh about even when I'm by myself.	,280	-,234	,053	,281

Extraction Method: Principal Axis Factoring.

a. 4 factors extracted. 5 iterations required.

Which factor rotation to use?

We might find that some factors are hard to interpret or that some items contribution to the factor is hard to assess if for example some items load on multiple factors to the same extent, or if most items load on a single factor and other factors only have high loadings from a handful of items. In this case (in fact in most cases in EFA) factor rotation can be helpful to improve the interpretability of the factors.

Factor rotation rotates the dimensions to achieve some goal. For example the most commonly used varimax rotation rotates the coordinate system to minimize the number of items with extreme loadings. This often produces factors that are easier to interpret than using no rotation.

You can set a factor rotation method in the Rotation menu. A rotation method is almost always used in EFA. There are two general types of rotations: orthogonal rotations and oblique rotations. In orthogonal rotation factors are not allowed to correlate with each other, while in oblique they are allowed to correlate.

You should chose the one that makes more sense theoretically (whether the latent factors should be correlated or not). If there is no theoretical bases to make a decision, usually the orthogonal rotations are preferred because they produce factors that are uncorrelated, thus, the factors become easier to tell apart from each other and interpret.

Orthogonal rotation options:

- **Varimax** rotation is the most commonly used. It minimizes the number of items with extreme loadings. This often produces factors that are easier to interpret compared with the other rotations.
- **Quantimax** is less often used. This rotation tries to minimize the number of factors needed to explain all the items well. It often creates a "general factor" on which all items have moderate to high loading, which is often hard to interpret.

Oblique rotation options:

- **Promax** rotation is the most commonly used because it is computationally faster and handles large datasets well compared to the alternatives.
- **Direct oblimin** is also an alternative, less commonly used, but still accepted in the literature.

Which items to keep in the analysis?

PCA and EFA tries to account for the variance within all variables that are entered into the analysis. According to the communalities table this is achieved better for some variables (the ones with higher post-extraction communalities) than for others. It is possible however that the factors underlying most observed variables are not related to the some of the observed variables in the dataset or that the common variance component of some observed variables is not high enough, and so they seem to be loading on all factors to a little bit. This makes it hard to interpret the factors, and it makes the factors less clear. Thus, sometimes it is a good idea to exclude some of the items from the analysis, to get more clear-cut factors and aid interpretability. This can be especially important if the sample size is small (<250) to create a good factor solution.

A rule of thumb by MacCallum et al. is that if the total sample size is smaller than 250, we should strive to achieve an average post-extraction communality of 0.6. In our example we have 993 valid cases, so we allow for lower average communality and still hope for a good factor solution.

Another indicator that we can use to decide whether to keep or exclude an item are the item loadings. We need to look at the Rotated factor matrix in the case of Orthogonal rotations or the Pattern matrix in the case of Oblique rotations for the item loadings. If there are items that load very poorly on all factors, we may consider dropping that item from factor analysis. Also, if our goal is to get well interpretable factors and factors that are well distinguished from each other, we might even consider dropping items that have a moderate but very similar loadings on multiple factors. This could improve the distinguishability of the factors from each other and may make the factors easier to interpret, but at the cost of excluding the variance of that variable from the factor model.

Naming the factors

In the end of the iterative process, we arrive at the final factor solution that we are satisfied with. This can take many ties. When we are done, the final task is to name the factors, that is, to give them a name or label that corresponds to their interpretation.

The names should be informative and should capture the main aspect in which the items loading onto a given factor are similar to each other. For example, the original authors devising the questionnaire found 4 factor, and named them: Affiliative humor, Self-enhancing humor, Aggressive humor, Self-defeating humor.

Factor analysis with categorical or ordinal data

PCA and EFA requires the observed variables to be continuous. Many cases however, this is not the case. In some cases we would like to include variables in factor analysis which are ordinal. For example, if we have Likert-scale measures where people have to rate some statement on a scale of e.g. 1-4 where they can only give discrete numbers as a response. If the scale has at least 5 levels (a scale of 1-5 for example) people would often do a regular factor analysis, just as if the data was continuous. But expert recommend that if ordinal variables are entered into factor analysis, the factor analysis should be done on the polychoric correlation matrix. By default, the factor analysis in most software use the pearson correlation matrix as a basis for factor analysis (this is built into the factor procedure in SPSS as well). SPSS does not have a readily available option to use the polychoric correlation matrix in factor analysis. So if you have ordinal data, you have the following options:

- Use the regular factor analysis procedure in SPSS, and report in the limitations section of your research report that the variables entered into the factor analysis are ordinal in nature. The researchers in the current project used the Pearson correlation matrix in factor analysis, but with ordinal data the Polychoric correlation matrix returns a more accurate result. This is not ideal and would probably not be accepted in most research journals, unless the ordinal variable has many levels (5+), but even then, the reviewers might ask you to do the analysis with the Polychoric correlation matrix.

- Use another software for data analysis. There are very good online tutorials for using factor analysis in other software, for example using the free software FACTOR: https://www.youtube.com/watch?v=X-Y2UXg9z5I

- There is an R extension for SPSS, and using that extension you can utilizes the great flexibility of R from SPSS. Here is a paper about an extension specifically designed to do factor analysis in SPSS using the R extension: https://www.jstatsoft.org/article/view/v046i04

- SPSS also has a comment about doing factor analysis using ordinal and categorical data, which is available on this link: https://www.ibm.com/support/pages/exploratory-factor-analysis-categorical-variables