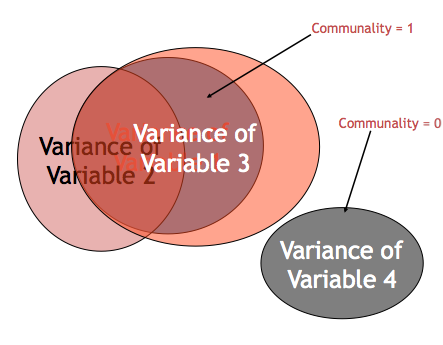
PCA and EFA

**Description:**

* Both PCA and EFA are descriptive analyses used to understand the underlying pattern in the data. They both group variables together based on the high correlations between patterns of answers on those variables. They are used for data reduction.
* PCA: Principle Components Analysis
  + Components are combinations of correlated variables, and the variables are thought to *cause* the components.
  + Components are produced, which is the term to use when writing up.
  + All the variance in the variables in analyzed, therefore the *total* variance is used.
* EFA: Exploratory Factor Analysis – describes the data and summarizes “factors”, often used as a first step on a scale or data.
  + Factors are thought to *cause* the variables, and the underlying construct is what creates the scores on each variable.
  + Factors are produced, which is the term to use when writing up.
  + Only shared variance and *unique* variance is analyzed, the left over variance is considered error.

**Definitions/Abbreviations:**

* Variance Types
  + Common variance = overlapping variance between items (systematic variance)
  + Unique variance = variance only related to that item (error variance)
  + Communality – the common variance for the item
    - You can think of it as R2 for that item
  + EFA = describes the common variance
  + PCA = describes common variance + unique variance



* Correlation Matrices
  + Observed correlation matrix – the correlations between all of the variables (very similar to doing a bivariate correlation chart).
  + Reproduced correlation matrix – correlation matrix created from the factors created.
  + Residual correlation matrix – the difference between original and reproduced correlation matrix. This matrix will be very small if you had a good fit for your model. The residual matrices are used to calculate how well the analysis went for you.
* Eigenvalues – A mathematical representation of the variance accounted for by that grouping of items
  + Confusing part: You will see the number of eigenvalues as you have items because they are calculated before extraction.
  + Only a few should be large.

**Research Questions:**

* Number of underlying patterns (factors/components): How many best fit the data?
  + Does this match the expected theory?
* Scale development: building a new measure, does it match your expected theory? Does it measure what you are expecting it to measure?
  + What are the underlying pieces? How do the questions group together?
  + What questions can we eliminate as not being important?

**Power:**

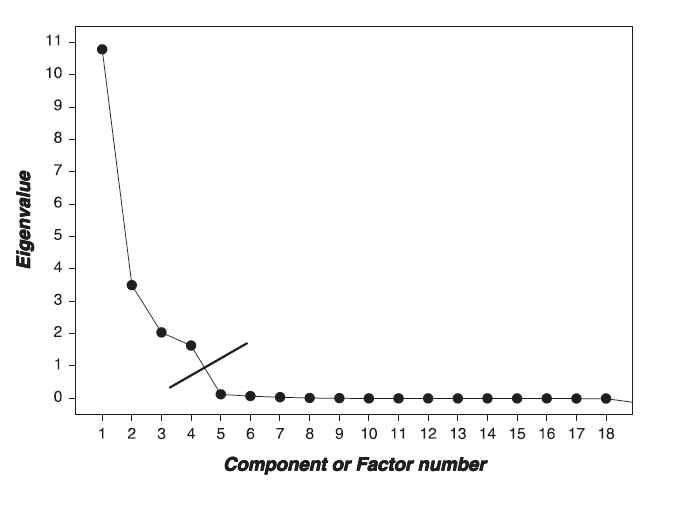
* Generally, power is a problem of sample size, since PCA/EFA are testing model fit. Either your model is going to fit in an expected way or not. If power is a concern, the solution is to test more participants.
* Large sample sizes are needed for either analysis, and usually scales are tested several times. If you have a large dataset, people will often randomly split them to get two tests of the model as well.
* Rules of thumb:
  + 10-15 participants per item
  + <100 is not acceptable (believe me, I know this from experience).
  + 300 is generally agreed upon as the best; however, most people see it as the gold criteria and are ok with less.

**Assumptions:**

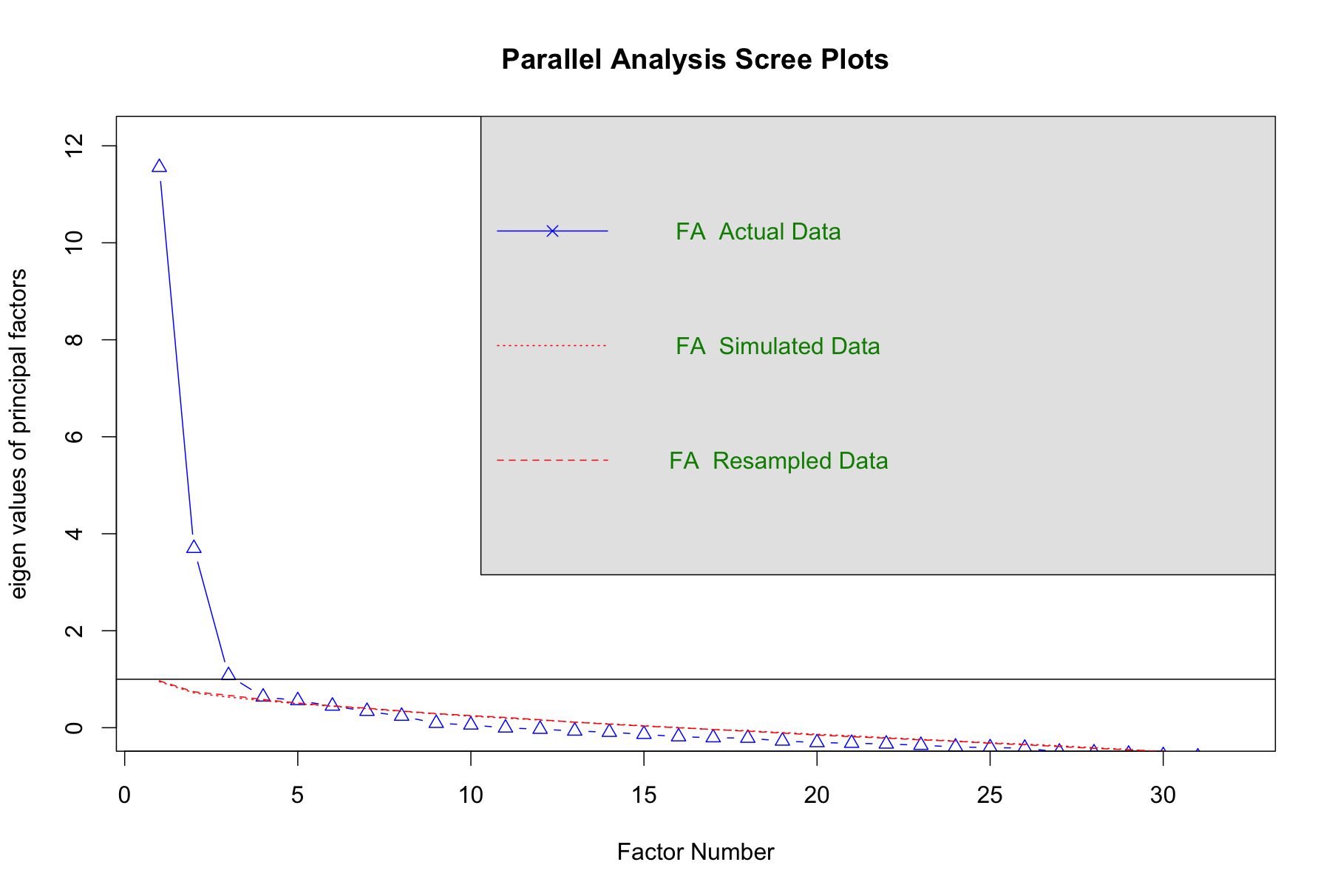
1. Variables:
   1. Number of variables – EFA/PCA group variables into factors/components, so only using 5 variables does not allow you to create those clusters easily. At least 10 variables are recommended.
   2. Types of variables – without running special types of analysis, interval or ratio data are recommended for EFA/PCA.
2. Sample size – see above under power.
3. Normal data screening:
   1. Screen: Accuracy, missing, outliers
   2. Assumptions: Additivity\*, Linearity, Normality, Homogeneity, and Homoscedasticity
   3. \* Additivity is checked to make sure that variables aren’t perfectly correlated so the analysis will run. You expect variables to be highly correlated, as that is the point of the analysis.

**Rules/Questions to ask yourself:**

1. **Do I have a good dataset for EFA/PCA?**
   1. Correlation adequacy – do I have strong enough correlations to be able to group them together into meaningful factors or components?
      1. Use Bartlett’s test of correlation adequacy. Significant values (*p* < .05) are good.
   2. Do I have a large enough sample?
      1. Look at the number of participants. Do you have at least 100? 10-15 per item?
      2. You can also use the Kaiser-Meyer-Olkin test where you want to find high values close to 1, but at least above .70.
2. **How many factors/components do I have?**
   1. Theory – You might have a theory on how many factors you expect from the scale. If you do have a theory, people generally run that many factors and then two more (1 more factor, 1 less factor).
   2. Scree Plots – scree plots are a visual depiction of the eigenvalues. You will look for the large drop off to figure out how many to use:



* 1. Parallel Analysis – this analysis tells you how many factors are greater than chance, which you can use in combination with a scree plot to look at the number of factors.
     1. The dark line is set at one, which is part of the Kaiser criterion listed below.
     2. The red dotted line is the random data set used to test this analysis. Your data is randomly reordered to see how many factors are better than chance.
     3. The blue line and triangles are your eigenvalues from the real dataset.
     4. You want to look at where the blue and red lines cross.



* 1. Kaiser criterion – this method is an older rule of thumb that is not well supported anymore. You would look at the number of eigenvalues that are greater than 1 (or .70 in new literature). This rule tends to overestimate the number of factors/components needed.

1. **Can I achieve *simple structure*?**
   1. Simple structure is the final solution of a factor analysis that has the simplest solution.
   2. **How can I get to simple structure (how to set up the analysis)?**
      1. Factor rotation – process by which the solution is made “better” (smaller residuals) without changing the mathematical properties.
         1. **Oblique** – Oblimin is the most common rotation. Factors are allowed to be correlation when they are rotated.
         2. Orthogonal – Varimax is the most common rotation. Orthogonal rotation holds factors completely uncorrelated.
      2. Fitting estimation = MATH that is used to determine factor loadings.
         1. For EFA – maximum likelihood is the most common fitting estimation.
         2. For PCA – principle components is the most common fitting estimation.
         3. Half-half – principle axis factoring is the type of fitting estimation that’s sort of both analyses.



* 1. **How do I tell if my set up achieved simple structure?**
     1. Variable Loading - variables “load” on a factor when they have a value over >.300.
     2. You want variables to load onto only one factor (and only one; hence the simple solution name).
     3. Split variables – you want to get rid of variables that load onto two or more factors.
     4. Non-Loading Variables – you want to get rid of variables that don’t load on any factor.
     5. Factors/Components with only one/two items loading onto it are considered *unique.* Three to four items are suggested for each factor/component.
     6. What to do if bad items?
        1. In this step you might run **several rounds** of analyses. Find the bad items, run the EFA/PCA again without them.

1. **How adequate is my model?**
   1. So is that simple structure any good? This step is akin to checking *p*-values and effect sizes to determine if that structure is appropriate (i.e. your means might be different in ANOVA, but you cannot say that unless it is significant or has a large effect size).
   2. Fit indices – a measure of how well the rotated matrix matches the original matrix
      1. Goodness of fit statistics – want large values, compares reproduced correlation matrix to real correlation matrix
      2. Residual statistics – want small values, look at the residual matrix (i.e. reproduced – real correlation table)

Goodness of fit statistics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fit** | **Name** | **Excellent** | **Acceptable** | **Poor** |
| NNFI/TLI | Non-normed fit index, Tucker-Lewis index | >.95 | >.90 | <.90 |
| CFI | Comparative fix index | >.95 | >.90 | <.90 |

Residual fit statistics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fit** | **Name** | **Excellent** | **Acceptable** | **Poor** |
| RMSEA | Root mean square error of approximation | <.06 | .06-.08 | >.10 |
| RMSR | Root mean square of the residual | <.06 | .06-.08 | >.10 |

* 1. Reliability – an estimate of how much your items “hang together” and might replicate
     1. Cronbach’s alpha most common
     2. .70 or .80 is acceptable
  2. Theory –Do the item loadings make any sense? Can you label the factor/component?

# Complete Example

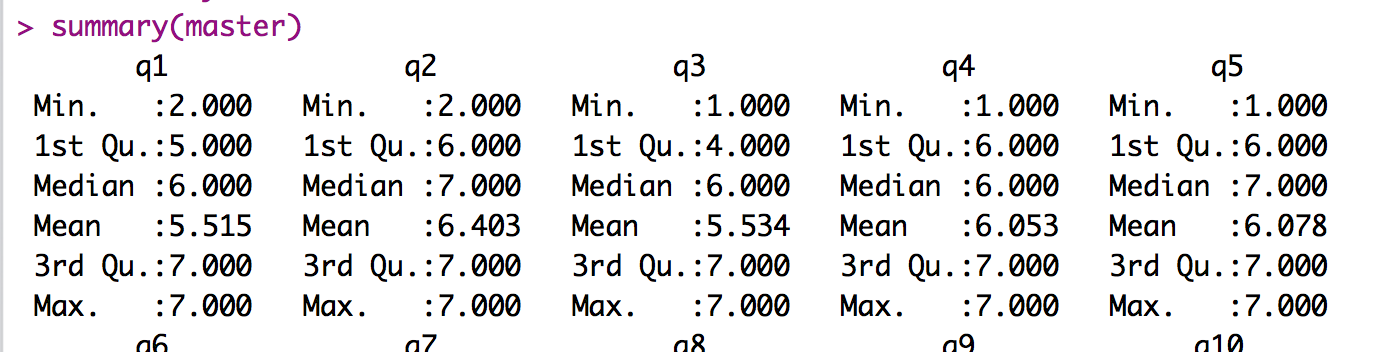
# Exploratory Factor Analysis

**Research Question:**

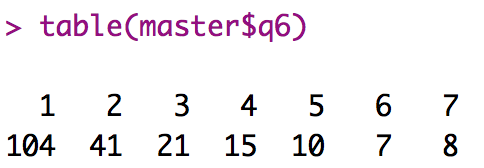
Many people are interested in self-determination theory – a theory of motivation that “is concerned with supporting our natural or intrinsic tendencies to behave in effective and healthy ways”. In line with this idea, a scale for assessing why students are in college was developed. Here we will test if those questions fit a factor structure. See the attached guide for the list of questions used.

**Assumptions:**

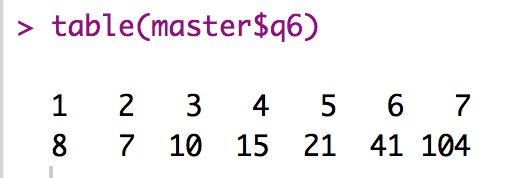
1. Accuracy:
   1. Use the summary(*dataset name*) function to get the basic information for the data.
   2. Let’s check out minimum and maximum.
      1. All of our questions are 1-7, so it seems ok.
   3. Check for any reverse scored items.
      1. Reverse scored items are ones that are worded so that high scores match low scores on other questions. For example:
         1. Because I experience pleasure and satisfaction while learning new things.
         2. I wonder what I’m doing in college; actually I find it boring.
      2. The second question is negatively worded, so we would want to switch it to be positive if they scored low on that question.
      3. There are recode functions in R, but the easiest thing to do, is to think about the scaling.
      4. The rule is:
         1. Max score + minimum – participant score
         2. So, in our example, 7 + 1 – participant score
         3. If a person scored a 7 (8 – 7 = 1), if a person scored a 4 (8 – 4 = 4), if a person scored a 1 (8 – 1 = 7).
      5. On the scale, we see that 6, 9, 17, and 27 are reverse coded.
      6. Let’s reverse code them!



* 1. I looked at a table to make sure what the distribution was before I started.



* 1. Then I ran the following:
     1. master[ , c(*column numbers to reverse*)] = 8 - master[ , c(*column numbers to reverse*)]
     2. Because R uses matrix algebra, it will subtract 8 – participant score for each column and row.
  2. Here’s a table to show what happened:

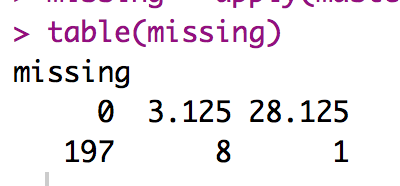


1. Missing data:
   1. Because we tend to have large datasets and many columns, EFA is a great place to replace missing data.
   2. First, subset out participants that have too much missing data:

percentmissing = function (x){ sum(is.na(x))/length(x) \* 100}

missing = apply(*dataset*, 1, percentmissing)

table(missing)

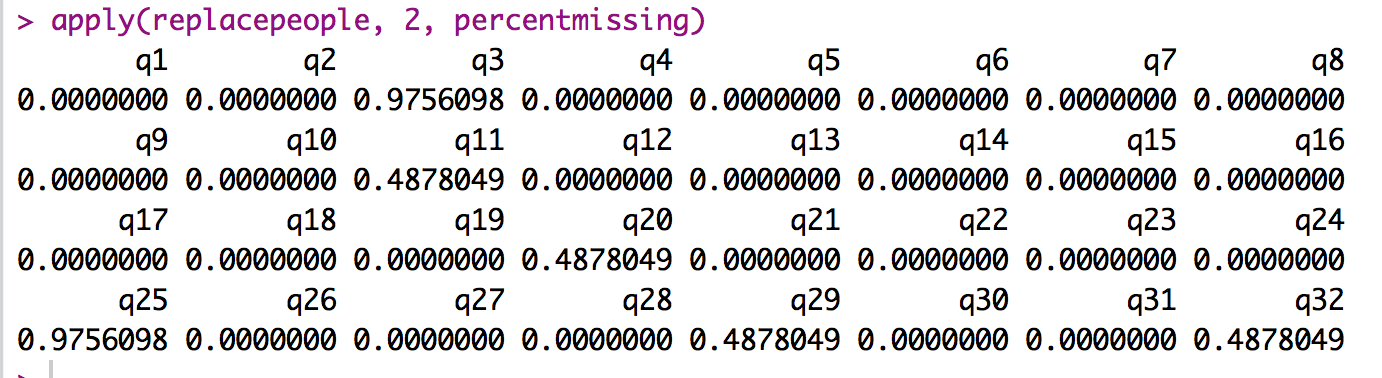


* 1. We have 8 participants we can replace, and one to get rid of because they skipped a lot of the scale.

replacepeople = subset(*dataset*, missing < = 5)

* 1. Then we should make sure we don’t have too much missing by column.

apply(replacepeople, 2, percentmissing)



* 1. Remember these values are percentages, so all of these are less than 5%.
  2. Mice the data!

library(mice)

tempnomiss = mice(replacepeople)

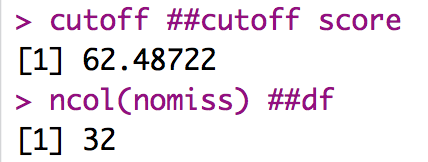
nomiss = complete(tempnomiss, 1)

1. Outliers:
   1. We have 32 columns here to check for weird patterns of scores – remember that Mahalanobis allows us to figure out if someone’s pattern of data is strange to eliminate them.
   2. Create the Mahalanobis values:
      1. mahal = mahalanobis(*dataset*,

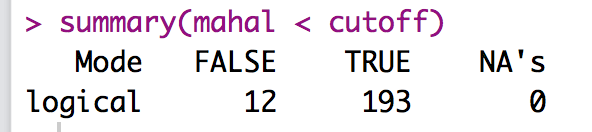
colMeans(*dataset*, na.rm = T),

cov(*dataset*, use = “pairwise.complete.obs”))

* 1. Create the cut off score:
     1. cutoff = qchisq(1-.001, ncol(*dataset*))
  2. Remember you can use:
     1. cutoff to get the cutoff score
     2. ncol(*dataset*) to get the *df*



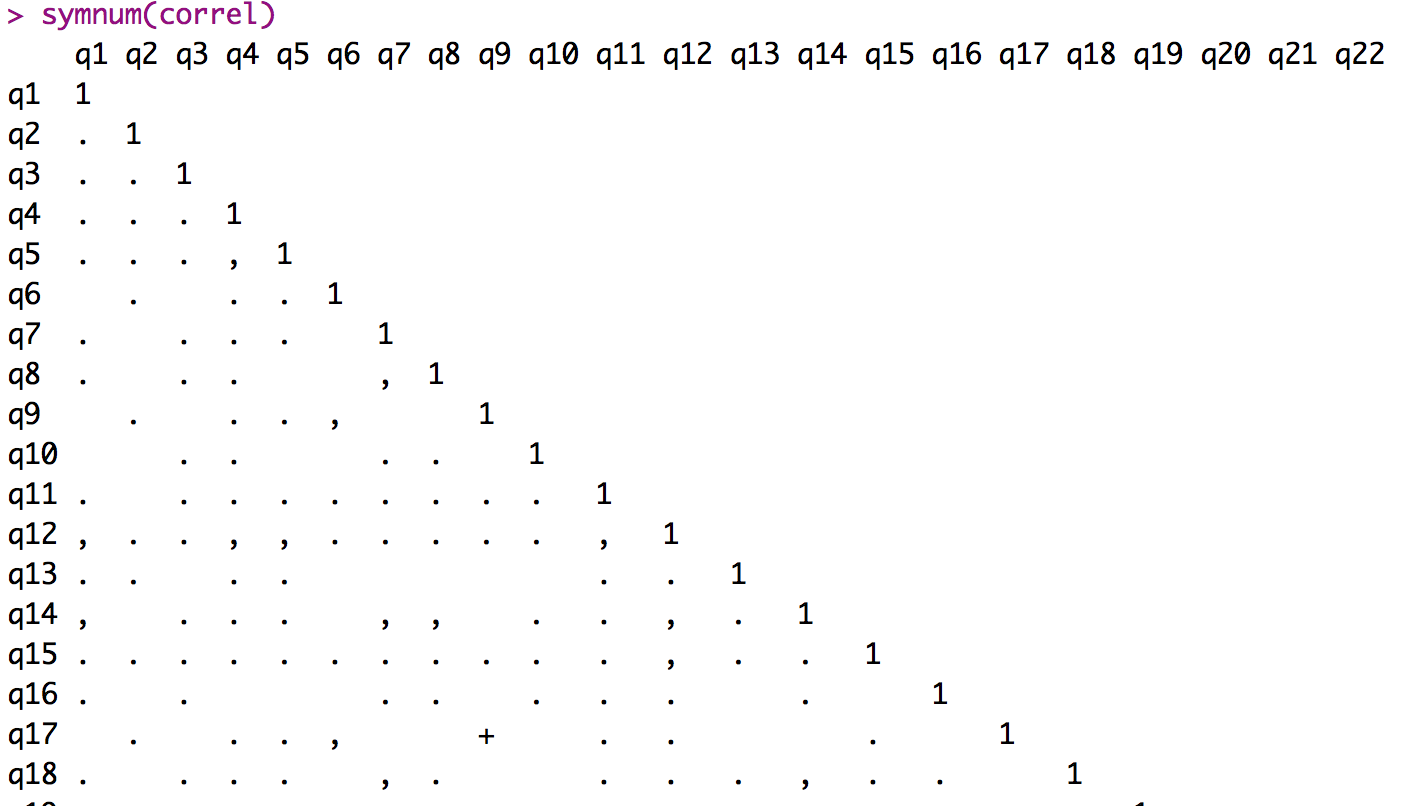
* 1. See how many outliers you have:
     1. summary(mahal < cutoff)



* + 1. Remember FALSE is bad.
    2. I have 12 outliers!
       1. Exclude them – however, they don’t often make a huge difference in these types of analyses if you need the sample size.

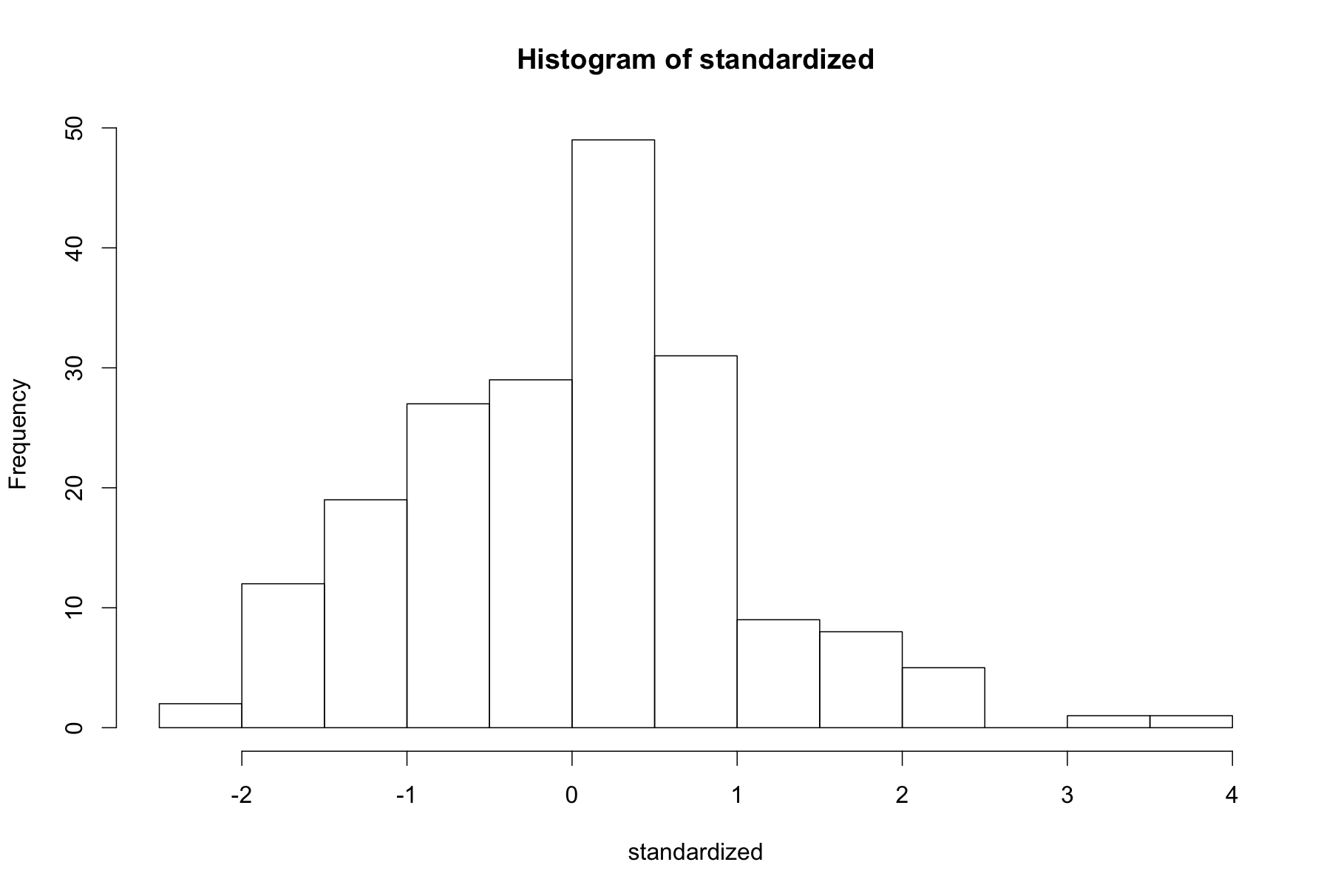
noout = subset(nomiss, mahal < cutoff).

1. Additivity
   1. In general, you *want* the questions to be correlated – that’s the idea behind EFA, is that they are all tapping into the same construct.
   2. However, they cannot be perfectly correlated or the EFA will not run.
   3. Mainly we are checking that we don’t get any 1s other than the diagonal in our symbols chart. So, basically, the rule is the *r* < .999.
   4. Get the correlations:
      1. correl = cor(*dataset*, use = “pairwise.complete.obs”)
   5. Get the symbols chart:
      1. symnum(correl)
   6. Look for 1s NOT on the diagonal:

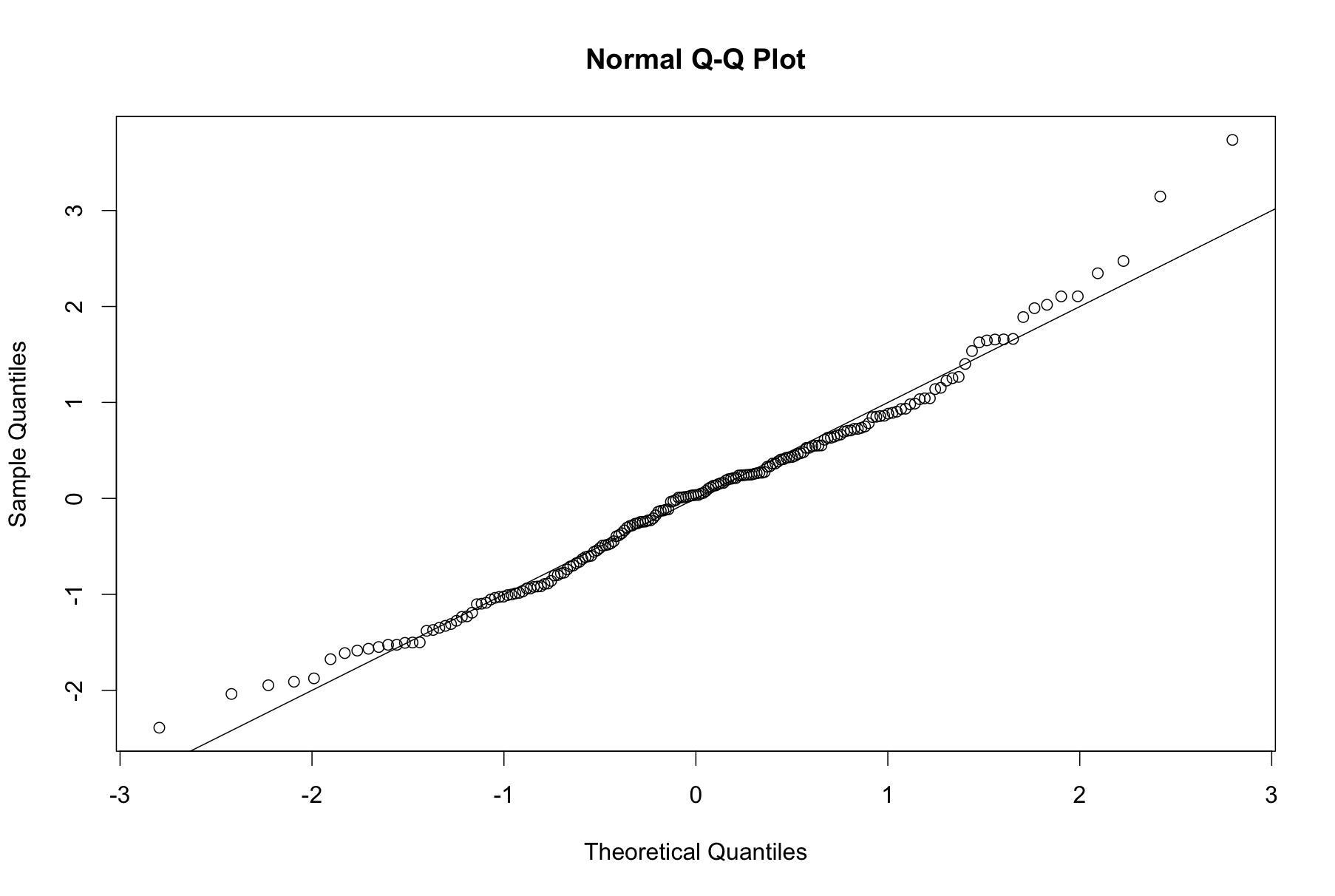


* 1. The ones marked in blue are ok, they are the column correlated with itself (which should be 1).
  2. So, our numbers appear ok.

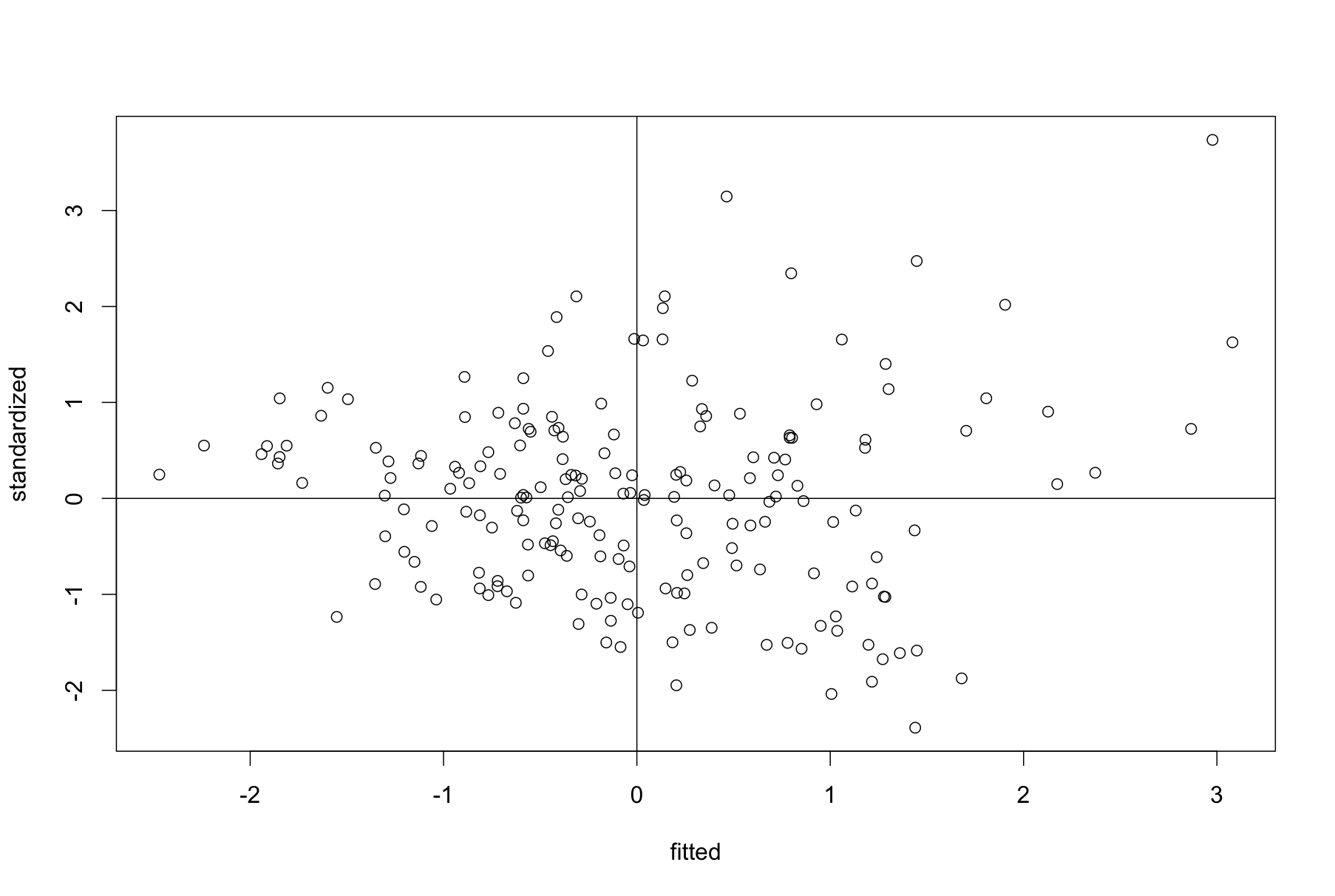
1. Set up the rest of the assumptions – we must use a fake regression analysis because, while EFA is regression to the extreme, you still have to screen it with a regular regression analysis.
   1. Make a random variable:
      1. random = rchisq(nrow(*dataset*), 7)
   2. Run a fake regression:
      1. fake = lm(random~., data = *dataset*)
   3. Create the standardized residuals:
      1. standardized = rstudent(fake)
   4. Create the fitted values:
      1. fitted = scale(fake$fitted.values)
2. Normality:
   1. hist(standardized)
   2. Most of the data is between -2 and 2 and is centered over 0 – still a couple wonky people, but it’s ok.
   3. Because we have more than 30 people, we do not have to worry because of the central limit theorem.



* 1. Linearity:
     1. qqnorm(standardized)
     2. abline(0,1)
     3. This graph looks good!



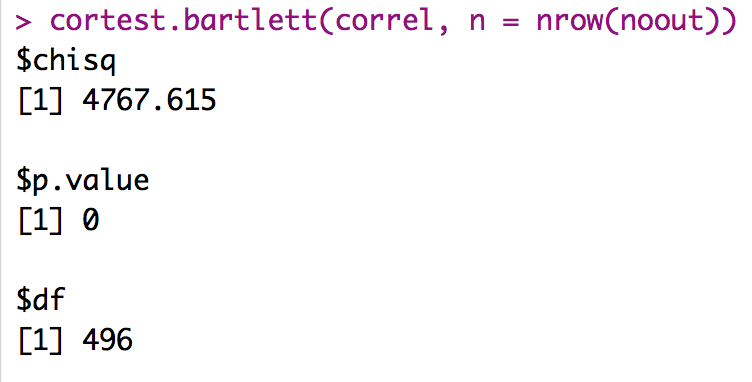
* 1. Homogeneity:
     1. plot(fitted,standardized)
     2. abline(0,0)
     3. abline(v = 0)
     4. The data is ok – not great but ok.
  2. Homoscedasticity:
     1. Yeah, no.
     2. There is a definite megaphone shape to the data – which might be explain by the different factors clumping together, but the assumption test with all the questions involved shows heteroscedasticity.



**EFA:**

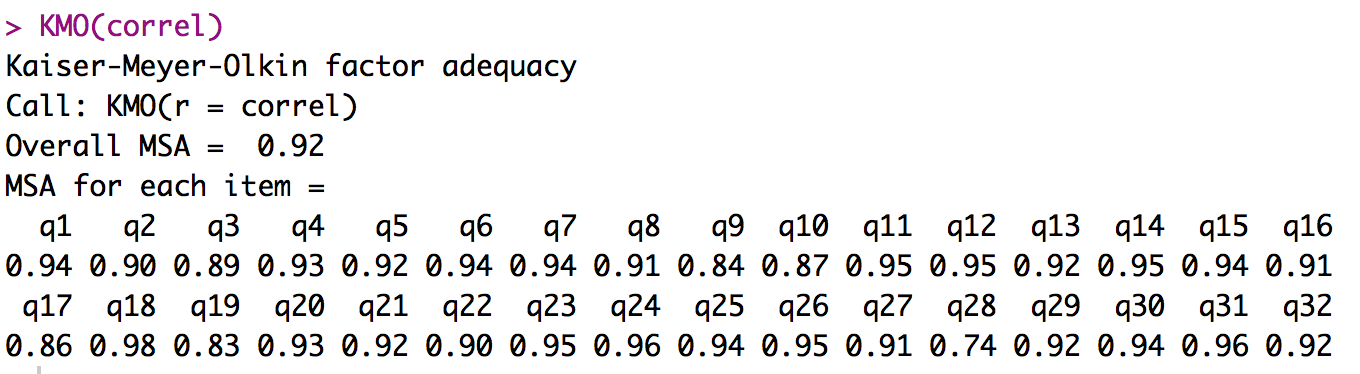
**Do I have a good dataset for EFA?**

1. Checking variables and sample size:
   1. Number of variables – 32 questions, so we are good.
   2. Types of variables – 1 to 7 Likert scales, which are at least interval.
   3. Sample Size – 193 – also at least 100, not quite 320 for 10 for each item.
2. Load the psych and GPArotation libraries.
3. Correlation adequacy:
   1. Bartlett’s test determines if your correlations are large enough for EFA.
   2. cortest.bartlett(correl, n = nrow(*dataset*))
      1. Remember, we saved correl earlier in data screening.



* 1. Bartlett’s test was significant *X2*(496) = 4767.62, *p* < .001.
  2. That results implies we have large enough correlations for EFA.

1. Sampling adequacy:
   1. KMO (Kaiser-Meyer-Olkin) test determines if you have a good sample for EFA. You want high values close to one.
   2. KMO(correl)

****

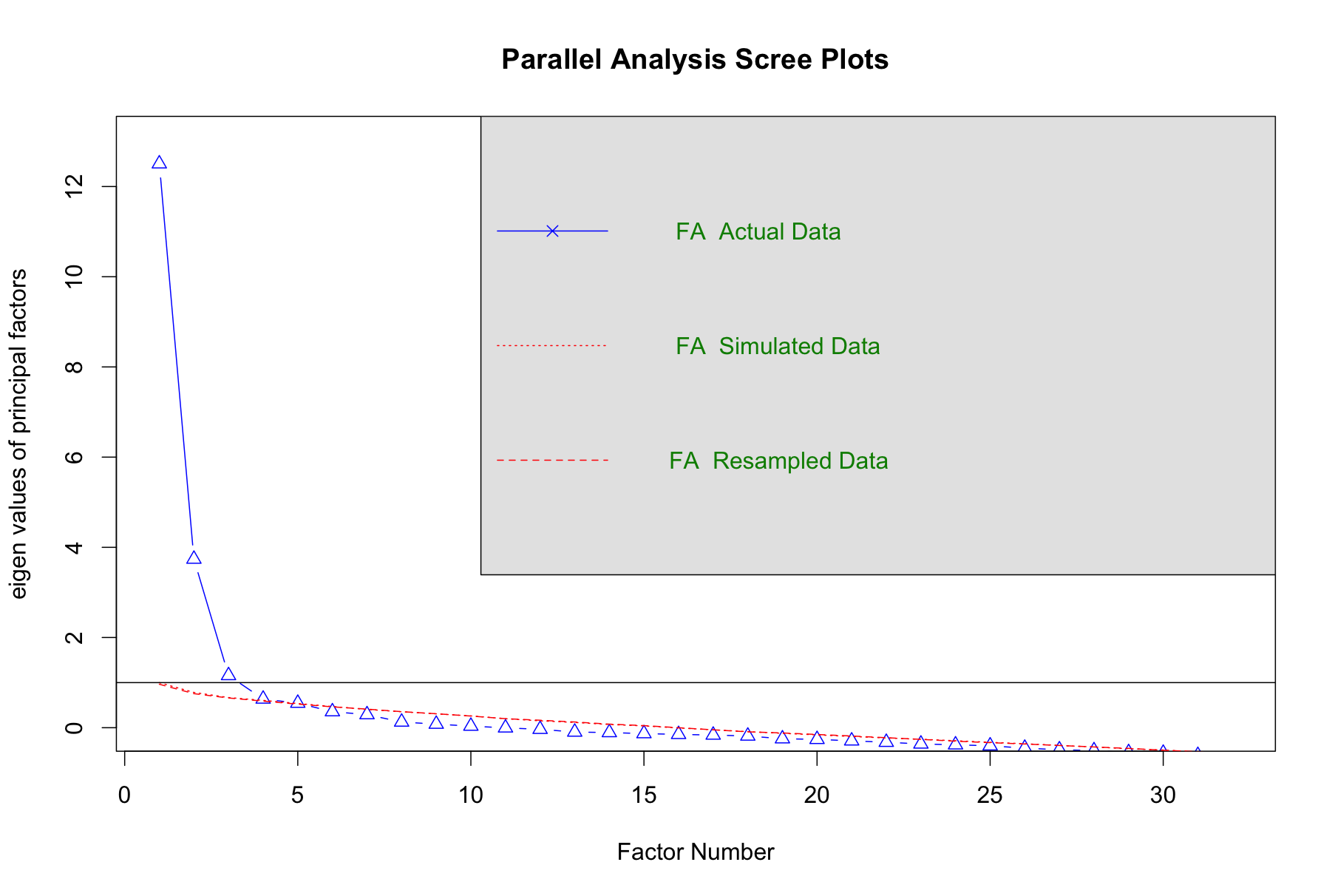
* 1. The mean sampling adequacy (MSA) was .92, which is a good score.

**How many factors/components do I have?**

1. Theory: self-determination theory suggests three factors.
2. Parallel: You will want to use a parallel analysis to determine the number of factors. This analysis will also make a scree plot for you to examine.
   1. Function: fa.parallel()
      1. Arguments = (*dataset*, fm = “ml”, fa = “fa”).
      2. FM = type of math you are going to use – ml = maximum likelihood
      3. FA = type of analysis you are going to use – fa = factor analysis
   2. The parallel analysis suggests five but the scree plot appears to be three.

nofactors = fa.parallel(noout, fm="ml", fa="fa")

Parallel analysis suggests that the number of factors = 5 and the number of components = NA



1. Kaiser criterion:
   1. We can sum up the number of eigenvalues over 1 (old rule) and .7 (new rule).
      1. sum(nofactors$fa.values > 1.0) ##old kaiser criterion
      2. sum(nofactors$fa.values > .7) ##new kaiser criterion
      3. Both of these indicators suggest three factors.
2. What to do when they don't agree?
   1. Try a couple of models (i.e. 3 factors compared to 5 factors) and see which one fits the best (see below).
   2. Here we have theory (3), parallel (5), scree (3), and Kaiser (3). Three seems to be the common ground.

**Can I achieve simple structure?**

1. Set up for simple structure:
2. You are actually going to run the factor analysis at this point.
   1. You want to use the fa() function.
   2. Arguments: (data set name, nfactors = #, rotate = “type”, fm= “type”)
      1. Nfactors = how many ever factors you picked from the step above.
      2. Rotate = the most popular type is oblimin, but there are many options (see the help page for all the options).
      3. FM = type of math you are going to use, try ml for maximum likelihood to start.
      4. A lot of times, you do not have to do this = that, but since fa has many arguments you may not want to use, be sure to leave in the rotate = type and such.
   3. You don’t have to save the function, but I like to save them as a variable, so I can get the output in smaller pieces if I wanted to.

round1 = fa(noout, nfactors=3, rotate = "oblimin", fm = "ml")

1. Did I get simple structure?
   * 1. Let’s look at the output. The first thing you get is the standardized loadings (which is what you want to check for variables to “load” onto a factor).
     2. You will get the factors listed by the name of the rotation (hence, ML1, ML2) but they are ordered by the size of the factor. So factor 3 takes up more variance than factor 2.
     3. H2 = communality – the amount of variance accounted for by factors.
     4. U2 = uniqueness – the amount of variance not accounted for by factors.
        1. These values should add up to 1 because we can only have 100 percent variance.
     5. Com = item complexity (not used).
2. So I want to go through and figure out which variables have loaded on: ONE and only one factor with at least .300 loading.

Factor Analysis using method = ml

Call: fa(r = noout, nfactors = 3, rotate = "oblimin", fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix

ML1 ML3 ML2 h2 u2 com

q1 0.54 0.25 0.10 0.53 0.47 1.5

q2 -0.06 0.52 0.17 0.36 0.64 1.2

q3 0.62 0.03 0.02 0.41 0.59 1.0

q4 0.32 0.37 0.21 0.49 0.51 2.6

q5 0.22 0.51 0.19 0.56 0.44 1.7

q6 0.01 0.01 0.78 0.63 0.37 1.0

q7 0.84 -0.06 -0.07 0.65 0.35 1.0

q8 0.75 -0.06 -0.04 0.53 0.47 1.0

q9 0.01 -0.04 0.95 0.88 0.12 1.0

q10 0.52 0.03 -0.05 0.27 0.73 1.0

q11 0.54 0.14 0.27 0.55 0.45 1.6

q12 0.54 0.29 0.27 0.72 0.28 2.1

q13 -0.06 0.91 -0.12 0.70 0.30 1.0

q14 0.74 0.12 0.05 0.66 0.34 1.1

q15 0.35 0.39 0.22 0.56 0.44 2.6

q16 0.79 -0.18 -0.16 0.55 0.45 1.2

q17 -0.02 -0.07 0.93 0.81 0.19 1.0

q18 0.71 0.11 0.03 0.59 0.41 1.1

q19 0.02 0.71 -0.13 0.44 0.56 1.1

q20 0.77 0.08 0.07 0.69 0.31 1.0

q21 0.84 -0.17 -0.19 0.62 0.38 1.2

q22 0.74 -0.08 0.05 0.51 0.49 1.0

q23 0.63 0.24 0.15 0.68 0.32 1.4

q24 0.46 0.23 0.20 0.48 0.52 1.9

q25 0.68 0.21 0.18 0.75 0.25 1.3

q26 0.14 0.63 0.21 0.68 0.32 1.3

q27 -0.11 0.03 0.74 0.55 0.45 1.0

q28 -0.07 0.62 -0.12 0.30 0.70 1.1

q29 0.64 0.09 0.02 0.48 0.52 1.0

q30 0.07 0.78 0.04 0.69 0.31 1.0

q31 0.51 0.13 0.09 0.38 0.62 1.2

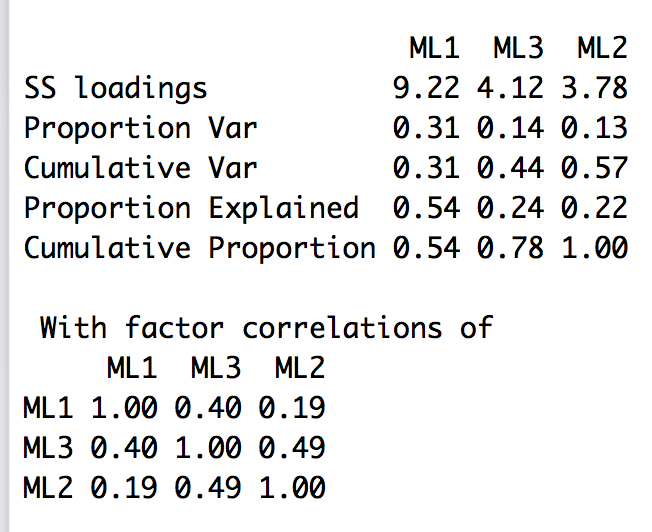
q32 0.74 -0.12 -0.19 0.49 0.51 1.2

|  |  |  |  |
| --- | --- | --- | --- |
|  | ML1 | ML3 | ML2 |
| q1 | 0.54 | 0.25 | 0.1 |
| q2 | -0.06 | 0.52 | 0.17 |
| q3 | 0.62 | 0.03 | 0.02 |
| q4 | 0.32 | 0.37 | 0.21 |
| q5 | 0.22 | 0.51 | 0.19 |
| q6 | 0.01 | 0.01 | 0.78 |
| q7 | 0.84 | -0.06 | -0.07 |
| q8 | 0.75 | -0.06 | -0.04 |
| q9 | 0.01 | -0.04 | 0.95 |
| q10 | 0.52 | 0.03 | -0.05 |
| q11 | 0.54 | 0.14 | 0.27 |
| q12 | 0.54 | 0.29 | 0.27 |
| q13 | -0.06 | 0.91 | -0.12 |
| q14 | 0.74 | 0.12 | 0.05 |
| q15 | 0.35 | 0.39 | 0.22 |
| q16 | 0.79 | -0.18 | -0.16 |
| q17 | -0.02 | -0.07 | 0.93 |
| q18 | 0.71 | 0.11 | 0.03 |
| q19 | 0.02 | 0.71 | -0.13 |
| q20 | 0.77 | 0.08 | 0.07 |
| q21 | 0.84 | -0.17 | -0.19 |
| q22 | 0.74 | -0.08 | 0.05 |
| q23 | 0.63 | 0.24 | 0.15 |
| q24 | 0.46 | 0.23 | 0.2 |
| q25 | 0.68 | 0.21 | 0.18 |
| q26 | 0.14 | 0.63 | 0.21 |
| q27 | -0.11 | 0.03 | 0.74 |
| q28 | -0.07 | 0.62 | -0.12 |
| q29 | 0.64 | 0.09 | 0.02 |
| q30 | 0.07 | 0.78 | 0.04 |
| q31 | 0.51 | 0.13 | 0.09 |
| q32 | 0.74 | -0.12 | -0.19 |

1. So, it appears that these questions are bad: Split: 4, 15.
2. So I would want to drop them from the analysis, and rerun it.

|  |  |  |  |
| --- | --- | --- | --- |
|  | ML1 | ML3 | ML2 |
| q1 | 0.55 | 0.22 | 0.11 |
| q2 | -0.04 | 0.52 | 0.17 |
| q3 | 0.62 | 0.03 | 0.02 |
| q5 | 0.25 | 0.47 | 0.21 |
| q6 | 0.02 | 0.01 | 0.78 |
| q7 | 0.84 | -0.06 | -0.07 |
| q8 | 0.75 | -0.06 | -0.04 |
| q9 | 0.01 | -0.03 | 0.95 |
| q10 | 0.52 | 0.02 | -0.05 |
| q11 | 0.56 | 0.12 | 0.28 |
| q12 | 0.56 | 0.26 | 0.29 |
| q13 | -0.04 | 0.91 | -0.11 |
| q14 | 0.75 | 0.11 | 0.05 |
| q16 | 0.79 | -0.18 | -0.17 |
| q17 | -0.01 | -0.06 | 0.93 |
| q18 | 0.71 | 0.11 | 0.03 |
| q19 | 0.02 | 0.73 | -0.13 |
| q20 | 0.78 | 0.08 | 0.07 |
| q21 | 0.83 | -0.16 | -0.2 |
| q22 | 0.73 | -0.07 | 0.04 |
| q23 | 0.65 | 0.21 | 0.17 |
| q24 | 0.47 | 0.22 | 0.21 |
| q25 | 0.69 | 0.2 | 0.19 |
| q26 | 0.16 | 0.61 | 0.23 |
| q27 | -0.1 | 0.03 | 0.74 |
| q28 | -0.06 | 0.63 | -0.12 |
| q29 | 0.64 | 0.1 | 0.02 |
| q30 | 0.09 | 0.76 | 0.06 |
| q31 | 0.52 | 0.13 | 0.09 |
| q32 | 0.74 | -0.12 | -0.19 |

1. Now, we see that everything loads on one and only factor, which is good.
2. Here’s some other important output right here that you’ll want to use, but **only the final round** of analyses:



1. The first part tells you how much variance each factor accounted for:
   * 1. ML1 = 0.31
     2. ML3 = 0.14
     3. ML2 = 0.13
2. The second part tells you how correlated your factors were, where ML2 and ML3 have a strong relationship *r* = 0.49.

**Is the solution adequate?**

1. Most of the numbers you’ll need for adequate solutions are in the output you get automatically with the fa() function.
2. Fit indices
3. The root mean square of the residuals (RMSR) is 0.05
   1. Tucker Lewis Index of factoring reliability = 0.828
   2. RMSEA index = 0.095 and the 90 % confidence intervals are 0.082 0.097
   3. CFI: 1 – ((yours χ2 – df) / (null χ2 – df))
      1. dof = degrees of freedom for your current model
      2. STATISTIC = chi square for your current model
      3. null.dof = degrees of freedom for your null model
      4. null.chisq = chi square for your null model
   4. You could use this function:
      1. 1 - ((finalmodel$STATISTIC-finalmodel$dof)/(finalmodel$null.chisq-finalmodel$null.dof))
   5. You can tell if you are close because it should be slightly better than TLI.

> 1 - ((finalmodel$STATISTIC-finalmodel$dof)/(finalmodel$null.chisq-finalmodel$null.dof))

[1] 0.8638744

1. Are the fit indices ok?
2. TLI/CFI – not really.
3. RMSR (excellent) and RMSEA (acceptable).
4. Reliability – the psych package has a Cronbach’s alpha function – alpha().
   1. You can first save each factor as a separate set of column numbers based on which column they loaded onto. For example, question 1 is factor 1, which is column 1.
   2. Then run psych::alpha(*dataset*[ , factor1]).

factor1 = c(1, 3, 7, 8, 10:12, 14, 16, 18, 20:25, 29, 31, 32)

factor2 = c(2, 5, 13, 19, 26, 28, 30)

factor3 = c(6, 9, 17, 27)

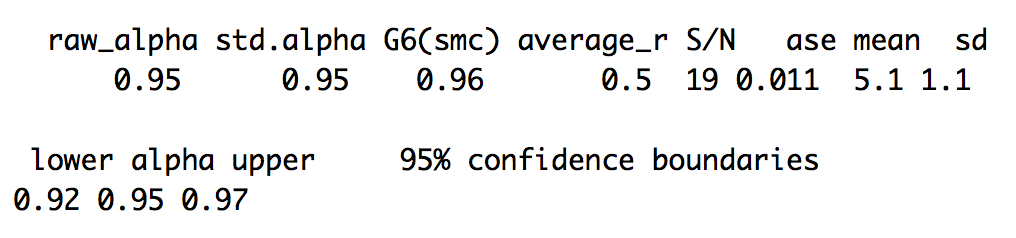
psych::alpha(noout[ , factor1])

psych::alpha(noout[ , factor2])

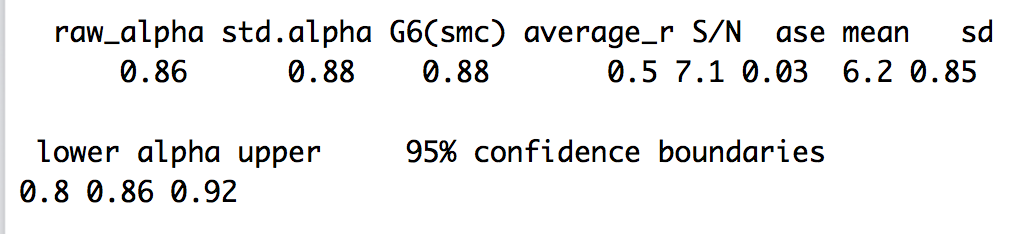
psych::alpha(noout[ , factor3])

* 1. Separating them out allows you to create factor scores as well, without typing the c(*numbers*) part over and over.
  2. Alpha creates a lot of output, but here’s the important part:

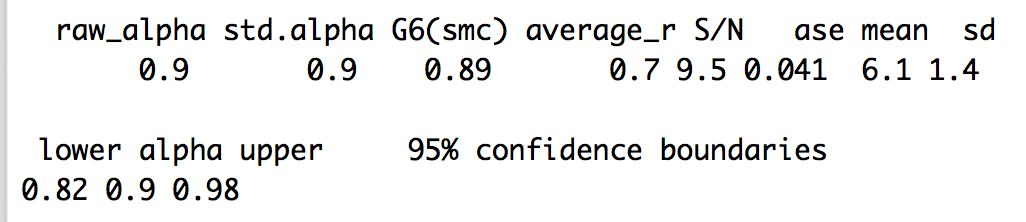
Factor 1:



Factor 2:



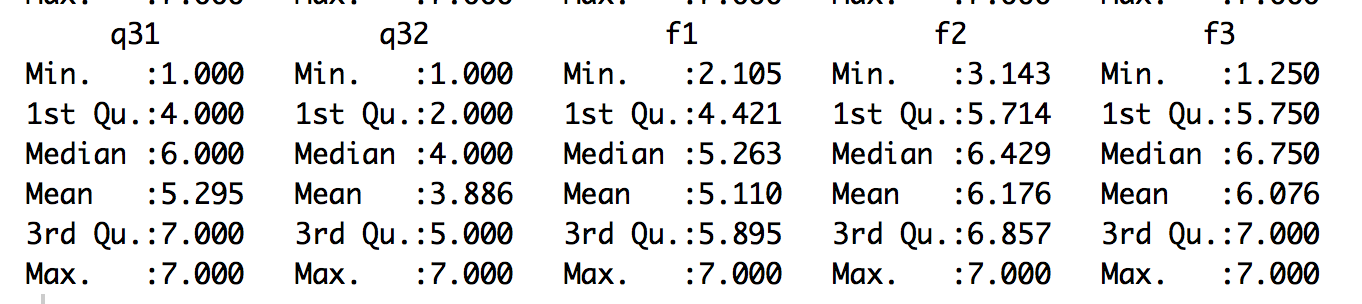
Factor 3:



* 1. They are all ok and over .80.

1. Theory: What do the factors mean? Look at question themes and give the factors a label.
   1. Factor 1: Intrinsic motivation/love of learning
      1. Success, pleasure of studies, intense feeling, learn lots, satisfaction, pleasure of reading/learning new things, etc.
   2. Factor 2: Career goals
      1. Job, skills for area, profession skills, degree, job, career, etc.
   3. Factor 3: Doubts about college
      1. Boring, wasting my time, don’t know what doing, should I keep going
2. Create new factor scores for each person:
   1. We’ve used apply to help us figure out percent missing, so now we can use it to create average or total scores for each person.
   2. *dataset*$f1 = apply(*dataset*[ , factor1], 1, mean) ##creates average scores
   3. *dataset*$f1 = apply(*dataset*[ , factor1], 1, sum) ##creates total scores
   4. Get the average and SD for the new factors:

summary(noout)



> sd(noout$f1)

[1] 1.081914

> sd(noout$f2)

[1] 0.8545663

> sd(noout$f3)

[1] 1.359161

**Example Write Up:**

**Results**

An exploratory factor analysis (EFA) was used to analyze the underlying factors in the self-determination motivation for college questionnaire using the *psych* package in *R*. Data were screened for multivariate assumptions (normality, linearity, homogeneity, and homoscedasticity), and all assumptions were met with slight problems of heteroscedasticity. Twelve multivariate outliers were detected using Mahalanobis distance (*X2*(32) = 62.49), and they were removed from further analyses. A small number of missing data values were replaced using the *mice* package. The following EFA analyses were conducted using guidelines outlined in Preacher and MacCallum (2003). Bartlett’s test indicated correlation adequacy, *X2*(496) = 4767.62, *p* < .001, and the KMO test indicated sampling adequacy, *MSA =* 0.92.

A parallel analysis and scree plot examination suggested three overall factors, and a 3-factor model was tested based on theory. Maximum likelihood estimation was used with direct oblimin rotation because of expected factor correlation. After testing all 32 questions, two items split across several factors (4, 15) using the criterion that loadings must be greater than .300. These items were eliminated from further analyses. Another 3-factor model was tested, and the factor loadings are presented in Table 1. This model achieved simple structure with each item loading on one and only one factor. This model had moderate fit: the RMSEA indicated moderate fit at .10, 90%CI[.08, .10], and SRMR with excellent fit (.05), while the CFI (.87) and TLI (.83) indicated room for improvement.

Factor 1 included 16 items that measured the intrinsic motivation of attending college with questions such as “For the pleasure that I experience when I feel completely absorbed by what certain authors have written” and “7. For the pleasure I experience while surpassing myself in my studies”. See Appendix A for the questionnaire. Factor 2 included eight items that assessed career goals for a student, including “Because I need a degree to get a good job” and “Because it was the only way to be considered for the career I want”. Finally, Factor 3 included four questions that appeared to assess a student’s doubt about motivation for college studies with questions like “I wonder what I am doing in college, I actually found it boring”. The reliability of all three factors was very high with .95, .86, and .91 for Factors 1, 2, and 3 respectively. The mean scores for each factor were: Factor 1 *M* = 5.11 (*SD* = 1.08), Factor 2 *M* = 6.18 (*SD* = 0.85), and Factor 3 *M* = 6.08 (*SD* = 1.36).

Table 1. *3-Factor Model Loadings.*

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Factor 1 | Factor 2 | Factor 3 |
| 3 | **0.509** | 0.199 | -0.021 |
| 7 | **0.820** | -0.046 | 0.018 |
| 8 | **0.797** | -0.075 | -0.020 |
| 10 | **0.528** | 0.099 | 0.068 |
| 11 | **0.529** | 0.226 | -0.239 |
| 12 | **0.532** | 0.277 | -0.261 |
| 14 | **0.681** | 0.218 | -0.028 |
| 16 | **0.828** | -0.178 | 0.139 |
| 18 | **0.631** | 0.171 | -0.069 |
| 20 | **0.709** | 0.163 | -0.078 |
| 21 | **0.848** | -0.164 | 0.132 |
| 22 | **0.736** | -0.031 | -0.073 |
| 24 | **0.439** | 0.240 | -0.235 |
| 29 | **0.619** | 0.156 | -0.056 |
| 31 | **0.469** | 0.171 | -0.105 |
| 32 | **0.791** | -0.170 | 0.132 |
| 2 | -0.076 | **0.658** | -0.101 |
| 4 | 0.286 | **0.456** | -0.182 |
| 5 | 0.164 | **0.525** | -0.181 |
| 13 | 0.005 | **0.804** | 0.120 |
| 19 | -0.004 | **0.755** | 0.103 |
| 26 | 0.065 | **0.714** | -0.184 |
| 28 | 0.004 | **0.563** | 0.053 |
| 30 | 0.055 | **0.805** | -0.011 |
| 6 | 0.018 | 0.012 | **0.801** |
| 9 | 0.014 | 0.049 | **0.946** |
| 17 | -0.030 | 0.101 | **0.886** |
| 27 | 0.121 | -0.007 | **0.754** |

*Note*. Factor loadings have been sorted and bolded for ease of reading.