HW 1 - AA

CJK

2023-05-21

## R Markdown

Homework Assignment 1 for Advanced Analytics:

It is important to note that the ‘first’ step below is simply the first one in an R script. Before beginning the assignment, I created a GitHub account, charlottesdataweb, and created a repository for this assignment called ‘some-pig’ - I downloaded the latest version of GitHub on my device and configured it on my device’s terminal. I created a project linked to the repository, and the ‘Git’ tab is now available in my Environment pane. After a (very) long time of troubleshooting the process and test-running some short scripts, GitHub and R are now synced up (restarting R helped with the first hurdle…the second hurdle was getting the .sav file to work. I had to ensure it was also a downloaded file in my ‘Git’ tab, I initially thought it had to pull from the same folder as my project…took awhile, but we got there!)

I am using GitKraken to stage changes to the markdown file and pull edits to my GitHub origin online. Now that R and GitHub have started a relationship and I understand the basics of how they work together, here’s the actual code breakdown:

The following are steps to perform an Exploratory Factor Analysis (EFA) on a dataset. The first steps involve gaining a broad understanding of the data, looking for anything concerning (like unusual values, missing data, etc), and examining the data to make sure it meets the key assumptions for an EFA. If the data meet the requirements, the next steps will be to gain some initial insights into what factor models may fit best with the data, to conduct the EFA and determine which factor model provides the best fit for the data, and to name the factors based on how they group the questions together. The last step will be a reliability analysis of each individual scale to determine if it meets the standard of reliability or if future steps should be taken to increase its reliability.

A. Step one is to set my seed. Setting the seed can offer some ‘control’ over the randomness of certain R functions. This would be especially useful if I was to perform a CFA right after my EFA. By using the same number, I can ensure I am working with the exact same split of training data (EFA) and test data (CFA), so there is no overlap

set.seed(69)  
knitr::opts\_chunk$set(cache.extra = knitr::rand\_seed)

B. Next step is to ensure all the needed packages are installed and called into the library. This should include the ‘haven’ package in order to properly import the .sav file (SPSS file) for this particular project. The ‘pacman’ package will install any packages on the list not already installed and call all packages into the library

tryCatch(require(pacman),finally=utils:::install.packages(pkgs='pacman',repos='http://cran.r-project.org'));

## Loading required package: pacman

##   
## The downloaded binary packages are in  
## /var/folders/\_6/7mn81ts936g1klwprl9hyq7m0000gn/T//RtmpaenyqF/downloaded\_packages

require(pacman)  
  
pacman::p\_load(Hmisc,  
 checkmate,  
 corrr,  
 conflicted,  
 readxl,  
 dplyr,  
 tidyr,  
 ggplot2,  
 knitr,  
 evaluate,  
 iopsych,  
 psych,  
 quantreg,  
 lavaan,  
 xtable,  
 reshape2,  
 GPArotation,  
 Amelia,  
 expss,  
 multilevel,  
 janitor,  
 mice,  
 lmtest,  
 haven,  
 tidylog  
)  
  
#' <!-- #Loading from GitHub -->  
#' <!-- #pacman::p\_load\_current\_gh("trinker/lexicon", "trinker/sentimentr") -->

Note: The last package should be ‘tidylog’ and it’s optional but useful to provide a log of information and feedback about different ‘diplyr’ and ‘tidyr’ functions that are used!

C. Now I’m all ready to load the dataset into R. I’m loading the data and calling the initial dataset ‘hwdata’

hwdata <- read\_sav("~/Documents/DirectoryR/UGA IOMP 2023 AA/some-pig/HW 1/SAQ.sav")

D. Now that the dataset is loaded, I want to get some broad overview information about it so I know what I’m working with. The str() function gives me some basic information about the structure of the data: it tells me what the questions are, the range of the scale for each item, etc.

str(hwdata)

## tibble [2,571 × 31] (S3: tbl\_df/tbl/data.frame)  
## $ Question\_01: dbl+lbl [1:2571] 2, 1, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 3, 1, 2,...  
## ..@ label : chr "Statistics makes me cry"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:6] 1 2 3 4 5 9  
## .. ..- attr(\*, "names")= chr [1:6] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_02: dbl+lbl [1:2571] 1, 1, 3, 1, 1, 1, 3, 2, 3, 4, 1, 1, 1, 2, 2, 1, 2, 2,...  
## ..@ label : chr "My friends will think I'm stupid for not being able to cope with SPSS"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_03: dbl+lbl [1:2571] 4, 4, 2, 1, 3, 3, 3, 3, 1, 4, 5, 3, 3, 1, 3, 2, 5, 3,...  
## ..@ label : chr "Standard deviations excite me"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_04: dbl+lbl [1:2571] 2, 3, 2, 4, 2, 2, 2, 2, 4, 3, 2, 3, 4, 2, 4, 2, 2, 3,...  
## ..@ label : chr "I dream that Pearson is attacking me with correlation coefficients"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:6] 1 2 3 4 5 9  
## .. ..- attr(\*, "names")= chr [1:6] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_05: dbl+lbl [1:2571] 2, 2, 4, 3, 2, 4, 2, 2, 5, 2, 2, 4, 3, 2, 2, 2, 1, 3,...  
## ..@ label : chr "I don't understand statistics"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_06: dbl+lbl [1:2571] 2, 2, 1, 3, 3, 4, 2, 2, 3, 1, 1, 3, 2, 2, 2, 2, 1, 4,...  
## ..@ label : chr "I have little experience of computers"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_07: dbl+lbl [1:2571] 3, 2, 2, 4, 3, 4, 2, 2, 5, 2, 2, 3, 3, 3, 3, 2, 1, 3,...  
## ..@ label : chr "All computers hate me"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_08: dbl+lbl [1:2571] 1, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 1, 3, 2, 2, 2, 1, 2,...  
## ..@ label : chr "I have never been good at mathematics"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_09: dbl+lbl [1:2571] 1, 5, 2, 2, 4, 4, 3, 4, 3, 3, 5, 3, 2, 2, 2, 2, 4, 5,...  
## ..@ label : chr "My friends are better at statistics than me"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_10: dbl+lbl [1:2571] 2, 2, 2, 4, 2, 3, 2, 2, 3, 2, 2, 2, 3, 3, 3, 3, 1, 2,...  
## ..@ label : chr "Computers are useful only for playing games"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_11: dbl+lbl [1:2571] 1, 2, 3, 2, 2, 2, 2, 2, 5, 2, 1, 2, 3, 2, 2, 2, 1, 3,...  
## ..@ label : chr "I did badly at mathematics at school"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_12: dbl+lbl [1:2571] 2, 3, 3, 2, 3, 4, 2, 3, 5, 3, 3, 3, 4, 4, 3, 3, 2, 3,...  
## ..@ label : chr "People try to tell you that SPSS makes statistics easier to understand but it doesn't"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_13: dbl+lbl [1:2571] 2, 1, 2, 2, 3, 3, 2, 2, 5, 2, 1, 2, 4, 2, 2, 2, 1, 3,...  
## ..@ label : chr "I worry that I will cause irreparable damage because of my incompetence with computers"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_14: dbl+lbl [1:2571] 2, 3, 4, 3, 2, 3, 2, 2, 5, 1, 2, 2, 4, 4, 3, 3, 1, 3,...  
## ..@ label : chr "Computers have minds of their own and deliberately go wrong whenever I use them"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_15: dbl+lbl [1:2571] 2, 4, 2, 3, 2, 5, 2, 3, 5, 2, 1, 3, 4, 4, 3, 2, 1, 4,...  
## ..@ label : chr "Computers are out to get me"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_16: dbl+lbl [1:2571] 3, 3, 3, 3, 2, 2, 2, 2, 5, 3, 2, 3, 4, 4, 4, 3, 2, 3,...  
## ..@ label : chr "I weep openly at the mention of central tendency"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:6] 1 2 3 4 5 9  
## .. ..- attr(\*, "names")= chr [1:6] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_17: dbl+lbl [1:2571] 1, 2, 2, 2, 2, 3, 2, 2, 5, 2, 2, 2, 3, 2, 2, 2, 2, 2,...  
## ..@ label : chr "I slip into a coma whenever I see an equation"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_18: dbl+lbl [1:2571] 2, 2, 3, 4, 3, 5, 2, 2, 5, 2, 2, 2, 3, 4, 3, 3, 1, 2,...  
## ..@ label : chr "SPSS always crashes when I try to use it"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_19: dbl+lbl [1:2571] 3, 3, 1, 2, 3, 1, 3, 4, 2, 3, 5, 3, 2, 1, 3, 2, 4, 2,...  
## ..@ label : chr "Everybody looks at me when I use SPSS"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_20: dbl+lbl [1:2571] 2, 4, 4, 4, 4, 5, 2, 3, 5, 3, 3, 4, 4, 5, 4, 3, 2, 3,...  
## ..@ label : chr "I can't sleep for thoughts of eigenvectors"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_21: dbl+lbl [1:2571] 2, 4, 3, 4, 2, 3, 2, 2, 5, 2, 2, 3, 4, 5, 4, 2, 1, 3,...  
## ..@ label : chr "I wake up under my duvet thinking that I am trapped under a normal distribution"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_22: dbl+lbl [1:2571] 2, 4, 2, 4, 4, 1, 4, 4, 3, 4, 5, 4, 3, 3, 4, 3, 4, 3,...  
## ..@ label : chr "My friends are better at SPSS than I am"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:6] 1 2 3 4 5 9  
## .. ..- attr(\*, "names")= chr [1:6] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ Question\_23: dbl+lbl [1:2571] 5, 2, 2, 3, 4, 4, 4, 4, 3, 4, 5, 4, 4, 1, 4, 4, 4, 4,...  
## ..@ label : chr "If I'm good at statistics my friends will think I'm a nerd"  
## ..@ format.spss: chr "F1.0"  
## ..@ labels : Named num [1:5] 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...  
## $ FAC1\_1 : num [1:2571] -1.1297 -0.0448 0.1562 0.7937 -0.9825 ...  
## ..- attr(\*, "label")= chr "A-R factor score 1 for analysis 1"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13  
## $ FAC2\_1 : num [1:2571] 0.0509 -0.4774 -0.7224 0.6118 0.6628 ...  
## ..- attr(\*, "label")= chr "A-R factor score 2 for analysis 1"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13  
## $ FAC3\_1 : num [1:2571] -1.586 -0.221 0.083 -0.793 -0.358 ...  
## ..- attr(\*, "label")= chr "A-R factor score 3 for analysis 1"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13  
## $ FAC4\_1 : num [1:2571] -0.552 0.641 -0.909 -0.318 0.548 ...  
## ..- attr(\*, "label")= chr "A-R factor score 4 for analysis 1"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13  
## $ FAC1\_2 : num [1:2571] -0.9452 -0.0171 0.2749 0.8158 -1.0391 ...  
## ..- attr(\*, "label")= chr "A-R factor score 1 for analysis 2"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13  
## $ FAC2\_2 : num [1:2571] -0.618 0.61 -0.934 -0.227 0.516 ...  
## ..- attr(\*, "label")= chr "A-R factor score 2 for analysis 2"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13  
## $ FAC3\_2 : num [1:2571] 0.0378 -0.5085 -0.6518 0.7289 0.5483 ...  
## ..- attr(\*, "label")= chr "A-R factor score 3 for analysis 2"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13  
## $ FAC4\_2 : num [1:2571] 1.6807 0.2401 -0.0804 0.6975 0.4316 ...  
## ..- attr(\*, "label")= chr "A-R factor score 4 for analysis 2"  
## ..- attr(\*, "format.spss")= chr "F11.5"  
## ..- attr(\*, "display\_width")= int 13

E. Now I know we’re dealing with 23 questions, all on a scale of 1-5, from ‘Strongly agree’ to ‘Strongly disagree’ - my first question/point of clarification after viewing this information is if I should conclude from this that Strongly agree = 1 (vs. Strongly disagree = 1). Based on the way it’s written, it looks like that’s correct, but I will double check. This will be useful information to help with further interpretations. Also, most of the items had a 1-5 numerical label, but a few said that the options were 1, 2, 3, 4, 5, and 9. However, I didn’t see any 9s when I looked at the data; however, I will examine more about this below. I also know that there are 8 factor scores pertaining to 2 different analyses. These may be useful later on, but for now, I plan to focus on the first 23 items. This means that I will likely need to build a variant of this dataset with only the first 23 questions.

To help me better understand what the scale means (and why some of them have a ‘9’ option), I took a look at the attributes of one of the questions (Q4) that had a ‘9’ option.

#making a scale label variable for question 4 to get info about the corresponding number/response  
scalelabels <- attributes(hwdata$Question\_04)$labels  
print(scalelabels)

## Strongly agree Agree Neither Disagree   
## 1 2 3 4   
## Strongly disagree Not answered   
## 5 9

This helped a ton! Now I know that 1 = Strongly agree (often times it’s the inverse, so that’s good to know). I also know that 9 corresponds with not answered/NA. At a cursory glance, I didn’t see any ‘9’ responses so it looks like maybe those were already taken care of.

I’m going to use the summary function to make sure all questions (1-23) have a max of 5 (and not 9).

summary(hwdata)

## Question\_01 Question\_02 Question\_03 Question\_04   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:2.000 1st Qu.:1.000 1st Qu.:2.000 1st Qu.:2.000   
## Median :2.000 Median :1.000 Median :3.000 Median :3.000   
## Mean :2.374 Mean :1.623 Mean :2.585 Mean :2.786   
## 3rd Qu.:3.000 3rd Qu.:2.000 3rd Qu.:3.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## Question\_05 Question\_06 Question\_07 Question\_08   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:2.000 1st Qu.:1.000 1st Qu.:2.000 1st Qu.:2.000   
## Median :3.000 Median :2.000 Median :3.000 Median :2.000   
## Mean :2.722 Mean :2.227 Mean :2.924 Mean :2.237   
## 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:4.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## Question\_09 Question\_10 Question\_11 Question\_12   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:3.000   
## Median :3.000 Median :2.000 Median :2.000 Median :3.000   
## Mean :2.846 Mean :2.281 Mean :2.255 Mean :3.159   
## 3rd Qu.:4.000 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:4.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## Question\_13 Question\_14 Question\_15 Question\_16   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000   
## Median :2.000 Median :3.000 Median :3.000 Median :3.000   
## Mean :2.449 Mean :2.876 Mean :2.766 Mean :2.879   
## 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## Question\_17 Question\_18 Question\_19 Question\_20   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:1.000 1st Qu.:3.000   
## Median :2.000 Median :2.000 Median :2.000 Median :4.000   
## Mean :2.467 Mean :2.569 Mean :2.292 Mean :3.624   
## 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:4.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## Question\_21 Question\_22 Question\_23 FAC1\_1   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :-3.10373   
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:3.000 1st Qu.:-0.69008   
## Median :3.000 Median :3.000 Median :4.000 Median :-0.07904   
## Mean :3.171 Mean :2.888 Mean :3.434 Mean : 0.00000   
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.: 0.65248   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. : 4.60083   
## FAC2\_1 FAC3\_1 FAC4\_1 FAC1\_2   
## Min. :-2.5530 Min. :-3.0874 Min. :-3.231645 Min. :-3.17112   
## 1st Qu.:-0.6801 1st Qu.:-0.5885 1st Qu.:-0.711371 1st Qu.:-0.69145   
## Median :-0.1712 Median :-0.1170 Median : 0.000857 Median :-0.06044   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.000000 Mean : 0.00000   
## 3rd Qu.: 0.5419 3rd Qu.: 0.4734 3rd Qu.: 0.678753 3rd Qu.: 0.64106   
## Max. : 3.7057 Max. : 4.3461 Max. : 3.433388 Max. : 4.48099   
## FAC2\_2 FAC3\_2 FAC4\_2   
## Min. :-3.310160 Min. :-2.3033 Min. :-4.1028   
## 1st Qu.:-0.703546 1st Qu.:-0.6682 1st Qu.:-0.5066   
## Median :-0.005424 Median :-0.1888 Median : 0.1579   
## Mean : 0.000000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.668124 3rd Qu.: 0.5281 3rd Qu.: 0.5814   
## Max. : 3.458260 Max. : 3.6224 Max. : 2.8292

The max is 5 across the board, so I have no 9s to worry about or need to edit.

F. Now that I have a good overview of the questions and how to interpret their responses, I can glimpse the data and get column names. I’m just getting to know the dataset and looking for anything that may require further clarification or scrutiny.

glimpse(hwdata)

## Rows: 2,571  
## Columns: 31  
## $ Question\_01 <dbl+lbl> 2, 1, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 3, 1, 2, …  
## $ Question\_02 <dbl+lbl> 1, 1, 3, 1, 1, 1, 3, 2, 3, 4, 1, 1, 1, 2, 2, 1, 2, 2, …  
## $ Question\_03 <dbl+lbl> 4, 4, 2, 1, 3, 3, 3, 3, 1, 4, 5, 3, 3, 1, 3, 2, 5, 3, …  
## $ Question\_04 <dbl+lbl> 2, 3, 2, 4, 2, 2, 2, 2, 4, 3, 2, 3, 4, 2, 4, 2, 2, 3, …  
## $ Question\_05 <dbl+lbl> 2, 2, 4, 3, 2, 4, 2, 2, 5, 2, 2, 4, 3, 2, 2, 2, 1, 3, …  
## $ Question\_06 <dbl+lbl> 2, 2, 1, 3, 3, 4, 2, 2, 3, 1, 1, 3, 2, 2, 2, 2, 1, 4, …  
## $ Question\_07 <dbl+lbl> 3, 2, 2, 4, 3, 4, 2, 2, 5, 2, 2, 3, 3, 3, 3, 2, 1, 3, …  
## $ Question\_08 <dbl+lbl> 1, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 1, 3, 2, 2, 2, 1, 2, …  
## $ Question\_09 <dbl+lbl> 1, 5, 2, 2, 4, 4, 3, 4, 3, 3, 5, 3, 2, 2, 2, 2, 4, 5, …  
## $ Question\_10 <dbl+lbl> 2, 2, 2, 4, 2, 3, 2, 2, 3, 2, 2, 2, 3, 3, 3, 3, 1, 2, …  
## $ Question\_11 <dbl+lbl> 1, 2, 3, 2, 2, 2, 2, 2, 5, 2, 1, 2, 3, 2, 2, 2, 1, 3, …  
## $ Question\_12 <dbl+lbl> 2, 3, 3, 2, 3, 4, 2, 3, 5, 3, 3, 3, 4, 4, 3, 3, 2, 3, …  
## $ Question\_13 <dbl+lbl> 2, 1, 2, 2, 3, 3, 2, 2, 5, 2, 1, 2, 4, 2, 2, 2, 1, 3, …  
## $ Question\_14 <dbl+lbl> 2, 3, 4, 3, 2, 3, 2, 2, 5, 1, 2, 2, 4, 4, 3, 3, 1, 3, …  
## $ Question\_15 <dbl+lbl> 2, 4, 2, 3, 2, 5, 2, 3, 5, 2, 1, 3, 4, 4, 3, 2, 1, 4, …  
## $ Question\_16 <dbl+lbl> 3, 3, 3, 3, 2, 2, 2, 2, 5, 3, 2, 3, 4, 4, 4, 3, 2, 3, …  
## $ Question\_17 <dbl+lbl> 1, 2, 2, 2, 2, 3, 2, 2, 5, 2, 2, 2, 3, 2, 2, 2, 2, 2, …  
## $ Question\_18 <dbl+lbl> 2, 2, 3, 4, 3, 5, 2, 2, 5, 2, 2, 2, 3, 4, 3, 3, 1, 2, …  
## $ Question\_19 <dbl+lbl> 3, 3, 1, 2, 3, 1, 3, 4, 2, 3, 5, 3, 2, 1, 3, 2, 4, 2, …  
## $ Question\_20 <dbl+lbl> 2, 4, 4, 4, 4, 5, 2, 3, 5, 3, 3, 4, 4, 5, 4, 3, 2, 3, …  
## $ Question\_21 <dbl+lbl> 2, 4, 3, 4, 2, 3, 2, 2, 5, 2, 2, 3, 4, 5, 4, 2, 1, 3, …  
## $ Question\_22 <dbl+lbl> 2, 4, 2, 4, 4, 1, 4, 4, 3, 4, 5, 4, 3, 3, 4, 3, 4, 3, …  
## $ Question\_23 <dbl+lbl> 5, 2, 2, 3, 4, 4, 4, 4, 3, 4, 5, 4, 4, 1, 4, 4, 4, 4, …  
## $ FAC1\_1 <dbl> -1.12973669, -0.04484331, 0.15619586, 0.79369971, -0.98250…  
## $ FAC2\_1 <dbl> 0.05090334, -0.47738650, -0.72239801, 0.61177764, 0.662841…  
## $ FAC3\_1 <dbl> -1.58646169, -0.22125910, 0.08298662, -0.79340882, -0.3581…  
## $ FAC4\_1 <dbl> -0.55242061, 0.64055340, -0.90900640, -0.31779452, 0.54788…  
## $ FAC1\_2 <dbl> -0.94520232, -0.01713883, 0.27492721, 0.81581663, -1.03907…  
## $ FAC2\_2 <dbl> -0.61761674, 0.61044497, -0.93408252, -0.22690610, 0.51633…  
## $ FAC3\_2 <dbl> 0.03775347, -0.50851122, -0.65182212, 0.72885731, 0.548270…  
## $ FAC4\_2 <dbl> 1.680667173, 0.240109522, -0.080425498, 0.697456742, 0.431…

dput(colnames(hwdata))

## c("Question\_01", "Question\_02", "Question\_03", "Question\_04",   
## "Question\_05", "Question\_06", "Question\_07", "Question\_08", "Question\_09",   
## "Question\_10", "Question\_11", "Question\_12", "Question\_13", "Question\_14",   
## "Question\_15", "Question\_16", "Question\_17", "Question\_18", "Question\_19",   
## "Question\_20", "Question\_21", "Question\_22", "Question\_23", "FAC1\_1",   
## "FAC2\_1", "FAC3\_1", "FAC4\_1", "FAC1\_2", "FAC2\_2", "FAC3\_2", "FAC4\_2"  
## )

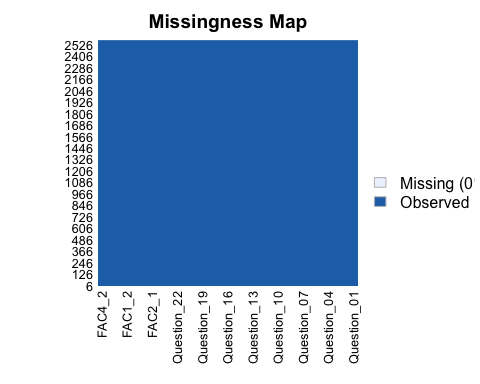
G. At first glance from the ‘glimpse’ information, what I can see looks like normal values on the Likert scale of 1-5. Nothing weird so far. Still not sure what the 8 analysis numbers are. The column names don’t provide too much information since they mostly just show the question #, but the questions are sequential with no number skipped from 1-2, and I’ve already examined the actual questions above using the str() function.

H. And last step in examining the broad overview of information is looking for missing data. I can do this using the ‘Amelia’ package (too soon…)

library(Amelia)  
  
missmap(hwdata)

## Warning: Unknown or uninitialised column: `arguments`.  
## Unknown or uninitialised column: `arguments`.

## Warning: Unknown or uninitialised column: `imputations`.

 I. This is great news (and rare). There’s no missing data! Now I can continue with the process of determining if the data meets the required assumptions for performing an Exploratory Factor Analysis (EFA).

1. Before performing an EFA, I will be evaluating the data to ensure it meets the necessary assumptions for an EFA. The data will be analyzed for the following:
2. Outliers
3. Linearity
4. Correlation
5. Sampling
6. First, I will search for outliers by measuring the Mahalanobis distance. Mahalanobis distance is how far away the data point is from the overall distribution of the data. Are there any cases of extreme or unusual responses that may affect the results of an EFA down the line?

# set.seed(69)  
#   
#   
# cutoff = qchisq(1-.001, ncol(hwdata))  
# mahal = mahalanobis(hwdata,  
# colMeans(hwdata),  
# cov(hwdata))  
# cutoff   
# ncol(hwdata)  
# summary(mahal < cutoff)

I got an error that said the system is computationally singular. Then I remembered that the dataset had the survey responses, as well as the factor analysis scores. So I will build a new dataset with only the survey responses and rerun the Mahalanobis code with that to see if that will work.

Also, I commented the chunk above because, when I tried to knit the markdown file, it wouldn’t allow it to knit with this type of error. Lots of lessons to be learned!

To create a new dataframe, I’m going to ‘borrow’ the code we learned from the dplyr package to deselect the last eight columns so I have a new dataset that is only survey responses. Note that, in the code below, I have a minus sign (-) before I concatenate c(). This creates a new data frame (called hwdataitems) that excludes the 8 analysis columns.

library(dplyr)  
hwdataitems <- hwdata %>%  
 dplyr::select(-c(FAC1\_1,FAC2\_1,FAC3\_1,FAC4\_1,FAC1\_2,FAC2\_2,FAC3\_2,FAC4\_2))

That worked! I’m now going rerun the Mahalnobis with hwdataitems instead.

set.seed(69)  
  
  
cutoff = qchisq(1-.001, ncol(hwdataitems))  
mahal = mahalanobis(hwdataitems,  
 colMeans(hwdataitems),  
 cov(hwdataitems))  
cutoff ##cutoff score

## [1] 49.72823

ncol(hwdata) ##df

## [1] 31

summary(mahal < cutoff)

## Mode FALSE TRUE   
## logical 97 2474

My output (97 FALSE) indicates 97 potential outliers, but I’ll take a deeper look at them. The code below creates a new dataframe with the Mahalanobis distance values as a new column. The ‘24’ in the code below denotes that it will be the 24th column in the new dataset. The column will be named ‘mahal’ - I had to specify the rename function from the dplyr package.

hwdataitems\_mahal <- hwdataitems %>%  
 bind\_cols(mahal) %>%  
 dplyr::rename(mahal = `...24`)

## New names:  
## • `` -> `...24`

Now I can look only at the 97 FALSE folks to examine their responses more closely. I will do this using the filter function. The filter function is present in several packages, so I will specify the dplyr package to perform the function (dplyr::filter) The arrange() function will sort them from highest to lowest

mahal\_out <- hwdataitems\_mahal %>%  
 dplyr::filter(mahal > cutoff) %>%  
 arrange(desc(mahal))   
View(mahal\_out)

I definitely see some extreme responses, as well as some unusual patterns of responses in the potential outliers. Some respondents selected almost exclusively 1s and 5s, and their selection patterns don’t make a lot of sense. For instance, some have the same or similar responses to Q1 (‘Statistics makes me cry’) and to Q3 (‘Standard deviations excite me’) even though they are much more likely to be more opposing views. Or some selected opposing responses to Q8 (‘I have never been good at mathematics’) and Q11 (‘I did badly at mathematics in school’) even though intuitively, those responses should align similarly with one another. This makes me question if the outliers were responding carefully and consistently, and my inclination is to assume that they weren’t.

Because of this, it is probably best practice to remove the outliers. Additionally, it is noteworthy that the initial sample size (2571 respondents) can easily lend itself to removing almost 100 respondents. I will use the filter function from dplyr once more, this time to create a dataframe that does not include the identified outliers.

#bye bye outliers!  
nomoreout <- hwdataitems %>%  
 dplyr::filter(mahal < cutoff)  
View(nomoreout)

My new data set, nomoreout, has 2474 observations instead of the 2571. Now that I have identified and handled the outliers, I can move on to the other assumption checks before pressing on with an EFA.

1. Next, I’m going to examine the additivity and linearity of the data. I will start with additivity. Overall, it is the concept that the effects of the individual predictors are independent of each other, that their effects are additive (simply the sum of their individual effects) with no interactions and no high collinearity. I can look at this using a correlation matrix that identifies if any items are too highly correlated with any other item(s). I will do this with the cor() and synum() function. I’m looking for any higher than usual correlations outside of the diagonal line of 1s (the key is provided in the output).

correl = cor(nomoreout, use = "pairwise.complete.obs")  
  
symnum(correl)

## Q\_01 Q\_02 Q\_03 Q\_04 Q\_05 Q\_06 Q\_07 Q\_08 Q\_09 Q\_10 Q\_11 Q\_12 Q\_13  
## Question\_01 1   
## Question\_02 1   
## Question\_03 . . 1   
## Question\_04 . . 1   
## Question\_05 . . . 1   
## Question\_06 1   
## Question\_07 . . . . . 1   
## Question\_08 . . . 1   
## Question\_09 . . 1   
## Question\_10 . 1   
## Question\_11 . . . . . . , 1   
## Question\_12 . . . . . . . 1   
## Question\_13 . . . . . . . . . . 1   
## Question\_14 . . . . . . . . .   
## Question\_15 . . . . . . . . .   
## Question\_16 . . . . . . . . .   
## Question\_17 . . . . . , , . .   
## Question\_18 . . . . . . . . . .   
## Question\_19 .   
## Question\_20 . .   
## Question\_21 . . . . . . . . .   
## Question\_22   
## Question\_23   
## Q\_14 Q\_15 Q\_16 Q\_17 Q\_18 Q\_19 Q\_20 Q\_21 Q\_22 Q\_23  
## Question\_01   
## Question\_02   
## Question\_03   
## Question\_04   
## Question\_05   
## Question\_06   
## Question\_07   
## Question\_08   
## Question\_09   
## Question\_10   
## Question\_11   
## Question\_12   
## Question\_13   
## Question\_14 1   
## Question\_15 . 1   
## Question\_16 . . 1   
## Question\_17 . . . 1   
## Question\_18 . . . . 1   
## Question\_19 1   
## Question\_20 1   
## Question\_21 . . . . . . 1   
## Question\_22 1   
## Question\_23 1   
## attr(,"legend")  
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '\*' 0.95 'B' 1

correl

## Question\_01 Question\_02 Question\_03 Question\_04 Question\_05  
## Question\_01 1.00000000 -0.09564516 -0.3394120 0.44356524 0.40683612  
## Question\_02 -0.09564516 1.00000000 0.3403275 -0.11535211 -0.12750749  
## Question\_03 -0.33941198 0.34032755 1.0000000 -0.38501916 -0.31188888  
## Question\_04 0.44356524 -0.11535211 -0.3850192 1.00000000 0.41311634  
## Question\_05 0.40683612 -0.12750749 -0.3118889 0.41311634 1.00000000  
## Question\_06 0.21642795 -0.09409007 -0.2364366 0.27450232 0.25262203  
## Question\_07 0.31830059 -0.15146503 -0.3888337 0.41073259 0.35683807  
## Question\_08 0.35970427 -0.05150263 -0.2718204 0.35532444 0.28485684  
## Question\_09 -0.09735681 0.34107119 0.3121586 -0.13853293 -0.10449784  
## Question\_10 0.20718614 -0.09690482 -0.2000051 0.21945052 0.24494916  
## Question\_11 0.37639745 -0.14986851 -0.3593173 0.37921625 0.30764567  
## Question\_12 0.36296740 -0.20647399 -0.4213259 0.44652908 0.35086220  
## Question\_13 0.36393293 -0.16223708 -0.3365851 0.34860223 0.30253499  
## Question\_14 0.35652805 -0.18215985 -0.3762200 0.35739840 0.32248586  
## Question\_15 0.25226994 -0.17019220 -0.3178184 0.34153652 0.26939559  
## Question\_16 0.49696876 -0.17745629 -0.4208369 0.42290818 0.40795250  
## Question\_17 0.38340729 -0.09727568 -0.3311392 0.37644360 0.32028257  
## Question\_18 0.37623424 -0.16729714 -0.3976746 0.39166256 0.34466207  
## Question\_19 -0.19273206 0.22571071 0.3581167 -0.20759579 -0.18525677  
## Question\_20 0.22712216 -0.21523224 -0.3266163 0.25872701 0.21311727  
## Question\_21 0.34426480 -0.21391337 -0.4336355 0.41958580 0.36194548  
## Question\_22 -0.10457188 0.25028516 0.2155666 -0.12649321 -0.14277734  
## Question\_23 -0.01544328 0.10567285 0.1583330 -0.04247334 -0.05693478  
## Question\_06 Question\_07 Question\_08 Question\_09 Question\_10  
## Question\_01 0.21642795 0.31830059 0.359704271 -0.097356807 0.20718614  
## Question\_02 -0.09409007 -0.15146503 -0.051502627 0.341071191 -0.09690482  
## Question\_03 -0.23643658 -0.38883370 -0.271820371 0.312158588 -0.20000510  
## Question\_04 0.27450232 0.41073259 0.355324438 -0.138532934 0.21945052  
## Question\_05 0.25262203 0.35683807 0.284856845 -0.104497841 0.24494916  
## Question\_06 1.00000000 0.53363050 0.235239319 -0.134138078 0.32562494  
## Question\_07 0.53363050 1.00000000 0.311551945 -0.137951567 0.28570275  
## Question\_08 0.23523932 0.31155195 1.000000000 0.004414116 0.17208292  
## Question\_09 -0.13413808 -0.13795157 0.004414116 1.000000000 -0.14437725  
## Question\_10 0.32562494 0.28570275 0.172082923 -0.144377253 1.00000000  
## Question\_11 0.34720392 0.36394420 0.645561669 -0.125660682 0.26278685  
## Question\_12 0.32463303 0.42835549 0.260041296 -0.188030097 0.25545078  
## Question\_13 0.48119806 0.46913918 0.335912534 -0.191141492 0.30106099  
## Question\_14 0.41067316 0.45132805 0.295700965 -0.140583541 0.26148441  
## Question\_15 0.37132397 0.39896365 0.312609295 -0.209655036 0.30124219  
## Question\_16 0.26157255 0.39026576 0.341856316 -0.196389345 0.28888431  
## Question\_17 0.29033521 0.40050472 0.607264448 -0.060008831 0.22004286  
## Question\_18 0.53564484 0.51924098 0.296357802 -0.166331510 0.32773688  
## Question\_19 -0.19434313 -0.28179738 -0.182826882 0.257208925 -0.13302464  
## Question\_20 0.09960499 0.22221657 0.200285197 -0.168290274 0.09134585  
## Question\_21 0.28028089 0.48323678 0.309998901 -0.151430462 0.20332338  
## Question\_22 -0.18495653 -0.18544749 -0.105454105 0.285347499 -0.12178841  
## Question\_23 -0.07307044 -0.06563211 -0.050427999 0.165983438 -0.06049506  
## Question\_11 Question\_12 Question\_13 Question\_14 Question\_15  
## Question\_01 0.3763974 0.36296740 0.36393293 0.3565280 0.25226994  
## Question\_02 -0.1498685 -0.20647399 -0.16223708 -0.1821598 -0.17019220  
## Question\_03 -0.3593173 -0.42132586 -0.33658510 -0.3762200 -0.31781841  
## Question\_04 0.3792163 0.44652908 0.34860223 0.3573984 0.34153652  
## Question\_05 0.3076457 0.35086220 0.30253499 0.3224859 0.26939559  
## Question\_06 0.3472039 0.32463303 0.48119806 0.4106732 0.37132397  
## Question\_07 0.3639442 0.42835549 0.46913918 0.4513281 0.39896365  
## Question\_08 0.6455617 0.26004130 0.33591253 0.2957010 0.31260929  
## Question\_09 -0.1256607 -0.18803010 -0.19114149 -0.1405835 -0.20965504  
## Question\_10 0.2627868 0.25545078 0.30106099 0.2614844 0.30124219  
## Question\_11 1.0000000 0.34140893 0.44362602 0.3463327 0.37438219  
## Question\_12 0.3414089 1.00000000 0.50259162 0.4405299 0.34756600  
## Question\_13 0.4436260 0.50259162 1.00000000 0.4465351 0.36827082  
## Question\_14 0.3463327 0.44052991 0.44653507 1.0000000 0.38799703  
## Question\_15 0.3743822 0.34756600 0.36827082 0.3879970 1.00000000  
## Question\_16 0.3771590 0.41419718 0.36951002 0.4202687 0.45003587  
## Question\_17 0.6093433 0.34043554 0.41248389 0.3494621 0.38806347  
## Question\_18 0.3978584 0.49956410 0.55019249 0.5026376 0.36303891  
## Question\_19 -0.2122948 -0.28553047 -0.25575048 -0.2694598 -0.22293558  
## Question\_20 0.2675185 0.30152135 0.21500906 0.2281466 0.22559760  
## Question\_21 0.3591919 0.43850465 0.38500498 0.3960666 0.31533342  
## Question\_22 -0.1802655 -0.19247298 -0.20970778 -0.1789014 -0.18108675  
## Question\_23 -0.0924981 -0.04901162 -0.06044059 -0.0600950 -0.07152116  
## Question\_16 Question\_17 Question\_18 Question\_19 Question\_20  
## Question\_01 0.49696876 0.38340729 0.37623424 -0.1927321 0.22712216  
## Question\_02 -0.17745629 -0.09727568 -0.16729714 0.2257107 -0.21523224  
## Question\_03 -0.42083690 -0.33113924 -0.39767457 0.3581167 -0.32661632  
## Question\_04 0.42290818 0.37644360 0.39166256 -0.2075958 0.25872701  
## Question\_05 0.40795250 0.32028257 0.34466207 -0.1852568 0.21311727  
## Question\_06 0.26157255 0.29033521 0.53564484 -0.1943431 0.09960499  
## Question\_07 0.39026576 0.40050472 0.51924098 -0.2817974 0.22221657  
## Question\_08 0.34185632 0.60726445 0.29635780 -0.1828269 0.20028520  
## Question\_09 -0.19638935 -0.06000883 -0.16633151 0.2572089 -0.16829027  
## Question\_10 0.28888431 0.22004286 0.32773688 -0.1330246 0.09134585  
## Question\_11 0.37715901 0.60934335 0.39785838 -0.2122948 0.26751848  
## Question\_12 0.41419718 0.34043554 0.49956410 -0.2855305 0.30152135  
## Question\_13 0.36951002 0.41248389 0.55019249 -0.2557505 0.21500906  
## Question\_14 0.42026874 0.34946207 0.50263764 -0.2694598 0.22814660  
## Question\_15 0.45003587 0.38806347 0.36303891 -0.2229356 0.22559760  
## Question\_16 1.00000000 0.41795652 0.43213319 -0.2750407 0.27996205  
## Question\_17 0.41795652 1.00000000 0.38775677 -0.1875743 0.22485820  
## Question\_18 0.43213319 0.38775677 1.00000000 -0.2819246 0.24510669  
## Question\_19 -0.27504071 -0.18757433 -0.28192464 1.0000000 -0.25279285  
## Question\_20 0.27996205 0.22485820 0.24510669 -0.2527928 1.00000000  
## Question\_21 0.42221207 0.36837433 0.43288139 -0.2814537 0.49045434  
## Question\_22 -0.17204435 -0.13450689 -0.19099701 0.2402900 -0.10112202  
## Question\_23 -0.09718849 -0.10123323 -0.08827266 0.1367694 -0.04440591  
## Question\_21 Question\_22 Question\_23  
## Question\_01 0.34426480 -0.1045719 -0.01544328  
## Question\_02 -0.21391337 0.2502852 0.10567285  
## Question\_03 -0.43363547 0.2155666 0.15833302  
## Question\_04 0.41958580 -0.1264932 -0.04247334  
## Question\_05 0.36194548 -0.1427773 -0.05693478  
## Question\_06 0.28028089 -0.1849565 -0.07307044  
## Question\_07 0.48323678 -0.1854475 -0.06563211  
## Question\_08 0.30999890 -0.1054541 -0.05042800  
## Question\_09 -0.15143046 0.2853475 0.16598344  
## Question\_10 0.20332338 -0.1217884 -0.06049506  
## Question\_11 0.35919193 -0.1802655 -0.09249810  
## Question\_12 0.43850465 -0.1924730 -0.04901162  
## Question\_13 0.38500498 -0.2097078 -0.06044059  
## Question\_14 0.39606664 -0.1789014 -0.06009500  
## Question\_15 0.31533342 -0.1810868 -0.07152116  
## Question\_16 0.42221207 -0.1720443 -0.09718849  
## Question\_17 0.36837433 -0.1345069 -0.10123323  
## Question\_18 0.43288139 -0.1909970 -0.08827266  
## Question\_19 -0.28145374 0.2402900 0.13676942  
## Question\_20 0.49045434 -0.1011220 -0.04440591  
## Question\_21 1.00000000 -0.1386220 -0.06999931  
## Question\_22 -0.13862196 1.0000000 0.23243564  
## Question\_23 -0.06999931 0.2324356 1.00000000

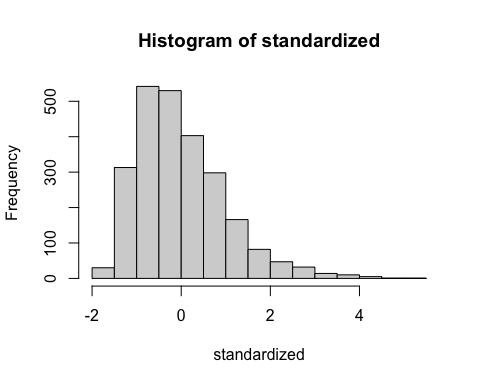
Good news, I don’t see anything that indicates too high correlations. There are a few .3 and .6 correlations, but nothing higher than that, so nothing to be concerned about here. We can assume an additive relationship.

Checking for linearity has a few more steps. The first is to set up a fake regression analysis and standardize it to examine a histogram for a normality analysis.

random = rchisq(nrow(nomoreout), 7)  
fake = lm(random~., data = nomoreout)  
standardized = rstudent(fake)  
fitted = scale(fake$fitted.values)

Now my ‘fake’ regression is all set up so I can view a histogram of the standardized results. I am looking for a normal distribution.

hist(standardized)

 We’re looking approximately normally distributed. There is a bit of a positive skew, but it’s slight, so I will move forward as is.

Next, I’ll test for heteroscedasticity with the Breusch-Pagan Test. Heteroscedasticity is unequal variance of the residuals. So if residuals were more dispersed at lower X-values but less dispersed at higher X-values, that would constistute heteroscedasticity. This is a violation of the assumption of equal variance, so we are hoping for homoscedasticity, or similar variance of residuals, within the model.

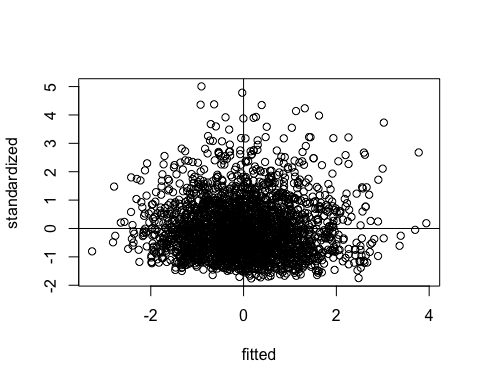
bptest(fake)

##   
## studentized Breusch-Pagan test  
##   
## data: fake  
## BP = 22.52, df = 23, p-value = 0.4891

Our test statistic is BP = 22.82, and our p-value = 0.472. Because p > .05, we fail to reject the null hypothesis. There is not sufficient evidence of the presence of heteroscedasticity in the model.

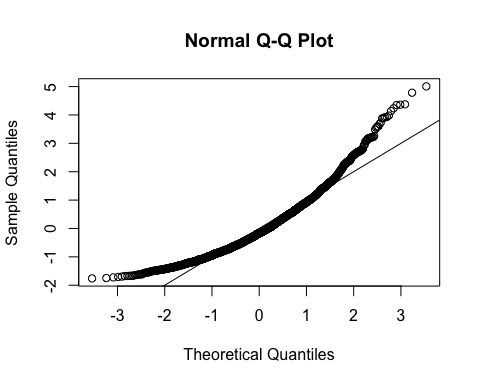
To further ensure there is no problem with heteroscedasticity, I will also plot the residuals. Any pattern or shape, such as a U-shape, funnel shape, or fan shape in the plots, would indicate unequal variance. So I am hoping for random points, no pattern in the residual plot.

plot(fitted,standardized)  
abline(0,0)  
abline(v = 0)

 We are looking pretty random, but not perfect. There seems to be skewness going on with less disperssion below the 0 standardized line and greater dispersion above 0. However, there is not a fan shape or anything funky going on left to right, so we will move forward with what I’ll call ‘pretty good homogeneity’ for now, especially since the results of the Breusch-Pagan test did not identify any heteroscedasticity. If a defined pattern emerged from the residual plots, I could transform my response variable, but for now I will press forward with the data as is.

Now I can look at linearity with a Q-Q plot. A Q-Q plot is a plot that looks at two different samples to see if they are likely from the same population. Here we have a theoretical sample and actual sample. We are hoping for something highly resembling a straight line along the ab line to provide sufficient evidence of linearity.

qqnorm(standardized)  
abline(0,1)



We’ve got a little bit of a curve at the high and low ends, but the line is roughly straight so I will move forward.

Overall, there is presence of linearity here. Is it perfect? No. But it is likely sufficient enough to continue with the other assumptions for an EFA.

1. Now I can look at correlation with Bartlett’s test. I am hoping that the items in the data are correlated sufficiently with each other for factor analysis results to make sense. If they are not correlated enough, the below results will be non-significant (which would indicate a problem…our items would not be related enough for EFA results to make sense). It’s worth noting that we have a very large sample size (2474 respondents) so very likely the results will be significant even if the inter-item correlations are smaller.

cortest.bartlett(correl, n = nrow(nomoreout))

## $chisq  
## [1] 19534.27  
##   
## $p.value  
## [1] 0  
##   
## $df  
## [1] 253

There is statistical significance, p > .05, so correlations are sufficient for EFA. I also know from the test for additivity that the items are not too correlated with each other to indicate problems with collinearity or redundancy. There is a relationship among the items, but they aren’t too related either. I am ready for the final step to ensure the data is well-fitted for an EFA.

1. Next, I will run the Kaiser–Meyer–Olkin (KMO) Measure of Sampling Adequacy Test. This helps determine if there is enough of a relationship among the variables in the data to run (and be able to trust the results of) a factor analysis. It does this by looking at the strength of the partial correlations of each variable. Are these variables (and the relationships among the variables) explained by underlying factors? Do they share enough common variance for EFA to make sense? Let’s see.

I am hoping for an overall MSA of .90 or higher.

KMO(correl)

## Kaiser-Meyer-Olkin factor adequacy  
## Call: KMO(r = correl)  
## Overall MSA = 0.93  
## MSA for each item =   
## Question\_01 Question\_02 Question\_03 Question\_04 Question\_05 Question\_06   
## 0.94 0.88 0.95 0.96 0.96 0.90   
## Question\_07 Question\_08 Question\_09 Question\_10 Question\_11 Question\_12   
## 0.94 0.88 0.85 0.96 0.91 0.96   
## Question\_13 Question\_14 Question\_15 Question\_16 Question\_17 Question\_18   
## 0.95 0.97 0.95 0.94 0.93 0.95   
## Question\_19 Question\_20 Question\_21 Question\_22 Question\_23   
## 0.95 0.89 0.93 0.89 0.79

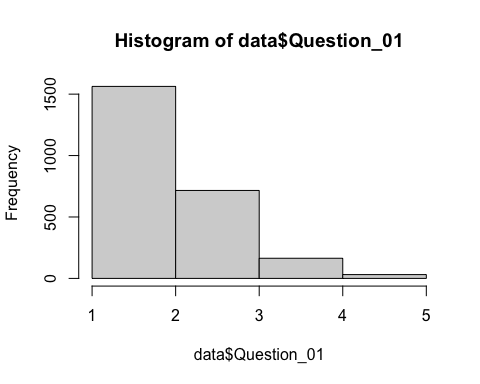
The overall MSA = .93. Great! There is solid sampling adequacy to perform an EFA.

Now that all the assumption checks are covered, I’m going to rename the data set, nomoreout, to ‘data’ for ease of use and understanding.

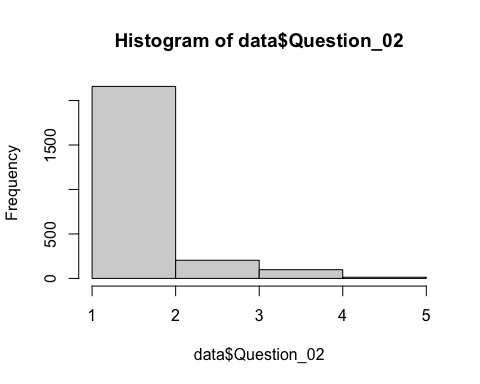
data <- nomoreout

1. Now I can take a look at the histograms of each item. I should have 23 lines of code, one for each question, and I will set breaks as 5 since they are all on a 5-point Likert scale.

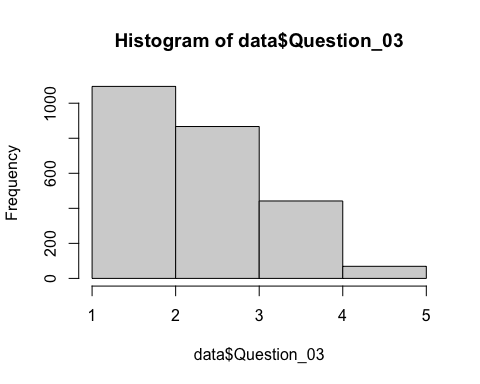
hist(data$Question\_01, breaks = 5)



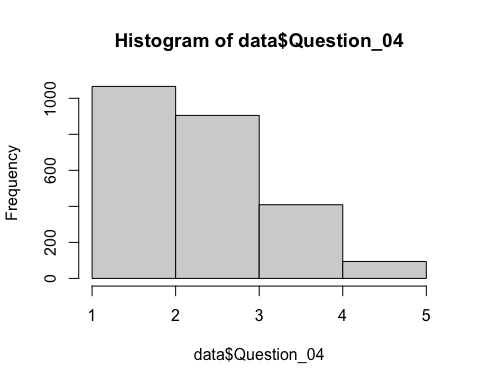
hist(data$Question\_02, breaks = 5)



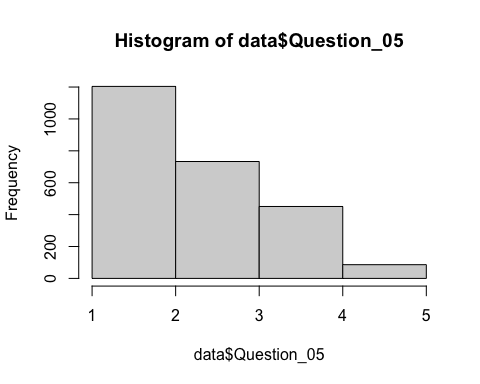
hist(data$Question\_03, breaks = 5)



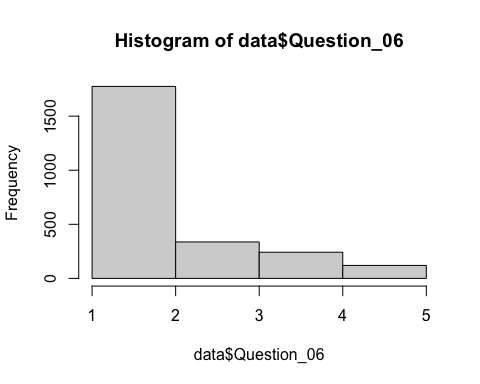
hist(data$Question\_04, breaks = 5)



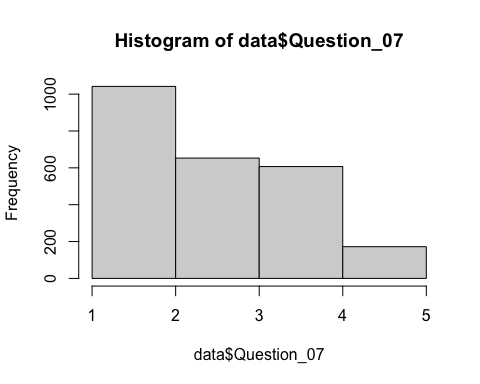
hist(data$Question\_05, breaks = 5)



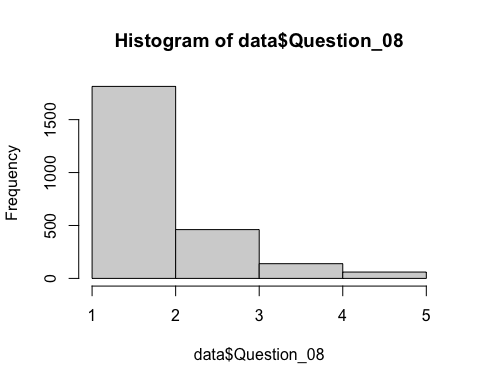
hist(data$Question\_06, breaks = 5)



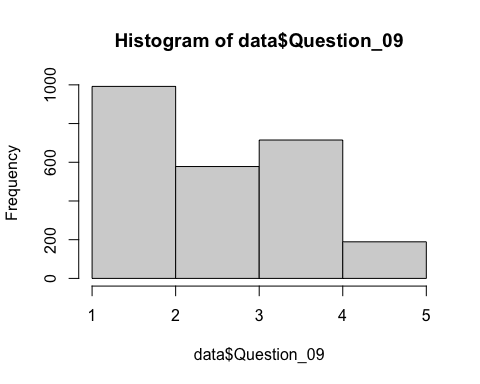
hist(data$Question\_07, breaks = 5)



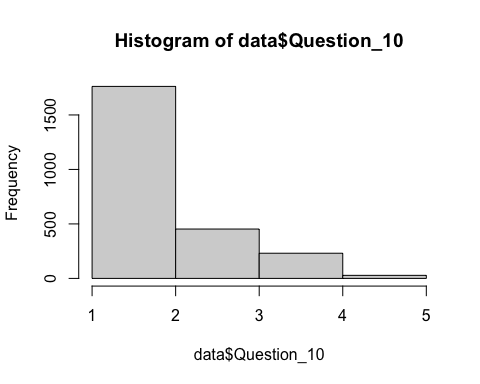
hist(data$Question\_08, breaks = 5)



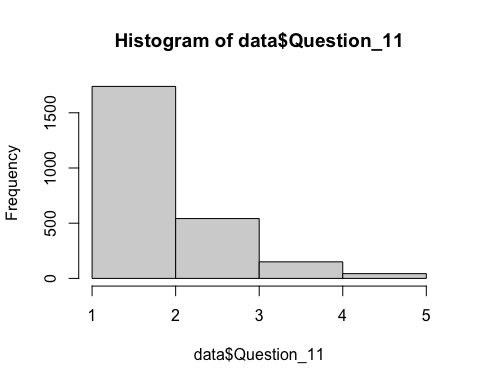
hist(data$Question\_09, breaks = 5)



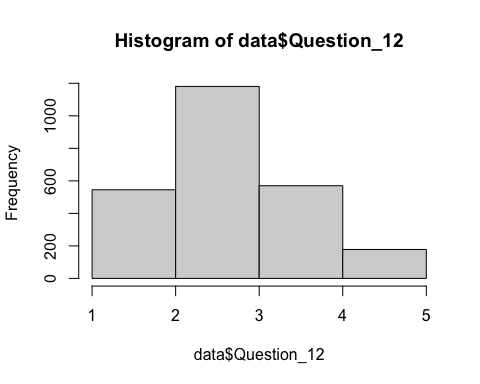
hist(data$Question\_10, breaks = 5)



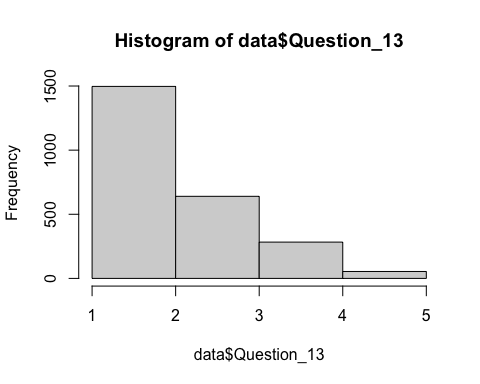
hist(data$Question\_11, breaks = 5)



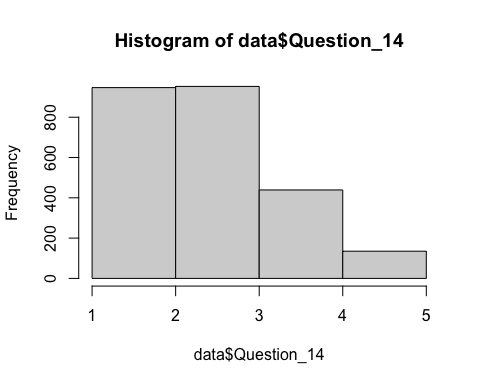
hist(data$Question\_12, breaks = 5)



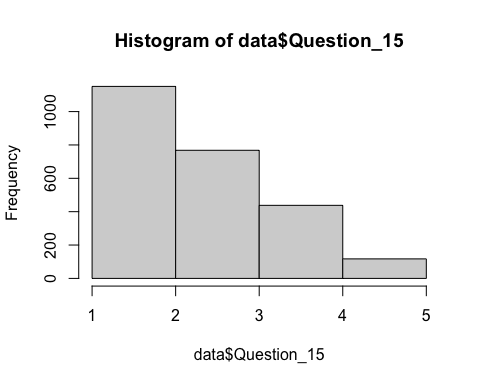
hist(data$Question\_13, breaks = 5)



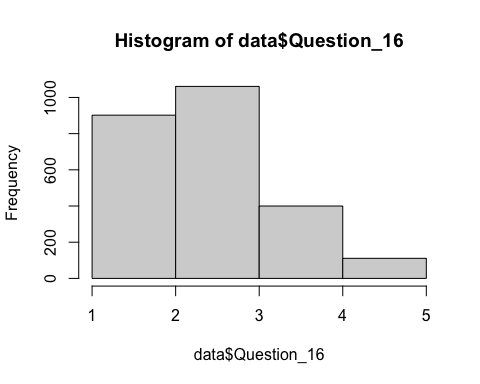
hist(data$Question\_14, breaks = 5)



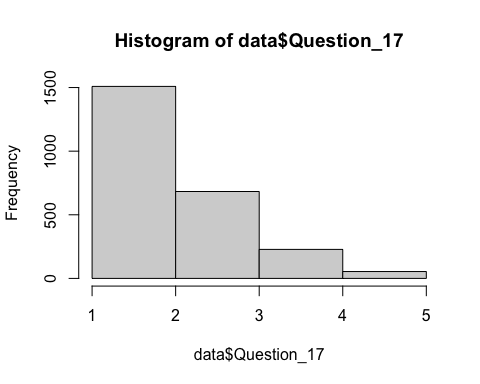
hist(data$Question\_15, breaks = 5)



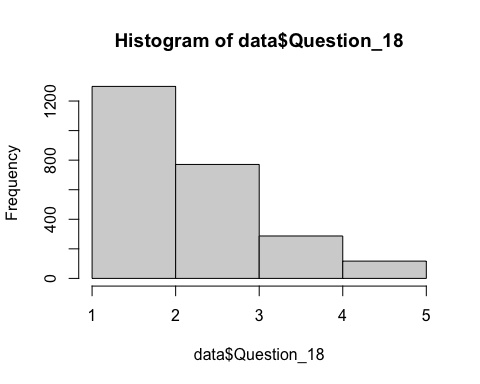
hist(data$Question\_16, breaks = 5)



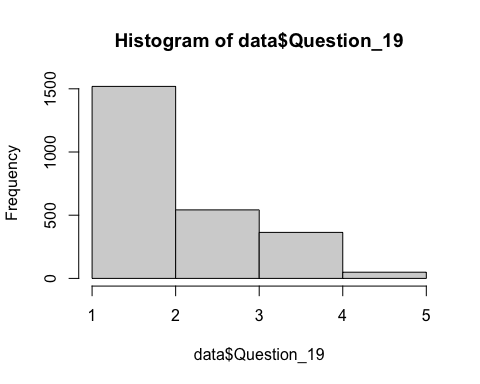
hist(data$Question\_17, breaks = 5)



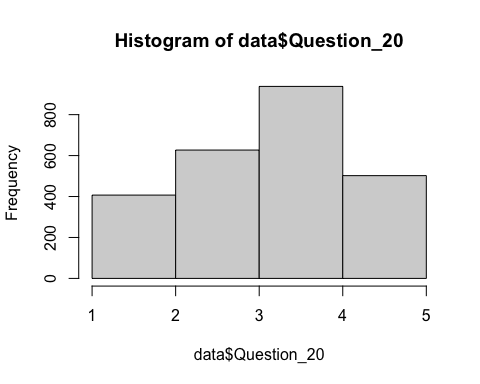
hist(data$Question\_18, breaks = 5)



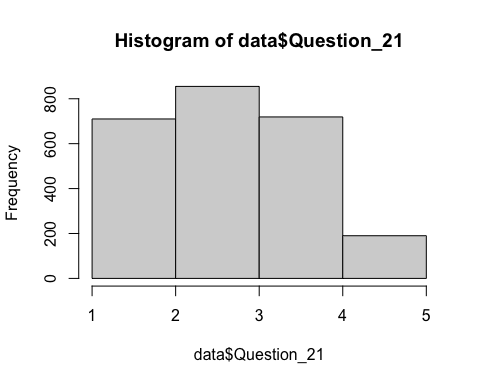
hist(data$Question\_19, breaks = 5)



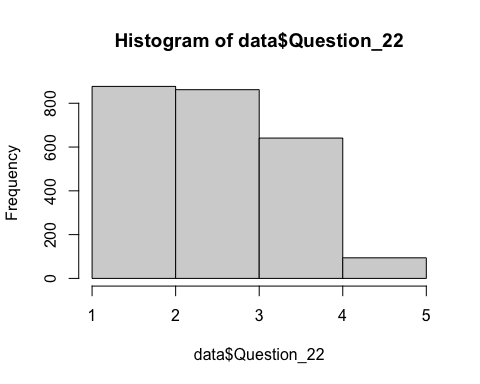
hist(data$Question\_20, breaks = 5)



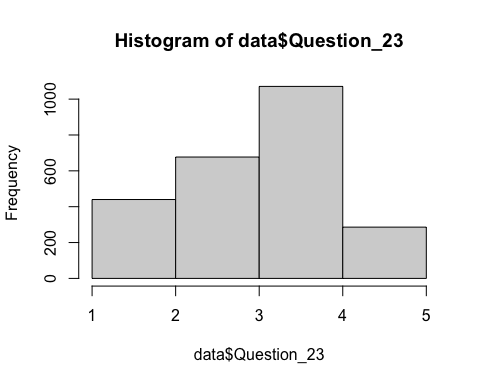
hist(data$Question\_21, breaks = 5)



hist(data$Question\_22, breaks = 5)



hist(data$Question\_23, breaks = 5)



Now I will try the code that will show all the histograms in one plot to see if it appears in the markdown file.

# par(mfrow =c(5,5),mar = c(3,4,3,4))  
#   
# hist(data$Question\_01, breaks = 5)  
# hist(data$Question\_02, breaks = 5)  
# hist(data$Question\_03, breaks = 5)  
# hist(data$Question\_04, breaks = 5)  
# hist(data$Question\_05, breaks = 5)  
# hist(data$Question\_06, breaks = 5)  
# hist(data$Question\_07, breaks = 5)  
# hist(data$Question\_08, breaks = 5)  
# hist(data$Question\_09, breaks = 5)  
# hist(data$Question\_10, breaks = 5)  
# hist(data$Question\_11, breaks = 5)  
# hist(data$Question\_12, breaks = 5)  
# hist(data$Question\_13, breaks = 5)  
# hist(data$Question\_14, breaks = 5)  
# hist(data$Question\_15, breaks = 5)  
# hist(data$Question\_16, breaks = 5)  
# hist(data$Question\_17, breaks = 5)  
# hist(data$Question\_18, breaks = 5)  
# hist(data$Question\_19, breaks = 5)  
# hist(data$Question\_20, breaks = 5)  
# hist(data$Question\_21, breaks = 5)  
# hist(data$Question\_22, breaks = 5)  
# hist(data$Question\_23, breaks = 5)

I had to adjust the margins with the mar = # syntax to get it to work. However, it only worked if the margins were set pretty small. Because of this, the question# is cut off from each, and it’s a little difficult to interpret the histograms. Because of this, I also put this code in a blank script and got the plot to show up in my plots pane, and I saved that plot. I will include that too since it’s bigger/easier to understand. Additionally, when I tried to knit the markdown file, the error or ‘figure margins too large’ would not let me knit. So, similar to the mahalanobis code getting an error, I have commented the code out jsut to keep track of what doesn’t work but also allow the file to knit.

Insights from the histograms are:

It is important to remember that 1 corresponds to strongly agree (1 is high, 5 is low). This will be noteworthy as I look more at the item and scale analyses.

It is noteworthy that there is a positive skew on quite a few items, with a majority of responses clustered around 1 and 2, and the response numbers seem to peter off as they get higher. This was a common trend I saw in the initial visualization, especially in the first 19 questions.

There does seem to be some relationships among the variables and nothing looks too unusual or out of place. Some items seem to be behaving in a similar fashion, which is good, but there also may be some items that are behaving too similarly - are they redundant? No major red flags yet on this end though. Another potential issue to stay mindful of is thatI could have some cross-loadings happen with such a high number of items that seem to be behaving in similar fashions, but I will know more when I cross that bridge. So far, so good.

Ok, I’m all ready to start the EFA process now. First are some preliminary steps to prepare my data and split it into two subsets.

I will set my seed first before dividing my data into my training (EFA) and testing (CFA) sets. I want to make sure the ‘randomness’ is the same ‘randomness’ for both testing and training analyses.

set.seed(69)

Next, I will create an ID variable and move it to the front (so it’s the first column). I can do this with a little help from the dplyr package. First step is to create the ID variable, then use the select function to move it before ‘everything’ so it’s the first column. Then I can use colnames() to make sure it’s first.

data <- data %>%   
 dplyr::mutate(ID = row\_number())  
  
data <- data %>%  
 dplyr::select(ID, everything())  
  
colnames(data)

## [1] "ID" "Question\_01" "Question\_02" "Question\_03" "Question\_04"  
## [6] "Question\_05" "Question\_06" "Question\_07" "Question\_08" "Question\_09"  
## [11] "Question\_10" "Question\_11" "Question\_12" "Question\_13" "Question\_14"  
## [16] "Question\_15" "Question\_16" "Question\_17" "Question\_18" "Question\_19"  
## [21] "Question\_20" "Question\_21" "Question\_22" "Question\_23"

Now, I can split my data.

1. I will split the data as 70/30. It is usually recommended to have a larger sample for the EFA (training), so I have chosen this uneven split that favors a bigger sample for the ‘test’ set. Also, since we practiced the 50/50 split in class, I wanted to try a different one that would more likely be applicable to best practices in the ‘real world’ - the split is also sometimes done as an 80/20 split. I opted for the slightly more conservative approach just to ensure enough observations remained for the testing set.

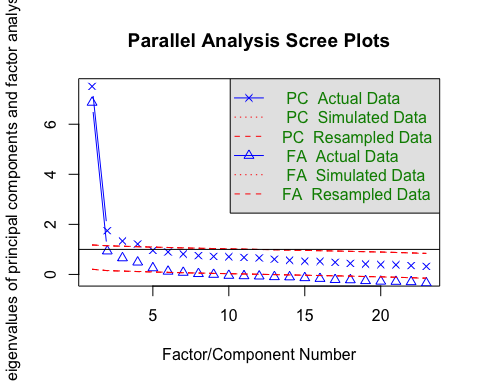
#70 percent of the sample is now labeled with the 'training' value  
training <- sample(data$ID, length(data$ID)\*0.7)  
  
#datatrain is now my new set that has the 70% above who are labeled as 'training' - and those not labeled as training are in the datatest set (with the %in% meaning in/contains and the ! meaning not in/does not contain)  
datatrain <- subset(data, ID %in% training)  
datatest <- subset(data, !(ID %in% training))

I can now check on my two sets of data and know that the split worked correctly right since datatrain has 1731 observations and datatest has 743 observations.

View(datatrain)  
View(datatest)

Before I run my EFA, I will run a Parallel Analysis just to get an idea of roughly how many factors I might be looking at. I am not familiar with this set of questions (and they seem very hilarious, by the way), so I’m not really going in with any previous knowledge to pull from. Hopefully the Scree Plot will give me a good starting point.

fa.parallel(data[c(2:24)])



## Parallel analysis suggests that the number of factors = 6 and the number of components = 4

Parallel analysis suggests that the number of factors = 6 and the number of components = 4 6 feels quite high for 23 items; it may be more like 4 or 5 I will start with 4, and work from there.

1. When selecting the method for my EFA, I opted for Maximum Likelihood (ML). ML is good for large datasets, it is well-suited when the assumptions for EFA have been met (as checked above), and it can handle unequal loadings within factors. I opted to not use Least Squares because it’s more commonly used when there is no factor correlation and no large cross-loadings (both of which which may occur based on my views of the histograms). I also opted not to use Principal Axis (PA) because it can often underestimate factor loadings and may not provide an accurate view of the underlying factors. ML does operate best when assumptions of normality are met. My hope is that the data is not so skewed that ML will inaccurate. If I run an initial EFA that seems to create relationships with variables that I don’t think are logical, I may revisit this decision and see what PA will provide for me. However, ML is the most commonly used method, especially in large sets of data.

Here’s information on the naming scheme for the data as I move forward:

* fa = Factor Analysis
* ml = Maximum Likelihood (the method of factor analysis I am using; if I change to principal axis, this will b pa instead)
* 4 = the number of factors I think are in the data (we will change this as we adjust the number of factors)
* trn = the training data (as opposed to the test data)

fa\_ml\_4\_trn <- fa(data[c(2:24)], nfactors = 4, fm="ml")  
print(fa\_ml\_4\_trn)

## Factor Analysis using method = ml  
## Call: fa(r = data[c(2:24)], nfactors = 4, fm = "ml")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## ML1 ML3 ML2 ML4 h2 u2 com  
## Question\_01 0.57 -0.02 0.16 0.09 0.405 0.59 1.2  
## Question\_02 -0.05 0.06 0.01 0.56 0.311 0.69 1.0  
## Question\_03 -0.40 0.02 -0.08 0.39 0.477 0.52 2.1  
## Question\_04 0.57 0.05 0.11 0.04 0.426 0.57 1.1  
## Question\_05 0.51 0.07 0.06 0.04 0.323 0.68 1.1  
## Question\_06 -0.10 0.83 0.04 0.00 0.637 0.36 1.0  
## Question\_07 0.30 0.49 0.03 -0.02 0.514 0.49 1.7  
## Question\_08 0.00 -0.06 0.85 0.07 0.671 0.33 1.0  
## Question\_09 0.03 -0.02 0.05 0.62 0.365 0.64 1.0  
## Question\_10 0.09 0.33 0.05 -0.05 0.193 0.81 1.3  
## Question\_11 -0.04 0.08 0.77 -0.10 0.663 0.34 1.1  
## Question\_12 0.51 0.22 -0.04 -0.10 0.459 0.54 1.4  
## Question\_13 0.18 0.46 0.15 -0.07 0.488 0.51 1.6  
## Question\_14 0.36 0.35 0.03 -0.05 0.424 0.58 2.1  
## Question\_15 0.17 0.25 0.19 -0.14 0.330 0.67 3.4  
## Question\_16 0.55 0.03 0.12 -0.08 0.464 0.54 1.1  
## Question\_17 0.11 0.05 0.67 0.02 0.582 0.42 1.1  
## Question\_18 0.32 0.53 0.00 -0.04 0.573 0.43 1.7  
## Question\_19 -0.17 -0.05 -0.03 0.36 0.253 0.75 1.5  
## Question\_20 0.44 -0.16 0.05 -0.21 0.267 0.73 1.8  
## Question\_21 0.58 0.05 0.03 -0.10 0.458 0.54 1.1  
## Question\_22 0.13 -0.12 -0.09 0.45 0.232 0.77 1.4  
## Question\_23 0.14 -0.01 -0.10 0.32 0.095 0.90 1.6  
##   
## ML1 ML3 ML2 ML4  
## SS loadings 3.35 2.31 2.30 1.64  
## Proportion Var 0.15 0.10 0.10 0.07  
## Cumulative Var 0.15 0.25 0.35 0.42  
## Proportion Explained 0.35 0.24 0.24 0.17  
## Cumulative Proportion 0.35 0.59 0.83 1.00  
##   
## With factor correlations of   
## ML1 ML3 ML2 ML4  
## ML1 1.00 0.51 0.57 -0.41  
## ML3 0.51 1.00 0.44 -0.32  
## ML2 0.57 0.44 1.00 -0.21  
## ML4 -0.41 -0.32 -0.21 1.00  
##   
## Mean item complexity = 1.4  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## df null model = 253 with the objective function = 7.93 with Chi Square = 19534.27  
## df of the model are 167 and the objective function was 0.43   
##   
## The root mean square of the residuals (RMSR) is 0.03   
## The df corrected root mean square of the residuals is 0.03   
##   
## The harmonic n.obs is 2474 with the empirical chi square 794.58 with prob < 2.2e-82   
## The total n.obs was 2474 with Likelihood Chi Square = 1053.65 with prob < 1.5e-128   
##   
## Tucker Lewis Index of factoring reliability = 0.93  
## RMSEA index = 0.046 and the 90 % confidence intervals are 0.044 0.049  
## BIC = -251.22  
## Fit based upon off diagonal values = 0.99  
## Measures of factor score adequacy   
## ML1 ML3 ML2 ML4  
## Correlation of (regression) scores with factors 0.92 0.90 0.92 0.84  
## Multiple R square of scores with factors 0.84 0.82 0.85 0.70  
## Minimum correlation of possible factor scores 0.69 0.63 0.71 0.40

Now I can look at only the loadings .30 and higher

print(fa\_ml\_4\_trn$loadings, cutoff = .3)

##   
## Loadings:  
## ML1 ML3 ML2 ML4   
## Question\_01 0.572   
## Question\_02 0.556  
## Question\_03 -0.401 0.386  
## Question\_04 0.568   
## Question\_05 0.509   
## Question\_06 0.829   
## Question\_07 0.303 0.487   
## Question\_08 0.854   
## Question\_09 0.618  
## Question\_10 0.329   
## Question\_11 0.774   
## Question\_12 0.514   
## Question\_13 0.461   
## Question\_14 0.358 0.346   
## Question\_15   
## Question\_16 0.554   
## Question\_17 0.671   
## Question\_18 0.317 0.527   
## Question\_19 0.365  
## Question\_20 0.436   
## Question\_21 0.581   
## Question\_22 0.455  
## Question\_23 0.320  
##   
## ML1 ML3 ML2 ML4  
## SS loadings 2.654 1.824 1.933 1.411  
## Proportion Var 0.115 0.079 0.084 0.061  
## Cumulative Var 0.115 0.195 0.279 0.340

There is a lot of cross-loading and A LOT of variables on the first factor, but right now, the structure is not easy to interpret. So I will rotate the factors first, then follow-up with my thoughts and interpretations.

1. I opted for an oblique rotation because I saw some correlation between items and similar patterns in the histograms, so I am operating under the idea that my factors will likely be more correlated (if there was evidence they would be less correlated, I could opt for an orthogonal rotation). Per the recommendation of James Gaskin’s EFA video, I chose a promax rotation to maximize interpretability.

fa\_ml\_4\_trn <- fa(data[c(2:24)], nfactors = 4, fm="ml", rotate="promax")  
print(fa\_ml\_4\_trn)

## Factor Analysis using method = ml  
## Call: fa(r = data[c(2:24)], nfactors = 4, rotate = "promax", fm = "ml")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## ML3 ML1 ML2 ML4 h2 u2 com  
## Question\_01 0.64 -0.03 0.12 0.16 0.405 0.59 1.2  
## Question\_02 -0.13 0.10 0.06 0.54 0.311 0.69 1.2  
## Question\_03 -0.50 0.05 -0.01 0.33 0.477 0.52 1.8  
## Question\_04 0.62 0.05 0.07 0.11 0.426 0.57 1.1  
## Question\_05 0.55 0.08 0.02 0.10 0.323 0.68 1.1  
## Question\_06 -0.30 0.99 -0.02 0.01 0.637 0.36 1.2  
## Question\_07 0.23 0.58 -0.03 0.04 0.514 0.49 1.3  
## Question\_08 0.08 -0.10 0.84 0.06 0.671 0.33 1.1  
## Question\_09 -0.02 0.00 0.12 0.61 0.365 0.64 1.1  
## Question\_10 0.04 0.39 0.01 -0.03 0.193 0.81 1.0  
## Question\_11 0.01 0.07 0.73 -0.11 0.663 0.34 1.1  
## Question\_12 0.53 0.25 -0.10 -0.02 0.459 0.54 1.5  
## Question\_13 0.12 0.54 0.08 -0.03 0.488 0.51 1.2  
## Question\_14 0.32 0.41 -0.03 0.01 0.424 0.58 1.9  
## Question\_15 0.16 0.29 0.14 -0.11 0.330 0.67 2.4  
## Question\_16 0.62 0.03 0.06 -0.01 0.464 0.54 1.0  
## Question\_17 0.17 0.04 0.64 0.03 0.582 0.42 1.2  
## Question\_18 0.23 0.63 -0.07 0.02 0.573 0.43 1.3  
## Question\_19 -0.22 -0.04 0.02 0.34 0.253 0.75 1.8  
## Question\_20 0.55 -0.20 0.01 -0.15 0.267 0.73 1.4  
## Question\_21 0.64 0.05 -0.03 -0.02 0.458 0.54 1.0  
## Question\_22 0.11 -0.12 -0.04 0.46 0.232 0.77 1.3  
## Question\_23 0.11 0.00 -0.07 0.33 0.095 0.90 1.3  
##   
## ML3 ML1 ML2 ML4  
## SS loadings 3.63 2.77 1.85 1.36  
## Proportion Var 0.16 0.12 0.08 0.06  
## Cumulative Var 0.16 0.28 0.36 0.42  
## Proportion Explained 0.38 0.29 0.19 0.14  
## Cumulative Proportion 0.38 0.67 0.86 1.00  
##   
## With factor correlations of   
## ML3 ML1 ML2 ML4  
## ML3 1.00 0.69 0.58 -0.49  
## ML1 0.69 1.00 0.56 -0.43  
## ML2 0.58 0.56 1.00 -0.25  
## ML4 -0.49 -0.43 -0.25 1.00  
##   
## Mean item complexity = 1.3  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## df null model = 253 with the objective function = 7.93 with Chi Square = 19534.27  
## df of the model are 167 and the objective function was 0.43   
##   
## The root mean square of the residuals (RMSR) is 0.03   
## The df corrected root mean square of the residuals is 0.03   
##   
## The harmonic n.obs is 2474 with the empirical chi square 794.58 with prob < 2.2e-82   
## The total n.obs was 2474 with Likelihood Chi Square = 1053.65 with prob < 1.5e-128   
##   
## Tucker Lewis Index of factoring reliability = 0.93  
## RMSEA index = 0.046 and the 90 % confidence intervals are 0.044 0.049  
## BIC = -251.22  
## Fit based upon off diagonal values = 0.99  
## Measures of factor score adequacy   
## ML3 ML1 ML2 ML4  
## Correlation of (regression) scores with factors 0.93 0.93 0.92 0.83  
## Multiple R square of scores with factors 0.87 0.87 0.84 0.70  
## Minimum correlation of possible factor scores 0.75 0.74 0.69 0.39

And once more, I will look at loadings above .3

print(fa\_ml\_4\_trn$loadings, cutoff = .3)

##   
## Loadings:  
## ML3 ML1 ML2 ML4   
## Question\_01 0.639   
## Question\_02 0.543  
## Question\_03 -0.498 0.329  
## Question\_04 0.621   
## Question\_05 0.547   
## Question\_06 0.990   
## Question\_07 0.577   
## Question\_08 0.836   
## Question\_09 0.612  
## Question\_10 0.389   
## Question\_11 0.733   
## Question\_12 0.527   
## Question\_13 0.542   
## Question\_14 0.325 0.407   
## Question\_15   
## Question\_16 0.624   
## Question\_17 0.638   
## Question\_18 0.625   
## Question\_19 0.335  
## Question\_20 0.545   
## Question\_21 0.643   
## Question\_22 0.460  
## Question\_23 0.333  
##   
## ML3 ML1 ML2 ML4  
## SS loadings 3.188 2.561 1.737 1.319  
## Proportion Var 0.139 0.111 0.076 0.057  
## Cumulative Var 0.139 0.250 0.326 0.383

This is better. The RMSEA = 0.046 (0.08 or less is preferred) The TLI = 0.93 (0.95 or higher is preferred, sometimes 0.90 is acceptable)

Not bad, but we do still have some cross loadings, and if the TLI could be above the 0.95 threshold, that would definitely be the preferred structure. Let’s run this again with a 5 factor model instead of 4.

fa\_ml\_5\_trn <- fa(data[c(2:24)], nfactors = 5, fm="ml", rotate="promax")  
  
print(fa\_ml\_5\_trn)

## Factor Analysis using method = ml  
## Call: fa(r = data[c(2:24)], nfactors = 5, rotate = "promax", fm = "ml")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## ML1 ML5 ML2 ML3 ML4 h2 u2 com  
## Question\_01 -0.13 0.86 0.07 -0.11 0.10 0.482 0.52 1.1  
## Question\_02 0.09 0.03 0.06 -0.15 0.54 0.308 0.69 1.3  
## Question\_03 0.03 -0.28 -0.02 -0.22 0.33 0.466 0.53 2.7  
## Question\_04 0.04 0.55 0.06 0.09 0.08 0.424 0.58 1.1  
## Question\_05 0.04 0.58 -0.01 0.02 0.07 0.339 0.66 1.0  
## Question\_06 1.06 -0.32 0.00 -0.08 0.05 0.623 0.38 1.2  
## Question\_07 0.68 -0.04 -0.01 0.22 0.10 0.533 0.47 1.3  
## Question\_08 -0.09 0.05 0.84 0.04 0.08 0.672 0.33 1.1  
## Question\_09 0.04 -0.06 0.13 0.05 0.67 0.389 0.61 1.1  
## Question\_10 0.37 0.17 -0.01 -0.15 -0.06 0.208 0.79 1.8  
## Question\_11 0.09 -0.04 0.74 0.04 -0.09 0.664 0.34 1.1  
## Question\_12 0.28 0.34 -0.09 0.15 -0.04 0.447 0.55 2.6  
## Question\_13 0.58 0.05 0.09 0.01 -0.03 0.489 0.51 1.1  
## Question\_14 0.43 0.24 -0.03 0.05 0.00 0.423 0.58 1.6  
## Question\_15 0.29 0.18 0.13 -0.04 -0.13 0.337 0.66 2.7  
## Question\_16 -0.06 0.75 0.01 -0.05 -0.07 0.520 0.48 1.0  
## Question\_17 0.05 0.14 0.63 0.04 0.04 0.581 0.42 1.1  
## Question\_18 0.68 0.12 -0.05 0.05 0.03 0.571 0.43 1.1  
## Question\_19 -0.07 -0.05 0.01 -0.16 0.33 0.251 0.75 1.6  
## Question\_20 -0.14 -0.04 0.06 0.66 -0.06 0.387 0.61 1.1  
## Question\_21 0.16 -0.03 0.01 0.80 0.14 0.669 0.33 1.2  
## Question\_22 -0.10 0.05 -0.03 0.09 0.48 0.237 0.76 1.2  
## Question\_23 0.02 0.07 -0.06 0.04 0.34 0.093 0.91 1.2  
##   
## ML1 ML5 ML2 ML3 ML4  
## SS loadings 2.98 2.48 1.85 1.46 1.34  
## Proportion Var 0.13 0.11 0.08 0.06 0.06  
## Cumulative Var 0.13 0.24 0.32 0.38 0.44  
## Proportion Explained 0.29 0.25 0.18 0.14 0.13  
## Cumulative Proportion 0.29 0.54 0.72 0.87 1.00  
##   
## With factor correlations of   
## ML1 ML5 ML2 ML3 ML4  
## ML1 1.00 0.75 0.56 0.56 -0.51  
## ML5 0.75 1.00 0.60 0.71 -0.49  
## ML2 0.56 0.60 1.00 0.42 -0.30  
## ML3 0.56 0.71 0.42 1.00 -0.51  
## ML4 -0.51 -0.49 -0.30 -0.51 1.00  
##   
## Mean item complexity = 1.4  
## Test of the hypothesis that 5 factors are sufficient.  
##   
## df null model = 253 with the objective function = 7.93 with Chi Square = 19534.27  
## df of the model are 148 and the objective function was 0.27   
##   
## The root mean square of the residuals (RMSR) is 0.02   
## The df corrected root mean square of the residuals is 0.02   
##   
## The harmonic n.obs is 2474 with the empirical chi square 453.33 with prob < 1e-32   
## The total n.obs was 2474 with Likelihood Chi Square = 670.33 with prob < 1.7e-67   
##   
## Tucker Lewis Index of factoring reliability = 0.954  
## RMSEA index = 0.038 and the 90 % confidence intervals are 0.035 0.041  
## BIC = -486.08  
## Fit based upon off diagonal values = 1  
## Measures of factor score adequacy   
## ML1 ML5 ML2 ML3 ML4  
## Correlation of (regression) scores with factors 0.94 0.93 0.92 0.89 0.85  
## Multiple R square of scores with factors 0.88 0.86 0.84 0.80 0.71  
## Minimum correlation of possible factor scores 0.77 0.73 0.69 0.59 0.43

And now, let’s look only at loadings of .30 or higher

print(fa\_ml\_5\_trn$loadings, cutoff = .3)

##   
## Loadings:  
## ML1 ML5 ML2 ML3 ML4   
## Question\_01 0.859   
## Question\_02 0.538  
## Question\_03 0.331  
## Question\_04 0.554   
## Question\_05 0.577   
## Question\_06 1.057 -0.320   
## Question\_07 0.677   
## Question\_08 0.837   
## Question\_09 0.666  
## Question\_10 0.373   
## Question\_11 0.738   
## Question\_12 0.340   
## Question\_13 0.584   
## Question\_14 0.433   
## Question\_15   
## Question\_16 0.754   
## Question\_17 0.634   
## Question\_18 0.676   
## Question\_19 0.328  
## Question\_20 0.664   
## Question\_21 0.802   
## Question\_22 0.483  
## Question\_23 0.339  
##   
## ML1 ML5 ML2 ML3 ML4  
## SS loadings 2.977 2.418 1.721 1.328 1.399  
## Proportion Var 0.129 0.105 0.075 0.058 0.061  
## Cumulative Var 0.129 0.235 0.309 0.367 0.428

The fit indices are improved, and we’re down to just one cross-loading, but one factor only has 2 variables, one has 3 (right at the minimum requires number). I’m going to try principal axis and see if that looks any better.

I think that, based on the small quantity of variables loaded onto two of the factors, a 6-factor model does not make sense to examine. I will stick to the 4 and 5 factor models and just see if the method changes the results significantly between ML and PA.

fa\_pa\_4\_trn <- fa(data[c(2:24)], nfactors = 4, fm="pa", rotate="promax")  
  
print(fa\_pa\_4\_trn)

## Factor Analysis using method = pa  
## Call: fa(r = data[c(2:24)], nfactors = 4, rotate = "promax", fm = "pa")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## PA1 PA3 PA4 PA2 h2 u2 com  
## Question\_01 0.58 0.01 0.15 0.16 0.39 0.61 1.3  
## Question\_02 -0.18 0.10 0.08 0.50 0.30 0.70 1.4  
## Question\_03 -0.51 0.04 -0.01 0.31 0.48 0.52 1.7  
## Question\_04 0.59 0.08 0.08 0.12 0.42 0.58 1.2  
## Question\_05 0.49 0.11 0.04 0.09 0.32 0.68 1.2  
## Question\_06 -0.30 0.98 -0.04 0.00 0.61 0.39 1.2  
## Question\_07 0.21 0.58 -0.03 0.04 0.51 0.49 1.3  
## Question\_08 0.08 -0.11 0.84 0.05 0.67 0.33 1.1  
## Question\_09 -0.05 -0.01 0.12 0.59 0.36 0.64 1.1  
## Question\_10 0.01 0.41 0.02 -0.03 0.20 0.80 1.0  
## Question\_11 0.03 0.05 0.72 -0.10 0.65 0.35 1.1  
## Question\_12 0.50 0.29 -0.10 -0.01 0.46 0.54 1.7  
## Question\_13 0.09 0.57 0.07 -0.03 0.49 0.51 1.1  
## Question\_14 0.29 0.44 -0.03 0.00 0.42 0.58 1.7  
## Question\_15 0.14 0.31 0.14 -0.11 0.33 0.67 2.1  
## Question\_16 0.56 0.07 0.09 -0.01 0.46 0.54 1.1  
## Question\_17 0.14 0.04 0.66 0.02 0.59 0.41 1.1  
## Question\_18 0.20 0.65 -0.07 0.02 0.57 0.43 1.2  
## Question\_19 -0.24 -0.05 0.03 0.33 0.26 0.74 1.9  
## Question\_20 0.60 -0.22 0.00 -0.12 0.28 0.72 1.3  
## Question\_21 0.69 0.04 -0.03 0.01 0.48 0.52 1.0  
## Question\_22 0.12 -0.13 -0.05 0.49 0.26 0.74 1.3  
## Question\_23 0.12 0.00 -0.09 0.35 0.11 0.89 1.4  
##   
## PA1 PA3 PA4 PA2  
## SS loadings 3.47 2.93 1.89 1.32  
## Proportion Var 0.15 0.13 0.08 0.06  
## Cumulative Var 0.15 0.28 0.36 0.42  
## Proportion Explained 0.36 0.30 0.20 0.14  
## Cumulative Proportion 0.36 0.67 0.86 1.00  
##   
## With factor correlations of   
## PA1 PA3 PA4 PA2  
## PA1 1.00 0.69 0.57 -0.46  
## PA3 0.69 1.00 0.58 -0.41  
## PA4 0.57 0.58 1.00 -0.23  
## PA2 -0.46 -0.41 -0.23 1.00  
##   
## Mean item complexity = 1.3  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## df null model = 253 with the objective function = 7.93 with Chi Square = 19534.27  
## df of the model are 167 and the objective function was 0.43   
##   
## The root mean square of the residuals (RMSR) is 0.02   
## The df corrected root mean square of the residuals is 0.03   
##   
## The harmonic n.obs is 2474 with the empirical chi square 779.09 with prob < 1e-79   
## The total n.obs was 2474 with Likelihood Chi Square = 1065.48 with prob < 1e-130   
##   
## Tucker Lewis Index of factoring reliability = 0.929  
## RMSEA index = 0.047 and the 90 % confidence intervals are 0.044 0.049  
## BIC = -239.39  
## Fit based upon off diagonal values = 0.99  
## Measures of factor score adequacy   
## PA1 PA3 PA4 PA2  
## Correlation of (regression) scores with factors 0.93 0.93 0.92 0.83  
## Multiple R square of scores with factors 0.87 0.87 0.84 0.68  
## Minimum correlation of possible factor scores 0.73 0.75 0.69 0.37

And look at loadings above .30

print(fa\_pa\_4\_trn$loadings, cutoff = .3)

##   
## Loadings:  
## PA1 PA3 PA4 PA2   
## Question\_01 0.579   
## Question\_02 0.505  
## Question\_03 -0.512 0.311  
## Question\_04 0.586   
## Question\_05 0.494   
## Question\_06 0.979   
## Question\_07 0.583   
## Question\_08 0.842   
## Question\_09 0.593  
## Question\_10 0.414   
## Question\_11 0.724   
## Question\_12 0.495   
## Question\_13 0.570   
## Question\_14 0.439   
## Question\_15 0.315   
## Question\_16 0.559   
## Question\_17 0.657   
## Question\_18 0.655   
## Question\_19 0.329  
## Question\_20 0.600   
## Question\_21 0.691   
## Question\_22 0.486  
## Question\_23 0.348  
##   
## PA1 PA3 PA4 PA2  
## SS loadings 3.013 2.721 1.779 1.261  
## Proportion Var 0.131 0.118 0.077 0.055  
## Cumulative Var 0.131 0.249 0.327 0.382

Fit statistics are quite comparable to what they were for the ML 4-factor model. Last one, I’ll look at PA 5-factor model.

fa\_pa\_5\_trn <- fa(data[c(2:24)], nfactors = 5, fm="pa", rotate="promax")  
  
print(fa\_pa\_5\_trn)

## Factor Analysis using method = pa  
## Call: fa(r = data[c(2:24)], nfactors = 5, rotate = "promax", fm = "pa")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## PA3 PA1 PA4 PA5 PA2 h2 u2 com  
## Question\_01 -0.13 0.84 0.06 -0.09 0.09 0.48 0.52 1.1  
## Question\_02 0.08 0.04 0.06 -0.19 0.50 0.29 0.71 1.4  
## Question\_03 0.03 -0.25 -0.02 -0.27 0.31 0.47 0.53 2.9  
## Question\_04 0.04 0.56 0.05 0.09 0.08 0.43 0.57 1.1  
## Question\_05 0.03 0.59 -0.01 -0.01 0.04 0.34 0.66 1.0  
## Question\_06 1.05 -0.30 0.00 -0.11 0.04 0.61 0.39 1.2  
## Question\_07 0.67 -0.03 0.00 0.18 0.08 0.52 0.48 1.2  
## Question\_08 -0.09 0.05 0.83 0.06 0.07 0.67 0.33 1.1  
## Question\_09 0.03 -0.06 0.14 0.02 0.64 0.38 0.62 1.1  
## Question\_10 0.38 0.18 -0.01 -0.18 -0.07 0.21 0.79 2.0  
## Question\_11 0.09 -0.04 0.73 0.06 -0.08 0.66 0.34 1.1  
## Question\_12 0.31 0.29 -0.09 0.20 -0.01 0.45 0.55 2.9  
## Question\_13 0.61 0.01 0.09 0.03 -0.02 0.49 0.51 1.1  
## Question\_14 0.45 0.20 -0.03 0.07 0.01 0.42 0.58 1.4  
## Question\_15 0.30 0.17 0.12 -0.03 -0.13 0.34 0.66 2.4  
## Question\_16 -0.05 0.74 0.01 -0.05 -0.09 0.52 0.48 1.0  
## Question\_17 0.06 0.12 0.64 0.04 0.03 0.59 0.41 1.1  
## Question\_18 0.70 0.08 -0.05 0.07 0.04 0.57 0.43 1.1  
## Question\_19 -0.08 -0.01 0.01 -0.20 0.32 0.26 0.74 1.8  
## Question\_20 -0.14 -0.06 0.07 0.70 -0.02 0.40 0.60 1.1  
## Question\_21 0.16 -0.01 0.03 0.74 0.14 0.62 0.38 1.2  
## Question\_22 -0.10 0.04 -0.03 0.10 0.51 0.26 0.74 1.2  
## Question\_23 0.03 0.05 -0.07 0.07 0.36 0.11 0.89 1.2  
##   
## PA3 PA1 PA4 PA5 PA2  
## SS loadings 3.06 2.36 1.84 1.53 1.30  
## Proportion Var 0.13 0.10 0.08 0.07 0.06  
## Cumulative Var 0.13 0.24 0.32 0.38 0.44  
## Proportion Explained 0.30 0.23 0.18 0.15 0.13  
## Cumulative Proportion 0.30 0.54 0.72 0.87 1.00  
##   
## With factor correlations of   
## PA3 PA1 PA4 PA5 PA2  
## PA3 1.00 0.76 0.56 0.58 -0.50  
## PA1 0.76 1.00 0.60 0.70 -0.46  
## PA4 0.56 0.60 1.00 0.40 -0.28  
## PA5 0.58 0.70 0.40 1.00 -0.50  
## PA2 -0.50 -0.46 -0.28 -0.50 1.00  
##   
## Mean item complexity = 1.4  
## Test of the hypothesis that 5 factors are sufficient.  
##   
## df null model = 253 with the objective function = 7.93 with Chi Square = 19534.27  
## df of the model are 148 and the objective function was 0.28   
##   
## The root mean square of the residuals (RMSR) is 0.02   
## The df corrected root mean square of the residuals is 0.02   
##   
## The harmonic n.obs is 2474 with the empirical chi square 440.55 with prob < 7.9e-31   
## The total n.obs was 2474 with Likelihood Chi Square = 680.89 with prob < 2.8e-69   
##   
## Tucker Lewis Index of factoring reliability = 0.953  
## RMSEA index = 0.038 and the 90 % confidence intervals are 0.035 0.041  
## BIC = -475.52  
## Fit based upon off diagonal values = 1  
## Measures of factor score adequacy   
## PA3 PA1 PA4 PA5 PA2  
## Correlation of (regression) scores with factors 0.94 0.93 0.92 0.89 0.84  
## Multiple R square of scores with factors 0.89 0.86 0.84 0.79 0.70  
## Minimum correlation of possible factor scores 0.77 0.72 0.68 0.58 0.41

And look at loadings above .30

print(fa\_pa\_5\_trn$loadings, cutoff = .3)

##   
## Loadings:  
## PA3 PA1 PA4 PA5 PA2   
## Question\_01 0.837   
## Question\_02 0.498  
## Question\_03 0.312  
## Question\_04 0.564   
## Question\_05 0.594   
## Question\_06 1.052 -0.304   
## Question\_07 0.669   
## Question\_08 0.830   
## Question\_09 0.641  
## Question\_10 0.379   
## Question\_11 0.725   
## Question\_12 0.314   
## Question\_13 0.610   
## Question\_14 0.454   
## Question\_15 0.302   
## Question\_16 0.742   
## Question\_17 0.638   
## Question\_18 0.698   
## Question\_19 0.316  
## Question\_20 0.704   
## Question\_21 0.742   
## Question\_22 0.515  
## Question\_23 0.364  
##   
## PA3 PA1 PA4 PA5 PA2  
## SS loadings 3.064 2.301 1.696 1.365 1.342  
## Proportion Var 0.133 0.100 0.074 0.059 0.058  
## Cumulative Var 0.133 0.233 0.307 0.366 0.425

Once more, there is a factor with only two variables and one with three. I’m leaning toward a 4-factor model because of this, but I will still take a closer look at the four options I have run so far (4-factor ML, 5-factor ML, 4-factor PA, 5-factor PA). I may also remove some items if they seem to be cross-loading, acting unusually, only associated with low loadings overall, or if the question is way too different or way too redundant compared to the others.

So now, I will input these four into one excel file to better examine the pros and cons of each.

fa\_ml\_4faclds <- as.data.frame(round(unclass(fa\_ml\_4\_trn$loadings), 3))  
  
fa\_ml\_4faclds

## ML3 ML1 ML2 ML4  
## Question\_01 0.639 -0.027 0.123 0.162  
## Question\_02 -0.131 0.097 0.065 0.543  
## Question\_03 -0.498 0.049 -0.009 0.329  
## Question\_04 0.621 0.053 0.067 0.114  
## Question\_05 0.547 0.076 0.018 0.104  
## Question\_06 -0.297 0.990 -0.024 0.011  
## Question\_07 0.229 0.577 -0.033 0.039  
## Question\_08 0.080 -0.097 0.836 0.060  
## Question\_09 -0.024 -0.004 0.117 0.612  
## Question\_10 0.040 0.389 0.012 -0.030  
## Question\_11 0.008 0.068 0.733 -0.107  
## Question\_12 0.527 0.251 -0.104 -0.024  
## Question\_13 0.118 0.542 0.084 -0.034  
## Question\_14 0.325 0.407 -0.034 0.006  
## Question\_15 0.162 0.291 0.137 -0.111  
## Question\_16 0.624 0.025 0.060 -0.006  
## Question\_17 0.169 0.038 0.638 0.032  
## Question\_18 0.235 0.625 -0.067 0.023  
## Question\_19 -0.223 -0.044 0.023 0.335  
## Question\_20 0.545 -0.202 0.007 -0.152  
## Question\_21 0.643 0.053 -0.025 -0.020  
## Question\_22 0.108 -0.125 -0.040 0.460  
## Question\_23 0.110 0.001 -0.074 0.333

openxlsx::write.xlsx(fa\_ml\_4faclds,"~/Documents/DirectoryR/UGA IOMP 2023 AA/fa\_ml\_4faclds.xlsx")

fa\_ml\_5faclds <- as.data.frame(round(unclass(fa\_ml\_5\_trn$loadings), 3))  
  
fa\_ml\_5faclds

## ML1 ML5 ML2 ML3 ML4  
## Question\_01 -0.132 0.859 0.065 -0.113 0.095  
## Question\_02 0.089 0.032 0.055 -0.154 0.538  
## Question\_03 0.034 -0.281 -0.017 -0.218 0.331  
## Question\_04 0.040 0.554 0.055 0.093 0.081  
## Question\_05 0.038 0.577 -0.009 0.016 0.066  
## Question\_06 1.057 -0.320 -0.003 -0.083 0.050  
## Question\_07 0.677 -0.044 -0.006 0.216 0.098  
## Question\_08 -0.087 0.053 0.837 0.041 0.078  
## Question\_09 0.039 -0.058 0.134 0.050 0.666  
## Question\_10 0.373 0.168 -0.009 -0.153 -0.062  
## Question\_11 0.087 -0.040 0.738 0.037 -0.087  
## Question\_12 0.282 0.340 -0.090 0.155 -0.038  
## Question\_13 0.584 0.050 0.093 0.006 -0.029  
## Question\_14 0.433 0.239 -0.031 0.052 -0.001  
## Question\_15 0.287 0.179 0.127 -0.039 -0.134  
## Question\_16 -0.059 0.754 0.010 -0.046 -0.072  
## Question\_17 0.050 0.135 0.634 0.039 0.041  
## Question\_18 0.676 0.118 -0.054 0.054 0.032  
## Question\_19 -0.066 -0.047 0.010 -0.158 0.328  
## Question\_20 -0.135 -0.042 0.058 0.664 -0.058  
## Question\_21 0.165 -0.033 0.012 0.802 0.140  
## Question\_22 -0.105 0.045 -0.027 0.089 0.483  
## Question\_23 0.016 0.071 -0.064 0.042 0.339

openxlsx::write.xlsx(fa\_ml\_5faclds,"~/Documents/DirectoryR/UGA IOMP 2023 AA/fa\_ml\_5faclds.xlsx")

fa\_pa\_4faclds <- as.data.frame(round(unclass(fa\_pa\_4\_trn$loadings), 3))  
  
fa\_pa\_4faclds

## PA1 PA3 PA4 PA2  
## Question\_01 0.579 0.012 0.147 0.156  
## Question\_02 -0.179 0.097 0.077 0.505  
## Question\_03 -0.512 0.041 -0.011 0.311  
## Question\_04 0.586 0.082 0.082 0.117  
## Question\_05 0.494 0.107 0.041 0.092  
## Question\_06 -0.298 0.979 -0.042 0.000  
## Question\_07 0.215 0.583 -0.035 0.036  
## Question\_08 0.080 -0.114 0.842 0.049  
## Question\_09 -0.049 -0.010 0.118 0.593  
## Question\_10 0.010 0.414 0.015 -0.033  
## Question\_11 0.027 0.053 0.724 -0.105  
## Question\_12 0.495 0.293 -0.102 -0.012  
## Question\_13 0.093 0.570 0.073 -0.033  
## Question\_14 0.289 0.439 -0.032 0.005  
## Question\_15 0.139 0.315 0.140 -0.111  
## Question\_16 0.559 0.074 0.092 -0.012  
## Question\_17 0.140 0.042 0.657 0.019  
## Question\_18 0.204 0.655 -0.073 0.023  
## Question\_19 -0.236 -0.047 0.025 0.329  
## Question\_20 0.600 -0.218 -0.004 -0.118  
## Question\_21 0.691 0.036 -0.035 0.006  
## Question\_22 0.119 -0.128 -0.047 0.486  
## Question\_23 0.118 0.004 -0.087 0.348

openxlsx::write.xlsx(fa\_pa\_4faclds,"~/Documents/DirectoryR/UGA IOMP 2023 AA/fa\_pa\_4faclds.xlsx")

f

fa\_pa\_5faclds <- as.data.frame(round(unclass(fa\_pa\_5\_trn$loadings), 3))  
  
fa\_pa\_5faclds

## PA3 PA1 PA4 PA5 PA2  
## Question\_01 -0.128 0.837 0.061 -0.087 0.090  
## Question\_02 0.080 0.039 0.061 -0.194 0.498  
## Question\_03 0.028 -0.254 -0.017 -0.267 0.312  
## Question\_04 0.040 0.564 0.047 0.093 0.082  
## Question\_05 0.034 0.594 -0.012 -0.009 0.044  
## Question\_06 1.052 -0.304 -0.004 -0.114 0.041  
## Question\_07 0.669 -0.027 0.003 0.177 0.085  
## Question\_08 -0.089 0.052 0.830 0.055 0.071  
## Question\_09 0.034 -0.056 0.137 0.018 0.641  
## Question\_10 0.379 0.183 -0.014 -0.175 -0.067  
## Question\_11 0.090 -0.041 0.725 0.063 -0.078  
## Question\_12 0.314 0.288 -0.088 0.200 -0.011  
## Question\_13 0.610 0.013 0.090 0.029 -0.016  
## Question\_14 0.454 0.198 -0.026 0.073 0.005  
## Question\_15 0.302 0.169 0.124 -0.032 -0.128  
## Question\_16 -0.046 0.742 0.014 -0.051 -0.085  
## Question\_17 0.058 0.122 0.638 0.041 0.030  
## Question\_18 0.698 0.079 -0.051 0.073 0.042  
## Question\_19 -0.075 -0.013 0.006 -0.200 0.316  
## Question\_20 -0.136 -0.059 0.066 0.704 -0.022  
## Question\_21 0.160 -0.005 0.028 0.742 0.139  
## Question\_22 -0.099 0.040 -0.029 0.101 0.515  
## Question\_23 0.026 0.055 -0.071 0.070 0.364

openxlsx::write.xlsx(fa\_pa\_5faclds,"~/Documents/DirectoryR/UGA IOMP 2023 AA/fa\_pa\_5faclds.xlsx")

Additionally, I copied the fit statistics from all four models to compare and contrast them more eaasily.

The examinations of the four potential models were very useful to determine how to proceed. The first note was that the difference between ML and PA was very slight, so I chose to proceed with my initial selected method of ML. However, it was useful to see both methods and gain some context for how similar the results are. Also, the fit indices only improved slightly when moving from the 4-factor to the 5-factor model. The major issue with the 5-factor model is the factor that only has two items - this does not meet the standard of 3 items per factor. With this in mind, the ML method 4-factor model is likely what I will proceed with.

The examinations of the 4-factor model still pointed to some potential problems. Namely: Q3 had cross-loadings on factors 3 and 4; additionally, it was the only negative loading in the entire model (sometimes the reverse scored items will act weirdly and be more challenging to interpret, so this may be why it is cross-loading). The loadings were also on the lower end (-0.498 and 0.329) relative to other loadings. Q14 had cross-loadings on factors 1 and 3; the loadings were relatively low as well (0.407, 0.325) Q15 did not have any factor loadings that reached the threshold of .30.

My plan is to examine those questions more carefully and see if any should be removed from the data. Q3 - Standard deviations excite me Q14 - Computers have minds of their own and deliberately go wrong whenever I use them Q15 - Computers are out to get me

I think Q3 should be removed. It is the only reverse scored item, which makes it stand out and potentially behave unusually within the dataset. It is also cross-loading on two factors that already have a sufficient number of other factors. Q14 and Q15 are very similar. They both have the ‘evil sentient computer’ sentiment behind them. It seems that other factors are more strongly picking up frustration/ineptitude with technology (like Q7, for instance, ‘All computers hate me’). Let’s see what happens when they’re both removed.

datatrain\_mod <- datatrain %>%  
 dplyr::select(-c(Question\_03,Question\_14,Question\_15))

I’m rerunning the 4-factor ML method EFA with the three questions removed. I adjusted the column numbers accordingly in the code below.

fa\_ml\_4\_trn\_MOD <- fa(datatrain\_mod[c(2:21)], nfactors =4, fm="ml", rotate="promax")   
  
print(fa\_ml\_4\_trn\_MOD)

## Factor Analysis using method = ml  
## Call: fa(r = datatrain\_mod[c(2:21)], nfactors = 4, rotate = "promax",   
## fm = "ml")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## ML3 ML1 ML2 ML4 h2 u2 com  
## Question\_01 0.66 -0.05 0.10 0.15 0.41 0.59 1.2  
## Question\_02 -0.09 0.07 0.07 0.53 0.29 0.71 1.1  
## Question\_04 0.60 0.07 0.05 0.09 0.42 0.58 1.1  
## Question\_05 0.59 0.04 0.00 0.10 0.34 0.66 1.1  
## Question\_06 -0.24 0.92 0.00 0.02 0.61 0.39 1.1  
## Question\_07 0.27 0.54 -0.03 0.04 0.51 0.49 1.5  
## Question\_08 0.08 -0.07 0.80 0.06 0.64 0.36 1.0  
## Question\_09 0.00 -0.01 0.11 0.62 0.37 0.63 1.1  
## Question\_10 0.05 0.37 0.02 -0.04 0.19 0.81 1.1  
## Question\_11 -0.02 0.10 0.74 -0.11 0.67 0.33 1.1  
## Question\_12 0.50 0.25 -0.08 -0.06 0.46 0.54 1.6  
## Question\_13 0.15 0.51 0.09 -0.05 0.48 0.52 1.3  
## Question\_16 0.61 0.02 0.06 -0.03 0.46 0.54 1.0  
## Question\_17 0.17 0.05 0.63 0.04 0.58 0.42 1.2  
## Question\_18 0.23 0.61 -0.04 0.01 0.58 0.42 1.3  
## Question\_19 -0.21 -0.06 0.02 0.32 0.24 0.76 1.8  
## Question\_20 0.49 -0.20 0.03 -0.21 0.27 0.73 1.7  
## Question\_21 0.67 0.02 -0.04 -0.06 0.47 0.53 1.0  
## Question\_22 0.09 -0.12 -0.03 0.48 0.25 0.75 1.2  
## Question\_23 0.09 0.02 -0.09 0.34 0.10 0.90 1.3  
##   
## ML3 ML1 ML2 ML4  
## SS loadings 3.08 2.25 1.77 1.22  
## Proportion Var 0.15 0.11 0.09 0.06  
## Cumulative Var 0.15 0.27 0.35 0.42  
## Proportion Explained 0.37 0.27 0.21 0.15  
## Cumulative Proportion 0.37 0.64 0.85 1.00  
##   
## With factor correlations of   
## ML3 ML1 ML2 ML4  
## ML3 1.00 0.67 0.59 -0.45  
## ML1 0.67 1.00 0.50 -0.39  
## ML2 0.59 0.50 1.00 -0.24  
## ML4 -0.45 -0.39 -0.24 1.00  
##   
## Mean item complexity = 1.2  
## Test of the hypothesis that 4 factors are sufficient.  
##   
## df null model = 190 with the objective function = 6.4 with Chi Square = 11020.08  
## df of the model are 116 and the objective function was 0.35   
##   
## The root mean square of the residuals (RMSR) is 0.03   
## The df corrected root mean square of the residuals is 0.03   
##   
## The harmonic n.obs is 1731 with the empirical chi square 457.34 with prob < 4.8e-42   
## The total n.obs was 1731 with Likelihood Chi Square = 599.78 with prob < 2.7e-66   
##   
## Tucker Lewis Index of factoring reliability = 0.927  
## RMSEA index = 0.049 and the 90 % confidence intervals are 0.045 0.053  
## BIC = -265.17  
## Fit based upon off diagonal values = 0.99  
## Measures of factor score adequacy   
## ML3 ML1 ML2 ML4  
## Correlation of (regression) scores with factors 0.93 0.92 0.91 0.82  
## Multiple R square of scores with factors 0.86 0.84 0.83 0.67  
## Minimum correlation of possible factor scores 0.72 0.69 0.67 0.33

Now once more, let’s look at loaadings above .30

print(fa\_ml\_4\_trn\_MOD$loadings, cutoff = .3)

##   
## Loadings:  
## ML3 ML1 ML2 ML4   
## Question\_01 0.656   
## Question\_02 0.534  
## Question\_04 0.604   
## Question\_05 0.588   
## Question\_06 0.923   
## Question\_07 0.536   
## Question\_08 0.795   
## Question\_09 0.618  
## Question\_10 0.369   
## Question\_11 0.740   
## Question\_12 0.499   
## Question\_13 0.510   
## Question\_16 0.614   
## Question\_17 0.626   
## Question\_18 0.613   
## Question\_19 0.317  
## Question\_20 0.492   
## Question\_21 0.668   
## Question\_22 0.475  
## Question\_23 0.338  
##   
## ML3 ML1 ML2 ML4  
## SS loadings 2.775 2.063 1.635 1.223  
## Proportion Var 0.139 0.103 0.082 0.061  
## Cumulative Var 0.139 0.242 0.324 0.385

Things are looking good! There is no cross-loading, there is no item missing a >.30 loading, and each factor still has at least 3 items. This is the simple structure that I will proceed with.

I’ll talk about the fit statistics and related details later on as well. For now, I will explore what the dimensions of the scale look like to help shed some light on the underlying constructs of each.

1. My final simple structure has 4 factors. Here are the names for each factor, followed by their corresponding items:

Factor 1: Fear and Incompetence Toward Computers Q6 - I have little experience of computers Q7 - All computers hate me Q10 - Computers are useful only for playing games Q13 - I worry that I will cause irreparable damage because of my incompetence with computers Q18 - SPSS always crashes when I try to use it

Factor 2: Fear and Incompetence Toward Mathematics Q8 - I have never been good at mathematics Q11 - I did badly at mathematics at school Q17 - I slip into a coma whenever I see an equation

Factor 3: Fear and Incompetence Toward Statistics Q1 - Statistics makes me cry Q4 - I dream that Pearson is attacking me with correlation coefficients Q5 - I don’t understand statistics Q12 - People try to tell you that SPSS makes statistics easier to understand but it doesn’t Q16 - I weep openly at the mention of central tendency Q20 - I can’t sleep for thoughts of eigenvectors Q21 - I wake up under my duvet thinking that I am trapped under a normal distribution

Factor 4: Social Fear Toward Statistics  
Q2 - My friends will think I’m stupid for not being able to cope with SPSS Q9 - My friends are better at statistics than me Q19 - Everybody looks at me when I use SPSS Q22 - My friends are better at SPSS than I am Q23 - If I’m good at statistics my friends will think I’m a nerd

The factor names are a tad long, but I do feel that they accurately and pretty fully capture the underlying constructs of each. One way to condense them could be to have some overall description language before the entire list of factors, so it would read: The self-assessment questionnaire asks respondents about their fears and negative attitudes about: computers, mathematics, statistics, and social pressures around statistics. The titles as written above also do the job though!

I also noticed how well the 4-factor model grouped the items together. The identification of each underlying factor was not difficult. This is a good sign that my simple structure is logical, interpretable, and accurate.

Below is some information about the fit statistics for the final model. As a reminder, the final model is the 4-factor model, ML method, with the three items removed.

1. The Tucker Lewis Index ranges from 0 to 1, and it provides information about the goodness of fit for a factor model. For the final model, the TLI = 0.927. The preferred standard for a good TLI is 0.95 or greater. However, a value of 0.90 or greater may be acceptable as well. It is noteworthy that the obtained value does not reach the preferred standard, but it is above 0.90 so it is considered acceptable. Although the TLI was higher in the 5 factor model, the results were problematic in other ways (they were not as easily interpretable, and one factor would not meet the criteria for the minimum number of items, only having two items loading high enough on that factor instead of the standard of 3 or more).
2. The RMSEA Index is another method of evaluating the goodness of fit for a factor model. It ranges from 0 to infinity, with lower numbers indicating better fit. For the final model, the RMSEA = 0.049, 90%CI [0.045,0.053]. The preferred standard is less than 0.08. Some standards of practice even call for the RMSEA to be 0.05 or below. The good news is that our obtained RMSEA is lower than both.

The conclusion from the TLI and RMSEA is that the 4-factor model is an acceptable fit for the data.

1. Here is the information about my scales: I’ve chosen to name the overall scale ‘Fear and Incompetence Toward Mathematics and Technology’ (FITMAT) Scale Based on my factor information above, the scale is comprised of 4 subscales: The ‘Fear and Incompetence Toward Computers’ (FITC) Scale The ‘Fear and Incompetence Toward Mathematics’ (FITM) Scale The ‘Fear and Incompetence Toward Statistics’ (FITS) Scale And the ‘Social Fear Toward Statistics’ (SFTS) Scale

All four are scored on a scale of 1 (Strongly agree) to 5 (Strongly disagree). For the full list of questions associated with each scale, see the answer to question 6 above. I will pop the short list of what question numbers correspond with each factor just so I have this info readily available for the scale analysis below. Note that the first set of numbers is the Question\_#, and the second set of numbers is the new column number due to the removal of three items. This short list will also help me keep the column numbers straight as I move forward. Factor 1 - FITC: 6, 7, 10, 13, 18 [corresponds to columns 5,6,9,12,15] Factor 2 - FITM: 8, 11, 17 [corresponds to columns 7,10,14] Factor 3 - FITS: 1, 4, 5, 12, 16, 20, 21 [corresponds to columns 1,3,4,11,13,17,18] Factor 4 - SFTS: 2, 9, 19, 22, 23 [corresponds to columns 2,8,16,19,20]

1. The last step is some scale analysis, evaluating the reliability of each subscale by calculating Cronbach’s alpha. To help with this, I will build a data set with the revised list of items only (making sure it doesn’t include the 3 removed items and the ID number column).

dataitems <- datatrain\_mod %>%  
 dplyr::select(-c(ID))

Now I can run a quick check for any items that need reverse-scoring. I feel like I know the items well at this point, but I can double-check with the skimr package.

library(skimr)  
  
skim(dataitems)

Data summary

|  |  |
| --- | --- |
| Name | dataitems |
| Number of rows | 1731 |
| Number of columns | 20 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| numeric | 20 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Question\_01 | 0 | 1 | 2.37 | 0.81 | 1 | 2 | 2 | 3 | 5 | ▂▇▅▁▁ |
| Question\_02 | 0 | 1 | 1.63 | 0.85 | 1 | 1 | 1 | 2 | 5 | ▇▅▁▁▁ |
| Question\_04 | 0 | 1 | 2.77 | 0.93 | 1 | 2 | 3 | 3 | 5 | ▁▇▇▃▁ |
| Question\_05 | 0 | 1 | 2.70 | 0.93 | 1 | 2 | 3 | 3 | 5 | ▁▇▆▃▁ |
| Question\_06 | 0 | 1 | 2.21 | 1.09 | 1 | 1 | 2 | 3 | 5 | ▅▇▂▂▁ |
| Question\_07 | 0 | 1 | 2.91 | 1.06 | 1 | 2 | 3 | 4 | 5 | ▂▇▆▆▂ |
| Question\_08 | 0 | 1 | 2.25 | 0.86 | 1 | 2 | 2 | 3 | 5 | ▂▇▂▁▁ |
| Question\_09 | 0 | 1 | 2.86 | 1.25 | 1 | 2 | 3 | 4 | 5 | ▅▆▇▇▂ |
| Question\_10 | 0 | 1 | 2.27 | 0.84 | 1 | 2 | 2 | 3 | 5 | ▂▇▂▁▁ |
| Question\_11 | 0 | 1 | 2.24 | 0.84 | 1 | 2 | 2 | 3 | 5 | ▂▇▃▁▁ |
| Question\_12 | 0 | 1 | 3.15 | 0.88 | 1 | 3 | 3 | 4 | 5 | ▁▃▇▅▁ |
| Question\_13 | 0 | 1 | 2.43 | 0.91 | 1 | 2 | 2 | 3 | 5 | ▂▇▅▂▁ |
| Question\_16 | 0 | 1 | 2.87 | 0.89 | 1 | 2 | 3 | 3 | 5 | ▁▆▇▃▁ |
| Question\_17 | 0 | 1 | 2.45 | 0.85 | 1 | 2 | 2 | 3 | 5 | ▁▇▅▂▁ |
| Question\_18 | 0 | 1 | 2.56 | 1.02 | 1 | 2 | 2 | 3 | 5 | ▃▇▆▂▁ |
| Question\_19 | 0 | 1 | 2.30 | 1.07 | 1 | 1 | 2 | 3 | 5 | ▆▇▅▃▁ |
| Question\_20 | 0 | 1 | 3.62 | 1.02 | 1 | 3 | 4 | 4 | 5 | ▁▃▅▇▅ |
| Question\_21 | 0 | 1 | 3.16 | 0.95 | 1 | 2 | 3 | 4 | 5 | ▁▆▇▇▂ |
| Question\_22 | 0 | 1 | 2.89 | 1.02 | 1 | 2 | 3 | 4 | 5 | ▂▆▇▆▁ |
| Question\_23 | 0 | 1 | 3.43 | 1.04 | 1 | 3 | 4 | 4 | 5 | ▁▂▅▇▂ |

A few of the item histograms have less of a positive skew, namely 12 and 20, but when I evaluated the two questions, I do not believe they should be reverse scored – I just think they happened to have less of a level of agreement than other similar items. The items in question were: Q12 - People try to tell you that SPSS makes statistics easier to understand but it doesn’t Q20 - I can’t sleep for thoughts of eigenvectors

I’m mostly noting these so if something suspicious crops up in my reliability analyses, I have a good idea of where to start in the troubleshooting process.

Now I can group them together by factors and make keys for my reliability analyses. For now, I will not reverse score anything. \*\*\*Note that this corresponds to the column numbers above (not the question numbers)

fitmat\_keys\_list <- list(fitc = c(5, 6, 9, 12, 15),  
 fitm = c(7, 10, 14),  
 fits = c(1, 3, 4, 11, 13, 17, 18),  
 sfts = c(2, 8, 16, 19, 20)  
 )  
  
fitmat\_keys <- make.keys(dataitems, fitmat\_keys\_list, item.labels = colnames(dataitems))

And now I will score the items for each subscale and put them in their own dataframe.

scores <- scoreItems(fitmat\_keys, dataitems, impute = "none",   
 min = 1, max = 5, digits = 3)  
  
head(scores$scores)

## fitc fitm fits sfts  
## [1,] 2.2 1.000000 2.142857 2.4  
## [2,] 3.4 2.000000 3.285714 2.4  
## [3,] 3.8 2.333333 3.142857 2.2  
## [4,] 2.0 2.000000 2.000000 3.4  
## [5,] 2.0 2.000000 2.285714 3.6  
## [6,] 4.2 5.000000 4.571429 2.8

scores\_df <- as.data.frame(scores$scores)

Now I’m going to divide the data into its four factors/subscales. I’m going to use the dplyr package with the select function (used above to create the final list of items). This time, instead of -c() I will use c() since the number of items to include is fewer than the number of items to exclude from each data frame

FITC <- dataitems %>%  
 dplyr::select(c(Question\_06,Question\_07,Question\_10,Question\_13,Question\_18))  
  
FITM <- dataitems %>%  
 dplyr::select(c(Question\_08,Question\_11,Question\_17))  
  
FITS <- dataitems %>%  
 dplyr::select(c(Question\_01,Question\_04,Question\_05,Question\_12,Question\_16,Question\_20,Question\_21))  
  
SFTS <- dataitems %>%  
 dplyr::select(c(Question\_02,Question\_09,Question\_19,Question\_22,Question\_23))

Now I can start the process of evaluating each subscale and gaining information about it’s reliability. I’ll start with the FITC scale.

At first, I did play around with going right to the psych::alpha functions, but I got an error message. So I will actually make the keys for each subscale one at a time. I will leave the code above since doing it in aggregate may come in handy later on if I can get the code to work, but for now I will override that with the code below.

#fitmat\_keys\_list <- list(fitc=c(5, 6, 9, 12, 15))   
#fitmat\_keys <- make.keys(FITC, fitmat\_keys\_list, item.labels = colnames(FITC))

Ok, I got an error there that says my subscript is out of bounds. After some sleuthing, I found that this means I’m calling for column numbers that don’t exist (are ‘out of the bounds’ of my data). It turns out that these numbers do not need to match the column numbers from the main data set, instead they just need to be 1-X (with X being the number of items in that subscale). Lesson learned. Let’s try that again. (Once more, I have commented out the error code so it will knit, but leaving it there to understand what went wrong and how to fix it).

fitmat\_keys\_list <- list(fitc=c(1,2,3,4,5))   
fitmat\_keys <- make.keys(FITC, fitmat\_keys\_list, item.labels = colnames(FITC))

Ok that worked. Now I can run the initial analysis for reliability with the alpha() function of the psych package. The syntax psych::alpha ensures that I’m using the alpha function from psych and not from ggplot2.

FITC\_ALPHA <- psych::alpha(x = FITC[, abs(fitmat\_keys\_list$fitc)], keys = fitmat\_keys)

And now I can make FITC\_ALPHA into a dataframe and add some scale and item level statistics to get a full picture of its reliability.

FITC\_total <- round(as.data.frame(FITC\_ALPHA$total), 3)  
FITC\_alpha\_drop <- round(as.data.frame(FITC\_ALPHA$alpha.drop), 3)  
FITC\_item\_stat <- round(as.data.frame(FITC\_ALPHA$item.stats), 3)  
  
FITC\_ALPHA

##   
## Reliability analysis   
## Call: psych::alpha(x = FITC[, abs(fitmat\_keys\_list$fitc)], keys = fitmat\_keys)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.79 0.79 0.76 0.43 3.7 0.0077 2.5 0.73 0.46  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.78 0.79 0.81  
## Duhachek 0.78 0.79 0.81  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## Question\_06 0.73 0.73 0.68 0.41 2.7 0.0101 0.0128 0.39  
## Question\_07 0.74 0.74 0.69 0.41 2.8 0.0098 0.0127 0.39  
## Question\_10 0.80 0.80 0.76 0.51 4.1 0.0077 0.0013 0.52  
## Question\_13 0.75 0.74 0.70 0.42 2.9 0.0096 0.0139 0.42  
## Question\_18 0.72 0.72 0.67 0.39 2.6 0.0104 0.0102 0.39  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## Question\_06 1731 0.79 0.77 0.70 0.62 2.2 1.09  
## Question\_07 1731 0.77 0.76 0.67 0.60 2.9 1.06  
## Question\_10 1731 0.57 0.61 0.43 0.39 2.3 0.84  
## Question\_13 1731 0.74 0.75 0.67 0.59 2.4 0.91  
## Question\_18 1731 0.80 0.79 0.73 0.65 2.6 1.02  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## Question\_06 0.27 0.45 0.14 0.09 0.05 0  
## Question\_07 0.07 0.33 0.28 0.25 0.07 0  
## Question\_10 0.13 0.58 0.19 0.09 0.01 0  
## Question\_13 0.12 0.49 0.26 0.11 0.02 0  
## Question\_18 0.14 0.38 0.31 0.12 0.05 0

Below are some notes for interpreting the above results, along with the corresponding values from the info above:

Scale Info - \* raw\_alpha is alpha based upon the covariance (0.79) \* std.alpha is the standardized alpha based upon the correlations (0.79) \* G6(smc) is Guttman’s Lamda 6 reliability (0.76) \* average\_r is the average interitem correlation (0.43) \* median\_r is the median interitem correlation (0.46)

Item Info - (numbers are in order of Questions: 6,7,10,13,18) \* raw.r is the correlation of each item with the total score, not corrected for item overlap (0.79,0.77,0.57,0.74,0.80) \* std.r is the correlation of each item with the total score (not corrected for item overlap) if the items were all standardized (0.77,0.75,0.61,0.750.79) \* r.cor is item whole correlation corrected for item overlap and scale reliability (0.70,0.67,0.43,0.67,0.73) \* r.drop is item whole correlation for this item against the scale without this item (0.62,0.60,0.39,0.59,0.65)

The Cronbach’s alpha for the FITC scale is 0.79. An acceptable alpha is 0.70 or higher (0.80 or higher is even better). So an alpha = 0.79 is considered adequate. Additionally, the r.drop values are all lower than the current alpha. This means that removing any of these items would result in a less reliable scale, so I feel confident that the best move is to keep all five items of the FITC scale.

For FITM:

fitmat\_keys\_list <- list(fitm=c(1,2,3))   
fitmat\_keys <- make.keys(FITM, fitmat\_keys\_list, item.labels = colnames(FITM))

Note that I had to adjust the numbers to align with how many items are in the scale

FITM\_ALPHA <- psych::alpha(x = FITM[, abs(fitmat\_keys\_list$fitm)], keys = fitmat\_keys)

Now I will repeat the process I did for FITC of creating a dataframe with scale and item level statistics.

FITM\_total <- round(as.data.frame(FITM\_ALPHA$total), 3)  
FITM\_alpha\_drop <- round(as.data.frame(FITM\_ALPHA$alpha.drop), 3)  
FITM\_item\_stat <- round(as.data.frame(FITM\_ALPHA$item.stats), 3)  
  
FITM\_ALPHA

##   
## Reliability analysis   
## Call: psych::alpha(x = FITM[, abs(fitmat\_keys\_list$fitm)], keys = fitmat\_keys)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.82 0.82 0.76 0.61 4.7 0.0073 2.3 0.73 0.61  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.81 0.82 0.84  
## Duhachek 0.81 0.82 0.84  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## Question\_08 0.76 0.76 0.61 0.61 3.1 0.012 NA 0.61  
## Question\_11 0.74 0.74 0.59 0.59 2.9 0.012 NA 0.59  
## Question\_17 0.77 0.77 0.63 0.63 3.4 0.011 NA 0.63  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## Question\_08 1731 0.86 0.86 0.75 0.68 2.2 0.86  
## Question\_11 1731 0.87 0.87 0.77 0.69 2.2 0.84  
## Question\_17 1731 0.85 0.85 0.73 0.66 2.5 0.85  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## Question\_08 0.14 0.59 0.19 0.06 0.03 0  
## Question\_11 0.15 0.56 0.21 0.07 0.02 0  
## Question\_17 0.08 0.52 0.28 0.09 0.02 0

Once more, here’s some notable information from above: Scale Info - \* raw\_alpha (0.82) \* std.alpha (0.82) \* G6(smc) (0.76) \* average\_r (0.61) \* median\_r (0.61)

Item Info - (numbers are in order of Questions: 8,11,17) \* raw.r (0.86,0.87,0.85) \* std.r (0.86,0.87,0.85) \* r.cor (0.75,0.77,0.73) \* r.drop (0.68,0.69,0.66)

The Cronbach’s alpha of the FITM scale is 0.82. This meets the standard for an acceptable alpha (>=0.70), and it also meets the ‘better’ standard (>=0.80). Once more, this is good news. Additionally, the r.drop values indicate that no item removal would increase alpha, so all items of the FITM scale will remain. Also note that if one of the r.drop values would increase alpha, the decision to remove an item from this particular subscale would not be as cut and dry as others. Because this scale is already at the minimum accepted number of items (3), removal of an item would likely mean replacing that item and potentially require rerunning the EFA and reliability analyses. The good news is that this does not apply here.

FITS is next:

fitmat\_keys\_list <- list(fits=c(1,2,3,4,5,6,7))   
fitmat\_keys <- make.keys(FITS, fitmat\_keys\_list, item.labels = colnames(FITS))

FITS\_ALPHA <- psych::alpha(x = FITS[, abs(fitmat\_keys\_list$fits)], keys = fitmat\_keys)

Now I will repeat the process I did for the first two scales, creating a dataframe with scale and item level statistics.

FITS\_total <- round(as.data.frame(FITS\_ALPHA$total), 3)  
FITS\_alpha\_drop <- round(as.data.frame(FITS\_ALPHA$alpha.drop), 3)  
FITS\_item\_stat <- round(as.data.frame(FITS\_ALPHA$item.stats), 3)  
  
FITS\_ALPHA

##   
## Reliability analysis   
## Call: psych::alpha(x = FITS[, abs(fitmat\_keys\_list$fits)], keys = fitmat\_keys)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.8 0.81 0.79 0.37 4.2 0.0072 2.9 0.62 0.41  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.79 0.8 0.82  
## Duhachek 0.79 0.8 0.82  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## Question\_01 0.78 0.78 0.76 0.37 3.6 0.0082 0.0065 0.41  
## Question\_04 0.77 0.78 0.76 0.37 3.5 0.0085 0.0080 0.38  
## Question\_05 0.78 0.79 0.77 0.38 3.7 0.0081 0.0077 0.42  
## Question\_12 0.78 0.78 0.76 0.37 3.5 0.0084 0.0087 0.41  
## Question\_16 0.77 0.77 0.75 0.36 3.4 0.0086 0.0073 0.38  
## Question\_20 0.80 0.81 0.78 0.41 4.2 0.0073 0.0016 0.41  
## Question\_21 0.76 0.77 0.74 0.36 3.4 0.0088 0.0079 0.40  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## Question\_01 1731 0.66 0.68 0.61 0.54 2.4 0.81  
## Question\_04 1731 0.70 0.70 0.64 0.57 2.8 0.93  
## Question\_05 1731 0.66 0.66 0.57 0.51 2.7 0.93  
## Question\_12 1731 0.69 0.69 0.61 0.55 3.1 0.88  
## Question\_16 1731 0.71 0.72 0.66 0.59 2.9 0.89  
## Question\_20 1731 0.60 0.57 0.47 0.41 3.6 1.02  
## Question\_21 1731 0.74 0.73 0.68 0.62 3.2 0.95  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## Question\_01 0.10 0.53 0.29 0.06 0.02 0  
## Question\_04 0.05 0.37 0.37 0.16 0.04 0  
## Question\_05 0.05 0.44 0.30 0.18 0.03 0  
## Question\_12 0.02 0.20 0.47 0.24 0.07 0  
## Question\_16 0.03 0.33 0.42 0.17 0.05 0  
## Question\_20 0.02 0.15 0.25 0.38 0.21 0  
## Question\_21 0.01 0.27 0.35 0.29 0.08 0

Scale Info - \* raw\_alpha (0.80) \* std.alpha (0.81) \* G6(smc) (0.79) \* average\_r (0.37) \* median\_r (0.41)

Item Info - (numbers are in order of Questions: 1,4,5,12,16,20,21) \* raw.r (0.66,0.70,0.66,0.69,0.71,0.60,0.74) \* std.r (0.68,0.70,0.66,0.69,0.72,0.57,0.73) \* r.cor (0.61,0.64,0.57,0.61,0.66,0.47,0.68) \* r.drop (0.54,0.57,0.51,0.55,0.59,0.41,0.62)

The Cronbach’s alpha for FITS is 0.80. This meets the standard for an acceptable alpha (>=0.70), and it also meets the higher standard (>=0.80). Additionally, the r.drop values indicate that no item removal would increase alpha, so all items from the FITS scale will remain.

Last subscale, SFTS:

fitmat\_keys\_list <- list(sfts=c(1,2,3,4,5))   
fitmat\_keys <- make.keys(SFTS, fitmat\_keys\_list, item.labels = colnames(SFTS))

SFTS\_ALPHA <- psych::alpha(x = SFTS[, abs(fitmat\_keys\_list$sfts)], keys = fitmat\_keys)

Now I will repeat the process I did for the first two scales, creating a dataframe with scale and item level statistics.

SFTS\_total <- round(as.data.frame(SFTS\_ALPHA$total), 3)  
SFTS\_alpha\_drop <- round(as.data.frame(SFTS\_ALPHA$alpha.drop), 3)  
SFTS\_item\_stat <- round(as.data.frame(SFTS\_ALPHA$item.stats), 3)  
  
SFTS\_ALPHA

##   
## Reliability analysis   
## Call: psych::alpha(x = SFTS[, abs(fitmat\_keys\_list$sfts)], keys = fitmat\_keys)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.59 0.59 0.54 0.22 1.4 0.015 2.6 0.65 0.23  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.55 0.59 0.62  
## Duhachek 0.56 0.59 0.62  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## Question\_02 0.53 0.53 0.46 0.22 1.11 0.018 0.0027 0.22  
## Question\_09 0.49 0.49 0.42 0.19 0.96 0.020 0.0031 0.21  
## Question\_19 0.55 0.55 0.49 0.24 1.24 0.017 0.0069 0.24  
## Question\_22 0.51 0.51 0.45 0.21 1.04 0.019 0.0067 0.20  
## Question\_23 0.58 0.58 0.52 0.26 1.40 0.016 0.0027 0.24  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## Question\_02 1731 0.58 0.62 0.47 0.36 1.6 0.85  
## Question\_09 1731 0.71 0.67 0.55 0.42 2.9 1.25  
## Question\_19 1731 0.59 0.59 0.40 0.31 2.3 1.07  
## Question\_22 1731 0.64 0.65 0.50 0.39 2.9 1.02  
## Question\_23 1731 0.54 0.54 0.33 0.26 3.4 1.04  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## Question\_02 0.55 0.32 0.08 0.04 0.01 0  
## Question\_09 0.19 0.20 0.24 0.28 0.08 0  
## Question\_19 0.27 0.35 0.22 0.14 0.02 0  
## Question\_22 0.09 0.27 0.34 0.26 0.04 0  
## Question\_23 0.06 0.12 0.27 0.43 0.12 0

Scale Info - \* raw\_alpha (0.59) \* std.alpha (0.59) \* G6(smc) (0.54) \* average\_r (0.22) \* median\_r (0.23)

Item Info - (numbers are in order of Questions: 2,9,19,22,23) \* raw.r (0.58,0.71,0.59,0.64,0.54) \* std.r (0.62,0.67,0.59,0.65,0.54) \* r.cor (0.47,0.55,0.40,0.50,0.33) \* r.drop (0.36,0.42,0.31,0.39,0.26)

The Cronbach’s alpha for SFTS is 0.59. This does not meet the standard for acceptable reliability since it is not greater than or equal to 0.70. Also, the r.drop values indicate that removing any of the five items will not lead to an increase in reliability. There are a few methods I could explore to increase the alpha of my SFTS scale. The first step is straightforward, which is simply to make sure that I have input everything correctly. I checked my work for errors and didn’t see any in the code above. I double-checked my scales and the five items in the analysis align with the five items that loaded onto this factor. The next step is to reexamine the reverse-scoring information to see if any of the five items should be reverse scored. I reexamined the histograms, looked at the five items, and do not see adequate evidence that any should be reverse scored.

The following potential steps would be more involved, with some requiring extensive work and re-analyzing of the data. I could redistribute the survey, as is, to another sample to see if the results are consistent with my above finding. Namely, does this SFTS scale consistently have inadequately low reliability, or is this current data set more anomalous? Is this data set reflective of the ‘exception’ or the ‘rule’? If it’s the exception based on other datasets, then I will take the above findings as an anomaly and move forward. If it aligns with other findings, then I can move to some of the steps below. I could add one or two well-constructed items to the subscale, and this often leads to increase in reliability. Unfortunately, this would also require that I go through a round of item writing, redistribute the questions, and repeat the full process of EFA and reliability analyses above. I could rewrite any items that may be performing poorly to see if the rewrites would increase reliability. Once more, this would require redistributing the survey and re-running analyses. However, these modifications may make the questions more clear, or perhaps more applicable (could changing ‘SPSS’ to ‘R’ be more accurate based on what software people are using to run statistical analyses?)

I could also reexamine my EFA options and see if a three-factor model would fit and increase the reliability of the subscales. However, I would need to ensure that my TLI is still high and my RMSEA is still low to indicate a good fit for a three factor model. Additionally, when examining the questions for the four factor model, I feel that the division of questions by factors makes logical sense, and I don’t foresee my results having clear delineations of three factors fitting into three distinct categories/subscales. I could potentially remove a few items from this scale and see if this helps with the alignment of a three-factor model. I could choose to remove the entire subscale. This would require repeating the EFA and subscale analyses. I would also need to ensure that the data collected from this subscale is not providing essential information. This is a major decision that I would not make lightly, and it is not one that I would recommend in a majority of cases. The scale is highly different from the other three scales, picking up on a social fear around statistics instead of a more internal/personal fear or anxiety around something. But I question if that is sufficient for deciding to remove the whole subscale. A better solution would be to find a way to remedy it. Lastly, I could leave the lower reliability scale in the measure, but I would need to fully document the reliability results and have a good reason for continuing to use a subscale that does not meet the standards for reliability. Once more, this is not a decision that I would recommend. In my summary below, I give final recommendations regarding methods to increase the reliability of the SFTS scale. These recommendations are out of the scope of this project, but they do point to valuable next steps for assessing and improving the subscale.

In summation regarding the reliability analyses for the subscales, the FITC scale meets the adequate standard for reliability (0.79), and the FITM and FITS scales both meet the higher ‘good’ standard for reliability (0.82 and 0.80, respectively). The reliability results for the FITC, FITM, and FITS did not flag any items for potential removal. The ‘problem child’ is the SFTS scale; it did not reach the minimum acceptable standard for reliability (0.59), and the results did not flag any items for potential removal. From the potential options listed above regarding methods for increasing the reliability of SFTS, the three recommended paths forward would be: - Rerun the EFA using a three factor model, potentially removing additional items to ensure each factor captures a clear and delineated construct with appropriately high loadings and no cross loadings. - Readminister the measure as it currently stands to see if the results are consistent of inconsistent with the above findings. - Reword current items in the SFTS scale and add 1-2 new items to the scale as well. Readminister the measure and rerun the full EFA and reliability analyses above. This would require more work and effort, but the process of scale development and refinement is cyclical, and steps often need to be repeated to ensure that all scales and subscales are reliable and comprised of accurate, thorough, and well-performing items.

Overall, the four-factor model appears to be a good fit for the 23 question scale. The factor analysis picked up on four distinct factors that grouped items accordingly. Three of the four scales emerged with adequate reliability, and one scale emerged with a problematic low Cronbach’s alpha that will require further adjustments and evaluations.