

ReCentering Psych Stats: Psychometrics

Lynette H. Bikos, PhD, ABPP

16 Feb 2022

Contents

BOOK COVER

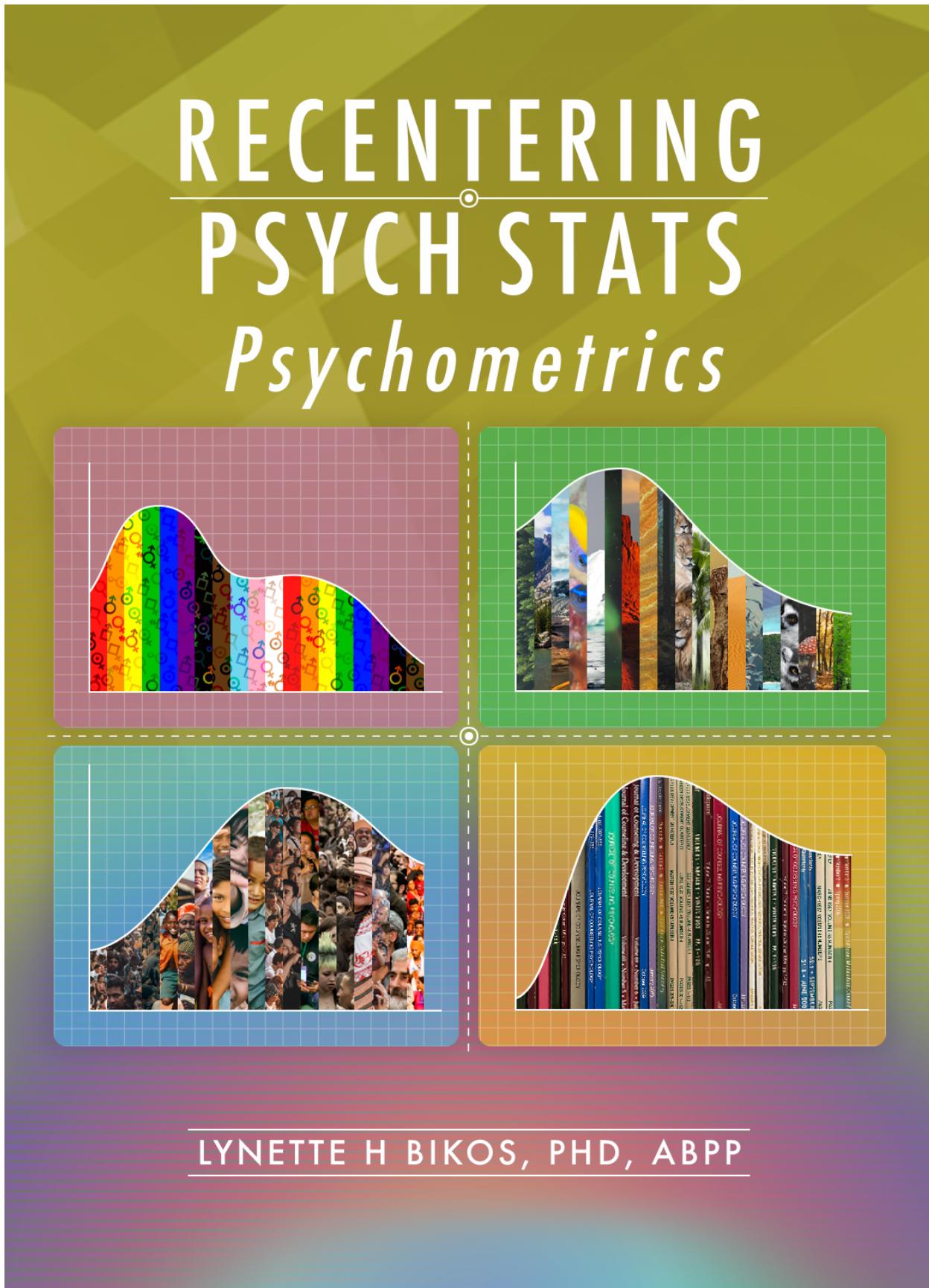


Figure 1: An image of the book cover. It includes four quadrants of non-normal distributions representing gender, race/ethnicity, sustainability/global concerns, and journal articles

Preface

If you are viewing this document, you should know that this is a book-in-progress. Early drafts are released for the purpose teaching my classes and gaining formative feedback from a host of stakeholders. The document was last updated on 16 Feb 2022. Emerging volumes on other statistics are posted on the [ReCentering Psych Stats](#) page at my research team's website.

[Screencasted Lecture Link](#)

To *center* a variable in regression means to set its value at zero and interpret all other values in relation to this reference point. Regarding race and gender, researchers often center male and White at zero. Further, it is typical that research vignettes in statistics textbooks are similarly seated in a White, Western (frequently U.S.), heteronormative, framework. The purpose of this project is to create a set of open educational resources (OER) appropriate for doctoral and post-doctoral training that contribute to a socially responsive pedagogy – that is, it contributes to justice, equity, diversity, and inclusion.

Statistics training in doctoral programs are frequently taught with fee-for-use programs (e.g., SPSS/AMOS, SAS, MPlus) that may not be readily available to the post-doctoral professional. In recent years, there has been an increase and improvement in R packages (e.g., *psych*, *lavaan*) used for in analyses common to psychological research. Correspondingly, many graduate programs are transitioning to statistics training in R (free and open source). This is a challenge for post-doctoral psychologists who were trained with other software. This OER will offer statistics training with R and be freely available (specifically in a GitHub repository and posted through GitHub Pages) under a Creative Commons Attribution - Non Commercial - Share Alike license [CC BY-NC-SA 4.0].

Training models for doctoral programs in HSP are commonly scholar-practitioner, scientist-practitioner, or clinical-scientist. An emerging model, the *scientist-practitioner-advocacy* training model incorporates social justice advocacy so that graduates are equipped to recognize and address the sociocultural context of oppression and unjust distribution of resources and opportunities (?). In statistics textbooks, the use of research vignettes engages the learner around a tangible scenario for identifying independent variables, dependent variables, covariates, and potential mechanisms of change. Many students recall examples in Field's (?) popular statistics text: Viagra to teach one-way ANOVA, beer goggles for two-way ANOVA, and bushtucker for repeated measures. What if the research vignettes were more socially responsive?

In this OER, research vignettes will be from recently published articles where:

- the author's identity is from a group where scholarship is historically marginalized (e.g., BIPOC, LGBTQ+, LMIC[low-middle income countries]),

- the research is responsive to issues of justice, equity, inclusion, diversity,
- the lesson's statistic is used in the article, and
- there is sufficient information in the article to simulate the data for the chapter example(s) and practice problem(s); or it is publicly available.

In training for multicultural competence, the saying, “A fish doesn’t know that it’s wet” is often used to convey the notion that we are often unaware of our own cultural characteristics. In recent months and years, there has been an increased awakening to the institutional and systemic racism that our systems are perpetuating. Queuing from the water metaphor, I am hopeful that a text that is recentered in the ways I have described can contribute to *changing the water* in higher education and in the profession of psychology.

Copyright with Open Access

This book is published under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. This means that this book can be reused, remixed, retained, revised and redistributed (including commercially) as long as appropriate credit is given to the authors. If you remix, or modify the original version of this open textbook, you must redistribute all versions of this open textbook under the same license - CC BY-SA.

A [GitHub open-source repository](#) contains all of the text and source code for the book, including data and images.

ACKNOWLEDGEMENTS

As a doctoral student at the University of Kansas (1992-2005), I learned that “a foreign language” was a graduation requirement. *Please note that as one who studies the intersections of global, vocational, and sustainable psychology, I regret that I do not have language skills beyond English.* This could have been met with credit from high school my rural, mid-Missouri high school did not offer such classes. This requirement would have typically been met with courses taken during an undergraduate program – but my non-teaching degree in the University of Missouri’s School of Education was exempt from this. The requirement could have also been met with a computer language (fortran, C++) – I did not have any of those either. There was a tiny footnote on my doctoral degree plan that indicated that a 2-credit course, “SPSS for Windows” would substitute for the language requirement. Given that it was taught by my one of my favorite professors, I readily signed up. As it turns out, Samuel B. Green, PhD, was using the course to draft chapters in the textbook (?) that has been so helpful for so many. Unfortunately, Drs. Green (1947 - 2018) and Salkind (2947 - 2017) are no longer with us. I have worn out numerous versions of their text. Another favorite text of mine was Dr. Barbara Byrne’s (?), “Structural Equation Modeling with AMOS.” I loved the way she worked through each problem and paired it with a published journal article, so that the user could see how the statistical evaluation fit within the larger project/article. I took my tea-stained text with me to a workshop she taught at APA and was proud of the signature she added to it (a little catfur might have fallen out). Dr. Byrne created SEM texts for a number of statistical programs (e.g., LISREL, EQS, MPlus). As I was learning R, I wrote Dr. Byrne, asking if she had an edition teaching SEM/CFA with R. She promptly wrote back, saying that she did not have the bandwidth to learn a new statistics package. We lost Dr. Byrne in December 2020. I am so grateful to these role models for their contributions to my statistical training. I am also grateful for the doctoral students who have taken my courses and are continuing to provide input for how to improve the materials.

The inspiration for training materials that re*center statistics and research methods came from the [Academics for Black Survival and Wellness Initiative](#). This project, co-founded by Della V. Mosley, Ph.D., and Pearis L. Bellamy, M.S., made clear the necessity and urgency for change in higher education and the profession of psychology.

At very practical levels, I am indebted to SPU’s Library, and more specifically, SPU’s Education, Technology, and Media Department. Assistant Dean for Instructional Design and Emerging Technologies, R. John Robertson, MSc, MCS, has offered unlimited consultation, support, and connection. Senior Instructional Designer in Graphics & Illustrations, Dominic Wilkinson, designed the logo and bookcover. Psychology and Scholarly Communications Librarian, Kristin Hoffman, MLIS, has provided consultation on topics ranging from OERS to citations. I am also indebted to Associate Vice President, Teaching and Learning at Kwantlen Polytechnic University, Rajiv Jhangiani, PhD. Dr. Jhangiani’s text (?) was the first OER I ever used and I was grateful for his

encouraging conversation.

Financial support for this project has been provided the following:

- *Call to Action on Equity, Inclusion, Diversity, Justice, and Social Responsivity Request for Proposals* grant from the Association of Psychology Postdoctoral and Internship Centers (2021-2022).
- *Diversity Seed Grant*, Office of Inclusive Excellence and Advisory Council for Diversity and Reconciliation (ACDR), Seattle Pacific University.
- *ETM Open Textbook & OER Development Funding*, Office of Education, Technology, & Media, Seattle Pacific University.

Chapter 1

Introduction

[Screencasted Lecture Link](#)

1.1 What to expect in each chapter

This textbook is intended as *applied*, in that a primary goal is to help the scientist-practitioner-advocate use a variety of statistics in research problems and *writing them up* for a program evaluation, dissertation, or journal article. In support of that goal, I try to provide just enough conceptual information so that the researcher can select the appropriate statistic (i.e., distinguishing between when ANOVA is appropriate and when regression is appropriate) and assign variables to their proper role (e.g., covariate, moderator, mediator).

This conceptual approach does include occasional, step-by-step, *hand-calculations* (only we calculate them arithmetically in R) to provide a *visceral feeling* of what is happening within the statistical algorithm that may be invisible to the researcher. Additionally, the conceptual review includes a review of the assumptions about the characteristics of the data and research design that are required for the statistic. Statistics can be daunting, so I have worked hard to establish a *workflow* through each analysis. When possible, I include a flowchart that is referenced frequently in each chapter and assists the researcher keep track of their place in the many steps and choices that accompany even the simplest of analyses.

As with many statistics texts, each chapter includes a *research vignette*. Somewhat unique to this resource is that the vignettes are selected from recently published articles. Each vignette is chosen with the intent to meet as many of the following criteria as possible:

- the statistic that is the focus of the chapter was properly used in the article,
- the author's identity is from a group where scholarship is historically marginalized (e.g., BIPOC, LGBTQ+, LMIC [low middle income countries]),
- the research has a justice, equity, inclusion, diversity, and social responsibility focus and will contribute positively to a social justice pedagogy, and
- the data is available in a repository or there is sufficient information in the article to simulate the data for the chapter example(s) and practice problem(s).

In each chapter we employ *R* packages that will efficiently calculate the statistic and the dashboard of metrics (e.g., effect sizes, confidence intervals) that are typically reported in psychological science.

1.2 Strategies for Accessing and Using this OER

There are a number of ways you can access this resource. You may wish to try several strategies and then select which works best for you. I demonstrate these in the screencast that accompanies this chapter.

1. Simply follow along in the .html formatted document that is available on via GitHug Pages, and then
 - open a fresh .rmd file of your own, copying (or retyping) the script and running it
2. Locate the original documents at the [GitHub repository](#). You can
 - open them to simply take note of the “behind the scenes” script
 - copy/download individual documents that are of interest to you
 - fork a copy of the entire project to your own GitHub site and further download it (in its entirety) to your personal workspace. The [GitHub Desktop app](#) makes this easy!
3. Listen to the accompanying lectures (I think sound best when the speed is 1.75). The lectures are being recorded in Panopto and should include the closed captioning.
4. Provide feedback to me! If you fork a copy to your own GitHub repository, you can
 - open up an editing tool and mark up the document with your edits,
 - start a discussion by leaving comments/questions, and then
 - sending them back to me by committing and saving. I get an e-mail notifying me of this action. I can then review (accepting or rejecting) them and, if a discussion is appropriate, reply back to you.

1.3 If You are New to R

R can be overwhelming. Jumping right into advanced statistics might not be the easiest way to start. However, in these chapters, I provide complete code for every step of the process, starting with uploading the data. To help explain what R script is doing, I sometimes write it in the chapter text; sometimes leave hashtags-comments in the chunks; and, particularly in the accompanying screencasted lectures, try to take time to narrate what the R script is doing.

I've found that, somewhere on the internet, there's almost always a solution to what I'm trying to do. I am frequently stuck and stumped and have spent hours searching the internet for even the tiniest of things. When you watch my videos, you may notice that in my R studio, there is a “scRiptuRe” file. I takes notes on the solutions and scripts here – using keywords that are meaningful to me so that when I need to repeat the task, I can hopefully search my own prior solutions and find a fix or a hint.

1.3.1 Base R

The base program is free and is available here: <https://www.r-project.org/>

Because R is already on my machine (and because the instructions are sufficient), I will not walk through the installation, but I will point out a few things.

- Follow the instructions for your operating system (Mac, Windows, Linux)
- The “cran” (I think “cranium”) is the *Comprehensive R Archive Network*. In order for R to run on your computer, you have to choose a location. Because proximity is somewhat related to processing speed, select one that is geographically “close to you.”
- You will see the results of this download on your desktop (or elsewhere if you chose to not have it appear there) but you won’t ever use R through this platform.

1.3.2 R Studio

R Studio is the desktop application I work in R. It’s a separate download. Choose the free, desktop, option that is appropriate for your operating system: <https://www.rstudio.com/products/RStudio/>

- Upper right window: Includes several tabs; we frequently monitor the
 - Environment: it lists the *objects* that are available to you (e.g., dataframes)
- Lower right window: has a number of helpful tabs.
 - Files: Displays the file structure in your computer’s environment. Make it a practice to (a) organize your work in small folders and (b) navigating to that small folder that is holding your project when you are working on it.
 - Packages: Lists the packages that have been installed. If you navigate to it, you can see if it is “on.” You can also access information about the package (e.g., available functions, examples of script used with the package) in this menu. This information opens in the Help window.
 - Viewer and Plots are helpful, later, when we can simultaneously look at our output and still work on our script.
- Primary window
 - R Studio runs in the background(in the console). Very occasionally, I can find useful troubleshooting information here.
 - More commonly, I open my R Markdown document so that it takes the whole screen and I work directly, right here.
- *R Markdown* is the way that many analysts write *script*, conduct analyses, and even write up results. These are saved as .rmd files.
 - In R Studio, open an R Markdown document through File/New File/R Markdown
 - Specify the details of your document (title, author, desired output)
 - In a separate step, SAVE this document [File/Save] into a NEW FILE FOLDER that will contain anything else you need for your project (e.g., the data).
 - *Packages* are at the heart of working in R. Installing and activating packages require writing script.

1.3.3 R Hygiene

Many initial problems in R can be solved with good R hygiene. Here are some suggestions for basic practices. It can be tempting to “skip this.” However, in the first few weeks of class, these are the solutions I am presenting to my students.

1.3.3.1 Everything is documented in the .rmd file

Although others do it differently, everything is in my .rmd file. That is, for uploading data and opening packages I write the code in my .rmd file. Why? Because when I read about what I did hours or years later, I have a permanent record of very critical things like (a) where my data is located, (b) what version I was using, and (c) what package was associated with the functions.

1.3.3.2 File organization

File organization is a critical key to this:

- Create a project file folder.
- Put the data file in it.
- Open an R Markdown file.
- Save it in the same file folder.
- When your data and .rmd files are in the same folder (not your desktop, but a shared folder), they can be connected.

1.3.3.3 Chunks

The R Markdown document is an incredible tool for integrating text, tables, and analyses. This entire OER is written in R Markdown. A central feature of this is “chunks.”

The easiest way to insert a chunk is to use the INSERT/R command at the top of this editor box. You can also insert a chunk with the keyboard shortcut: CTRL/ALT/i

“Chunks” start and end with those three tic marks and will show up in a shaded box, like this:

```
#hashtags let me write comments to remind myself what I did
#here I am simply demonstrating arithmetic (but I would normally be running code)
2021 - 1966
```

```
## [1] 55
```

Each chunk must open and close. If one or more of your tic marks get deleted, your chunk won’t be read as such and your script will not run. The only thing in the chunks should be script for running R; you can hashtag-out script so it won’t run.

Although unnecessary, you can add a brief title for the chunk in the opening row, after the “r.” These create something of a table of contents of all the chunks – making it easier to find what you did. You can access them in the “Chunks” tab at the bottom left of R Studio. If you wish to knit a document, you cannot have identical chunk titles.

You can put almost anything you want in the space outside of tics. Syntax for simple formatting in the text areas (e.g., using italics, making headings, bold, etc.) is found here: https://rmarkdown.rstudio.com/authoring_basics.html

1.3.3.4 Packages

As scientist-practitioners (and not coders), we will rely on *packages* to do our work for us. At first you may feel overwhelmed about the large number of packages that are available. Soon, though, you will become accustomed to the ones most applicable to our work (e.g., psych, tidyverse, lavaan, apaTables).

Researchers treat packages differently. In these lectures, I list all the packages we will use in an opening chunk that asks R to check to see if the package is installed, and if not, installs it.

```
if(!require(psych)){install.packages("psych")}
```

```
## Loading required package: psych
```

To make a package operable, you need to open it through the library. This process must be repeated each time you restart R. I don't open the package (through the "library(package_name)") command until it is time to use it. Especially for new users, I think it's important to connect the functions with the specific packages.

```
#install.packages ("psych")
library (psych)
```

If you type in your own "install.packages" code, hashtag it out once it's been installed. It is problematic to continue to re-run this code .

1.3.3.5 Knitting

An incredible feature of R Markdown is its capacity to *knit* to HTML, powerpoint, or word. If you access the .rmd files for this OER, you can use annotate or revise them to suit your purposes. If you redistribute them, though, please honor the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License with a citation.

1.3.4 tRoubleshooting in R maRkdown

Hiccups are normal. Here are some ideas that I have found useful in getting unstuck.

- In an R script, you must have everything in order – Every. Single. Time.
 - All the packages have to be in your library and activated; if you restart R, you need to reload each package.
 - If you open an .rmd file and want a boxplot, you cannot just scroll down to that script. You need to run any *prerequisite* script (like loading the package, importing data, putting the data in the global environment, etc.)
 - Do you feel lost? clear your global environment (broom) and start at the top of the R script. Frequent, fresh starts are good.

- Your .rmd file and your data need to be stored in the same file folder. These should be separate for separate projects, no matter how small.
- Type any warnings you get into a search engine. Odds are, you'll get some decent hints in a manner of seconds. Especially at first, these are common errors:
 - The package isn't loaded (if you restarted R, you need to reload your packages)
 - The .rmd file has been saved yet, or isn't saved in the same folder as the data
 - Errors of punctuation or spelling
- Restart R (it's quick – not like restarting your computer)
- If you receive an error indicating that a function isn't working or recognized, and you have loaded the package, type the name of the package in front of the function with two colons (e.g., psych::describe(df)). If multiple packages are loaded with functions that have the same name, R can get confused.

1.3.5 stRategies for success

- Engage with R, but don't let it overwhelm you.
 - The *mechanical is also the conceptual*. Especially when it is *simpler*, do try to retype the script into your own .rmd file and run it. Track down the errors you are making and fix them.
 - If this stresses you out, move to simply copying the code into the .rmd file and running it. If you continue to have errors, you may have violated one of the best practices above (Is the package loaded? Are the data and .rmd files in the same place? Is all the prerequisite script run?).
 - Still overwhelmed? Keep moving forward by downloading a copy of the .rmd file that accompanies any given chapter and just “run it along” with the lecture. Spend your mental power trying to understand what each piece does. Then select a practice problem that is appropriate for your next level of growth.
- Copy script that works elsewhere and replace it with your datafile, variables, etc.
- The learning curve is steep, but not impossible. Gladwell(?) reminds us that it takes about 10,000 hours to get GREAT at something (2,000 to get reasonably competent). Practice. Practice. Practice.
- Updates to R, R Studio, and the packages are NECESSARY, but can also be problematic. It could very well be that updates cause programs/script to fail (e.g., “X has been deprecated for version X.XX”). Moreover, this very well could have happened between my distribution of these resources and your attempt to use it. My personal practice is to update R, R Studio, and the packages a week or two before each academic term.
- Embrace your downward dog. Also, walk away, then come back.

1.3.6 Resources for getting staRted

R for Data Science: <https://r4ds.had.co.nz/>

R Cookbook: <http://shop.oreilly.com/product/9780596809164.do>

R Markdown homepage with tutorials: <https://rmarkdown.rstudio.com/index.html>

R has cheatsheets for everything, here's one for R Markdown: <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

R Markdown Reference guide: <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>

Using R Markdown for writing reproducible scientific papers: <https://libscie.github.io/rmarkdown-workshop/handout.html>

LaTeX equation editor: <https://www.codecogs.com/latex/eqneditor.php>

Chapter 2

Questionnaire Construction: The Fundamentals

[Screencasted Lecture Link](#)

I found this lesson to be more of a struggle to prepare than I expected. Why? There is a great deal of lore about what increases response rates and participation. Yet, research over the years had both supported and contradicted these claims. One example is where to include “sensitive items.” Historically, textbook authors have recommended that these should come last so that respondents would be engaged in the process and be more willing to complete the survey (??). Yet, research has shown that this has not held up in employee groups (?) nor among members of the National Association of Social Workers (?).

Given these contradictions, this lecture starts with the overall structure of a survey. The core of the lecture focuses on recent, evidence-based support for item-level decisions. I briefly discuss construct-specific guidance and discuss specific considerations for the on-line environment. I then close by addressing some of the decisions that I routinely make in survey construction and provide my rationale for why. Because this lesson occurs at the beginning of a text on psychometrics – this “skips over and around” the issue of reliability and validity.

2.1 Navigating this Lesson

There is just under one hour of lecture.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER’s [introduction](#)

2.1.1 Learning Objectives

Focusing on this week’s materials, make sure you can:

- Outline the overall structure/components of a questionnaire,
- Articulate test construction myths (e.g., location of sensitive items, “requirement” to have reverse scored items) and their evidence-based solutions (when they have them)
- List elements to consider when the questionnaire is administered online

2.1.2 Planning for Practice

This is a two-part lesson on questionnaire construction. After the second lesson, a detailed suggestion for practice will be provided that lists criteria for creating and piloting a survey of your own.

2.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (when possible, linked) in the text with complete citations in the reference list.

- Chyung, S. Y., Roberts, K., Swanson, I., & Hankinson, A. (2017). Evidence-Based Survey Design: The Use of a Midpoint on the Likert Scale. *Performance Improvement*, 56(10), 15–23. <https://doi.org/10.1002/pfi.21727>
- Chyung, S. Y., Barkin, J. R., & Shamsy, J. A. (2018a). Evidence-Based Survey Design: The Use of Negatively Worded Items in Surveys. *Performance Improvement*, 57(3), 16–25. <https://doi.org/10.1002/pfi.21749>
- Chung, S. Y., Kennedy, M., & Campbell, I (2018b). Evidence-based survey design: The use of ascending or descending order of Likert-type response options. *Performance Improvement*, 57(9), 9-16. <https://doi.org/10.1002/pfi.21800>
- Chyung, S. Y., Swanson, I., Roberts, K., & Hankinson, A. (2018c). Evidence-Based Survey Design: The Use of Continuous Rating Scales in Surveys. *Performance Improvement*, 57(5), 38–48. <https://doi.org/10.1002/pfi.21763>
 - Finding the Chyung et al. series was like finding a pot of gold! They provide empirical support for guiding choices about survey construction. And they are current! If you don’t have time to read them in detail, I recommend you scan them and archive them for future reference.

2.2 Components of the Questionnaire

Let’s start by examining the components of a questionnaire and the general guidelines for their construction(??):

Title

- reflect the content of the instrument
- be concisely worded
- be written in language easily understood by the respondents
- should not be offensive or off-putting
- should be formatted clearly at the top/beginning of the document

Introductory Statement

- include a brief summary of the instrument's purpose
- contain an appropriate statement concerning the confidentiality of the respondent's information (informed consent)
- be motivating such that respondents are inspired/willing to complete the items
- specify the approximate amount of time required to complete the instrument

Directions

- complete, unambiguous, concise
- written at a language level appropriate to the respondents
- tell the respondents how to return the instrument once they have completed it (surprisingly, in Qualtrics, this is also important; submission requires hitting that last little “→”)

Items (discussed in a later section)

Closing Statement

- thank the participants for their participation
- remind participants that their information is valuable and perhaps remind about
 - next steps or follow-up
 - confidentiality

Overall Structure/Look

- should be coherent with an easy-to-follow layout
- not crowded
- professional appearance
 - not crowded, plenty of white space
 - avoiding a “slick look”
 - numbering and headings to provide a sense of progress
 - breaks between every 4-6 questions (or shading alternate items)
 - in a sense, inviting and “easy on the eye”

Pershing and Pershing (?) reviewed 50 *reactionnaires* that were used by training evaluators at a “prestigious medical school.” Their purpose was to determine the degree to which the survey design adhered to the recommendations. The results suggested that:

- 72% did not include an introductory statement; an additional 16% were “minimal”
- 78% had no closing statement
- 30% had no directions; another 54% of directions were “minimal”
- 8% were professional in appearance

In summary, the formatting of the reactionnaires were not designed in a way that would maximize respondent engagement. In turn, we might expect this to threaten the psychometric reliability and validity.

2.3 What Improves (or Threatens) Response Rates and Bias?

It's not always clear, but well-designed studies and formal reviews of literature are revealing, sometimes surprising findings to old debates. When we design survey instruments based on our own preference rather than research-based evidence, we may get less than optimal data (?).

Chyung et al. (2018) reviewed the five steps (?) that survey respondents engage when answering structured, closed-ended survey items.

1. Interpreting the question
2. Retrieving information from their memory
3. Integrating the information
4. Selecting one of the given response options
5. Editing the answer for reasons of social desirability

Chyung and colleagues appear to be starting such a systematic review. What follows are their evidence based evaluations regarding some of the most common questions about questionnaire construction.

2.3.1 Should Likert-type scales include a midpoint?

Chyung et al. (?) reviewed the literature on whether/not to use a **midpoint** ("neutral" or "neither disagree nor agree"). Examining their article, we can see variants of Likert-style scaling for a scale of agreement. They look something like this:

Type						
No midpoint (4 pt)	Strongly Disagree	Disagree	<i>skipped</i>	Agree	Strongly Agree	
Midpoint (5 pt)	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree	

Chyung and colleagues quickly suggest that the question is not "Should I use a midpoint?" but rather "When should I use a midpoint?"

The article is more detailed, but essentially, a midpoint is appropriate when:

- the measurement scale is interval (instead of ordinal; this is a good statistical property to have)
- the question content is such that the midpoint is a *true* midpoint and not a point for hedging or avoiding

If a true midpoint is impossible, then consider adding an option such as "I don't know" or "It depends." Of course these introduce scoring dilemmas (e.g., needing to be recoded as missing; then having a plan for managing missingness) that will need to be addressed in preparing the data for analysis.

2.3.2 Should *continuous rating scales* be used in surveys?

First, let's consider the distinction between *discrete*, *continuous*, and *numerical* scales. Figure 4 in the Chyung, Swanson, Roberts, and Hankinson (?) article illustrate the major differences and some variations.

- **Discrete** scales are Likert-type scales that range between 2 and 11 *discrete* options. Clasically, respondents pick *words* (e.g., pain rated as *no pain, mild, moderate, severe, extreme, worst pain possible*).
 - 6-point discrete rating scales result in a collection of six *ordered values*
 - thus they are on the ordinal measurement scale and (as discussed above)
 - they should be analyzed with non-parametric statistical procedures (parametric approaches can be used if the data are normally distributed and there is a mid-point)
- **Continuous** scales allow respondents to indicate a response anywhere within a given range – usually by marking a place on a horizontal line on a continuum of a minimum of 100 points. There are no discrete categories defined by words or numbers. In Qualtrics there is a slider option.
 - Continuous scales result in precise numbers (e.g., 26 or 26.8 if the scale is 0 to 100)
 - these are on an interval-level measurement scale AND
 - they can be evaluated with parametric statistical analyses
 - *visual analog scales (VAS; aka graphic rating scales, GRS)* are another variant of continuous rating scales if they allow the participants to make “anywhere on the line.” Some VAS scales have verbal descriptors to guide the marking; some have numbers (hence, *numerical response scales*)

Which is better? The mixed results are summarized in Chyung et al's (?) Table 1. With a focus on the types of research I encounter in my program, here is my take-away:

- Continuous scales provide better data (more precise/full information, more likely to be normally distributed, better reliability) for statistical analysis.
 - *Caveat:* If the response scale on a Likert scale is increased to 11, there is a better chance to have normally distributed responses.
 - *Caveat:* When “simple descriptives” are desired (e.g., histograms, frequency distributions) the discrete scale may be the best choice.
- Both are easy to use, except in the case where respondents complete the surveys on mobile devices.
 - *Caveat:* there has been more missing data with sliders (compared to radio buttons)
 - *Caveat:* respondents are more likely to change their responses on sliders. If this means there is greater accuracy or more careful responding, this is desirable.
- In both circumstances adding “don’t know,” “prefer not to respond,” or “not applicable” may improve the validity of the responses

2.3.3 Should Likert-type response options use an ascending or descending order?

Let's first look at the difference (?):

Type						
Ascending	Strongly Disagree	Disagree	Neither Disagree Nor Agree	Agree	Strongly Agree	
Descending	Strongly Agree	Agree	Neither Agree Nor Disagree	Disagree	Strongly Disagree	

In the consideration of the choice between ascending/descending, we are concerned with *response-order effects*. Let's first examine these conceptually/theoretically.

Recency effect is the tendency of survey respondents to select the options that they see at the end of the response-option list. This is expected when options are presented orally (e.g., during interviews, people tend to choose from the last-offered options)

Primacy effect is the survey respondents' tendency to select the options that are presented at the beginning of the response-option list. This is expected when options are presented visually. For example, people tend to choose among the first-presented categories in self-administered written survey questionnaires.

- *left-sided selection bias* occurs when respondents read text from left-to-right and are more inclined to select options from the left.
- *satisficing theory* occurs when individuals seek solutions that are “simply satisfactory” so as to minimize psychological costs. Thus, respondents may
 - select the first option that seems “reasonable enough”,
 - select the “I don’t know” response, or
 - randomly select one of the options.
- *acquiescence bias* is the tendency for respondents to agree with the statement provided—aka yea-saying bias (e.g., being polite).
 - Closely related is *social-desirability bias*, the tendency for respondents to select among the options they think are more socially acceptable or desirable (instead of true responses).
 - In surveys, this generally is selecting *agree* or *strongly agree*.

Considering these response biases together, Chyung et al. suggest that when the response options are presented in descending order (*Strongly agree, Agree, Neutral, Disagree, Strongly disagree*), respondents would (theoretically) see a positive option immediately on the left side of the response scale and perceive it to be socially desirable and satisfactory, resulting in their decision to select it without having to spend more time to choose a true response. However, the same effects may or may not happen when the response options are presented in ascending order (*Strongly disagree, Disagree, Neutral, Agree, Strongly agree*).

After reviewing 13 studies, Chyung et al. observed that many studies (paper and web based, with children and adults, in English and other language):

- revealed response-order effects in self administered surveys, especially the primacy effect, associated with left-side selection bias, acquiescence bias, and satisficing.
- showed more positive average scores from descending-ordered scales

Recommendations:

- present response scales in ascending order
 - when a number line is used, lower and negative numbers should be on the left
- when using descended order scales
 - keep respondents motivated to complete items accurately
 - present half items with descended-ordered scales and the other half with ascended-ordered scales
 - assign half of participants with descended-ordered scales; half with ascended-ordered scales
 - present response options vertically rather than horizontally

2.3.4 Should surveys include negatively worded items?

In examining this question, Chyung et al. (?) made a distinction between (see Table 1 in the article):

- **Statement format** the same response scale such as a Likert scale
- **Question format** different response scales that are tailored to individual survey questions. A challenge with this format is the difficulty in calculating an average score of data obtained from multiple survey items.

The advent of negatively-worded items began with Rensis Likert in 1932. He was an American social psychologist who, in attempt to mitigate acquiescence/yea-saying biases, recommended designing one half of survey items to be associated with agreement and the other half with disagreement. Although Likert recommended “straightforward statements,” incorporating negative words can become quickly complicated. Table 2 in the Chyung paper shows that there are four ways of wording survey statements:

Reverse-coding, which is necessary when including negatively worded items in a scale, assumes that agreeing to a positively worded statement and disagreeing to its negatively worded counterpart are the same. Tables 3 and 4 in the ms show how this assumption may be faulty.

A review of the negatively-worded-item literature suggested the following:

- scales with all positively worded items yielded greater accuracy when compared with all negatively worded items or mixed worded items

- scores on positively and negatively worded items are not the same (e.g., strongly disagreeing to a positively worded statement is different from strongly agreeing to a negatively worded statement)
- positively worded items produce higher means than negatively worded items. This may be due to
 - carelessness and fatigue in reading items
 - the cognitive processing of positive and negative items may be different
- a *method factor* has shown itself where in factor analyses of scales with factor structures unrelated to the wording, exploratory approaches to factor analysis have produced separate factors with the negatively worded (or otherwise ambiguous) items, this results in a threat to construct validity and reliability.

Chyung, Barkin, and Ramsey (?) noted that respondent performance declines approximately 12 minutes after respondents start a survey. This relates to negatively worded items because it appeared that respondents increasingly failed to notice negatively worded statements even when there were efforts to draw their attention to them via bolding, underlining, or capitalizing the negated element (e.g., *not*). Thus, when negatively worded items are used, they should probably be presented early in the protocol.

Chyung et al (?) also cautioned about a response set bias that can occur when using all positively worded items. They recommended making design choices that enhance bias-free and accurate responding based on the research design.

- For example, attributes to be measured in some constructs (e.g., depression, anxiety) are, themselves, negative and so a negatively worded item may be most clear and appropriate.
- The inclusion (and subsequent analysis) of negatively phrased items may help *detect* acquiescence bias.
- Table 5 in the manuscript provides some guidelines that are more nuanced when negative items must be included. For example,
 - ensure that negatively worded items are true polar opposites and symmetrical (so they can be analyzed with the positively worded items)
 - group negative items together (and forewarn/format so they are recognized as such)
 - administer the survey when respondents are not fatigued
 - analyze the effect of the negatively worded items

2.4 Construct-specific guidance

Self-efficacy is domain-specific construct. That is, even though there are some *general self-efficacy scales* Bandura's original definition suggests that it should be task specific (i.e., career decision-making self-efficacy, math self-efficacy).

This construct is an example where Bandura, himself (?), provided specific guidelines for creating these task-specific assessments. Generally they included:

1. phrasing items as “can do” rather than “will do,”

2. maintaining consistency with the self-efficacy construct definition (e.g., domain specific, a focus on capability rather than self-worth),
3. including items that reflect gradations of challenge,
4. asking individuals to rate their current (as opposed to future) operative capabilities
5. scaling should be on a 100 point scale

2.5 Surveying in the Online Environment

Nearly a decade ago, a survey of human subjects review boards suggested that 94% of the IRB applications reviewed involved online or Web-based surveys (?). Thus, it is important to give some specifical conceptual consideration to the online environment.

A first set of considerations involve data security, identity, and permission (implicit and explicit).

The **IP address** as well as **longitude/latitude** has been a contentious issue for a number of years (?). EU data protection laws consider IP addresses as personally identifiable data; in the U.S., IP addresses typically fall outside the definition of “personal information.” In Qualtrics, the default is to collect and download the IP address (the “anonymize response” option can prevent this data from being collected). On the one hand it is helpful to know geographically “from where” participants are responding; on the other, some consider its capture to be a violation of privacy.

Relatedly, what is considered to be **fully informed consent** (?). Is it ethical to capture paradata (e.g., typing speed, changed answers, response times) or metadata without explicitly saying so?

The tool being used to collect the data is concerning. Buchanan and Hvizzak (?) argued that until each tool is vetted and its privacy policies and data security policies are understood, we cannot be 100% certain how security, consent, and privacy are instantiated within the individual tools. For example, it is possible that tool creators *could* gather respondent data and repurpose it for their own marketing, for sale to other researchers, and so forth

Online and web-based protocols increase our reach geographically and cross-culturally. For now, I will bracket out the issue of cultural translation, for the purpose of online the question is about **access** (?). Think about the decades of psychological research based on White, college-educated, males. Are we creating another strata of privileged research with technology that may not be accessible in terms of both internet/technology as well as capacity/fluency with the tool? On the other hand, what are the risks of not adopting new technologies before everyone has them.

When paper/pencil measures were administered in face-to-face settings (individually or in auditoriums of students) there was some degree of **standardized protocol**. This is lost when surveys are administered online and we cannot guarantee *who* is taking the survey. To what degree does this matter?

When respondents are remote, what happens if they have a **negative reaction to the survey**? In a face-to-face context, debriefings can occur and referrals can be made. How is this managed remotely?

Security of test items might also be concerning. It is not ok to use proprietary items without the permission of the testmaker. If the security of items is important (e.g., SAT/GRE, intelligence test items, inkblots) because they are central to administration, how can this be protected in the virtual environment?

Consequently, when students in our programs write doctoral dissertations they are to narrate the following in their Method section.

- Describe how informed consent will be obtained in the online environment.
- Describe the level of identification that is collected. If the claim of “anonymous” or “de-identified” indicate whether/not this includes capturing the IP address; some researchers believe that capturing a computer’s IP address threatens anonymity.
- Describe the steps to be taken to ensure that respondents met the inclusion/exclusion criteria of the study.
- Anticipate and describe how the online (e.g., uncontrolled, public, distractions) setting might affect responses.
- Particularly if the survey contained sensitive materials, describe how respondents might access resources for debriefing or referral.
- Identify the permissions (from original authors or copyright holders) granted to reformat and post (on the internet) existing surveys. If items are considered to be secure (e.g., those on the MMPI or WAIS), identify steps taken to protect them.

2.6 In my Surveys

Because there isn’t empirical data on every decision that we make in survey construction, I thought it might be useful for me to address some of the decisions that I find myself making in the online surveys I use in my own research.

2.6.1 Demographics and Background Information

A core value that I hope to reflect in the *ReCentering Psych Stats* series is to promote socially and culturally responsive research. Correspondingly, the information we collect from participants should ensure that they feel that their identities are authentically reflected in the survey. Naively, when I first considered how to capture race/ethnicity in my surveys, I looked to the categories used in the U.S. Census. Immediately, I learned that this is problematic. Rather than elaborating here, I invite you to take a listen to NPR’s [Code Switch](#) podcast had a series of episodes that included a discussion of the evolution of the race and ethnicity question. Two of the episodes review how the race/ethnicity question is used and why it is problematic: [Census Watch 2020](#) and [The U.S. Census and Our Sense of Us](#). As made clear in the Code Switch podcasts, the race/ethnicity question in the U.S. Census *erases* people when their identities are not included.

My last few surveys have captured race/ethnicity data differently. Each time, I engage in several practices that (I hope) will continue to shape the item in a socially and culturally responsive way. Systematically, I:

- conduct a quick internet search to see if there is an emerging best practice (even though I may have also searched weeks or months prior),
- consider who the intended research population is in relationship to the topic of investigation,
- look to recently published, similar, research to see what other researchers are doing, and
- ask for a collegial, formative review from individuals who hold marginalized identities, whose data will be requested in the survey.

When I engage in research, I try to balance the need to quantify (with discrete categories) who is participating in the survey and inviting respondents to state (in their own words) their identity. This is consistent with my view that variables like race, ethnicity, and gender identity are socially constructed. In addition to this particular worldview, Parent (?) has suggested that the worst possible kind of missing data pattern (MNAR – missing not at random) may be caused when items are *unanswerable* to particular person. Therefore, it is essential that all individuals find their place in the items that assess demographic variables.

My last survey was directed toward community members (including students, alumni, staff, faculty) of my predominantly White, liberal arts, Christian institution. After reviewing recently published research articles and consulting with a handful of individuals, I chose to include the following categories – *each with a text write-in box* so that individuals could select the category(ies) that fit best and have the opportunity to refine it(them). I am excited to review this data because such responses may inform my next survey. The categories included:

- Asian or Pacific Islander
- Black or African American
- Hispanic or Latino
- Native American or Alaskan Native
- White or Caucasian
- Biracial or multiracial
- An international/global identity that does not fit in the U.S. categorization of race/ethnicity
- A race/ethnicity not listed above

Note that individuals could select multiple categories

RaceEthnicity ★

Which of the following best describes you?

Owing to the limitations of surveys, we have provided categories typically used in survey construction. Please select the category(ies) that is(are) closest to the manner in which you identify. You are welcome to refine the response in the text box.

Asian or Pacific Islander
 Black or African American
 Hispanic or Latino
 Native American or Alaskan Native
 White or Caucasian
 Biracial or multiracial
 An international/global identity that does not fit in the U.S. categorization of race/ethnicity
 A race/ethnicity not listed above

Figure 2.1: Image of a survey item inquiring about race/ethnicity from a survey. Each option has an option for the respondent to clarify.

The option to select multiple boxes will result in some extra coding when we prepare the data for analysis. I am taking approach that we will *listen* to the data and decide, based on the results, how to report the findings in a way that will efficiently fit into an APA style empirical paper and honor the participants.

The population of interest for this particular study are those who are engaged in protest activities regarding hiring practices and policies that result in discrimination to members of the LGBTQIA+ community. This means that questions of gender identity, pronouns, and relationship to the LGBTQIA+ community are important to the research, but needed to be asked sensitively and with great security of the data.

Regarding gender identity, I used a similar approach to the race/ethnicity question, allowing individuals to select multiple categories and offering write-in boxes for each. The categories included:

- female
- male
- nonbinary
- transwoman
- transman
- something else
- prefer not to say

Additionally, I invited individuals to identify their pronouns. Again, write-in boxes were offered with each option.

- they/them/theirs
- she/her/hers
- he/him/his
- they/she/he
- neo pronouns (e.g., xe/xem/xyr, ze/hir/hirs, ey/em/eir)
- something else

Finally, we wanted individuals to indicate their relationship to the LGBTQIA+ community. We asked them to select all that apply. Only the “something else” box had a write-in option:

- Member
- Exploring/Questioning
- Ally
- Not related
- Something else (with a write-in box)

I expect that my future surveys may inquire about these variables differently. If you have found a different way to ask, please consider e-mailing me. I would love to provide different options and give credit to contributors.

2.6.2 Survey Order

Historically, demographic information has been first or last in the survey. Although some research has reported no differences in response rates when demographic and sensitive data are at the beginning or end (??), I am inclined to open the survey with questionnaire items that are closely related to the topic listed on the recruitment materials and end the survey with the demographic

information. Why? It makes sense to me that if someone has responded positively to the survey topic, they expect to answer questions (right away) about that topic.

In between that opening survey and closing demographic items, I consider if there are any *order effects* that would engage in undue *priming* of responses. If there are such concerns, I think through the order to minimize them. If there are no such concerns, I put my surveys in blocks and then ask my survey program to randomly present the blocks. This serves two purposes:

- counterbalancing possible order effects, and
- distributing missingness for individuals who do not complete the survey.

2.6.3 Forced Responses

Programs like Qualtrics are able to engage in a variety of *content validation* procedures. If these are in place, they may require the person to enter a properly formatted response (e.g., phone number, e-mail address, numerical response between 0 and 100) before responding. These are extremely helpful tools in collecting data that will be closest-to-being-ready-for-analysis. These same procedures can *force* or *request* a response.

Requiring a response is tempting. However, doing so violate IRB requirements that allow a person to skip or “quit at any time without penalty.” They may also anger a person such that they stop responding. Some researchers get around this by *requiring* the response, but including a “Not applicable” or “Prefer to not answer” column. Because I worry that (a) the respondent may confuse that option with one extreme of the scale and/or (b) my research team and I will forget to code it as missing data, I prefer the *request* response alternative.

In Qualtrics in particular, I turn on the “Request response” feature for each of the questions. If an item is skipped, a simple warning is displayed that invites the respondent to review the page of answers to see if they would like to answer the question. If not, they can simply move forward.

2.7 Practice Problems

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. With each of these options I encourage you to:

This is a two-part lesson on questionnaire construction. After the [second lesson](#), a detailed suggestion for practice will be provided that lists criteria for creating and piloting a survey of your own.

Chapter 3

Be a QualTRIXter

[Screencasted Lecture Link](#)

The focus of this lecture is on the technical and mechanical tools available in Qualtrics (and likely other survey platforms) to increase the effectiveness of your survey.

3.1 Navigating this Lesson

This lecture is just under one hour. Plan for another 30 minutes for *intRavenous qualtRics* practice.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

3.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Utilize basic Qualtrics tools (e.g., question type, use of headers) so that surveys are present materials clearly to the respondent.
- Incorporate more advanced tools (e.g., display logic, randomization) that may increase the respondent's ability to complete the survey and provide accurate responses.
- Provide a rationale for survey options that protect (or possibly reveal) an individual's identity.

3.1.2 Planning for Practice

This is the second of a two-part lesson on questionnaire construction. At the end of this lesson is a detailed suggestion for practice that lists criteria for creating and piloting a survey of your own. There are four essential criteria for your survey:

- Adhere to the evidence-based practices identified in the lesson on [questionnaire construction](#).

- Utilize four techniques (in the context of Qualtrics, I term these *qualTRIXter skills*) that increase the flow, effectiveness, and appearance of your survey.
- Pilot and consider feedback provided by those who took the survey.
- Import the data into the R environment.

3.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the tutorials available at the [Qualtrics support site](#). I have tried to link them throughout the presentation. It is likely they could change at any time and/or they might not work on your particular browser (as I write this, half of them will not work on FireFox, but they do on Chrome and Edge).

3.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(qualtRics)){install.packages("qualtRics")}
```

3.2 Research Vignette

I will demonstrate the qual”TRIX” by using a Qualtrics account at my institution, Seattle Pacific University. The only surveys in this account are for the *Recentering Psych Stats* chapters and lessons. All surveys are designed to not capture personally identifying information and not collecting IP addresses nor longitude/latitude. I use this survey in several lessons in this OER. If you haven’t taken the survey yet, [I invite you to do so, now](#).

As a teaching activity for the ReCentering Psych Stats OER, the topic of the survey was selected to be consistent with the overall theme of OER. Specifically, the purpose of this study is to understand the campus climate for students whose identities make them vulnerable to bias and discrimination. These include students who are Black, non-Black students of color, LGBTQ+ students, international students, and students with disabilities.

After consulting with a diverse group of stakeholders and subject matter experts (and revising the response options numerous times) I have attempted to center anti-Black racism in the U.S. (??). In fact, the display logic does not present the race items when the course is offered outside the U.S. There are only five options for race: *biracial/multiracial*, *Black*, *non-Black person(s) of color*, *White*, and *I did not notice* (intended to capture a color-blind response). One unintended negative consequence of this design is that the response options could contribute to *colorism* (??). Another possibility is that the limited options may erase, or make invisible, other identities. At the time that I wrote up the first description of this survey, the murder of six Asian American women in Atlanta had just occurred. The Center for the Study of Hate and Extremeism has documented that while overall hate crimes dropped by 7% in 2020, anti-Asian hate crimes reported to the police in America’s largest cities increased by 149% (?). These incidents have occurred not only in cities, but

in our neighborhoods and on our campus (???). While this survey is intended to assess campus climate as a function of race, it unfortunately does not distinguish between many identities that experience marginalization.

Although the dataset should provide the opportunity to test a number of statistical models, one working hypothesis that framed the study is that there will be a greater sense of belonging and less bias and discrimination when there is similar representation (of identities that are often marginalized) in the instructional faculty and student body. Termed, “structural diversity” (?) this is likely an oversimplification. In fact, an increase in diverse representation without attention to interacting factors can increase hostility on campus (?). Thus, the task of rating of a single course relates to the larger campus along the dimensions of belonging and bias/discrimination. For example, if a single class has higher ratings on issues of inclusivity, diversity, and respect, we would expect that sentiment to be echoed in the broader institution.

The survey design has notable limitations. You will likely notice that we ask about demographic characteristics of the instructional staff and classmates in the course rated, but we do not ask about the demographic characteristics of the respondent. In making this decision, we likely lose important information. For example, Iacobino and James (?) have noted that White students perceive campus more favorably than Black student counterparts.

The decision to not collect demographic details about the respondent was about protecting their (your) identity. As you will see, you have the opportunity to download and analyze the data. If a faculty member asked an entire class to take the survey, the datestamp and a handful of demographic identifiers could very likely identify a student. In certain circumstances, this might be risky in that private information (i.e., gender nonconformity, disclosure of a disability) along with course evaluation data and a datestamp could be related back to the student.

Further, the items that ask respondents to *guess* the identities of the instructional staff and classmates are limited, and contrary to best practices in survey construction that recommend providing the option of a “write-in” a response.

In parallel, the items asking respondents to identify characteristics of the instructional staff along dimensions of gender, international status, and disability are “large buckets” and do not include “write-in” options. Similarly, there was no intent to cause harm by erasing or making invisible individuals whose identities are better defined by different descriptors. Further, no write-in items were allowed. This was also intentional to prevent potential harm caused by people who could leave inappropriate, racist, or otherwise harmful comments.

As I review Qualtrics essentials and trix, I will their use (if used) in the ReCentering Psych Stats survey.

3.3 Qualtrics Essentials

Qualtrics is a powerful program and I find that many of the surveys we distribute don’t capitalize on the features Qualtrics has to offer. Qualtrics has detailed tutorials and instructions that are well worth the investment of a weekend to review them.

In this lecture I will point you to the elements that I think are critical to constructing online surveys. Because Qualtrics tutorials are (a) clear and thorough and (b) frequently updated, I will (a) point you to the tutorials that are available at the time of this lecture prep, (b) tell you why I think they are appropriate, and (c) show you how we have used them in some of our own surveys.

Even if you think you know what you are doing, start here (and then always take the time to “look around” at all the options on each window):

Survey Basic Overview: The link below will give you an overview. From there, you can follow all kinds of leads, looking for things you want to do with your survey – and getting ideas for what will improve it. <https://www.qualtrics.com/support/survey-platform/survey-module/survey-module-overview/>

Blocks are the basic organizational tool in Qualtrics surveys. Blocks have two purposes: (a) grouping items shown on “one page,” and (b) using the block for easy ordering and/or random selection/presentation.

Question types: Take a time to look at ALL the options. You might be surprised to learn that there is a better choice than you might have imagined.

Let's take a look at super basic/helpful question types:

- **Text/graphic:** These are the types you should use for providing information (e.g., informed consent) to the participants.
- **Matrix table :** A more efficient way to use the Likert-style items (than multiple choice). There is some controversy about whether not to use matrix tables vs. multiple choice dropdowns...
 - Make sure to select a reasonable amount of header repetition. This allows the respondent the maximum opportunity to see the column descriptors while they are responding.
- **Slider :** This gets you continuous data on that 1 to 100 scale. If the scale you are using is already published, and has not been psychometrically evaluated for slider use, you should probably stick with the format recommended in the publication. But if YOU are designing a survey, think about this option.
- **Text Entry Questions** have multiple options for length of answer. Don't miss the options that include forms and content validation
- **Validation:** Allows the user to allow certain types of information and specify their formats (e.g., numbers, e-mail addresses, dates). There is a balancing between being overly restricting and ensuring that the data is entered in the most clear and consistent way possible. A validation option I frequently use is one that asks individuals if they intended to leave something blank. This is tool that helps prevent missingness without forcing an individual to respond to an item that (a) might not be clear to them, (b) might not be appropriate for them, and/or (c) might result in an answer that is untrue for their unique circumstance.

3.4 Qual-TRIX

Collaborating with other Qualtrics users in your institution is easy! Scroll down to “Collaborating Inside Your Organization” and follow the instructions for adding individuals to your survey (you must “own” the survey...your collaborators will not be able to add others).

The ability to **schedule survey distributions** is like having your very own GA! If you have a roster(contact list) you can schedule distributions, reminders, and thank yous. Qualtrics will keep track of who responds and send reminders to the non-responders. Here are resources for

- [E-mail overview](#)
- [E-mail distribution management](#)
- [Contact lists](#)

Personalizing invitations and surveys. [Piped text](#) is a way to personalize invitations and/or “carry forward” prior responses into new questions.

Randomization of blocks (or a subset of blocks) can be used for several purposes such as: (a) using random selection to display one or more blocks to respondents – as in a random clinical trial, (b) to randomly display a percentage of blocks or items to shorten the survey in a planned missing design, and (c) randomly display some or all of the blocks of the survey to all respondents so that when respondents experience test fatigue, when they quit responding, “the last items/surveys” aren’t always the same ones – thus distributing the missingness across surveys.

Randomization of items within a block can be used for similar purposes. You can also use this to display only some of the items (e.g., planned missingness).

File upload from respondents is an additional package that requires the institution to pay a higher fee. If available, this allows respondents to upload some sort of file (photo, powerpoint, .pdf). We use it for poster contests at professional contests (where students upload their poster for online judging in advance of the conference). A colleague of mine uses this function to collect application elements (i.e., resumes, cover letters, reference letters) to a fellowship program.

- As researchers, we can also upload files (e.g., hardcopy of informed consent, documents to be reviewed) for use by the respondent.

Display, Skip, and/or Branch Logic can be used to help display to respondents *only* the items that pertain to them. There are multiple approaches to doing this. Using a display logic approach may feel a bit *backward* where the logic is applied *from* the landing spot. We did this extensively in a study that involved two language versions and three age options.

Two other approaches for these issues are [skip logic](#) and [branch logic](#)

3.5 Even moRe, particularly relevant to iRb

We can use Qualtrics tools for purposes beyond collecting and downloading data. These tools are especially useful when I think about IRB applications and ethics related to data collection.

Exporting to Word Helpful for your IRB application (and perhaps in a cloud so that a team can use track changes to edit), it is super simple to export the survey to Microsoft Word. PLUS! You have options for including question numbers, recode values, logic, etc., so that it is essentially a codebook.

Anonymizing responses Another step toward an anonymous response is to withhold the IP address. This is accomplished in the Survey Options menu.

Prevent ballot box stuffing Want to make sure that respondents only answer once? In the same Survey Options window, you can prevent ballot box stuffing.

Other security options include

- Password protection
- HTTP Referer verification

Look also at:

- **Progress bar** to provide participants hope (or despair) for “how much longer.”
- **Survey termination** to connect custom endings and thank-you notes.
- **Partial completion** to specify how long the respondent has to complete the survey (after opening it) and whether it is recorded or deleted if it is not completed.
 - Related to this, back on the Data & Analysis tab, you can see both #s of [recorded responses and responses in progress](#). You also have options to manually determine how you want to include/exclude the response in progress.
 - ARGHGHGHGHGH!!!! That grubby little “->” submit and progress symbol is often the reason that surveys that are > 90% complete aren’t counted as “complete.” What to do? Options: (a) don’t say “Thanks and goodbye” on a page that has any items, and (b) provide instructions to look for the “->” symbol to continue.

Finally, **PREVIEW PREVIEW PREVIEW!** There is no better way to check your work than with previews.

3.6 intRavenous Qualtrics

Access credentials for the institutional account, individual user’s account, and survey are essential for getting the survey items and/or results to export into R. The Qualtrics website provides a tutorial for [generating an API token](#).

We need two pieces of information: the **root_url** and an **API token**.

- Log into your respective qualtrics.com account.
- Select Account Settings
- Choose “Qualtrics IDs” from the user name dropdown

We need the **root_url**. This is the first part of the web address for the Qualtrics account. For our institution it is: spupsych.az1.qualtrics.com

The API token is in the box labeled, “API.” If it is empty, select, “Generate Token.” If you do not have this option, locate the *brand administrator* for your Qualtrics account. They will need to set up your account so that you have API privileges.

BE CAREFUL WITH THE API TOKEN This is the key to your Qualtrics accounts. If you leave it in an .rmd file that you forward to someone else, this key and the base URL gives access to every survey in your account. If you share it, you could be releasing survey data to others that would violate confidentiality promises in an IRB application.

If you mistakenly give out your API token you can generate a new one within your Qualtrics account and re-protect all its contents.

You do need to change the API key/token if you want to download data from a different Qualtrics account. If your list of surveys generates the wrong set of surveys, restart R, make sure you have the correct API token and try again.

```
#only have to run this ONCE to draw from the same Qualtrics account...but will need to get diff
library(qualtRics)
qualtRics::qualtrics_api_credentials(api_key = "mUgPMySYkiWpMFkwHale1QE5HNmh5LRUaA8d9PDg",
                                      base_url = "spupsych.az1.qualtrics.com", overwrite = TRUE, install = TRUE)
readRenviron("~/Renvironment")
```

`all_surveys()` generates a dataframe containing information about all the surveys stored on your Qualtrics account.

```
surveys <- all_surveys()
#View this as an object (found in the right: Environment).
#Get survey id # for the next command
#If this is showing you the WRONG list of surveys, you are pulling from the wrong Qualtrics ac
surveys
```

To retrieve the survey, use the `fetch_survey()` function.

```
#obtained with the survey ID
#"surveyID" should be the ID from above
#"verbose" prints messages to the R console
#"label", when TRUE, imports data as text responses; if FALSE prints the data as numerical resp
#"convert", when TRUE, attempts to convert certain question types to the "proper" data type in
#"force_request", when TRUE, always downloads the survey from the API instead of from a tempor
# "import_id", when TRUE includes the unique Qualtrics-assigned ID; since I have provided labe

#Out of the blue, I started getting an error, that R couldn't find function "fetch_survey." A
QTRX_df <- qualtRics::fetch_survey(surveyID = "SV_b2cClqA1LGQ6nLU", time_zone = NULL, verbose =
```

```
-- Column specification -----
cols(
  .default = col_double(),
  StartDate = col_datetime(format = ""),
  EndDate = col_datetime(format = ""),
  RecordedDate = col_datetime(format = ""),
  ResponseId = col_character(),
  DistributionChannel = col_character(),
  UserLanguage = col_character(),
  Virtual_6 = col_logical(),
  `5_iPronouns` = col_logical(),
  `5_iGenderConf` = col_logical(),
  `5_iRace` = col_logical(),
  `5_iUS` = col_logical(),
  `5_iDis` = col_logical(),
```

```

`6_iPronouns` = col_logical(),
`6_iGenderConf` = col_logical(),
`6_iRace` = col_logical(),
`6_iUS` = col_logical(),
`6_iDis` = col_logical(),
`7_iPronouns` = col_logical(),
`7_iGenderConf` = col_logical(),
`7_iRace` = col_logical()
# ... with 17 more columns
)
i Use `spec()` for the full column specifications.

```

Warning in file.remove(survey.fpath): cannot remove file 'C:/Users/lhbikos/AppData/Local/Temp/RtmpoN3kVl/Recent Course Eval.csv', reason 'Permission denied'

```
#useLocalTime = TRUE,
```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```

#write the simulated data as a .csv
#write.table(QTRX_df, file="QTRX_df.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#QTRX_df <- read.csv ("QTRX_df.csv", header = TRUE)

```

```

#to save the df as an .rds (think "R object") file on your computer; it should save in the same directory as the .csv file
#saveRDS(QTRX_df, "QTRX_df.rds")
#bring back the simulated dat from an .rds file
#QTRX_df <- readRDS("QTRX_df.rds")

```

3.6.1 The Codebook

In order to prepare data from a survey, it is critical to know about its content, scoring directions for scales/subscales, and its design. As I demonstrated above, we can export a **codebook**, that is, a Word (or PDF) version of the survey with all the coding. In Qualtrics the protocol is: Survey/Tools/ImportExport/Export Survey to Word. Then select all the options you want (especially “Show Coded Values”). A tutorial provided by Qualtrics can be found [here](#). This same process can be used to print the PDF example I used above.

I recommend providing custom variable names and recode values directly in Qualtrics before exporting them into R. A Qualtrics tutorial for this is provided [here](#). In general, consider these qualities when creating variable names:

- Brevity: historically, SPSS variable names could be a maximum of 8 characters.

- Intuitive: although variables can be renamed in R (e.g., for use in charts and tables), it is helpful when the name imported from Qualtrics provides some indication of what the variable is.
- Systematic: start items in a scale with the same stem, followed by the item number – ITEM1, ITEM2, ITEM3.

More complete information about data preparation is covered in chapters in the [ReCentering Psych Stats: Multivariate Modeling](#) text.

3.6.2 Using data from an exported Qualtrics .csv file

The lecture focused on the “intRavenous” import. It is also possible to download the Qualtrics data in a variety of formats (e.g., CSV, Excel, SPSS). Since I got started using files with the CSV extension (think “Excel” lite), that is my preference.

In Qualtrics, these are the steps to download the data: Projects/YOURsurvey/Data & Analysis/Export & Import/Export data/CSV/Use numeric values

I think that it is critical that to save this file in the same folder as the .rmd file that you will use with the data.

R is sensitive to characters used in filenames. As downloaded, my Qualtrics .csv file had a long name with spaces and symbols that are not allowed. Therefore, I gave it a simple, sensible, filename, “ReC_Download210319.csv”. An idiosyncrasy of mine is to datestamp filenames. I use two-digit representations of the year, month, and date so that if the letters preceding the date are the same, the files would alphabetize automatically.

```
#library(qualtrics)
#QTRX_csv <- read_survey("ReC_Download210319.csv", strip_html = TRUE, import_id = FALSE, time_...
```

Although minor tweaking may be required, the same script above should be applicable to this version of the data.

3.6.3 Tweaking Data Format

Two general approaches:

1. Inside Qualtrics: Use the recode values option (found in the item’s gearbox, to the left of the block) to specify variable names and recode values. These should be preserved on the download. <https://www.qualtrics.com/support/survey-platform/survey-module/question-options/recode-values/>
2. In the R script: In another lecture I demonstrate how to change the formats of data (character, string), selecting only the variables in which we are interested (e.g., excluding the meta-data), and renaming variables sensibly.

Both work! Just a preference – and probably have an explicit process approach within your research team/collaborators.

3.7 Practice Problems

The suggestion for practice is to develop a questionnaire, format it, pilot it, and download it. Essentially you will be

- Formatting a survey on Qualtrics using all the best practices identified in the lecture
 - these include having an introductory statement (to include statement of confidentiality), directions for each sub-survey (if more than one), and closing statement.
 - selecting the most appropriate question type for the items. For example, matrix instead of multiple choice.
 - within question type, using the appropriate options for proper formatting (e.g., the anchors in a matrix should be topically consistent and equal-interval)
- The survey should include minimum of 3 of the qualTRIXter skills (identified in lecture); choose from
 - establishing collaboration
 - scheduling e-mail distribution and follow-up
 - personalizing the survey in some way
 - randomization of blocks or items
 - integrating display, skip, or branch logic (e.g., having males and females take a different route)
 - exporting the survey to Word
 - recoding variables in the item controls
 - anonymize the responses
 - prevent ballot box stuffing
 - include a progress bar
 - create a custom ending, e-mail, or thank-you note
 - something else that YOU discovered that isn't in the lecture
- Piloting it, getting their feedback, and identifying what problems are (and how you might fix them)
 - with 3 folks from your RVT, cohort, or this class
 - with 3 additional folks who aren't quite as "research savvy"
 - collect their feedback (ideally in a text-item directly on the survey itself) and write a brief summary (3 paragraphs max) of their impressions and how you might improve the survey
- Import the Qualtrics data directly R
 - preferably, directly from Qualtrics with the API token, base URL, and survey ID
 - alternatively (for the same # of points) from the exported CSV file *via the qualtrics package* (required)

Assignment Component	Points Possible	Points Earned
1. Qualtrics survey best practices	5	
2. QualTRIXter skills (at least 3)	5	

Assignment Component	Points Possible	Points Earned
3. Minimum of 6 pilot respondents	5	
4. Summary of pilot feedback	5	
5. Import of Qualtrics data into R	5	
6. Explanation to grader	5	
Totals		20

Chapter 4

Psychometric Validity: Basic Concepts

[Screencasted Lecture Link](#)

The focus of this lecture is to provide an introduction to validity. This includes understanding some of the concerns of validity, different aspects of validity, and factors as they affect validity coefficients.

4.1 Navigating this Lesson

There is just over one hour of lecture. If you work through the materials with me it would be plan for an additional hour.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

4.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Distinguish between different types of validity based on short descriptions.
- Compute and interpret validity coefficients.
- Evaluate the incremental validity of an instrument-of-interest.
- Define and interpret the standard error of estimate.
- Develop a rationale that defends importance of establishing the validity of a measuring instrument.

4.1.2 Planning for Practice

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. With each of these options I encourage you to interpret examine aspects of the construct validity through the creation and interpretation of validity coefficients. Ideally, you will examine both convergent/discriminant as well as incremental validity.

4.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (when possible, linked) in the text with complete citations in the reference list.

- Jhangiani, R. S., Chiang, I.-C. A., Cuttler, C., & Leighton, D. C. (2019). Reliability and Validity. In *Research Methods in Psychology*. <https://doi.org/10.17605/OSF.IO/HF7DQ>
- Clark, L. A. & Watson, D. (1995). Constructing validity: Basic issues in objective scale development. *Psychological Assessment*, 7, 309-319.
 - In this manuscript, Clark and Watson (1995) create a beautiful blend of theoretical issues and practical suggestions for creating measures that evidence construct validity. From the practical perspective, the authors first guide potential scale constructors through the literature review and creating an item pool (including tips on writing items). The authors address structural validity by first beginning with strategies for constructing the test. In this section, the authors revisit the issue of dimensionality (i.e., alpha vs. factor analysis). Finally, the authors look at initial data collection (addressing sample size) and psychometric evaluation.

4.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(tidyverse)){install.packages("tidyverse")}
#if(!require(MASS)){install.packages("MASS")}
#if(!require(psych)){install.packages("psych")}
```

4.2 Research Vignette

Explorations of validity are frequently correlational in nature. This lesson provides descriptions of numerous pathways for establishing an instrument's validity. Best practices involving numerous demonstrations of validity. Across several lessons, we will rework several of the analyses reported in the research vignette. For this lesson in particular, the research vignette allows both convergent/discriminant validity as well as incremental validity.

The research vignette for this lesson is the development and psychometric evaluation of the Perceptions of the LGBTQ College Campus Climate Scale (?). The scale is six items with responses rated on a 7-point Likert scale ranging from 1 (*strongly disagree*) to 7 (*strongly agree*). Higher scores indicate more negative perceptions of the LGBTQ campus climate. Szymanski and Bissonette have suggested that the psychometric evaluation supports using the scale in its entirety or as subscales composed of the following items:

- College response to LGBTQ students:
 - My university/college is cold and uncaring toward LGBTQ students.
 - My university/college is unresponsive to the needs of LGBTQ students.
 - My university/college provides a supportive environment for LGBTQ students.
- LGBTQ Stigma:
 - Negative attitudes toward LGBTQ persons are openly expressed on my university/college campus.
 - Heterosexism, homophobia, biphobia, transphobia, and cissexism are visible on my university/college campus.
 - LGBTQ students are harassed on my university/college campus.

A preprint of the article is available at ResearchGate.

In the **lesson** on reliability, I simulate item-level data. However, this lesson we will be interested in the correlations between the total and subscale scores with five additional scales:

- LGBTQ victimization
- Satisfaction with college
- Intention to persist in college
- Generalized anxiety disorder symptoms
- Symptoms of depression

```
Szy_mu <- c(3.16, 2.71, 3.61, .11, 5.61, 4.41, 1.45, 1.29)
Szy_sd <- c(1.26, 1.33, 1.51, .23, 1.15, .53, .80, .78)
Szy_r_mat <- matrix(c(1, .88, .90, .35, -.56, -.27, .25, .24,
                     .88, 1, .58, .25, -.59, -.29, .17, .18,
                     .90, .58, 1, .37, -.41, -.19, .27, .24,
                     .35, .25, .37, 1, -.22, -.04, .23, .21,
                     -.56, -.59, -.41, -.22, 1, .53, -.29, -.32,
                     -.27, -.29, -.19, -.04, .53, 1, -.22, -.26,
                     .25, .17, .27, .23, -.29, -.22, 1, .76,
                     .24, .18, .24, .21, -.32, -.26, .76, 1), ncol = 8)
Szy_cov_mat <- Szy_sd %*% t(Szy_sd) * Szy_r_mat

set.seed(210907)
SzyDF <- round(as.data.frame(MASS::mvrnorm(n = 646, mu=Szy_mu, Sigma=Szy_cov_mat, tol=1e-3, emp))
SzyDF <- round(dplyr::rename(SzyDF, CClimate = V1, CRResponse = V2, Stigma = V3, Victimization =
#round(cor(SzyDF),2)
```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```
#write the simulated data as a .csv
#write.table(SzyDF, file="SzyDF.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#SzyDF <- read.csv ("SzyDF.csv", header = TRUE)

#to save the df as an .rds (think "R object") file on your computer; it should save in the same directory as the .csv file
#saveRDS(SzyDF, "SzyDF.rds")
#bring back the simulated dat from an .rds file
#SzyDF <- readRDS("SzyDF.rds")
```

4.3 Fundamentals of Validity

Validity (the classic definition) is the ability of a test to measure what it purports to measure. Supporting that definition are these notions:

- Validity is extent of matching, congruence, or “goodness of fit” between the operational definition and concept it is supposed to measure.
- An instrument is said to be valid if it taps the concept it claims to measure.
- Validity is the appropriateness of the interpretation of the results of an assessment procedure for a given group of individuals, not to the procedure itself.
- Validity is a matter of degree; it does not exist on an all-or-none basis.
- Validity is always specific to some particular use or interpretation.
- Validity is a unitary concept.
- Validity involves an overall evaluative judgment.

Over the years (and, perhaps within each construct), validity has somewhat of an *evolutionary* path from a focus on content, to prediction, to theory and hypothesis testing.

When the focus is on **content**, we are concerned with the:

- assessment of what individuals had learned in specific content areas
- relevance of its content (i.e., we compare the content to the content domain)

When the focus is on **prediction**, we are concerned with:

- how different persons respond in a given situation (now or later).
- the correlation coefficient between test scores (predictor) and the assessment of a criterion (performance in a situation)

A focus on **theory and hypothesis testing** adds:

- a strengthened theoretical orientation

- a close linkage between psychological theory and verification through empirical and experimental hypothesis testing
- an emphasis on constructs in describing and understanding human behavior.

Constructs are broad categories, derived from the common features shared by directly observable behavioral variables. They are theoretical entities and not directly observable. **Construct validity** is at the heart of psychometric evaluation. We define **construct validity** as the fundamental and all-inclusive validity concept, insofar as it specifies what the test measures. Content and predictive validation procedures are among the many sources of information that contribute to the understanding of the constructs assessed by a test.

4.4 Validity Criteria

We have just stated that validity is an overall, evaluative judgment. Within that umbrella are different criteria by which we judge the validity of a measure. We casually refer to them as *types*, but each speaks to that unitary concept.

4.4.1 Content Validity

Content validity is concerned with the representativeness of the domain being assessed. Content validation procedures may differ depending on whether the test is in the educational/achievement context or if it is more of an attitude/behavioral survey.

In the educational/achievement context, content validation seeks to ensure the items on an exam are appropriate for the content domain being assessed.

A **table of specifications** is a two-way chart which indicates the instructionally relevant learning tasks to be measured. Percentages in the table indicate the relative degree of emphasis that each content area

Let's imagine that I was creating a table of specifications for items on a quiz for this very chapter. The columns represent the types of outcomes we might expect. The American Psychological Association often talks about *KSAs* (knowledge, skills, attitudes), so I will utilize those as a framework. You'll notice that the number of items and percentages do not align mathematically. Because, in the exam, I would likely weight application items (e.g., "work the problem") more highly than knowledge items (e.g., multiple choice), the relative weighting may differ.

Table of Specifications

Learning Objectives	Knowledge	Skills	Attitudes	% of test
Distinguish between different types of validity based on short descriptions.	6 items			30%
Compute and interpret validity coefficients.		2 items		15%
Evaluate the incremental validity of an instrument-of-interest.		1 item		20%

Learning Objectives	Knowledge	Skills	Attitudes	% of test
Define and interpret the standard error of estimate.	1 item			15%
Develop a rationale that defends importance of establishing the validity of a measuring instrument.			1 item	20%
TOTALS	7 items	3 items	1 item	100%

Subject matter experts (SMEs) are individuals chosen to evaluate items based on their degree of knowledge of the subject being assessed. If used:

- report how many and professional qualifications;
- to classify items, report the directions they were given and the extent of agreement among judges.

Empirical procedures for enhancing content validity of educational assessments may include:

- comparing item-level and total scores with grades; lower grades should get lower scores;
- analyzing individual errors;
- observing student work methods (have the students “think aloud” in front of an examiner);
- evaluating the role of speed, noting how many do not complete the test in the time allowed;
- correlating the scores with a reading comprehension test (if the exam is highly correlated, then it may be a test of reading and not another subject). Alternatively, if it is a reading comprehension test, give the student the questions (without the passage) to see how well they answered the questions on the basis of prior knowledge.

For surveys and tests outside of educational settings, content validation procedures ask, “Does the test cover a representative sample of the specified skills and knowledge?” and “Is test performance reasonably free from the influence of irrelevant variables?” Naturally, SMEs might be used.

An example of content validation from Industrial-Organizational Psychology is the job analysis which precedes the development of test for employee selection and classification. Not all tests require content analysis. In aptitude and personality tests we are probably more interested in other types of validity evaluation.

4.4.2 Face Validity: The “Un”validity

Face validity is concerned with the question, “How does an assessment look on the ‘face of it’?” Let’s imagine that on a qualification exam for electricians, a math item asks the electrician to estimate the amount of yarn needed to complete a project. The item may be more *face valid* if the calculation was with wire. Thus, face validity can often be improved by reformulating test items in terms that appear relevant and plausible for the context.

Face validity should never be regarded as a substitute for objectively determined validity. In contrast, it should not be assumed that when a (valid and reliable) test has been modified to increase its face validity, that its objective validity and reliability is unaltered. That is, it must be reevaluated.

4.4.3 Criterion-Related Validity

Criterion-related validity has to do with the test's ability to *predict* an outcome (the criterion). If the criterion is something that occurs simultaneously, it is an assessment of **concurrent validity**; if it is in the future, it is an assessment of **predictive validity**.

A **criterion** is the “thing” that the test should, theoretically, be able to *predict*. That prediction could be occurring at the same time (*concurrent validity*) or at a future time (*predictive validity*). Regardless, the estimate of the criteria must be independent of the survey/assessment being evaluated. The table below provides examples of types of tests and concurrent and predictive validity criteria.

Type of Test	Concurrent Criteria Example	Predictive Criteria Example
A shorter (or cheaper) standardized achievement test	school grades, existing standardized tests	subsequent graduation/college admissions, cumulative GPA
Employee selection tests	decision made by a search committee	subsequent retention or promotion of the selected employee
Assessment of depression severity (shorter or cheaper)	diagnosis from a mental health professional; correlation with an established measure	inpatient hospitalization or act of self-harm

Contrasted groups is a specific type of criterion-related validity. Clearly differentiated groups (e.g., sales clerks versus executives; engineers versus musicians) are chosen to see if exam performance or profiles differ in predictable ways.

Criterion contamination occurs when test scores, themselves, are used to make decisions about the criteria. To prevent this:

- No person who participates in the assignment of criterion ratings can have any knowledge of the examinee's test scores.
- the test scores must be kept strictly confidential.

There are a number of issues related to criterion-related validity.

- Is the criterion choice appropriate?
 - Criterion validity is only as good as the validity of the criterion to which one is making a comparison. – In the 1980s and 1990s there was more attention in this area. That is critics questioned the quality of the criterion being used.
- To what degree can the results of criterion-related validity be generalized?
 - Most tests are developed (intentionally) for a local context, setting, or audience. Consequently, in the local context, the criterion-prediction sample is usually too small (i.e., 50 cases or less).

- Those who want to generalize the test to a broader population should evaluate the test in relation to the new purpose.
- Is there a role for meta-analysis?
 - Repeated validation studies of our tests, on different samples, results in a number of small-scale studies, each with their own validity coefficients.
 - We can use meta-analytic procedures in reporting the results of validity coefficients when they are used for establishing criterion validity.

4.4.4 Construct Validity

Construct validity was introduced in 1954 in the first edition of APA's testing standards and is defined as the extent to which the test may be said to measure a theoretical construct or trait. The overarching focus is on the role of *psychological theory* in test construction and the ability to formulate hypotheses that can be supported (or not) in the evaluation process. Construct validity is established by the accumulation of information from a variety of sources.

There are a number of sources that can be used to support construct validity.

4.4.5 Internal Consistency

In the next [chapter](#), you will learn that **internal consistency** is generally considered to be an index of reliability. In the context of criterion-related validity, a goal is to ensure that the criterion is the total score on the test itself. To that end, some of the following could also support this aspect of validity:

- Comparing high and low scorers. Items that fail to show a significantly greater proportion of “passes” in the upper than the lower group are considered invalid, and are modified or eliminated.
- Computing a biserial correlation between the item and total score.
- Correlating the subtest score with the total score. Any subtest whose correlation with the total score is too low is eliminated.

Although some take issue with this notion, the degree of *homogeneity* (the degree to which items assess the same thing) has some bearing on construct validity. There is a tradeoff between items that measure a narrow slice of the construct definition (internal consistency estimates are likely to be higher) and those that sample the construct definition more broadly (internal consistency estimates are likely to be lower).

Admittedly, the contribution of internal consistency data is limited. In absence of external data, it tells us little about **WHAT** the test measures.

4.4.6 Structural Validity

4.4.6.1 Exploratory Factor Analysis

Exploratory factor analysis (EFA) is used to simplify the description of behavior by reducing the number of categories (factors or dimensions) to as many as the numbers of the items to fewer. In instrument development, techniques like *principal components analysis* or *principal axis factoring* are used to identify clusters (latent factors) among items. We frequently treat these as scales and subscales.

Imagine the use of 20 tests to 300 people. There would be 190 correlations.

- Irrespective of content, we can probably summarize the intercorrelations of tests with 5-6 factors.
- When the clustering of tests includes vocabulary, analogies, opposites, and sentence completions, we might suggest a “verbal comprehension factor.”
- Factorial validity is the correlation of the test with whatever is common to a group of tests or other indices of behavior. If our single test has a correlation of .66 with the factor on which it loads, then the “factorial validity of the new test as a measure of the common trait is .66.”

When EFA is utilized, the items are “fed” into an iterative process that analyzes the relations and “reveals” (or suggests – we are the ones who interpret the data) how many factors (think scales/subscales) and which items comprise them.

4.4.6.2 Confirmatory Factor Analysis

Confirmatory factor analysis (CFA) involves specifying, *a priori*, a proposed relationship of items, scales, and subscales and then testing its *goodness of fit*. In CFA (a form of structural equation modeling [SEM]), the latent variables (usually the higher order scales and total scale score) are positioned to cause the responses on the indicators/items.

Subsequent lessons provide examples of both EFA and CFA approaches to psychometrics.

4.4.7 Experimental Interventions

Construct validity is also supported by hypothesis testing and experimentation. If we expect that the construct assessed by the instrument is malleable (e.g., depression) and that an intervention could change it, then a random clinical trial that evaluated the effectiveness of an intervention (and it worked – depression scores declined) would simultaneously provide support for the intervention as well as the instrument.

4.4.8 Convergent and Discriminant Validity

In a psychometric evaluation, we will often administer our instrument-of-interest along with a battery of instruments that are more-and-less related. **Convergent validity** is supported when there are *moderately high* correlations between our tests and the instruments with which we expect moderately high correlations. In contrast, **discriminant validity** is established by low and/or

non-significant correlations between our instrument-of-interest and instruments that should be unrelated. For example, we want a low and non-significant correlation between a quantitative reasoning test and scores on a reading comprehension test. Why? Because if the correlation is too high, the test cannot discriminate between reading comprehension and math.

There are no strict cut-offs to establish convergence or discrimination. We can even ask, “Could a correlation intended to support convergence be too high?” It is possible! Unless the instrument-of-interest offers advantages such as brevity or cost, then correlations that fall into the ranges of multicollinearity or singularity can indicate unnecessary duplication or redundancy.

In our research vignette, Szymanski and Bissonette (?) conduct a correlation matrix that reports the bivariate relations between the LGBTQ Campus Climate full-scale as well as the College Response and Stigma subscales with measures that assess (a) LGBTQ victimization, (b) satisfaction with college, (c) persistence attitudes, and (d) anxiety, and (e) depression.

```
apaTables::apa.cor.table(SzyDF, filename = "SzyCor.doc", table.number = 1, show.sig.stars=TRUE)
```

Table 1

Means, standard deviations, and correlations with confidence intervals

Variable	M	SD	1	2	3
1. CClimate	3.16	1.26			
2. CResponse	2.71	1.33	.88** [.86, .90]		
3. Stigma	3.61	1.51	.90** [.88, .91]	.58** [.53, .63]	
4. Victimization	0.11	0.23	.35** [.28, .42]	.25** [.18, .32]	.37** [.30, .44]
5. CollSat	5.61	1.15	-.56** [-.61, -.50]	-.59** [-.64, -.54]	-.41** [-.47, -.34]
6. Persistence	4.41	0.53	-.27** [-.34, -.20]	-.29** [-.36, -.22]	-.19** [-.26, -.11]
7. Anxiety	1.45	0.80	.25** [.18, .32]	.17** [.09, .24]	.27** [.20, .34]
8. Depression	1.29	0.78	.24** [.17, .31]	.18** [.10, .25]	.24** [.17, .31]
4	5	6		7	

-.22**				
[-.29, -.15]				
-.04	.53**			
[-.12, .04]	[.47, .58]			
.23**	-.29**	-.22**		
[.16, .30]	[-.36, -.22]	[-.29, -.15]		
.21**	-.32**	-.26**	.76**	
[.14, .28]	[-.39, -.25]	[-.33, -.19]	[.73, .79]	

Note. M and SD are used to represent mean and standard deviation, respectively.

Values in square brackets indicate the 95% confidence interval.

The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014).

* indicates $p < .05$. ** indicates $p < .01$.

Examination of these values (which align well with the results in the table) follow the expected pattern. That is, higher scores on the overall Campus Climate, College Response, and Stigma scales result in higher levels of victimization, anxiety, and depression. Conversely, they are associated with lower college satisfaction and persistence.

The College Response and Stigma subscales' relations with satisfaction with college (-.59, -.41, respectively) and persistence attitudes (-.29*, -.19, respectively) are examples of the convergent and discriminant patterns (College Response has higher relations with these than Stigma) that support construct validity.

The **multitrait-multimethod matrix** is a systematic experimental design for the dual approach of convergent and discriminant validation, which requires the assessment of two or more traits (classically, math, English, and reading scores) by two more methods (self, parent, and teacher). Conducting a web-based image search on this term will show a matrix of alpha coefficients and correlation coefficients that are interpreted in relationship to each other. Roughly:

- alpha coefficients (internal consistency) should be the highest,

- validity coefficients (correlations of the same trait assessed by different methods) should be higher than correlations between different traits measured by different methods,
- validity coefficients (correlations of the same trait assessed by different methods) should be higher than different traits measured by the same method.

4.4.9 Incremental Validity

Incremental validity is the increase in predictive validity attributable to the test. It indicates the contribution the test makes to the selection of individuals who will meet the minimum standards in criterion performance. There are different ways to assess this – one of the most common is to first enter known predictors and then see if the instrument-of-interest continues to account variance over-and-above those that are entered.

In the Szymanski and Bissonette (?) psychometric evaluation, the negative relations with satisfaction with college and intention to persist in college as well as positive relations with both anxiety and depression persisted even after controlling for LGBTQ victimization experiences.

I will demonstrate this procedure, predicting the contribution that the LGBTQ Campus Climate total scale score has on predicting intention to persist in college, over and above LGBTQ victimization.

The process is to use hierarchical linear regression. Two models are built. In the first mode (“PfV” stands [in my mind] for “Persistence from Victimization”), persistence is predicted from victimization. The second model adds the LGBTQ Campus Climate Scale. I asked for summaries of each model. Then the *anova()* function compares the model.

```
PfV <- lm(Persistence ~ Victimization, data = SzyDF)
PfVC <- lm(Persistence ~ Victimization + CClimate, data = SzyDF)
summary(PfV)
```

Call:

```
lm(formula = Persistence ~ Victimization, data = SzyDF)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.59249	-0.34872	-0.01686	0.35461	1.65781

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.42029	0.02312	191.183	<0.0000000000000002 ***
Victimization	-0.09367	0.09079	-1.032	0.303

Signif. codes:	0 ***	0.001 **	0.01 *	0.05 .
	'	'	'	'

Residual standard error: 0.5302 on 644 degrees of freedom

Multiple R-squared: 0.00165, Adjusted R-squared: 0.0001002

F-statistic: 1.065 on 1 and 644 DF, p-value: 0.3025

From the PfVCmodel we learn that victimization has a non-significant effect on intentions to persist in college Further, the R^2 is quite small (0.002).

```
summary(PfVC)
```

Call:

```
lm(formula = Persistence ~ Victimization + CClimate, data = SzyDF)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.54270	-0.33725	-0.01329	0.34526	1.65393

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.78193	0.05489	87.119 < 0.0000000000000002	***
Victimization	0.14182	0.09330	1.520	0.129
CClimate	-0.12263	0.01701	-7.208	0.0000000000016 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5104 on 643 degrees of freedom

Multiple R-squared: 0.07628, Adjusted R-squared: 0.07341

F-statistic: 26.55 on 2 and 643 DF, p-value: 0.00000000008341

In the PfVC model, we see that the LGBTQ Campus Climate full scale score has a significant impact on intentions to persist. Specifically, foreach additional point higher on the Campus Climate Score, intentions to persist decrease by .13 points. Together, the model accounts for 7% of the variance (this is also an R^2 change of 7%).

```
anova(PfV, PfVC)
```

Analysis of Variance Table

Model 1: Persistence ~ Victimization

Model 2: Persistence ~ Victimization + CClimate

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	644	181.01			
2	643	167.48	1	13.531 51.95	0.000000000001602 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

We see that there is a statistically significant difference between the models.

Let's try another model.

```
AfV <- lm(Anxiety ~ Victimization, data = SzyDF)
AfVC <- lm(Anxiety ~ Victimization + CClimate, data = SzyDF)
summary(AfV)
```

Call:
`lm(formula = Anxiety ~ Victimization, data = SzyDF)`

Residuals:

Min	1Q	Median	3Q	Max
-2.40178	-0.51888	0.04366	0.54301	2.51019

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	1.36192	0.03398	40.081	< 0.0000000000000002 ***		
Victimization	0.80273	0.13342	6.016	0.00000000299 ***		

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1

Residual standard error: 0.7791 on 644 degrees of freedom
Multiple R-squared: 0.05322, Adjusted R-squared: 0.05175
F-statistic: 36.2 on 1 and 644 DF, p-value: 0.000000002991

```
summary(AfVC)
```

Call:
`lm(formula = Anxiety ~ Victimization + CClimate, data = SzyDF)`

Residuals:

Min	1Q	Median	3Q	Max
-2.47857	-0.48764	0.01947	0.52990	2.50069

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	1.00143	0.08241	12.152	< 0.0000000000000002 ***		
Victimization	0.56799	0.14008	4.055	0.00005635 ***		
CClimate	0.12224	0.02554	4.786	0.00000212 ***		

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1

Residual standard error: 0.7662 on 643 degrees of freedom
Multiple R-squared: 0.08578, Adjusted R-squared: 0.08293
F-statistic: 30.16 on 2 and 643 DF, p-value: 0.0000000000003007

```
anova(AfV, AfVC)
```

Analysis of Variance Table

```

Model 1: Anxiety ~ Victimization
Model 2: Anxiety ~ Victimization + CClimate
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     644 390.94
2     643 377.50  1    13.445 22.901 0.000002117 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

This model is a little more exciting in that our first model (AfV) is statistically significant. That is, victimization has a statistically significant effect on anxiety, accounting for 5% of the variance. Even so, when added, the LGBTQ Campus Climate total scale score is also significant, and accounts for an additional 4% of variance (ΔR^2), $R^2 = 0.85$. There is a statistically significant difference between models ($F[1, 643] = 22.98, p < .001$).

4.4.10 Considering the Individual and Social Consequences of Testing

Messick (?) and others recommend that the “consequences of testing” be included in the concept of test validity. Messick’s point was to consider the unintended consequences of specific uses. That is, their use may be detrimental to individuals or to members of certain ethnic or other populations with diverse experiential backgrounds. Examples of inappropriate use have included:

- The California Psychological Inventory (CPI) being used as a screening tool for employment as a security job. Two of its items inquired about same-sex activities and the employer was using this to screen out gay men. Applicants were able to demonstrate, in court, a consistent rejection of gay applicants.
- While this is not a psychological test, urine samples are often collected as drug screening tools. In reality, urine can reveal a number of things, such as pregnancy.

The issue begs “conflicting goals.” In this case, the problem was not caused by the test but rather by its misuse. Studying the “consequences” of testing is one that is not necessarily answerable by empirical data/statistical analysis. It requires critical observation, human judgment, and systematic debate.

4.5 Factors Affecting Validity Coefficients

Keeping in mind that a *validity coefficient* is merely the correlation between the test and some criteria, the same elements that impact the magnitude and significance of a correlation coefficient will similarly effect a validity coefficient.

Nature of the group. A test that has high validity in predicting a particular criterion in one population, may have little or no validity in predicting the same criterion in another population. If a test is designed for use in diverse populations, information about the population generalizability should be reported in the technical manuals.

Sample heterogeneity Other things being equal, if there is a linear relationship between X and Y, it will have a greater magnitude when the sample is heterogeneous.

Pre-selection. Just like internal and external validity in a research design can be threatened by selection issues, pre-selection can also impact the validity coefficients of a measure. For example, if we are evaluating a new test for job selection, we may select a group of newly hired employees. We plan to collect some measure of job performance at a later date. Our results may be limited by the criteria used to select the employees. Were they volunteers? Were they only those hired? Were they ALL of the applicants?

Validity coefficients may change over time. Consider the relationship between the college boards and grade point average at Yale University. Fifty years ago, $r_{xy} = .72$; today $r_{xy} = .52$. Why? The nature of the student body has become more diverse (50 years ago, the student body was predominantly White, high SES, and male).

The **form of the relationship** matters. The Pearson R assumes the relationship between the predictor and criterion variables is linear, uniform, and homoschedastic (equal variability throughout the range of a bivariate distribution). When the variability is unequal throughout the range of the distribution the relationship is heteroscedastic.

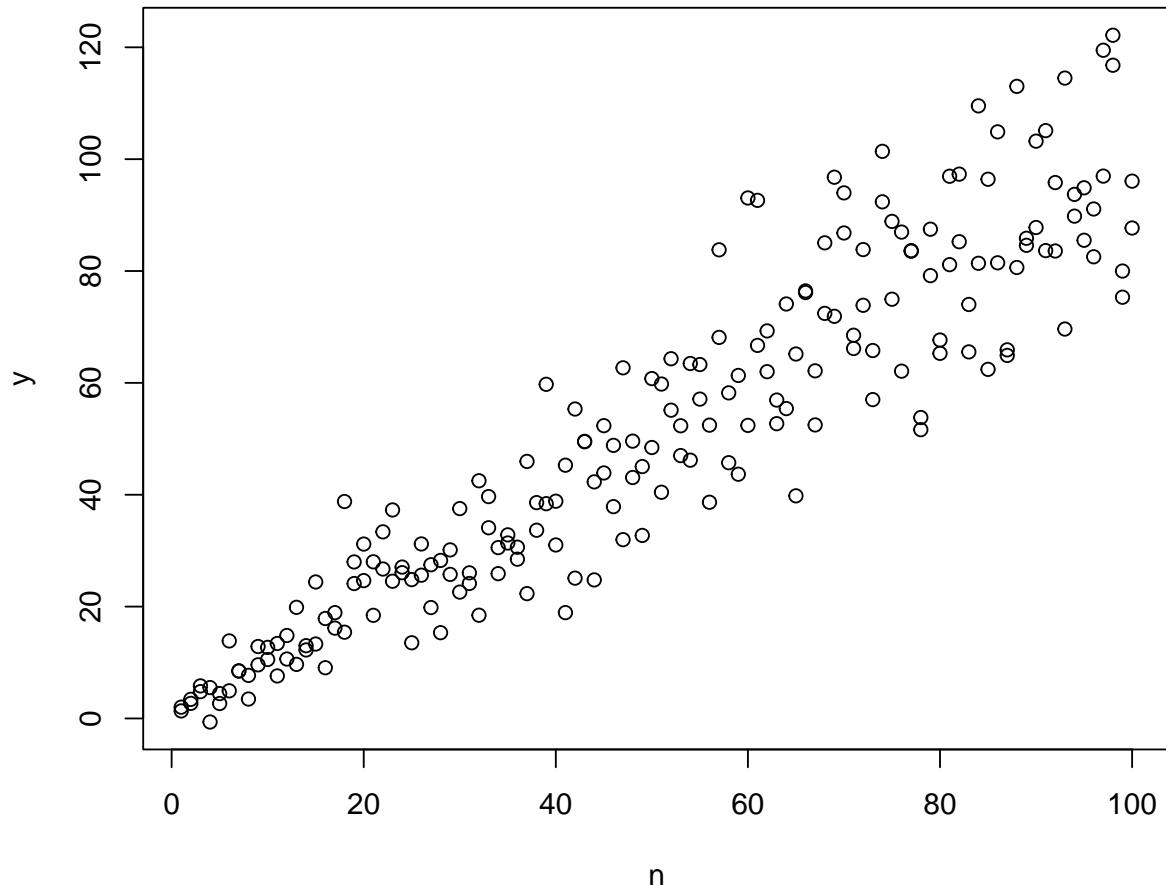


Figure 4.1: Illustration of heteroscedasticity

There could also be other factors involved in the relationship between the instrument and the criterion:

- curvilinearity
- an undetected mechanism, such as a moderator

Finally, what is our threshold for acceptability?

- Consider statistical significance – but also its limitations (e.g., power, Type I error, Type II error)
- Consider the magnitude of the correlation; and also R^2 (the proportion of variance accounted for)
- Consider error:
 - The standard error of the estimate shows the margin of error to be expected in the individuals predicted criterion score as the result of the imperfect validity of the instrument.

$$SE_{est} = SD_y \sqrt{1 - r_{xy}^2}$$

Where r_{xy}^2 is the square of the validity coefficient SD_y is the standard deviation of the criterion scores

If the validity were perfect ($r_{xy}^2 = 1.00$), the error of estimate would be 0.00. If the validity were zero, the error of estimate would equal SD_y .

Interpreting SE_{est}

If $r_{xy} = .80$, then $\sqrt{1 - r_{xy}^2} = .60$

Error is 60% as large as it would be by chance. Stated another way, predicting an individual's criterion performance has a margin of error that is 40% smaller than it would be by chance.

To obtain the SE_{est} , we merely multiply by the SD_y . This puts error in the metric of the criterion variable.

Your Turn If $r_{xy} = .25$, then $\sqrt{1 - r_{xy}^2} = ??$

Make a statement about chance. Make a statement about margin of error.

4.6 Practice Problems

In each of these lessons, I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. With each of these options, I encourage you to interpret examine aspects of the construct validity through the creation and interpretation of validity coefficients. Ideally, you will examine both convergent/discriminant validity as well as incremental validity.

4.6.1 Problem #1: Play around with this simulation.

Copy the script for the simulation and then change (at least) one thing in the simulation to see how it impacts the results.

If calculating is new to you, perhaps you just change the number in “set.seed(210907)” from 210907 to something else. Your results should parallel those obtained in the lecture, making it easier for you to check your work as you go.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Create a correlation matrix that includes the instrument-of-interest and the variables that will have varying degrees of relation	5	_____
3. Interpret the validity coefficients	5	_____
4. With at least three variables, evaluate the degree to which the instrument demonstrates incremental validity (this should involve two regression equations and their statistical comparison)	5	_____
5. Explanation to grader	5	_____
Totals	25	_____

4.6.2 Problem #2: Conduct the reliability analysis selecting different variables.

The Szymanski and Bissonette (?) article conducted a handful of incremental validity assessments. Select different outcome variables (e.g., depression) and/or use the subscales as the instrument-of-interest.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Create a correlation matrix that includes the instrument-of-interest and the variables that will have varying degrees of relation	5	_____
3. Interpret the validity coefficients	5	_____
4. With at least three variables, evaluate the degree to which the instrument demonstrates incremental validity (this should involve two regression equations and their statistical comparison)	5	_____
5. Explanation to grader	5	_____
Totals	25	_____
5. Explanation to grader	5	_____
Totals	25	_____

4.6.3 Problem #3: Try something entirely new.

Using data for which you have permission and access (e.g., IRB approved data you have collected or from your lab; data you simulate from a published article; data from an open science repository; data from other chapters in this OER), create validity coefficients and use three variables to estimate the incremental validity of the instrument-of-interest.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Create a correlation matrix that includes the instrument-of-interest and the variables that will have varying degrees of relation	5	_____
3. Interpret the validity coefficients	5	_____
4. With at least three variables, evaluate the degree to which the instrument demonstrates incremental validity (this should involve two regression equations and their statistical comparison)	5	_____
5. Explanation to grader	5	_____
Totals	25	_____

Chapter 5

Reliability

Screencasted Lecture Link

The focus of this lecture is the assessment of reliability. We start by defining *classical test theory* and examining several forms of reliability. While the majority of our time is spent considering estimates of internal consistency, we also examine retest reliability and interrater reliability.

5.1 Navigating this Lesson

There is one hour and twenty minutes of lecture. If you work through the materials with me it would be plan for an additional hour.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's introduction

5.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Define “reliability”
- Identify broad classes of reliability
- Interpret reliability coefficients
- Describe the strengths and limitations of the alpha coefficient

5.1.2 Planning for Practice

In each of these lessons, I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. The practice problems are the start of a larger project that spans multiple lessons. Therefore, if possible, please use a dataset that has item-level data for which there is a theorized total scale score as well as two or more subscales. With each of these options I encourage you to:

- Format (i.e., rescore, if necessary) a dataset so that it is possible to calculate estimates of internal consistency
- Calculate and report the alpha coefficient for a total scale scores and subscales (if the scale has them)
- Calculate and report ω_t and ω_h . With these two determine what proportion of the variance is due to all the factors, error, and g .
- Calculate total and subscale scores.
- Describe other reliability estimates that would be appropriate for the measure you are evaluating.

Again, I encourage you to use data that allows for the possibility of a total scale score as well as two or more subscales. This will allow you to continue using it in some of the lessons that follow.

5.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (and linked, when possible) in the text with complete citations in the reference list.

- Jhangiani, R. S., Chiang, I.-C. A., Cuttler, C., & Leighton, D. C. (2019). Reliability and Validity. In *Research Methods in Psychology*. <https://doi.org/10.17605/OSF.IO/HF7DQ>
- Revelle, W., & Condon, D. M. (2019a). Reliability from to : A tutorial. Psychological Assessment. <https://doi.org/10.1037/pas0000754>
 - A full-text preprint is available [here](#).
- Revelle, W., & Condon, D. M. (2019b). Reliability from to : A tutorial. Online supplement. Psychological Assessment. <https://doi.org/10.1037/pas0000754>
- Revelle, William. (n.d.). Reliability. In An introduction to psychometric theory with applications in R. Retrieved from <http://www.personality-project.org/dev/r/book/#chapter7>
 - All three documents provide a practical integration of conceptual and mechanical.
- Szymanski, D. M., & Bissonette, D. (2020). Perceptions of the LGBTQ College Campus Climate Scale: Development and psychometric evaluation. *Journal of Homosexuality*, 67(10), 1412–1428. <https://doi.org/10.1080/00918369.2019.1591788>
 - The research vignette for this lesson.

5.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
if(!require(psych)){install.packages("psych")}
if(!require(tidyverse)){install.packages("tidyverse")}
```

```
Loading required package: tidyverse

-- Attaching packages ----- tidyverse 1.3.1 --

v ggplot2 3.3.5     v purrr   0.3.4
v tibble   3.1.4     v dplyr    1.0.7
v tidyr    1.1.3     v stringr  1.4.0
v readr    2.0.1     v forcats 0.5.1

-- Conflicts ----- tidyverse_conflicts() --
x ggplot2:::+%()  masks psych:::+%()
x ggplot2::alpha() masks psych::alpha()
x dplyr::filter()  masks stats::filter()
x dplyr::lag()     masks stats::lag()

if(!require(MASS)){install.packages("MASS")}
```

Loading required package: MASS

Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

select

```
if(!require(sjstats)){install.packages("sjstats")}
```

Loading required package: sjstats

Registered S3 methods overwritten by 'parameters':

method	from
as.double.parameters_kurtosis	datawizard
as.double.parameters_skewness	datawizard
as.double.parameters_smoothness	datawizard
as.numeric.parameters_kurtosis	datawizard
as.numeric.parameters_skewness	datawizard
as.numeric.parameters_smoothness	datawizard
print.parameters_distribution	datawizard
print.parameters_kurtosis	datawizard
print.parameters_skewness	datawizard
summary.parameters_kurtosis	datawizard
summary.parameters_skewness	datawizard

Attaching package: 'sjstats'

The following object is masked from 'package:psych':

phi

```
if(!require(apaTables)){install.packages("apaTables")}
```

Loading required package: apaTables

```
if(!require(qualtRics)){install.packages("qualtRics")}
```

Loading required package: qualtRics

5.2 Defining Reliability

5.2.1 Begins with Classical Test Theory (CTT)

CTT is based on Spearman's (1904) *true-score model* where:

- an observed score consists of two components – a true component and an error component
- $X = T + E$
 - X = the fallible, observed/manifest score, obtained under ideal or perfect conditions of measurement (these conditions never exist);
 - T = the true/latent score (that will likely remain unknown); and
 - E = random error
- In CTT, we assume that the traits measured are constant and the errors random.
 - Therefore, the mean of measurement errors for any individual (upon numerous repeated testings) would be ????.
- That said, in CTT, the true score would be equal to the mean of the observed scores over an indefinite number of repeated measures.
 - Caveat: this is based on the assumption that when individuals are repeatedly measured, their true scores remain unchanged.
- In classic test theory, true score can be estimated over multiple trials. However, if errors are systematically biased, the true score will remain unknown.

5.2.2 Why are we concerned with reliability? Error!

- Measurements are imperfect and every observation has some unknown amount of error associated with it. Two components in error:
 - **random/unsystematic**: varies in unpredictable and inconsistent ways upon repeated measurements; sources are unknown

- **systematic:** recurs upon repeated measurements reflecting situational or individual effects that, theoretically, could be specified.
- Correlations are attenuated from the true correlation if the observations contain error.
- Knowing the reliability of an instruments allows us to:
 - estimate the degree to which measured at one time and place with one instrument predict scores at another time and/or place and perhaps measured with a different instrument
 - estimate the consistency of scores
 - estimate “...the degree to which test scores are free from errors of measurement” (APA, 1985, p. 19)

Figure 7.1a in Revelle’s chapter illustrates the *attenuation* of the correlation between the variables p and q as a function of reliability.

- circles (latent variables) represent the *true score*
- observed/measured/manifest variables are represented by squares and each has an associated error; not illustrated are the *random* and *systematic* components of error
- a true score is composed of a measured variable and its error
- the relationship between the true scores would be stronger than the one between the measured variables
- moving to 7.1b, the correlation between LV p and the observed ‘’ can be estimated from the correlation of p' with a parallel test (this is the reliability piece)

Figure 7.2 in Revelle’s Chapter 7 (?) illustrates the conceptual effect of reliability on the estimation of a true score.

5.2.3 The Reliability Coefficient

The symbol for reliability, r_{xx} , sums up the big-picture definition that reliability is the correlation of a measure with itself. There are a number of ways to think about it:

- a “theoretical validity” of a measure because it refers to a relationship between observed scores and scores on a latent variable or construct,
- represents the fraction of an observed score variance that is not error,
- ranges from 0-1
 - 1, when all observed variance is due to true-score variance; there are no random errors,
 - 0, when all observed variance is due to random errors of measurement,
- represents the squared correlation between observed scores and true scores,
- the ratio between true-score variance and observed-score variance (for a formulaic rendition see (?)),

$$r_{xt}^2 = r_{xx} = \frac{\sigma_2^t}{\sigma_2^x}$$

where r_{xt}^2 is the proportion of variance between observed scores ($t + e$) and true scores (t); its square root is the correlation

r_{xx} is the reliability of a measure

σ_2^t is the variance of true scores

σ_2^x is the variance of observed scores

- The reliability coefficient is interpreted as the proportion of systematic variance in the observed score.
 - .8 means that 80% of the variance of the observed scores is systematic;
 - .2 (e.g., 1.00 - .8) is the proportion of variance due to random errors;
 - the reliability coefficient is population specific.

To restate the first portion of the formula: although reliability is expressed as a correlation between observed scores, it is also the ratio of reliable variance to total variance.

5.3 Research Vignette

The research vignette for this lesson is the development and psychometric evaluation of the Perceptions of the LGBTQ College Campus Climate Scale (?). The scale is six items with responses rated on a 7-point Likert scale ranging from 1 (*strongly disagree*) to 7 (*strongly agree*). Higher scores indicate more negative perceptions of the LGBTQ campus climate. Szymanski and Bissonette (?) have suggested that the psychometric evaluation supports using the scale in its entirety or as subscales composed of the following items:

- College response to LGBTQ students:
 - My university/college is cold and uncaring toward LGBTQ students. (cold)
 - My university/college is unresponsive to the needs of LGBTQ students. (unresponsive)
 - My university/college provides a supportive environment for LGBTQ students. [un]supportive; must be reverse-scored
- LGBTQ Stigma:
 - Negative attitudes toward LGBTQ persons are openly expressed on my university/college campus. (negative)
 - Heterosexism, homophobia, biphobia, transphobia, and cissexism are visible on my university/college campus. (heterosexism)
 - LGBTQ students are harassed on my university/college campus. (harassed)

A preprint of the article is available at ResearchGate. Below is the script for simulating item-level data from the factor loadings, means, and sample size presented in the published article.

```

set.seed(210827)
SzyT1 <- matrix(c(.88, .73, .73, -.07, -.02, .16, -.03, .10, -.04, .86, .76, .71), ncol=2) #prior code I replaced with above
rownames(SzyT1) <- c("cold", "unresponsive", "supportiveNR", "negative", "heterosexism", "harassment")
#rownames(Szyf2) <- paste("V", seq(1:6), sep=" ") #prior code I replaced with above
colnames(SzyT1) <- c("F1", "F2")
SzyCorMat <- SzyT1 %*% t(SzyT1) #create the correlation matrix
diag(SzyCorMat) <- 1
#SzyCorMat #prints the correlation matrix
SzyM <- c(2.31, 3.11, 2.40, 3.18, 4.44, 3.02) #item means
SzySD <- c(1.35, 1.46, 1.26, 1.60, 1.75, 1.50) #item standard deviations; turns out we won't need these
SzyCovMat <- SzySD %*% t(SzySD) * SzyCorMat #creates a covariance matrix from the correlation matrix
#SzyCovMat #displays the covariance matrix

dfSzyT1 <- as.data.frame(round(MASS::mvrnorm(n=646, mu = SzyM, Sigma = SzyCovMat, empirical = TRUE)))
dfSzyT1[dfSzyT1>7] <- 7 #restricts the upperbound of all variables to be 7 or less
dfSzyT1[dfSzyT1<1] <- 1 #restricts the lowerbound of all variable to be 1 or greater
#colMeans(dfSzyT1) #displays column means

library(tidyverse)
library(dplyr)
dfSzyT1 <- dfSzyT1 %>% dplyr::mutate(ID = row_number()) #add ID to each row
dfSzyT1 <- dfSzyT1 %>% dplyr::select(ID, everything()) #moving the ID number to the first column;
dfSzyT1 <- dfSzyT1 %>%
  dplyr::mutate(supportive = 8 - supportiveNR) #because the original scale had 1 reversed item
dfSzyT1 <- dfSzyT1 %>%
  dplyr::select(-supportiveNR)

```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```

#write the simulated data as a .csv
#write.table(dfSzyT1, file="dfSzyT1.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#dfSzyT1 <- read.csv ("dfSzyT1.csv", header = TRUE)

#to save the df as an .rds (think "R object") file on your computer; it should save in the same folder
#saveRDS(dfSzyT1, "dfSzyT1.rds")
#bring back the simulated dat from an .rds file
#sdfSzyT1 <- readRDS("dfSzyT1.rds")

psych::describe(dfSzyT1)

```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
ID	1	646	323.50	186.63	323.5	323.50	239.44	1	646	645	0.00
cold	2	646	2.42	1.17	2.0	2.34	1.48	1	6	5	0.49
unresponsive	3	646	3.17	1.43	3.0	3.13	1.48	1	7	6	0.20

negative	4	646	3.22	1.52	3.0	3.16	1.48	1	7	6	0.33
heterosexism	5	646	4.41	1.61	4.0	4.45	1.48	1	7	6	-0.20
harassed	6	646	3.07	1.42	3.0	3.02	1.48	1	7	6	0.33
supportive	7	646	5.52	1.14	6.0	5.57	1.48	2	7	5	-0.30
kurtosis se											
ID			-1.21	7.34							
cold			-0.52	0.05							
unresponsive			-0.65	0.06							
negative			-0.47	0.06							
heterosexism			-0.60	0.06							
harassed			-0.36	0.06							
supportive			-0.76	0.04							

If we look at the information about this particular scale, we recognize that the *supportive* item is scaled in the opposite direction of the rest of the items. That is, a higher score on *supportive* would indicate a positive perception of the campus climate for LGBTQ individuals, whereas higher scores on the remaining items indicate a more negative perception. Before moving forward, we must reverse score this item.

In doing this, I will briefly note that in this case I have given my variables one-word names that represent each item. Many researchers (including myself) will often give variable names that are alpha numerical: LGBTQ1, LGBTQ2, LGBTQn. Either is acceptable. In the psychometric case, the one-word names may be useful shortcuts as one begins to understand the inter-item relations.

In reverse-scoring the *supportive* item, I will rename it “*unsupportive*” as an indication of its reversed direction.

```
library(tidyverse)

dfSzyT1<- dfSzyT1 %>%
  dplyr::mutate(unsupported = 8 - supportive)#when reverse-coding, subtract the variable from 8

psych::describe(dfSzyT1)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
ID	1	646	323.50	186.63	323.5	323.50	239.44	1	646	645	0.00
cold	2	646	2.42	1.17	2.0	2.34	1.48	1	6	5	0.49
unresponsive	3	646	3.17	1.43	3.0	3.13	1.48	1	7	6	0.20
negative	4	646	3.22	1.52	3.0	3.16	1.48	1	7	6	0.33
heterosexism	5	646	4.41	1.61	4.0	4.45	1.48	1	7	6	-0.20
harassed	6	646	3.07	1.42	3.0	3.02	1.48	1	7	6	0.33
supportive	7	646	5.52	1.14	6.0	5.57	1.48	2	7	5	-0.30
unsupported	8	646	2.48	1.14	2.0	2.43	1.48	1	6	5	0.30
kurtosis se											
ID			-1.21	7.34							
cold			-0.52	0.05							
unresponsive			-0.65	0.06							
negative			-0.47	0.06							

```

heterosexism   -0.60 0.06
harassed       -0.36 0.06
supportive      -0.76 0.04
unsupportive    -0.76 0.04

```

Next, I will create dfs that each contain the items of the total and subscales. These will be useful in the reliability estimates that follow.

```

LGBTQT1 <- dplyr::select(dfSzyT1, cold, unresponsive, unsupportive, negative, heterosexism, ha
ResponseT1 <- dplyr::select(dfSzyT1, cold, unresponsive, unsupportive)
StigmaT1 <- dplyr::select(dfSzyT1, negative, heterosexism, harassed)

```

5.4 The Big Parade of Reliability Coefficients

While I cluster the reliability coefficients into large groups, please understand that these are somewhat overlapping.

Table 1 in Revelle and Condon's (?) article provides a summary of the type of reliability tested, the findings, and the function used in the *psych* package.

5.5 Reliability Options for a Single Administration

If reliability is defined as the correlation between a test and a test just like it, how do we estimate the reliability of a single test, given only one time (?)? It may help to keep in mind that reliability is the ratio of true score variance to test score variance (or 1 - the ratio of error variance). Thus, the goal is to estimate the amount of error variance in the test. In this case we can investigate:

- a correlation between two random parts of the test
- internal consistency
- the internal structure of the test

5.5.1 Split-half reliability

Split-half reliability is splitting a test into two random halves, correlating the two halves, and adjusting the correlation with the *Spearman-Brown* prophecy formula. Abundant formulaic detail in Revelle's Chapter 7/Reliability (?).

An important question to split-half is "How to split?" Revelle terms it a "combinatorially difficult problem." There are 126 possible splits for a 10 item scale, 6,345 possible splits for a 16 item scale, and over 4.5 billion for a 36 item scale! The *psych* package's *splitHalf()* function will try all possible splits for scales of up to 16 items, then sample 10,000 splits for scales longer than that.

```

split <- psych::splitHalf (LGBTQT1, raw = TRUE, brute = TRUE)
split #show the results of the analysis

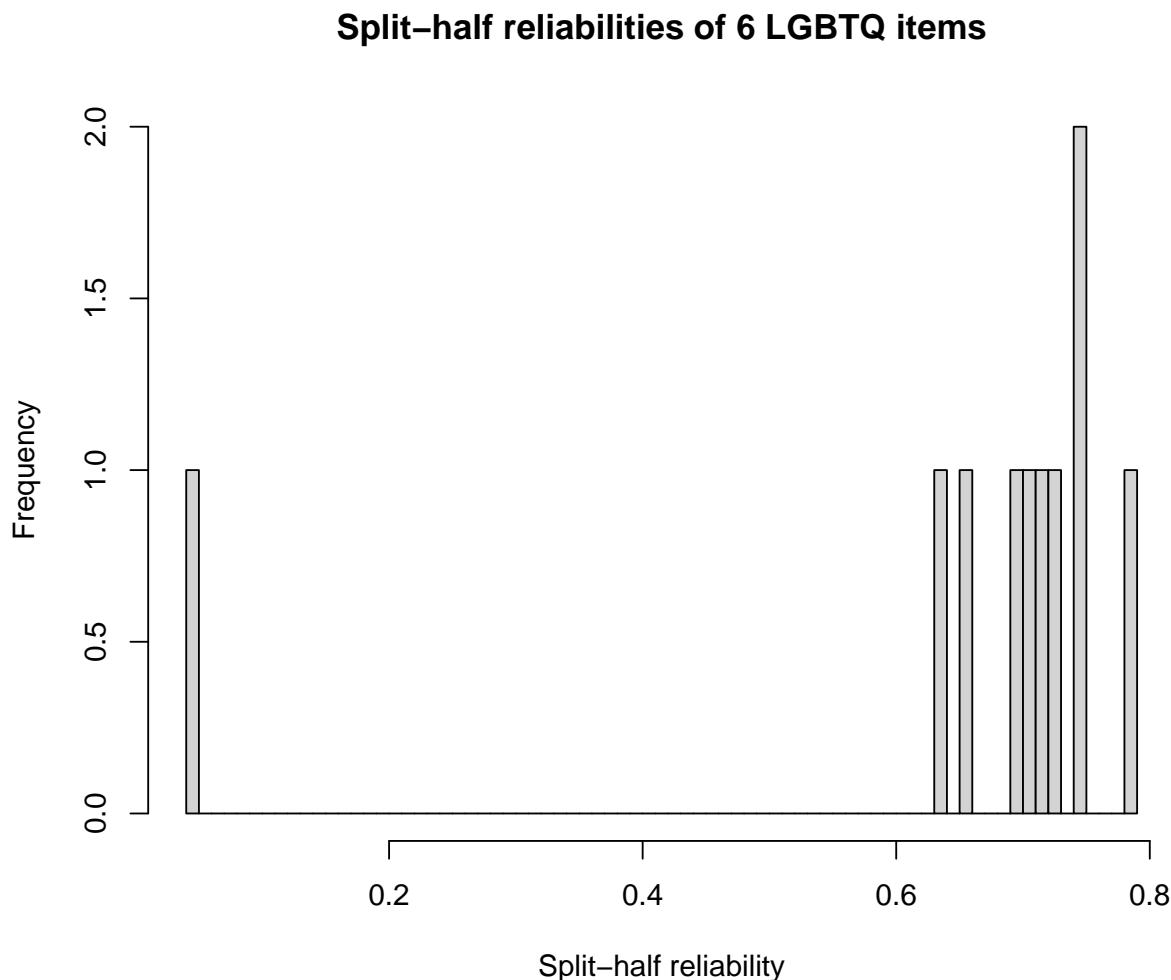
```

```
Split half reliabilities
```

```
Call: psych::splitHalf(r = LGBTQT1, raw = TRUE, brute = TRUE)
```

```
Maximum split half reliability (lambda 4) = 0.78
Guttman lambda 6                      = 0.73
Average split half reliability        = 0.64
Guttman lambda 3 (alpha)              = 0.64
Guttman lambda 2                      = 0.7
Minimum split half reliability (beta) = 0.04
Average interitem r = 0.23 with median = 0.09
                                         2.5% 50% 97.5%
Quantiles of split half reliability   = 0.17 0.71 0.78
```

```
hist(split$raw, breaks = 101, xlab = "Split-half reliability",
main = "Split-half reliabilities of 6 LGBTQ items")
```



Results of the split-half can provide some indication of whether not the scale is unidimensional.

In this case, the maximum reliability coefficient is .78, the average .64, and the lowest is .04. Similarly, we can examine the quantiles: .17, .71, .78.

The split-half output also includes the classic Cronbach's (1951) alpha coefficient (.64; aka Guttman lambda 3) and average interitem correlations (.24). The figure plots the frequencies of the reliability coefficient values.

While I did not find guidelines on what constitutes a “high enough lowerbound” to establish homogeneity, Revelle suggested that a scale with .85, .80, and .65 had “strong evidence for a relatively homogeneous scale.” When the values were .81, .73, .42, Revelle indicated that there was “strong evidence for non-homogeneity” (? , p. 11). In making this declaration, Revelle was also looking at the strength of the inter-item correlation and for a rather tight, bell-shaped, distribution at the higher ($> .73$) end of the figure. *We don't quite have that.*

What happens when we examine the split-half estimates of the subscales? With only three items, there's not much of a split and so the associated histogram will not be helpful.

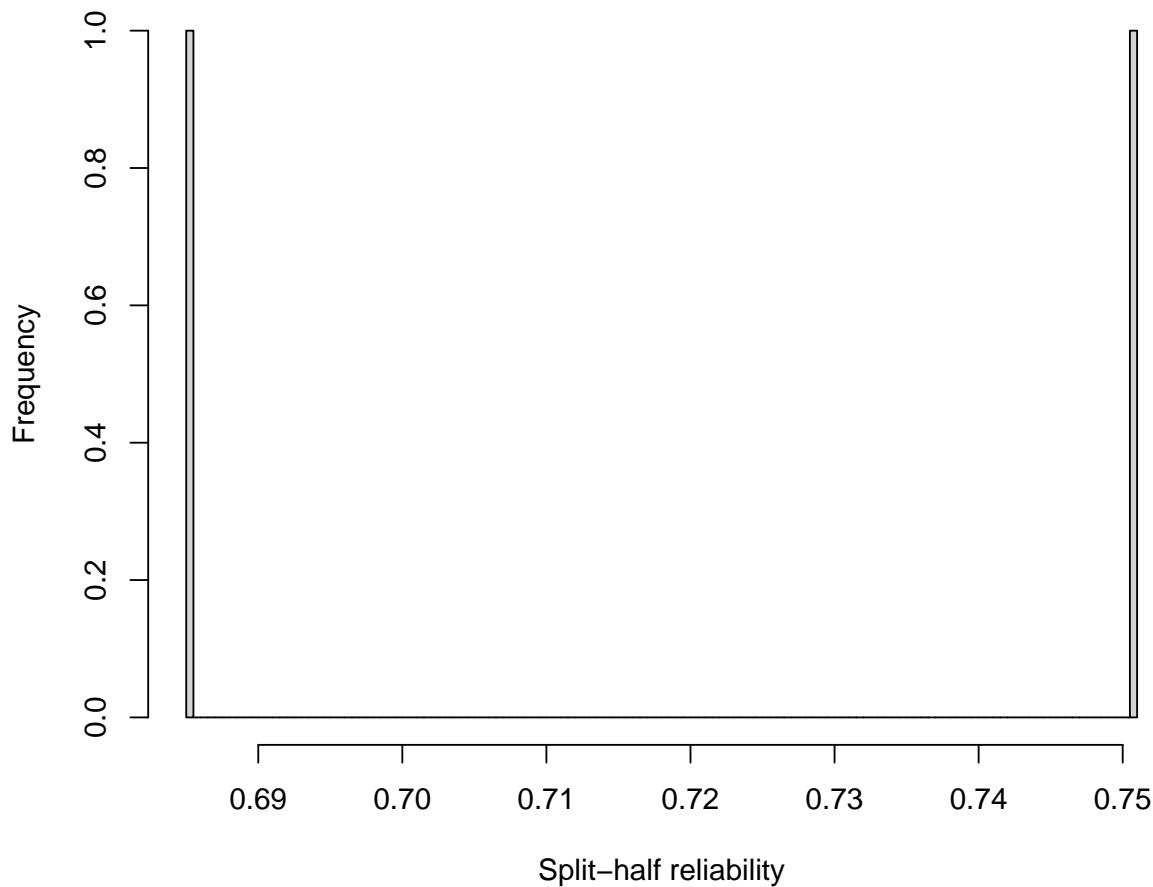
```
splitRx <- psych::splitHalf (ResponseT1, raw = TRUE, brute = TRUE)
splitRx #show the results of the analysis
```

```
Split half reliabilities
Call: psych::splitHalf(r = ResponseT1, raw = TRUE, brute = TRUE)

Maximum split half reliability (lambda 4) = 0.75
Guttman lambda 6 = 0.72
Average split half reliability = 0.96
Guttman lambda 3 (alpha) = 0.79
Guttman lambda 2 = 0.79
Minimum split half reliability (beta) = 0.69
Average interitem r = 0.56 with median = 0.58
                                         2.5% 50% 97.5%
Quantiles of split half reliability = 0.69 0.72 0.75
```

```
hist(splitRx$raw,breaks = 101, xlab = "Split-half reliability",
main = "Split-half reliabilities of 3 items of the College Response subscale")
```

Split-half reliabilities of 3 items of the College Response subscale



The alpha is higher – .79 The range of splits for max, ave, and low are .75, .96, and .69 and the quantiles are 0.69 0.72 0.75. The inter-item correlations have an average of .57.

Let's look at the split-half reliabilities for the Stigma subscale.

```
splitSt <- psych::splitHalf (StigmaT1, raw = TRUE, brute = TRUE)
splitSt #show the results of the analysis
```

```
Split half reliabilities
Call: psych::splitHalf(r = StigmaT1, raw = TRUE, brute = TRUE)

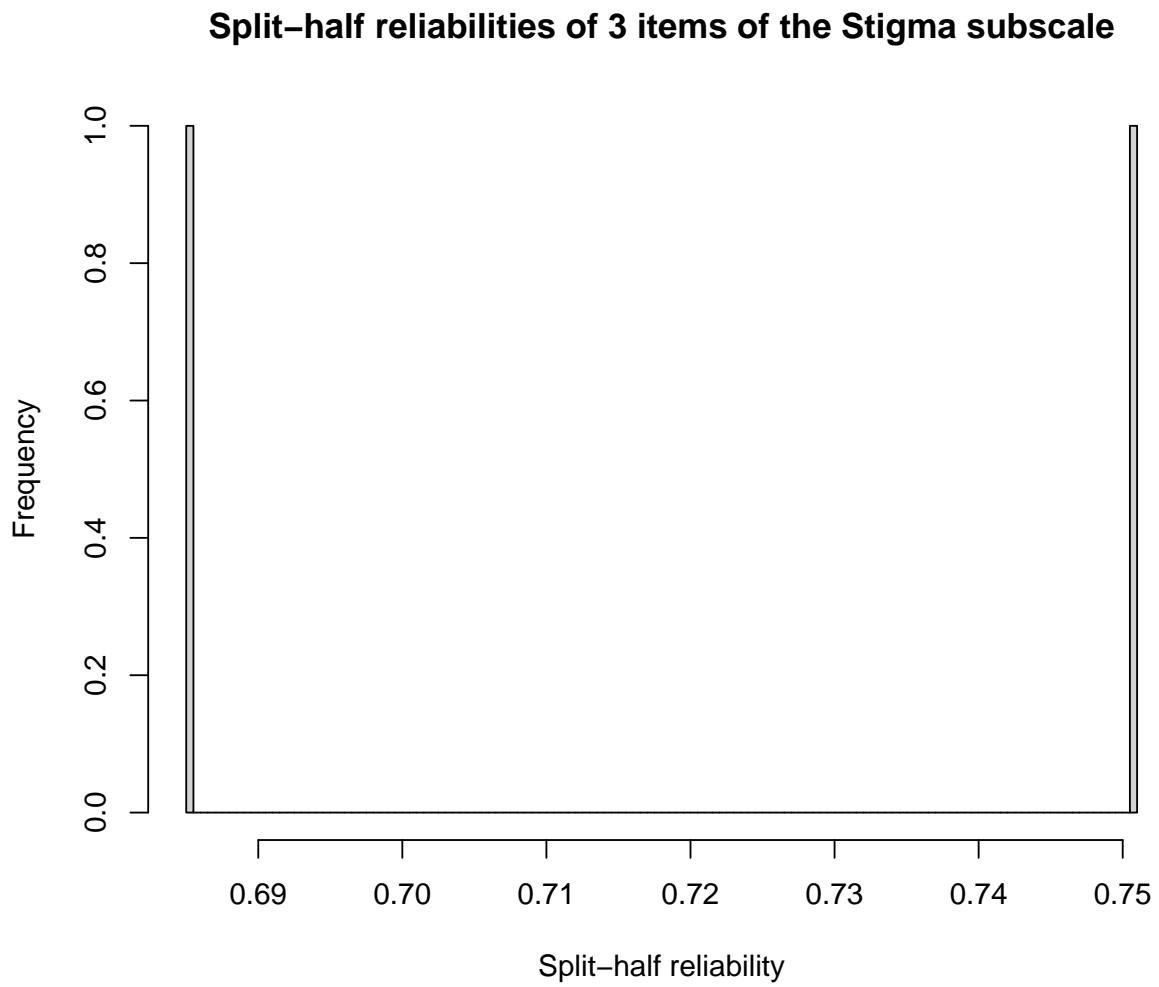
Maximum split half reliability (lambda 4) =  0.75
Guttman lambda 6                      =  0.72
Average split half reliability        =  0.96
Guttman lambda 3 (alpha)              =  0.79
Guttman lambda 2                      =  0.79
Minimum split half reliability (beta) =  0.7
Average interitem r =  0.56  with median =  0.57
```

```

Quantiles of split half reliability      =  2.5% 50% 97.5%
                                         =  0.7  0.72 0.75

hist(splitRx$raw, breaks = 101, xlab = "Split-half reliability",
main = "Split-half reliabilities of 3 items of the Stigma subscale")

```



The maximum, average, and minimum split-half reliabilities were .74, .96, and .70; quantiles were at .70, .72, and .74. The average interitem correlation was .56.

Because the alpha coefficient can be defined as the “average of all possible split-half coefficients” for the groups tested, it is common for researchers to not provide split-half results in their papers – this is true for our research vignette. I continue to teach the split-half because it can be a stepping stone in the conceptualization of internal consistency as an estimate of reliability.

5.5.2 From alpha

The most common methods to assess internal consistency are the *KR20* (for dichotomous items) and α (for Likert scaling); alpha has an alias, λ_3 (the Guttman lambda 3).

Alpha and the Guttman 3 (used for scales with Likert-type scaling) may be thought of as:

- a function of the number of items and the average correlation between the items
- the correlation of a test with a non-existent test just like it
- average of all possible split-half coefficients for the groups tested

Although the *psych* package has an incredible and thorough *alpha()* function, Revelle is not a fan of alpha. In fact, his alpha function reports a 95% CI around alpha as well as bootstrapped alpha results.

Let's grab alpha coefficients for our total and subscales.

```
psych::alpha (LGBTQT1)
```

Reliability analysis

Call: psych::alpha(x = LGBTQT1)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.64	0.64	0.73	0.23	1.8	0.023	3.1	0.83	0.089

lower alpha	upper	95% confidence boundaries			
0.6	0.64	0.69			

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
cold	0.62	0.60	0.66	0.23	1.5	0.024	0.075	0.111	
unresponsive	0.60	0.58	0.67	0.22	1.4	0.024	0.094	0.076	
unsupportive	0.63	0.62	0.70	0.25	1.6	0.023	0.083	0.124	
negative	0.59	0.62	0.67	0.24	1.6	0.026	0.070	0.124	
heterosexism	0.60	0.61	0.69	0.24	1.6	0.026	0.081	0.124	
harassed	0.55	0.57	0.68	0.21	1.3	0.030	0.100	0.033	

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
cold	646	0.52	0.60	0.53	0.32	2.4	1.2
unresponsive	646	0.60	0.64	0.56	0.37	3.2	1.4
unsupportedive	646	0.48	0.56	0.45	0.28	2.5	1.1
negative	646	0.63	0.57	0.50	0.39	3.2	1.5
heterosexism	646	0.64	0.58	0.48	0.39	4.4	1.6
harassed	646	0.70	0.66	0.57	0.50	3.1	1.4

Non missing response frequency for each item

	1	2	3	4	5	6	7	miss
cold	0.26	0.29	0.25	0.14	0.04	0.00	0.00	0
unresponsive	0.15	0.20	0.24	0.23	0.13	0.04	0.01	0
unsupportedive	0.24	0.28	0.28	0.16	0.04	0.00	0.00	0
negative	0.15	0.19	0.24	0.22	0.12	0.05	0.02	0

heterosexism	0.05	0.08	0.14	0.25	0.22	0.14	0.12	0
harassed		0.16	0.21	0.24	0.24	0.11	0.02	0.02

The second screen of output shows the information we are interested in:

- **raw_alpha**, .64 is based on the covariances
- **std.alpha**, .64 is based on correlations
- **average_r**, .24 is the average inter-item correlation (i.e., all possible pairwise combinations of items)

```
psych::alpha(ResponseT1)
```

Reliability analysis

Call: psych::alpha(x = ResponseT1)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r	
0.78	0.79	0.72		0.56	3.8	0.015	2.7	1	0.58

lower alpha	upper	95% confidence boundaries
0.75	0.78	0.81

Reliability if an item is dropped:

raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
cold	0.64	0.65	0.48	0.48	1.8	0.028	NA	0.48
unresponsive	0.74	0.74	0.58	0.58	2.8	0.021	NA	0.58
unsupportive	0.75	0.76	0.61	0.61	3.1	0.019	NA	0.61

Item statistics

n	raw.r	std.r	r.cor	r.drop	mean	sd	
cold	646	0.86	0.87	0.78	0.69	2.4	1.2
unresponsive	646	0.86	0.83	0.69	0.61	3.2	1.4
unsupportive	646	0.80	0.82	0.67	0.59	2.5	1.1

Non missing response frequency for each item

1	2	3	4	5	6	7	miss	
cold	0.26	0.29	0.25	0.14	0.04	0.00	0.00	0
unresponsive	0.15	0.20	0.24	0.23	0.13	0.04	0.01	0
unsupportive	0.24	0.28	0.28	0.16	0.04	0.00	0.00	0

In the case of the College Response subscale:

- **raw_alpha**, .79 is based on the covariances
- **std.alpha**, .80 is based on correlations
- **average_r**, .57 is the average interitem correlation

```
psych::alpha(StigmaT1)
```

Reliability analysis

Call: psych::alpha(x = StigmaT1)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.79	0.79	0.72		0.56	3.8	0.014	3.6	1.3
								0.57

lower	alpha	upper	95% confidence boundaries			
0.76	0.79	0.82				

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
negative	0.66	0.66	0.49		0.49	1.9	0.027	NA	0.49
heterosexism	0.72	0.72	0.57		0.57	2.6	0.022	NA	0.57
harassed	0.76	0.76	0.62		0.62	3.2	0.019	NA	0.62

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
negative	646	0.87	0.87	0.77	0.69	3.2	1.5
heterosexism	646	0.85	0.84	0.71	0.63	4.4	1.6
harassed	646	0.80	0.82	0.66	0.59	3.1	1.4

Non missing response frequency for each item

	1	2	3	4	5	6	7	miss
negative	0.15	0.19	0.24	0.22	0.12	0.05	0.02	0
heterosexism	0.05	0.08	0.14	0.25	0.22	0.14	0.12	0
harassed	0.16	0.21	0.24	0.24	0.11	0.02	0.02	0

In the case of the Stigma subscale:

- **raw_alpha**, .79 is based on the covariances
- **std.alpha**, .79 is based on correlations
- **average_r**, .56 is the average interitem correlation

The documentation for this package is incredible. Scroll down near the bottom of the *alpha()* function to learn what these are.

Especially useful are item-level statistics:

- **r.drop** is the corrected item-total correlation ([in the next lesson](#)) for this item against the scale without this item ***mean** and **sd** are the mean and standard deviation of each item across all individuals.

But don't get too excited the popularity of alpha emerged when tools available for calculation were less sophisticated – alpha can be misleading:

- alpha inflates, somewhat artificially, even when inter-item correlations are low.
 - a 14-item scale will have an alpha of at least .70, even if it has two orthogonal (i.e., unrelated) scales (?)
- alpha assumes a unidimensional factor structure,
- the same alpha can be obtained for dramatically different underlying factor structures (see graphs in Revelle's Chapter 7)

The proper use of alpha requires the following:

- *tau equivalence*, that is, equal covariances with the latent score represented by the test, and
- *unidimensionality*, equal factor loadings on the single factor of the test

When either of these is violated, alpha underestimates reliability and overestimates the fraction of test variance that is associated with the general variance in the test.

It is curious that the subscale estimates are stronger than the total scale estimates. This early evidence supports the two-scale solution.

Alpha and the split halves are *internal consistency* estimates. Moving to *model-based* techniques allows us to take into consideration the factor structure of the scale. In the original article (?), results were as follows (note that the alphas are stronger than in our simulation):

Scale (<i>n</i>)	Alpha	Inter-item correlation range	Average inter-item correlation
Total (6)	.85	.27 to .66	.49
College Response (3)	.82	.56 to .67	.61
Stigma (3)	.83	.60 to .66	.63

In the article, we can see the boost that alpha gets (.85) when the number of items is double, even though the average inter-item correlation is lower (.49)

5.5.3 To Omega

Assessing reliability with the *omega* (ω) statistics falls into a larger realm of *composite reliability* where reliability is assessed from a ratio of the variability explained by the items compared with the total variance of the entire scale (?). Members of the omega family of reliability estimates come from factor exploratory (i.e., EFA) and confirmatory (i.e., CFA; structural equation modeling [SEM]) factor analytic approaches. This lesson precedes the lessons on CFA and SEM. Therefore, my explanations and demonstrations will be somewhat brief. I intend to revisit omega output in the CFA and SEM lessons and encourage you to review this section now, then return to this section again after learning more about CFA and SEM.

In the context of *psychometrics* it may be useful (albeit an oversimplification) to think of factors as scales/subscales where *g* refers to the amount of variance in the *general* factor (or total scale score) and subcales to be items that have something in common that is separate from what is *g*.

Model-based estimates examine the correlations or covariances of the items and decompose the test variance into that which is:

- common to all items (**g**, a general factor),
- specific to some items (**f**, orthogonal group factors), and
- unique to each item (confounding **s** specific, and **e** error variance)

ω is something of a shapeshifter. In the *psych* package:

- ω_t represents the total reliability of the test (ω_t)
 - In the *psych* package, this is calculated from a bifactor model where there is one general *g* factor (i.e., each item loads on the single general factor), one or more group factors (*f*), and an item-specific factor(*s*).
- ω_h extracts a higher-order factor from the correlation matrix of lower level factors, then applies the Schmid and Leiman (1957) transformation to find the general loadings on the original items. Stated another way, it is a measure of the general factor saturation (*g*; the amount of variance attributable to one comon factor). The subscript “*h*” acknowledges the hierarchical nature of the approach.
 - the ω_h approach is exploratory and defined if there are three or more group factors (with only two group factors, the default is to assume they are equally important, hence the factor loadings of those subscales will be equal)
 - Najera Catalan (?) suggests that ω_h is the best measure of reliability when dealing with multiple dimensions.
- ω_g is an estimate that uses a bifactor solution via the SEM package *lavaan* and tends to be a larger (because it forces all the cross loadings of lower level factors to be 0)
 - the ω_g is confirmatory, requiring the specification of which variables load on each group factor

Two commands in *psych* get us the results:

- *omega()* reports only the EFA solution
- *omegaSem()* reports both EFA and CFA solutions
 - We will use the *omegaSem()* function

Note that in our specification, we indicate there are two factors. We do not tell it (anywhere!) what items belong to what factors (think, *subsubscales*). One test will be to see if the items align with their respective factors.

```
psych::omegaSem(LGBTQT1, nfactors=2)
```

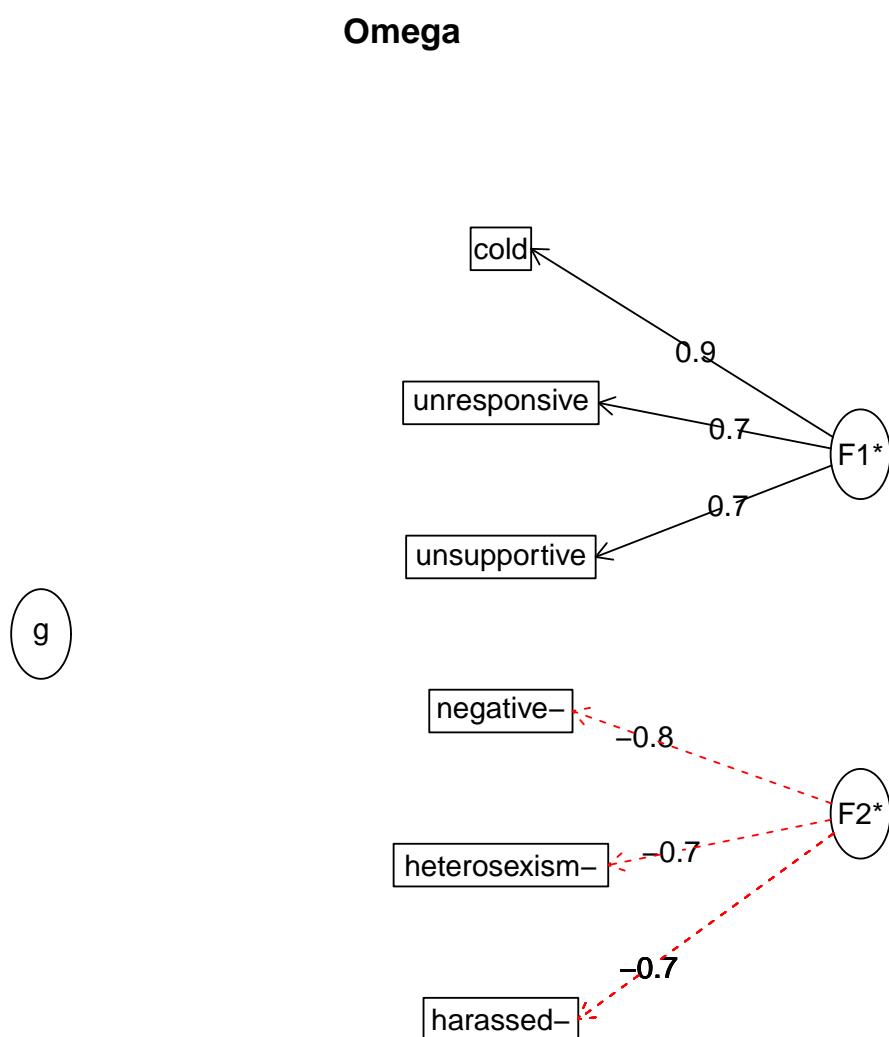
Loading required namespace: lavaan

Loading required namespace: GPArotation

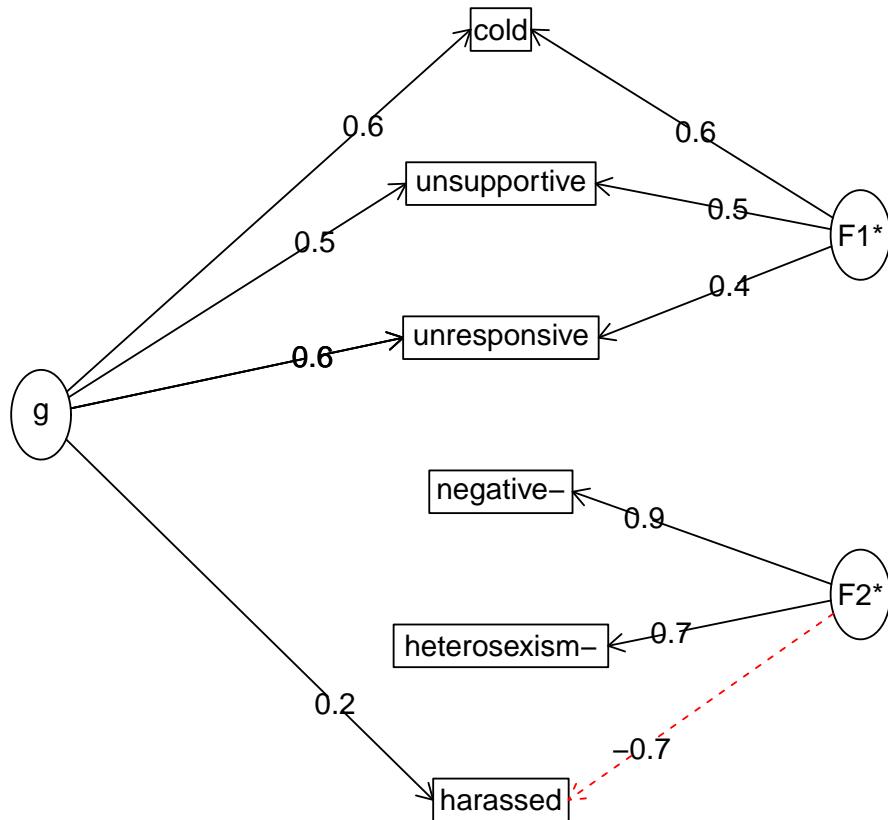
Three factors are required for identification -- general factor loadings set to be equal.
Proceed with caution.

Think about redoing the analysis with alternative values of the 'option' setting.

Warning in lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats, : lavaan WARNING
 Could not compute standard errors! The information matrix could
 not be inverted. This may be a symptom that the model is not
 identified.



Omega from SEM



```

Call: psych::omegaSem(m = LGBTQT1, nfactors = 2)
Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
  digits = digits, title = title, sl = sl, labels = labels,
  plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
  covar = covar)

```

Alpha:	0.62
G.6:	0.72
Omega Hierarchical:	0.01
Omega H asymptotic:	0.01
Omega Total	0.79

Schmid Leiman Factor loadings greater than 0.2

	g	F1*	F2*	h2	u2	p2
cold		0.85		0.73	0.27	0.01
unresponsive		0.71		0.52	0.48	0.01

unsupportive	0.68	0.46	0.54	0.01
negative-	-0.84	0.72	0.28	0.01
heterosexism-	-0.73	0.53	0.47	0.01
harassed-	-0.68	0.48	0.52	0.00

With eigenvalues of:

g	F1*	F2*
0.02	1.71	1.71

general/max 0.01 max/min = 1
 mean percent general = 0.01 with sd = 0 and cv of 0.49
 Explained Common Variance of the general factor = 0.01

The degrees of freedom are 4 and the fit is 0
 The number of observations was 646 with Chi Square = 2.59 with prob < 0.63
 The root mean square of the residuals is 0.01
 The df corrected root mean square of the residuals is 0.01
 RMSEA index = 0 and the 10 % confidence intervals are 0 0.049
 BIC = -23.29

Compare this with the adequacy of just a general factor and no group factors
 The degrees of freedom for just the general factor are 9 and the fit is 1.87
 The number of observations was 646 with Chi Square = 1198.08 with prob < 0.0000000000000000
 The root mean square of the residuals is 0.36
 The df corrected root mean square of the residuals is 0.46

RMSEA index = 0.452 and the 10 % confidence intervals are 0.431 0.474
 BIC = 1139.84

Measures of factor score adequacy

	g	F1*	F2*
Correlation of scores with factors	0.10	0.91	0.90
Multiple R square of scores with factors	0.01	0.82	0.82
Minimum correlation of factor score estimates	-0.98	0.64	0.63

Total, General and Subset omega for each subset

	g	F1*	F2*
Omega total for total scores and subscales	0.79	0.80	0.80
Omega general for total scores and subscales	0.01	0.01	0.01
Omega group for total scores and subscales	0.81	0.79	0.79

The following analyses were done using the lavaan package

Omega Hierarchical from a confirmatory model using sem = 0.37
 Omega Total from a confirmatory model using sem = 0.73
 With loadings of

	g	F1*	F2*	h2	u2	p2
cold	0.57	0.65		0.74	0.26	0.44

unresponsive	0.58	0.43	0.52	0.48	0.65
unsupportive	0.46	0.50	0.46	0.54	0.46
negative-			0.85	0.73	0.27 0.01
heterosexism-			0.72	0.52	0.48 0.00
harassed	0.21		-0.68	0.51	0.49 0.09

With sum of squared loadings of:

g	F1*	F2*
0.91	0.86	1.71

The degrees of freedom of the confirmatory model are 3 and the fit is 12.09965 with p = 0
 general/max 0.53 max/min = 2
 mean percent general = 0.27 with sd = 0.28 and cv of 1.01
 Explained Common Variance of the general factor = 0.26

Measures of factor score adequacy

	g	F1*	F2*
Correlation of scores with factors	0.67	0.68	0.92
Multiple R square of scores with factors	0.45	0.46	0.84
Minimum correlation of factor score estimates	-0.10	-0.08	0.68

Total, General and Subset omega for each subset

	g	F1*	F2*
Omega total for total scores and subscales	0.73	0.80	0.42
Omega general for total scores and subscales	0.37	0.40	0.04
Omega group for total scores and subscales	0.35	0.39	0.38

To get the standard sem fit statistics, ask for summary on the fitted object

There's a ton of output! How do we make sense of it?

First, our items aligned perfectly with their respective factors (subscales). That is, it would be problematic if the items switched factors.

Second, we can interpret our results. Like alpha, the omegas range from 0 to 1, where values closer to 1 represent good reliability (?). For unidimensional measures, * ω_t values above 0.80 seem to be an indicator of good reliability. For multidimensional measures with well-defined dimensions, we strive for ω_h values above 0.65 (and $\omega_t > 0.8$). These recommendations are based on a Monte Carlo study that examined a host of reliability indicators and how their values corresponded with accurate predictions of poverty status. With this in mind, let's examine the output related to our simulated research vignette.

Let's start with the output in the lower portion where the values are "from a confirmatory model using sem."

Omega is a reliability estimate for factor analysis that represents the proportion of variance in the LGBTQ scale attributable to common variance rather than error. The omega for the total reliability of the test (ω_t ; which included the general factors and the subscale factors) was .72, meaning that 72% of the variance in the total scale is due to the factors and 28% (100% - 72%) is attributable to error.

Omega hierarchical (ω_h) estimates are the proportion of variance in the LGBTQ score attributable to the general factor, which in effect treats the subscales as error. ω_h for the the LGBTQ total scale was .40. A quick calculation with ω_h (.37) and ω_t (.72; $.40/.72 = .56$) lets us know that that 56% of the reliable variance in the LGBTQ total scale is attributable to the general factor.

.4/.72

[1] 0.555556

Amongst the output is the Cronbach's alpha coefficient (.66). Szymanski and Bissonette (?) did not report omega results; this may be because there were only two subfactors and/or they did not feel like a bifactor analysis would be appropriate.

5.5.4 Some summary statements about reliability from single administrations

- With the exception of the worst split-half reliability and ω_g or ω_h , all of the reliability estimates are functions of test length and will tend asymptotically towards 1 as the number of items increases
- the omega output provides a great deal more information about reliability than a simple alpha
 - Figure 7.5 in Revelle's chapter shows four different structural representations of measures that have equal alphas (all .72)
- $\omega_{(h)}$, β , and the worst split-half reliability are estimates of the amount of general factor variance in the test scores
- in the case of low general factor saturation, the EFA based $\omega_{(h)}$ is positively biased, so the CFA-based estimate, $\omega_{(g)}$, should be used
- $\omega_{(t)}$ is the model-based estimate of the greatest lower bound of the total reliability of the test; so is the best split-half reliability

Revelle and Condon's (?) recommendations to researchers:

- report at least two coefficients (e.g., $\omega_{(h)}$ and $\omega_{(t)}$) and discuss why each is appropriate for the inference that is being made,
- report more than "just alpha" unless you can demonstrate that the measure is tau equivalent and unidimensional

5.6 Reliability Options for Two or more Administrations

5.6.1 Test-retest of total scores

The purpose of test-retest reliability is to understand the stability of the measure over time. With two time points, T1 and T2, the test-retest correlation is an unknown mixture of trait, state, and specific variance, and is a function of the length of time between two measures.

- With two time points we cannot distinguish between trait and state effects, that said

- we would expect a high degree of stability if the retest is (relatively) immediate
- With three time points we can leverage some SEM tools to distinguish between trait and state components
- A large test-retest correlation over a long period of time indicates temporal stability;
 - expected if we are assessing something trait like (e.g., cognitive ability, personality trait)
 - not expected if we are assessing something state like (e.g., emotional state, mood)
 - not expected if there was an intervention (or condition) and the T1 and T2 administrations are part of a pre- and post-test design.

There are some *methodological* concerns about test-retest reliability. For example, owing to memory and learning effects, the average response time to a second administration of identical items is about 80% the time of the first administration.

Szymanski and Bissonette (?) did not assess retest reliability. We can, though, imagine how this might work. Let's imagine that both waves were taken in the same academic term, approximately two weeks apart.

With both sets of data we need to create scores for the total scale score and the two subscales. We would also need to join the two datasets into a single dataframe. We could do either first. I think I would create the scale scores in each df, separately.

In preparing this lesson, I considered several options. While I could (and actually did, but then deleted it) simulate item-level T2 data, I don't have an easy way to correlate it with the T1 data. The resulting test-retest is absurdly low. So, I will quickly demonstrate how you would score the item-level data for the total and subscale scores, then resimulate scale-level data that is correlated to demonstrate the retest reliability.

The code below presumes that you would have missing data in your raw dataset. Using an available information approach (AIA; (?)) where is common to allow 20-25% missingness, we might allow the total scale score to calculate if there is 1 variable missing; but none for the subscale scores.

```
LGBTQvars <- c('cold', 'unresponsive', 'negative', 'heterosexism', 'harassed', 'unsupported')
ResponseVars <- c('cold', 'unresponsive', 'unsupported')
Stigmavars <- c('negative', 'heterosexism', 'harassed')

dfSzyT1$TotalT1 <- sjstats::mean_n(dfSzyT1[,LGBTQvars], .80)#will create the mean for each individual
dfSzyT1$ResponseT1 <- sjstats::mean_n(dfSzyT1[,ResponseVars], .80)#will create the mean for each individual
dfSzyT1$StigmaT1 <- sjstats::mean_n(dfSzyT1[,Stigmavars], .80)#will create the mean for each individual
```

We would need to repeat this process with our retest (T2) data, save baby dfs with our scale and total scale scores, and then join them.

To demonstrate the retest reliability, I have taken a different path. In order for us to get sensible answers, I went ahead and simulated a new dataset with total and subscale scores for our variables for both waves. This next script is simply that simulation (i.e., you can skip over it).

```
SimCor_mu <- c(3.13, 2.68, 3.58, 3.16, 2.66, 2.76)
SimCor_sd <- c(0.82, 1.04, 1.26, 0.83, 1.05, .99)
simCor <- matrix (c(1, 0.64, 0.77, 0.44, 0.33, 0.29,
```

```

  0.64,   1,  0.53,  0.35,  0.46,  0.34,
  0.77,  0.53,   1,  0.27,  0.4,   0.47,
  0.44,  0.35,   0.27,   1,  0.63,  0.62,
  0.33,  0.46,   0.4,   0.63,   1,  0.57,
  0.29,  0.34,   0.47,  0.62,  0.57,   1),
  ncol = 6)
scovMat <- SimCor_sd %*% t(SimCor_sd)*simCor
set.seed(210829)
retest_df <- MASS::mvrnorm(n = 646, mu = SimCor_mu, Sigma = scovMat, empirical = TRUE)
colnames(retest_df) <- c("TotalT1", "ResponseT1", "StigmaT1", "TotalT2", "ResponseT2", "StigmaT2")
retest_df <- as.data.frame(retest_df) #converts to a df so we can use in R
library(dplyr)
retest_df <- retest_df %>% dplyr::mutate(ID = row_number()) #add ID to each row
retest_df <- retest_df %>% dplyr::select(ID, everything())#moving the ID number to the first column

```

Examining our df, we can see the ID variable and the three sets of scores for each wave of analysis. Now we simply ask for their correlations. There are a number of ways to do this – the *apaTables* package can do the calculations and pop it into a manuscript-ready table.

We won't want the ID variable to be in the table.

```

retest_df2 <- retest_df %>%
  dplyr::select (-ID)

apaTables::apa.cor.table(data = retest_df2, landscape=TRUE, table.number = 1, filename="Table_1.Rmd")

```

Table 1

Means, standard deviations, and correlations with confidence intervals

Variable	M	SD	1	2	3	4
1. TotalT1	3.13	0.82				
2. ResponseT1	2.68	1.04	.64**			
			[.59, .68]			
3. StigmaT1	3.58	1.26	.77**	.53**		
			[.74, .80]	[.47, .58]		
4. TotalT2	3.16	0.83	.44**	.35**	.27**	
			[.38, .50]	[.28, .42]	[.20, .34]	
5. ResponseT2	2.66	1.05	.33**	.46**	.40**	.63**
			[.26, .40]	[.40, .52]	[.33, .46]	[.58, .67]

6. StigmaT2	2.76	0.99	.29**	.34**	.47**	.62**
			[.22, .36]	[.27, .41]	[.41, .53]	[.57, .67]

5

.57**
[.52, .62]

Note. M and SD are used to represent mean and standard deviation, respectively. Values in square brackets indicate the 95% confidence interval. The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014). * indicates $p < .05$. ** indicates $p < .01$.

As expected in this simulation,

- the strongest correlations are within each scale at their respective time, that is to
 - the T1 variables correlate with each other;
 - the T2 variables correlate with each other.
- the next strongest correlations are with the same scale/subscale configuration across time, for example
 - TotalT1 with TotalT2
 - ResponseT1 with ResponseT2
 - StigmaT1 with StigmaT2
- the lowest correlations are different scales at T1 and T2
 - ResponseT1 with StigmaT2

5.6.2 Test Retest Recap

Here are some summary notions for retest reliability:

- increases in the interval will lower the reliability coefficient,
 - an experimental intervention that is designed to impact the retest assessment will lower the reliability coefficient,
 - state measures will have lower retest coefficients than trait measures,
 - and those all interact with each other

Note: there are numerous demonstrations in the Revelle and Condon (?; ?) materials (Table 1). In addition to the myriad of vignettes used to illustrate foci on state, trait, items, whole scale, etc., there were demos on duplicated items, assessing for consistency, and parallel/alternate forms.

If you are asking, “Hey, is parallel/alternate forms really a variant of test retest?” Great question! In fact, split-half could be seen as test-retest! Once you get in the weeds, the distinctions become less clear.

5.7 Interrater Reliability

5.7.1 Cohen's kappa

Cohen's kappa coefficient is used to calculate proportions of agreement corrected for chance. This type of analysis occurs in research designs where there is some kind of (usually) categorical designation of a response. I don't have a research vignette for this. In the past, I was involved in research where members of the research team coded counselor utterances according to Hill's *helping skills* system designed by Clara Hill (?). In the helping skills system, 15 different helping skills are divided into three larger groups that generally reflect the counseling trajectory: *exploration, insight, action*. One of our analyses divided counselor utterances into these categories. Let's look at a fabricated (not based on any real data) simulation where four raters each evaluated 12 counselor utterances (that represent the arch of a nonsensically speedy counseling session).

```
Rater1 <- c("exploration", "exploration", "exploration", "exploration", "exploration", "exploration")
Rater2 <- c("exploration", "exploration", "exploration", "insight", "exploration", "insight", "explorat
Rater3 <- c("exploration", "insight", "exploration", "exploration", "exploration", "exploration", "explorat
Rater4 <- c("exploration", "exploration", "exploration", "exploration", "exploration", "exploration")
ratings <- data.frame(Rater1, Rater2, Rater3, Rater4)
```

Historically, kappa could only be calculated for 2 raters at a time. Presently, though, it appears there can be any number of raters and the average agreement is reported.

Let's take a look at the data, then run the analysis, and interpret the results.

```
psych::cohen.kappa(ratings)
```

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Cohen Kappa (below the diagonal) and Weighted Kappa (above the diagonal)
For confidence intervals and detail print with all=TRUE

	Rater1	Rater2	Rater3	Rater4
Rater1	1.00	0.40	0.21	0.62
Rater2	0.14	1.00	0.00	0.57
Rater3	0.48	0.00	1.00	0.30
Rater4	0.54	0.45	0.43	1.00

Average Cohen kappa for all raters 0.34

Average weighted kappa for all raters 0.35

Kappa can range from -1.00 to 1.00.

- K = .00 indicates that the observed agreement is exactly equal to the agreement that could be observed by chance.
- Negative kappa indicates that observed kappa is less than the expected chance agreement.
- K = 1.00 equals perfect agreement between judges.

On using kappa:

- research teams set a standard (maybe .85) and “train up” until kappa is achieved
 - then periodically reassess and retrain
- really difficult to obtain an adequate kappa level when the number of categories achieve
 - example is Hill’s *Helping Skills System* when all 15 categories (not just the big three) are used
- really difficult to obtain an adequate kappa when *infrequent* categories (e.g., “insight”) exist

Our kappa of .35 indicates that this rating team has a 35% chance of agreement, corrected for by chance. This is substantially below the standard. Let’s imagine that the team spends time with their dictionaries, examines common errors, and makes some decision rules.

Here’s the resimulation...

```
Rater1b <- c("exploration", "exploration", "exploration", "exploration", "exploration", "exploration")
Rater2b <- c("exploration", "exploration", "exploration", "exploration", "exploration", "insight", "insight")
Rater3b <- c("exploration", "exploration", "exploration", "exploration", "exploration", "exploration", "exploration")
Rater4b <- c("exploration", "exploration", "exploration", "exploration", "exploration", "exploration", "exploration")
after_training <- data.frame(Rater1b, Rater2b, Rater3b, Rater4b)
```

Now run it again.

```
psych::cohen.kappa(after_training)
```

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in psych::cohen.kappa(after_training): No variance detected in cells 2 1

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in psych::cohen.kappa(after_training): No variance detected in cells 4 1

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

Warning in cohen.kappa1(x1, w = w, n.obs = n.obs, alpha = alpha, levels = levels): upper or lower confidence interval exceed abs(1) and set to +/- 1.

At least one item had no variance. Try describe(your.data) to find the problem.

Cohen Kappa (below the diagonal) and Weighted Kappa (above the diagonal)

For confidence intervals and detail print with all=TRUE

	Rater1b	Rater2b	Rater3b	Rater4b
Rater1b	1.00	0.83	0.55	0.80
Rater2b	0.73	1.00	0.36	0.60
Rater3b	0.72	0.45	1.00	0.46
Rater4b	0.71	0.43	0.70	1.00

Average Cohen kappa for all raters 0.62

Average weighted kappa for all raters 0.6

Hmmm. There was improvement, but this team needs more training!

5.7.2 Intraclass correlation (ICC)

Yes! This is the same ICC we used in multilevel modeling! The ICC is used when we have numerical ratings.

In our fabricated vignette below, five raters are evaluating the campus climate for LGBTQIA+ individuals for 10 units/departments on a college campus. Using the ICC can help us determine the degree of leniency and variability within judges.

Here's the resimulation (you can ignore this)...

```
Rater1 <- c(1, 1, 1, 4, 2, 3, 1, 3, 3, 5)
Rater2 <- c(1, 1, 2, 1, 4, 4, 4, 4, 5, 5)
Rater3 <- c(3, 3, 3, 2, 3, 3, 6, 4, 4, 5)
Rater4 <- c(3, 5, 4, 2, 3, 6, 6, 6, 5, 5)
Rater5 <- c(2, 3, 3, 3, 4, 4, 4, 4, 5, 5)
ICC_df <- data.frame(Rater1, Rater2, Rater3, Rater4, Rater5)
```

```
psych::ICC(ICC_df[1:10, 1:5], lmer = TRUE) #find the ICCs for the 10 campus units and 5 judges
```

```
Call: psych::ICC(x = ICC_df[1:10, 1:5], lmer = TRUE)
```

Intraclass correlation coefficients

	type	ICC	F	df1	df2	p	lower bound	upper bound
Single_raters_absolute	ICC1	0.34	3.5	9	40	0.00259	0.12	0.64
Single_random_raters	ICC2	0.37	5.4	9	36	0.00011	0.15	0.66
Single_fixed_raters	ICC3	0.47	5.4	9	36	0.00011	0.23	0.74
Average_raters_absolute	ICC1k	0.72	3.5	9	40	0.00259	0.40	0.90
Average_random_raters	ICC2k	0.74	5.4	9	36	0.00011	0.47	0.91
Average_fixed_raters	ICC3k	0.81	5.4	9	36	0.00011	0.60	0.93

```
Number of subjects = 10      Number of Judges = 5
```

In the output, reliability for a single judge ICC_1 is the ratio of person variance to total variance. Reliability for multiple judges ICC_{1k} adjusts the residual variance by the number of judges.

The ICC function reports six reliability coefficients: 3 for the case of single judges and 3 for the case of multiple judges. It also reports the results in terms of a traditional ANOVA as well as a mixed effects linear model, and CIs for each coefficient.

Like most correlation coefficients, the ICC ranges from 0 to 1.

- An ICC close to 1 indicates high similarity between values from the same group.
- An ICC close to zero means that values from the same group are not similar.

5.8 What do we do with these coefficients?

5.8.1 Corrections for attenuation

Circa 1904, Spearman created the reliability coefficient out of a need to adjust observed correlations between related constructs for the error of measurement in each construct. This is only appropriate if the measure is seen as the expected value of a single underlying construct. However, “under the hood,” SEM programs model the pattern of observed correlations in terms of a measurement (reliability) model as well as a structural (validity) model.

5.8.2 Predicting true scores (and their CIs)

True scores remain unknown and so the reliability coefficient is used in a couple of ways to estimate the true score (and the CI around that true score).

Take a quick look at the formula for predicting a true score and observe that the reliability coefficient is used within. It generally serves to nudge the observed score a bit closer to the mean: $T' = (1 - r_{xx})\bar{X} + r_{xx}X$

The CI around that true score includes some estimate of standard error: $CI_{95} = T' + / - z_{cv}(s_e)$

Whether that term is the standard error of estimate $s_e = s_x \sqrt{r_{xx}(1 - r_{xx})}$; standard deviation of predicted true scores for a given observed score),

OR, the standard error of measurement ($s_m = s_x \sqrt{(1 - r_{xx})}$; an estimate of the amount of variation to be expected in test scores; aka, the standard deviation of the errors of measurement),

the reliability coefficient is also a player.

I can hear you asking What is the difference between s_e and s_m ?

- Because r_{xx} is almost always a fraction, s_e is smaller than s_m
- When the reliability is high, the two standard errors are fairly similar to each other.
- Using s_m will result in wider confidence intervals.

5.8.3 How do I keep it all straight?

Table 1 in Revelle and Condon’s (?) article helps us connect the type of reliability we are seeking with the statistic(s) and the R function within the *psych* package.

5.9 Practice Problems

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. The practice problems are the start of a larger project that spans multiple lessons. Therefore, if possible, please use a dataset that has item-level data for which there is a theorized total scale score as well as two or more subscales. With each of these options I encourage you to:

- Format (i.e., rescore if necessary) a dataset so that it is possible to calculate estimates of internal consistency
- Calculate and report the alpha coefficient for a total scale scores and subscales (if the scale has them)
- Calculate and report ω_t and ω_h . With these two determine what proportion of the variance is due to all the factors, error, and g .
- Calculate total and subscale scores.
- Describe other reliability estimates that would be appropriate for the measure you are evaluating.

5.9.1 Problem #1: Play around with this simulation.

If evaluating internal consistency is new to you, copy the script for the simulation and then change (at least) one thing in the simulation to see how it impacts the results. Perhaps you just change the number in “set.seed(210827)” from 210827 to something else. Your results should parallel those obtained in the lecture, making it easier for you to check your work as you go.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Calculate and report the alpha coefficient for a total scale scores and subscales (if the scale has them)	5	_____
3. Calculate and report ω_t and ω_h . With these two determine what proportion of the variance is due to all the factors, error, and g .	5	_____
4. Calculate total and subscale scores.	5	_____
5. Describe other reliability estimates that would be appropriate for the measure you are evaluating.	5	_____
6. Explanation to grader	5	_____
Totals	30	_____

5.9.2 Problem #2: Use the data from the live ReCentering Psych Stats survey.

The script below pulls live data directly from the ReCentering Psych Stats survey on Qualtrics. As described in the [Scrubbing and Scoring chapters](#) of the ReCentering Psych Stats Multivariate Modeling volume, the Perceptions of the LGBTQ College Campus Climate Scale (?) was included (LGBTQ) and further adapted to assess perceptions of campus climate for Black students (BLst), non-Black students of color (nBSoC), international students (INTst), and students disabilities (wDIS). Consider conducting the analyses on one of these scales or merging them together.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Calculate and report the alpha coefficient for a total scale scores and subscales (if the scale has them)	5	_____

Assignment Component	Points Possible	Points Earned
3.Calculate and report ω_t and ω_h . With these two determine what proportion of the variance is due to all the factors, error, and g .	5	_____
4. Calculate total and subscale scores.	5	_____
5.Describe other reliability estimates that would be appropriate for the measure you are evaluating.	5	_____
6. Explanation to grader	5	_____
Totals	30	_____

```
library(tidyverse)
#only have to run this ONCE to draw from the same Qualtrics account...but will need to get diff
library(qualtRics)
#qualtrics_api_credentials(api_key = "mUgPMMySYkiWpMFkwHale1QE5HNmh5LRUaA8d9PDg",
#                           base_url = "spuppsych.az1.qualtrics.com", overwrite = TRUE, install = TRUE)
QTRX_df <- qualtRics::fetch_survey(surveyID = "SV_b2cClqAllGQ6nLU", time_zone = NULL, verbose = TRUE)
climate_df <- QTRX_df%>%
  select('Blst_1', 'Blst_2', 'Blst_3', 'Blst_4', 'Blst_5', 'Blst_6',
         'nBSoC_1', 'nBSoC_2', 'nBSoC_3', 'nBSoC_4', 'nBSoC_5', 'nBSoC_6',
         'INTst_1', 'INTst_2', 'INTst_3', 'INTst_4', 'INTst_5', 'INTst_6',
         'wDIS_1', 'wDIS_2', 'wDIS_3', 'wDIS_4', 'wDIS_5', 'wDIS_6',
         'LGBTQ_1', 'LGBTQ_2', 'LGBTQ_3', 'LGBTQ_4', 'LGBTQ_5', 'LGBTQ_6')
#Item numbers are supported with the following items:
#_1 "My campus unit provides a supportive environment for ___ students"
#_2 "_____ is visible in my campus unit"
#_3 "Negative attitudes toward persons who are ___ are openly expressed in my campus unit."
#_4 "My campus unit is unresponsive to the needs of ___ students."
#_5 "Students who are_____ are harassed in my campus unit."
#_6 "My campus unit is cold and uncaring toward ___ students."
#Item 1 on each subscale should be reverse coded.
#The College Response scale is composed of items 1, 4, 6,
#The Stigma scale is composed of items 2,3, 5
```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```
#write the simulated data as a .csv
#write.table(climate_df, file="climate_df.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#climate_df <- read.csv ("climate_df.csv", header = TRUE)
```

```
#to save the df as an .rds (think "R object") file on your computer; it should save in the sam
#saveRDS(climate_df, "climate_df.rds")
#bring back the simulated dat from an .rds file
#climate_df <- readRDS("climate_df.rds")
```

5.9.3 Problem #3: Try something entirely new.

Complete the same steps using data for which you have permission and access. This might be data of your own, from your lab, simulated from an article, or located on an open repository.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Calculate and report the alpha coefficient for a total scale scores and subscales (if the scale has them)	5	_____
3. Calculate and report ω_t and ω_h . With these two determine what proportion of the variance is due to all the factors, error, and g .	5	_____
4. Calculate total and subscale scores.	5	_____
5. Describe other reliability estimates that would be appropriate for the measure you are evaluating.	5	_____
6. Explanation to grader	5	_____
Totals	30	_____

Chapter 6

Item Analysis for Educational Achievement Tests (Exams)

[Screencasted Lecture Link](#)

In this lecture I walk through some procedures for analyzing the quality of multiple choice (including true/false) exam items. We look at item difficulty and item discrimination. We also look at item coverage as it relates to the learning objectives for an educational endeavor.

6.1 Navigating this Lesson

There is about one hour of lecture. If you work through the materials with me it would be plan for an additional 30 minutes.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

6.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Provide a rationale for why having a *test bank* might be a good idea.
- Describe the effects of skewness on the interpretation of exam results.
- Evaluate the the quality of a multiple choice item on the basis of item difficulty, correlation, and discrimination.
- Discuss the challenges of identifying an *ideal* difficulty level for test items. Further elaborate how guessing, speeded tests, interitem correlations, and the purposes of the test influence the *ideal difficulty*.

6.1.2 Planning for Practice

Practice suggestions for this lesson encourage you to think about the exams in your life: those you might be taking; those you might be writing or proctoring.

6.1.3 Readings & Resources

Classic psychometric texts tend to not cover item analysis for achievement tests and/or they skip over these fundamentals and move straight to item response theory/Rasch modeling (IRT). After scouring the internet, I landed on these two resources as concise, accessible, summaries.

- Understanding item analysis. Office of Educational Assessment, University of Washington. Retrieved September 20, 2019. Retrieved from <https://www.washington.edu/assessment/scanning-scoring/scoring/reports/item-analysis/>
 - It is common for excellent instructions/descriptions to accompany the scoring software used by institutions. UW appears to use ScorePak and this resource provides both conceptual and interpretive information.
- Revelle, W. (2017). An overview of the psych package. Retrieved from <http://personality-project.org/r/overview.pdf>
 - Pages 85-85 provide a vignette for conducting item analysis on multiple choice items.

6.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(psych)){install.packages("psych")}
```

6.2 Research Vignette

This lesson's research vignette is from my own class. Especially in the early years of my teaching, I gave high(er) stakes mid-term and final exams. There were usually 40 (or so) multiple choice or true/false items, 2-3 applied problems or short essays, and 1 longer essay. Today's vignette are an array of exam items from a statistics exam that demonstrate the desirable and undesirable elements we want in objective items.

6.3 Item Analysis in the Educational/Achievement Context

Multiple choice, true/false, and other *objectively* formatted/scored items are part-n-parcel to educational/achievement assessment. But how do we know if the items are performing the way they

should? This lecture focuses on item analysis in the context of multiple choice and true/false items. Using these practices can help you identify what selection of items you'd like for your exams. These can be critical tools in helping you improve your ability to assess student performance. In-so-doing, we walk through a bit of "what we used to do," to current common practices, to a glimpse of our future. We owe much of this to rapid advances in technology.

Test banks are instructor-created resources for developing/storing/protecting items for use in future exams. We create test banks when we carefully distribute/collect/protect items "that work" (from statistical perspective). Why would we want to do this?

- Once a test is "out" it's out. Instructors can presume that resourceful students are using it to study; yet all students won't have equal access to it.
- Developing "good" items takes a good deal of time; does the instructor want to redo this each term?
- Should we be piloting *all new items* on students each term and then having the debates about whether the item should be rescored?
 - Better is to introduce a proportion of new items each year and evaluate them for inclusion in the test bank; EPPP, SAT, GRE do this.
- A challenge is providing students appropriate study tools – old exams are favorites of students (but maybe there are other ways – worksheets, Jeopardy).

The conceptual portions of this lecture, particularly the interpretation of the difficulty and discrimination statistics are based in Anastasi's work (?)

6.3.1 And now a quiz! Please take it.

Let's start with some items from an early version of the exam I gave when I taught CPY7020/Statistical Methods.

Item 5 A grouping variable such as men or women that uses dummy coding of 1 and 0 to categorize the groups is an example of _____ scaling.

- a) Nominal
- b) Ordinal
- c) Interval
- d) Ratio

Item 11 The term "grade inflation" has frequently been applied to describe the distribution of grades in graduate school. Which of the following best describes this distribution.

- a) negatively skewed
- b) uniform/rectangular
- c) positively skewed and leptokurtic

- d) uniform and platykurtic

Item 19 All distributions of Z-scores will have the identical

- a) Mean
- b) Variance
- c) Standard deviation
- d) All of the above

Item 21 The most appropriate score for comparing scores across two or more distributions (e.g., exam scores in math and art classes) is the:

- a) mean
- b) percentile rank
- c) raw score
- d) z-score

Item 37 Of the following, what statement best describes $r^2 = .49$

- a) strong positive correlation
- b) strong positive or negative correlation
- c) weak positive or negative correlation
- d) weak negative correlation

Item 38 When there are no ties among ranks, what is the relationship between the Spearman rho (ρ) and the Pearson (r)?

- a) $\rho = r$
- ii) $\rho > r$
- a) $\rho < r$
- b) no relationship

6.4 Item Difficulty

6.4.1 Percent passing

Item difficulty index is the proportion of test takers who answer an item correctly. It is calculated by dividing the number of people who passed the item (e.g., 55) by the total number of people (e.g., 100).

- If 55% pass an item, we write $p = .55$
- The easier the item, the larger the percentage will be.

What is an ideal pass rate (and this “ideal” is the *statistical ideal* mostly for norm-referenced tests like the ACT, SAT, GRE)?

- The closer the difficulty of an item approaches 1.00 or 0, the less differential information about test takers it contributes.
 - If, out of 100 people, 50 pass an item and 50 fail ($p = .50$)...we have 50 X 50 or 2,500 paired comparisons or differential bits of information.
- How much information would we have for an item passed by:
 - 70% of the people ($70 * 30 = ???$)
 - 90% of the people ($90 * 10 = ???$)
- For maximum differentiation, one would choose all items at the .50 level (but hold up...)

6.4.2 Several factors prevent .50 from being the ideal difficulty level

Speeded tests complicate the interpretation of item difficulty because items are usually of equivalent difficulty and there are so many that no one could complete them all. Thus later items should be considered to be more difficult – but item difficulty is probably not the best assessment of item/scale quality.

Guessing the correct answer in true/false and multiple choice contexts interferes with the goal of $p = .50$. In a 1952 issue of *Psychometrika*, Lord provided this guide for optimal p values based on the number of choices in the objective context:

Optimal p values

Number of Choices	Optimal Mean Difficulty Level
2 (T/F)	0.85
3	0.77
4	0.74
5	0.69
Constructed response essay	0.5

The purpose of the testing changes the ideal difficulty level.

- If the test is *norm-referenced* (ACT, SAT, GRE), .50 is very useful.
- If the test is mastery oriented, p values may be as high as 0.90 since student performance is a function of repeated attempts with feedback.

Item intercorrelations impacts interpretation of item difficulty.

- The more homogeneous the test, the higher these intercorrelations will be. If all items were perfectly intercorrelated and all were of the .50 difficulty level:
 - the same 50 persons out of 100 would pass each item, that is,
 - half of the test takers would obtain perfect scores, the other half zero scores
- It is best to select items with a moderate spread of difficulty but whose AVERAGE difficulty level is .50
- The percentage of persons passing an item expresses the item difficulty in terms of which statistical scale of measurement? Is it nominal, ordinal, interval, or ratio?
 - Because of this issue, we can correctly indicate the rank order or relative difficulty of the items
 - However, we cannot infer that the difference in difficulty between Items 1 and 2 is equal to the difference between Items 2 and 3.
- We can make an *equal-interval inference* with the table of normal curve frequencies (i.e., translating the proportion to z-scores). Z-scores would be used as the units if an equal interval inference was required in the analysis. For example,
 - $p = .84$ is equal to -1 SD
 - $p = .16$ is equal to $+1\text{ SD}$

Seem a little upside down? Recall that we are calculating the percent passing and starting the count “from the top.” So a relatively easy item where 84% passed, would have an standard deviation of -1.

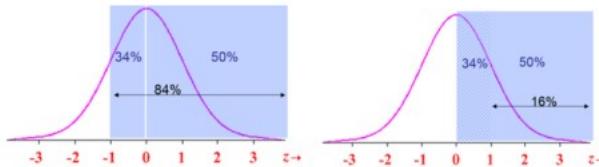


Figure 6.1: Image of graphs where $p = .84$ and $p = .16$

6.5 Item Discrimination

The degree to which an item differentiates correctly among test takers in the behavior that the test is designed to measure.

- the *criterion* can be internal or external to the test itself

- under some conditions, the two approaches lead to opposite results because (a) items chosen to maximize the validity of the test tend to be the ones rejected on the basis of internal consistency, and (b) rejecting items with low correlations with the total score tends to homogenize the test (we are more likely to keep items with the highest average intercorrelations)
- *internal* criteria maximizes internal consistency or homogeneity of the test.
 - Example: achievement test, where criteria is total score itself
- *external* criteria maximize the validity of an external criterion.
 - Example: a different assessment of the same ability being assessed

6.5.1 Index of Discrimination

- Compare the proportion of cases that pass an item in contrasting criterion groups
 - upper (U) and lower (L) criterion groups are selected from the extremes of the distribution
 - traditionally these groups are created from the 27% from each of those sides of the distribution
- This *index of discrimination (D)* can be expressed as a difference of raw frequencies (U - L), or (more conventionally) as the difference of percentages of those who scored it correctly in the upper 27% and lower 27% groups
 - when all members of the U group and none of the members of the L group pass, D = 100
 - when all members of the L group and none of the members of the U group pass, D = 0
 - optimum point at which these two conditions reach balance is with the upper and lower 27%

<hr/> <hr/> Optimal Discrimination <hr/>	
Difficulty	Discrimination
0.40 and larger	Excellent
0.30 - 0.39	Good
0.11 - 0.29	Fair
0.00 -0.10	Poor
Negative values	Mis-keyed or other major flaw

6.5.2 Application of Item Difficulty and Discrimination

Earlier I asked you to “take the quiz.” To keep it engaging, I encourage you to look at your own answers and compare them to “what happened” from in this actual exam administration. I will

KEY		A	A	D	D	B	A		
	ID	Missed	Score	Item5	Item11	Item19	Item21	Item37	Item38
	1	3	38	A	A	D	B	B	A
	10	3	38	A	C	D	D	B	D
	11	3	38	A	A	D	D	C	D
	8	4	37	A	A	D	D	C	D
	5	5	36	A	A	D	B	C	
	2	6	35	A	A	D	D	B	D
	7	7	34	A	D	D	B	B	D
	6	8	33	A	C	A	B	A	D
	9	10	31	A	C	D	B	C	A
	4	12	29	A	A	D	D	C	D
	12	12	29	A	C	D	D	A	A
	3	16	25	A	D	D	D	A	D

Figure 6.2: Image of scores and responses of 6 items from 12 students.

demonstrate how to evaluate my exam items with these indices of difficulty and discrimination. I have intentionally selected items with a variety of desirable (and undesirable) characteristics.

Item 5 A grouping variable such as men or women that uses dummy coding of 1 and 0 to categorize the groups is an example of _____ scaling.

- a) Nominal
- b) Ordinal
- c) Interval
- d) Ratio

If we wanted to hand-calculate the index of discrimination for Item #5, we find that 3 people (100%) in the upper group selected the correct answer and 3 people (100%) in the lower group selected the correct answer: $3 - 3 = 0$. If you prefer percentages: $100\% - 100\% = 0\%$. This means there is no discrimination in performance of the upper and lower performing groupings.

Older scoring systems (e.g., Scantron) used to provide this information.

Considering what we have learned already, Item #5 is:

- too easy
- does not discriminate between upper and lower performance
- *Yes, there is more data on here, but we will save it for the next level of review...just a few moments.*

Item 11 The term “grade inflation” has frequently been applied to describe the distribution of grades in graduate school. Which of the following best describes this distribution.

- a) negatively skewed
- b) uniform/rectangular

Item 5	Answer: A Point Biserial: 0.00	Correct Responses: Total Group: Upper 27% of Group: Lower 27% of Group:
Distractor Analysis - * indicates correct answer:		
	*A B C D E	
Pt-Biser.:	0.00 0.00 0.00 0.00 0.00	
Frequency%:	100.00 0.00 0.00 0.00 0.00	

Figure 6.3: Scantron image of item analysis for exam item #5

- c) positively skewed and leptokurtic
- d) uniform and platykurtic

For Item #11, 2 people (~66%) from the upper group selected the correct answer, 1 person (~33%) from the lower group selected the correct answer. Thus, the U-L was +1 (+33%) and the item is working in the proper direction.

Item 11	Answer: A Point Biserial: 0.44	Correct Responses: Total Group: 50.00% Upper 27% of Group: 66.67% Lower 27% of Group: 33.33%
Distractor Analysis - * indicates correct answer:		
	*A B C D E	
Pt-Biser.:	0.44 0.00 -0.15 -0.39 0.00	
Frequency%:	50.00 0.00 33.33 16.67 0.00	

Figure 6.4: Scantron image of item analysis for exam item #11

Considering what we have learned already, Item #11 is:

- difficult (50% overall selected the correct item)
- does discriminate between upper and lower performance, with more individuals in the upper groups selecting the correct answer than in the lower group

Item 19 All distributions of Z-scores will have the identical

- a) Mean
- b) Variance
- c) Standard deviation
- d) All of the above

Item 19	Answer: D Point Biserial: -0.06	Correct Responses: Total Group: 91.87% Upper 27% of Group: 100.00% Lower 27% of Group: 100.00%
Distractor Analysis - * indicates correct answer:		
A	B	C
Pt-Biser.: 0.06	0.00	0.00
Frequency%: 8.33	0.00	91.67
D		
E		
Pt-Biser.: 0.06 0.00 0.00 -0.06 0.00		
Frequency%: 8.33 0.00 0.00 91.67 0.00		
** Review item(s): A		

Figure 6.5: Scantron image of item analysis for exam item #19

Hand calculation: Upper = 3 (100%), Lower = 3 (100%). Difference = 0.

Considering what we have learned already, Item #19 is:

- somewhat easy (92% overall selected the correct item)
- using the U - L discrimination index, it does not discriminate between upper and lower performance

Item 21 The most appropriate score for comparing scores across two or more distributions (e.g., exam scores in math and art classes) is the:

- a) mean
- b) percentile rank
- c) raw score
- d) z-score

Hand calculation: Upper = 2 (66%), Lower = 3 (100%). Difference = -33%. This item is upside down. This is different than the Scantron snip below because uppers and lowers were likely calculated on exam total that included subjectively scored items (essays; and I no longer have that data).

Item 21	Answer: D Point Biserial: -0.13	Correct Responses: Total Group: 58.33% Upper 27% of Group: 66.67% Lower 27% of Group: 66.67%
Distractor Analysis - * indicates correct answer:		
A	B	C
Pt-Biser.: 0.00	0.13	0.00
Frequency%: 0.00	41.67	0.00
D	E	
Pt-Biser.: 0.00 0.13 0.00 -0.13 0.00		
Frequency%: 0.00 41.67 0.00 58.33 0.00		
** Review item(s): B		

Figure 6.6: Scantron image of item analysis for exam item #21

Considering what we have learned already, Item #21 is:

- somewhat difficult (58% overall selected the correct item)
- on the basis of the hand-calculations it does not discriminate between uppers and lowers

Item 37 Of the following, what statement best describes $r^2 = .49$

- a) strong positive correlation

- b) strong positive or negative correlation
- c) weak positive or negative correlation
- d) weak negative correlation

Hand calculation: Upper = 2 (66%), Lower = 0 (0%). Difference = 66%.

Item 37	Answer: B Point Biserial: 0.48	Correct Responses: Total Group: 33.33% Upper 27% of Group: 66.67% Lower 27% of Group: 0.00%		
Distractor Analysis - * indicates correct answer:				
A	*B	C	D	E
Pt-Biser.: -0.52	0.48	0.00	0.00	0.00

Figure 6.7: Scantron image of item analysis for exam item #37

Considering what we have learned already, Item #37 is:

- very difficult (33% overall selected the correct item)
- on the basis of the hand-calculations, this completely discriminates the uppers from the lowers)

Item 38 When there are no ties among ranks, what is the relationship between the Spearman rho (ρ) and the Pearson r (r)?

- a) $\rho = r$
- ii) $\rho > r$
- a) $\rho < r$
- b) no relationship

Hand calculation: Upper = 1 (33%), Lower = 1 (33%). Difference = 0%.

Item 38	Answer: A Point Biserial: -0.21	Correct Responses: Total Group: 25.00% Upper 27% of Group: 33.33% Lower 27% of Group: 33.33%		
Distractor Analysis - * indicates correct answer:				
'A	B	C	D	E
Pt-Biser.: -0.19	0.00	0.00	0.19	0.00

Figure 6.8: Scantron image of item analysis for exam item #38

Considering what we have learned already, Item #21 is:

- very difficult (25% overall selected the correct item)
- on the basis of the hand-calculations, this does not discriminate the uppers from the lowers

6.6 In the *psych* Package

Using the *score.multiple.choice()* function in the *psych* package. Documentation is pp. 85-86 in <http://personality-project.org/r/overview.pdf>

A multiple choice exam presumes that there is one correct response. We start with a dataset that records the students' responses. It *appears* that the psych package requires these responses to be numerical (rather than A, B, C, D).

```
#For portability of the lesson, I hand-entered the exam score data.
#Variables are items (not students), so the entry is the 41 items for the 12 students
Item1 <- c(1,1,4,1,1,1,1,1,1,1,1)
Item2 <- c(4,4,1,4,4,4,4,4,4,4,4)
Item3 <- c(1,1,4,1,1,1,1,3,1,1,1)
Item4 <- c(2,3,2,2,2,2,2,2,2,2,2)
Item5 <- c(1,1,1,1,1,1,1,1,1,1,1)
Item6 <- c(1,1,1,1,1,1,1,1,1,1,1)
Item7 <- c(3,3,4,4,4,3,3,3,3,3,2)
Item8 <- c(1,2,2,4,2,2,2,2,1,4,2,2)
Item9 <- c(1,1,4,4,1,4,1,1,1,1,4)
Item10 <- c(3,3,3,2,3,2,3,2,2,3,3,3)
Item11 <- c(1,1,4,1,1,3,4,1,3,3,1,3)
Item12 <- c(2,1,2,4,2,2,2,2,2,2,2,2)
Item13 <- c(2,2,3,3,2,2,2,2,2,2,2,1)
Item14 <- c(2,2,2,2,2,2,3,2,2,2,2,2)
Item15 <- c(2,1,1,3,2,4,2,2,2,2,4,2)
Item16 <- c(2,2,2,4,4,2,2,2,4,2,2,1)
Item17 <- c(1,1,1,1,1,1,1,1,1,1,1,1)
Item18 <- c(3,3,3,3,3,3,3,3,3,3,3,3)
Item19 <- c(4,4,4,4,4,1,4,4,4,4,4,4)
Item20 <- c(1,1,1,1,1,1,1,1,1,1,1,1)
Item21 <- c(2,4,4,4,2,2,2,4,2,4,4,4)
Item22 <- c(3,3,3,3,3,3,3,3,3,3,3,1)
Item23 <- c(3,3,2,3,3,3,3,3,2,3,3,2)
Item24 <- c(3,3,1,3,3,3,2,3,3,3,3,1)
Item25 <- c(2,2,2,2,3,2,2,2,2,2,2,2)
Item26 <- c(4,4,4,4,4,4,4,4,1,4,4,1)
Item27 <- c(4,4,1,4,4,4,4,4,4,4,4,4)
Item28 <- c(1,1,1,1,1,1,1,1,1,1,1,1)
Item29 <- c(1,3,1,1,1,1,1,1,1,1,1,1)
Item30 <- c(2,2,2,2,2,2,2,2,2,2,2,2)
Item31 <- c(1,1,1,2,1,1,3,1,1,1,1,2)
Item32 <- c(1,1,3,1,1,1,3,1,1,1,1,1)
Item33 <- c(3,3,3,3,3,3,3,3,3,3,3,3)
Item34 <- c(3,3,3,3,3,3,3,3,3,3,3,3)
Item35 <- c(3,3,3,3,3,3,3,3,3,3,3,3)
Item36 <- c(2,2,2,2,2,2,2,2,2,2,2,2)
Item37 <- c(2,2,1,3,3,1,2,3,3,2,3,1)
```

```
Item38 <- c(1,4,4,4,NA,4,4,4,1,4,4,4,1)
Item39 <- c(3,3,3,3,3,3,3,3,3,3,3,3,3)
Item40 <- c(3,3,4,3,3,3,3,3,3,3,3,3,3)
Item41 <- c(2,1,2,2,2,4,4,2,2,4,4,2)
```

```
exam <- data.frame(Item1, Item2, Item3, Item4, Item5, Item6, Item7, Item8, Item9, Item10, Item11)
```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```
#write the simulated data as a .csv
#write.table(exam, file="exam.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#exam <- read.csv ("exam.csv", header = TRUE)
```

```
#to save the df as an .rds (think "R object") file on your computer; it should save in the same directory as the R script
#saveRDS(exam, "exam.rds")
#bring back the simulated dat from an .rds file
#exam <- readRDS("exam.rds")
```

We create a key of the correct answers.

```
exam.keys <- c(1,4,1,2,1,1,3,2,1,3,1,2,2,2,2,2,1,3,4,1,4,3,3,3,1,4,4,1,1,2,1,1,3,3,3,2,2,1,3,3,3)
```

We then insert that key into the *psych* package’s *score.multiple.choice()* function.

```
results <- psych::score.multiple.choice(exam.keys, exam, score = TRUE, short = FALSE, skew = TRUE)
```

```
Warning in cor(items, scores, use = "pairwise"): the standard deviation is zero
```

```
results
```

Call: NULL

```
(Unstandardized) Alpha:
[1] 0.73
```

```
Average item correlation:
[1] 0.06
```

```
item statistics
    key    1     2     3     4 miss      r   n mean    sd skew kurtosis    se
Item1     1 0.92 0.00 0.00 0.08 0.00  0.65 12 0.92 0.29 -2.65      5.48 0.08
Item2     4 0.08 0.00 0.00 0.92 0.00  0.65 12 0.92 0.29 -2.65      5.48 0.08
Item3     1 0.83 0.00 0.08 0.08 0.00  0.34 12 0.83 0.39 -1.57      0.53 0.11
```

Item4	2	0.00	0.92	0.08	0.00	0.00	-0.11	12	0.92	0.29	-2.65	5.48	0.08
Item5	1	1.00	0.00	0.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item6	1	1.00	0.00	0.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item7	3	0.00	0.08	0.67	0.25	0.00	0.72	12	0.67	0.49	-0.62	-1.74	0.14
Item8	2	0.17	0.67	0.00	0.17	0.00	-0.13	12	0.67	0.49	-0.62	-1.74	0.14
Item9	1	0.67	0.00	0.00	0.33	0.00	0.81	12	0.67	0.49	-0.62	-1.74	0.14
Item10	3	0.00	0.33	0.67	0.00	0.00	0.18	12	0.67	0.49	-0.62	-1.74	0.14
Item11	1	0.50	0.00	0.33	0.17	0.00	0.42	12	0.50	0.52	0.00	-2.16	0.15
Item12	2	0.08	0.83	0.00	0.08	0.00	0.17	12	0.83	0.39	-1.57	0.53	0.11
Item13	2	0.08	0.75	0.17	0.00	0.00	0.85	12	0.75	0.45	-1.01	-1.04	0.13
Item14	2	0.00	0.92	0.08	0.00	0.00	-0.04	12	0.92	0.29	-2.65	5.48	0.08
Item15	2	0.17	0.58	0.08	0.17	0.00	0.32	12	0.58	0.51	-0.30	-2.06	0.15
Item16	2	0.08	0.67	0.00	0.25	0.00	0.40	12	0.67	0.49	-0.62	-1.74	0.14
Item17	1	1.00	0.00	0.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item18	3	0.00	0.00	1.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item19	4	0.08	0.00	0.00	0.92	0.00	0.04	12	0.92	0.29	-2.65	5.48	0.08
Item20	1	1.00	0.00	0.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item21	4	0.00	0.42	0.00	0.58	0.00	-0.19	12	0.58	0.51	-0.30	-2.06	0.15
Item22	3	0.08	0.00	0.92	0.00	0.00	0.34	12	0.92	0.29	-2.65	5.48	0.08
Item23	3	0.00	0.25	0.75	0.00	0.00	0.71	12	0.75	0.45	-1.01	-1.04	0.13
Item24	3	0.17	0.08	0.75	0.00	0.00	0.61	12	0.75	0.45	-1.01	-1.04	0.13
Item25	1	0.00	0.92	0.08	0.00	0.00	NA	12	0.00	0.00	NaN	NaN	0.00
Item26	4	0.17	0.00	0.00	0.83	0.00	0.34	12	0.83	0.39	-1.57	0.53	0.11
Item27	4	0.08	0.00	0.00	0.92	0.00	0.65	12	0.92	0.29	-2.65	5.48	0.08
Item28	1	1.00	0.00	0.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item29	1	0.92	0.00	0.08	0.00	0.00	-0.11	12	0.92	0.29	-2.65	5.48	0.08
Item30	2	0.00	1.00	0.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item31	1	0.75	0.17	0.08	0.00	0.00	0.41	12	0.75	0.45	-1.01	-1.04	0.13
Item32	1	0.83	0.00	0.17	0.00	0.00	0.45	12	0.83	0.39	-1.57	0.53	0.11
Item33	3	0.00	0.00	1.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item34	3	0.00	0.00	1.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item35	3	0.00	0.00	1.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item36	2	0.00	1.00	0.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item37	2	0.25	0.33	0.42	0.00	0.00	0.49	12	0.33	0.49	0.62	-1.74	0.14
Item38	1	0.27	0.00	0.00	0.73	0.08	-0.07	11	0.27	0.47	0.88	-1.31	0.14
Item39	3	0.00	0.00	1.00	0.00	0.00	NA	12	1.00	0.00	NaN	NaN	0.00
Item40	3	0.00	0.00	0.92	0.08	0.00	0.65	12	0.92	0.29	-2.65	5.48	0.08
Item41	4	0.08	0.58	0.00	0.33	0.00	0.40	12	0.33	0.49	0.62	-1.74	0.14

#short=FALSE allows us to produce scores; we will use these later in some IRT analyses
#names(results)

The first screen of output provides an alpha. In this context, *alpha* should tell us the consistency of getting answers right or wrong. Technically, the alpha is reduced to a KR-20 (Kuder Richardson 20). We interpret it the same. Alpha is directly effected by:

- *interitem correlations* among the items – a large number of positive correlations between items increases alpha

- test length – more items produce higher reliability (all things else equal)
- test content – the more diverse/broad, the lower the reliability coefficient

In the context of the classroom, reliabilities above .70 are probably adequate and above .80 are good. Reliabilities below .60 suggest that items should be investigated and additional measures (tests, homework assignments) should be included in assigning grades.

Focus instead on the second screen of output.

key indicates which answer was correct.

1, 2, 3, 4 (there would be as many as there are options in the multiple choice exam) provide a *distractor analysis* by indicating the percentage of time that answer was chosen. For item 1, option 1 was correct, and it was chosen 92% of the time. No individuals chose options 2 or 3. Option 4 was chosen 8% of the time.

miss indicates how many times the item was skipped.

r is a point-biserial correlation with a dichotomous correct/incorrect correlated with the continuously scaled total scale score. Positively scored items let us know that the item is working in the proper direction; the students who got the item correct, did better on the overall total score and vice versa.

- one of the best indicators of an item's ability to *discriminate* (hence, **item discrimination**) among the criterion assessed on the test
- it is important to investigate those with values close to zero (no relation between item performance with overall test performance) and those with negative values (meaning that those who had the correct answer on the item were those who scored lower on the exam).

n tells us how many participants completed the item (this would necessarily be the inverse of “miss”).

mean repeats the proportion of individuals who scored correctly; it would be the same as the percentage in the item keyed as the correct one. This is an indication of **item difficulty**.

sd gives an indication of the variability around that mean

It is important to look at the *r* and *mean* columns, together to understand the degree of difficulty and how well each item is discriminating between performance levels.

skew can provide an indication of ceiling and floor effects.

If a score has a significant negative skew (long tail to the left), then there may be a piling up of items at the upper end of the scale. This would indicate an *insufficient ceiling* and make it more difficult to discriminate among differences among the higher performers.

If a score has a significant positive skew (long tail to the right), then there may be a piling up of items at the low end, indicating an *insufficient floor*. That is, it lacks the ability to discriminate between poorer performers.

How do you tell what is significant?

A general rule of thumb says that anything greater or less than the absolute value of 1.0 is significantly skewed. A formal z-test can be conducted this way: $z_{\text{skewness}} = \frac{S-0}{SE_{\text{skewness}}}$

In our exam dataset, -2.65 is the most extremely negatively skewed item and its *se* = 0.08.

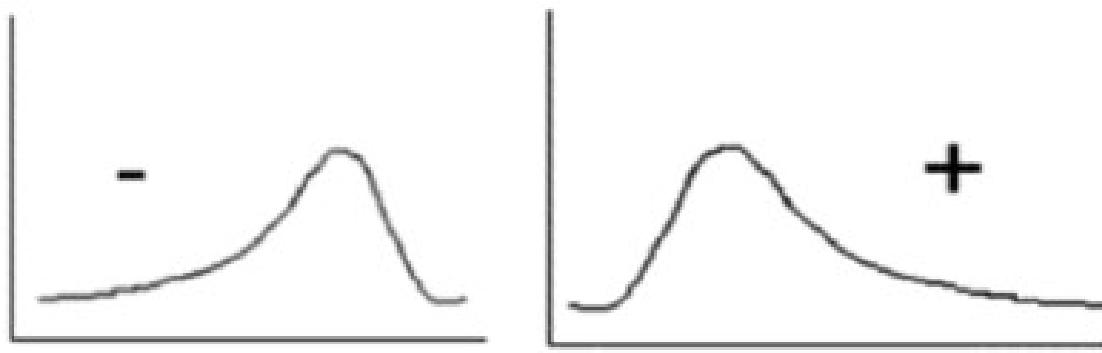


Figure 6.9: Image of two graphs illustrating positive and negative skew

```
-2.65/0.08
```

```
[1] -33.125
```

Considering that anything greater than $+/- 1.96$ is statistically significant, it is safe to say that this item has an insufficient ceiling.

What about the items with -0.30 ($se = 0.15$)?

```
-.30/.15
```

```
[1] -2
```

This is not as extreme (and recall my $N = 12$, so I should probably look up a critical t value), but there is still some question about whether my exam items can discriminate among high performers.

Please note, because these distributions are *dichotomous* (correct/incorrect) they will never be normally distributed, but, like the difficulty index, they give another glimpse of the ability to discriminate.

Before we look at the specific exam items and their output from the scoring function, let me introduce you to the features of the psych package that draw from *item response theory* (IRT).

6.6.1 A Mini-Introduction to IRT

To recap – at the instructional level, the combination of percent passing (mean) and point-biserial correlation (discrimination index) is status quo for evaluating/improving the items.

The *psych* package draws from its IRT capacity to conduct distractor analysis. IRT models individual responses to items by estimating individual ability (θ) and item difficulty ($diff$) parameters.

In these graphs, θ is on the X axis. θ is the standard unit of the IRT model that represents the level of the domain being measured. Like a z-score, a θ unit of “1” is the SD of the calibrated sample.

The pattern of responses to multiple choice ability items can show that some items have poor distractors. This may be done by using the `irt.responses` function. A good distractor is one that is negatively related to ability.

As we look at each of the exam items, we will look at the psych input from the scoring function as well as use the `results` objects to create the IRT graphs.

Item 5 A grouping variable such as men or women that uses dummy coding of 1 and 0 to categorize the groups is an example of _____ scaling.

- a) Nominal
- b) Ordinal
- c) Interval
- d) Ratio

Mean = 1.0 (much too easy), $r = \text{NA}$, Distractors: 1.00 0.00 0.00 0.00, skew = -2.65

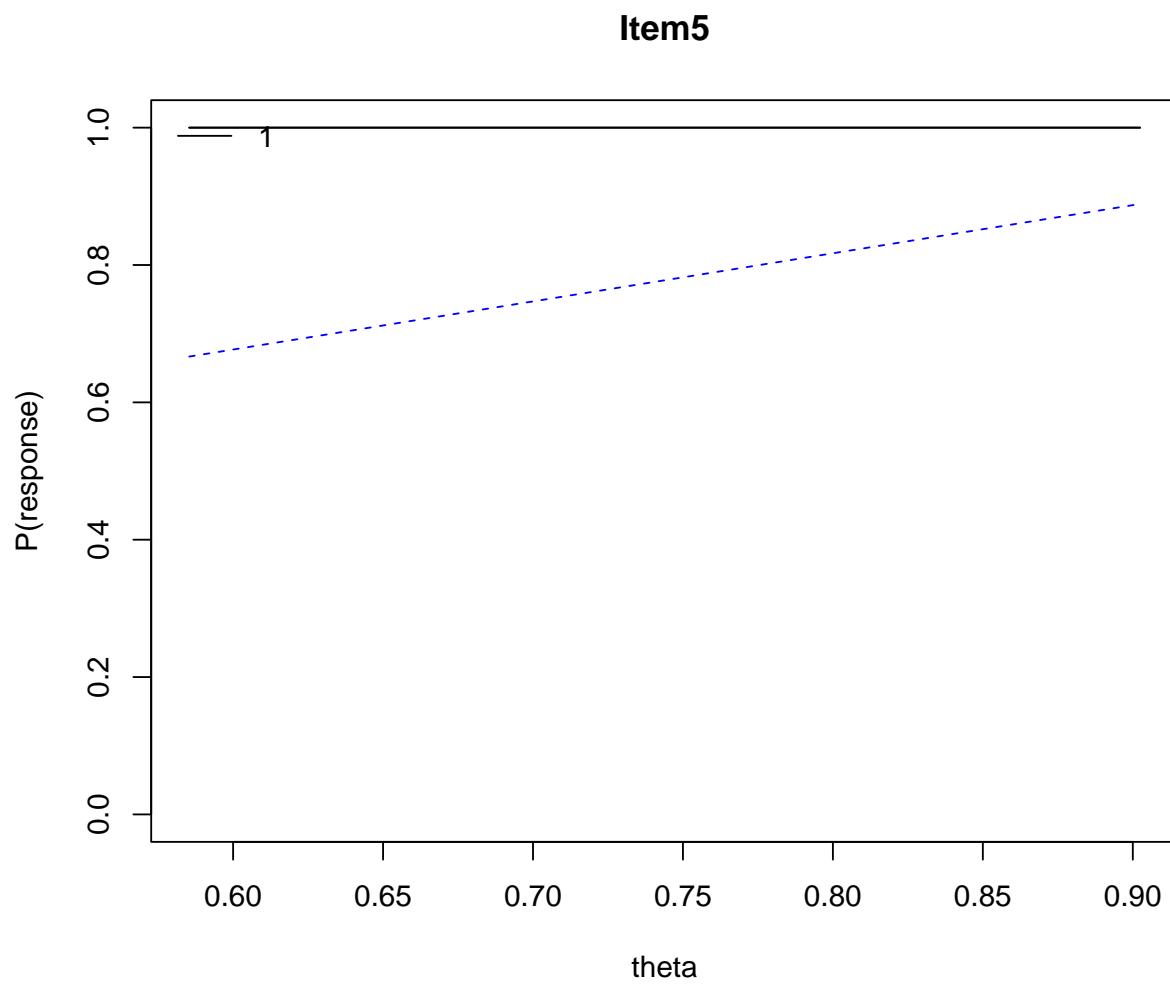
```
#irt.responses(scores$scores, exam[5], breaks = 2)
psych::irt.responses(results$scores, exam[5], breaks = 2)
```

Number of categories should be increased in order to count frequencies.

Warning in `rbind(items, dummy)`: number of columns of result is not a multiple of vector length (arg 1)

Number of categories should be increased in order to count frequencies.

Warning in `rbind(items, dummy)`: number of columns of result is not a multiple of vector length (arg 1)



With Item #5, 100% responded correctly (the flat, solid line at the top); there is not much to see.

Item 11 The term “grade inflation” has frequently been applied to describe the distribution of grades in graduate school. Which of the following best describes this distribution.

- a) negatively skewed
- b) uniform/rectangular
- c) positively skewed and leptokurtic
- d) uniform and platykurtic

Mean = .50, $r = .42$, Distractors: 0.50 0.00 0.33 0.17, skew = 0.00

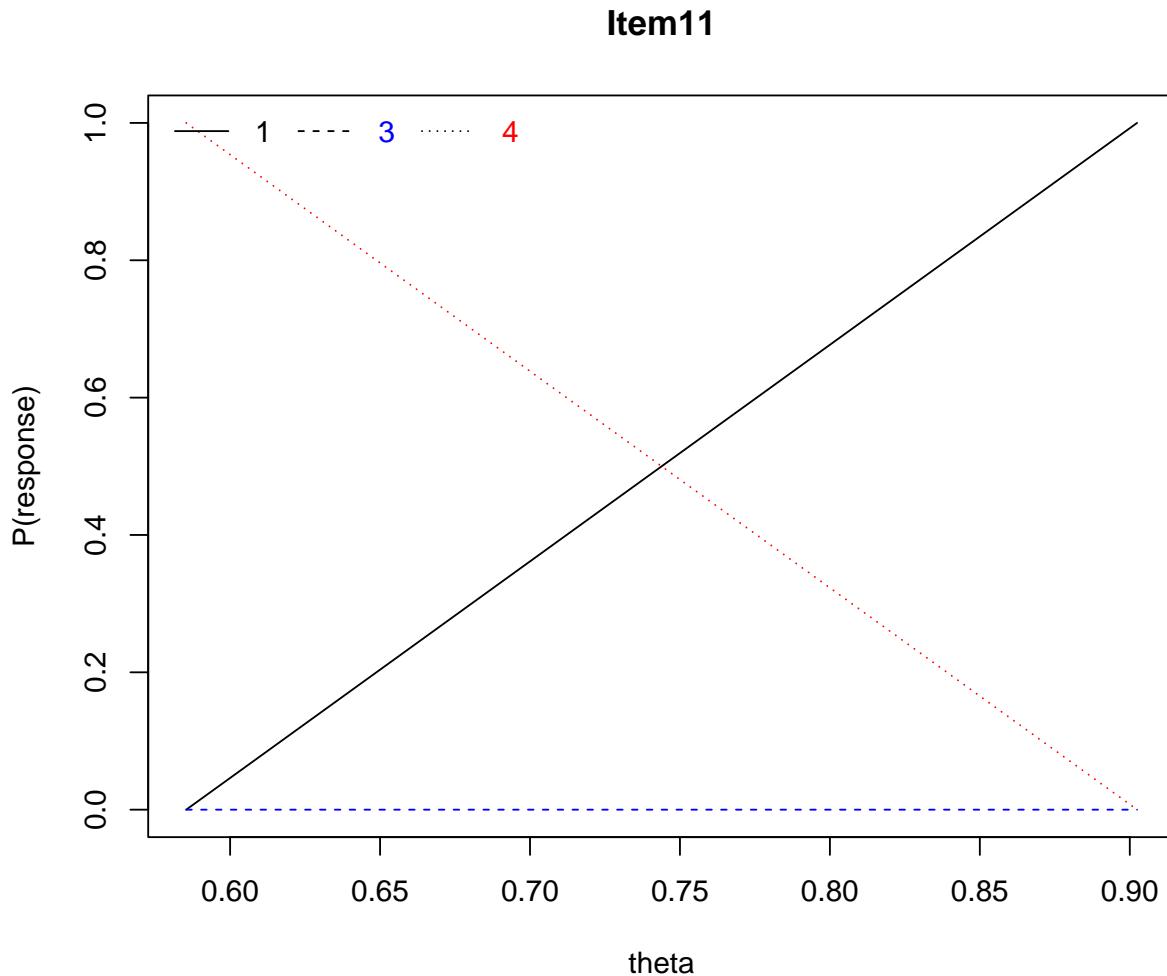
```
psych::irt.responses(results$scores, exam[11], breaks = 2)
```

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)



With Item #11, there is a positive relationship between 1/A (correct answer) and ability (theta), no relationship between 3/C and ability, and a negative relationship between 4/D and ability (indicating that 4/D is a good distractor). These map onto each of the point-biserial correlations associated with the distractors in the Scantron output.

Item 19 All distributions of Z-scores will have the identical

- a) Mean
- b) Variance

- c) Standard deviation
- d) All of the above

Mean = .92, $r = .04$, Distractors: 0.08 0.00 0.00 0.92 , skew = -2.65

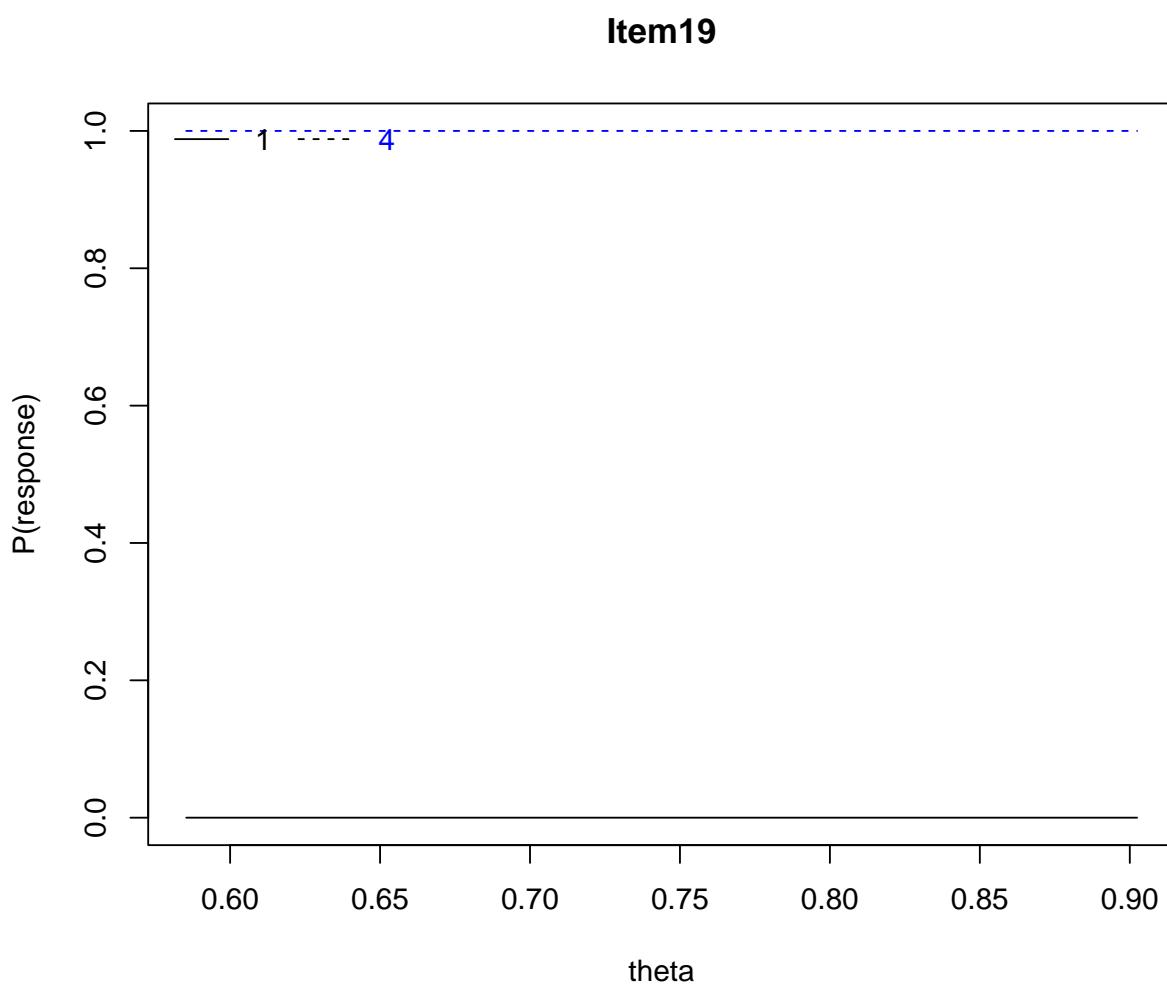
```
psych::irt.responses(results$scores, exam[19], breaks = 2)
```

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)



Item #19 shows rather flat (no relationship) relations with ability for the correct item and the lone distractor.

Item 21 The most appropriate score for comparing scores across two or more distributions (e.g., exam scores in math and art classes) is the:

- a) mean
- b) percentile rank
- c) raw score
- d) z-score

Mean = .58, $r = -.19$, Distractors: 0.00 0.42 0.00 0.58, skew = -0.30

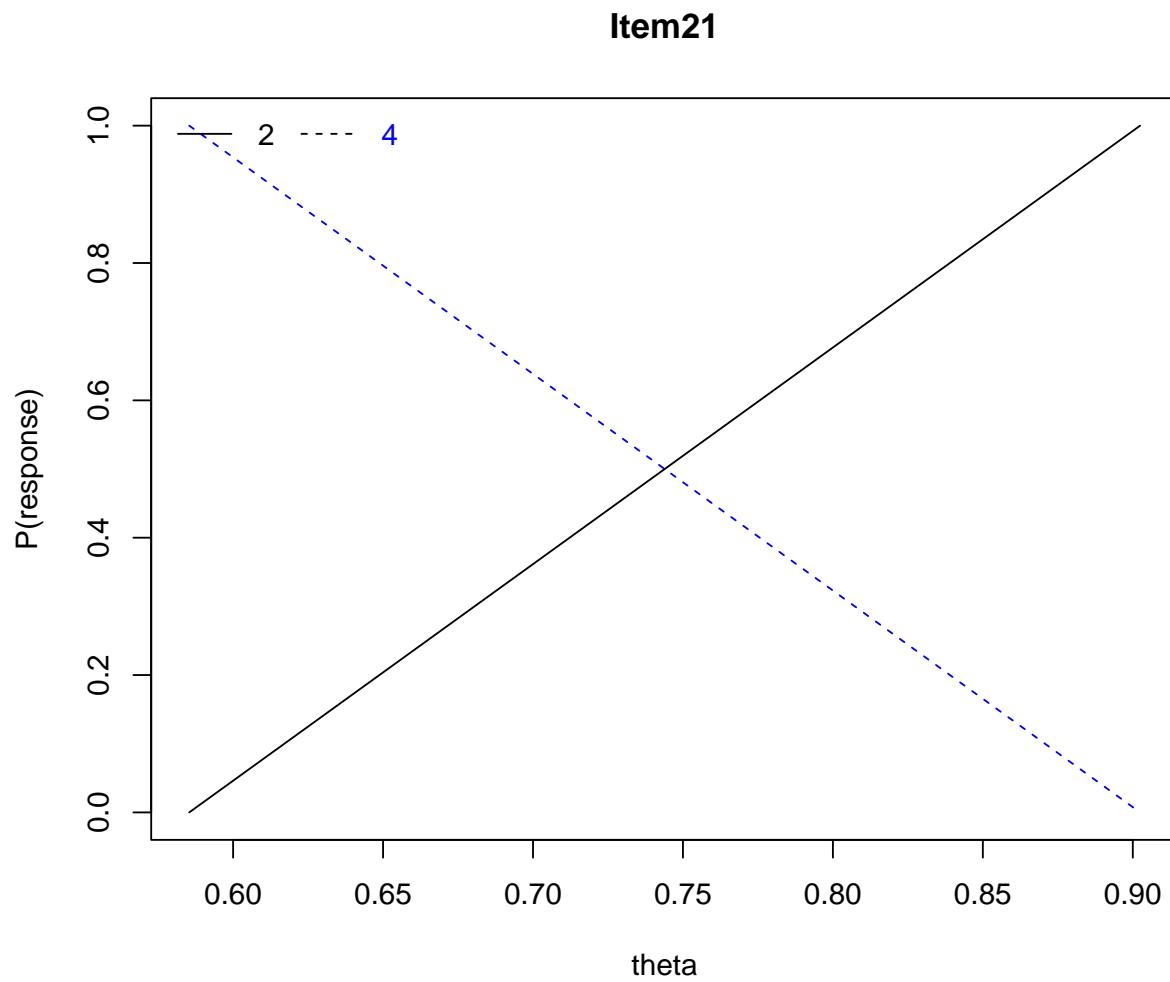
```
psych::irt.responses(results$scores, exam[21], breaks = 2)
```

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)



For Item #21, a positive relationship between the WRONG answer (2/B) and ability (theta) and a negative relationship between 4/D (incorrect answer) and ability. This makes sense as the point biserial for the overall item was 0-.13.

Item 37 Of the following, what statement best describes $r^2 = .49$

- a) strong positive correlation
- b) strong positive or negative correlation
- c) weak positive or negative correlation
- d) weak negative correlation

Mean = .33, $r = .49$, Distractors: 0.25 0.33 0.42 0.00, skew = .62

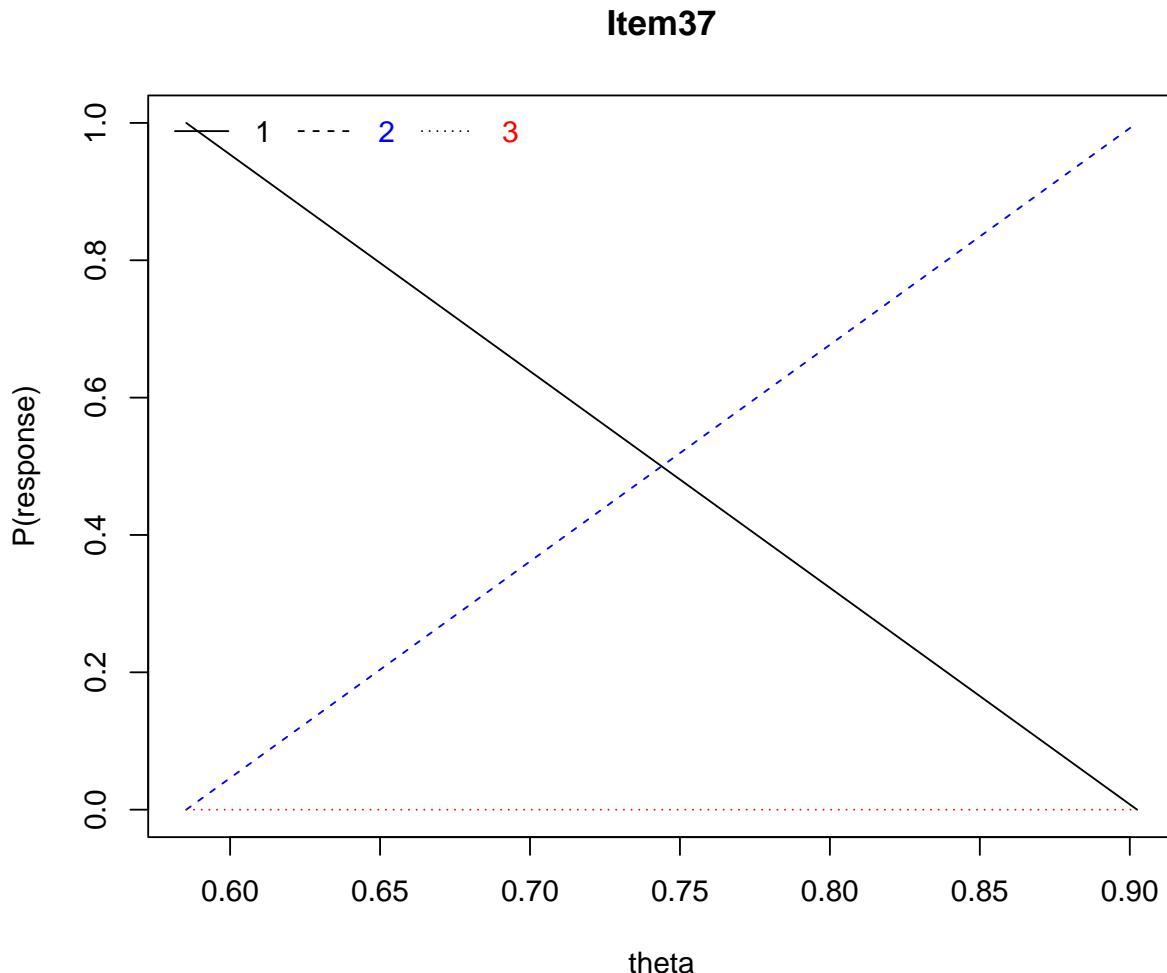
```
psych::irt.responses(results$scores, exam[37], breaks = 2)
```

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)



For Item #37, a negative relation between endorsing 1/A and ability (a good distractor). No relationship with ability for endorsing 3/C. A positive relation with ability for those endorsing 2/B (correct answer).

Item 38 When there are no ties among ranks, what is the relationship between the Spearman rho (ρ) and the Pearson r (r)?

- a) $\rho = r$
- ii) $\rho > r$

- a) $\rho < r$
- b) no relationship

Mean = .27, $r = -.07$, Distractors: 0.27 0.00 0.00 0.73, skew = .68

Notice anything else that's funky about Item #38?

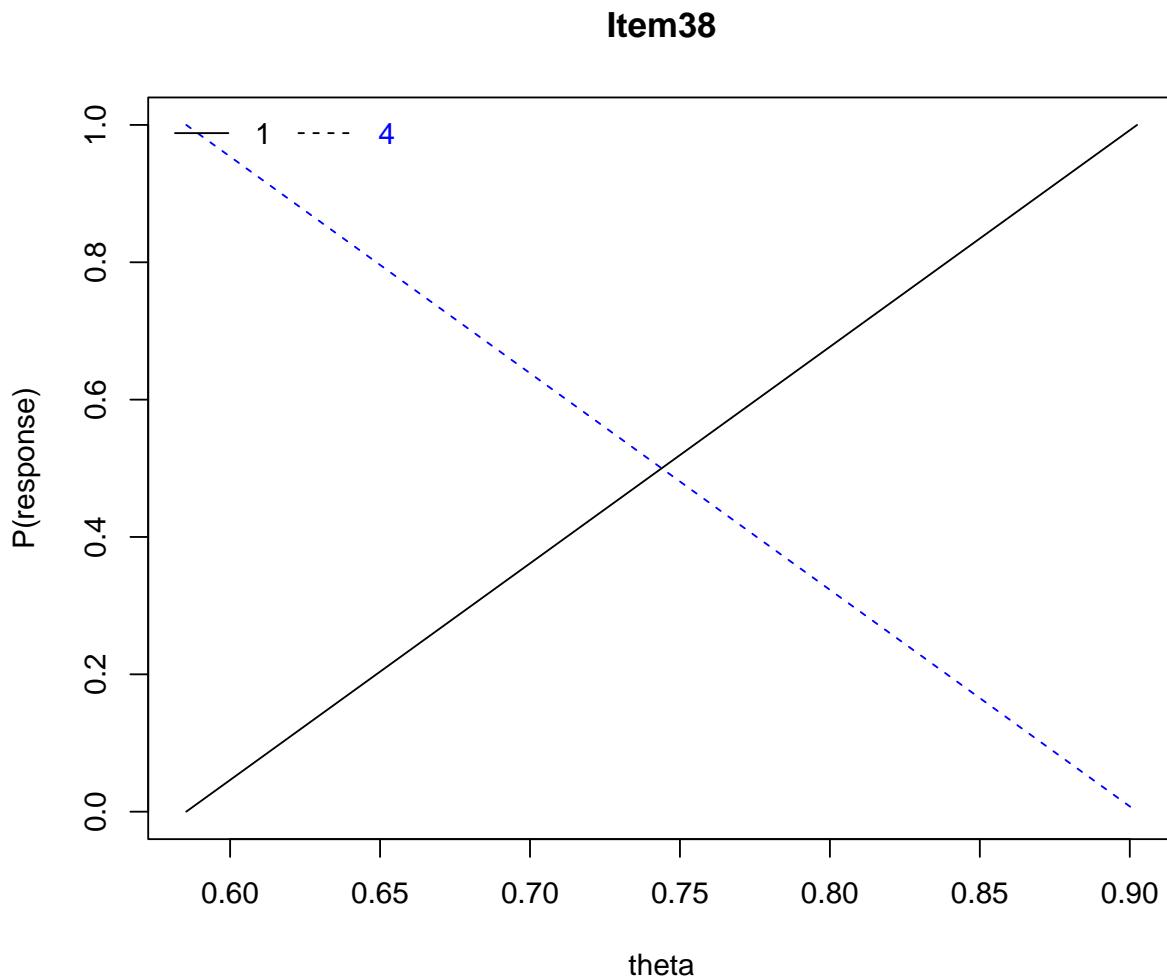
```
psych::irt.responses(results$scores, exam[38], breaks = 2)
```

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)

Number of categories should be increased in order to count frequencies.

Warning in rbind(items, dummy): number of columns of result is not a multiple of vector length (arg 1)



6.7. CLOSING THOUGHTS ON DEVELOPING MEASURES IN THE EDUCATION/ACHIEVEMENT CONTEXT

For Item #38, there is a positive relationship with ability for endorsing 1/A (correct answer) and a negative relationship with ability for 4/D (incorrect answer).

Regarding overall test characteristics

3340 Final Exam Detailed Item Analysis Report On Final Exam Version A Version A				
Course #:		Instructor:	Bikos	
Course Title:		Description:		
Day/Time:		Term/Year:		
Total Possible Points:	56.00	Median Score:	49.00	Mean Score: 46.83
Student in this group:	12	Mode Score:	53.00	Variance: 37.14
Standard Deviation:	6.09	Highest Score:	53.00	Lowest Score: 35.00
Measure of Skewness:	-0.64	Degree of Kurtosis:	-1.07	Reliability Coefficient (KR20): 0.77

6.7 Closing Thoughts on Developing Measures in the Education/Achievement Context

Item analysis tends to be an assessment of *reliability*. However, in the context of educational assessment and achievement exams, there are also *validity* issues.

Content validity is concerned with whether or not the scale adequately represents the entirety of the *domain* to be assessed.

In educational and achievement contexts, this is often accomplished with a *table of specifications*. I introduced this in the [Validity lesson](#). As a refresher, I will include another example – imagining that I am going to write a quiz or short exam based on the learning objectives of this, single, lesson. There are a number of different ways to organize the types of knowledge that is being assessed. Since the American Psychological Association (and others) work in “KSAs” (knowledge, skills, attitudes) in their accreditation standards, I will use those.

In creating a table of specifications, we start with the learning objectives. Then we decide what type of items to write and what type of performance level they satisfy. This helps us ensure that all learning objectives are proportionately covered, using a variety of assessment approaches. Otherwise, we might be tempted to include the items that come easily to us or that are from our favorite topics. Personally, I find that when I work on the exam, and am informed by the learning objectives and table of specifications, I find myself tinkering with all three. I am inclined to believe that this results in an ever-increasingly-improved pedagogy.

Table of Specifications

Learning Objectives	Knowledge	Skills	Attitudes	% of test
Provide a rationale for why having a <i>test bank</i> might be a good idea.			1 item	30%
Describe the effects of skewness on the interpretation of exam results.	2 items			10%
Evaluate the quality of a multiple choice item on the basis of item difficulty, correlation, and discrimination.		5 items		25%

Learning Objectives	Knowledge	Skills	Attitudes	% of test
Discuss the challenges of identifying an <i>ideal</i> difficulty level for test items. Further elaborate how guessing, speeded tests, interitem correlations, and the purposes of the test influence the <i>ideal difficulty</i> .	2 items		1 item	35%
TOTALS	4 items	5 items	2 items	100%

There are a variety of free resources that help with this process. Below are some that I find helpful:

- [Bloom's Taxonomy Verbs](#), freely available from Fractus Learning.
- [The Bloom's Taxonomy Verbs Poster for Teachers](#)
- If you have “writer’s block” for writing objectives, here is a [learning outcome generator](#) that may help get you started.
- From APA’s Education Directorate, [Guidance for Writing Behavioral Learning Objectives](#). The APA Guidance really emphasizes key components of well-written behavioral leaning objectives. These include:
 - **observable and measurable**, using action verbs that describe measureable behaviors. The APA CE office disallows the use of “understand” as an action verb,
 - statements that clearly describe what the learner will know or be able to do **as a result** of having participated,
 - focused on the learner and learning (as opposed to what the trainer is doing or leading),
 - appropriate in breadth (not too few or too many)

Takeaway message: Together, mapping out exam coverage in a table of specifications PLUS item analysis (difficulty/discrimination) can be powerful tools in educational assessment.

6.8 Practice Problems

For this particular lesson, I think some of the most meaningful practice comes from multiple choice and true/false exams that occur in your life. If you are in a class, see if your instructor is willing to share item analysis information that they have received. Learning management systems like Canvas, automatically calculate these.

If you are an instructor, calculate and review item analysis data on your own items. Think about how you might improve items between exams and cconsider how the dificulty and discrimination capacity of the item changes.

Chapter 7

Item Analysis for Likert Type Scale Construction

[Screencasted Lecture Link](#)

The focus of this lecture is on item analysis for surveys. We use information about alpha coefficients and item-total correlations (within and across subscales) to help assess what we might consider to be *within-scale convergent and discriminant validity* (although we tend to think of it as an assessment of reliability).

7.1 Navigating this Lesson

There is about 45 minutes of lecture. If you work through the materials with me it would be plan for an additional hour.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

7.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Define the corrected item-total correlation and compare it to an item-total correlation.
- List the preliminary steps essential for scale construction, beginning with item development.
- Name the type(s; e.g., reliability, validity) of psychometric evaluation that item analytic procedures assess..
- Identify threats to the interpretation of item-total correlations and alpha coefficients.
- Make decisions about item retention, deletion, and revision that balances statistical output with construct definitions.

7.1.2 Planning for Practice

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. For this lesson, please locate item-level data for a scale that has the potential for at least two subscales and a total-scale score. Ideally, the data you utilized in one or more of the prior lessons (e.g., changing the random seed in the lesson data, downloading the data from the *ReCentering Psych Stats* survey, or data you found elsewhere) will allow you to continue with these analyses. Then, please examine the following:

- produce alpha coefficients, average inter-item correlations, and corrected item-total correlations for the total and subscales, separately
- produce correlations between the individual items of one subscale and the subscale scores of all other scales
- draft an APA style results section with an accompanying table.

In my example there were only two subscales. If you have more, you will need to compare each subscale with all the others. For example, if you had three subscales: A, B, C, you would need to compare A/B, B/C, and A/C.

7.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (when possible, linked) in the text with complete citations in the reference list.

- Green & Salkind (2018). Lesson 38: Item analysis using the reliability Procedure. In S.B. Green and N.J. Salkind's, "Using SPSS for Windows and Macintosh: Analyzing and understanding data (8th ed). New York: Pearson.
 - Even though the operation of the chapter uses SPSS, the narration of the "what" and "why" of item analysis is clear and concise. Further, I have not found another chapter (not even in psychometrics texts) that addresses this as completely.
- Lewis, J. A., & Neville, H. A. (2015). Construction and initial validation of the Gendered Racial Microaggressions Scale for Black women. *Journal of Counseling Psychology*, 62(2), 289–302. <https://doi.org/10.1037/cou0000062>
 - Our research vignette for this lesson.

7.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(tidyverse)){install.packages("tidyverse")}
#if(!require(MASS)){install.packages("MASS")}
```

```
#if(!require(psych)){install.packages("psych")}
#if(!require(apaTables)){install.packages("apaTables")}
#if(!require(sjstats)){install.packages("sjstats")}
#if(!require(qualtRics)){install.packages("qualtRics")}
```

7.2 Intro to Item Analysis for Survey Development

Item analysis can be used to decide which items to include and exclude from a scale or subscale. The goal is to select a set of items that yields a summary score (total or mean) that is strongly related to the construct identified and defined in the scale.

- Item analysis is somewhat limiting because we usually cannot relate our items to a direct (external) measure of a construct to select our items.
- Instead, we *trust* (term used lightly) that the items we have chosen, together, represent the construct and we make decisions about the relative strength of each item's correlation to the total score.
- This makes it imperative that we look to both statistics and our construct definition (e.g., how well does each item map onto the construct definition)

7.3 Research Vignette

This lesson will also use the research vignette that presented the development and psychometric evaluation of the Perceptions of the LGBTQ College Campus Climate Scale (?).

The scale is six items with responses rated on a 7-point Likert scale ranging from 1 (*strongly disagree*) to 7 (*strongly agree*). Higher scores indicate more negative perceptions of the LGBTQ campus climate. Szymanski and Bissonette (?) have suggested that the psychometric evaluation supports using the scale in its entirety or as subscales composed of the following items:

- College response to LGBTQ students:
 - My university/college is cold and uncaring toward LGBTQ students. (cold)
 - My university/college is unresponsive to the needs of LGBTQ students. (unresponsive)
 - My university/college provides a supportive environment for LGBTQ students. [un]supportive; must be reverse-scored
- LGBTQ Stigma:
 - Negative attitudes toward LGBTQ persons are openly expressed on my university/college campus. (negative)
 - Heterosexism, homophobia, biphobia, transphobia, and cissexism are visible on my university/college campus. (heterosexism)
 - LGBTQ students are harassed on my university/college campus. (harassed)

A preprint of the article is available at ResearchGate. Below is the script for simulating item-level data from the factor loadings, means, and sample size presented in the published article.

```

set.seed(210827)
SzyT1 <- matrix(c(.88, .73, .73, -.07, -.02, .16, -.03, .10, -.04, .86, .76, .71), ncol=2) #prior code I replaced with above
rownames(SzyT1) <- c("cold", "unresponsive", "supportiveNR", "negative", "heterosexism", "harassment")
#rownames(Szyf2) <- paste("V", seq(1:6), sep=" ") #prior code I replaced with above
colnames(SzyT1) <- c("F1", "F2")
SzyCorMat <- SzyT1 %*% t(SzyT1) #create the correlation matrix
diag(SzyCorMat) <- 1
#SzyCorMat #prints the correlation matrix
SzyM <- c(2.31, 3.11, 2.40, 3.18, 4.44, 3.02) #item means
SzySD <- c(1.35, 1.46, 1.26, 1.60, 1.75, 1.50) #item standard deviations; turns out we won't need these
SzyCovMat <- SzySD %*% t(SzySD) * SzyCorMat #creates a covariance matrix from the correlation matrix
#SzyCovMat #displays the covariance matrix

dfSzyT1 <- as.data.frame(round(MASS::mvrnorm(n=646, mu = SzyM, Sigma = SzyCovMat, empirical = TRUE)))
dfSzyT1[dfSzyT1>7] <- 7 #restricts the upperbound of all variables to be 7 or less
dfSzyT1[dfSzyT1<1] <- 1 #restricts the lowerbound of all variable to be 1 or greater
#colMeans(dfSzy) #displays column means

library(tidyverse)
dfSzyT1 <- dfSzyT1 %>% dplyr::mutate(ID = row_number()) #add ID to each row
dfSzyT1 <- dfSzyT1 %>% dplyr::select(ID, everything()) #moving the ID number to the first column;
dfSzyT1 <- dfSzyT1 %>%
  dplyr::mutate(supportive = 8 - supportiveNR) #because the original scale had 1 reversed item
dfSzyT1 <- dfSzyT1 %>%
  dplyr::select(-supportiveNR)

```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```

#write the simulated data as a .csv
#write.table(dfSzyT1, file="dfSzyT1.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#dfSzyT1 <- read.csv ("dfSzyT1.csv", header = TRUE)

#to save the df as an .rds (think "R object") file on your computer; it should save in the same folder
#saveRDS(dfSzyT1, "dfSzyT1.rds")
#bring back the simulated dat from an .rds file
#dfSzyT1 <- readRDS("dfSzyT1.rds")

```

```
psych::describe(dfSzyT1)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
ID		1	646	323.50	186.63	323.5	323.50	239.44	1	646	645
cold		2	646	2.42	1.17	2.0	2.34	1.48	1	6	5
unresponsive		3	646	3.17	1.43	3.0	3.13	1.48	1	7	6
negative		4	646	3.22	1.52	3.0	3.16	1.48	1	7	6

heterosexism	5	646	4.41	1.61	4.0	4.45	1.48	1	7	6	-0.20
harassed	6	646	3.07	1.42	3.0	3.02	1.48	1	7	6	0.33
supportive	7	646	5.52	1.14	6.0	5.57	1.48	2	7	5	-0.30
kurtosis se											
ID			-1.21	7.34							
cold			-0.52	0.05							
unresponsive			-0.65	0.06							
negative			-0.47	0.06							
heterosexism			-0.60	0.06							
harassed			-0.36	0.06							
supportive			-0.76	0.04							

Although Szymanski and Bissonette report inter-item correlations, it does not appear that they used item analysis to guide their selection of items. In fact, it is not necessary to do so. I like to teach item analysis because I think it provides a conceptual grounding for future lessons on exploratory and confirmatory factor analysis.

7.4 Steps to Item Analysis

If this is initial scale development, the researcher(s) are wise to write more items than needed so that there is flexibility in selecting items with optimal functioning. Szymaski and Bissonette (?) do this. Their article narrates how they began with 36 items, narrowed it to 24, and – on the basis of subject matter expertise and peer review – further narrowed it to 10. The reduction of additional items happened on the basis of exploratory factor analysis.

7.4.1 Step I: Corrected item-total correlations**

A within-scale version of convergent validity.

- If needed, transform any items (i.e., reverse-coding) and calculate a total score.
- Calculate *corrected item-total correlations* by correlating each item to the total score LESS the item being evaluated.
 - to the degree that the item total represents the construct of interest, the items should be strongly correlated with the corrected total score
- Make decisions about items and scales. For items that have low or negative correlations
 - consider deletion
 - consider revision (requires readministering the scale)
- Each time an item is deleted, the process needs to be run again because it changes the total-scale score.
 - In fact, it's a very iterative process. It could be that you “add back” a previously deleted item (once others are deleted) because with each deletion/addition the statistical construct definition is evolving.

- In multidimensional scales, if the total-scale score is ever used, researchers should conduct these separately for both the total-scale score and the subscales scores.

There are reasons to not “blindly follow the results of an item analysis” (?).

- **Method factors** (aka *method effects*) are common methods (e.g., negatively word items, common phrasing such as “My supervisor tells me” versus “I receive feedback”) that are irrelevant to the characteristics or traits being measured – yet when analyzed they share variance (?).
- Adequacy of construct representation. That is, how broad is the construct and to what degree do the items represent the entire construct? If the construct is broad, but there may be a tendency to:
 - write items on a particular, narrow, aspect of the construct, ignoring others
 - one or more items to strongly correlate, tempting us to delete items that are not as strongly correlated (although they represent the construct)

This means we should think carefully and simultaneously about:

- statistical properties of the item and overall scale
- construct definition
- scale structure (unidimensional? multidimensional? hierarchical?)

Step II: Correlation of each subscale's items with other subscale totals

This provides a within-scale version of discriminant validity.

- Calculate scale scores for each of the subscales of a measure.
- Focusing on one subscale at a time, correlate each of the subscale's items with the total scores of all the other subscales.
- Comparing to the results of Step I's corrected item-total process, each item should have stronger correlations with its own items (i.e., the corrected item-total correlation) than with the other subscale total scores.

7.4.2 Data Prep

Let's do the operational work to get all the pieces we need:

1. Reverse-code the *supportive* variable
2. Raw data for each of the scales
 - total-scale score
 - campus response subscale
 - stigma subscale
3. Mean scale scores for total and subscale scores
4. A merged dataset with the raw, item-level, data PLUS the three mean scores (total, college response, stigma)

If we look at the information about this particular scale, we recognize that the *supportive* item is scaled in the opposite direction of the rest of the items. That is, a higher score on *supportive* would indicate a positive perception of the campus climate for LGBTQ individuals whereas higher scores on the remaining items indicate a more negative perception. Before moving forward, we must reverse score this item.

In doing this, I will briefly note that in this case I have given my variables one-word names that represent each item. Many researchers (including myself) will often give variable names that are alpha numerical: LGBTQ1, LGBTQ2, LGBTQn. Either is acceptable. In the psychometrics case, the one-word names may be useful shortcuts as one begins to understand the inter-item relations.

In reverse-scoring the *supportive* item, I will rename it “*unsupportive*” as an indication of its reversed direction.

```
dfSzyT1<- dfSzyT1 %>%
  dplyr::mutate(unsupportive = 8 - supportive) #scaling 1 to 7; so we subtract from 8

psych::describe(dfSzyT1)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
ID	1	646	323.50	186.63	323.5	323.50	239.44	1	646	645	0.00
cold	2	646	2.42	1.17	2.0	2.34	1.48	1	6	5	0.49
unresponsive	3	646	3.17	1.43	3.0	3.13	1.48	1	7	6	0.20
negative	4	646	3.22	1.52	3.0	3.16	1.48	1	7	6	0.33
heterosexism	5	646	4.41	1.61	4.0	4.45	1.48	1	7	6	-0.20
harassed	6	646	3.07	1.42	3.0	3.02	1.48	1	7	6	0.33
supportive	7	646	5.52	1.14	6.0	5.57	1.48	2	7	5	-0.30
unsupportive	8	646	2.48	1.14	2.0	2.43	1.48	1	6	5	0.30
	kurtosis	se									
ID	-1.21	7.34									
cold	-0.52	0.05									
unresponsive	-0.65	0.06									
negative	-0.47	0.06									
heterosexism	-0.60	0.06									
harassed	-0.36	0.06									
supportive	-0.76	0.04									
unsupportive	-0.76	0.04									

Next, we score the items. In our simulation, we have no missing data. Using an available information approach (AIA; (?)) where it is common to allow 20-25% missingness, we might allow the total-scale score to calculate if there is 1 variable missing; but none for the subscale scores. The *mean_n()* function in the *sjstats* packages is especially helpful for this.

```
LGBTQvars <- c('cold', 'unresponsive', 'negative', 'heterosexism', 'harassed', 'unsupportive')
ResponseVars <- c('cold', 'unresponsive', 'unsupportive')
Stigmavars <- c('negative', 'heterosexism', 'harassed')

dfSzyT1$TotalT1 <- sjstats::mean_n(dfSzyT1[,LGBTQvars], .80) #will create the mean for each ind
```

```
dfSzyT1$ResponseT1 <- sjstats::mean_n(dfSzyT1[,ResponseVars], .80) #will create the mean for each item
dfSzyT1$StigmaT1 <- sjstats::mean_n(dfSzyT1[,Stigmavars], .80) #will create the mean for each item
```

While we are at it, let's just create tiny dfs with just our variables of interest.

```
LGBTQT1 <- dplyr::select(dfSzyT1, cold, unresponsive, unsupportive, negative, heterosexism, harassed)
ResponseT1 <- dplyr::select(dfSzyT1, cold, unresponsive, unsupportive)
StigmaT1 <- dplyr::select(dfSzyT1, negative, heterosexism, harassed)
```

7.5 Calculating Item-Total Correlation Coefficients

Let's first ask, "Is there support for this instrument as a unidimensional measure?" To do that, we get an alpha for the whole scale score.

The easiest way to do this is apply the *alpha()* function to a tiny df with the variables in that particular scale or subscale. Any variables should be pre-reversed.

```
LGBTQalpha <- psych::alpha(LGBTQT1) #Although unnecessary, I have saved the output as objects below
LGBTQalpha
```

Reliability analysis

Call: psych::alpha(x = LGBTQT1)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.64	0.64	0.73	0.23	1.8	0.023	3.1	0.83	0.089

lower alpha	upper	95% confidence boundaries
0.6	0.64	0.69

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
cold	0.62	0.60	0.66	0.23	1.5	0.024	0.075	0.111	
unresponsive	0.60	0.58	0.67	0.22	1.4	0.024	0.094	0.076	
unsupportive	0.63	0.62	0.70	0.25	1.6	0.023	0.083	0.124	
negative	0.59	0.62	0.67	0.24	1.6	0.026	0.070	0.124	
heterosexism	0.60	0.61	0.69	0.24	1.6	0.026	0.081	0.124	
harassed	0.55	0.57	0.68	0.21	1.3	0.030	0.100	0.033	

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
cold	646	0.52	0.60	0.53	0.32	2.4	1.2
unresponsive	646	0.60	0.64	0.56	0.37	3.2	1.4
unsupportive	646	0.48	0.56	0.45	0.28	2.5	1.1
negative	646	0.63	0.57	0.50	0.39	3.2	1.5
heterosexism	646	0.64	0.58	0.48	0.39	4.4	1.6

```
harassed      646  0.70  0.66  0.57   0.50   3.1 1.4
```

	Non missing response frequency for each item							
	1	2	3	4	5	6	7 miss	
cold	0.26	0.29	0.25	0.14	0.04	0.00	0.00	0
unresponsive	0.15	0.20	0.24	0.23	0.13	0.04	0.01	0
unsupportedive	0.24	0.28	0.28	0.16	0.04	0.00	0.00	0
negative	0.15	0.19	0.24	0.22	0.12	0.05	0.02	0
heterosexism	0.05	0.08	0.14	0.25	0.22	0.14	0.12	0
harassed	0.16	0.21	0.24	0.24	0.11	0.02	0.02	0

Examining our list, the overall alpha is 0.64. Further, the average inter-item correlation (*average_r*) is .23.

And just hold up a minute, I thought you told us alpha was bad!

- While it is less than ideal, we still use it all the time,
 - keeping in mind its relative value (does it increase/decrease, holding other things [like sample size] constant) and
 - also looking at alpha alternatives (such as we obtained from the omega output)
- Why alpha in this context? Its information about *consistency* is essential. In evaluating a scale's reliability we do want to know if items (unidimensionally or across subscales) are responding consistently high/middle/low.

We take note of two columns:

- *r.cor* is the correlation between the item and the total-scale score WITH THIS ITEM INCLUDED. When our focus is on the contribution of a specific item, this information is not helpful since this column gets “extra credit” for the redundancy of the duplicated item.
- *r.drop* is the corrected item-total correlation. This is the better choice because it deletes the individual item being evaluated (eliminating the redundancy) prior to conducting the correlation.
 - Looking at the two columns, notice that the *r.drop* correlations are lower. This is the more honest correlation of the item with the *other* items.
 - In item analysis, we look for items that have relatively high (assessing redundancy or duplication) of items and relatively low (indicating they are unlike the other items) values.

If we thought an item was problematic, we could eliminate it and rerun the analysis. Because we are looking at a list of items that “made the cut,” we don’t have any items that are concerningly high or low. For demonstration purposes, though, the corrected item-total correlation (*r.drop*) of the *unresponsive* variable was the lowest (.28). Let’s re-run the analysis excluding this item.

```
minus_unresponsive <- dplyr::select(dfSzyT1, cold, unsupportedive, negative, heterosexism, harassed)
```

```
psych::alpha(minus_unresponsive)
```

Warning in psych::alpha(minus_unresponsive): Some items were negatively correlated with the total scale and probably should be reversed.

To do this, run the function again with the 'check.keys=TRUE' option

Some items (cold unsupportive) were negatively correlated with the total scale and probably should be reversed.

To do this, run the function again with the 'check.keys=TRUE' option

Reliability analysis

Call: psych::alpha(x = minus_unresponsive)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.6	0.58	0.67	0.22	1.4	0.024	3.1	0.86	0.076

lower alpha	upper	95% confidence boundaries	
0.55	0.6	0.65	

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
cold	0.64	0.60	0.62	0.27	1.48	0.021	0.104	0.277	
unsupportive	0.64	0.60	0.63	0.27	1.50	0.022	0.103	0.290	
negative	0.47	0.48	0.55	0.19	0.93	0.035	0.077	0.076	
heterosexism	0.48	0.49	0.58	0.19	0.95	0.034	0.093	0.076	
harassed	0.44	0.43	0.57	0.16	0.76	0.037	0.117	-0.049	

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
cold	646	0.41	0.51	0.36	0.15	2.4	1.2
unsupportive	646	0.40	0.50	0.35	0.15	2.5	1.1
negative	646	0.73	0.67	0.60	0.49	3.2	1.5
heterosexism	646	0.73	0.66	0.57	0.46	4.4	1.6
harassed	646	0.76	0.72	0.63	0.55	3.1	1.4

Non missing response frequency for each item

	1	2	3	4	5	6	7	miss
cold	0.26	0.29	0.25	0.14	0.04	0.00	0.00	0
unsupportive	0.24	0.28	0.28	0.16	0.04	0.00	0.00	0
negative	0.15	0.19	0.24	0.22	0.12	0.05	0.02	0
heterosexism	0.05	0.08	0.14	0.25	0.22	0.14	0.12	0
harassed	0.16	0.21	0.24	0.24	0.11	0.02	0.02	0

Both alpha (.58) and the overall inter-item correlations (average_r; .22) decrease. This decrease in alpha is an example of how sample size can effect the result.

Examining item-level statistics, we do see greater variability (.15 to .55) in the corrected item-total correlations (r.drop). What might this mean?

- The item we dropped (*unresponsive*) may be clustering with *cold* and *unsupportive* in a subordinate factor (think subscale).
- Although item analysis is more of a tool in assessing reliability, the statistical information that *unresponsive* provided may broaden the construct definition (definitions are a concern of *validity*) of perceptions of campus climate such that it is necessary to ground/anchor *cold* and *unsupportive*.

Tentative conclusion: there is evidence that this is not a unidimensional measure.

```
RESPalpha <- psych::alpha(ResponseT1)
RESPalpha
```

Reliability analysis
Call: psych::alpha(x = ResponseT1)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.78	0.79	0.72	0.56	3.8	0.015	2.7	1	0.58

lower alpha	upper	95% confidence boundaries
0.75	0.78	0.81

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
cold	0.64	0.65	0.48	0.48	1.8	0.028	NA	0.48	
unresponsive	0.74	0.74	0.58	0.58	2.8	0.021	NA	0.58	
unsupportedive	0.75	0.76	0.61	0.61	3.1	0.019	NA	0.61	

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
cold	646	0.86	0.87	0.78	0.69	2.4	1.2
unresponsive	646	0.86	0.83	0.69	0.61	3.2	1.4
unsupportedive	646	0.80	0.82	0.67	0.59	2.5	1.1

Non missing response frequency for each item

	1	2	3	4	5	6	7	miss
cold	0.26	0.29	0.25	0.14	0.04	0.00	0.00	0
unresponsive	0.15	0.20	0.24	0.23	0.13	0.04	0.01	0
unsupportedive	0.24	0.28	0.28	0.16	0.04	0.00	0.00	0

The alpha for the College Response subscale is .79 much higher than the .64 of the total-scale. Similarly the average inter-item correlation (*average_r*) is higher (.56 versus .23).

Examining the corrected item-total correlations indicates a strong correlation with the item (excluded) with the remaining variables (.59 to .69).

Let's look at the Stigma subscale.

```
STIGalpha <- psych::alpha(StigmaT1)
STIGalpha
```

Reliability analysis

Call: psych::alpha(x = StigmaT1)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r	
0.79	0.79	0.72		0.56	3.8	0.014	3.6	1.3	0.57

lower	alpha	upper	95% confidence boundaries
0.76	0.79	0.82	

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
negative	0.66	0.66	0.49		0.49	1.9	0.027	NA	0.49
heterosexism	0.72	0.72	0.57		0.57	2.6	0.022	NA	0.57
harassed	0.76	0.76	0.62		0.62	3.2	0.019	NA	0.62

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
negative	646	0.87	0.87	0.77	0.69	3.2	1.5
heterosexism	646	0.85	0.84	0.71	0.63	4.4	1.6
harassed	646	0.80	0.82	0.66	0.59	3.1	1.4

Non missing response frequency for each item

	1	2	3	4	5	6	7	miss
negative	0.15	0.19	0.24	0.22	0.12	0.05	0.02	0
heterosexism	0.05	0.08	0.14	0.25	0.22	0.14	0.12	0
harassed	0.16	0.21	0.24	0.24	0.11	0.02	0.02	0

The alpha for the Stigma subscale is also much higher (.79) than the alpha for the total-scale (.64). Similarly the average inter-item correlation (*average_r*) is higher (.56 versus .23).

Examining the corrected item-total correlations indicates a strong correlation with the item (excluded) with the remaining variables (.59 to .69).

In addition to needing strong inter-item correlations (which we just assessed) we want the individual items to correlate more strongly with themselves than with the other scale. Let's do that next.

7.6 Correlating Items with Other Scale Totals

In this first analysis, we will correlate the individual items of the College Response scale to the Stigma subscale score.

```
apaTables::apa.cor.table(dfSzyT1[c("cold", "unresponsive", "unsupportive", "StigmaT1")])
```

Means, standard deviations, and correlations with confidence intervals

Variable	M	SD	1	2	3
1. cold	2.42	1.17			
2. unresponsive	3.17	1.43	.61**		
			[.56, .65]		
3. unsupportive	2.48	1.14	.58**	.48**	
			[.53, .63]	[.42, .54]	
4. StigmaT1	3.57	1.28	-.02	.09*	-.02
			[-.10, .06]	[.01, .16]	[-.10, .05]

Note. M and SD are used to represent mean and standard deviation, respectively.

Values in square brackets indicate the 95% confidence interval.

The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014).

* indicates $p < .05$. ** indicates $p < .01$.

We want the items to have higher correlations with each other (.48 to .61) than with the StigmaT1 scale (-.02 to .09) with all three items). These are uncharacteristically low and are influenced by the simulation of raw data from factor loadings and the authors' excellent work in developing, evaluating, and retaining items that reflect two distinct factors.

Let's examine the individual items from the Stigma scale with the College Response subscale score.

```
apaTables::apa.cor.table(dfSzyT1[c("negative", "heterosexism", "harassed", "ResponseT1")])
```

Means, standard deviations, and correlations with confidence intervals

Variable	M	SD	1	2	3
1. negative	3.22	1.52			
2. heterosexism	4.41	1.61	.62**		
			[.57, .66]		
3. harassed	3.07	1.42	.57**	.49**	
			[.51, .62]	[.43, .55]	
4. ResponseT1	2.69	1.04	-.05	-.01	.13**
			[-.13, .03]	[-.09, .07]	[.05, .20]

Note. M and SD are used to represent mean and standard deviation, respectively.

Values in square brackets indicate the 95% confidence interval.

The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014).

* indicates $p < .05$. ** indicates $p < .01$.

Again, the inter-item correlations are strong (.49 to .62) while their correlation with the College Response scale is 0.00

7.7 Interpreting and Writing up the Results

Tabling these results can be really useful to present them effectively. As is customary in APA style tables, when the item is in bold, the value represents its relationship with its own factor. These values come from the corrected item-total (r_{drop}) values where the item is singled out and correlated with the remaining items in its subscale.

Item-Total Correlations of Items with their Own and Other Subscales

Variables	College Response	Stigma
cold	.69	-.02
unresponsive	.61	.09
unsupportive	.59	-.02
negative	-.05	.69
heterosexism	-.01	.63
harassed	.13	.59

Although I pitched this type of item-analysis as *reliability*, to some degree it assesses within-scale **convergent and discriminant validity** because we can see the item relates more strongly to members of its own scale (higher correlation coefficients indicate *convergence*) than to the subscale scores of the other scales. When this pattern occurs, we can argue that the items *discriminate* well.

Results

Item analyses were conducted on the six items hypothesized to assess perceptions of campus climate for members of the LGBTQ community. To assess the within-scale convergent and discriminant validity of the College Response and Stigma subscales, each item was correlated with its own scale (with the item removed) and with the other coping scale (see Table 1). In all cases, items were more highly correlated with their own scale than with the other scale. Coefficient alphas were .79, .79, and .64 for the College Response, Stigma, and total-scale scores, respectively. We concluded that the within-scale convergent and discriminant validity of this measure is strong.

For your consideration: You are at your dissertation defense. For one of your measures, the Cronbach's alpha is .45. A committee member asks, "So why was the alpha coefficient so low?" On the basis of what you have learned in this lesson, how do you respond?

7.8 Practice Problems

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. For this lesson, please locate item-level data for a scale that has the potential for at least two subscales and a total-scale score. Ideally, you selected such data for practice from the prior lesson. Then, please examine the following:

- produce alpha coefficients, average inter-item correlations, and corrected item-total correlations for the total and subscales, separately
- produce correlations between the individual items of one subscale and the subscale scores of all other scales
- draft an APA style results section with an accompanying table.

In my example, there were only two subscales. If you have more, you will need to compare each subscale with all the others. For example, if you had three subscales: A, B, C, you would need to compare A/B, B/C, and A/C.

7.8.1 Problem #1: Play around with this simulation.

Copy the script for the simulation and then change (at least) one thing in the simulation to see how it impacts the results.

If item analysis is new to you, copy the script for the simulation and then change (at least) one thing in the simulation to see how it impacts the results. Perhaps you just change the number in “set.seed(210827)” from 210827 to something else. Your results should parallel those obtained in the lecture, making it easier for you to check your work as you go.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Report alpha coefficients and average inter-item correlations for the total and subscales	5	_____
3. Produce and interpret corrected item-total correlations for total and subscales, separately	5	_____
4. Produce and interpret correlations between the individual items of a given subscale and the subscale scores of all other subscales	5	_____
5. APA style results section with table	5	_____
6. Explanation to grader	5	_____
Totals	30	_____

7.8.2 Problem #2: Use raw data from the ReCentering Psych Stats survey on Qualtrics.

The script below pulls live data directly from the ReCentering Psych Stats survey on Qualtrics. As described in the [Scrubbing and Scoring chapters](#) of the ReCentering Psych Stats Multivariate Modeling volume, the Perceptions of the LGBTQ College Campus Climate Scale (?) was included

(LGBTQ) and further adapted to assess perceptions of campus climate for Black students (BLst), non-Black students of color (nBSoc), international students (INTst), and students with disabilities (wDIS). Consider conducting the analyses on one of these scales or merging them together and imagining subscales according to identity/group (LGBTQ, Black, non-Black, disability, international) or College Response and Stigma across the different groups.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Report alpha coefficients and average inter-item correlations for the total and subscales	5	_____
3. Produce and interpret corrected item-total correlations for total and subscales, separately	5	_____
4. Produce and interpret correlations between the individual items of a given subscale and the subscale scores of all other subscales	5	_____
5. APA style results section with table	5	_____
6. Explanation to grader	5	_____
Totals	30	_____

```
library(tidyverse)
#only have to run this ONCE to draw from the same Qualtrics account...but will need to get diff
library(qualtRics)
#qualtrics_api_credentials(api_key = "mUgPMMySYkiWpMFkwHale1QE5HNmh5LRUaA8d9PDg",
#                           base_url = "spupsych.az1.qualtrics.com", overwrite = TRUE, install = TRUE)
QTRX_df <- qualtRics::fetch_survey(surveyID = "SV_b2cClqAllGQ6nLU", time_zone = NULL, verbose =
climate_df <- QTRX_df%>%
  select('BLst_1', 'BLst_2', 'BLst_3', 'BLst_4', 'BLst_5', 'BLst_6',
         'nBSoc_1', 'nBSoc_2', 'nBSoc_3', 'nBSoc_4', 'nBSoc_5', 'nBSoc_6',
         'INTst_1', 'INTst_2', 'INTst_3', 'INTst_4', 'INTst_5', 'INTst_6',
         'wDIS_1', 'wDIS_2', 'wDIS_3', 'wDIS_4', 'wDIS_5', 'wDIS_6',
         'LGBTQ_1', 'LGBTQ_2', 'LGBTQ_3', 'LGBTQ_4', 'LGBTQ_5', 'LGBTQ_6')
#Item numbers are supported with the following items:
#_1 "My campus unit provides a supportive environment for ___ students"
#_2 "____ is visible in my campus unit"
#_3 "Negative attitudes toward persons who are ___ are openly expressed in my campus unit."
#_4 "My campus unit is unresponsive to the needs of ___ students."
#_5 "Students who are___ are harassed in my campus unit."
#_6 "My campus unit is cold and uncaring toward ___ students."
#Item 1 on each subscale should be reverse coded.
#The College Response scale is composed of items 1, 4, 6,
#The Stigma scale is composed of items 2,3, 5
```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```
#write the simulated data as a .csv
#write.table(climate_df, file="climate_df.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#climate_df <- read.csv ("climate_df.csv", header = TRUE)

#to save the df as an .rds (think "R object") file on your computer; it should save in the same
#saveRDS(climate_df, "climate_df.rds")
#bring back the simulated dat from an .rds file
#climate_df <- readRDS("climate_df.rds")
```

7.8.3 Problem #3: Try something entirely new.

Complete the same steps using data for which you have permission and access. This might be data of your own, from your lab, simulated from an article, or located on an open repository.

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Report alpha coefficients and average inter-item correlations for the total and subscales	5	_____
3. Produce and interpret corrected item-total correlations for total and subscales, separately	5	_____
4. Produce and interpret correlations between the individual items of a given subscale and the subscale scores of all other subscales	5	_____
5. APA style results section with table	5	_____
6. Explanation to grader	5	_____
Totals	30	_____

7.9 Bonus Reel:

For our interpretation and results, I created the table by manually typing in the results. Since there were only two subscales, this was easy. However, it can be a very useful skill (and prevent typing errors) by leveraging R's capabilities to build a table.

The script below

- Creates a correlation matrix of the items of each scale and correlates them with the “other” subscale, separately for both subscales.
- Extracts the r.drop from each subscale
- Joins (adds more variables) the analyses across the corrected item-total and item-other subscale analyses
- Binds (adds more cases) the two sets of items together

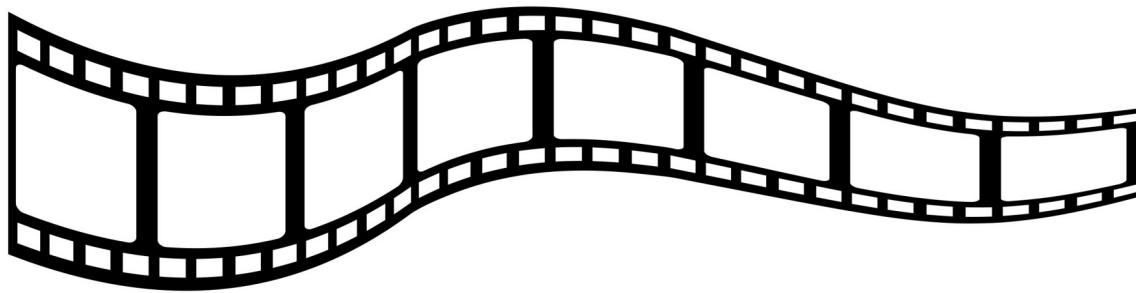


Figure 7.1: Image of a filmstrip

```

Resp_othR <- psych::corr.test(dfSzyT1[c("negative", "heterosexism", "harassed", "ResponseT1")])
Resp_othR <- as.data.frame(Resp_othR$r)#extracts the "r" matrix and makes it a df
Resp_othR$Items <- c("negative", "heterosexism", "harassed", "ResponseT1")#Assigning names to
Resp_othR <- Resp_othR[!Resp_othR$Items == "ResponseT1",]#Removing the subscale score as a a row
Resp_othR[, 'StigmaT1'] <- NA #We need a column for this to bind the items later
Resp_othR <- dplyr::select(Resp_othR, Items, ResponseT1, StigmaT1) #All we need is the item na
RESPalpha <- as.data.frame(RESPalpha$item.stats)#Grabbing the alpha objet we created earlier a
RESPalpha$Items <- c("cold", "unresponsive", "unsupportive")

Stig_othR <- psych::corr.test(dfSzyT1[c("cold", "unresponsive", "unsupportive", "StigmaT1")])#
Stig_othR <- as.data.frame(Stig_othR$r)#extracts the "r" matrix and makes it a df
Stig_othR$Items <- c("cold", "unresponsive", "unsupportive", "StigmaT1")#Assigning names to the
Stig_othR <- Stig_othR[!Stig_othR$Items == "StigmaT1",]#Removing the subscale score as a a row
Stig_othR[, 'ResponseT1'] <- NA #We need a column for this to bind the items later
Stig_othR <- dplyr::select(Stig_othR, Items, ResponseT1, StigmaT1) #All we need is the item na
STIGalpha <- as.data.frame(STIGalpha$item.stats)#Grabbing the alpha objet we created earlier a
STIGalpha$Items <- c("negative", "heterosexism", "harassed")

#Combining these four dfs
ResponseStats <- full_join(RESPalpha, Stig_othR, by = "Items")
ResponseStats$ResponseT1 <- ResponseStats$r.drop
ResponseStats <- dplyr::select(ResponseStats, Items, ResponseT1, StigmaT1)
StigmaStats <- full_join(STIGalpha, Resp_othR, by = "Items")
StigmaStats$StigmaT1 <- StigmaStats$r.drop
StigmaStats <- dplyr::select(StigmaStats, Items, ResponseT1, StigmaT1)
ItemAnalyses <- rbind(ResponseStats, StigmaStats)
ItemAnalyses

```

	Items	ResponseT1	StigmaT1
1	cold	0.691836225	-0.01933630
2	unresponsive	0.611240042	0.08696749
3	unsupportive	0.586410276	-0.02454147

```
4    negative -0.051291879  0.68802067
5 heterosexism -0.009656154  0.62875558
6    harassed  0.128551812  0.58709766
```

#Writing them to a .csv file allows post-r formatting

```
write.csv(ItemAnalyses, file = "LGBTQ_Climate_ItemAnalyses.csv", sep = ",", row.names=TRUE, col
```

```
Warning in write.csv(ItemAnalyses, file = "LGBTQ_Climate_ItemAnalyses.csv", :
attempt to set 'col.names' ignored
```

```
Warning in write.csv(ItemAnalyses, file = "LGBTQ_Climate_ItemAnalyses.csv", :
attempt to set 'sep' ignored
```


EXPLORATORY *FACTOR* ANALYSIS

The next two lessons are devoted to exploratory *factor* analysis. The two approaches are principal components analysis (PCA) and principal axis factoring (PAF). In truth, only PAF is considered factor analysis. I will explain why in the lesson.

These approaches are loosely termed *exploratory* because the statistical process (not the researcher) produces the factor (think scale or subscale) and identifies which items belong to it. This is contrasted with *confirmatory* approaches (which use structural equation modeling) where the researcher assigns items to factors and analyzes the goodness of fit.

Chapter 8

Principal Components Analysis

[Screencasted Lecture Link](#)

In this lesson on principal components analysis (PCA) I provide an introduction to the exploratory factor analysis (EFA) arena. We will review the theoretical and technical aspects of PCA, we will work through a research vignette, and then consider the relationship of PCA to item analysis and reliability coefficients.

Please note, although PCA is frequently grouped into EFA techniques, it is *exploratory* but it is not *factor analysis*. We'll discuss the difference in the lecture.

8.1 Navigating this Lesson

There is about two hours of lecture. If you work through the materials with me, I would be plan for an additional hour-and-a-half.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

8.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Distinguish between PCA and PAF on several levels:
 - which path diagram represents each best
 - keywords associated with each: factor loadings, linear components, describe versus explain.
- Recognize/define an identity matrix – what test would you use to diagnose it?
- Recognize/define multicollinearity and singularity – what test would you use to diagnose it?

- Describe the pattern of “loadings” (i.e., the relative weights of an item on its own scale compared to other scales) that supports the structure of the instrument.
- Compare the results from item analysis and PCA.

8.1.2 Planning for Practice

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. The least complex is to change the random seed in the research and rework the problem demonstrated in the lesson. The results *should* map onto the ones obtained in the lecture.

The second option involves utilizing one of the simulated datasets available in this OER. Szymanski and Bissonette’s (?)Perceptions of the LGBTQ College Campus Climate Scale: Development and psychometric evaluation was used as the research vignette for the validity, reliability, and item analysis lessons. Although I switched vignettes, the Szymanski and Bissonette example is ready for PCA.

As a third option, you are welcome to use data to which you have access and is suitable for PCA. These could include other vignettes from this OER, other simulated data, or your own data (presuming you have permission to use it). In either case, please plan to:

- Properly format and prepare the data.
- Conduct diagnostic tests to determine the suitability of the data for PCA.
- Conducting tests to guide the decisions about number of components to extract.
- Conducting orthogonal and oblique extractions (at least two each with different numbers of components).
- Selecting one solution and preparing an APA style results section (with table and figure).
- Compare your results in light of any other psychometrics lessons where you have used this data.

8.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (when possible, linked) in the text with complete citations in the reference list.

- Revelle, William. (n.d.). Chapter 6: Constructs, components, and factor models. In *An introduction to psychometric theory with applications in R*. Retrieved from <https://personality-project.org/r/book/#chapter6>
 - pp. 145 to 150 (we’ll continue with the rest in the next lecture). Stop at “6.2 Exploratory Factor Analysis.”
 - A simultaneously theoretical review of psychometric theory while working with R and data to understand the concepts.
- Revelle, W. (2019). *How To: Use the psych package for Factor Analysis and data reduction*.
 - pp. 13 through 24 provide technical information about what we are doing
- Dekay, Nicole (2021). Quick Reference Guide: The statistics for psychometrics <https://www.humanalysts.com/quick-reference-guide-the-statistics-for-psychometrics>

- Lewis, J. A., & Neville, H. A. (2015). Construction and initial validation of the Gendered Racial Microaggressions Scale for Black women. *Journal of Counseling Psychology*, 62(2), 289–302. <https://doi.org/10.1037/cou0000062>
 - Our research vignette for this lesson.

8.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(psych)){install.packages("psych")}
#if(!require(tidyverse)){install.packages("tidyverse")}
#if(!require(MASS)){install.packages("MASS")}
#if(!require(sjstats)){install.packages("sjstats")}
#if(!require(apaTables)){install.packages("apaTables")}
#if(!require(qualtRics)){install.packages("qualtRics")}
```

8.2 Exploratory Principal Components Analysis

The psychometric version of *parsimony* is seen in our attempt to *describe* (components) or to *explain* (factors) in the relationships between many observed variables in terms of a more limited set of components or latent factors.

That is, we are trying to:

- understand the structure of a set of variables,
- construct a questionnaire to measure an underlying latent variable, and
- reduce a data set to a more manageable size (think fewer columns...scale scores) while retaining as much of the information as possible

8.2.1 Some Framing Ideas (in very lay terms)

Exploratory versus *confirmatory* factor analysis.

- Both exploratory and confirmatory approaches to components/factor analysis are used in scale construction. Think of “scales” as being interchangeable with “factors” and “components.”
 - That said, “factors” and “components” are not interchangeable terms.
- **Exploratory:** Even though we may have an a priori model in mind, we *explore* the structure of the items by using diagnostics (KMO, Barlett’s determinant), factor extraction, and rotation to determine the number of scales (i.e., components or factors) that exist within the raw data or correlation matrix. The algorithms (including matrix algebra) determine the relationship of each item to its respective scales (i.e., components or factors).

- **Confirmatory:** Starting with an a priori theory, we specify the structure (i.e., number and levels of factors) and which items belong to factors. We use structural equation modeling as the framework. And we only work with “factors.” We evaluate the quality of the model with a number of fit indices.

Within the *exploratory* category we will focus on two further distinctions (there are even more): component analysis (principal components, or PCA) and principal axis factoring (PAF; one of the approaches that is commonly termed exploratory factor analysis, or EFA). In this first lecturette we focus on the differences between PCA and EFA.

- **Option #1/Component model:** PCA approximates the correlation matrix in terms of the product of components where each is a weighted linear sum of the variables. In the figure below, note how the arrows in the Components Analysis (a *path* model) point from variables to the component. Perhaps an oversimplification, think of each of these as a predictor variable contributing to an outcome.
- **Option #2/Factor model:** EFA (and in the next lesson, PAF/principal axis factoring) approximates the correlation matrix by the product of the two factors; this approach presumes that the factors are the causes (rather than as consequences). In the figure below, note how the arrows in the Factor Analysis model (a *structural* model) point from latent variable (or factor) to the observed variables (items). Factor analysis has been termed *causal modeling* because the latent variables are theorized to cause the responses to the individual items. There are other popular approaches, including parallel analysis (which is what the authors used in this lesson’s research vignette).

Well-crafted figures provide important clues to the analyses. In structural models, rectangles and squares indicate the presence of *observed* (also called *manifest*) variables. These are variables that have a column in the dataset. In our particular case, they are the responses to the 25 items in the GRMS.

Circles or ovals represent latent variables or factors. These were never raw data, but are composed of the relations of variables that were collected. They are more complex than mean or sum scores. Rather, they represent what the variables that represent them share in common.

Our focus today is on the principal component analysis (PCA) approach to scale construction.

8.3 PCA Workflow

Below is a screenshot of the workflow. The original document is located in the [Github site](#) that hosts the ReCentering Psych Stats: Psychometrics OER.

Steps in the process include:

- Creating an items only dataframe where any items are scaled in the same direction (e.g., negatively worded items are reverse-scored).
- Conducting tests that assess the statistical assumptions of PCA to ensure that the data is appropriate for PCA.

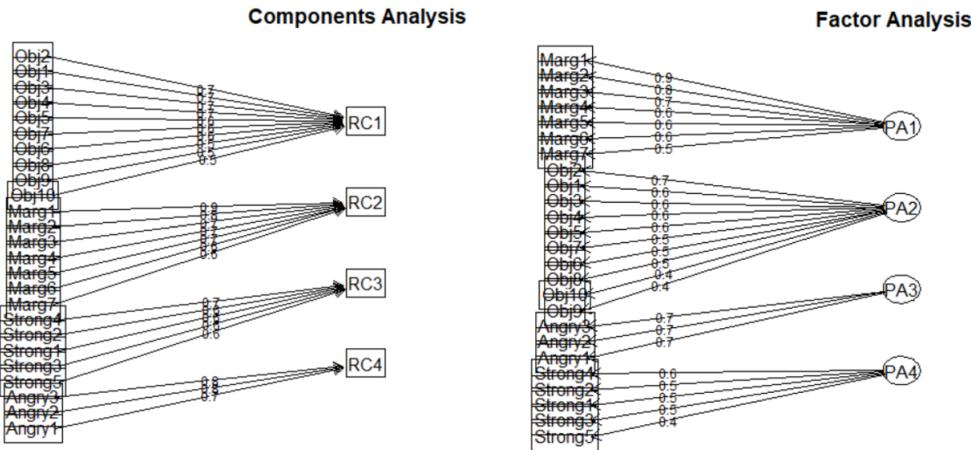


Figure 8.1: Comparison of path models for PCA and EFA for our research vignette

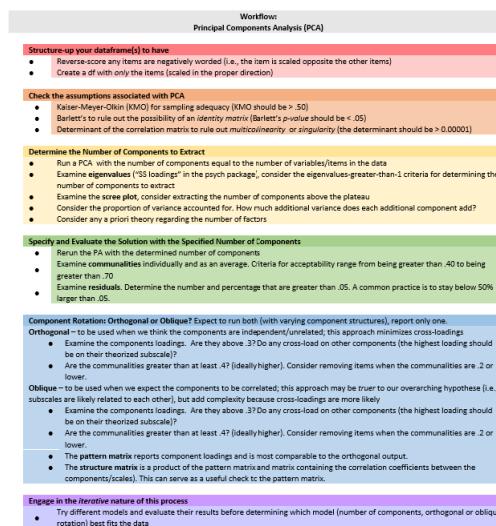


Figure 8.2: Image of the workflow for PCA

- Determining the number of components (think “subscales”) to extract.
- Conducting the component extraction – this process will likely occur iteratively,
 - exploring orthogonal (uncorrelated/independent) and oblique (correlated) components, and
 - changing the number of components to extract

Because the intended audience for the ReCentering Psych Stats OER is the scientist-practitioner-advocate, this lesson focuses on the workflow and decisions. As you might guess, the details of PCA can be quite complex. Some important notions to consider that may not be obvious from this lesson, are these:

- The values of component loadings are directly related to the correlation matrix.
 - Although I do not explain this in detail, nearly every analytic step attempts to convey this notion by presenting equivalent analytic options using the raw data and correlation matrix.
- PCA is about *dimension reduction* – our goal is fewer components (think subscales) than there are items.
 - In this lesson’s vignette there are 25 items on the scale and we will have 4 subscales.
- Principal component analysis is *exploratory*, but it is not “factor analysis.”
- Matrix algebra (e.g., using the transpose of a matrix, multiplying matrices together) plays a critical role in the analytic solution.

8.4 Research Vignette

This lesson’s research vignette emerges from Lewis and Neville’s Gendered Racial Microaggressions Scale for Black Women (?). The article reports on two separate studies that comprised the development, refinement, and psychometric evaluation of two parallel versions (stress appraisal, frequency) of the scale. We simulate data from the final construction of the stress appraisal version as the basis of the lecture.

Lewis and Neville (?) reported support for a total scale score (25 items) and four subscales. Below, I list the four subscales, their number of items, and a single example item. At the outset, let me provide a content warning. For those who hold this particular identity (or related identities) the content in the items may be upsetting. In other lessons, I often provide a variable name that gives an indication of the primary content of the item. In the case of the GRMS, I will simply provide an abbreviation of the subscale name and its respective item number. This will allow us to easily inspect the alignment of the item with its intended factor, and hopefully minimize discomfort. If you are not a member of this particular identity, I encourage you to learn about these microaggressions by reading the article in its entirety. Please do not ask members of this group to explain why these microaggressions are harmful or ask if they have encountered them. The four factors, number of items, and sample item are as follows:

- Assumptions of Beauty and Sexual Objectification

- 10 items
- “Objectified me based on physical features.”
- Abbreviated in the simulated data as “Obj#”
- Silenced and Marginalized
 - 7 items
 - “Someone has tried to ‘put me in my place.’”
 - Abbreviated in the simulated data as “Marg#”
- Strong Black Woman Stereotype
 - 5 items
 - “I have been told that I am too assertive.”
 - Abbreviated in the simulated data as “Strong#”
- Angry Black Woman Stereotype
 - 3 items
 - “Someone accused me of being angry when speaking calm.”
 - Abbreviated in the simulated data as “Angry#”

Below, I walk through the data simulation. This is not an essential portion of the lesson, but I will lecture it in case you are interested. None of the items are negatively worded (relative to the other items), so there is no need to reverse-score any items.

```
set.seed(210921)
GRMSmat <- matrix(c(.69, .69, .60, .59, .55, .55, .54, .50, .41, .41, .04, -.15, .06, .12, .20
                     .07, -.07, .00, .07, -.18, .22, .23, -.01, .03, .02, .93, .81, .69, .67, .61
                     -.08, -.08, .00, .06, .16, -.06, .08, .16, .22, .23, -.04, .01, -.05, -.11,
                     -.06, .03, .16, .01, .05, .09, -.08, -.06, .07, -.03, -.08, .18, .03, .06, .09
rownames(GRMSmat) <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9",
colnames(GRMSmat) <- c("Objectified", "Marginalized", "Strong", "Angry") #component (subscale)
GRMSCorMat <- GRMSmat %*% t(GRMSmat) #create the correlation matrix via some matrix algebra
diag(GRMSCorMat) <- 1
#SzyCorMat #prints the correlation matrix
GRMS_M <- c(1.78, 1.85, 1.97, 1.93, 2.01, 1.76, 1.91, 2.22, 1.83, 1.88, 2,
GRMS_SD <- c(1.11, 1.23, 0.97, 0.85, 1.19, 1.32, 1.04, 0.98, 1.01, 1.03, 1.01
GRMSCovMat <- GRMS_SD %*% t(GRMS_SD) * GRMSCorMat #creates a covariance matrix (with more matrix)
dfGRMS <- as.data.frame(round(MASS::mvrnorm(n=259, mu = GRMS_M, Sigma = GRMSCovMat, empirical = TRUE))
dfGRMS[dfGRMS>5] <- 5 #restricts the upperbound of all variables to be 5 or less
dfGRMS[dfGRMS<0] <- 0 #restricts the lowerbound of all variable to be 0 or greater
#colMeans(dfGRMS) #displays column means

#Below is code if you would like an ID number for each case. Especially at first, the ID numbers
#library(tidyverse)
#dfGRMS <- dfGRMS %>% dplyr::mutate(ID = row_number()) #add ID to each row
#dfGRMS <- dfGRMS %>% dplyr::select(ID, everything())#moving the ID number to the first column;
```

Let's take a quick peek at the data to see if everthing looks right.

```
psych::describe(dfGRMS)
```

```
Strong2 0.06
Strong3 0.05
Strong4 0.07
Strong5 0.05
Angry1 0.07
Angry2 0.07
Angry3 0.07
```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do). If you save the .csv file and bring it back in, you will lose any formatting (e.g., ordered factors will be interpreted as character variables).

```
#write the simulated data as a .csv
#write.table(dfGRMS, file="dfGRMS.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#dfGRMS <- read.csv ("dfGRMS.csv", header = TRUE)
```

An .rds file preserves all formatting to variables prior to the export and re-import. For the purpose of this chapter, you don’t need to do either. That is, you can re-simulate the data each time you work the problem.

```
#to save the df as an .rds (think "R object") file on your computer; it should save in the same directory as the .csv file
#saveRDS(dfGRMS, "dfGRMS.rds")
#bring back the simulated dat from an .rds file
#dfGRMS <- readRDS("dfGRMS.rds")
```

8.5 Working the Vignette

Below we will create a correlation matrix of our items. Whether we are conducting PCA or PAF, the *dimension-reduction* we are seeking is looking for clusters of correlated items in the *R*-matrix. Essentially, these are (?):

- statistical entities that can be plotted as classification axes where coordinates of variables along each axis represent the strength of the relationship between that variable to each component/factor.
- mathematical equations, resembling regression equations, where each variable is represented according to its relative weight

PCA in particular establishes which linear components exist within the data and how a particular variable might contribute to that component.

Here is the correlation matrix of our items. It would be quite a daunting exercise to visually inspect this and manually cluster the correlations of items.

```
GRMSmatrix<-cor(dfGRMS) #correlation matrix created and saved as object
round(GRMSmatrix, 2)
```

	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6	Obj7	Obj8	Obj9	Obj10	Marg1	Marg2
Obj1	1.00	0.41	0.38	0.33	0.34	0.34	0.37	0.35	0.25	0.24	0.11	0.00
Obj2	0.41	1.00	0.43	0.40	0.40	0.34	0.35	0.29	0.28	0.24	-0.02	-0.13
Obj3	0.38	0.43	1.00	0.36	0.34	0.31	0.30	0.28	0.22	0.24	0.01	-0.03
Obj4	0.33	0.40	0.36	1.00	0.32	0.32	0.32	0.25	0.27	0.28	0.06	-0.03
Obj5	0.34	0.40	0.34	0.32	1.00	0.23	0.26	0.26	0.27	0.25	-0.14	-0.18
Obj6	0.34	0.34	0.31	0.32	0.23	1.00	0.31	0.24	0.19	0.16	0.20	0.16
Obj7	0.37	0.35	0.30	0.32	0.26	0.31	1.00	0.26	0.19	0.26	0.23	0.11
Obj8	0.35	0.29	0.28	0.25	0.26	0.24	0.26	1.00	0.21	0.22	0.00	-0.09
Obj9	0.25	0.28	0.22	0.27	0.27	0.19	0.19	0.21	1.00	0.22	0.07	0.01
Obj10	0.24	0.24	0.24	0.28	0.25	0.16	0.26	0.22	0.22	1.00	0.06	-0.03
Marg1	0.11	-0.02	0.01	0.06	-0.14	0.20	0.23	0.00	0.07	0.06	1.00	0.66
Marg2	0.00	-0.13	-0.03	-0.03	-0.18	0.16	0.11	-0.09	0.01	-0.03	0.66	1.00
Marg3	0.05	-0.01	0.02	0.04	-0.11	0.16	0.12	-0.03	0.07	0.02	0.62	0.50
Marg4	0.17	0.05	0.10	0.07	-0.03	0.21	0.21	0.05	0.05	0.07	0.58	0.47
Marg5	0.20	0.07	0.11	0.13	0.00	0.21	0.22	0.04	0.10	0.12	0.55	0.41
Marg6	0.00	-0.09	-0.06	0.03	-0.11	0.06	0.16	0.03	0.05	0.13	0.51	0.43
Marg7	-0.14	-0.22	-0.09	-0.13	-0.17	0.00	-0.01	-0.10	-0.04	-0.04	0.42	0.42
Strong1	-0.05	-0.07	0.03	-0.02	0.11	-0.05	0.01	0.03	0.14	0.10	-0.04	0.05
Strong2	0.01	-0.06	0.01	0.04	0.13	-0.01	0.04	0.13	0.12	0.13	-0.10	-0.06
Strong3	0.04	0.05	0.07	0.14	0.15	0.02	0.10	0.10	0.13	0.14	-0.06	-0.09
Strong4	-0.11	-0.11	0.01	-0.04	0.06	-0.04	-0.05	0.05	0.11	0.09	-0.04	0.05
Strong5	0.02	0.00	-0.01	0.01	0.04	0.00	0.10	0.11	0.05	0.11	0.14	0.10
Angry1	0.11	0.15	0.20	0.15	0.11	0.15	0.01	0.08	0.08	0.03	-0.04	0.06
Angry2	-0.12	-0.04	0.08	-0.02	0.01	0.01	-0.07	-0.01	0.07	-0.05	-0.03	0.11
Angry3	-0.12	-0.09	0.03	-0.06	-0.05	0.02	-0.12	-0.11	0.05	-0.07	0.02	0.21
	Marg3	Marg4	Marg5	Marg6	Marg7	Strong1	Strong2	Strong3	Strong4	Strong5		
Obj1	0.05	0.17	0.20	0.00	-0.14	-0.05	0.01	0.04	-0.11	0.02		
Obj2	-0.01	0.05	0.07	-0.09	-0.22	-0.07	-0.06	0.05	-0.11	0.00		
Obj3	0.02	0.10	0.11	-0.06	-0.09	0.03	0.01	0.07	0.01	-0.01		
Obj4	0.04	0.07	0.13	0.03	-0.13	-0.02	0.04	0.14	-0.04	0.01		
Obj5	-0.11	-0.03	0.00	-0.11	-0.17	0.11	0.13	0.15	0.06	0.04		
Obj6	0.16	0.21	0.21	0.06	0.00	-0.05	-0.01	0.02	-0.04	0.00		
Obj7	0.12	0.21	0.22	0.16	-0.01	0.01	0.04	0.10	-0.05	0.10		
Obj8	-0.03	0.05	0.04	0.03	-0.10	0.03	0.13	0.10	0.05	0.11		
Obj9	0.07	0.05	0.10	0.05	-0.04	0.14	0.12	0.13	0.11	0.05		
Obj10	0.02	0.07	0.12	0.13	-0.04	0.10	0.13	0.14	0.09	0.11		
Marg1	0.62	0.58	0.55	0.51	0.42	-0.04	-0.10	-0.06	-0.04	0.14		
Marg2	0.50	0.47	0.41	0.43	0.42	0.05	-0.06	-0.09	0.05	0.10		
Marg3	1.00	0.42	0.40	0.35	0.34	-0.06	-0.09	-0.09	-0.02	0.07		
Marg4	0.42	1.00	0.44	0.28	0.27	-0.03	-0.08	-0.08	-0.04	0.06		
Marg5	0.40	0.44	1.00	0.25	0.26	-0.10	-0.08	-0.10	-0.06	0.01		
Marg6	0.35	0.28	0.25	1.00	0.29	0.10	0.09	0.10	0.11	0.23		
Marg7	0.34	0.27	0.26	0.29	1.00	0.09	0.08	-0.01	0.19	0.13		

Strong1	-0.06	-0.03	-0.10	0.10	0.09	1.00	0.26	0.24	0.34	0.19
Strong2	-0.09	-0.08	-0.08	0.09	0.08	0.26	1.00	0.27	0.31	0.21
Strong3	-0.09	-0.08	-0.10	0.10	-0.01	0.24	0.27	1.00	0.25	0.27
Strong4	-0.02	-0.04	-0.06	0.11	0.19	0.34	0.31	0.25	1.00	0.22
Strong5	0.07	0.06	0.01	0.23	0.13	0.19	0.21	0.27	0.22	1.00
Angry1	0.05	0.08	0.07	-0.16	0.08	0.09	-0.05	-0.10	0.06	-0.11
Angry2	0.02	0.06	-0.01	-0.08	0.19	0.22	0.05	-0.02	0.24	-0.02
Angry3	0.05	0.07	0.04	-0.09	0.20	0.18	0.03	-0.07	0.20	-0.05
	Angry1	Angry2	Angry3							
Obj1	0.11	-0.12	-0.12							
Obj2	0.15	-0.04	-0.09							
Obj3	0.20	0.08	0.03							
Obj4	0.15	-0.02	-0.06							
Obj5	0.11	0.01	-0.05							
Obj6	0.15	0.01	0.02							
Obj7	0.01	-0.07	-0.12							
Obj8	0.08	-0.01	-0.11							
Obj9	0.08	0.07	0.05							
Obj10	0.03	-0.05	-0.07							
Marg1	-0.04	-0.03	0.02							
Marg2	0.06	0.11	0.21							
Marg3	0.05	0.02	0.05							
Marg4	0.08	0.06	0.07							
Marg5	0.07	-0.01	0.04							
Marg6	-0.16	-0.08	-0.09							
Marg7	0.08	0.19	0.20							
Strong1	0.09	0.22	0.18							
Strong2	-0.05	0.05	0.03							
Strong3	-0.10	-0.02	-0.07							
Strong4	0.06	0.24	0.20							
Strong5	-0.11	-0.02	-0.05							
Angry1	1.00	0.44	0.43							
Angry2	0.44	1.00	0.47							
Angry3	0.43	0.47	1.00							

This correlation matrix is so big that you might wish to write code so that you can examine it in sections

```
#round(GRMSmatrix[,1:8], 2)
#round(GRMSmatrix[,9:16], 2)
#round(GRMSmatrix[,17:25], 2)
```

With component and factor analytic procedures we can analyze the data with either raw data or correlation matrix. Producing the matrix helps us see how this is a “structural” analysis. That is, we are trying to see if our more parsimonious extraction “reproduces” this original correlation matrix.

8.5.1 Three Diagnostic Tests to Evaluate the Appropriateness of the Data for Component-or-Factor Analysis

Below is a snip from the workflow to remind us where we are in the steps to PCA.

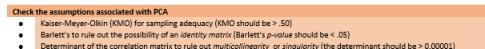


Figure 8.3: Image of an excerpt from the workflow

8.5.1.1 Is my sample adequate for PCA?

There have been a number of generic guidelines (some supported by analyses, some not) about “how big” the sample size should be:

- 10-15 participants per variable
- 10 times as many participants as variables (Nunnally, 1978)
- 5 and 10 participants per variable up to 300 (Kass & Tinsley, 1979)
- 300 (Tabachnick & Fidell, 2007)
- 1000 = excellent, 300 = good, 100 = poor (Comrey & Lee, 1992)

Of course it is more complicated. Monte Carlo studies have shown that:

- if factor loadings are large (~.6), the solution is reliable regardless of size
- if communalities are large (~.6), relatively small samples (~100) are sufficient, but when they are lower (well below .5), then larger samples (>500 are indicated).

The **Kaiser-Meyer-Olkin** index (KMO) is an index of *sampling adequacy* that can be used with the actual sample to let us know if the sample size is sufficient relative to the statistical characteristics of the data. If it is below the threshold, we should probably collect more data to see if it can achieve a satisfactory value.

Kaiser’s 1974 recommendations were:

- bare minimum of .5
- values between .5 and .7 as mediocre
- values between .7 and .8 as good
- values above .9 are superb

Revelle has included a KMO test in the psych package. The function can use either raw or matrix data. Either way, the only variables in the matrix should be the items of interest. This means that everything else (e.g., total or subscale scores, ID numbers) should be removed.

```
psych::KMO(dfGRMS)
```

Kaiser-Meyer-Olkin factor adequacy
 Call: psych::KMO(r = dfGRMS)
 Overall MSA = 0.86
 MSA for each item =

Obj1	Obj2	Obj3	Obj4	Obj5	Obj6	Obj7	Obj8	Obj9	Obj10
0.89	0.88	0.90	0.90	0.91	0.92	0.92	0.90	0.88	0.90
Marg1	Marg2	Marg3	Marg4	Marg5	Marg6	Marg7	Strong1	Strong2	Strong3
0.83	0.88	0.91	0.91	0.91	0.87	0.90	0.79	0.79	0.80
Strong4	Strong5	Angry1	Angry2	Angry3					
0.79	0.81	0.72	0.75	0.75					

```
#psych::KMO(GRMSmatrix)
```

We examine the KMO values for both the overall matrix and the individual items.

At the matrix level, our $KMO = .86$, which falls into Kaiser's definition of *good*.

At the item level, the KMO should be $> .50$. Variables with values below $.50$ should be evaluated for exclusion from the analysis (or run the analysis with and without the variable and compare the difference). Because removing and adding variables impacts the KMO, be sure to re-evaluate the sampling adequacy if changes are made to the items (and/or sample size).

At the item level, our KMO values range between .72 (Angry1) and .92 (Obj6, Obj7).

Considering both item and matrix levels, we conclude that the sample size and the data are adequate for component (or factor) analysis.

8.5.1.2 Are there correlations among the variables that are large enough to be analyzed?

Bartlett's test lets us know if a matrix is an *identity matrix*. In an identity matrix all correlation coefficients (everything on the off-diagonal) would be 0.0 (and everything on the diagonal would be 1.0).

A significant Bartlett's (i.e., $p < .05$) tells that the R -matrix is not an identity matrix. That is, there are some relationships between variables that can be analyzed.

The `cortest.bartlett()` function is in the `psych` package and can be run either from the raw data or R matrix formats.

```
psych::cortest.bartlett(dfGRMS) #from the raw data
```

R was not square, finding R from data

```
$chisq  
[1] 1683.76
```

```
$df
[1] 300
```

```
#raw data produces the warning "R was not square, finding R from data." This means nothing oth
#psych::cortest.bartlett(GRMSmatrix, n = 259) #if using the matrix, must specify sample
```

Our Bartlett's test is significant: $\chi^2(300) = 1683.76, p < .001$. This supports a component-or-factor analytic approach for investigating the data.

8.5.1.3 Is there multicollinearity or singularity in my data?

The **determinant of the correlation matrix** should be greater than 0.00001 (that would be 4 zeros, then the 1). If it is smaller than 0.00001 then we may have an issue with *multicollinearity* (i.e., variables that are too highly correlated) or *singularity* (variables that are perfectly correlated).

The determinant function we use comes from base R. It is easiest to compute when the correlation matrix is the object. However, it is also possible to specify the command to work with the raw data.

```
det(GRMSmatrix)
```

```
[1] 0.001151581
```

```
#det(cor(dfGRMS))#if using the raw data
```

With a value of 0.00115, our determinant is greater than the 0.00001 requirement. If it were not, then we could identify problematic variables (i.e., those correlating too highly with others; those not correlating sufficiently with others) and re-run the diagnostic statistics.

Summary: Data screening were conducted to determine the suitability of the data for this analyses. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO; Kaiser, 1970) represents the ratio of the squared correlation between variables to the squared partial correlation between variables. KMO ranges from 0.00 to 1.00; values closer to 1.00 indicate that the patterns of correlations are relatively compact and that component analysis should yield distinct and reliable components (Field, 2012). In our dataset, the KMO value was .86, indicating acceptable sampling adequacy. The Barlett's Test of Sphericity examines whether the population correlation matrix resembles an identity matrix (Field, 2012). When the p value for the Bartlett's test is $< .05$, we are fairly certain we have clusters of correlated variables. In our dataset, $\chi^2(300) = 1683.76, p < .001$, indicating the correlations between items are sufficiently large enough for principal components analysis. The determinant of the correlation matrix alerts us to any issues of multicollinearity or singularity and should be larger than 0.00001. Our determinant was 0.00115 and, again, indicated that our data was suitable for the analysis.

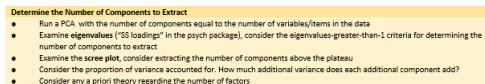


Figure 8.4: Image of an excerpt from the workflow

8.5.2 Principal Components Analysis

Below is a snip from the workflow to remind us where we are in the steps to PCA.

We can use the *principal()* function from the *psych* package with raw or matrix data.

We start by creating a principal components model that has the same number of components as there are variables in the data. This allows us to inspect the component's eigenvalues and make decisions about which to extract.

- Note, this is different than actual *factor* analysis where you *must* extract fewer factors than variables (e.g., extracting 18 [an arbitrary number] instead of 25).

```
#The numerous codes all result in the same. They simply swap out using the df or r-matrix, and
#pca1 <- psych::principal(GRMSmatrix, nfactors=25, rotate = "none") #using the matrix form of
#pca1 <- psych::principal(GRMSmatrix, nfactors=length(GRMSmatrix[,1]), rotate = "none") #using
#pca1 <- psych::principal(dfGRMS, nfactors=25, rotate="none") #using raw data and specifying #
pca1 <- psych::principal(dfGRMS, nfactors=length(dfGRMS), rotate="none")# using raw data and l
pca1
```

Principal Components Analysis

Call: *psych::principal(r = dfGRMS, nfactors = length(dfGRMS), rotate = "none")*
Standardized loadings (pattern matrix) based upon correlation matrix

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Obj1	0.55	0.40	-0.17	0.02	0.12	-0.17	-0.13	-0.12	-0.16	0.02	0.14	-0.01
Obj2	0.46	0.54	-0.15	0.14	0.02	0.18	-0.03	-0.09	0.03	0.10	-0.15	0.03
Obj3	0.48	0.43	0.04	0.21	0.13	0.08	0.11	0.01	-0.16	0.05	-0.20	-0.29
Obj4	0.50	0.42	-0.03	0.04	-0.15	0.25	0.12	0.27	0.24	-0.06	0.16	0.00
Obj5	0.33	0.57	0.13	0.02	-0.08	-0.01	-0.05	-0.01	-0.32	0.23	-0.19	0.04
Obj6	0.56	0.19	-0.09	0.15	0.17	0.12	-0.19	0.33	0.06	-0.27	-0.31	0.21
Obj7	0.59	0.22	-0.09	-0.13	0.11	0.09	0.15	-0.03	-0.21	-0.13	0.13	0.27
Obj8	0.38	0.40	0.06	-0.09	0.37	-0.44	-0.14	-0.15	0.32	-0.20	0.09	-0.25
Obj9	0.40	0.28	0.21	-0.01	-0.55	-0.05	-0.48	-0.17	0.23	0.12	0.03	0.06
Obj10	0.40	0.28	0.12	-0.19	-0.33	-0.24	0.62	-0.06	0.17	0.02	-0.09	0.04
Marg1	0.63	-0.62	-0.13	-0.10	-0.02	0.04	-0.03	-0.03	0.00	-0.01	0.02	-0.03
Marg2	0.47	-0.66	0.05	0.05	0.00	0.07	-0.06	-0.02	-0.03	-0.14	-0.03	0.05
Marg3	0.50	-0.53	-0.11	0.02	-0.08	0.06	-0.10	0.01	0.12	0.08	-0.08	-0.11
Marg4	0.57	-0.42	-0.10	0.08	0.10	-0.05	0.00	-0.09	-0.23	0.02	0.17	-0.11


```
Marg1   -0.43  1 -0.00000000000000133 3.0
Marg2    0.16  1 -0.00000000000000067 3.5
Marg3    0.10  1 -0.00000000000000067 4.9
Marg4    0.08  1 -0.000000000000000133 5.9
Marg5    0.08  1  0.00000000000000011 6.3
Marg6    0.07  1 -0.00000000000000089 8.3
Marg7    0.04  1  0.000000000000000133 5.5
Strong1   0.00  1  0.00000000000000200 4.3
Strong2  -0.02  1  0.00000000000000155 6.6
Strong3   0.01  1 -0.00000000000000022 7.2
Strong4  -0.02  1  0.00000000000000011 3.6
Strong5   0.02  1  0.00000000000000022 7.6
Angry1   -0.03  1  0.00000000000000044 3.6
Angry2   -0.02  1  0.00000000000000078 4.9
Angry3   -0.01  1  0.00000000000000011 5.2
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
SS loadings	4.04	3.61	2.41	2.09	0.88	0.84	0.80	0.78	0.75	0.71	0.68
Proportion Var	0.16	0.14	0.10	0.08	0.04	0.03	0.03	0.03	0.03	0.03	0.03
Cumulative Var	0.16	0.31	0.40	0.49	0.52	0.55	0.59	0.62	0.65	0.68	0.70
Proportion Explained	0.16	0.14	0.10	0.08	0.04	0.03	0.03	0.03	0.03	0.03	0.03
Cumulative Proportion	0.16	0.31	0.40	0.49	0.52	0.55	0.59	0.62	0.65	0.68	0.70
	PC12	PC13	PC14	PC15	PC16	PC17	PC18	PC19	PC20	PC21	PC22
SS loadings	0.67	0.65	0.63	0.61	0.59	0.56	0.56	0.54	0.52	0.51	0.50
Proportion Var	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Cumulative Var	0.73	0.76	0.78	0.81	0.83	0.85	0.87	0.90	0.92	0.94	0.96
Proportion Explained	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Cumulative Proportion	0.73	0.76	0.78	0.81	0.83	0.85	0.87	0.90	0.92	0.94	0.96
	PC23	PC24	PC25								
SS loadings	0.45	0.39	0.25								
Proportion Var	0.02	0.02	0.01								
Cumulative Var	0.97	0.99	1.00								
Proportion Explained	0.02	0.02	0.01								
Cumulative Proportion	0.97	0.99	1.00								

Mean item complexity = 5.7

Test of the hypothesis that 25 components are sufficient.

The root mean square of the residuals (RMSR) is 0
with the empirical chi square 0 with prob < NA

Fit based upon off diagonal values = 1

The total variance for a particular variable will have two components: some of it will be shared with other variables (common variance, h²) and some of it will be specific to that measure (unique variance, u²). Random variance is also specific to one item, but not reliably so.

We can examine this most easily by examining the matrix (second screen).

The columns PC1 thru PC25 are the (uninteresting at this point) unrotated loadings. PC stands for “principal component.” Although these don’t align with the specific items, at this point in the procedure, there are as many components as variables.

Communalities are represented as h^2 . These are the proportions of common variance present in the variables. A variable that has no specific (or random) variance would have a communality of 1.0. If a variable shares none of its variance with any other variable its communality would be 0.0.

Because we extracted the same number components as variables, they all equal 1.0. That is we have explained all the variance in each variable. When we specify fewer components, the value of the communalities will decrease.

**Uniquenesses* are represented as $u2$. These are the amount of unique variance for each variable. They are calculated as $1 - h^2$ (or 1 minus the communality). Technically (at this point in the analysis where we have an equal number of components as items), they should all be zero, but the *psych* package is very quantsy and decimals are reported to the 15th and 16th decimal places! (hence the u2 for Q1 is -0.0000000000000028865799).

The final column, *com*, represents *item complexity*. This is an indication of how well an item reflects a single construct. If it is 1.0 then the item loads only on one component, if it is 2.0, it loads evenly on two components, and so forth. For now, we can ignore this. *I mostly wanted to reassure you that “com” is not “communality”; h^2 is communality.*

Let’s switch to the first screen of output.

Eigenvalues are displayed in the row called *SS loadings* (i.e., the sum of squared loadings). They represent the variance explained by the particular linear component. PC1 explains 4.04 units of variance (out of a possible 25, the total of components). As a proportion, this is $4.04/25 = 0.16$ (reported in the *Proportion Var* row).

```
4.04/25
```

```
[1] 0.1616
```

Note:

- *Cumulative Var* is helpful in determining how many components we would like to retain to balance parsimony (few as possible) with the amount of variance we want to explain
- The eigenvalues are in descending order. If we were to use the *eigenvalue > 1.0* (aka, “Kaiser’s”) criteria to determine how many components to extract, we would select 4. Joliffe’s criteria was 0.7 (thus, we would select 10 components). Eigenvalues are only one criteria, let’s look at he scree plot.

Scree plot: We can gain another view of how many components to extract by creating a scree plot.

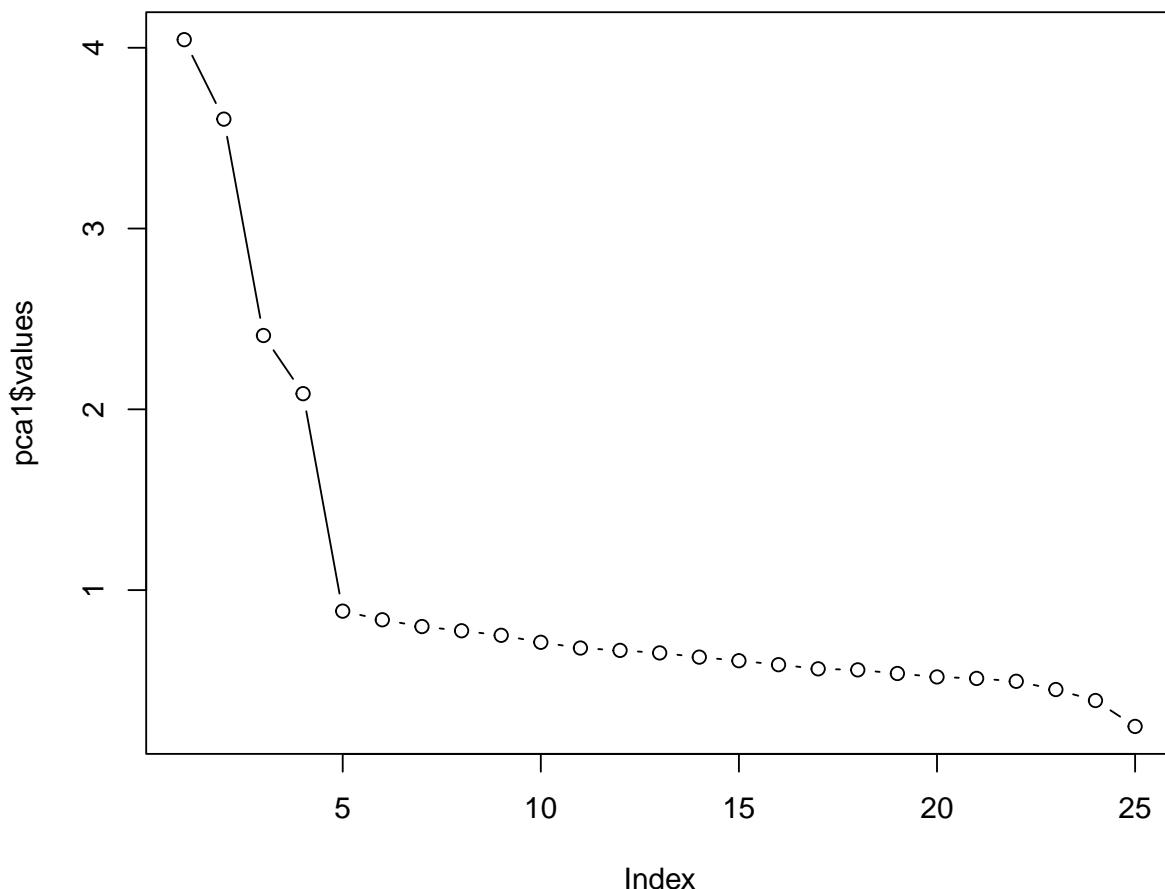
Eigenvalues are stored in the *pca1* object’s variable, “values”. We can see all the values captured by this object with the *names()* function:

```
names(pca1)
```

```
[1] "values"      "rotation"     "n.obs"       "communality"   "loadings"
[6] "fit"         "fit.off"      "fn"          "Call"        "uniquenesses"
[11] "complexity"  "chi"         "EPVAL"       "R2"          "objective"
[16] "residual"    "rms"         "factors"     "dof"         "null.dof"
[21] "null.model"  "criteria"    "STATISTIC"   "PVAL"        "weights"
[26] "r.scores"    "Vaccounted" "Structure"   "scores"
```

Plotting the eigenvalues produces a scree plot. We can use this to further gauge the number of factors we should extract.

```
plot(pca1$values, type="b") #type = "b" gives us "both" lines and points; type = "l" gives li
```



We look for the point of *inflection*. That is, where the baseline levels out into a plateau. There are four components above the plateau.

8.5.3 Specifying the Number of Components

Below is a snip from the workflow to remind us where we are in the steps to PCA.



Figure 8.5: Image of an excerpt from the workflow

Having determined the number of components, we re-run the analysis with this specification. Especially when researchers may not have a clear theoretical structure that guides the process, researchers may do this iteratively with varying numbers of factors. Lewis and Neville (?) examined solutions with 2, 3, 4, and 5 factors (they did a parallel *factor* analysis – we are still examining components).

```
#pca2 <- psych::principal(GRMSmatrix, nfactors=4, rotate="none")
pca2 <- psych::principal(dfGRMS, nfactors=4, rotate="none") #can copy prior script, but change
pca2
```

Principal Components Analysis

Call: psych::principal(r = dfGRMS, nfactors = 4, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix

	PC1	PC2	PC3	PC4	h2	u2	com
Obj1	0.55	0.40	-0.17	0.02	0.50	0.50	2.1
Obj2	0.46	0.54	-0.15	0.14	0.55	0.45	2.2
Obj3	0.48	0.43	0.04	0.21	0.46	0.54	2.4
Obj4	0.50	0.42	-0.03	0.04	0.42	0.58	2.0
Obj5	0.33	0.57	0.13	0.02	0.44	0.56	1.7
Obj6	0.56	0.19	-0.09	0.15	0.38	0.62	1.4
Obj7	0.59	0.22	-0.09	-0.13	0.43	0.57	1.5
Obj8	0.38	0.40	0.06	-0.09	0.32	0.68	2.2
Obj9	0.40	0.28	0.21	-0.01	0.28	0.72	2.4
Obj10	0.40	0.28	0.12	-0.19	0.30	0.70	2.5
Marg1	0.63	-0.62	-0.13	-0.10	0.80	0.20	2.1
Marg2	0.47	-0.66	0.05	0.05	0.66	0.34	1.8
Marg3	0.50	-0.53	-0.11	0.02	0.54	0.46	2.1
Marg4	0.57	-0.42	-0.10	0.08	0.52	0.48	2.0
Marg5	0.58	-0.35	-0.17	0.08	0.49	0.51	1.9
Marg6	0.40	-0.44	0.07	-0.42	0.54	0.46	3.0
Marg7	0.23	-0.59	0.28	0.01	0.48	0.52	1.7
Strong1	0.04	0.02	0.65	-0.13	0.45	0.55	1.1
Strong2	0.04	0.13	0.53	-0.35	0.42	0.58	1.9
Strong3	0.09	0.21	0.41	-0.45	0.42	0.58	2.5
Strong4	0.02	-0.04	0.70	-0.15	0.51	0.49	1.1
Strong5	0.18	-0.06	0.34	-0.47	0.38	0.62	2.2
Angry1	0.19	0.07	0.30	0.70	0.62	0.38	1.5
Angry2	0.03	-0.11	0.55	0.55	0.61	0.39	2.1
Angry3	0.02	-0.21	0.49	0.59	0.63	0.37	2.2

SS loadings	PC1	PC2	PC3	PC4
	4.04	3.61	2.41	2.09

```

Proportion Var      0.16 0.14 0.10 0.08
Cumulative Var     0.16 0.31 0.40 0.49
Proportion Explained 0.33 0.30 0.20 0.17
Cumulative Proportion 0.33 0.63 0.83 1.00

```

Mean item complexity = 2

Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is 0.05
 with the empirical chi square 360.58 with prob < 0.00000000015

Fit based upon off diagonal values = 0.94

Our eigenvalues/SS loadings remain the same. With 4 components, we explain 49% of the variance (we can see this in the “Cumulative Var” row).

Communality is the proportion of common variance within a variable. Principal components analysis assumes that all variance is common. Before extraction, all variance was set at 1.0, therefore, changing from 25 to 4 components will change this value (h^2) as well as its associated *uniqueness* (u^2), which is calculated as “1.0 minus the communality.”

The *communalities* (h^2) and *uniquenesses* (u^2) have changed.

Now we see that 50% of the variance associate with Obj1 is common/shared (the h^2 value).

Recall that we could represent this scale with all 25 items as components, but we want a more *parsimonious* explanation. By respecifying a smaller number of components, we lose some information. That is, the retained components (now 4) cannot explain all of the variance present in the data (as we saw, it explains about 50%, cumulatively). The amount of variance explained in each variable is represented by the communalities after extraction.

We can examine the communalities through the lens of Kaiser’s criterion (the eigenvalue > 1 criteria) to see if we think that “four” was a good number of components to extract.

Kaiser’s criterion is believed to be accurate if:

- when there are fewer than 30 variables (we had 25) and, after extraction, the communalities are greater than .70
 - looking at our data, only 1 communality (Marg1) is $> .70$, so, this does not support extracting four components
- when the sample size is greater than 250 (ours was 259) and the average communality is $> .60$
 - we can extract the communalities from our object and calculate the mean the average communality

Using the *names()* function again, we see that “communality” is available. Thus, we can easily calculate their mean. To get this value let’s first examine the possible contents of the object we created from this PCA analysis by asking for its names.

```
names(pca2)
```

```
[1] "values"      "rotation"     "n.obs"       "communality"   "loadings"
[6] "fit"         "fit.off"      "fn"          "Call"        "uniquenesses"
[11] "complexity"  "chi"         "EPVAL"       "R2"          "objective"
[16] "residual"    "rms"         "factors"     "dof"         "null.dof"
[21] "null.model"  "criteria"    "STATISTIC"   "PVAL"        "weights"
[26] "r.scores"    "Vaccounted" "Structure"   "scores"
```

We see that it includes communalities. Thus, we can easily calculate their mean.

```
mean(pca2$communality)
```

```
[1] 0.4857967
```

```
#sum(pca2$communality) #checking my work by calculating the sum and dividing by 25
#12.14492/25
```

We see that the average communality is 0.48. These two criteria would suggest that we may not have the best solution. That said (in our defense):

- We used the scree plot as a guide and it was very clear.
- We have an adequate sample size and that was supported with the KMO.
- Are the number of components consistent with theory? We have not yet inspected the component loadings. This will provide us with more information.

We could do several things:

- re-run with a different number of components (recall Lewis and Neville (?) ran models with 2, 3, 4, and 5 factors)
- conduct more diagnostics
 - reproduced correlation matrix
 - the difference between the reproduced correlation matrix and the correlation matrix in the data

The *factor.model()* function in *psych* produces the *reproduced correlation matrix* by using the *loadings* in our extracted object. Conceptually, this matrix is the correlations that should be produced if we did not have the raw data but we only had the component loadings. We could do fancy matrix algebra and produce these.

The questions, though, is: How close did we get? How different is the *reproduced correlation matrix* from *GRMSmatrix* – the *R*-matrix produced from our raw data.

```
round(psych::factor.model(pca2$loadings),3) #produces the reproduced correlation matrix
```

	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6	Obj7	Obj8	Obj9	Obj10
Obj1	0.495	0.501	0.435	0.447	0.386	0.403	0.428	0.359	0.294	0.311
Obj2	0.501	0.547	0.477	0.466	0.442	0.395	0.390	0.372	0.303	0.297
Obj3	0.435	0.477	0.460	0.426	0.408	0.380	0.349	0.339	0.318	0.281
Obj4	0.447	0.466	0.426	0.424	0.396	0.367	0.384	0.351	0.307	0.307
Obj5	0.386	0.442	0.408	0.396	0.444	0.282	0.306	0.358	0.315	0.307
Obj6	0.403	0.395	0.380	0.367	0.282	0.383	0.362	0.271	0.257	0.240
Obj7	0.428	0.390	0.349	0.384	0.306	0.362	0.426	0.322	0.278	0.317
Obj8	0.359	0.372	0.339	0.351	0.358	0.271	0.322	0.319	0.278	0.295
Obj9	0.294	0.303	0.318	0.307	0.315	0.257	0.278	0.278	0.281	0.267
Obj10	0.311	0.297	0.281	0.307	0.307	0.240	0.317	0.295	0.267	0.298
Marg1	0.116	-0.042	0.010	0.052	-0.162	0.232	0.258	-0.005	0.051	0.082
Marg2	-0.016	-0.143	-0.045	-0.043	-0.213	0.144	0.118	-0.086	0.012	-0.003
Marg3	0.081	-0.038	0.015	0.031	-0.148	0.195	0.184	-0.028	0.030	0.034
Marg4	0.165	0.062	0.109	0.115	-0.060	0.264	0.243	0.038	0.089	0.084
Marg5	0.208	0.112	0.137	0.149	-0.029	0.286	0.270	0.064	0.096	0.098
Marg6	0.020	-0.123	-0.082	-0.007	-0.114	0.070	0.187	0.022	0.054	0.128
Marg7	-0.164	-0.258	-0.133	-0.145	-0.226	-0.007	-0.028	-0.135	-0.017	-0.046
Strong1	-0.086	-0.081	0.028	0.003	0.110	-0.049	-0.016	0.078	0.161	0.129
Strong2	-0.028	-0.037	0.020	0.040	0.148	-0.055	0.047	0.132	0.163	0.185
Strong3	0.056	0.039	0.059	0.103	0.198	-0.011	0.124	0.189	0.186	0.237
Strong4	-0.131	-0.136	-0.013	-0.037	0.070	-0.080	-0.043	0.048	0.144	0.112
Strong5	0.003	-0.065	-0.026	0.031	0.060	-0.013	0.123	0.109	0.128	0.189
Angry1	0.094	0.175	0.278	0.143	0.152	0.200	0.005	0.054	0.154	-0.003
Angry2	-0.108	-0.048	0.106	-0.022	0.030	0.037	-0.129	-0.047	0.096	-0.056
Angry3	-0.147	-0.098	0.059	-0.069	-0.040	0.018	-0.161	-0.101	0.048	-0.106
	Marg1	Marg2	Marg3	Marg4	Marg5	Marg6	Marg7	Strong1	Strong2	
Obj1	0.116	-0.016	0.081	0.165	0.208	0.020	-0.164	-0.086	-0.028	
Obj2	-0.042	-0.143	-0.038	0.062	0.112	-0.123	-0.258	-0.081	-0.037	
Obj3	0.010	-0.045	0.015	0.109	0.137	-0.082	-0.133	0.028	0.020	
Obj4	0.052	-0.043	0.031	0.115	0.149	-0.007	-0.145	0.003	0.040	
Obj5	-0.162	-0.213	-0.148	-0.060	-0.029	-0.114	-0.226	0.110	0.148	
Obj6	0.232	0.144	0.195	0.264	0.286	0.070	-0.007	-0.049	-0.055	
Obj7	0.258	0.118	0.184	0.243	0.270	0.187	-0.028	-0.016	0.047	
Obj8	-0.005	-0.086	-0.028	0.038	0.064	0.022	-0.135	0.078	0.132	
Obj9	0.051	0.012	0.030	0.089	0.096	0.054	-0.017	0.161	0.163	
Obj10	0.082	-0.003	0.034	0.084	0.098	0.128	-0.046	0.129	0.185	
Marg1	0.800	0.690	0.648	0.619	0.592	0.554	0.469	-0.062	-0.089	
Marg2	0.690	0.664	0.579	0.544	0.498	0.458	0.512	0.027	-0.063	
Marg3	0.648	0.579	0.538	0.517	0.492	0.412	0.396	-0.065	-0.114	
Marg4	0.619	0.544	0.517	0.516	0.499	0.368	0.349	-0.063	-0.114	
Marg5	0.592	0.498	0.492	0.499	0.491	0.338	0.290	-0.108	-0.142	
Marg6	0.554	0.458	0.412	0.368	0.338	0.537	0.365	0.108	0.146	
Marg7	0.469	0.512	0.396	0.349	0.290	0.365	0.481	0.175	0.073	
Strong1	-0.062	0.027	-0.065	-0.063	-0.108	0.108	0.175	0.447	0.394	

Strong2	-0.089	-0.063	-0.114	-0.114	-0.142	0.146	0.073	0.394	0.419
Strong3	-0.080	-0.102	-0.119	-0.113	-0.125	0.164	0.002	0.334	0.404
Strong4	-0.035	0.064	-0.043	-0.052	-0.105	0.142	0.223	0.476	0.415
Strong5	0.157	0.117	0.077	0.057	0.032	0.326	0.167	0.289	0.344
Angry1	-0.040	0.092	0.040	0.103	0.083	-0.232	0.092	0.118	-0.071
Angry2	-0.042	0.142	0.028	0.053	0.004	-0.133	0.229	0.286	0.081
Angry3	0.014	0.200	0.080	0.095	0.042	-0.115	0.270	0.240	0.024
	Strong3	Strong4	Strong5	Angry1	Angry2	Angry3			
Obj1	0.056	-0.131	0.003	0.094	-0.108	-0.147			
Obj2	0.039	-0.136	-0.065	0.175	-0.048	-0.098			
Obj3	0.059	-0.013	-0.026	0.278	0.106	0.059			
Obj4	0.103	-0.037	0.031	0.143	-0.022	-0.069			
Obj5	0.198	0.070	0.060	0.152	0.030	-0.040			
Obj6	-0.011	-0.080	-0.013	0.200	0.037	0.018			
Obj7	0.124	-0.043	0.123	0.005	-0.129	-0.161			
Obj8	0.189	0.048	0.109	0.054	-0.047	-0.101			
Obj9	0.186	0.144	0.128	0.154	0.096	0.048			
Obj10	0.237	0.112	0.189	-0.003	-0.056	-0.106			
Marg1	-0.080	-0.035	0.157	-0.040	-0.042	0.014			
Marg2	-0.102	0.064	0.117	0.092	0.142	0.200			
Marg3	-0.119	-0.043	0.077	0.040	0.028	0.080			
Marg4	-0.113	-0.052	0.057	0.103	0.053	0.095			
Marg5	-0.125	-0.105	0.032	0.083	0.004	0.042			
Marg6	0.164	0.142	0.326	-0.232	-0.133	-0.115			
Marg7	0.002	0.223	0.167	0.092	0.229	0.270			
Strong1	0.334	0.476	0.289	0.118	0.286	0.240			
Strong2	0.404	0.415	0.344	-0.071	0.081	0.024			
Strong3	0.423	0.345	0.355	-0.158	-0.043	-0.106			
Strong4	0.345	0.511	0.314	0.108	0.305	0.263			
Strong5	0.355	0.314	0.377	-0.201	-0.063	-0.096			
Angry1	-0.158	0.108	-0.201	0.624	0.551	0.548			
Angry2	-0.043	0.305	-0.063	0.551	0.615	0.613			
Angry3	-0.106	0.263	-0.096	0.548	0.613	0.626			

We're not really interested in this matrix. We just need it to compare it to the *GRMSmatrix* to produce the residuals. We do that next.

Residuals are the difference between the reproduced (i.e., those created from our component loadings) and *R*-matrix produced by the raw data.

If we look at the $r_{Obj1Obj2}$ in our original correlation matrix (theoretically from the raw data [although we simulated data]), the value is 0.41. The reproduced correlation that we just calculated for this pair is 0.50. The difference is -0.09.

.41 - .50

[1] -0.09

By using the `factor.residuals()` function we can calculate the residuals. Here we will see this difference calculated for us, for all the elements in the matrix.

```
round(psych::factor.residuals(GRMSmatrix, pca2$loadings), 3)
```

	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6	Obj7	Obj8	Obj9	Obj10
Obj1	0.505	-0.093	-0.052	-0.118	-0.047	-0.067	-0.062	-0.010	-0.039	-0.070
Obj2	-0.093	0.453	-0.047	-0.066	-0.045	-0.055	-0.038	-0.079	-0.026	-0.060
Obj3	-0.052	-0.047	0.540	-0.068	-0.067	-0.068	-0.049	-0.059	-0.102	-0.045
Obj4	-0.118	-0.066	-0.068	0.576	-0.074	-0.046	-0.059	-0.097	-0.038	-0.030
Obj5	-0.047	-0.045	-0.067	-0.074	0.556	-0.054	-0.050	-0.097	-0.049	-0.060
Obj6	-0.067	-0.055	-0.068	-0.046	-0.054	0.617	-0.057	-0.033	-0.064	-0.077
Obj7	-0.062	-0.038	-0.049	-0.059	-0.050	-0.057	0.574	-0.067	-0.086	-0.056
Obj8	-0.010	-0.079	-0.059	-0.097	-0.097	-0.033	-0.067	0.681	-0.070	-0.070
Obj9	-0.039	-0.026	-0.102	-0.038	-0.049	-0.064	-0.086	-0.070	0.719	-0.047
Obj10	-0.070	-0.060	-0.045	-0.030	-0.060	-0.077	-0.056	-0.070	-0.047	0.702
Marg1	-0.011	0.017	0.002	0.005	0.018	-0.030	-0.027	0.004	0.015	-0.018
Marg2	0.016	0.016	0.011	0.014	0.030	0.011	-0.012	0.001	-0.002	-0.024
Marg3	-0.026	0.025	0.003	0.011	0.039	-0.034	-0.060	0.001	0.036	-0.010
Marg4	0.002	-0.013	-0.005	-0.044	0.028	-0.054	-0.028	0.014	-0.044	-0.018
Marg5	-0.012	-0.039	-0.029	-0.023	0.032	-0.078	-0.045	-0.019	0.009	0.027
Marg6	-0.018	0.035	0.020	0.032	0.007	-0.007	-0.028	0.004	-0.003	0.005
Marg7	0.020	0.036	0.043	0.014	0.054	0.008	0.016	0.031	-0.023	0.006
Strong1	0.040	0.015	0.002	-0.026	-0.001	-0.005	0.021	-0.043	-0.019	-0.030
Strong2	0.034	-0.023	-0.008	0.001	-0.020	0.042	-0.008	-0.001	-0.044	-0.056
Strong3	-0.015	0.015	0.011	0.038	-0.053	0.032	-0.019	-0.085	-0.056	-0.095
Strong4	0.026	0.031	0.021	-0.003	-0.008	0.038	-0.005	-0.003	-0.038	-0.025
Strong5	0.019	0.062	0.021	-0.020	-0.022	0.014	-0.020	0.002	-0.074	-0.080
Angry1	0.015	-0.026	-0.077	0.006	-0.045	-0.046	0.006	0.022	-0.072	0.028
Angry2	-0.007	0.012	-0.030	-0.002	-0.019	-0.031	0.061	0.040	-0.030	0.011
Angry3	0.023	0.006	-0.027	0.008	-0.010	-0.001	0.040	-0.008	0.001	0.040
	Marg1	Marg2	Marg3	Marg4	Marg5	Marg6	Marg7	Strong1	Strong2	
Obj1	-0.011	0.016	-0.026	0.002	-0.012	-0.018	0.020	0.040	0.034	
Obj2	0.017	0.016	0.025	-0.013	-0.039	0.035	0.036	0.015	-0.023	
Obj3	0.002	0.011	0.003	-0.005	-0.029	0.020	0.043	0.002	-0.008	
Obj4	0.005	0.014	0.011	-0.044	-0.023	0.032	0.014	-0.026	0.001	
Obj5	0.018	0.030	0.039	0.028	0.032	0.007	0.054	-0.001	-0.020	
Obj6	-0.030	0.011	-0.034	-0.054	-0.078	-0.007	0.008	-0.005	0.042	
Obj7	-0.027	-0.012	-0.060	-0.028	-0.045	-0.028	0.016	0.021	-0.008	
Obj8	0.004	0.001	0.001	0.014	-0.019	0.004	0.031	-0.043	-0.001	
Obj9	0.015	-0.002	0.036	-0.044	0.009	-0.003	-0.023	-0.019	-0.044	
Obj10	-0.018	-0.024	-0.010	-0.018	0.027	0.005	0.006	-0.030	-0.056	
Marg1	0.200	-0.026	-0.033	-0.041	-0.044	-0.041	-0.049	0.020	-0.006	
Marg2	-0.026	0.336	-0.077	-0.070	-0.090	-0.026	-0.092	0.020	0.002	
Marg3	-0.033	-0.077	0.462	-0.101	-0.091	-0.066	-0.058	0.004	0.019	
Marg4	-0.041	-0.070	-0.101	0.484	-0.055	-0.087	-0.074	0.038	0.032	
Marg5	-0.044	-0.090	-0.091	-0.055	0.509	-0.093	-0.031	0.010	0.066	

Marg6	-0.041	-0.026	-0.066	-0.087	-0.093	0.463	-0.074	-0.007	-0.056
Marg7	-0.049	-0.092	-0.058	-0.074	-0.031	-0.074	0.519	-0.085	0.006
Strong1	0.020	0.020	0.004	0.038	0.010	-0.007	-0.085	0.553	-0.133
Strong2	-0.006	0.002	0.019	0.032	0.066	-0.056	0.006	-0.133	0.581
Strong3	0.022	0.013	0.033	0.034	0.024	-0.062	-0.016	-0.090	-0.135
Strong4	-0.005	-0.012	0.019	0.015	0.040	-0.037	-0.032	-0.133	-0.100
Strong5	-0.013	-0.020	-0.011	0.001	-0.020	-0.096	-0.032	-0.096	-0.130
Angry1	0.003	-0.029	0.009	-0.023	-0.013	0.076	-0.014	-0.030	0.021
Angry2	0.009	-0.029	-0.005	0.006	-0.010	0.053	-0.044	-0.071	-0.031
Angry3	0.007	0.006	-0.027	-0.026	-0.004	0.029	-0.070	-0.061	0.009
	Strong3	Strong4	Strong5	Angry1	Angry2	Angry3			
Obj1	-0.015	0.026	0.019	0.015	-0.007	0.023			
Obj2	0.015	0.031	0.062	-0.026	0.012	0.006			
Obj3	0.011	0.021	0.021	-0.077	-0.030	-0.027			
Obj4	0.038	-0.003	-0.020	0.006	-0.002	0.008			
Obj5	-0.053	-0.008	-0.022	-0.045	-0.019	-0.010			
Obj6	0.032	0.038	0.014	-0.046	-0.031	-0.001			
Obj7	-0.019	-0.005	-0.020	0.006	0.061	0.040			
Obj8	-0.085	-0.003	0.002	0.022	0.040	-0.008			
Obj9	-0.056	-0.038	-0.074	-0.072	-0.030	0.001			
Obj10	-0.095	-0.025	-0.080	0.028	0.011	0.040			
Marg1	0.022	-0.005	-0.013	0.003	0.009	0.007			
Marg2	0.013	-0.012	-0.020	-0.029	-0.029	0.006			
Marg3	0.033	0.019	-0.011	0.009	-0.005	-0.027			
Marg4	0.034	0.015	0.001	-0.023	0.006	-0.026			
Marg5	0.024	0.040	-0.020	-0.013	-0.010	-0.004			
Marg6	-0.062	-0.037	-0.096	0.076	0.053	0.029			
Marg7	-0.016	-0.032	-0.032	-0.014	-0.044	-0.070			
Strong1	-0.090	-0.133	-0.096	-0.030	-0.071	-0.061			
Strong2	-0.135	-0.100	-0.130	0.021	-0.031	0.009			
Strong3	0.577	-0.096	-0.087	0.056	0.024	0.040			
Strong4	-0.096	0.489	-0.097	-0.047	-0.069	-0.064			
Strong5	-0.087	-0.097	0.623	0.092	0.041	0.044			
Angry1	0.056	-0.047	0.092	0.376	-0.108	-0.115			
Angry2	0.024	-0.069	0.041	-0.108	0.385	-0.147			
Angry3	0.040	-0.064	0.044	-0.115	-0.147	0.374			

There are several strategies to evaluate this matrix:

- see how large the residuals are compared to the original correlations
 - the worst possible model would occur if we extracted no components and would be the size of the original correlations
 - if the correlations were small to start with, we expect small residuals
 - if the correlations were large to start with, the residuals will be relatively larger (this is not terribly problematic)
- comparing residuals requires squaring them first (because residuals can be both positive and negative)

- the sum of the squared residuals divided by the sum of the squared correlations is an estimate of model fit. Subtracting this from 1.0 means that it ranges from 0 to 1. Values $> .95$ are an indication of good fit.

Analyzing the residuals means we need to extract only the upper right of the triangle of the matrix into an object. We can do this in steps.

```
pca2_resids <- psych::factor.residuals(GRMSmatrix, pca2$loadings)#first extract the resids
pca2_resids <- as.matrix(pca2_resids[upper.tri(pca2_resids)])#the object has the residuals in
head(pca2_resids)
```

```
[,1]
[1,] -0.09259539
[2,] -0.05244358
[3,] -0.04745227
[4,] -0.11845843
[5,] -0.06581435
[6,] -0.06842963
```

One criteria of residual analysis is to see how many residuals there are that are greater than an absolute value of 0.05. The result will be a single column with TRUE if it is $> |0.05|$ and false if it is smaller. The sum function will tell us how many TRUE responses are in the matrix. Further, we can write script to obtain the proportion of total number of residuals.

```
large.resid <- abs(pca2_resids) > 0.05
#large.resid
sum(large.resid)
```

```
[1] 85
```

```
round(sum(large.resid)/nrow(pca2_resids),3)
```

```
[1] 0.283
```

We learn that there are 85 residuals greater than the absolute value of 0.05. This represents 28% of the total number of residuals.

There are no hard rules about what proportion of residuals can be greater than 0.05. A common practice is to stay below 50% (?).

Another approach to analyzing residuals is to look at their mean. Because of the +/- valences, we need to square them (to eliminate the negative), take the average, then take the square root.

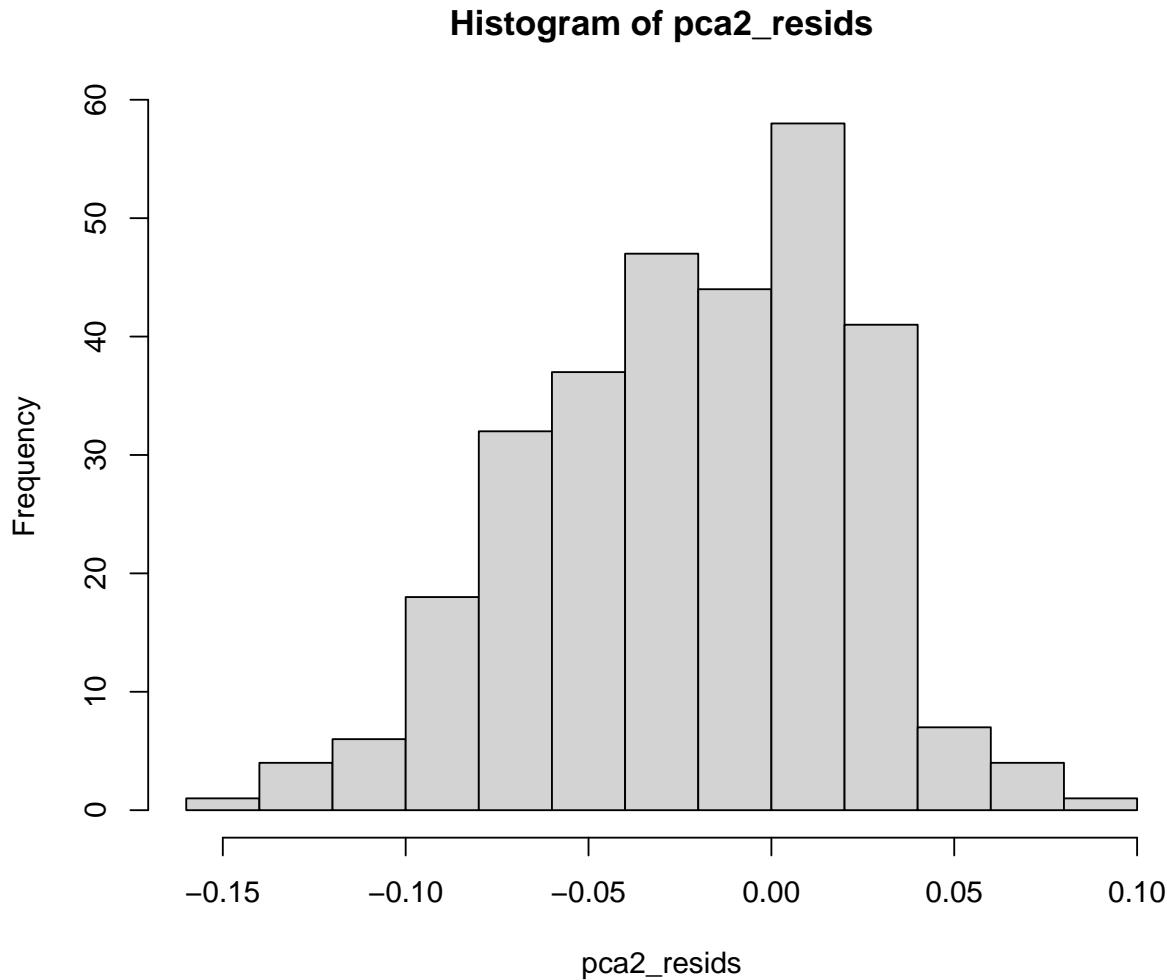
```
round(sqrt(mean(pca2_resids^2)),3)
```

```
[1] 0.048
```

While there are no clear guidelines to interpret these, one recommendation is to consider extracting more components if the value is higher than 0.08 (?).

Finally, we expect our residuals to be normally distributed. A histogram can help us inspect the distribution.

```
hist(pca2_resids)
```



Not bad! Looks reasonably normal. No outliers.

8.5.3.1 Quick recap of how to evaluate the # of components we extracted

- If fewer than 30 variables, the eigenvalue > 1 (Kaiser's) criteria is fine, so long as communalities are all $> .70$.
- If sample size > 250 and the average communalities are $.6$ or greater, this is fine.
- When $N > 200$, the scree plot can be used.
- Regarding residuals

- fewer than 50% should have absolute values > 0.05
- model fit should be > 0.90

8.5.4 Component Rotation

Below is a snip from the workflow to remind us where we are in the steps to PCA.

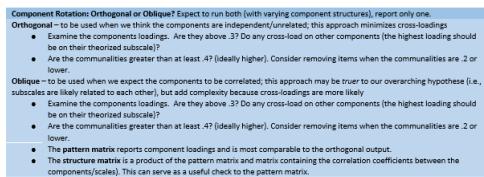


Figure 8.6: Image of an excerpt from the workflow

Rotation improves the interpretation of the components by maximizing the loading on each variable on one of the extracted components while minimizing the loading on all other components. Rotation works by changing the absolute values of the variables while keeping their differential values constant.

There are two big choices and we need to make them on theoretical grounds:

- Orthogonal rotation if you think that the components are independent/unrelated.
 - varimax is the most common orthogonal rotation
- Oblique rotation if you think that the components are related correlated.
 - oblimin and promax are common oblique rotations

Which to do?

- Orthogonal is “easy” because it minimizes cross-loadings, but
- Can you think of a measure where the subscales would *not* be correlated?

8.5.4.1 Orthogonal rotation

```
#pcaORTH <- psych::principal(GRMSmatrix, nfactors = 4, rotate = "varimax")
pcaORTH <- psych::principal(dfGRMS, nfactors = 4, rotate = "varimax")
pcaORTH
```

Principal Components Analysis

Call: psych::principal(r = dfGRMS, nfactors = 4, rotate = "varimax")

Standardized loadings (pattern matrix) based upon correlation matrix

	RC1	RC2	RC3	RC4	h2	u2	com
Obj1	0.69	0.08	-0.10	-0.09	0.50	0.50	1.1
Obj2	0.72	-0.10	-0.14	0.00	0.55	0.45	1.1

Obj3	0.65	-0.01	-0.04	0.17	0.46	0.54	1.1
Obj4	0.65	0.02	0.00	0.00	0.42	0.58	1.0
Obj5	0.61	-0.21	0.14	0.05	0.44	0.56	1.4
Obj6	0.56	0.23	-0.11	0.08	0.38	0.62	1.5
Obj7	0.58	0.24	0.05	-0.16	0.43	0.57	1.5
Obj8	0.54	-0.04	0.15	-0.06	0.32	0.68	1.2
Obj9	0.47	0.04	0.22	0.11	0.28	0.72	1.5
Obj10	0.47	0.06	0.25	-0.10	0.30	0.70	1.7
Marg1	0.06	0.89	-0.02	-0.08	0.80	0.20	1.0
Marg2	-0.09	0.80	0.02	0.14	0.66	0.34	1.1
Marg3	0.03	0.73	-0.08	0.02	0.54	0.46	1.0
Marg4	0.16	0.69	-0.10	0.07	0.52	0.48	1.2
Marg5	0.21	0.65	-0.15	0.01	0.49	0.51	1.3
Marg6	-0.03	0.61	0.32	-0.25	0.54	0.46	1.9
Marg7	-0.24	0.57	0.22	0.23	0.48	0.52	2.0
Strong1	0.00	-0.03	0.61	0.27	0.45	0.55	1.4
Strong2	0.06	-0.09	0.64	0.00	0.42	0.58	1.1
Strong3	0.16	-0.10	0.60	-0.15	0.42	0.58	1.3
Strong4	-0.06	0.01	0.66	0.28	0.51	0.49	1.4
Strong5	0.04	0.17	0.56	-0.18	0.38	0.62	1.4
Angry1	0.22	0.01	-0.13	0.75	0.62	0.38	1.2
Angry2	-0.03	0.03	0.13	0.77	0.61	0.39	1.1
Angry3	-0.11	0.10	0.06	0.77	0.63	0.37	1.1

	RC1	RC2	RC3	RC4
SS loadings	3.84	3.79	2.32	2.20
Proportion Var	0.15	0.15	0.09	0.09
Cumulative Var	0.15	0.31	0.40	0.49
Proportion Explained	0.32	0.31	0.19	0.18
Cumulative Proportion	0.32	0.63	0.82	1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is 0.05
with the empirical chi square 360.58 with prob < 0.00000000015

Fit based upon off diagonal values = 0.94

Essentially, we have the same information as before, except that loadings are calculated after rotation (which adjusts the absolute values of the component loadings while keeping their differential values constant). Our communality and uniqueness values remain the same. The eigenvalues (SS loadings) should even out, but the proportion of variance explained and cumulative variance will remain the same.

The *print.psych()* function facilitates interpretation and prioritizes the information about which we care most:

- “cut” will display loadings above .3

- if some items load on no factors
- if some items have cross-loadings (and their relative weights)
- “sort” will reorder the loadings to make it clearer (to the best of its ability...in the case of ties) to which component/scale it belongs

```
pca_table <- psych::print.psych(pcaORTH, cut = 0.3, sort=TRUE)
```

```
Principal Components Analysis
Call: psych::principal(r = dfGRMS, nfactors = 4, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
    item   RC1   RC2   RC3   RC4   h2   u2 com
Obj2      2  0.72           0.55 0.45 1.1
Obj1      1  0.69           0.50 0.50 1.1
Obj3      3  0.65           0.46 0.54 1.1
Obj4      4  0.65           0.42 0.58 1.0
Obj5      5  0.61           0.44 0.56 1.4
Obj7      7  0.58           0.43 0.57 1.5
Obj6      6  0.56           0.38 0.62 1.5
Obj8      8  0.54           0.32 0.68 1.2
Obj9      9  0.47           0.28 0.72 1.5
Obj10     10 0.47           0.30 0.70 1.7
Marg1     11  0.89           0.80 0.20 1.0
Marg2     12  0.80           0.66 0.34 1.1
Marg3     13  0.73           0.54 0.46 1.0
Marg4     14  0.69           0.52 0.48 1.2
Marg5     15  0.65           0.49 0.51 1.3
Marg6     16  0.61  0.32       0.54 0.46 1.9
Marg7     17  0.57           0.48 0.52 2.0
Strong4   21      0.66       0.51 0.49 1.4
Strong2   19      0.64       0.42 0.58 1.1
Strong1   18      0.61       0.45 0.55 1.4
Strong3   20      0.60       0.42 0.58 1.3
Strong5   22      0.56       0.38 0.62 1.4
Angry3    25            0.77 0.63 0.37 1.1
Angry2    24            0.77 0.61 0.39 1.1
Angry1    23            0.75 0.62 0.38 1.2

                    RC1   RC2   RC3   RC4
SS loadings      3.84 3.79 2.32 2.20
Proportion Var  0.15 0.15 0.09 0.09
Cumulative Var  0.15 0.31 0.40 0.49
Proportion Explained 0.32 0.31 0.19 0.18
Cumulative Proportion 0.32 0.63 0.82 1.00

Mean item complexity = 1.3
Test of the hypothesis that 4 components are sufficient.
```

The root mean square of the residuals (RMSR) is 0.05
with the empirical chi square 360.58 with prob < 0.00000000015

Fit based upon off diagonal values = 0.94

In the unrotated solution, most variables loaded on the first component. After rotation, there are four clear components/scales. Further, there is clear (or at least reasonable) component/scale membership for each item. The item Marg6 was the only one that included cross-loadings. However, the difference was so great (.61 on its primary factor; .32 on the secondary factor) that the item's membership is clearly on the second component.

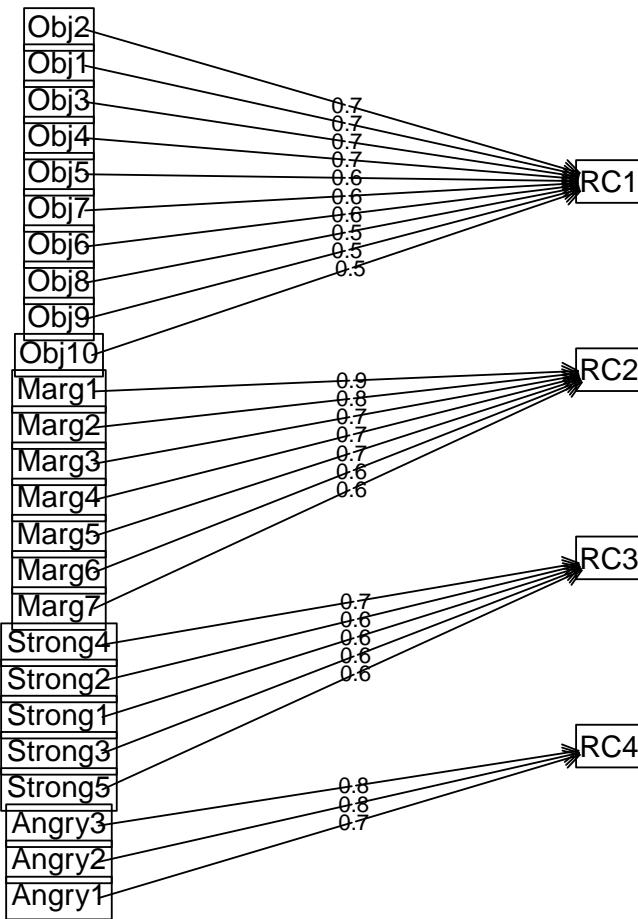
If this were a new scale and we had not yet established ideas for subscales, the next step would be to examine the items, themselves, and try to name the scales/components. If our scale construction included a priori/planned subscales, this is where we hope that the items fall where they were hypothesized to do so. Our simulated data worked perfectly and replicated the four scales that Lewis and Neville (?) reported in the article.

- Assumptions of Beauty and Sexual Objectification
- Silenced and Marginalized
- Strong Woman Stereotype
- Angry Woman Stereotype

We can also create a figure of the result.

```
psych::fa.diagram(pcaORTH)
```

Components Analysis



We can extract the component loadings and write them to a table. This can be useful in preparing an APA style table for a manuscript or presentation.

```
names(pcaORTH)
```

```
[1] "values"      "rotation"     "n.obs"       "communality"   "loadings"
[6] "fit"         "fit.off"      "fn"          "Call"         "uniquenesses"
[11] "complexity"  "chi"        "EPVAL"       "R2"           "objective"
[16] "residual"    "rms"        "factors"     "dof"          "null.dof"
[21] "null.model"  "criteria"    "STATISTIC"   "PVAL"         "weights"
[26] "r.scores"    "rot.mat"     "Vaccounted" "Structure"    "scores"
```

```
pcaORTH_table <- round(pcaORTH$loadings, 3)
```

```
write.table(pcaORTH_table, file="pcaORTH_table.csv", sep=",", col.names=TRUE, row.names=FALSE)
pcaORTH_table
```

Loadings:

	RC1	RC2	RC3	RC4
Obj1	0.686		-0.100	
Obj2	0.720		-0.142	
Obj3	0.654			0.174
Obj4	0.651			
Obj5	0.614	-0.212	0.142	
Obj6	0.558	0.231	-0.109	
Obj7	0.583	0.241		-0.159
Obj8	0.541		0.147	
Obj9	0.470		0.216	0.107
Obj10	0.470		0.253	
Marg1		0.888		
Marg2		0.797		0.145
Marg3		0.728		
Marg4	0.158	0.691		
Marg5	0.211	0.651	-0.153	
Marg6		0.609	0.317	-0.253
Marg7	-0.240	0.570	0.217	0.226
Strong1			0.613	0.266
Strong2			0.638	
Strong3	0.160		0.604	-0.150
Strong4			0.656	0.278
Strong5		0.171	0.559	-0.184
Angry1	0.219		-0.133	0.747
Angry2			0.134	0.771
Angry3	-0.110			0.775

	RC1	RC2	RC3	RC4
SS loadings	3.839	3.788	2.316	2.204
Proportion Var	0.154	0.152	0.093	0.088
Cumulative Var	0.154	0.305	0.398	0.486

8.5.4.2 Oblique rotation

Whereas the orthogonal rotation sought to maximize the independence/unrelatedness of the components, an oblique rotation will allow them to be correlated. Researchers often explore both solutions but then only report one.

```
#pcaOBL <- psych::principal(GRMSmatrix, nfactors = 4, rotate = "oblimin")
pcaOBL <- psych::principal(dfGRMS, nfactors = 4, rotate = "oblimin")
pcaOBL
```

```
Principal Components Analysis
Call: psych::principal(r = dfGRMS, nfactors = 4, rotate = "oblimin")
Standardized loadings (pattern matrix) based upon correlation matrix
      TC1   TC2   TC3   TC4     h2    u2 com
Obj1  0.68  0.08 -0.07 -0.08  0.50  0.50  1.1
```

Obj2	0.72	-0.09	-0.11	0.02	0.55	0.45	1.1
Obj3	0.66	-0.02	-0.01	0.19	0.46	0.54	1.2
Obj4	0.65	0.02	0.03	0.01	0.42	0.58	1.0
Obj5	0.62	-0.21	0.17	0.06	0.44	0.56	1.4
Obj6	0.55	0.23	-0.09	0.10	0.38	0.62	1.5
Obj7	0.58	0.25	0.07	-0.15	0.43	0.57	1.5
Obj8	0.54	-0.04	0.17	-0.06	0.32	0.68	1.2
Obj9	0.47	0.04	0.23	0.11	0.28	0.72	1.6
Obj10	0.46	0.06	0.27	-0.10	0.30	0.70	1.8
Marg1	0.03	0.89	-0.03	-0.08	0.80	0.20	1.0
Marg2	-0.11	0.79	0.01	0.14	0.66	0.34	1.1
Marg3	0.01	0.73	-0.08	0.03	0.54	0.46	1.0
Marg4	0.14	0.69	-0.10	0.07	0.52	0.48	1.1
Marg5	0.20	0.65	-0.15	0.02	0.49	0.51	1.3
Marg6	-0.05	0.61	0.31	-0.26	0.54	0.46	1.9
Marg7	-0.26	0.56	0.20	0.21	0.48	0.52	2.0
Strong1	0.00	-0.04	0.61	0.24	0.45	0.55	1.3
Strong2	0.06	-0.10	0.64	-0.02	0.42	0.58	1.1
Strong3	0.15	-0.10	0.61	-0.17	0.42	0.58	1.3
Strong4	-0.06	-0.01	0.65	0.25	0.51	0.49	1.3
Strong5	0.03	0.17	0.56	-0.20	0.38	0.62	1.5
Angry1	0.23	-0.01	-0.12	0.76	0.62	0.38	1.2
Angry2	-0.03	0.01	0.13	0.77	0.61	0.39	1.1
Angry3	-0.11	0.08	0.06	0.77	0.63	0.37	1.1

	TC1	TC2	TC3	TC4
SS loadings	3.82	3.79	2.33	2.20
Proportion Var	0.15	0.15	0.09	0.09
Cumulative Var	0.15	0.30	0.40	0.49
Proportion Explained	0.31	0.31	0.19	0.18
Cumulative Proportion	0.31	0.63	0.82	1.00

With component correlations of

	TC1	TC2	TC3	TC4
TC1	1.00	0.02	-0.03	-0.02
TC2	0.02	1.00	0.02	0.03
TC3	-0.03	0.02	1.00	0.03
TC4	-0.02	0.03	0.03	1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is 0.05
with the empirical chi square 360.58 with prob < 0.000000000015

Fit based upon off diagonal values = 0.94

We can make it a little easier to interpret by removing all factor loadings below .30.

```
psych::print.psych(pcaOBJ, cut = 0.3, sort=TRUE)
```

Principal Components Analysis

Call: psych::principal(r = dfGRMS, nfactors = 4, rotate = "oblimin")

Standardized loadings (pattern matrix) based upon correlation matrix

item	TC1	TC2	TC3	TC4	h2	u2	com	
Obj2	2	0.72			0.55	0.45	1.1	
Obj1	1	0.68			0.50	0.50	1.1	
Obj3	3	0.66			0.46	0.54	1.2	
Obj4	4	0.65			0.42	0.58	1.0	
Obj5	5	0.62			0.44	0.56	1.4	
Obj7	7	0.58			0.43	0.57	1.5	
Obj6	6	0.55			0.38	0.62	1.5	
Obj8	8	0.54			0.32	0.68	1.2	
Obj9	9	0.47			0.28	0.72	1.6	
Obj10	10	0.46			0.30	0.70	1.8	
Marg1	11		0.89		0.80	0.20	1.0	
Marg2	12		0.79		0.66	0.34	1.1	
Marg3	13		0.73		0.54	0.46	1.0	
Marg4	14		0.69		0.52	0.48	1.1	
Marg5	15		0.65		0.49	0.51	1.3	
Marg6	16		0.61	0.31	0.54	0.46	1.9	
Marg7	17		0.56		0.48	0.52	2.0	
Strong4	21			0.65	0.51	0.49	1.3	
Strong2	19			0.64	0.42	0.58	1.1	
Strong1	18			0.61	0.45	0.55	1.3	
Strong3	20			0.61	0.42	0.58	1.3	
Strong5	22			0.56	0.38	0.62	1.5	
Angry3	25				0.77	0.63	0.37	1.1
Angry2	24				0.77	0.61	0.39	1.1
Angry1	23				0.76	0.62	0.38	1.2

TC1 TC2 TC3 TC4

SS loadings 3.82 3.79 2.33 2.20

Proportion Var 0.15 0.15 0.09 0.09

Cumulative Var 0.15 0.30 0.40 0.49

Proportion Explained 0.31 0.31 0.19 0.18

Cumulative Proportion 0.31 0.63 0.82 1.00

With component correlations of

TC1 TC2 TC3 TC4

TC1 1.00 0.02 -0.03 -0.02

TC2 0.02 1.00 0.02 0.03

TC3 -0.03 0.02 1.00 0.03

TC4 -0.02 0.03 0.03 1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is 0.05
 with the empirical chi square 360.58 with prob < 0.00000000015

Fit based upon off diagonal values = 0.94

All of the items stayed in their respective components. Note, though, that because our specification included “sort=TRUE” that the relative weights wiggled around and so the items are listed in a different order than in the orthogonal rotation.

Let's create a table and write it to our file.

```
pcaOBL_table <- round(pcaOBL$loadings, 3)
write.table(pcaOBL_table, file="pcaOBL_table.csv", sep=",", col.names=TRUE, row.names=FALSE)
pcaOBL_table
```

Loadings:

	TC1	TC2	TC3	TC4
Obj1	0.685			
Obj2	0.724		-0.112	
Obj3	0.657			0.185
Obj4	0.650			
Obj5	0.618	-0.213	0.169	
Obj6	0.554	0.232		
Obj7	0.576	0.246		-0.151
Obj8	0.540		0.169	
Obj9	0.467		0.235	0.107
Obj10	0.465		0.272	
Marg1		0.891		
Marg2	-0.108	0.793		0.143
Marg3		0.729		
Marg4	0.142	0.691		
Marg5	0.197	0.653	-0.149	
Marg6		0.612	0.311	-0.265
Marg7	-0.255	0.561	0.204	0.215
Strong1		0.614	0.245	
Strong2		0.641		
Strong3	0.154	-0.102	0.611	-0.169
Strong4			0.654	0.254
Strong5		0.169	0.559	-0.203
Angry1	0.226		-0.123	0.756
Angry2			0.134	0.766
Angry3	-0.108			0.771

	TC1	TC2	TC3	TC4
SS loadings	3.829	3.786	2.334	2.198
Proportion Var	0.153	0.151	0.093	0.088

```
Cumulative Var 0.153 0.305 0.398 0.486
```

The same four components/scales have emerged, but they are in different order.

The oblique rotation allows us to see the correlation between the components/scales. This was not available in the orthogonal rotation because the assumption of the orthogonal/varimax rotation is that the scales/components are uncorrelated; hence in the analysis they were fixed to 0.0.

We can see that all the scales have almost no relation with each other. That is the the correlations range between -0.03 to 0.03. This is unusual and likely a biproduct of simulating data. It does, though, support the orthogonal rotation as the preferred one.

Of course, there is always a little complexity. In oblique rotations, there is a distinction between the *pattern* matrix (which reports component loadings and is comparable to the matrix we interpreted for the orthogonal rotation) and the *structure* matrix (takes into account the relationship between the components/scales – it is a product of the pattern matrix and the matrix containing the correlation coefficients between the components/scales). Most interpret the pattern matrix because it is simpler; however, it could be that values in the pattern matrix are suppressed because of relations between the components. Therefore, the structure matrix can be a useful check and some editors will request it.

Obtaining the structure matrix requires two steps. First, multiply the factor loadings with the phi matrix.

```
#names(pcaOBL)
pcaOBL$loadings %*% pcaOBL$Phi
```

	TC1	TC2	TC3	TC4
Obj1	0.69050128	0.092733819	-0.09342837	-0.096688897
Obj2	0.72526181	-0.078285639	-0.13389319	-0.007970087
Obj3	0.65222185	0.002163388	-0.02080291	0.169195306
Obj4	0.64980993	0.034592152	0.01021951	-0.003172452
Obj5	0.60724873	-0.195994047	0.14916017	0.040795794
Obj6	0.55920407	0.244637871	-0.09553755	0.084850366
Obj7	0.58233119	0.255604967	0.05567917	-0.156139162
Obj8	0.53550142	-0.026606024	0.15167765	-0.064536944
Obj9	0.45868906	0.055743636	0.22592591	0.104498059
Obj10	0.46065458	0.072992922	0.25626058	-0.099641255
Marg1	0.05429308	0.889387150	-0.01272292	-0.060136145
Marg2	-0.09540568	0.794155634	0.03439142	0.165860367
Marg3	0.02785178	0.728248568	-0.06944307	0.041476497
Marg4	0.15779653	0.693656404	-0.08609067	0.083060621
Marg5	0.21422529	0.654667296	-0.14130873	0.030721117
Marg6	-0.03873213	0.610566300	0.31476620	-0.237541460
Marg7	-0.25458376	0.564865999	0.22877602	0.242123251
Strong1	-0.02493831	-0.026459940	0.62139121	0.264583641
Strong2	0.03727021	-0.083266239	0.63714719	-0.002354881
Strong3	0.13898024	-0.091205429	0.59914813	-0.154079964
Strong4	-0.08957417	0.008131805	0.66462011	0.277765084

```
Strong5  0.02206318  0.175205559  0.55465506 -0.180203186
Angry1   0.21157554  0.012544720 -0.10378683  0.745823358
Angry2   -0.05377743  0.030179345  0.16057125  0.771738772
Angry3   -0.12591995  0.094122499  0.08785893  0.777830061
```

Then use Field's (?) function to produce the matrix.

```
#Field's function to produce the structure matrix
factor.structure <- function(fa, cut = 0.2, decimals = 2){
  structure.matrix <- psych::fa.sort(fa$loadings %*% fa$Phi)
  structure.matrix <- data.frame(ifelse(abs(structure.matrix) < cut, "", round(structure.matrix, decimals)))
  return(structure.matrix)
}

factor.structure(pca0BL, cut = 0.3)
```

	TC1	TC2	TC3	TC4
Obj2	0.73			
Obj1	0.69			
Obj3	0.65			
Obj4	0.65			
Obj5	0.61			
Obj7	0.58			
Obj6	0.56			
Obj8	0.54			
Obj10	0.46			
Obj9	0.46			
Marg1		0.89		
Marg2		0.79		
Marg3		0.73		
Marg4		0.69		
Marg5		0.65		
Marg6		0.61	0.31	
Marg7		0.56		
Strong4			0.66	
Strong2			0.64	
Strong1			0.62	
Strong3			0.6	
Strong5			0.55	
Angry3				0.78
Angry2				0.77
Angry1				0.75

Although some of the relative values changed, our items were stable regarding their component membership.

8.5.5 Component Scores

Component *scores* (PC scores) can be created for each case (row) on each component (column). These can be used to assess the relative standing of one person on the construct/variable to another. We can also use them in regression (in place of means or sums) when groups of predictors correlate so highly that there is multicollinearity.

Computation involves multiplying an individual's item-level responses by the component loadings we obtained through the PCA process. The results will be one score per component for each row/case.

```
pcaOBL <- psych::principal(dfGRMS, nfactors=4, rotate="oblimin", scores=TRUE)
head(pcaOBL$scores, 10) #shows us only the first 10 (of N = 2571)
```

	TC1	TC2	TC3	TC4
[1,]	-0.434736430	1.2702286	0.83668623	-0.01950134
[2,]	-0.358346641	0.6998443	-1.28684407	-0.10546432
[3,]	-1.245491638	-0.1908764	-1.42929276	-2.51420936
[4,]	-0.004981576	-0.1127983	-1.38956225	1.86355065
[5,]	0.296789107	0.5137755	0.08833053	-2.09774692
[6,]	0.684949888	1.6400602	0.44203340	-0.27258969
[7,]	-0.771083947	-1.1922357	1.09617794	0.36910174
[8,]	-1.284265060	-0.4144262	0.08708796	-1.07524322
[9,]	0.666522142	0.2999858	1.81030433	1.46348684
[10,]	0.292763466	-0.3285169	0.30101493	0.23847094

```
dfGRMS <- cbind(dfGRMS, pcaOBL$scores) #adds them to our raw dataset
```

To bring this full circle, we can see the correlation of the component scores; the pattern maps onto what we saw previously.

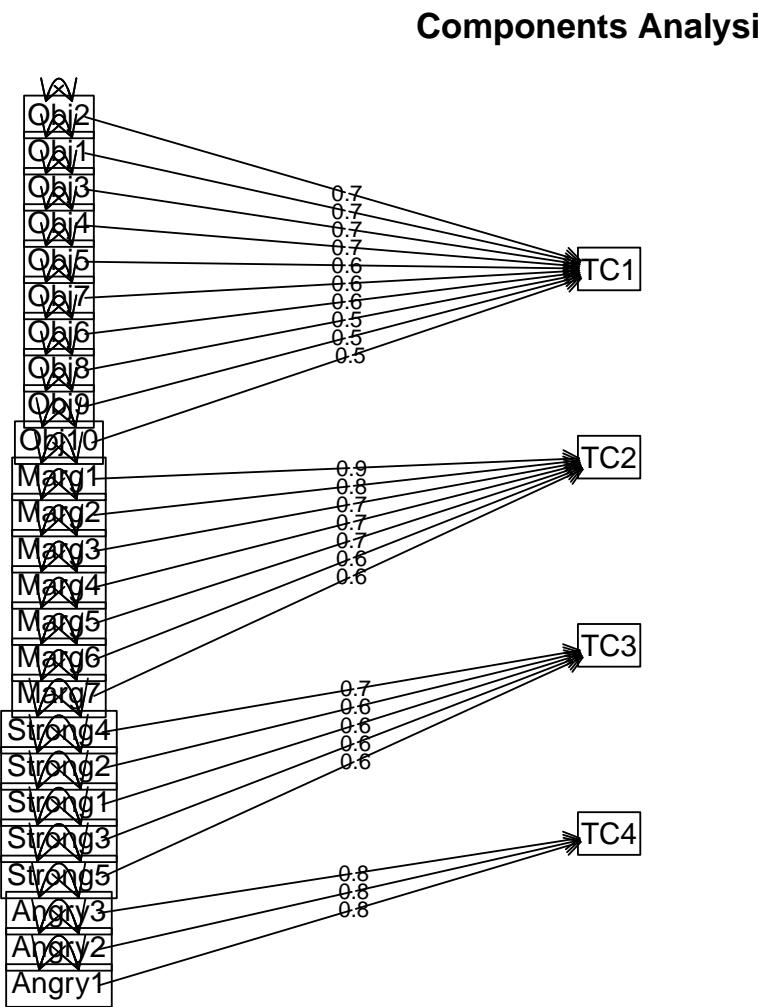
```
psych::corr.test(dfGRMS [c("TC1", "TC4", "TC3", "TC2")])
```

```
Call:psych::corr.test(x = dfGRMS[c("TC1", "TC4", "TC3", "TC2")])
Correlation matrix
  TC1  TC4  TC3  TC2
TC1  1.00 -0.02 -0.03  0.02
TC4 -0.02  1.00  0.03  0.03
TC3 -0.03  0.03  1.00  0.02
TC2  0.02  0.03  0.02  1.00
Sample Size
[1] 259
Probability values (Entries above the diagonal are adjusted for multiple tests.)
  TC1  TC4  TC3  TC2
TC1  0.00 1.00 1.00    1
TC4  0.70 0.00 1.00    1
TC3  0.65 0.59 0.00    1
```

```
TC2 0.74 0.69 0.76 0
```

To see confidence intervals of the correlations, print with the short=FALSE option

```
psych::fa.diagram (pcaOBJ, error=TRUE, side=3)
```



8.6 APA Style Results

Results

The dimensionality of the 25 items from the Gendered Racial Microaggressions Scale for Black Women was analyzed using principal components analysis. First, data were screened to determine the suitability of the data for this analyses. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO; Kaiser, 1970) represents the ratio of the squared correlation between variables to the squared partial correlation between variables. KMO ranges from 0.00 to 1.00; values closer to 1.00 indicate that the patterns of correlations are relatively compact and that component analysis

should yield distinct and reliable components (Field, 2012). In our dataset, the KMO value was .86, indicating acceptable sampling adequacy. The Bartlett's Test of Sphericity examines whether the population correlation matrix resembles an identity matrix (Field, 2012). When the p value for the Bartlett's test is $< .05$, we are fairly certain we have clusters of correlated variables. In our dataset, $\chi^2(300) = 1683.76, p < .001$, indicating the correlations between items are sufficiently large enough for principal components analysis. The determinant of the correlation matrix alerts us to any issues of multicollinearity or singularity and should be larger than 0.00001. Our determinant was 0.00115 and, again, indicated that our data was suitable for the analysis.

Four criteria were used to determine the number of components to extract: a priori theory, the scree test, the eigenvalue-greater-than-one criteria, and the interpretability of the solution. Kaiser's eigenvalue-greater-than-one criteria suggested four components, and, in combination explained 49% of the variance. The inflexion in the scree plot justified retaining four components. Based on the convergence of these decisions, four components were extracted. We investigated each with orthogonal (varimax) and oblique (oblimin) procedures. Given the non-significant correlations (ranging from -0.03 to 0.03) and the clear component loadings in the orthogonal rotation, we determined that an orthogonal solution was most appropriate.

The rotated solution, as shown in Table 1 and Figure 1, yielded four interpretable components, each listed with the proportion of variance accounted for: assumptions of beauty and sexual objectification (15%), silenced and marginalized (15%), strong woman stereotype (9%), and angry woman stereotype (15%).

Regarding the Table 1, I would include a table with ALL the values, bolding those with component membership. This is easy, though, because it is how the table was exported when we wrote it to a .csv file.

8.7 Back to the FutuRe: The relationship between PCA and item analysis

I included the lesson on item analysis because I find it to be a useful stepping stone into principal components and principal factor analyses. How do the results we obtained from PCA compare to those found in item analysis?

First, we score the total and subscales using the dataset we simulated above (dfGRMS).

```
library(tidyverse)
GRMSVars <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9", "Obj10", "Obj11")
ObjectifiedVars <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9", "Obj10", "Obj11")
MarginalizedVars <- c("Marg1", "Marg2", "Marg3", "Marg4", "Marg5", "Marg6", "Marg7")
StrongVars <- c("Strong1", "Strong2", "Strong3", "Strong4", "Strong5")
AngryVars <- c("Angry1", "Angry2", "Angry3")

dfGRMS$GRMStot <- sjstats::mean_n(dfGRMS[,GRMSVars], .80)#will create the mean for each individual
dfGRMS$Objectified <- sjstats::mean_n(dfGRMS[,ObjectifiedVars], .80)#will create the mean for each individual
dfGRMS$Marginalized <- sjstats::mean_n(dfGRMS[,MarginalizedVars], .80)#will create the mean for each individual
dfGRMS$Strong <- sjstats::mean_n(dfGRMS[,StrongVars], .80)#will create the mean for each individual
dfGRMS$Angry <- sjstats::mean_n(dfGRMS[,AngryVars], .80)#will create the mean for each individual
```

While we are at it, let's just create tiny dfs with just our variables of interest.

```
GRMStotal <- dplyr::select(dfGRMS, Obj1:Angry3)
Objectification <- dplyr::select(dfGRMS, Obj1:Obj10)
Marginalization <- dplyr::select(dfGRMS, Marg1:Marg7)
Strong <- dplyr::select(dfGRMS, Strong1:Strong5)
Angry <- dplyr::select(dfGRMS, Angry1:Angry3)
```

8.7.1 Calculating and Extracting Item-Total Correlation Coefficients

8.7.1.1 Corrected item-total correlations from the *psych::alpha()*

Let's first ask, "Is there support for this instrument as a unidimensional measure?" To do that, we get an alpha for the whole scale score.

```
GRMSalpha <- psych::alpha(GRMStotal) #creating an object from this analysis so I can extract alpha
```

Reliability analysis

Call: psych::alpha(x = GRMStotal)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.75	0.75	0.81	0.11	3	0.022	2.1	0.42	0.071

lower	alpha	upper	95% confidence boundaries		
0.71	0.75	0.8			

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
Obj1	0.74	0.74	0.80	0.11	2.8	0.023	0.026	0.070	
Obj2	0.74	0.74	0.80	0.11	2.9	0.023	0.025	0.073	
Obj3	0.74	0.74	0.80	0.11	2.8	0.023	0.027	0.069	
Obj4	0.74	0.74	0.80	0.11	2.8	0.023	0.027	0.070	
Obj5	0.75	0.74	0.81	0.11	2.9	0.023	0.026	0.070	
Obj6	0.74	0.74	0.80	0.10	2.8	0.024	0.027	0.067	
Obj7	0.74	0.74	0.80	0.10	2.8	0.024	0.027	0.066	
Obj8	0.74	0.74	0.81	0.11	2.9	0.023	0.027	0.070	
Obj9	0.74	0.74	0.80	0.11	2.8	0.023	0.028	0.067	
Obj10	0.74	0.74	0.81	0.11	2.9	0.023	0.028	0.066	
Marg1	0.73	0.74	0.79	0.10	2.8	0.024	0.023	0.074	
Marg2	0.74	0.74	0.80	0.11	2.8	0.024	0.024	0.072	
Marg3	0.74	0.74	0.80	0.11	2.9	0.023	0.025	0.078	
Marg4	0.74	0.74	0.80	0.10	2.8	0.024	0.026	0.071	
Marg5	0.74	0.74	0.80	0.11	2.8	0.024	0.026	0.067	
Marg6	0.74	0.74	0.80	0.11	2.9	0.023	0.026	0.070	
Marg7	0.75	0.75	0.81	0.11	3.0	0.023	0.026	0.070	

Strong1	0.75	0.75	0.81	0.11	3.0	0.022	0.028	0.071
Strong2	0.75	0.75	0.81	0.11	3.0	0.022	0.027	0.074
Strong3	0.75	0.75	0.81	0.11	3.0	0.022	0.027	0.072
Strong4	0.75	0.75	0.81	0.11	3.0	0.022	0.027	0.074
Strong5	0.75	0.75	0.81	0.11	3.0	0.023	0.028	0.072
Angry1	0.75	0.75	0.81	0.11	3.0	0.023	0.027	0.069
Angry2	0.75	0.75	0.81	0.11	3.0	0.022	0.027	0.079
Angry3	0.75	0.75	0.81	0.11	3.1	0.022	0.026	0.079

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
Obj1	259	0.44	0.44	0.41	0.34	1.8	1.09
Obj2	259	0.38	0.38	0.35	0.28	1.9	1.14
Obj3	259	0.45	0.46	0.43	0.36	2.0	1.01
Obj4	259	0.42	0.44	0.41	0.35	1.9	0.89
Obj5	259	0.35	0.37	0.32	0.25	2.1	1.19
Obj6	259	0.49	0.48	0.44	0.39	1.8	1.29
Obj7	259	0.49	0.49	0.47	0.40	2.0	1.09
Obj8	259	0.37	0.39	0.34	0.28	2.2	1.00
Obj9	259	0.42	0.44	0.39	0.34	1.8	0.99
Obj10	259	0.40	0.41	0.37	0.31	1.9	1.05
Marg1	259	0.53	0.50	0.52	0.46	2.0	1.02
Marg2	259	0.46	0.43	0.43	0.38	3.5	0.99
Marg3	259	0.45	0.41	0.39	0.34	2.4	1.30
Marg4	259	0.50	0.47	0.45	0.41	3.3	1.17
Marg5	259	0.49	0.45	0.43	0.38	2.4	1.31
Marg6	259	0.40	0.38	0.35	0.28	2.9	1.37
Marg7	259	0.34	0.32	0.27	0.23	2.7	1.21
Strong1	259	0.26	0.29	0.23	0.18	1.3	0.88
Strong2	259	0.22	0.25	0.19	0.13	2.3	0.95
Strong3	259	0.21	0.26	0.20	0.13	1.5	0.83
Strong4	259	0.27	0.29	0.24	0.17	1.6	1.10
Strong5	259	0.27	0.31	0.24	0.19	1.4	0.83
Angry1	259	0.33	0.31	0.27	0.22	2.0	1.15
Angry2	259	0.27	0.26	0.22	0.16	2.5	1.20
Angry3	259	0.24	0.23	0.18	0.13	2.4	1.16

Non missing response frequency for each item

	0	1	2	3	4	5	miss
Obj1	0.12	0.30	0.30	0.22	0.05	0.00	0
Obj2	0.12	0.25	0.31	0.24	0.07	0.01	0
Obj3	0.06	0.28	0.36	0.24	0.07	0.00	0
Obj4	0.04	0.28	0.45	0.19	0.04	0.00	0
Obj5	0.12	0.19	0.35	0.22	0.12	0.01	0
Obj6	0.17	0.27	0.24	0.20	0.09	0.02	0
Obj7	0.09	0.24	0.37	0.22	0.06	0.01	0
Obj8	0.04	0.20	0.35	0.33	0.08	0.00	0
Obj9	0.10	0.26	0.41	0.19	0.04	0.00	0

Obj10	0.09	0.27	0.39	0.17	0.08	0.00	0
Marg1	0.06	0.24	0.41	0.22	0.08	0.00	0
Marg2	0.00	0.03	0.12	0.34	0.36	0.15	0
Marg3	0.07	0.17	0.27	0.25	0.18	0.05	0
Marg4	0.00	0.05	0.20	0.29	0.25	0.20	0
Marg5	0.08	0.17	0.31	0.24	0.14	0.07	0
Marg6	0.06	0.10	0.24	0.26	0.22	0.13	0
Marg7	0.03	0.12	0.27	0.35	0.14	0.08	0
Strong1	0.20	0.42	0.31	0.08	0.00	0.00	0
Strong2	0.03	0.18	0.35	0.37	0.07	0.00	0
Strong3	0.12	0.41	0.38	0.09	0.00	0.00	0
Strong4	0.18	0.29	0.32	0.16	0.04	0.00	0
Strong5	0.14	0.38	0.41	0.07	0.00	0.00	0
Angry1	0.09	0.25	0.31	0.26	0.08	0.02	0
Angry2	0.05	0.14	0.31	0.28	0.18	0.04	0
Angry3	0.06	0.15	0.33	0.28	0.15	0.02	0

And now each of the subscales:

```
ObjAlpha <- psych::alpha(Objetification) #creating an object from this analysis so I can extract it later
ObjAlpha
```

Reliability analysis
Call: psych::alpha(x = Objetification)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r	
0.8	0.8	0.79		0.29	4.1	0.018	1.9	0.65	0.28

lower alpha	upper	95% confidence boundaries
0.77	0.8	0.84

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
Obj1	0.77	0.78	0.76		0.28	3.5	0.021	0.0038	0.27
Obj2	0.77	0.77	0.76		0.28	3.4	0.021	0.0030	0.26
Obj3	0.78	0.78	0.77		0.28	3.6	0.020	0.0038	0.27
Obj4	0.78	0.78	0.77		0.28	3.6	0.020	0.0044	0.26
Obj5	0.78	0.79	0.77		0.29	3.7	0.020	0.0044	0.28
Obj6	0.79	0.79	0.78		0.29	3.8	0.019	0.0040	0.28
Obj7	0.78	0.79	0.77		0.29	3.7	0.020	0.0044	0.28
Obj8	0.79	0.79	0.78		0.30	3.8	0.019	0.0045	0.30
Obj9	0.79	0.80	0.78		0.30	3.9	0.019	0.0038	0.31
Obj10	0.79	0.80	0.78		0.30	3.9	0.019	0.0039	0.31

Item statistics

n	raw.r	std.r	r.cor	r.drop	mean	sd	
Obj1	259	0.67	0.67	0.62	0.56	1.8	1.09

```

Obj2 259 0.69 0.69 0.65 0.58 1.9 1.14
Obj3 259 0.64 0.64 0.59 0.53 2.0 1.01
Obj4 259 0.62 0.64 0.59 0.53 1.9 0.89
Obj5 259 0.62 0.61 0.54 0.48 2.1 1.19
Obj6 259 0.60 0.57 0.50 0.44 1.8 1.29
Obj7 259 0.60 0.60 0.53 0.48 2.0 1.09
Obj8 259 0.55 0.56 0.48 0.43 2.2 1.00
Obj9 259 0.50 0.52 0.42 0.38 1.8 0.99
Obj10 259 0.51 0.52 0.42 0.37 1.9 1.05

```

Non missing response frequency for each item

	0	1	2	3	4	5	miss
Obj1	0.12	0.30	0.30	0.22	0.05	0.00	0
Obj2	0.12	0.25	0.31	0.24	0.07	0.01	0
Obj3	0.06	0.28	0.36	0.24	0.07	0.00	0
Obj4	0.04	0.28	0.45	0.19	0.04	0.00	0
Obj5	0.12	0.19	0.35	0.22	0.12	0.01	0
Obj6	0.17	0.27	0.24	0.20	0.09	0.02	0
Obj7	0.09	0.24	0.37	0.22	0.06	0.01	0
Obj8	0.04	0.20	0.35	0.33	0.08	0.00	0
Obj9	0.10	0.26	0.41	0.19	0.04	0.00	0
Obj10	0.09	0.27	0.39	0.17	0.08	0.00	0

```

MargAlpha <- psych::alpha(Marginalization) #creating an object from this analysis so I can ext
MargAlpha

```

Reliability analysis

Call: psych::alpha(x = Marginalization)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.83	0.84	0.83		0.42	5.1	0.017	2.7	0.84
								0.42

lower alpha	upper	95% confidence boundaries
0.79	0.83	0.86

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
Marg1	0.77	0.78	0.75		0.37	3.5	0.022	0.0069	0.40
Marg2	0.79	0.80	0.79		0.40	4.0	0.020	0.0148	0.40
Marg3	0.80	0.81	0.80		0.42	4.3	0.020	0.0162	0.42
Marg4	0.80	0.82	0.81		0.43	4.5	0.019	0.0153	0.42
Marg5	0.81	0.82	0.81		0.44	4.7	0.018	0.0147	0.42
Marg6	0.82	0.83	0.82		0.45	4.9	0.017	0.0135	0.42
Marg7	0.82	0.84	0.82		0.46	5.1	0.017	0.0137	0.44

Item statistics

n	raw.r	std.r	r.cor	r.drop	mean	sd
---	-------	-------	-------	--------	------	----

```
Marg1 259  0.86  0.87  0.88   0.81  2.0 1.02
Marg2 259  0.77  0.78  0.75   0.68  3.5 0.99
Marg3 259  0.73  0.73  0.67   0.60  2.4 1.30
Marg4 259  0.69  0.70  0.63   0.56  3.3 1.17
Marg5 259  0.67  0.66  0.58   0.52  2.4 1.31
Marg6 259  0.64  0.62  0.53   0.47  2.9 1.37
Marg7 259  0.60  0.60  0.49   0.45  2.7 1.21
```

Non missing response frequency for each item

	0	1	2	3	4	5	miss
Marg1	0.06	0.24	0.41	0.22	0.08	0.00	0
Marg2	0.00	0.03	0.12	0.34	0.36	0.15	0
Marg3	0.07	0.17	0.27	0.25	0.18	0.05	0
Marg4	0.00	0.05	0.20	0.29	0.25	0.20	0
Marg5	0.08	0.17	0.31	0.24	0.14	0.07	0
Marg6	0.06	0.10	0.24	0.26	0.22	0.13	0
Marg7	0.03	0.12	0.27	0.35	0.14	0.08	0

```
StrongAlpha <- psych::alpha(Strong) #creating an object from this analysis so I can extract an alpha value later
```

Reliability analysis

Call: psych::alpha(x = Strong)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.63	0.63	0.59		0.26	1.7	0.035	1.6	0.59
								0.26

lower	alpha	upper	95% confidence boundaries		
0.56	0.63	0.7			

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
Strong1	0.57	0.58	0.51		0.26	1.4	0.043	0.0014	0.26
Strong2	0.57	0.57	0.51		0.25	1.3	0.043	0.0027	0.25
Strong3	0.58	0.58	0.51		0.26	1.4	0.041	0.0036	0.24
Strong4	0.56	0.56	0.49		0.24	1.3	0.045	0.0010	0.25
Strong5	0.61	0.61	0.54		0.28	1.6	0.039	0.0016	0.27

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
Strong1	259	0.64	0.64	0.50	0.40	1.3	0.88
Strong2	259	0.66	0.65	0.50	0.40	2.3	0.95
Strong3	259	0.61	0.64	0.49	0.38	1.5	0.83
Strong4	259	0.71	0.67	0.54	0.43	1.6	1.10
Strong5	259	0.57	0.59	0.41	0.33	1.4	0.83

Non missing response frequency for each item

	0	1	2	3	4	5	miss
Strong1	0.20	0.42	0.31	0.08	0.00	0	0
Strong2	0.03	0.18	0.35	0.37	0.07	0	0
Strong3	0.12	0.41	0.38	0.09	0.00	0	0
Strong4	0.18	0.29	0.32	0.16	0.04	0	0
Strong5	0.14	0.38	0.41	0.07	0.00	0	0

```
AngryAlpha <- psych::alpha(Angry) #creating an object from this analysis so I can extract and manipulate it
```

Reliability analysis
Call: psych::alpha(x = Angry)

raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
0.71	0.71	0.62	0.45	2.4	0.031	2.3	0.93	0.44

lower alpha upper 95% confidence boundaries
0.65 0.71 0.77

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
Angry1	0.63	0.64	0.47	0.47	1.7	0.045	NA	0.47	
Angry2	0.60	0.60	0.43	0.43	1.5	0.049	NA	0.43	
Angry3	0.61	0.61	0.44	0.44	1.6	0.048	NA	0.44	

Item statistics

n	raw.r	std.r	r.cor	r.drop	mean	sd	
Angry1	259	0.78	0.79	0.61	0.51	2.0	1.2
Angry2	259	0.81	0.80	0.64	0.54	2.5	1.2
Angry3	259	0.79	0.80	0.63	0.53	2.4	1.2

Non missing response frequency for each item

	0	1	2	3	4	5	miss
Angry1	0.09	0.25	0.31	0.26	0.08	0.02	0
Angry2	0.05	0.14	0.31	0.28	0.18	0.04	0
Angry3	0.06	0.15	0.33	0.28	0.15	0.02	0

8.7.1.2 Correlating items with other subscale totals

```
Obj_othR <- psych::corr.test(dfGRMS[c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9", "Obj10"), c("Marg1", "Marg2", "Marg3", "Marg4", "Marg5", "Marg6", "Marg7", "Marg8", "Marg9", "Marg10")])
```

```
Marg_othR <- psych::corr.test(dfGRMS[c("Marg1", "Marg2", "Marg3", "Marg4", "Marg5", "Marg6", "Marg7", "Marg8", "Marg9", "Marg10"), c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9", "Obj10")])
```

```
Str_othR <- psych::corr.test(dfGRMS[c("Strong1", "Strong2", "Strong3", "Strong4", "Strong5", "Strong6", "Strong7", "Strong8", "Strong9", "Strong10"),])
Ang_othR <- psych::corr.test(dfGRMS[c("Angry1", "Angry2", "Angry3", "Objectified", "Marginalized", "Total")])
```

8.7.1.3 Extracting values, binding them together, and joining the files

```
#names(Obj_other)
#Extracting the item-level statistics from the alpha object
Obj_othR <- as.data.frame(Obj_othR$r)#Makes the item-total(other) correlation matrix a df
#Adding variable names so we don't get lost
Obj_othR$Items <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9", "Obj10")
#deleting the rows with the total scale scores
Obj_othR <- Obj_othR[!Obj_othR$Items == "Marginalized",]
Obj_othR <- Obj_othR[!Obj_othR$Items == "Strong",]
Obj_othR <- Obj_othR[!Obj_othR$Items == "Angry",]
Obj_othR[, 'Objectified'] <- NA #We need a column for this to bind the items, later.
Obj_othR <- dplyr::select(Obj_othR, Items, Objectified, Marginalized, Strong, Angry) #Putting them all in one place
#Item Corrected Total Correlations
ObjAlpha <- as.data.frame(ObjAlpha$item.stats)#Grabbing the alpha objet we created earlier and putting it in a df
ObjAlpha$Items <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9", "Obj10")
#Joining the two and selecting the vars of interest
ObjStats <- full_join(ObjAlpha, Obj_othR, by = "Items")
ObjStats$Objectified <- ObjStats$r.drop #Copy the item-corrected total (r.drop) into the ObjStats$Objectified
ObjStats <- dplyr::select(ObjStats, Items, Objectified, Marginalized, Strong, Angry)
#rm(ObjAlpha, Obj_othR) #It's messay, dropping all the no-longer-necessary objects from the Global Environment

#Extracting the item-level statistics from the alpha object
Marg_othR <- as.data.frame(Marg_othR$r)#Makes the item-total(other) correlation matrix a df
#Adding variable names so we don't get lost
Marg_othR$Items <- c("Marg1", "Marg2", "Marg3", "Marg4", "Marg5", "Marg6", "Marg7", "Obj")
#deleting the rows with the total scale scores
Marg_othR <- Marg_othR[!Marg_othR$Items == "Objectified",]
Marg_othR <- Marg_othR[!Marg_othR$Items == "Strong",]
Marg_othR <- Marg_othR[!Marg_othR$Items == "Angry",]
Marg_othR[, 'Marginalized'] <- NA #We need a column for this to bind the items, later.
Marg_othR <- dplyr::select(Marg_othR, Items, Objectified, Marginalized, Strong, Angry) #Putting them all in one place
#Item Corrected Total Correlations
MargAlpha <- as.data.frame(MargAlpha$item.stats)#Grabbing the alpha objet we created earlier and putting it in a df
MargAlpha$Items <- c("Marg1", "Marg2", "Marg3", "Marg4", "Marg5", "Marg6", "Marg7")
#Joining the two and selecting the vars of interest
MargStats <- full_join(MargAlpha, Marg_othR, by = "Items")
MargStats$Marginalized <- MargStats$r.drop #Copy the item-corrected total (r.drop) into the MargStats$Marginalized
MargStats <- dplyr::select(MargStats, Items, Objectified, Marginalized, Strong, Angry)
#rm(MargAlpha, Marg_othR) #It's messay, dropping all the no-longer-necessary objects from the Global Environment
```

```

Str_othR <- as.data.frame(Str_othR$r) #Makes the item-total(other) correlation matrix a df
#Adding variable names so we don't get lost
Str_othR$Items <- c("Strong1", "Strong2", "Strong3", "Strong4", "Strong5", "Objectified", "Marginalized")
#deleting the rows with the total scale scores
Str_othR <- Str_othR[!Str_othR$Items == "Objectified",]
Str_othR <- Str_othR[!Str_othR$Items == "Marginalized",]
Str_othR <- Str_othR[!Str_othR$Items == "Angry",]
Str_othR[, 'Strong'] <- NA
Str_othR <- dplyr::select(Str_othR, Items, Objectified, Marginalized, Strong, Angry)
#Item Corrected Total Correlations
StrongAlpha <- as.data.frame(StrongAlpha$item.stats) #Grabbing the alpha objet we created earlier
StrongAlpha$Items <- c("Strong1", "Strong2", "Strong3", "Strong4", "Strong5")
#Joining the two and selecting the vars of interest
StrStats <- full_join(StrongAlpha, Str_othR, by = "Items")
StrStats$Strong <- StrStats$r.drop #Copy the item-corrected total (r.drop) into the Strong variable
StrStats <- dplyr::select(StrStats, Items, Objectified, Marginalized, Strong, Angry)
rm(StrongAlpha, Str_othR) #It's messay, dropping all the no-longer-necessary objects from the Global Env

Ang_othR <- as.data.frame(Ang_othR$r) #Makes the item-total(other) correlation matrix a df
#Adding variable names so we don't get lost
Ang_othR$Items <- c("Angry1", "Angry2", "Angry3", "Objectified", "Marginalized", "Strong")
#deleting the rows with the total scale scores
Ang_othR <- Ang_othR[!Ang_othR$Items == "Objectified",]
Ang_othR <- Ang_othR[!Ang_othR$Items == "Marginalized",]
Ang_othR <- Ang_othR[!Ang_othR$Items == "Strong",]
Ang_othR[, 'Angry'] <- NA
Ang_othR <- dplyr::select(Ang_othR, Items, Objectified, Marginalized, Strong, Angry)
#Item Corrected Total Correlations
AngryAlpha <- as.data.frame(AngryAlpha$item.stats) #Grabbing the alpha objet we created earlier
AngryAlpha$Items <- c("Angry1", "Angry2", "Angry3")
#Joining the two and selecting the vars of interest
AngStats <- full_join(AngryAlpha, Ang_othR, by = "Items")
AngStats$Angry <- AngStats$r.drop #Copy the item-corrected total (r.drop) into the Angry variable
AngStats <- dplyr::select(AngStats, Items, Objectified, Marginalized, Strong, Angry)
rm(AngryAlpha, Ang_othR) #It's messay, dropping all the no-longer-necessary objects from the Global Env

#Adding all the variables into a single table
ItemAnalysis <- rbind(ObjStats, MargStats, StrStats, AngStats)

#Preparing and adding the r.drop for total scale score
TotAlpha <- as.data.frame(GRMSalpha$item.stats)
TotAlpha$Items <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9", "Obj10")
TotAlpha <- dplyr::select(TotAlpha, Items, r.drop) #deleting the rows with the total scale scores

#Adding the r.drop for the total scale score
ItemAnalysis <- full_join(TotAlpha, ItemAnalysis, by = "Items")

```

```

#Adding the values from the Orthogonal rotation
pcaORTH_loadings <- data.frame(unclass(pcaORTH$loadings)) #I had to add "unclass" to the loadings
pcaORTH_loadings$Items <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9")
#Deleting those lower rows
#pcaORTH_loadings <- pcaORTH_loadings[!pcaORTH_loadings$Items == "GRMSTot",]
#pcaORTH_loadings <- pcaORTH_loadings[!pcaORTH_loadings$Items == "Objectified",]
#pcaORTH_loadings <- pcaORTH_loadings[!pcaORTH_loadings$Items == "Marginalized",]
#pcaORTH_loadings <- pcaORTH_loadings[!pcaORTH_loadings$Items == "Strong",]
#pcaORTH_loadings <- pcaORTH_loadings[!pcaORTH_loadings$Items == "Angry",]
pcaORTH_loadings <- rename(pcaORTH_loadings, objORTH = RC1, margORTH = RC2, strORTH = RC3, angORTH = RC4)

#Joining with the Item Stats
Comparisons <- full_join(ItemAnalysis, pcaORTH_loadings, by = "Items")#I had to add "unclass" to the loadings

#Adding the oblique loadings
pcaOBLQ_loadings <- data.frame(unclass(pcaOBLQ$loadings)) #I had to add "unclass" to the loadings
pcaOBLQ_loadings$Items <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9")
#Deleting those lower rows
#pcaOBLQ_loadings <- pcaOBLQ_loadings[!pcaOBLQ_loadings$Items == "GRMSTot",]
#pcaOBLQ_loadings <- pcaOBLQ_loadings[!pcaOBLQ_loadings$Items == "Objectified",]
#pcaOBLQ_loadings <- pcaOBLQ_loadings[!pcaOBLQ_loadings$Items == "Marginalized",]
#pcaOBLQ_loadings <- pcaOBLQ_loadings[!pcaOBLQ_loadings$Items == "Strong",]
#pcaOBLQ_loadings <- pcaOBLQ_loadings[!pcaOBLQ_loadings$Items == "Angry",]
pcaOBLQ_loadings <- rename(pcaOBLQ_loadings, objOBLQ = TC1, margOBLQ = TC2, strOBLQ = TC3, angOBLQ = TC4)

#Joining with the Item Stats
Comparisons <- full_join(Comparisons, pcaOBLQ_loadings, by = "Items")#I had to add "unclass" to the loadings

write.csv(Comparisons, file = "GRMS_Comparisons.csv", sep = ",", row.names=FALSE, col.names=TRUE)

```

Warning in write.csv(Comparisons, file = "GRMS_Comparisons.csv", sep = ",", :
attempt to set 'col.names' ignored

Warning in write.csv(Comparisons, file = "GRMS_Comparisons.csv", sep = ",", :
attempt to set 'sep' ignored

```
saveRDS(Comparisons, "GRMS_Comparisons.rds")#Writes the file as an .rds so that if anything is changed, it can be loaded back in
```

8.7.1.4 Interpreting the result

The result of this work is a table that includes:

- **r.drop** Corrected item-total (entire GRMS) coefficients
- **Item-total correlations** of the items correlated with their own subscale (bold; correlation does not include the item being correlated) and the other subscales

- **PCA: Orthogonal rotation** factor loadings of the four-scales with a rotation that maximizes the independents (uncorrelatedness) of the scales
- **PCA: Oblique rotation** factor loadings of the four-scales with a rotation that permits correlation between subscales

Items	Total Scale	Item Corrected-Total (bold) and Item-Other-Scale Correlations				Principal Components Analysis: Orthogonal Rotation				Principal Components Analysis: Oblique Rotation			
		r.drop	Objectified	Marginalized	Strong	Angry	objORTH	margORTH	strORTH	angORTH2	objOBLQ	margOBLQ	strOBLQ
Obj1	0.34	0.56	0.08	-0.03	-0.06	0.69	0.08	-0.10	-0.09	0.68	0.08	-0.07	-0.08
Obj2	0.28	0.58	-0.07	-0.06	0.01	0.72	-0.10	-0.14	0.00	0.72	-0.09	-0.11	0.02
Obj3	0.36	0.53	0.01	0.03	0.13	0.65	-0.01	-0.04	0.17	0.66	-0.02	-0.01	0.19
Obj4	0.35	0.53	0.04	0.03	0.03	0.65	0.02	0.00	0.00	0.65	0.02	0.03	0.01
Obj5	0.25	0.48	-0.15	0.15	0.03	0.61	-0.21	0.14	0.05	0.62	-0.21	0.17	0.06
Obj6	0.39	0.44	0.20	-0.03	0.07	0.56	0.23	-0.11	0.08	0.55	0.23	-0.09	0.10
Obj7	0.40	0.48	0.21	0.06	-0.07	0.58	0.24	0.05	-0.16	0.58	0.25	0.07	-0.15
Obj8	0.28	0.43	-0.02	0.13	-0.02	0.54	-0.04	0.15	-0.06	0.54	-0.04	0.17	-0.06
Obj9	0.34	0.38	0.06	0.17	0.08	0.47	0.04	0.22	0.11	0.47	0.04	0.23	0.11
Obj10	0.31	0.37	0.08	0.18	-0.04	0.47	0.06	0.25	-0.10	0.46	0.06	0.27	-0.10
Marg1	0.46	0.10	0.81	-0.03	-0.02	0.06	0.89	-0.02	-0.08	0.03	0.89	-0.03	-0.08
Marg2	0.38	-0.03	0.68	0.02	0.16	-0.09	0.80	0.02	0.14	-0.11	0.79	0.01	0.14
Marg3	0.34	0.06	0.60	-0.06	0.05	0.03	0.73	-0.08	0.02	0.01	0.73	-0.08	0.03
Marg4	0.41	0.16	0.56	-0.05	0.09	0.16	0.69	-0.10	0.07	0.14	0.69	-0.10	0.07
Marg5	0.38	0.20	0.52	-0.10	0.04	0.21	0.65	-0.15	0.01	0.20	0.65	-0.15	0.02
Marg6	0.28	0.03	0.47	0.19	-0.13	-0.03	0.61	0.32	-0.25	-0.05	0.61	0.31	-0.26
Marg7	0.23	-0.16	0.45	0.16	0.20	-0.24	0.57	0.22	0.23	-0.26	0.56	0.20	0.21
Strong1	0.18	0.04	0.00	0.40	0.20	0.00	-0.03	0.61	0.27	0.00	-0.04	0.61	0.24
Strong2	0.13	0.09	-0.04	0.40	0.02	0.06	-0.09	0.64	0.00	0.06	-0.10	0.64	-0.02
Strong3	0.13	0.15	-0.06	0.38	-0.08	0.16	-0.10	0.60	-0.15	0.15	-0.10	0.61	-0.17
Strong4	0.17	-0.01	0.04	0.43	0.21	-0.06	0.01	0.66	0.28	-0.06	-0.01	0.65	0.25
Strong5	0.19	0.07	0.15	0.33	-0.08	0.04	0.17	0.56	-0.18	0.03	0.17	0.56	-0.20
Angry1	0.22	0.18	0.03	-0.03	0.51	0.22	0.01	-0.13	0.75	0.23	-0.01	-0.12	0.76
Angry2	0.16	-0.02	0.05	0.16	0.54	-0.03	0.03	0.13	0.77	-0.03	0.01	0.13	0.77
Angry3	0.13	-0.09	0.09	0.11	0.53	-0.11	0.10	0.06	0.77	-0.11	0.08	0.06	0.77

Figure 8.7: Image of a table of values from the item analysis and PCA solutions with orthogonal and oblique rotations

We are looking for:

- items that *load* higher on their own scales than they do on other scales
- when they are “close” or have a number of strong loadings, it means that it’s not going to discriminate well (think within-in scale discriminant validity).
 - if there are a number of these, there will likely be stronger correlations between subscales (indicating that the oblique rotation was an appropriate choice).
 - with low/no cross-loadings, this supports the choices of an orthogonal (uncorrelated) solution
- when the item has a strong, positive loading on its own scale, it supports within-scale convergent validity.
- similarities and differences across the item-analysis, PCA orthogonal, and PCA oblique solutions. Our biggest interest is in whether items change scale membership and/or have cross-loadings. This scale is performing extremely well with a great deal of stability
 - This, in part, is likely facilitated by the data simulation where we had the benefit of factors “telling” items where to load.

8.8 Practice Problems

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. The least complex is to change the random seed in the research and rework the problem demonstrated in the lesson. The most complex is to use data of your own. In either case, please plan to:

- Properly format and prepare the data.
- Conduct diagnostic tests to determine the suitability of the data for PCA.
- Conducting tests to guide the decisions about number of components to extract.
- Conducting orthogonal and oblique extractions (at least two each with different numbers of components).
- Selecting one solution and preparing an APA style results section (with table and figure).

8.8.1 Problem #1: Play around with this simulation.

Copy the script for the simulation and then change (at least) one thing in the simulation to see how it impacts the results. If PCA is new to you, perhaps you just change the number in “set.seed(210921)” from 210921 to something else. Your results should parallel those obtained in the lecture, making it easier for you to check your work as you go.

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Conduct and interpret the three diagnostic tests to determine if PCA is appropriate as an analysis (KMO, Bartlett’s, determinant).	5	_____
3. Determine how many components to extract (e.g., scree plot, eigenvalues, theory).	5	_____
4. Conduct an orthogonal extraction and rotation.	5	_____
5. Conduct an oblique extraction and rotation.	5	_____
6. Repeat the orthogonal and oblique extractions/rotations with a different number of specified factors.	5	_____
7. APA style results section with table and figure of one of the solutions.	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

8.8.2 Problem #2: Conduct a PCA with the Szymanski and Bissonette (?) research vignette that was used in prior lessons.

The second option involves utilizing one of the simulated datasets available in this OER. Szymanski and Bissonette’s (?)Perceptions of the LGBTQ College Campus Climate Scale: Development and

psychometric evaluation was used as the research vignette for the validity, reliability, and item analysis lessons. Although I switched vignettes, the Szymanski and Bissonette example is ready for PCA.

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Conduct and interpret the three diagnostic tests to determine if PCA is appropriate as an analysis (KMO, Bartlett's, determinant).	5	_____
3. Determine how many components to extract (e.g., scree plot, eigenvalues, theory).	5	_____
4. Conduct an orthogonal extraction and rotation.	5	_____
5. Conduct an oblique extraction and rotation.	5	_____
6. Repeat the orthogonal and oblique extractions/rotations with a different number of specified factors.	5	_____
7. APA style results section with table and figure of one of the solutions.	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

8.8.3 Problem #3: Try something entirely new.

Using data for which you have permission and access (e.g., IRB approved data you have collected or from your lab; data you simulate from a published article; data from an open science repository; data from other chapters in this OER), complete PCA. The data should allow for at least two (ideally three) components/subscales.

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Conduct and interpret the three diagnostic tests to determine if PCA is appropriate as an analysis (KMO, Bartlett's, determinant).	5	_____
3. Determine how many components to extract (e.g., scree plot, eigenvalues, theory).	5	_____
4. Conduct an orthogonal extraction and rotation.	5	_____
5. Conduct an oblique extraction and rotation.	5	_____
6. Repeat the orthogonal and oblique extractions/rotations with a different number of specified factors.	5	_____

Assignment Component	Points Possible	Points Earned
7. APA style results section with table and figure of one of the solutions.	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

Chapter 9

Principal Axis Factoring

[Screencasted Lecture Link](#)

This is the second week of *exploratory* principal components analysis (PCA) and factor analysis (EFA). This time the focus is on actual *factor analysis*. There are numerous approaches. I will be demonstrating principal axis factoring (PAF).

9.1 Navigating this Lesson

There is about an hour-and-a-half of lecture. If you work through the materials with me it would be plan for an additional two hours.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

9.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Distinguish between PCA and EFA on several levels:
 - recognize PCA and EFA from a path diagram
 - define keywords associated with each: factor loadings, linear components, describe v. explain.
- Recognize/define an identity matrix – what test would you use to diagnose it?
- Recognize/define multicollinearity and singularity – what test would you use to diagnose it?
- Describe the desired pattern of “loadings” (i.e., the relative weights of an item on its own scale compared to other scales)
- Compare the results from item analysis, PCA, PAF, and omega.

9.1.2 Planning for Practice

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. Whichever you choose, it would be terrific if you used the same dataframe across as many psychometrics lessons as possible so you can compare the results.

The least complex is to change the random seed and rework the problem demonstrated in the lesson. The results *should* map onto the ones obtained in the lecture.

The second option involves utilizing one of the simulated datasets available in this OER. Szymanski and Bissonette's (?) Perceptions of the LGBTQ College Campus Climate Scale: Development and Psychometric Evaluation was used as the research vignette for the validity, reliability, and item analysis lessons. Although I switched vignettes, the Szymanski and Bissonette example is ready for PCA.

As a third option, you are welcome to use data to which you have access and is suitable for PCA. These could include other vignettes from this OER, other simulated data, or your own data (presuming you have permission to use it). In either case, please plan to:

- Properly format and prepare the data.
- Conduct diagnostic tests to determine the suitability of the data for PCA.
- Conducting tests to guide the decisions about number of components to extract.
- Conducting orthogonal and oblique extractions (at least two each with different numbers of components).
- Selecting one solution and preparing an APA style results section (with table and figure).
- Compare your results in light of any other psychometrics lessons where you have used this data (especially the **item analysis** and **PCA** lessons).

9.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (when possible, linked) in the text with complete citations in the reference list.

- Revelle, William. (n.d.). Chapter 6: Constructs, components, and factor models. In *An introduction to psychometric theory with applications in R*. Retrieved from <https://personality-project.org/r/book/#chapter6>
 - pp. 150 to 167. Stop at “Non-Simple Structure Solutions: The Simplex and Circumplex.”
 - A simultaneously theoretical review of psychometric theory while working with R and data to understand the concepts.
- Revelle, W. (2019). *How To: Use the psych package for Factor Analysis and data reduction*.
 - Treat as reference. Pages 13 through 24 provide technical information about what we are doing.

9.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(psych)){install.packages("psych")}
#if(!require(tidyverse)){install.packages("tidyverse")}
#if(!require(MASS)){install.packages("MASS")}
#if(!require(sjstats)){install.packages("sjstats")}
#if(!require(apaTables)){install.packages("apaTables")}
#if(!require(qualTRics)){install.packages("qualTRics")}
```

9.2 Exploratory Factor Analysis (with a quick contrast to PCA)

Whereas principal components analysis (PCA) is a regression analysis technique, principal factor analysis is “...a latent variable model” (?).

Exploratory factor analysis has a rich history. In 1904, Spearman used it for a single factor. In 1947, Thurstone generalized it to multiple factors. Factor analysis is frequently used and controversial.

Factor analysis and principal components are commonly confused:

Principal components

- linear sums of variables,
- solved with an eigenvalue or singular decomposition
- represents a $n * n$ matrix in terms of the first k components and attempts to reproduce all of the R matrix.
- paths point from the items to a total scale score – all represented as observed/manifest (square) variables

Factor analysis

- linear sums of unknown factors
- estimated as best fitting solutions, normally through iterative procedures.
- Controversial because
 - at the *structural* level (i.e., covariance or correlation matrix), there are normally more observed variables than parameters to estimate them and the procedure seeks to find the best fitting solution using ordinary least squares, weighted least squares, or maximum likelihood
 - at the *data* level, the model is indeterminate, although scores can be estimated
 - this leads some to argue for using principal components; but fans of factor analysis suggest that it is useful for theory construction and evaluation
- attempts to model only the *common* part of the matrix, which means all of the off-diagonal elements and the common part of the diagonal (the *communalities*); the *uniquenesses* are the non-common (leftover) part
- Stated another way, the factor model partitions the correlation or covariance matrix into
 - *common factors*, FF' , and
 - that which is *unique*, U^2 (the diagonal matrix of *uniquenesses*)

- paths point from the latent variable (LV) representing the factor (oval) to the items (squares) illustrating that the factor/LV “causes” the item’s score

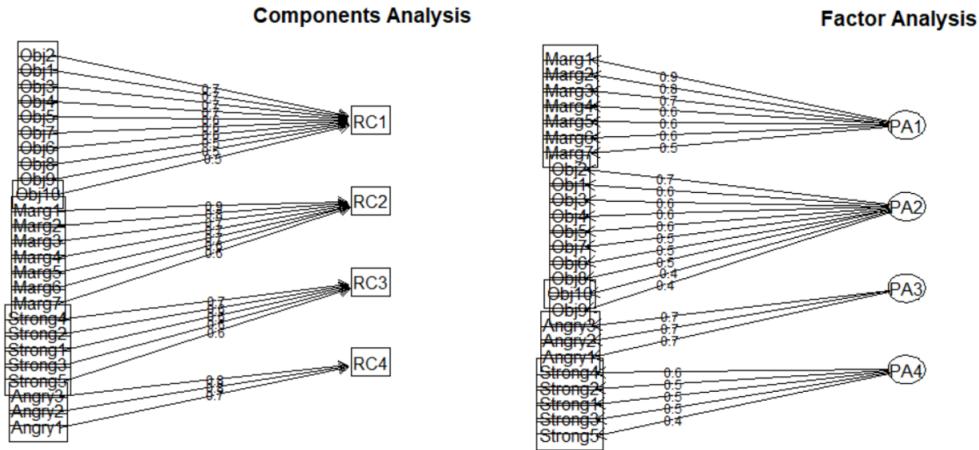


Figure 9.1: Comparison of path models for PCA and EFA

Our focus today is on the PAF approach to scale construction. By utilizing the same research vignette as in the [PCA lesson](#), we can identify similarities in differences in the approach, results, and interpretation. Let's first take a look at the workflow for PAF.

9.3 PAF Workflow

Below is a screenshot of the workflow. The original document is located in the [Github site](#) that hosts the ReCentering Psych Stats: Psychometrics OER. You may find it refreshing that, with the exception of the change from “components” to “factors,” the workflow for PCA and PAF are quite similar.

Steps in the process include:

- Creating an items only dataframe where all items are scaled in the same direction (i.e., negatively worded items are reverse-scored).
- Conducting tests that assess the statistical assumptions of PAF to ensure that the data is appropriate for PAF.
- Determining the number of factors (think “subscale”) to extract.
- Conducting the factor extraction – this process will likely occur iteratively,
 - exploring orthogonal (uncorrelated/independent) and oblique (correlated) factors, and
 - changing the number of factors to extract

Because the intended audience for the ReCentering Psych Stats OER is the scientist-practitioner-advocate, this lesson focuses on the workflow and decisions. As you might guess, the details of PAF can be quite complex. Some important notions to consider that may not be obvious from lesson, are these:

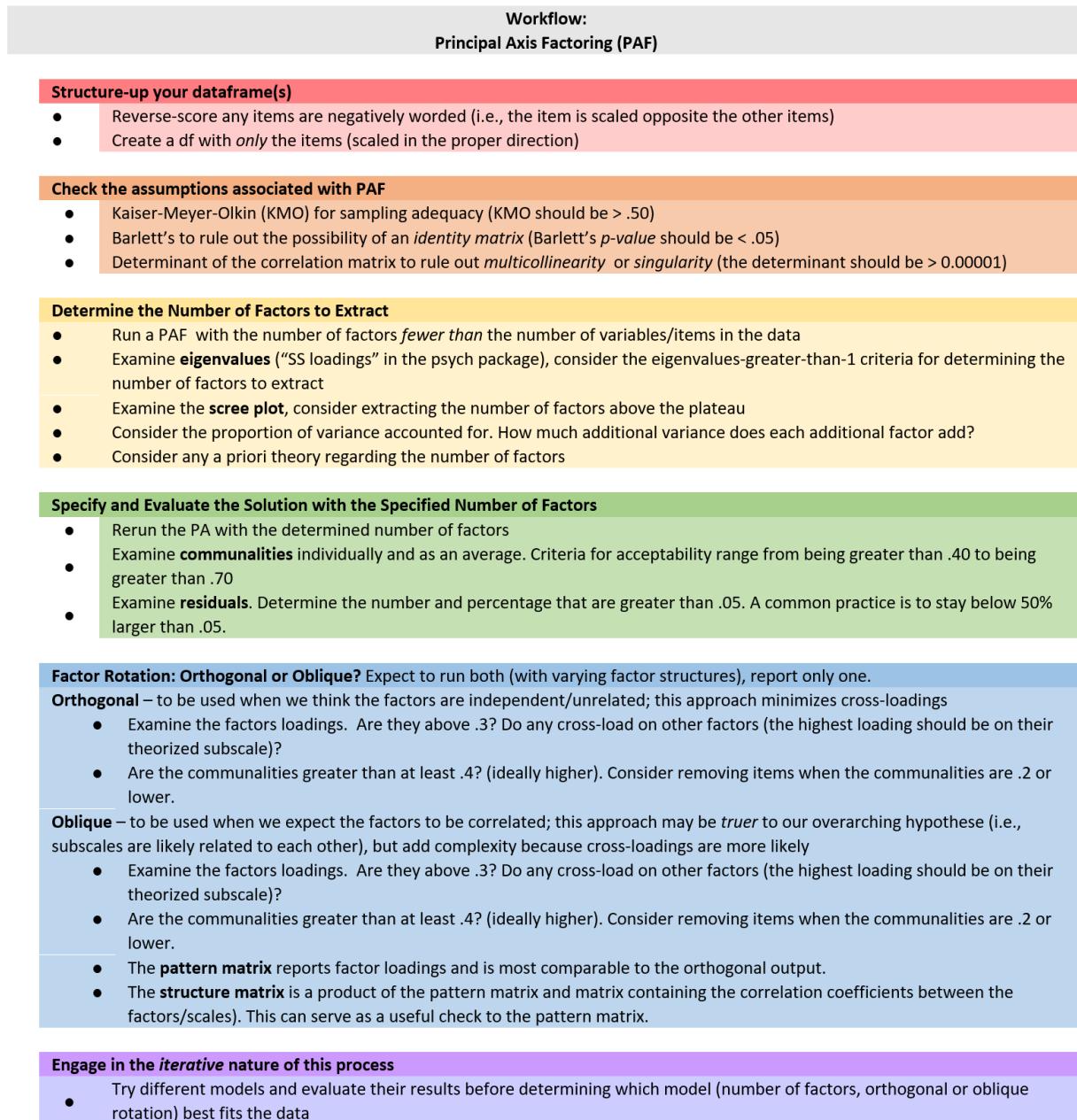


Figure 9.2: Image of the workflow for PAF

- The values of factor loadings are directly related to the correlation matrix.
 - Although I do not explain this in detail, nearly every analytic step attempts to convey this notion by presenting equivalent analytic options using the raw data and correlation matrix.
- PAF (like PCA and related EFA procedures) is about *dimension reduction* – our goal is fewer factors (think subscales) than there are items.
 - In this lesson's vignette there are 25 items on the scale and we will have 4 subscales.
- As a latent variable procedure, PAF is both *exploratory* and *factor analysis*. This is in contrast to our prior [PCA lesson](#). Recall that PCA is a regression-based model and therefore not “factor analysis.”
- Matrix algebra (e.g., using the transpose of a matrix, multiplying matrices together) plays a critical role in the analytic solution.

9.4 Research Vignette

This lesson's research vignette emerges from Lewis and Neville's, Gendered Racial Microaggressions Scale for Black Women (?). The article reports on two separate studies that comprised the development, refinement, and psychometric evaluation of two, parallel, versions (stress appraisal, frequency) of themselves. We simulate data from the final construction of the stress appraisal version as the basis of the lecture.

Lewis and Neville (?) reported support for a total scale score (25 items) and four subscales. Below, I list the four subscales, their number of items, and a single example item. At the outset, let me provide a content warning. For those who hold this particular identity (or related identities) the content in the items may be upsetting. In other lessons, I often provide a variable name that gives an indication of the primary content of the item. In the case of the GRMS, I will simply provide an abbreviation of the subscale name and its respective item number. This will allow us to easily inspect the alignment of the item with its intended factor/subscale, and hopefully minimize discomfort. If you are not a member of this particular identity, I encourage you to learn about these microaggressions by reading the article in its entirety. Please do not ask members of this group to explain why these microaggressions are harmful or ask if they have encountered them. The four factors, number of items, and sample item are as follows:

- Assumptions of Beauty and Sexual Objectification
 - 10 items
 - “Objectified me based on physical features.”
 - Abbreviated in the simulated data as “Obj#”
- Silenced and Marginalized
 - 7 items
 - “Someone has tried to ‘put me in my place.’”
 - Abbreviated in the simulated data as “Marg#”
- Strong Black Woman Stereotype

- 5 items
- “I have been told that I am too assertive.”
- Abbreviated in the simulated data as “Strong#”
- Angry Black Woman Stereotype
 - 3 items
 - “Someone accused me of being angry when speaking calm.”
 - Abbreviated in the simulated data as “Angry#”

Below, I walk through the data simulation. This is not an essential portion of the lesson, but I will lecture it in case you are interested. None of the items are negatively worded (relative to the other items), so there is no need to reverse-score any items. This is the same data used in the **PCA lesson**. If you have already simulated and exported it, you only need to import it.

```
set.seed(210921)
GRMSmat <- matrix(c(.69, .69, .60, .59, .55, .55, .54, .50, .41, .41, .04, -.15, .06, .12, .20
                     .07, -.07, .00, .07, -.18, .22, .23, -.01, .03, .02, .93, .81, .69, .67, .61
                     -.08, -.08, .00, .06, .16, -.06, .08, .16, .22, .23, -.04, .01, -.05, -.11,
                     -.06, .03, .16, .01, .05, .09, -.08, -.06, .07, -.03, -.08, .18, .03, .06, .09
rownames(GRMSmat) <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9",
#rownames(Szyf2) <- paste("V", seq(1:6), sep=" ") #prior code I replaced with above
colnames(GRMSmat) <- c("Objectified", "Marginalized", "Strong", "Angry")
GRMSCorMat <- GRMSmat %*% t(GRMSmat) #create the correlation matrix
diag(GRMSCorMat) <- 1
#SzyCorMat #prints the correlation matrix
GRMS_M <- c(1.78, 1.85, 1.97, 1.93, 2.01, 1.76, 1.91, 2.22, 1.83, 1.88, 2,
GRMS_SD <- c(1.11, 1.23, 0.97, 0.85, 1.19, 1.32, 1.04, 0.98, 1.01, 1.03, 1.01
GRMSCovMat <- GRMS_SD %*% t(GRMS_SD) * GRMSCorMat #creates a covariance matrix from the correlation matrix
#SzyCovMat #displays the covariance matrix

dfGRMS <- as.data.frame(round(MASS::mvrnorm(n=259, mu = GRMS_M, Sigma = GRMSCovMat, empirical = TRUE))
dfGRMS[dfGRMS>5] <- 5 #restricts the upperbound of all variables to be 5 or less
dfGRMS[dfGRMS<0] <- 0 #restricts the lowerbound of all variable to be 0 or greater
#colMeans(GRMS) #displays column means

#Below is code if you would like and ID number. For this lesson's purposes and ID number would be 1
#library(tidyverse)
#dfGRMS <- dfGRMS %>% dplyr::mutate(ID = row_number()) #add ID to each row
#dfGRMS <- dfGRMS %>% dplyr::select(ID, everything())#moving the ID number to the first column;
```

Let's take a quick peek at the data to see if everthing looks right.

```
psych::describe(dfGRMS)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis
Obj1	1	259	1.78	1.09	2	1.78	1.48	0	5	5	0.16	-0.63


```
Angry2 0.07
Angry3 0.07
```

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do).

```
#write the simulated data as a .csv
#write.table(dfGRMS, file="dfGRMS.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#dfGRMS <- read.csv ("dfGRMS.csv", header = TRUE)
```

```
#to save the df as an .rds (think "R object") file on your computer; it should save in the same
#saveRDS(dfGRMS, "dfGRMS.rds")
#bring back the simulated dat from an .rds file
#dfGRMS <- readRDS("dfGRMS.rds")
```

9.5 Working the Vignette

It may be useful to recall how we might understand factors in the psychometric sense:

- clusters of correlated items in an *R*-matrix
- statistical entities that can be plotted as classification axes where coordinates of variables along each axis represent the strength of the relationship between that variable to each factor.
- mathematical equations, resembling regression equations, where each variable is represented according to its relative weight

9.5.1 Data Prep

Since the first step is data preparation, let’s start by:

- reverse coding any items that are phrased in the opposite direction
- creating a *df* (as an object) that only contains the items in their properly scored direction (i.e., you might need to replace the original item with the reverse-coded item); there should be no other variables (e.g., ID, demographic variables, other scales) in this df
 - because the GRMS has no items like this we can skip these two steps

Our example today requires no reverse coding and the dataset I simulated only has item-level data (with no ID and no other variables). This means we are ready to start the PAF process.

```
psych::describe(dfGRMS)
```



```
Strong5 0.05
Angry1 0.07
Angry2 0.07
Angry3 0.07
```

Let's take a look at (and make an object of) the correlation matrix.

```
GRMSr <- cor(dfGRMS) #correlation matrix (with the negatively scored item already reversed)
round(GRMSr, 2)
```

	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6	Obj7	Obj8	Obj9	Obj10	Marg1	Marg2
Obj1	1.00	0.41	0.38	0.33	0.34	0.34	0.37	0.35	0.25	0.24	0.11	0.00
Obj2	0.41	1.00	0.43	0.40	0.40	0.34	0.35	0.29	0.28	0.24	-0.02	-0.13
Obj3	0.38	0.43	1.00	0.36	0.34	0.31	0.30	0.28	0.22	0.24	0.01	-0.03
Obj4	0.33	0.40	0.36	1.00	0.32	0.32	0.32	0.25	0.27	0.28	0.06	-0.03
Obj5	0.34	0.40	0.34	0.32	1.00	0.23	0.26	0.26	0.27	0.25	-0.14	-0.18
Obj6	0.34	0.34	0.31	0.32	0.23	1.00	0.31	0.24	0.19	0.16	0.20	0.16
Obj7	0.37	0.35	0.30	0.32	0.26	0.31	1.00	0.26	0.19	0.26	0.23	0.11
Obj8	0.35	0.29	0.28	0.25	0.26	0.24	0.26	1.00	0.21	0.22	0.00	-0.09
Obj9	0.25	0.28	0.22	0.27	0.27	0.19	0.19	0.21	1.00	0.22	0.07	0.01
Obj10	0.24	0.24	0.24	0.28	0.25	0.16	0.26	0.22	0.22	1.00	0.06	-0.03
Marg1	0.11	-0.02	0.01	0.06	-0.14	0.20	0.23	0.00	0.07	0.06	1.00	0.66
Marg2	0.00	-0.13	-0.03	-0.03	-0.18	0.16	0.11	-0.09	0.01	-0.03	0.66	1.00
Marg3	0.05	-0.01	0.02	0.04	-0.11	0.16	0.12	-0.03	0.07	0.02	0.62	0.50
Marg4	0.17	0.05	0.10	0.07	-0.03	0.21	0.21	0.05	0.05	0.07	0.58	0.47
Marg5	0.20	0.07	0.11	0.13	0.00	0.21	0.22	0.04	0.10	0.12	0.55	0.41
Marg6	0.00	-0.09	-0.06	0.03	-0.11	0.06	0.16	0.03	0.05	0.13	0.51	0.43
Marg7	-0.14	-0.22	-0.09	-0.13	-0.17	0.00	-0.01	-0.10	-0.04	-0.04	0.42	0.42
Strong1	-0.05	-0.07	0.03	-0.02	0.11	-0.05	0.01	0.03	0.14	0.10	-0.04	0.05
Strong2	0.01	-0.06	0.01	0.04	0.13	-0.01	0.04	0.13	0.12	0.13	-0.10	-0.06
Strong3	0.04	0.05	0.07	0.14	0.15	0.02	0.10	0.10	0.13	0.14	-0.06	-0.09
Strong4	-0.11	-0.11	0.01	-0.04	0.06	-0.04	-0.05	0.05	0.11	0.09	-0.04	0.05
Strong5	0.02	0.00	-0.01	0.01	0.04	0.00	0.10	0.11	0.05	0.11	0.14	0.10
Angry1	0.11	0.15	0.20	0.15	0.11	0.15	0.01	0.08	0.08	0.03	-0.04	0.06
Angry2	-0.12	-0.04	0.08	-0.02	0.01	0.01	-0.07	-0.01	0.07	-0.05	-0.03	0.11
Angry3	-0.12	-0.09	0.03	-0.06	-0.05	0.02	-0.12	-0.11	0.05	-0.07	0.02	0.21
	Marg3	Marg4	Marg5	Marg6	Marg7	Strong1	Strong2	Strong3	Strong4	Strong5		
Obj1	0.05	0.17	0.20	0.00	-0.14	-0.05	0.01	0.04	-0.11	0.02		
Obj2	-0.01	0.05	0.07	-0.09	-0.22	-0.07	-0.06	0.05	-0.11	0.00		
Obj3	0.02	0.10	0.11	-0.06	-0.09	0.03	0.01	0.07	0.01	-0.01		
Obj4	0.04	0.07	0.13	0.03	-0.13	-0.02	0.04	0.14	-0.04	0.01		
Obj5	-0.11	-0.03	0.00	-0.11	-0.17	0.11	0.13	0.15	0.06	0.04		
Obj6	0.16	0.21	0.21	0.06	0.00	-0.05	-0.01	0.02	-0.04	0.00		
Obj7	0.12	0.21	0.22	0.16	-0.01	0.01	0.04	0.10	-0.05	0.10		
Obj8	-0.03	0.05	0.04	0.03	-0.10	0.03	0.13	0.10	0.05	0.11		
Obj9	0.07	0.05	0.10	0.05	-0.04	0.14	0.12	0.13	0.11	0.05		
Obj10	0.02	0.07	0.12	0.13	-0.04	0.10	0.13	0.14	0.09	0.11		

Marg1	0.62	0.58	0.55	0.51	0.42	-0.04	-0.10	-0.06	-0.04	0.14
Marg2	0.50	0.47	0.41	0.43	0.42	0.05	-0.06	-0.09	0.05	0.10
Marg3	1.00	0.42	0.40	0.35	0.34	-0.06	-0.09	-0.09	-0.02	0.07
Marg4	0.42	1.00	0.44	0.28	0.27	-0.03	-0.08	-0.08	-0.04	0.06
Marg5	0.40	0.44	1.00	0.25	0.26	-0.10	-0.08	-0.10	-0.06	0.01
Marg6	0.35	0.28	0.25	1.00	0.29	0.10	0.09	0.10	0.11	0.23
Marg7	0.34	0.27	0.26	0.29	1.00	0.09	0.08	-0.01	0.19	0.13
Strong1	-0.06	-0.03	-0.10	0.10	0.09	1.00	0.26	0.24	0.34	0.19
Strong2	-0.09	-0.08	-0.08	0.09	0.08	0.26	1.00	0.27	0.31	0.21
Strong3	-0.09	-0.08	-0.10	0.10	-0.01	0.24	0.27	1.00	0.25	0.27
Strong4	-0.02	-0.04	-0.06	0.11	0.19	0.34	0.31	0.25	1.00	0.22
Strong5	0.07	0.06	0.01	0.23	0.13	0.19	0.21	0.27	0.22	1.00
Angry1	0.05	0.08	0.07	-0.16	0.08	0.09	-0.05	-0.10	0.06	-0.11
Angry2	0.02	0.06	-0.01	-0.08	0.19	0.22	0.05	-0.02	0.24	-0.02
Angry3	0.05	0.07	0.04	-0.09	0.20	0.18	0.03	-0.07	0.20	-0.05
	Angry1 Angry2 Angry3									
Obj1	0.11	-0.12	-0.12							
Obj2	0.15	-0.04	-0.09							
Obj3	0.20	0.08	0.03							
Obj4	0.15	-0.02	-0.06							
Obj5	0.11	0.01	-0.05							
Obj6	0.15	0.01	0.02							
Obj7	0.01	-0.07	-0.12							
Obj8	0.08	-0.01	-0.11							
Obj9	0.08	0.07	0.05							
Obj10	0.03	-0.05	-0.07							
Marg1	-0.04	-0.03	0.02							
Marg2	0.06	0.11	0.21							
Marg3	0.05	0.02	0.05							
Marg4	0.08	0.06	0.07							
Marg5	0.07	-0.01	0.04							
Marg6	-0.16	-0.08	-0.09							
Marg7	0.08	0.19	0.20							
Strong1	0.09	0.22	0.18							
Strong2	-0.05	0.05	0.03							
Strong3	-0.10	-0.02	-0.07							
Strong4	0.06	0.24	0.20							
Strong5	-0.11	-0.02	-0.05							
Angry1	1.00	0.44	0.43							
Angry2	0.44	1.00	0.47							
Angry3	0.43	0.47	1.00							

In case you want to examine it in sections (easier to view):

```
#round(GRMSr[,1:8], 2)
#round(GRMSr[,9:16], 2)
#round(GRMSr[,17:25], 2)
```

As with PCA, we can analyze the data with either raw data or correlation matrix. I will do both to demonstrate (a) that it's possible and to (b) continue emphasizing that this is a *structural* analysis. That is, we are trying to see if our more parsimonious extraction *reproduces* this original correlation matrix.

9.5.1.1 Three Diagnostic Tests to Evaluate the Appropriateness of the Data for Component (or Factor) Analysis

9.5.1.2 Is my sample adequate for PAF?

We return to the **KMO** (Kaiser-Meyer-Olkin), an index of *sampling adequacy* that can be used with the actual sample to let us know if the sample size is sufficient (or if we should collect more data).

Kaiser's 1974 recommendations were:

- bare minimum of .5
- values between .5 and .7 are mediocre
- values between .7 and .8 are good
- values above .9 are superb

We use the *KMO()* function from the *psych* package with either raw or matrix dat.

```
psych::KMO(dfGRMS)
```

```
Kaiser-Meyer-Olkin factor adequacy
Call: psych::KMO(r = dfGRMS)
Overall MSA =  0.86
MSA for each item =
  Obj1   Obj2   Obj3   Obj4   Obj5   Obj6   Obj7   Obj8   Obj9   Obj10
  0.89   0.88   0.90   0.90   0.91   0.92   0.92   0.90   0.88   0.90
  Marg1   Marg2   Marg3   Marg4   Marg5   Marg6   Marg7 Strong1 Strong2 Strong3
  0.83   0.88   0.91   0.91   0.91   0.87   0.90   0.79   0.79   0.80
Strong4 Strong5 Angry1 Angry2 Angry3
  0.79   0.81   0.72   0.75   0.75
```

```
#psych::KMO(GRMSr) #for the KMO function, do not specify sample size if using the matrix form
```

We examine the KMO values for both the overall matrix and the individual items.

At the matrix level, our $KMO = .86$, which falls in between Kaiser's definitions of *good* and *superb*.

At the item level, the KMO should be $> .50$. Variables with values below .5 should be evaluated for exclusion from the analysis (or run the analysis with and without the variable and compare the difference). Because removing/adding variables impacts the KMO, be sure to re-evaluate.

At the item level, our KMO values range between .72 (Angry1) and .92 (Obj6, Obj7).

Considering both item- and matrix- levels, we conclude that the sample size and the data are adequate for component (or factor) analysis.

9.5.1.3 Are there correlations among the variables that are big enough to be analyzed?

Bartlett's lets us know if a matrix is an *identity matrix*. In an identity matrix all correlation coefficients (everything on the off-diagonal) would be 0.0 (and everything on the diagonal would be 1.0).

A significant Barlett's (i.e., $p < .05$) tells that the R -matrix is not an identity matrix. That is, there are some relationships between variables that can be analyzed.

The `cortest.bartlett()` function in the `psych` package and can be run either from the raw data or R matrix formats.

```
psych::cortest.bartlett(dfGRMS) #from the raw data
```

R was not square, finding R from data

```
$chisq  
[1] 1683.76
```

```
$df  
[1] 300
```

#raw data produces the warning "R was not square, finding R from data." This means nothing other than that the raw data is not square.

```
#psych::cortest.bartlett(GRMSr, n = 259) #if using the matrix, must specify sample size
```

Our Bartlett's test is significant: $\chi^2(300) = 1683.76, p < .001$. This supports a component (or factor) analytic approach for investigating the data.

9.5.1.4 Is there multicollinearity or singularity in my data?

The **determinant of the correlation matrix** should be greater than 0.00001 (that would be 4 zeros before the 1). If it is smaller than 0.00001 then we may have an issue with *multicollinearity* (i.e., variables that are too highly correlated) or *singularity* (variables that are perfectly correlated).

The determinant function comes from base R. It is easiest to compute when the correlation matrix is the object. However, it is also possible to specify the command to work with the raw data.

```
#det(GRMSr)
det(cor(dfGRMS)) #if using the raw data
```

```
[1] 0.001151581
```

With a value of 0.00115, our determinant is greater than the 0.00001 requirement. If it were not, then we could identify problematic variables (i.e., those correlating too highly with others and those not correlating sufficiently with others) and re-run the diagnostic statistics.

Summary: Data screening were conducted to determine the suitability of the data for this analyses. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO; Kaiser, 1970) represents the ratio of the squared correlation between variables to the squared partial correlation between variables. KMO ranges from 0.00 to 1.00 – values closer to 1.00 indicate that the patterns of correlations are relatively compact and that component analysis should yield distinct and reliable components (Field, 2012). In our dataset, the KMO value was .86, indicating acceptable sampling adequacy. The Bartlett's Test of Sphericity examines whether the population correlation matrix resembles an identity matrix (Field, 2012). When the p value for the Bartlett's test is $< .05$, we are fairly certain we have clusters of correlated variables. In our dataset, $\chi^2(300) = 1683.76, p < .001$, indicating the correlations between items are sufficiently large enough for principal axis factoring. The determinant of the correlation matrix alerts us to any issues of multicollinearity or singularity and should be larger than 0.00001. Our determinant was 0.00115 and, again, indicated that our data was suitable for the analysis.

Note: If this looks familiar, it is! The same diagnostics are used in PAF and PCA.

9.5.2 Principal Axis Factoring (PAF)

We can use the `fa()` function, specifying `fm = "pa"` from the `psych` package with raw or matrix data.

One difference from PCA is that factor analysis will not (cannot) calculate as many factors as there are items. This means that we should select a reasonable number, like 20 (since there are 25 items). However, I received a number of errors/warnings and 10 is the first number that would run. I also received the warning, “maximum iteration exceeded.” Therefore I increased “max.iter” to 100.

Our goal is to begin to get an idea of the cumulative variance explained and number of factors to extract. If we think there are four factors, we simply need to specify more than four factors on the `nfactors = ##` command. As long as that number is less than the total number of items, it does not matter what that number is.

```
#grmsPAF1 <- psych::fa(GRMSr, nfactors=10, fm = "pa", max.iter = 100, rotate="none")# using the
#grmsPAF1 <- psych::fa(dfGRMS, nfactors = 10, fm = "pa", max.iter = 100, rotate = "none")# using the
#I received the warning "maximum iteration exceeded". It gave output, but it's best if we don't
grmsPAF1 #this object holds a great deal of information
```

```
Factor Analysis using method = pa
Call: psych::fa(r = dfGRMS, nfactors = 10, rotate = "none", max.iter = 100,
               fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
          PA1    PA2    PA3    PA4    PA5    PA6    PA7    PA8    PA9    PA10   h2    u2
Obj1     0.44   0.46  -0.14 -0.04   0.09  -0.15 -0.06  -0.09   0.00  -0.08  0.48  0.52
```

Obj2	0.35	0.59	-0.11	-0.14	-0.03	0.13	-0.08	-0.04	0.02	0.14	0.54	0.46	
Obj3	0.37	0.46	0.07	-0.15	0.08	0.06	0.02	-0.07	0.01	0.02	0.39	0.61	
Obj4	0.39	0.46	-0.01	-0.03	-0.09	0.20	0.13	0.15	0.01	-0.08	0.46	0.54	
Obj5	0.21	0.55	0.11	0.02	-0.03	-0.02	0.05	-0.12	0.02	0.02	0.38	0.62	
Obj6	0.47	0.26	-0.04	-0.13	0.06	0.06	-0.03	0.03	0.16	-0.10	0.35	0.65	
Obj7	0.49	0.29	-0.09	0.09	0.07	0.06	0.02	-0.06	-0.05	-0.01	0.36	0.64	
Obj8	0.29	0.42	0.05	0.10	0.22	-0.21	-0.15	0.19	-0.02	0.01	0.42	0.58	
Obj9	0.32	0.32	0.19	0.05	-0.43	-0.17	-0.10	0.06	0.04	0.03	0.48	0.52	
Obj10	0.31	0.31	0.08	0.18	-0.04	-0.01	0.14	0.05	-0.19	-0.03	0.29	0.71	
Marg1	0.74	-0.50	-0.15	0.08	-0.03	0.02	-0.03	0.00	-0.01	0.07	0.83	0.17	
Marg2	0.56	-0.56	0.06	-0.05	-0.01	0.07	-0.13	-0.06	0.05	-0.17	0.69	0.31	
Marg3	0.54	-0.39	-0.08	-0.04	-0.07	0.02	-0.01	0.07	0.05	0.09	0.47	0.53	
Marg4	0.58	-0.28	-0.07	-0.09	0.09	-0.06	0.01	-0.11	-0.02	0.03	0.45	0.55	
Marg5	0.59	-0.22	-0.14	-0.10	-0.03	-0.16	0.22	-0.06	-0.02	0.02	0.50	0.50	
Marg6	0.43	-0.33	0.01	0.38	-0.02	0.10	-0.07	0.12	-0.14	-0.06	0.50	0.50	
Marg7	0.29	-0.49	0.24	0.03	0.07	-0.04	0.10	0.05	0.08	0.08	0.41	0.59	
Strong1	0.03	0.02	0.53	0.22	-0.04	0.01	-0.09	-0.17	-0.13	-0.04	0.40	0.60	
Strong2	0.01	0.12	0.38	0.37	0.06	-0.12	0.11	0.04	0.10	-0.09	0.35	0.65	
Strong3	0.05	0.20	0.27	0.42	0.00	0.15	0.03	-0.03	0.08	0.04	0.32	0.68	
Strong4	0.02	-0.04	0.56	0.25	0.02	-0.03	0.05	-0.01	0.07	0.03	0.39	0.61	
Strong5	0.16	-0.03	0.21	0.41	0.12	0.05	-0.07	0.00	0.03	0.11	0.27	0.73	
Angry1	0.16	0.10	0.39	-0.56	0.08	0.03	0.01	0.09	-0.07	0.00	0.52	0.48	
Angry2	0.04	-0.09	0.57	-0.38	0.04	0.02	-0.03	0.05	-0.06	0.09	0.50	0.50	
Angry3	0.04	-0.19	0.53	-0.42	-0.04	0.02	0.00	-0.03	0.03	-0.08	0.50	0.50	
com													
Obj1	2.7												
Obj2	2.2												
Obj3	2.4												
Obj4	3.0												
Obj5	1.5												
Obj6	2.3												
Obj7	2.0												
Obj8	4.0												
Obj9	3.9												
Obj10	4.0												
Marg1	1.9												
Marg2	2.4												
Marg3	2.1												
Marg4	1.7												
Marg5	2.0												
Marg6	3.6												
Marg7	2.5												
Strong1	1.8												
Strong2	2.9												
Strong3	2.7												
Strong4	1.5												
Strong5	2.4												
Angry1	2.2												

Angry2 1.9
 Angry3 2.3

	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8	PA9	PA10
SS loadings	3.55	3.11	1.84	1.52	0.32	0.25	0.20	0.18	0.15	0.14
Proportion Var	0.14	0.12	0.07	0.06	0.01	0.01	0.01	0.01	0.01	0.01
Cumulative Var	0.14	0.27	0.34	0.40	0.41	0.42	0.43	0.44	0.44	0.45
Proportion Explained	0.32	0.28	0.16	0.14	0.03	0.02	0.02	0.02	0.01	0.01
Cumulative Proportion	0.32	0.59	0.75	0.89	0.92	0.94	0.96	0.97	0.99	1.00

Mean item complexity = 2.5

Test of the hypothesis that 10 factors are sufficient.

The degrees of freedom for the null model are 300 and the objective function was 6.77 with 0 degrees of freedom.
 The degrees of freedom for the model are 95 and the objective function was 0.05

The root mean square of the residuals (RMSR) is 0.01

The df corrected root mean square of the residuals is 0.01

The harmonic number of observations is 259 with the empirical chi square 8.19 with prob < 1
 The total number of observations was 259 with Likelihood Chi Square = 13.01 with prob < 1

Tucker Lewis Index of factoring reliability = 1.193

RMSEA index = 0 and the 90 % confidence intervals are 0 0

BIC = -514.88

Fit based upon off diagonal values = 1

Measures of factor score adequacy

	PA1	PA2	PA3	PA4	PA5
Correlation of (regression) scores with factors	0.95	0.93	0.88	0.86	0.61
Multiple R square of scores with factors	0.89	0.87	0.77	0.73	0.37
Minimum correlation of possible factor scores	0.79	0.74	0.54	0.46	-0.26
	PA6	PA7	PA8	PA9	PA10
Correlation of (regression) scores with factors	0.56	0.52	0.50	0.45	0.49
Multiple R square of scores with factors	0.32	0.27	0.25	0.20	0.24
Minimum correlation of possible factor scores	-0.36	-0.45	-0.51	-0.60	-0.52

The total variance for a particular variable will have two factors: some variance will be shared with other variables (common variance) and some variance will be specific to that measure (unique variance). Random variance is also specific to one item, but not reliably so.

We can examine this most easily by examining the matrix (second screen).

The columns PA1 thru PA10 are the (uninteresting at this point) unrotated loadings. These are the loading from each factor to each variable. PA stands for “principal axis.”

Scrolling to the far right we are interested in:

Communalities are represented as h^2 . These are the proportions of common variance present in the variables. A variable that has no specific (or random) variance would have a communality of 1.0. If a variable shares none of its variance with any other variable, its communality would be 0.0.

As a point of comparison, in PCA these started as 1.0 because we extracted the same number of components as items. In PAF, because we must extract fewer factors than items, these will have some value.

**Uniquenesses* are represented as u^2 . These are the amount of unique variance for each variable. They are calculated as $1 - h^2$ (or 1 minus the communality).

The final column, *com* represents *item complexity*. This is an indication of how well an item reflects a single construct. If it is 1.0 then the item loads only on one component, if it is 2.0, it loads evenly on two components, and so forth. For now, we can ignore this. *I mostly wanted to reassure you that “com” is not “communality” – h^2 is communality.*

Let's switch to the first screen of output.

Eigenvalues are displayed in the row called, *SS loadings* (i.e., the sum of squared loadings). They represent the variance explained by the particular linear component. PA1 explains 3.51 units of variance (out of a possible 25; the # of potential factors). As a proportion, this is $3.51/25 = 0.1404$ (reported in the *Proportion Var* row).

```
3.51/25
```

```
[1] 0.1404
```

Note.

We look at the eigenvalues to see how many are > 1.0 (Kaiser's eigenvalue > 1 criteria criteria). We see there are 4 that meet Kaiser's criteria and 4 that meet Joliffe's criteria (eigenvalues $> .77$).

Cumulative Var is helpful to determine how many factors we'd like to retain to balance parsimony (few as possible) with the amount of variance we want to explain. The eigenvalues are in descending order. Using both Kaiser's (eigenvalue > 1.0) and Joiliffe's (eigenvalue > 0.7) criteria, we landed on a four-factor solution. Extracting four factors (like we did with PCA will) will explain 40% of the variance. Eigenvalues are only one criteria, let's look at the scree plot.

Scree plot:

Eigenvalues are stored in the *grmsPAF1* object's variable, "values". We can see all the values captured by this object with the *names()* function:

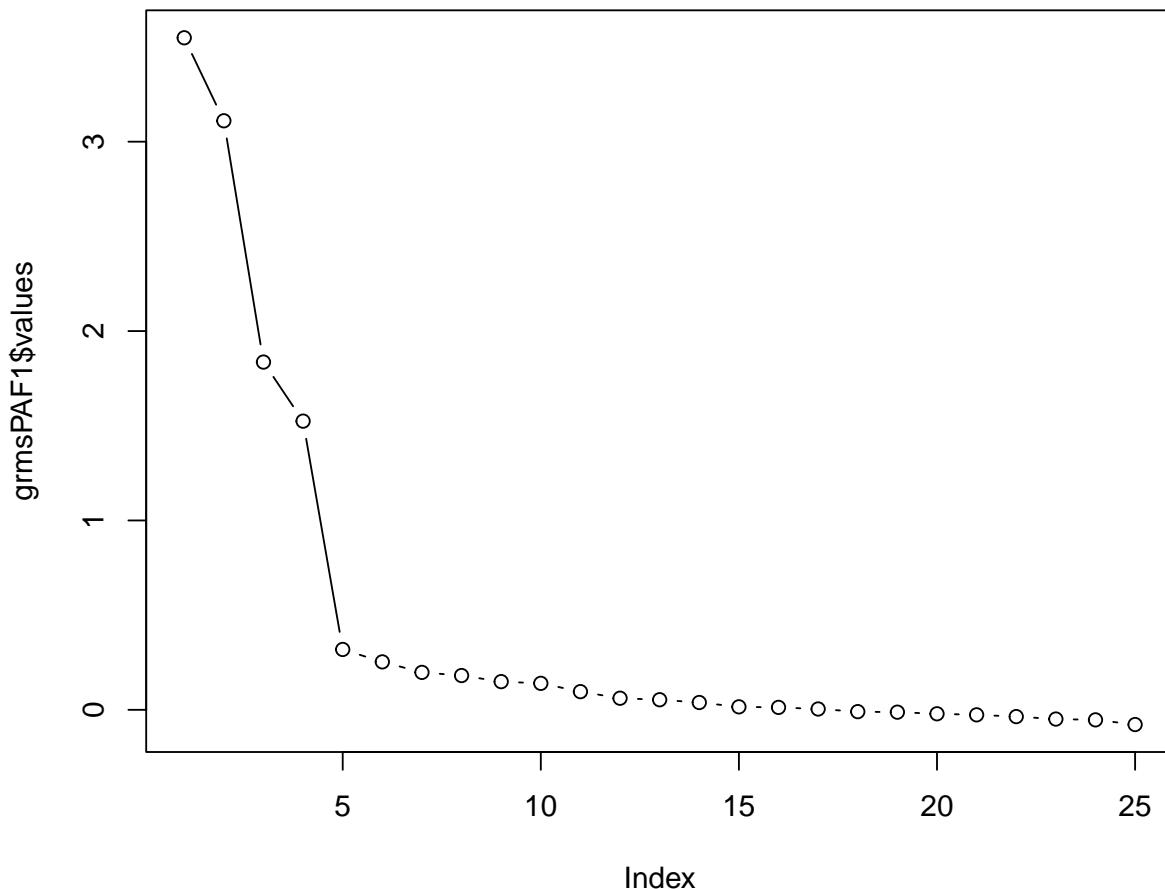
```
names(grmsPAF1)
```

[1] "residual"	"dof"	"chi"
[4] "nh"	"rms"	"EPVAL"
[7] "crms"	"EBIC"	"ESABIC"
[10] "fit"	"fit.off"	"sd"
[13] "factors"	"complexity"	"n.obs"
[16] "objective"	"criteria"	"STATISTIC"
[19] "PVAL"	"Call"	"null.model"
[22] "null.dof"	"null.chisq"	"TLI"
[25] "FO"	"RMSEA"	"BIC"
[28] "SABIC"	"r.scores"	"R2"

```
[31] "valid"                 "score.cor"           "weights"
[34] "rotation"              "communality"         "communalities"
[37] "uniquenesses"          "values"              "e.values"
[40] "loadings"               "model"               "fm"
[43] "Structure"             "communality.iterations" "method"
[46] "scores"                "R2.scores"           "r"
[49] "np.obs"                "fn"                  "Vaccounted"
```

Plotting the eigenvalues produces a scree plot. We can use this to further guage the number of factors we should extract.

```
plot(grmsPAF1$values, type = "b") #type = "b" gives us "both" lines and points; type = "l" gi
```



We look for the point of *inflection*. That is, where the baseline levels out into a plateau. As noted in the PCA lesson, this is one of the most clear scree plots (suggesting 4 factors) I've seen. We are benefitting from having created the simulated data from the factor results.

9.5.2.1 Specifying the number of factors

Having determined the number of components, we must rerun the analysis with this specification. Especially when researchers may not have a clear theoretical structure that guides the process, researchers may do this iteratively with varying numbers of factors. Lewis and Neville (?) examined solutions with 2, 3, 4, and 5 factors (they conducted a parallel *factor* analysis; in contrast this lesson demonstrates principal axis factoring).

```
#grmsPAF2 <- psych::fa(GRMSr, nfactors=4, fm = "pa", rotate="none")
grmsPAF2 <- psych::fa(dfGRMS, nfactors = 4, fm = "pa", rotate = "none") #can copy prior script
grmsPAF2
```

```
Factor Analysis using method = pa
Call: psych::fa(r = dfGRMS, nfactors = 4, rotate = "none", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
      PA1    PA2    PA3    PA4     h2    u2 com
Obj1   0.43   0.47 -0.13 -0.05  0.43  0.57 2.2
Obj2   0.33   0.59 -0.09 -0.14  0.49  0.51 1.7
Obj3   0.35   0.48  0.08 -0.15  0.38  0.62 2.1
Obj4   0.37   0.46 -0.01 -0.03  0.35  0.65 1.9
Obj5   0.19   0.56  0.12  0.03  0.37  0.63 1.3
Obj6   0.46   0.27 -0.03 -0.13  0.30  0.70 1.8
Obj7   0.48   0.31 -0.09  0.09  0.35  0.65 1.9
Obj8   0.26   0.41  0.04  0.09  0.25  0.75 1.8
Obj9   0.29   0.31  0.17  0.05  0.21  0.79 2.6
Obj10  0.30   0.31  0.07  0.17  0.22  0.78 2.7
Marg1   0.76  -0.48 -0.15  0.07  0.84  0.16 1.8
Marg2   0.57  -0.53  0.06 -0.04  0.61  0.39 2.0
Marg3   0.55  -0.37 -0.08 -0.05  0.45  0.55 1.8
Marg4   0.59  -0.26 -0.06 -0.09  0.43  0.57 1.4
Marg5   0.58  -0.19 -0.12 -0.10  0.40  0.60 1.4
Marg6   0.44  -0.32 -0.01  0.37  0.43  0.57 2.8
Marg7   0.30  -0.48  0.24  0.04  0.38  0.62 2.2
Strong1 0.03   0.02  0.51  0.24  0.31  0.69 1.4
Strong2  0.00   0.11  0.36  0.38  0.29  0.71 2.2
Strong3  0.04   0.20  0.25  0.43  0.29  0.71 2.1
Strong4  0.02  -0.04  0.56  0.27  0.39  0.61 1.5
Strong5  0.16  -0.02  0.19  0.41  0.23  0.77 1.8
Angry1   0.16   0.11  0.40 -0.54  0.49  0.51 2.1
Angry2   0.05  -0.09  0.58 -0.35  0.48  0.52 1.7
Angry3   0.05  -0.19  0.54 -0.40  0.50  0.50 2.1
```

	PA1	PA2	PA3	PA4
SS loadings	3.50	3.05	1.80	1.49
Proportion Var	0.14	0.12	0.07	0.06
Cumulative Var	0.14	0.26	0.33	0.39
Proportion Explained	0.36	0.31	0.18	0.15

Cumulative Proportion 0.36 0.67 0.85 1.00

Mean item complexity = 1.9

Test of the hypothesis that 4 factors are sufficient.

The degrees of freedom for the null model are 300 and the objective function was 6.77 with 0 degrees of freedom.
The degrees of freedom for the model are 206 and the objective function was 0.22

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 number of observations is 259 with the empirical chi square 43.59 with prob < 1
The total number of observations was 259 with Likelihood Chi Square = 53.89 with prob < 1

Tucker Lewis Index of factoring reliability = 1.162

RMSEA index = 0 and the 90 % confidence intervals are 0 0

BIC = -1090.82

Fit based upon off diagonal values = 0.99

Measures of factor score adequacy

	PA1	PA2	PA3	PA4
Correlation of (regression) scores with factors	0.94	0.92	0.87	0.84
Multiple R square of scores with factors	0.89	0.85	0.76	0.71
Minimum correlation of possible factor scores	0.78	0.71	0.52	0.42

Our eigenvalues/SS loadings wiggle around a bit from the initial run. With four factors, we now, cumulatively, explain 30% of the variance.

Communality is the proportion of common variance within a variable. Changing from 10 to 4 factors will change this value (h^2) as well as its associated *uniqueness* (u^2), which is calculated as “1.0 minus the communality.”

Now we see that 43% of the variance associated with Obj1 is common/shared (the h^2 value).

As a reminder of what we are doing, recall that we are looking for a more *parsimonious* explanation than 25 items on the GRMS. By respecifying a smaller number of factors, we lose some information. That is, the retained factors (now 4) cannot explain all of the variance present in the data (as we saw, it explains about 39%, cumulatively). The amount of variance explained in each variable is represented by the communalities after extraction.

We can also inspect the communalities through the lens of Kaiser’s criterion (the eigenvalue > 1 criteria) to see if we think that four was a good number of factors to extract.

Kaiser’s criterion is believed to be accurate if:

- when there are fewer than 30 variables (we had 25) and, after extraction, the communalities are greater than .70
 - looking at our data, only 1 communality (Marg1) is $> .70$, so, this does not support extracting four components
- when the sample size is greater than 250 (ours was 259) and the average communality is $> .60$

- we can extract the communalities from our object and calculate the mean the average communality

Using the `names()` function again, we see that “communality” is available for manipulation.

```
names(grmsPAF2)
```

[1] "residual"	"dof"	"chi"
[4] "nh"	"rms"	"EPVAL"
[7] "crms"	"EBIC"	"ESABIC"
[10] "fit"	"fit.off"	"sd"
[13] "factors"	"complexity"	"n.obs"
[16] "objective"	"criteria"	"STATISTIC"
[19] "PVAL"	"Call"	"null.model"
[22] "null.dof"	"null.chisq"	"TLI"
[25] "FO"	"RMSEA"	"BIC"
[28] "SABIC"	"r.scores"	"R2"
[31] "valid"	"score.cor"	"weights"
[34] "rotation"	"communality"	"communalities"
[37] "uniquenesses"	"values"	"e.values"
[40] "loadings"	"model"	"fm"
[43] "Structure"	"communality.iterations"	"method"
[46] "scores"	"R2.scores"	"r"
[49] "np.obs"	"fn"	"Vaccounted"

We can use this value to calculate their mean.

```
mean(grmsPAF2$communality)
```

```
[1] 0.3934452
```

```
#sum(grmsPAF2$communality) #checking my work by calculating the sum and dividing by 25
#9.836131/25
```

We see that our average communality is 0.39. These two criteria suggest that we may not have the best solution. That said (in our defense):

- We used the scree plot as a guide and it was very clear.
- We have an adequate sample size and that was supported with the KMO.
- Are the number of factors consistent with theory? We have not yet inspected the factor loadings. This will provide us with more information.

We could do several things:

- rerun with a different number of factors (recall Lewis and Neville (?) ran models with 2, 3, 4, and 5 factors)

- conduct more diagnostics tests
 - reproduced correlation matrix
 - the difference between the reproduced correlation matrix and the correlation matrix in the data

The `factor.model()` function in `psych` produces the *reproduced correlation matrix* by using the `loadings` in our extracted object. Conceptually, this matrix is the correlations that should be produced if we did not have the raw data but we only had the factor loadings. We could do fancy matrix algebra and produce these.

The questions, though, is: How close did we get? How different is the *reproduced correlation matrix* from `GRMSmatrix` – the *R*-matrix produced from our raw data.

```
round(psych::factor.model(grmsPAF2$loadings), 3) #produces the reproduced correlation matrix
```

	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6	Obj7	Obj8	Obj9	Obj10
Obj1	0.426	0.439	0.372	0.378	0.330	0.335	0.362	0.296	0.243	0.256
Obj2	0.439	0.487	0.411	0.400	0.381	0.333	0.339	0.312	0.253	0.252
Obj3	0.372	0.411	0.380	0.354	0.340	0.309	0.299	0.278	0.253	0.233
Obj4	0.378	0.400	0.354	0.350	0.329	0.299	0.320	0.283	0.245	0.247
Obj5	0.330	0.381	0.340	0.329	0.366	0.234	0.261	0.288	0.248	0.246
Obj6	0.335	0.333	0.309	0.299	0.234	0.303	0.299	0.219	0.205	0.197
Obj7	0.362	0.339	0.299	0.320	0.261	0.299	0.349	0.259	0.227	0.251
Obj8	0.296	0.312	0.278	0.283	0.288	0.219	0.259	0.246	0.213	0.224
Obj9	0.243	0.253	0.253	0.245	0.248	0.205	0.227	0.213	0.208	0.203
Obj10	0.256	0.252	0.233	0.247	0.246	0.197	0.251	0.224	0.203	0.221
Marg1	0.111	-0.035	0.016	0.053	-0.141	0.212	0.236	0.000	0.051	0.074
Marg2	-0.013	-0.126	-0.038	-0.034	-0.180	0.122	0.102	-0.068	0.012	0.000
Marg3	0.071	-0.027	0.018	0.031	-0.114	0.159	0.152	-0.017	0.030	0.032
Marg4	0.143	0.060	0.096	0.101	-0.040	0.215	0.203	0.039	0.078	0.074
Marg5	0.178	0.101	0.119	0.128	-0.014	0.231	0.223	0.059	0.084	0.085
Marg6	0.021	-0.094	-0.051	0.003	-0.084	0.068	0.148	0.018	0.049	0.094
Marg7	-0.132	-0.214	-0.109	-0.115	-0.182	-0.005	-0.021	-0.104	-0.017	-0.036
Strong1	-0.057	-0.058	0.025	0.009	0.084	-0.028	-0.003	0.059	0.112	0.093
Strong2	-0.011	-0.017	0.027	0.039	0.117	-0.027	0.041	0.096	0.116	0.129
Strong3	0.057	0.047	0.064	0.091	0.160	0.010	0.100	0.141	0.137	0.167
Strong4	-0.099	-0.107	-0.008	-0.024	0.055	-0.055	-0.027	0.037	0.102	0.082
Strong5	0.013	-0.035	0.000	0.034	0.052	0.009	0.093	0.078	0.094	0.127
Angry1	0.088	0.151	0.219	0.120	0.123	0.157	0.023	0.053	0.116	0.015
Angry2	-0.085	-0.044	0.072	-0.019	0.017	0.022	-0.091	-0.032	0.063	-0.034
Angry3	-0.122	-0.090	0.032	-0.061	-0.043	0.006	-0.121	-0.077	0.025	-0.075
	Marg1	Marg2	Marg3	Marg4	Marg5	Marg6	Marg7	Strong1	Strong2	
Obj1	0.111	-0.013	0.071	0.143	0.178	0.021	-0.132	-0.057	-0.011	
Obj2	-0.035	-0.126	-0.027	0.060	0.101	-0.094	-0.214	-0.058	-0.017	
Obj3	0.016	-0.038	0.018	0.096	0.119	-0.051	-0.109	0.025	0.027	
Obj4	0.053	-0.034	0.031	0.101	0.128	0.003	-0.115	0.009	0.039	
Obj5	-0.141	-0.180	-0.114	-0.040	-0.014	-0.084	-0.182	0.084	0.117	

Obj6	0.212	0.122	0.159	0.215	0.231	0.068	-0.005	-0.028	-0.027
Obj7	0.236	0.102	0.152	0.203	0.223	0.148	-0.021	-0.003	0.041
Obj8	0.000	-0.068	-0.017	0.039	0.059	0.018	-0.104	0.059	0.096
Obj9	0.051	0.012	0.030	0.078	0.084	0.049	-0.017	0.112	0.116
Obj10	0.074	0.000	0.032	0.074	0.085	0.094	-0.036	0.093	0.129
Marg1	0.835	0.674	0.604	0.574	0.544	0.511	0.427	-0.051	-0.079
Marg2	0.674	0.609	0.506	0.472	0.429	0.399	0.437	0.023	-0.051
Marg3	0.604	0.506	0.448	0.428	0.404	0.341	0.322	-0.046	-0.086
Marg4	0.574	0.472	0.428	0.426	0.409	0.306	0.281	-0.044	-0.083
Marg5	0.544	0.429	0.404	0.409	0.399	0.279	0.233	-0.076	-0.101
Marg6	0.511	0.399	0.341	0.306	0.279	0.427	0.297	0.088	0.103
Marg7	0.427	0.437	0.322	0.281	0.233	0.297	0.378	0.127	0.049
Strong1	-0.051	0.023	-0.046	-0.044	-0.076	0.088	0.127	0.314	0.275
Strong2	-0.079	-0.051	-0.086	-0.083	-0.101	0.103	0.049	0.275	0.286
Strong3	-0.070	-0.082	-0.090	-0.080	-0.087	0.113	-0.003	0.235	0.276
Strong4	-0.031	0.055	-0.032	-0.038	-0.078	0.118	0.169	0.349	0.302
Strong5	0.135	0.098	0.064	0.052	0.034	0.230	0.123	0.198	0.223
Angry1	-0.033	0.081	0.038	0.088	0.073	-0.168	0.068	0.084	-0.045
Angry2	-0.036	0.124	0.027	0.045	0.006	-0.086	0.179	0.211	0.068
Angry3	0.017	0.177	0.071	0.079	0.036	-0.073	0.215	0.177	0.023
	Strong3	Strong4	Strong5	Angry1	Angry2	Angry3			
Obj1	0.057	-0.099	0.013	0.088	-0.085	-0.122			
Obj2	0.047	-0.107	-0.035	0.151	-0.044	-0.090			
Obj3	0.064	-0.008	0.000	0.219	0.072	0.032			
Obj4	0.091	-0.024	0.034	0.120	-0.019	-0.061			
Obj5	0.160	0.055	0.052	0.123	0.017	-0.043			
Obj6	0.010	-0.055	0.009	0.157	0.022	0.006			
Obj7	0.100	-0.027	0.093	0.023	-0.091	-0.121			
Obj8	0.141	0.037	0.078	0.053	-0.032	-0.077			
Obj9	0.137	0.102	0.094	0.116	0.063	0.025			
Obj10	0.167	0.082	0.127	0.015	-0.034	-0.075			
Marg1	-0.070	-0.031	0.135	-0.033	-0.036	0.017			
Marg2	-0.082	0.055	0.098	0.081	0.124	0.177			
Marg3	-0.090	-0.032	0.064	0.038	0.027	0.071			
Marg4	-0.080	-0.038	0.052	0.088	0.045	0.079			
Marg5	-0.087	-0.078	0.034	0.073	0.006	0.036			
Marg6	0.113	0.118	0.230	-0.168	-0.086	-0.073			
Marg7	-0.003	0.169	0.123	0.068	0.179	0.215			
Strong1	0.235	0.349	0.198	0.084	0.211	0.177			
Strong2	0.276	0.302	0.223	-0.045	0.068	0.023			
Strong3	0.288	0.253	0.228	-0.102	-0.019	-0.070			
Strong4	0.253	0.392	0.224	0.078	0.235	0.203			
Strong5	0.228	0.224	0.234	-0.125	-0.026	-0.052			
Angry1	-0.102	0.078	-0.125	0.493	0.425	0.427			
Angry2	-0.019	0.235	-0.026	0.425	0.475	0.479			
Angry3	-0.070	0.203	-0.052	0.427	0.479	0.496			

We're not really interested in this matrix. We just need it to compare it to the *GRMSmatrix* to

produce the residuals. We do that next.

Residuals are the difference between the reproduced (i.e., those created from our factor loadings) and R -matrix produced by the raw data.

If we look at the $r_{Obj1Obj2}$ in our original correlation matrix (theoretically from the raw data [although we simulated data]), the value is 0.41. The reproduced correlation for this pair is 0.426. The difference is -0.016. Our table shows -0.013 because of rounding error.

```
.41 - .426
```

```
[1] -0.016
```

By using the `factor.residuals()` function we can calculate the residuals. Here we will see this difference calculated for us, for all the elements in the matrix.

```
round(psych::factor.residuals(GRMSr, grmsPAF2$loadings), 3)
```

	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6	Obj7	Obj8	Obj9	Obj10
Obj1	0.574	-0.031	0.010	-0.049	0.008	0.000	0.004	0.053	0.011	-0.016
Obj2	-0.031	0.513	0.019	0.001	0.016	0.007	0.013	-0.019	0.024	-0.015
Obj3	0.010	0.019	0.620	0.003	0.001	0.003	0.001	0.002	-0.037	0.003
Obj4	-0.049	0.001	0.003	0.650	-0.007	0.022	0.004	-0.029	0.023	0.029
Obj5	0.008	0.016	0.001	-0.007	0.634	-0.006	-0.005	-0.026	0.018	0.001
Obj6	0.000	0.007	0.003	0.022	-0.006	0.697	0.006	0.019	-0.011	-0.033
Obj7	0.004	0.013	0.001	0.004	-0.005	0.006	0.651	-0.004	-0.034	0.009
Obj8	0.053	-0.019	0.002	-0.029	-0.026	0.019	-0.004	0.754	-0.005	0.000
Obj9	0.011	0.024	-0.037	0.023	0.018	-0.011	-0.034	-0.005	0.792	0.017
Obj10	-0.016	-0.015	0.003	0.029	0.001	-0.033	0.009	0.000	0.017	0.779
Marg1	-0.006	0.010	-0.004	0.005	-0.003	-0.010	-0.005	-0.001	0.014	-0.010
Marg2	0.012	-0.001	0.004	0.006	-0.004	0.034	0.004	-0.017	-0.002	-0.028
Marg3	-0.017	0.014	0.000	0.012	0.005	0.002	-0.029	-0.010	0.036	-0.007
Marg4	0.023	-0.011	0.008	-0.030	0.008	-0.005	0.011	0.014	-0.032	-0.008
Marg5	0.018	-0.029	-0.011	-0.002	0.016	-0.023	0.002	-0.014	0.020	0.040
Marg6	-0.019	0.006	-0.011	0.023	-0.023	-0.005	0.010	0.008	0.002	0.039
Marg7	-0.012	-0.008	0.019	-0.016	0.010	0.007	0.009	0.000	-0.023	-0.003
Strong1	0.011	-0.009	0.005	-0.031	0.025	-0.026	0.008	-0.024	0.031	0.006
Strong2	0.017	-0.042	-0.015	0.002	0.011	0.015	-0.001	0.034	0.004	0.001
Strong3	-0.015	0.007	0.006	0.051	-0.015	0.011	0.005	-0.037	-0.008	-0.025
Strong4	-0.007	0.002	0.016	-0.015	0.008	0.014	-0.021	0.008	0.005	0.005
Strong5	0.009	0.032	-0.005	-0.023	-0.014	-0.007	0.009	0.033	-0.040	-0.017
Angry1	0.022	-0.002	-0.018	0.029	-0.016	-0.003	-0.012	0.022	-0.034	0.010
Angry2	-0.030	0.008	0.005	-0.004	-0.006	-0.016	0.023	0.025	0.003	-0.011
Angry3	-0.002	-0.002	0.001	0.000	-0.008	0.012	0.000	-0.031	0.024	0.008
	Marg1	Marg2	Marg3	Marg4	Marg5	Marg6	Marg7	Strong1	Strong2	
Obj1	-0.006	0.012	-0.017	0.023	0.018	-0.019	-0.012	0.011	0.017	
Obj2	0.010	-0.001	0.014	-0.011	-0.029	0.006	-0.008	-0.009	-0.042	
Obj3	-0.004	0.004	0.000	0.008	-0.011	-0.011	0.019	0.005	-0.015	

Obj4	0.005	0.006	0.012	-0.030	-0.002	0.023	-0.016	-0.031	0.002
Obj5	-0.003	-0.004	0.005	0.008	0.016	-0.023	0.010	0.025	0.011
Obj6	-0.010	0.034	0.002	-0.005	-0.023	-0.005	0.007	-0.026	0.015
Obj7	-0.005	0.004	-0.029	0.011	0.002	0.010	0.009	0.008	-0.001
Obj8	-0.001	-0.017	-0.010	0.014	-0.014	0.008	0.000	-0.024	0.034
Obj9	0.014	-0.002	0.036	-0.032	0.020	0.002	-0.023	0.031	0.004
Obj10	-0.010	-0.028	-0.007	-0.008	0.040	0.039	-0.003	0.006	0.001
Marg1	0.165	-0.010	0.011	0.004	0.004	0.002	-0.007	0.010	-0.016
Marg2	-0.010	0.391	-0.003	0.002	-0.020	0.033	-0.016	0.024	-0.010
Marg3	0.011	-0.003	0.552	-0.013	-0.003	0.005	0.016	-0.015	-0.009
Marg4	0.004	0.002	-0.013	0.574	0.035	-0.024	-0.007	0.019	0.001
Marg5	0.004	-0.020	-0.003	0.035	0.601	-0.033	0.026	-0.022	0.026
Marg6	0.002	0.033	0.005	-0.024	-0.033	0.573	-0.005	0.013	-0.013
Marg7	-0.007	-0.016	0.016	-0.007	0.026	-0.005	0.622	-0.036	0.030
Strong1	0.010	0.024	-0.015	0.019	-0.022	0.013	-0.036	0.686	-0.014
Strong2	-0.016	-0.010	-0.009	0.001	0.026	-0.013	0.030	-0.014	0.714
Strong3	0.012	-0.008	0.004	0.001	-0.015	-0.011	-0.011	0.008	-0.007
Strong4	-0.009	-0.003	0.008	0.001	0.013	-0.013	0.021	-0.006	0.013
Strong5	0.010	-0.001	0.002	0.006	-0.023	0.000	0.012	-0.005	-0.009
Angry1	-0.003	-0.018	0.011	-0.009	-0.003	0.012	0.011	0.004	-0.005
Angry2	0.003	-0.012	-0.004	0.015	-0.012	0.006	0.006	0.004	-0.017
Angry3	0.005	0.029	-0.017	-0.010	0.001	-0.013	-0.015	0.003	0.010
	Strong3	Strong4	Strong5	Angry1	Angry2	Angry3			
Obj1	-0.015	-0.007	0.009	0.022	-0.030	-0.002			
Obj2	0.007	0.002	0.032	-0.002	0.008	-0.002			
Obj3	0.006	0.016	-0.005	-0.018	0.005	0.001			
Obj4	0.051	-0.015	-0.023	0.029	-0.004	0.000			
Obj5	-0.015	0.008	-0.014	-0.016	-0.006	-0.008			
Obj6	0.011	0.014	-0.007	-0.003	-0.016	0.012			
Obj7	0.005	-0.021	0.009	-0.012	0.023	0.000			
Obj8	-0.037	0.008	0.033	0.022	0.025	-0.031			
Obj9	-0.008	0.005	-0.040	-0.034	0.003	0.024			
Obj10	-0.025	0.005	-0.017	0.010	-0.011	0.008			
Marg1	0.012	-0.009	0.010	-0.003	0.003	0.005			
Marg2	-0.008	-0.003	-0.001	-0.018	-0.012	0.029			
Marg3	0.004	0.008	0.002	0.011	-0.004	-0.017			
Marg4	0.001	0.001	0.006	-0.009	0.015	-0.010			
Marg5	-0.015	0.013	-0.023	-0.003	-0.012	0.001			
Marg6	-0.011	-0.013	0.000	0.012	0.006	-0.013			
Marg7	-0.011	0.021	0.012	0.011	0.006	-0.015			
Strong1	0.008	-0.006	-0.005	0.004	0.004	0.003			
Strong2	-0.007	0.013	-0.009	-0.005	-0.017	0.010			
Strong3	0.712	-0.004	0.040	0.000	0.000	0.003			
Strong4	-0.004	0.608	-0.007	-0.017	0.000	-0.004			
Strong5	0.040	-0.007	0.766	0.015	0.005	0.000			
Angry1	0.000	-0.017	0.015	0.507	0.018	0.006			
Angry2	0.000	0.000	0.005	0.018	0.525	-0.013			
Angry3	0.003	-0.004	0.000	0.006	-0.013	0.504			

There are several strategies to evaluate this matrix:

- see how large the residuals are, compared to the original correlations
 - the worst possible model would occur if we extracted no factors and would be the size of the original correlations
 - if the correlations were small to start with, we expect small residuals
 - if the correlations were large to start with, the residuals will be relatively larger (this is not terribly problematic)
- comparing residuals requires squaring them first (because residuals can be both positive and negative)
 - the sum of the squared residuals divided by the sum of the squared correlations is an estimate of model fit. Subtracting this from 1.0 means that it ranges from 0 to 1. Values $> .95$ are an indication of good fit.

Analyzing the residuals means we need to extract only the upper right of the triangle them into an object. We can do this in steps.

```
grmsPAF2_resids <- psych::factor.residuals(GRMSr, grmsPAF2$loadings)#first extract the resids
grmsPAF2_resids <- as.matrix(grmsPAF2_resids[upper.tri(grmsPAF2_resids)])#the object has the r
head(grmsPAF2_resids)
```

```
[,1]
[1,] -0.0309870563
[2,]  0.0100666134
[3,]  0.0185760343
[4,] -0.0490479537
[5,]  0.0008521008
[6,]  0.0030123717
```

One criteria of residual analysis is to see how many residuals there are that are greater than an absolute value of 0.05. The result will be a single column with TRUE if it is $> |0.05|$ and false if it is smaller. The sum function will tell us how many TRUE responses are in the matrix. Further, we can write script to obtain the proportion of total number of residuals.

```
large.resid <- abs(grmsPAF2_resids) > 0.05
#large.resid
sum(large.resid)
```

```
[1] 2
```

```
round(sum(large.resid) / nrow(grmsPAF2_resids),3)
```

```
[1] 0.007
```

We learn that there are 2 residuals greater than the absolute value of 0.05. This represents less than 1% of the total number of residuals.

There are no hard rules about what proportion of residuals can be greater than 0.05. Field recommends that it stay below 50% (?).

Another approach to analyzing residuals is to look at their mean. Because of the +/- valences, we need to square them (to eliminate the negative), take the average, then take the square root.

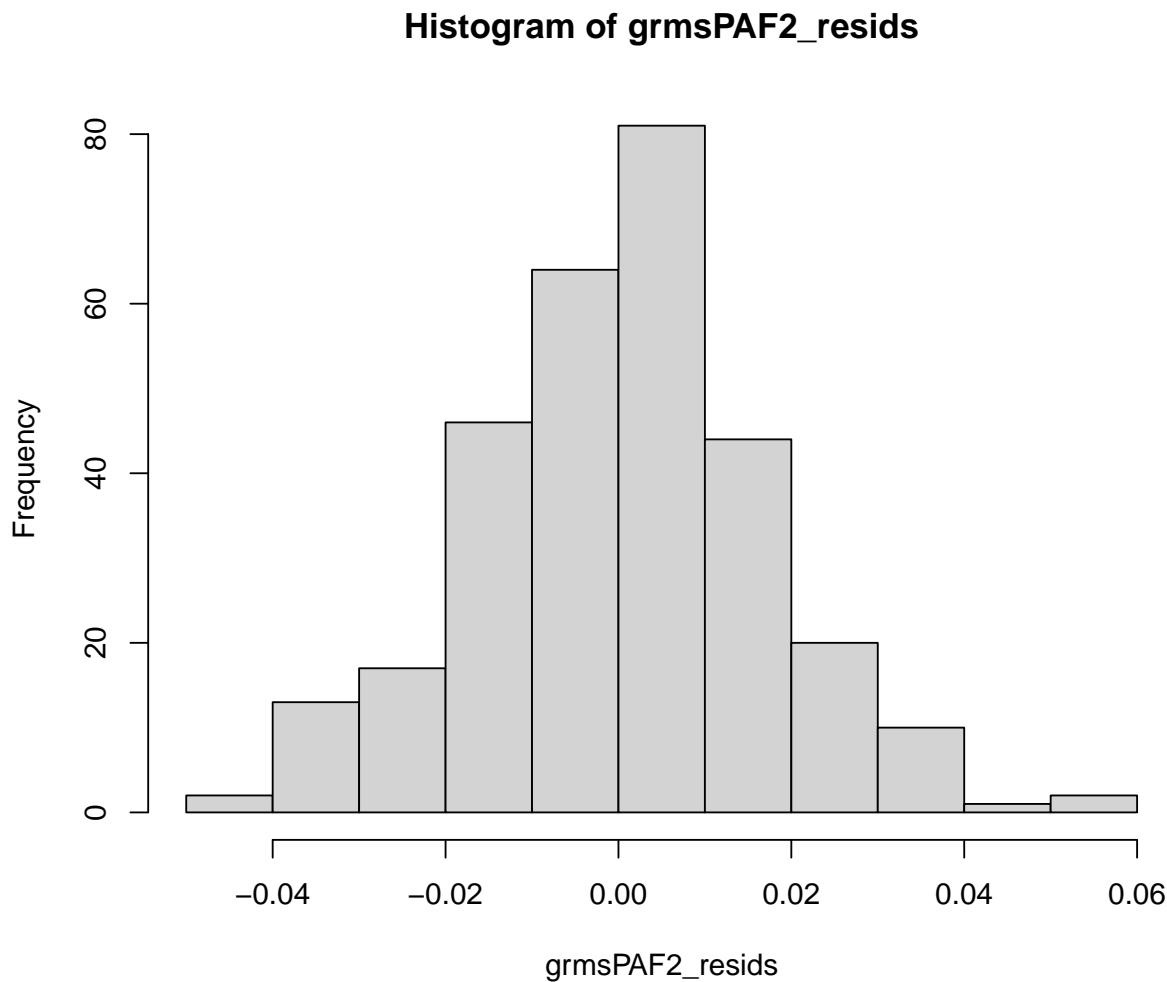
```
round(sqrt(mean(grmsPAF2_resids^2)), 3)
```

```
[1] 0.017
```

While there are no clear guidelines to interpret these, one recommendation is to consider extracting more components if the value is higher than 0.08 (?).

Finally, we expect our residuals to be normally distributed. A histogram can help us inspect the distribution.

```
hist(grmsPAF2_resids)
```



Not bad! It looks reasonably normal. No outliers.

9.5.2.2 Quick recap of how to evaluate the # of factors we extracted

- If fewer than 30 variables, the eigenvalue > 1 (Kaiser's) criteria is fine, so long as communalities are all $> .70$.
- If sample size > 250 and the average communalities are $.6$ or greater, this is fine.
- When $N > 200$, the scree plot can be used.
- Regarding residuals
 - fewer than 50% should have absolute values > 0.05
 - model fit should be > 0.90

9.5.3 Factor Rotation

The original solution of a principal components or principal axis factor analysis is a set of vectors that best account for the observed covariance or correlation matrix. Each additional component or

factor accounts for progressively less and less variance. The solution is efficient (yay) but difficult to interpret (boo).

Thanks to Thurstone's five rules toward a simple structure (circa 1947), interpretation of a matrix is facilitated by *rotation* (multiplying a matrix by a matrix of orthogonal vectors that preserve the communalities of each variable). Both the original matrix and the solution will be orthogonal.

Parsimony becomes a statistical consideration (an equation, in fact) and goal and is maximized when each variable has a 1.0 loading on one factor and the rest are zero.

Different rotation strategies emphasize different goals related to parsimony:

Quartimax seeks to maximize the notion of variable parsimony (each variable is associated with one factor) and permits the rotation toward a general factor (ignoring smaller factors). *Varimax* maximizes the variance of squared loadings taken over items instead of over factors and *avoids* a general factor.

Rotation improves the interpretation of the factor by maximizing the loading on each variable on one of the extracted factors while minimizing the loading on all other factors. Rotation works by changing the absolute values of the variables while keeping their differential values constant.

There are two big choices (to be made on theoretical grounds):

- Orthogonal rotation if you think that the factors are independent/unrelated.
 - varimax is the most common orthogonal rotation
- Oblique rotation if you think that the factors are related/correlated.
 - oblimin and promax are common oblique rotations

9.5.3.1 Orthogonal rotation

```
#grmsPAF2ORTH <- psych::fa(GRMSr, nfactors = 4, fm = "pa", rotate = "varimax")
grmsPAF2ORTH <- psych::fa(dfGRMS, nfactors = 4, fm = "pa", rotate = "varimax")
grmsPAF2ORTH
```

```
Factor Analysis using method = pa
Call: psych::fa(r = dfGRMS, nfactors = 4, rotate = "varimax", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
    PA1   PA2   PA3   PA4   h2   u2 com
Obj1    0.07  0.64 -0.08 -0.08  0.43  0.57  1.1
Obj2   -0.09  0.68  0.00 -0.12  0.49  0.51  1.1
Obj3   -0.01  0.60  0.14 -0.02  0.38  0.62  1.1
Obj4    0.02  0.59  0.00  0.01  0.35  0.65  1.0
Obj5   -0.19  0.56  0.04  0.13  0.37  0.63  1.3
Obj6    0.20  0.50  0.07 -0.08  0.30  0.70  1.4
Obj7    0.21  0.53 -0.13  0.05  0.35  0.65  1.5
Obj8   -0.03  0.48 -0.05  0.12  0.25  0.75  1.2
Obj9    0.04  0.41  0.08  0.18  0.21  0.79  1.5
```

Obj10	0.05	0.41	-0.07	0.20	0.22	0.78	1.6
Marg1	0.91	0.06	-0.08	-0.03	0.84	0.16	1.0
Marg2	0.76	-0.08	0.14	0.02	0.61	0.39	1.1
Marg3	0.66	0.03	0.03	-0.07	0.45	0.55	1.0
Marg4	0.63	0.15	0.06	-0.09	0.43	0.57	1.2
Marg5	0.58	0.20	0.02	-0.13	0.40	0.60	1.3
Marg6	0.55	-0.02	-0.21	0.28	0.43	0.57	1.8
Marg7	0.51	-0.21	0.19	0.19	0.38	0.62	2.0
Strong1	-0.02	0.01	0.21	0.52	0.31	0.69	1.3
Strong2	-0.07	0.06	0.00	0.53	0.29	0.71	1.1
Strong3	-0.08	0.15	-0.11	0.50	0.29	0.71	1.4
Strong4	0.01	-0.05	0.23	0.58	0.39	0.61	1.3
Strong5	0.15	0.05	-0.13	0.44	0.23	0.77	1.4
Angry1	0.01	0.21	0.66	-0.11	0.49	0.51	1.3
Angry2	0.03	-0.03	0.68	0.13	0.48	0.52	1.1
Angry3	0.09	-0.11	0.69	0.06	0.50	0.50	1.1

	PA1	PA2	PA3	PA4
SS loadings	3.33	3.20	1.67	1.64
Proportion Var	0.13	0.13	0.07	0.07
Cumulative Var	0.13	0.26	0.33	0.39
Proportion Explained	0.34	0.33	0.17	0.17
Cumulative Proportion	0.34	0.66	0.83	1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 factors are sufficient.

The degrees of freedom for the null model are 300 and the objective function was 6.77 with 0
The degrees of freedom for the model are 206 and the objective function was 0.22

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 number of observations is 259 with the empirical chi square 43.59 with prob < 1
The total number of observations was 259 with Likelihood Chi Square = 53.89 with prob < 1

Tucker Lewis Index of factoring reliability = 1.162

RMSEA index = 0 and the 90 % confidence intervals are 0 0

BIC = -1090.82

Fit based upon off diagonal values = 0.99

Measures of factor score adequacy

	PA1	PA2	PA3	PA4
Correlation of (regression) scores with factors	0.95	0.91	0.87	0.84
Multiple R square of scores with factors	0.91	0.84	0.76	0.71
Minimum correlation of possible factor scores	0.81	0.67	0.52	0.42

Essentially, we have the same information as before, except that loadings are calculated after rotation (which adjusts the absolute values of the factor loadings while keeping their differential

vales constant). Our communality and uniqueness values remain the same. The eigenvalues (SS loadings) should even out, but the proportion of variance explained and cumulative variance (39%) will remain the same.

The `print.psych()` function facilitates interpretation and prioritizes the information about which we care most:

- “cut” will display loadings above .3, this allows us to see
 - if some items load on no factors
 - if some items have cross-loadings (and their relative weights)
- “sort” will reorder the loadings to make it clearer (to the best of its ability...in the case of ties) to which factor/scale it belongs

```
grmsPAF2_table <- psych::print.psych(grmsPAF20RTH, cut = 0.3, sort = TRUE)
```

```
Factor Analysis using method = pa
Call: psych::fa(r = dfGRMS, nfactors = 4, rotate = "varimax", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
    item   PA1   PA2   PA3   PA4   h2   u2 com
Marg1     11  0.91           0.84 0.16 1.0
Marg2     12  0.76           0.61 0.39 1.1
Marg3     13  0.66           0.45 0.55 1.0
Marg4     14  0.63           0.43 0.57 1.2
Marg5     15  0.58           0.40 0.60 1.3
Marg6     16  0.55           0.43 0.57 1.8
Marg7     17  0.51           0.38 0.62 2.0
Obj2      2   0.68           0.49 0.51 1.1
Obj1      1   0.64           0.43 0.57 1.1
Obj3      3   0.60           0.38 0.62 1.1
Obj4      4   0.59           0.35 0.65 1.0
Obj5      5   0.56           0.37 0.63 1.3
Obj7      7   0.53           0.35 0.65 1.5
Obj6      6   0.50           0.30 0.70 1.4
Obj8      8   0.48           0.25 0.75 1.2
Obj10     10  0.41           0.22 0.78 1.6
Obj9      9   0.41           0.21 0.79 1.5
Angry3    25   0.69           0.50 0.50 1.1
Angry2    24   0.68           0.48 0.52 1.1
Angry1    23   0.66           0.49 0.51 1.3
Strong4   21           0.58 0.39 0.61 1.3
Strong2   19           0.53 0.29 0.71 1.1
Strong1   18           0.52 0.31 0.69 1.3
Strong3   20           0.50 0.29 0.71 1.4
Strong5   22           0.44 0.23 0.77 1.4
```

PA1 PA2 PA3 PA4

SS loadings	3.33	3.20	1.67	1.64
Proportion Var	0.13	0.13	0.07	0.07
Cumulative Var	0.13	0.26	0.33	0.39
Proportion Explained	0.34	0.33	0.17	0.17
Cumulative Proportion	0.34	0.66	0.83	1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 factors are sufficient.

The degrees of freedom for the null model are 300 and the objective function was 6.77 with 0 degrees of freedom for the model are 206 and the objective function was 0.22

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 number of observations is 259 with the empirical chi square 43.59 with prob < 1. The total number of observations was 259 with Likelihood Chi Square = 53.89 with prob < 1

Tucker Lewis Index of factoring reliability = 1.162

RMSEA index = 0 and the 90 % confidence intervals are 0 0

BIC = -1090.82

Fit based upon off diagonal values = 0.99

Measures of factor score adequacy

	PA1	PA2	PA3	PA4
Correlation of (regression) scores with factors	0.95	0.91	0.87	0.84
Multiple R square of scores with factors	0.91	0.84	0.76	0.71
Minimum correlation of possible factor scores	0.81	0.67	0.52	0.42

In the unrotated solution, most variables loaded on the first component. After rotation, there are four clear components/scales. Further, there is clear (or at least reasonable) component/scale membership for each item and no cross-loadings. This is a bit different than the PCA orthogonal rotation where the Marg7 item had a cross-loading.

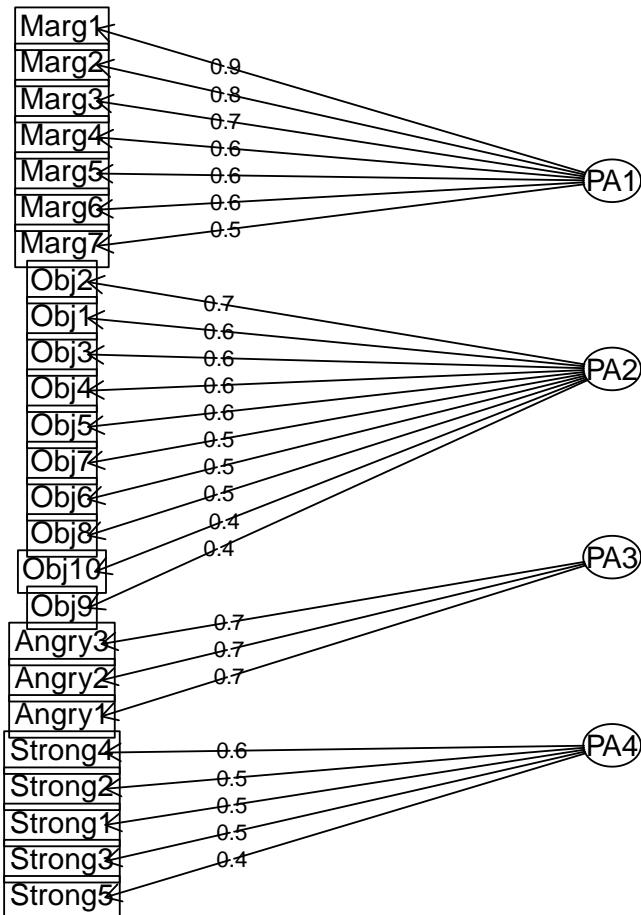
If this were a new scale and we had not yet established ideas for subscales, the next step is to look back at the items, themselves, and try to name the scales/components. If our scale construction included a priori/planned subscales, here's where we hope the items fall where they were hypothesized to do so. Our simulated data worked perfectly and replicated the four scales that Lewis and Neville (?) reported in the article.

- Assumptions of Beauty and Sexual Objectification
- Silenced and Marginalized
- Strong Woman Stereotype
- Angry Woman Stereotype

We can also create a figure of the result. Note the direction of the arrows from the factor (latent variable) to the items in PAF – in PCA the arrows went from item to component.

```
psych::fa.diagram(grmsPAF2ORTH)
```

Factor Analysis



We can extract the factor loadings and write them to a table. This can be useful in preparing an APA style table for a manuscript or presentation.

```
#names(grmsPAF2ORTH)
pafORTH_table <- round(grmsPAF2ORTH$loadings,3)
write.table(pafORTH_table, file="pafORTH_table.csv", sep=",", col.names=TRUE, row.names=FALSE)
pafORTH_table
```

Loadings:

	PA1	PA2	PA3	PA4
Obj1		0.638		
Obj2		0.681		-0.124
Obj3		0.600	0.142	
Obj4		0.591		

Obj5	-0.186	0.560		0.131
Obj6	0.204	0.501		
Obj7	0.214	0.534	-0.126	
Obj8		0.477		0.124
Obj9		0.410		0.179
Obj10		0.414		0.205
Marg1	0.907			
Marg2	0.762		0.142	
Marg3	0.664			
Marg4	0.626	0.153		
Marg5	0.584	0.200		-0.133
Marg6	0.553		-0.208	0.279
Marg7	0.509	-0.213	0.195	0.189
Strong1		0.213	0.518	
Strong2			0.526	
Strong3		0.149	-0.114	0.496
Strong4			0.232	0.579
Strong5	0.147		-0.129	0.440
Angry1		0.205	0.662	-0.113
Angry2			0.675	0.132
Angry3		-0.106	0.687	

	PA1	PA2	PA3	PA4
SS loadings	3.327	3.203	1.670	1.638
Proportion Var	0.133	0.128	0.067	0.066
Cumulative Var	0.133	0.261	0.328	0.393

9.5.3.2 Oblique rotation

Whereas the orthogonal rotation sought to maximize the independence/unrelatedness of the components, an oblique rotation will allow them to be correlated. Researchers often explore both solutions but only report one.

```
#grmsPAF2obl <- psych::fa(GRMSr, nfactors = 4, fm = "pa", rotate = "oblimin")
grmsPAF2obl <- psych::fa(dfGRMS, nfactors = 4, fm = "pa", rotate = "oblimin")
grmsPAF2obl
```

Factor Analysis using method = pa
Call: psych::fa(r = dfGRMS, nfactors = 4, rotate = "oblimin", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix

	PA1	PA2	PA3	PA4	h2	u2	com
Obj1	0.08	0.64	-0.07	-0.06	0.43	0.57	1.1
Obj2	-0.08	0.68	0.01	-0.10	0.49	0.51	1.1
Obj3	-0.01	0.60	0.15	0.00	0.38	0.62	1.1
Obj4	0.02	0.59	0.01	0.03	0.35	0.65	1.0
Obj5	-0.19	0.57	0.04	0.15	0.37	0.63	1.4
Obj6	0.21	0.49	0.08	-0.06	0.30	0.70	1.4

Obj7	0.22	0.53	-0.12	0.07	0.35	0.65	1.5
Obj8	-0.03	0.48	-0.04	0.14	0.25	0.75	1.2
Obj9	0.04	0.41	0.08	0.19	0.21	0.79	1.5
Obj10	0.05	0.41	-0.08	0.22	0.22	0.78	1.6
Marg1	0.91	0.03	-0.08	-0.02	0.84	0.16	1.0
Marg2	0.76	-0.11	0.14	0.02	0.61	0.39	1.1
Marg3	0.67	0.01	0.03	-0.07	0.45	0.55	1.0
Marg4	0.63	0.13	0.07	-0.08	0.43	0.57	1.2
Marg5	0.59	0.18	0.03	-0.12	0.40	0.60	1.3
Marg6	0.55	-0.04	-0.22	0.28	0.43	0.57	1.8
Marg7	0.50	-0.23	0.18	0.18	0.38	0.62	2.0
Strong1	-0.04	0.01	0.19	0.52	0.31	0.69	1.3
Strong2	-0.09	0.06	-0.02	0.53	0.29	0.71	1.1
Strong3	-0.09	0.15	-0.14	0.50	0.29	0.71	1.4
Strong4	-0.01	-0.05	0.20	0.58	0.39	0.61	1.3
Strong5	0.14	0.04	-0.15	0.44	0.23	0.77	1.5
Angry1	-0.01	0.21	0.67	-0.11	0.49	0.51	1.3
Angry2	0.00	-0.03	0.67	0.13	0.48	0.52	1.1
Angry3	0.06	-0.10	0.68	0.06	0.50	0.50	1.1

	PA1	PA2	PA3	PA4
SS loadings	3.33	3.19	1.67	1.65
Proportion Var	0.13	0.13	0.07	0.07
Cumulative Var	0.13	0.26	0.33	0.39
Proportion Explained	0.34	0.32	0.17	0.17
Cumulative Proportion	0.34	0.66	0.83	1.00

With factor correlations of

	PA1	PA2	PA3	PA4
PA1	1.00	0.03	0.03	0.02
PA2	0.03	1.00	-0.02	-0.04
PA3	0.03	-0.02	1.00	0.05
PA4	0.02	-0.04	0.05	1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 factors are sufficient.

The degrees of freedom for the null model are 300 and the objective function was 6.77 with 0
The degrees of freedom for the model are 206 and the objective function was 0.22

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 number of observations is 259 with the empirical chi square 43.59 with prob < 1
The total number of observations was 259 with Likelihood Chi Square = 53.89 with prob < 1

Tucker Lewis Index of factoring reliability = 1.162

RMSEA index = 0 and the 90 % confidence intervals are 0 0

```
BIC = -1090.82
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                         PA1  PA2  PA3  PA4
Correlation of (regression) scores with factors    0.95 0.91 0.87 0.84
Multiple R square of scores with factors          0.91 0.84 0.76 0.71
Minimum correlation of possible factor scores    0.82 0.67 0.52 0.42
```

We can make it a little easier to interpret by removing all factor loadings below .30.

```
psych::print.psych(grmsPAF2obl, cut = 0.3, sort=TRUE)
```

```
Factor Analysis using method = pa
Call: psych::fa(r = dfGRMS, nfactors = 4, rotate = "oblimin", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
```

item	PA1	PA2	PA3	PA4	h2	u2	com
Marg1	11	0.91			0.84	0.16	1.0
Marg2	12	0.76			0.61	0.39	1.1
Marg3	13	0.67			0.45	0.55	1.0
Marg4	14	0.63			0.43	0.57	1.2
Marg5	15	0.59			0.40	0.60	1.3
Marg6	16	0.55			0.43	0.57	1.8
Marg7	17	0.50			0.38	0.62	2.0
Obj2	2		0.68		0.49	0.51	1.1
Obj1	1		0.64		0.43	0.57	1.1
Obj3	3		0.60		0.38	0.62	1.1
Obj4	4		0.59		0.35	0.65	1.0
Obj5	5		0.57		0.37	0.63	1.4
Obj7	7		0.53		0.35	0.65	1.5
Obj6	6		0.49		0.30	0.70	1.4
Obj8	8		0.48		0.25	0.75	1.2
Obj10	10		0.41		0.22	0.78	1.6
Obj9	9		0.41		0.21	0.79	1.5
Angry3	25			0.68	0.50	0.50	1.1
Angry1	23			0.67	0.49	0.51	1.3
Angry2	24			0.67	0.48	0.52	1.1
Strong4	21				0.58	0.39	0.61
Strong2	19				0.53	0.29	0.71
Strong1	18				0.52	0.31	0.69
Strong3	20				0.50	0.29	0.71
Strong5	22				0.44	0.23	0.77

	PA1	PA2	PA3	PA4
SS loadings	3.33	3.19	1.67	1.65
Proportion Var	0.13	0.13	0.07	0.07
Cumulative Var	0.13	0.26	0.33	0.39
Proportion Explained	0.34	0.32	0.17	0.17

Cumulative Proportion 0.34 0.66 0.83 1.00

With factor correlations of

	PA1	PA2	PA3	PA4
PA1	1.00	0.03	0.03	0.02
PA2	0.03	1.00	-0.02	-0.04
PA3	0.03	-0.02	1.00	0.05
PA4	0.02	-0.04	0.05	1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 factors are sufficient.

The degrees of freedom for the null model are 300 and the objective function was 6.77 with 0 degrees of freedom.
The degrees of freedom for the model are 206 and the objective function was 0.22

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 number of observations is 259 with the empirical chi square 43.59 with prob < 1
The total number of observations was 259 with Likelihood Chi Square = 53.89 with prob < 1

Tucker Lewis Index of factoring reliability = 1.162

RMSEA index = 0 and the 90 % confidence intervals are 0 0

BIC = -1090.82

Fit based upon off diagonal values = 0.99

Measures of factor score adequacy

	PA1	PA2	PA3	PA4
Correlation of (regression) scores with factors	0.95	0.91	0.87	0.84
Multiple R square of scores with factors	0.91	0.84	0.76	0.71
Minimum correlation of possible factor scores	0.82	0.67	0.52	0.42

All of the items stayed in their respective factors. Note, though, that because our specification included "sort=TRUE", the relative weights wiggled around and so the items are listed in a different order than in the orthogonal rotation.

The oblique rotation allows us to see the correlation between the factors/scales. This was not available in the orthogonal rotation because the assumption of the orthogonal/varimax rotation is that the scales/factors are uncorrelated; hence in the analysis they were fixed to 0.0.

We can see that all the scales have almost no relation with each other. That is, the correlations range between -0.04 to 0.05. This is unusual and likely a biproduct of simulating data. It does, though, support the orthogonal rotation as the preferred one.

Of course there is always a little complexity. In oblique rotations, there is a distinction between the *pattern* matrix (which reports factor loadings and is comparable to the matrix we interpreted for the orthogonal rotation) and the *structure* matrix (takes into account the relationship between the factors/scales – it is a product of the pattern matrix and the matrix containing the correlation coefficients between the factors/scales). Most interpret the pattern matrix because it is simpler; however it could be that values in the pattern matrix are suppressed because of relations between the factors. Therefore, the structure matrix can be a useful check and some editors will request it.

Obtaining the structure matrix requires two steps. First, multiply the factor loadings with the phi matrix.

```
grmsPAF2obl$loadings %*% grmsPAF2obl$Phi
```

	PA1	PA2	PA3	PA4
Obj1	0.091617245	0.64179460	-0.082116054	-0.086955997
Obj2	-0.064381577	0.68533828	-0.010177709	-0.126827177
Obj3	0.009666359	0.59740193	0.137430240	-0.013192598
Obj4	0.038521534	0.59022701	-0.003149310	0.008964478
Obj5	-0.167648230	0.55513250	0.026889947	0.128051314
Obj6	0.220900534	0.50057516	0.069632821	-0.073189898
Obj7	0.231920793	0.53246553	-0.121211510	0.045773488
Obj8	-0.016732470	0.47326894	-0.049300632	0.120380779
Obj9	0.054345805	0.40181963	0.077792112	0.183004446
Obj10	0.066416194	0.40685467	-0.071121938	0.201990945
Marg1	0.909316145	0.05658834	-0.052531028	-0.009345473
Marg2	0.758728791	-0.09131277	0.169966028	0.045330248
Marg3	0.665014268	0.02942008	0.049650256	-0.055406392
Marg4	0.630496623	0.15006307	0.084401291	-0.068984727
Marg5	0.590748368	0.19980732	0.037862201	-0.117798546
Marg6	0.551643326	-0.02771386	-0.186799079	0.280811565
Marg7	0.500021599	-0.22688015	0.214615010	0.210736738
Strong1	-0.023734203	-0.01446513	0.214332979	0.527155113
Strong2	-0.074308426	0.04206983	0.002679449	0.523406328
Strong3	-0.079576059	0.13356854	-0.115836132	0.487340804
Strong4	0.003977093	-0.07474787	0.234704732	0.590024683
Strong5	0.147194163	0.03193109	-0.122335743	0.436688544
Angry1	0.013344627	0.19735056	0.659875942	-0.079698748
Angry2	0.026436508	-0.05154707	0.676344447	0.166509373
Angry3	0.085638675	-0.12109116	0.691072295	0.100726154

Then use Field's (?) function to produce the matrix.

```
#Field's function to produce the structure matrix
factor.structure <- function(fa, cut = 0.2, decimals = 2){
  structure.matrix <- psych::fa.sort(fa$loadings %*% fa$Phi)
  structure.matrix <- data.frame(ifelse(abs(structure.matrix) < cut, "", round(structure.matrix, decimals)))
  return(structure.matrix)
}

factor.structure(grmsPAF2obl, cut = 0.3)
```

	PA1	PA2	PA3	PA4
Marg1	0.91			
Marg2	0.76			

Marg3	0.67
Marg4	0.63
Marg5	0.59
Marg6	0.55
Marg7	0.5
Obj2	0.69
Obj1	0.64
Obj3	0.6
Obj4	0.59
Obj5	0.56
Obj7	0.53
Obj6	0.5
Obj8	0.47
Obj10	0.41
Obj9	0.4
Angry3	0.69
Angry2	0.68
Angry1	0.66
Strong4	0.59
Strong1	0.53
Strong2	0.52
Strong3	0.49
Strong5	0.44

Although some of the relative values changed, our items were stable regarding their component membership.

9.5.4 Factor Scores

Factor *scores* (PA scores) can be created for each case (row) on each component (column). These can be used to assess the relative standing of one person on the construct/variable to another. We can also use them in regression (in place of means or sums) when groups of predictors correlate so highly that there is multicolliearity.

Computation involves multiplying an individual's item-level response by the component loadings we obtained through the PAF process. The results will be one score per component for each row/case.

```
#in all of this, don't forget to be specifying the dataset that has the reverse-coded item rep
grmsPAF2obl <- psych::fa(dfGRMS, nfactors = 4, fm = "pa", rotate = "oblimin", scores = TRUE)
head(grmsPAF2obl$scores, 10) #shows us only the first 10 (of N = 2571)
```

	PA1	PA2	PA3	PA4
[1,]	1.064294757	-0.36271545	-0.05808223	0.646299885
[2,]	0.859678584	-0.36907029	-0.14101755	-1.060489782
[3,]	-0.207333849	-1.11352702	-2.20075890	-1.273124196
[4,]	0.003072714	-0.08097616	1.53024949	-1.097663890
[5,]	0.341575555	0.37185409	-1.68365215	-0.067680029

```
[6,] 1.388246787 0.61847412 -0.18638299 0.429949093
[7,] -1.094742433 -0.84883604 0.19227720 0.974728637
[8,] -0.298883211 -1.12083125 -0.88288628 -0.004756817
[9,] 0.250752016 0.54276969 1.27348563 1.554382437
[10,] -0.202795623 0.35651240 0.26740434 0.263447499
```

```
dfGRMS <- cbind(dfGRMS, grmsPAF2obl$scores) #adds them to our raw dataset
```

To bring this full circle, we can see the correlation of the component scores; the pattern maps onto what we saw previously in the correlations between factors in the oblique rotation.

```
psych::corr.test(dfGRMS [c("PA1", "PA2", "PA3", "PA4")])
```

```
Call:psych::corr.test(x = dfGRMS[c("PA1", "PA2", "PA3", "PA4")])
Correlation matrix
  PA1   PA2   PA3   PA4
PA1 1.00  0.03  0.04  0.02
PA2  0.03  1.00 -0.03 -0.03
PA3  0.04 -0.03  1.00  0.08
PA4  0.02 -0.03  0.08  1.00
Sample Size
[1] 259
Probability values (Entries above the diagonal are adjusted for multiple tests.)
  PA1   PA2   PA3   PA4
PA1 0.00  1.00  1.00    1
PA2 0.59  0.00  1.00    1
PA3 0.56  0.66  0.00    1
PA4 0.71  0.58  0.18    0
```

To see confidence intervals of the correlations, print with the short=FALSE option

We can extract the factor loadings and write them to a table. This can be useful in preparing an APA style table for a manuscript or presentation.

```
#names(grmsPAF2obl)
pafOBL_table <- round(grmsPAF2obl$loadings,3)
write.table(pafOBL_table, file="pafOBL_table.csv", sep=",", col.names=TRUE, row.names=FALSE)
pafOBL_table
```

Loadings:

	PA1	PA2	PA3	PA4
Obj1		0.636		
Obj2		0.684	-0.100	
Obj3		0.601	0.152	
Obj4		0.591		
Obj5	-0.187	0.567		0.151

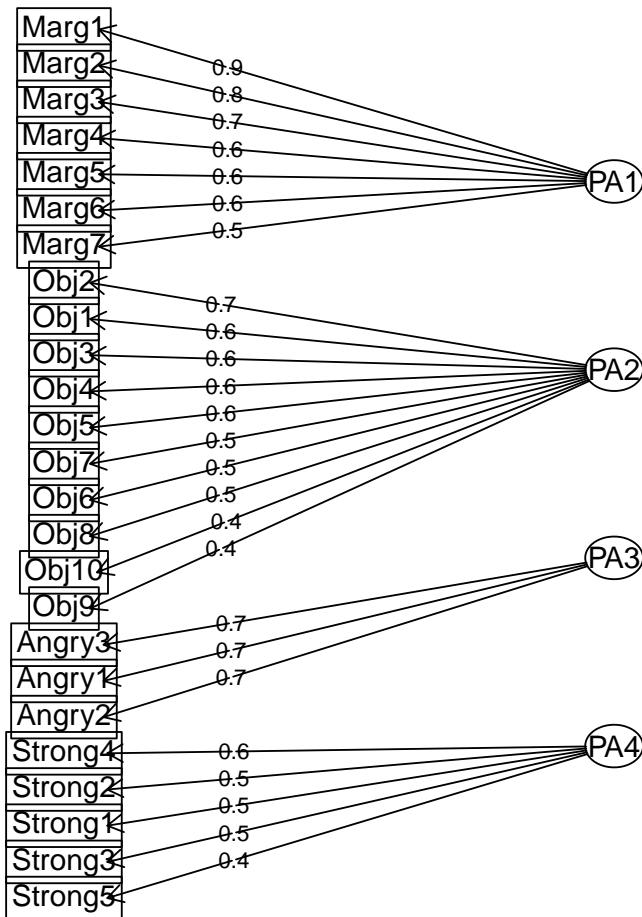
Obj6	0.207	0.495		
Obj7	0.221	0.526	-0.120	
Obj8		0.478		0.142
Obj9		0.410		0.194
Obj10		0.412		0.221
Marg1	0.912			
Marg2	0.757	-0.107	0.142	
Marg3	0.665			
Marg4	0.626	0.132		
Marg5	0.588	0.180		-0.125
Marg6	0.554		-0.221	0.280
Marg7	0.496	-0.229	0.183	0.182
Strong1		0.187	0.518	
Strong2			0.529	
Strong3		0.152	-0.137	0.502
Strong4			0.202	0.577
Strong5	0.142		-0.150	0.444
Angry1		0.210	0.671	-0.108
Angry2			0.668	0.129
Angry3		-0.104	0.683	

	PA1	PA2	PA3	PA4
SS loadings	3.324	3.196	1.660	1.654
Proportion Var	0.133	0.128	0.066	0.066
Cumulative Var	0.133	0.261	0.327	0.393

We can also obtain a figure of this PAF with oblique rotation.

```
psych::fa.diagram(grmsPAF2obl)
```

Factor Analysis



9.6 APA Style Results

Results

The dimensionality of the 25 items from the Gendered Racial Microaggressions Scale for Black Women was analyzed using principal axis factoring. First, data were screened to determine the suitability of the data for this analyses. The Kaiser-Meyer- Olkin measure of sampling adequacy (KMO; Kaiser, 1970) represents the ratio of the squared correlation between variables to the squared partial correlation between variables. KMO ranges from 0.00 to 1.00 – values closer to 1.00 indicate that the patterns of correlations are relatively compact and that component analysis should yield distinct and reliable components (Field, 2012). In our dataset, the KMO value was .86, indicating acceptable sampling adequacy. The Bartlett's Test of Sphericity examines whether the population correlation matrix resembles an identity matrix (Field, 2012). When the p value for the Bartlett's test is $< .05$, we are fairly certain we have clusters of correlated variables. In our dataset, $\chi^2(300) = 1683.76, p < .001$, indicating the correlations between items are sufficiently large enough for principal components analysis. The determinant of the correlation matrix alerts us to any issues

of multicollinearity or singularity and should be larger than 0.00001. Our determinant was 0.00115 and, again, indicated that our data was suitable for the analysis.

Four criteria were used to determine the number of factors to rotate: a priori theory, the scree test, the Eigenvalue-greater-than-one criteria, and the interpretability of the solution. Kaiser's eigenvalue-greater-than-one criteria suggested four components, and, in combination explained 39% of the variance. The scree plot was showed an inflexion that would justified retaining four components. Based on the convergence of these decisions, four components were extracted. We investigated each with orthogonal (varimax) and oblique (oblimin) procedures. Given the non-significant correlations (ranging from -0.04 to 0.05) and the clear component loadings in the orthogonal rotation, we determined that an orthogonal solution was most appropriate.

The rotated solution, as shown in Table 1 and Figure 1, yielded four interpretable components, each listed with the proportion of variance accounted for: assumptions of beauty and sexual objectification (13%), silenced and marginalized (13%), strong woman stereotype (7%), and angry woman stereotype (7%).

Regarding the Table 1, I would include a table with ALL the values, bolding those with component membership. This will be easy because we exported all those values to a .csv file.

9.6.1 Comparing FA and PCA

- FA drives a mathematical solution from which factors are estimated
 - Only FA can estimate underlying factors, but it relies on the various assumptions to be met
- PCA decomposes the original data into a set of linear variates
 - This limits its concern to establishing which linear components exist within the data and how a particular variable might contribute to that component
- Generally, FA and PCA result in similar solutions
 - When there are 30 or more variables and communalities are $> .7$ for all variables, different solutions are unlikely (Stevens, 2002)
 - When there are < 20 variables and low communalities ($< .4$) different solutions are likely to emerge
 - Both are inferential statistics
- Critics of PCA suggest
 - “at best it is a common factor analysis with some error added and at worst an unrecognizable hodgepodge of things from which nothing can be determined” (Cliff, 1987, p. 349)
 - PCA should never be described as FA and the resulting components should not be treated as reverently as true, latent variable, *factors*
 - To most of us (i.e., scientist-practitioners), the difference is largely from the algorithm used to drive the solutions. This is true for Field (?) also, who uses the terms interchangeably. My take: use whichever you like, just be precise in the language describing what you did.

9.7 Going Back to the Future: What, then, is Omega?

Now that we've had an introduction to factor analysis, let's revisit the ω grouping of reliability estimates. In the context of *psychometrics*, it may be useful to think of factors as scales/subscales where g refers to the amount of variance in the *general* factor (or total scale score) and subscales to be items that have something in common that is separate from what is g .

Model-based estimates examine the correlations or covariances of the items and decompose the test variance into that which is:

- common to all items (**g**, a general factor),
- specific to some items (**f**, orthogonal group factors), and
- unique to each item (confounding **s** specific, and **e** error variance)

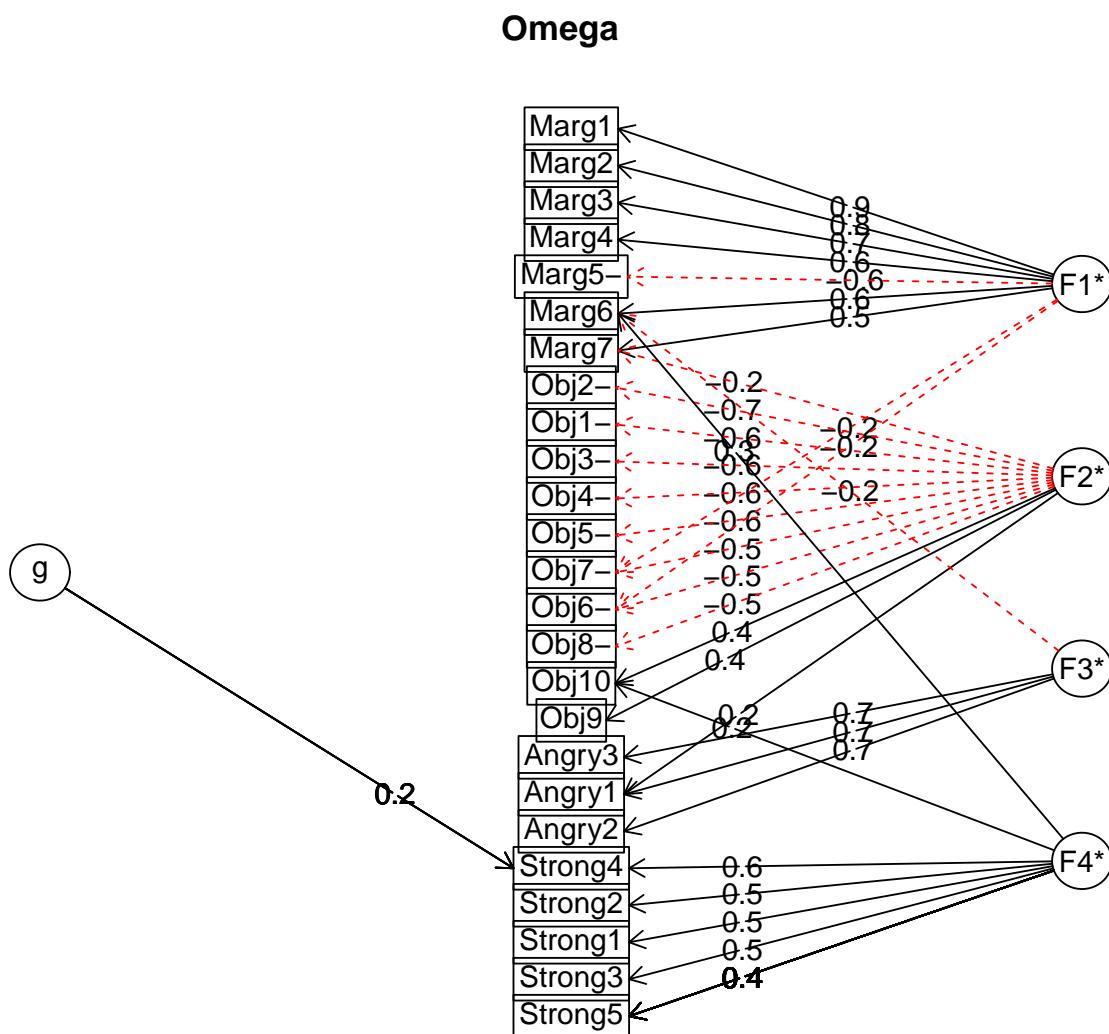
In the *psych* package

- ω_t represents the total reliability of the test (ω_t)
 - In the *psych* package, this is calculated from a bifactor model where there is one general g factor (i.e., each item loads on the single general factor), one or more group factors (f), and an item-specific factor (s).
- ω_h extracts a higher order factor from the correlation matrix of lower level factors, then applies the Schmid and Leiman (1957) transformation to find the general loadings on the original items. Stated another way, it is a measure of the general factor saturation (g ; the amount of variance attributable to one common factor). The subscript “ h ” acknowledges the hierarchical nature of the approach.
 - the ω_h approach is exploratory and defined if there are three or more group factors (with only two group factors, the default is to assume they are equally important, hence the factor loadings of those subscales will be equal)
 - Najera Catalan (?) suggests that ω_h is the best measure of reliability when dealing with multiple dimensions.
- ω_g is an estimate that uses a bifactor solution via the SEM package *lavaan* and tends to be a larger (because it forces all the cross loadings of lower level factors to be 0)
 - the ω_g is confirmatory, requiring the specification of which variables load on each group factor
- *psych::omegaSem()* reports both EFA and CFA solutions
 - We will use the *psych::omegaSem()* function

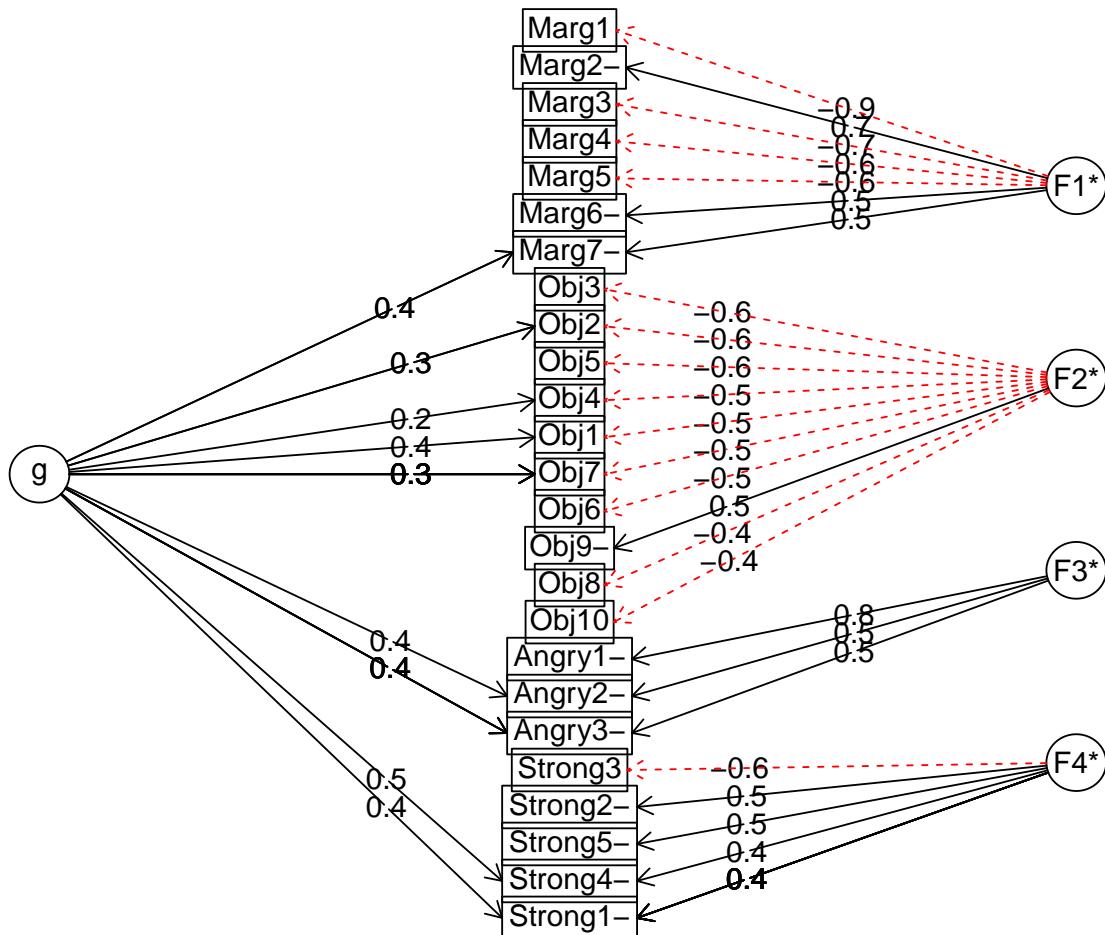
Note that in our specification, we indicate there are two factors. We do not tell it (anywhere!) what items belong to what factors (think, *subscales*). One test will be to see if the items align with their respective factors.

#Because we added the component scores to our df (and now it has more variables than just our observed variables)

```
psych::omegaSem(GRMSr, nfactors = 4, n.obs=259)
```



Omega from SEM



```

Call: psych::omegaSem(m = GRMSr, nfactors = 4, n.obs = 259)
Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
  digits = digits, title = title, sl = sl, labels = labels,
  plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
  covar = covar)

```

Alpha:	0.6
G.6:	0.71
Omega Hierarchical:	0.07
Omega H asymptotic:	0.09
Omega Total	0.74

Schmid Leiman Factor loadings greater than 0.2

	g	F1*	F2*	F3*	F4*	h2	u2	p2
Obj1-		-0.63		0.43	0.57	0.02		
Obj2-		-0.68		0.49	0.51	0.02		

Obj3-		-0.60		0.38	0.62	0.00	
Obj4-		-0.59		0.35	0.65	0.01	
Obj5-		-0.56		0.37	0.63	0.00	
Obj6-		-0.21	-0.49	0.30	0.70	0.00	
Obj7-		-0.22	-0.52	0.35	0.65	0.01	
Obj8-		-0.48		0.25	0.75	0.00	
Obj9		0.41		0.21	0.79	0.00	
Obj10		0.41		0.21	0.22	0.78	0.00
Marg1		0.91		0.84	0.16	0.00	
Marg2		0.75		0.61	0.39	0.02	
Marg3		0.66		0.45	0.55	0.00	
Marg4		0.62		0.43	0.57	0.00	
Marg5-		-0.59		0.40	0.60	0.00	
Marg6		0.55	-0.22	0.27	0.43	0.57	0.01
Marg7		0.49	-0.23		0.38	0.62	0.06
Strong1				0.50	0.31	0.69	0.10
Strong2				0.51	0.29	0.71	0.05
Strong3				0.48	0.29	0.71	0.02
Strong4	0.20		0.20	0.56	0.39	0.61	0.11
Strong5				0.43	0.23	0.77	0.04
Angry1		0.21	0.65		0.49	0.51	0.02
Angry2			0.65		0.48	0.52	0.07
Angry3			0.67		0.50	0.50	0.07

With eigenvalues of:

g	F1*	F2*	F3*	F4*
0.25	3.30	3.16	1.58	1.53

general/max 0.08 max/min = 2.15

mean percent general = 0.03 with sd = 0.03 and cv of 1.31

Explained Common Variance of the general factor = 0.03

The degrees of freedom are 206 and the fit is 0.22

The number of observations was 259 with Chi Square = 53.89 with prob < 1

The root mean square of the residuals is 0.02

The df corrected root mean square of the residuals is 0.02

RMSEA index = 0 and the 10 % confidence intervals are 0 0

BIC = -1090.82

Compare this with the adequacy of just a general factor and no group factors

The degrees of freedom for just the general factor are 275 and the fit is 6.44

The number of observations was 259 with Chi Square = 1598.25 with prob < 0.0000000000000000

The root mean square of the residuals is 0.19

The df corrected root mean square of the residuals is 0.2

RMSEA index = 0.136 and the 10 % confidence intervals are 0.13 0.143

BIC = 70.12

Measures of factor score adequacy

	g	F1*	F2*	F3*	F4*
Correlation of scores with factors		0.31	0.95	0.91	0.85
Multiple R square of scores with factors		0.10	0.90	0.83	0.72
Minimum correlation of factor score estimates	-0.80	0.81	0.66	0.45	0.32

Total, General and Subset omega for each subset

	g	F1*	F2*	F3*	F4*
Omega total for total scores and subscales	0.74	0.76	0.67	0.72	0.65
Omega general for total scores and subscales	0.07	0.01	0.01	0.04	0.05
Omega group for total scores and subscales	0.61	0.75	0.66	0.69	0.60

The following analyses were done using the lavaan package

Omega Hierarchical from a confirmatory model using sem = 0.4

Omega Total from a confirmatory model using sem = 0.73

With loadings of

	g	F1*	F2*	F3*	F4*	h2	u2	p2
Obj1	0.36		-0.54			0.42	0.58	0.31
Obj2	0.33		-0.59			0.46	0.54	0.24
Obj3			-0.60			0.37	0.63	0.04
Obj4	0.24		-0.54			0.35	0.65	0.16
Obj5			-0.55			0.31	0.69	0.03
Obj6			-0.47			0.25	0.75	0.13
Obj7	0.26		-0.47			0.29	0.71	0.23
Obj8			-0.45			0.23	0.77	0.11
Obj9-			0.46			0.21	0.79	0.00
Obj10			-0.41			0.18	0.82	0.04
Marg1		-0.91				0.84	0.16	0.01
Marg2-		0.75				0.59	0.41	0.05
Marg3		-0.67				0.45	0.55	0.00
Marg4		-0.63				0.40	0.60	0.02
Marg5		-0.58				0.37	0.63	0.07
Marg6-		0.54				0.30	0.70	0.00
Marg7-	0.36	0.50				0.38	0.62	0.34
Strong1-	0.35			0.41	0.29	0.71	0.42	
Strong2-				0.47	0.26	0.74	0.14	
Strong3				-0.58	0.33	0.67	0.00	
Strong4-	0.45			0.45	0.41	0.59	0.49	
Strong5-				0.45	0.20	0.80	0.00	
Angry1-			0.77			0.60	0.40	0.01
Angry2-	0.44		0.52			0.47	0.53	0.41
Angry3-	0.45		0.51			0.46	0.54	0.44

With sum of squared loadings of:

g F1* F2* F3* F4*
1.4 3.1 2.6 1.1 1.1

The degrees of freedom of the confirmatory model are 250 and the fit is 250.7259 with p = general/max 0.45 max/min = 2.78
mean percent general = 0.15 with sd = 0.16 and cv of 1.1
Explained Common Variance of the general factor = 0.15

Measures of factor score adequacy

	g	F1*	F2*	F3*	F4*
Correlation of scores with factors	0.78	1.06	0.96	0.91	0.85
Multiple R square of scores with factors	0.60	1.13	0.92	0.82	0.72
Minimum correlation of factor score estimates	0.20	1.26	0.84	0.64	0.44

Total, General and Subset omega for each subset

	g	F1*	F2*	F3*	F4*
Omega total for total scores and subscales	0.73	0.32	0.75	0.74	0.42
Omega general for total scores and subscales	0.40	0.15	0.12	0.17	0.18
Omega group for total scores and subscales	0.40	0.17	0.62	0.57	0.24

To get the standard sem fit statistics, ask for summary on the fitted object

There's a ton of output! How do we make sense of it?

First, our items aligned perfectly with their respective factors (subscales). That is, it would be problematic if the items switched factors.

Second, we can interpret our results. Like alpha, the omegas range from 0 to 1, where values closer to 1 represent good reliability (?). For unidimensional measures, ω_t values above 0.80 seem to be an indicator of good reliability. For multidimensional measures with well-defined dimensions we strive for ω_h values above 0.65 (and $\omega_t > 0.8$). These recommendations are based on a Monte Carlo study that examined a host of reliability indicators and how their values corresponded with accurate predictions of poverty status. With this in mind, let's examine the output related to our simulated research vignette.

Let's examine the output in the lower portion where the values are "from a confirmatory model using sem."

Omega is a reliability estimate for factor analysis that represents the proportion of variance in the GRMS scale attributable to common variance (rather than error). The omega for the total reliability of the test (ω_t ; which included the general factors and the subscale factors) was .73, meaning that 73% of the variance in the total scale is due to the factors and 27% (100% - 73%) is attributable to error.

Omega hierarchical (ω_h) estimates are the proportion of variance in the GRMS score attributable to the general factor, which in effect treats the subscales as error. ω_h for the the GRMS total scale was .40 A quick calculation with ω_h (.40) and ω_t (.73; $.40/.73 = .55$) lets us know that that 55% of the reliable variance in the GRMS total scale is attributable to the general factor.

.4/.73

[1] 0.5479452

Amongst the output is the Cronbach's alpha coefficient (.60). Lewis and Neville (?) did not report omega results. They reported an alpha of .92 for the version of the GRMS that assessed stress appraisal.

9.8 Comparing PFA to Item Analysis and PCA

In the lesson on PCA, we began a table that compared our item analysis (item corrected-total correlations with item-other scale correlations) and PCA results (both orthogonal and oblique). Let's now add our PAF results (both orthogonal and oblique).

In the prior lecture, I saved the file as both .rds and .csv objects. I will bring back in the .rds object and add to it.

```
GRMScomps <- readRDS("GRMS_Comparisons.rds")
grmsPAF2ORTH
```

```
Factor Analysis using method = pa
Call: psych::fa(r = dfGRMS, nfactors = 4, rotate = "varimax", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
    PA1   PA2   PA3   PA4   h2   u2 com
Obj1    0.07  0.64 -0.08 -0.08  0.43  0.57 1.1
Obj2   -0.09  0.68  0.00 -0.12  0.49  0.51 1.1
Obj3   -0.01  0.60  0.14 -0.02  0.38  0.62 1.1
Obj4    0.02  0.59  0.00  0.01  0.35  0.65 1.0
Obj5   -0.19  0.56  0.04  0.13  0.37  0.63 1.3
Obj6    0.20  0.50  0.07 -0.08  0.30  0.70 1.4
Obj7    0.21  0.53 -0.13  0.05  0.35  0.65 1.5
Obj8   -0.03  0.48 -0.05  0.12  0.25  0.75 1.2
Obj9    0.04  0.41  0.08  0.18  0.21  0.79 1.5
Obj10   0.05  0.41 -0.07  0.20  0.22  0.78 1.6
Marg1   0.91  0.06 -0.08 -0.03  0.84  0.16 1.0
Marg2   0.76 -0.08  0.14  0.02  0.61  0.39 1.1
Marg3   0.66  0.03  0.03 -0.07  0.45  0.55 1.0
Marg4   0.63  0.15  0.06 -0.09  0.43  0.57 1.2
Marg5   0.58  0.20  0.02 -0.13  0.40  0.60 1.3
Marg6   0.55 -0.02 -0.21  0.28  0.43  0.57 1.8
Marg7   0.51 -0.21  0.19  0.19  0.38  0.62 2.0
Strong1 -0.02  0.01  0.21  0.52  0.31  0.69 1.3
Strong2 -0.07  0.06  0.00  0.53  0.29  0.71 1.1
Strong3 -0.08  0.15 -0.11  0.50  0.29  0.71 1.4
Strong4  0.01 -0.05  0.23  0.58  0.39  0.61 1.3
Strong5  0.15  0.05 -0.13  0.44  0.23  0.77 1.4
Angry1   0.01  0.21  0.66 -0.11  0.49  0.51 1.3
Angry2   0.03 -0.03  0.68  0.13  0.48  0.52 1.1
Angry3   0.09 -0.11  0.69  0.06  0.50  0.50 1.1
```

	PA1	PA2	PA3	PA4
Obj1	0.07	0.64	-0.08	-0.08
Obj2	-0.09	0.68	0.00	-0.12
Obj3	-0.01	0.60	0.14	-0.02
Obj4	0.02	0.59	0.00	0.01
Obj5	-0.19	0.56	0.04	0.13
Obj6	0.20	0.50	0.07	-0.08
Obj7	0.21	0.53	-0.13	0.05
Obj8	-0.03	0.48	-0.05	0.12
Obj9	0.04	0.41	0.08	0.18
Obj10	0.05	0.41	-0.07	0.20
Marg1	0.91	0.06	-0.08	-0.03
Marg2	0.76	-0.08	0.14	0.02
Marg3	0.66	0.03	0.03	-0.07
Marg4	0.63	0.15	0.06	-0.09
Marg5	0.58	0.20	0.02	-0.13
Marg6	0.55	-0.02	-0.21	0.28
Marg7	0.51	-0.21	0.19	0.19
Strong1	-0.02	0.01	0.21	0.52
Strong2	-0.07	0.06	0.00	0.53
Strong3	-0.08	0.15	-0.11	0.50
Strong4	0.01	-0.05	0.23	0.58
Strong5	0.15	0.05	-0.13	0.44
Angry1	0.01	0.21	0.66	-0.11
Angry2	0.03	-0.03	0.68	0.13
Angry3	0.09	-0.11	0.69	0.06

SS loadings	3.33 3.20 1.67 1.64
Proportion Var	0.13 0.13 0.07 0.07
Cumulative Var	0.13 0.26 0.33 0.39
Proportion Explained	0.34 0.33 0.17 0.17
Cumulative Proportion	0.34 0.66 0.83 1.00

Mean item complexity = 1.3

Test of the hypothesis that 4 factors are sufficient.

The degrees of freedom for the null model are 300 and the objective function was 6.77 with 0.0000000
The degrees of freedom for the model are 206 and the objective function was 0.22

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 number of observations is 259 with the empirical chi square 43.59 with prob < 1.0000000
The total number of observations was 259 with Likelihood Chi Square = 53.89 with prob < 1.0000000

Tucker Lewis Index of factoring reliability = 1.162

RMSEA index = 0 and the 90 % confidence intervals are 0 0

BIC = -1090.82

Fit based upon off diagonal values = 0.99

Measures of factor score adequacy

	PA1	PA2	PA3	PA4
Correlation of (regression) scores with factors	0.95	0.91	0.87	0.84
Multiple R square of scores with factors	0.91	0.84	0.76	0.71
Minimum correlation of possible factor scores	0.81	0.67	0.52	0.42

```
#names(grmsPAF2ORTH)
pafORTH_loadings <- data.frame(unclass(grmsPAF2ORTH$loadings))#I had to add "unclass" to the loadings
pafORTH_loadings$Items <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9")
pafORTH_loadings <- dplyr::rename (pafORTH_loadings, PAF_OR_Mar = PA1, PAF_OR_Obj = PA2, PAF_OB_Mar = PA3, PAF_OB_Obj = PA4)
GRMScomps <- dplyr::full_join(GRMScomps, pafORTH_loadings, by = "Items")#I had to add "unclass" to the loadings
#Now adding the PAF oblique loadings
pafOBLQ_loadings <- data.frame(unclass(grmsPAF2obl$loadings))#I had to add "unclass" to the loadings
pafOBLQ_loadings$Items <- c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5", "Obj6", "Obj7", "Obj8", "Obj9")
pafOBLQ_loadings <- dplyr::rename (pafOBLQ_loadings, PAF_OB_Mar = PA1, PAF_OB_Obj = PA2, PAF_OB_Mar = PA3, PAF_OB_Obj = PA4)
GRMScomps <- dplyr::full_join(GRMScomps, pafOBLQ_loadings, by = "Items")#I had to add "unclass" to the loadings
write.csv(GRMScomps, file = "GRMS_Comps.csv", sep = ",", row.names=FALSE, col.names=TRUE)#Writing to CSV
```

Warning in write.csv(GRMScomps, file = "GRMS_Comps.csv", sep = ",", row.names = FALSE, : attempt to set 'col.names' ignored

Warning in write.csv(GRMScomps, file = "GRMS_Comps.csv", sep = ",", row.names = FALSE, : attempt to set 'sep' ignored

As a research vignette, this has worked extremely well, modeling consistency across the item analysis, principal components analysis (PCA), and principal axis factoring (PAF). That is, items load highest on their own scale (whether it is a component or factor), have no cross-loadings, and do not switch scale memberships from analysis to analysis.

Items	Total Scale r.drop	Item Corrected-Total (bold) and Item-Other-Scale Correlations				Principal Components Analysis: Orthogonal Rotation			Principal Components Analysis: Oblique Rotation			Principal Axis Factoring: Orthogonal Rotation			Principal Axis Factoring: Oblique Rotation						
		Objectified	Marginalized	Strong	Angry	Objectified	Marginalized	Strong	Angry	Objectified	Marginalized	Strong	Angry	Marginalized	Objectified	Angry	Strong	Marginalized	Objectified	Angry	Strong
Obj1	0.34	0.56	0.08	-0.03	-0.06	0.69	0.08	-0.10	-0.09	0.68	0.08	-0.07	-0.08	0.07	0.64	-0.08	-0.08	0.08	0.64	-0.07	-0.06
Obj2	0.28	0.58	-0.07	-0.06	0.01	0.72	-0.10	-0.14	0.00	0.72	-0.09	-0.11	0.02	-0.09	0.68	0.00	-0.12	-0.08	0.68	0.01	-0.10
Obj3	0.36	0.53	0.01	0.03	0.13	0.65	-0.01	-0.04	0.17	0.66	-0.02	-0.01	0.19	-0.01	0.60	0.14	-0.02	-0.01	0.60	0.15	0.00
Obj4	0.35	0.53	0.04	0.03	0.03	0.65	0.02	0.00	0.00	0.65	0.02	0.03	0.01	0.02	0.59	0.00	0.01	0.02	0.59	0.01	0.03
Obj5	0.25	0.48	-0.15	0.15	0.03	0.61	-0.21	0.14	0.05	0.62	-0.21	0.17	0.06	-0.19	0.56	0.04	0.13	-0.19	0.57	0.04	0.15
Obj6	0.39	0.44	0.20	-0.03	0.07	0.56	0.23	-0.11	0.08	0.55	0.23	-0.09	0.10	0.20	0.50	0.07	-0.08	0.21	0.49	0.08	-0.06
Obj7	0.40	0.48	0.21	0.06	-0.07	0.58	0.24	0.05	-0.16	0.58	0.25	0.07	-0.15	0.21	0.53	-0.13	0.05	0.22	0.53	-0.12	0.07
Obj8	0.28	0.43	-0.02	0.13	-0.02	0.54	-0.04	0.15	-0.06	0.54	-0.04	0.17	-0.06	-0.03	0.49	-0.05	0.12	-0.03	0.48	-0.04	0.14
Obj9	0.34	0.38	0.05	0.17	0.08	0.47	0.04	0.22	0.11	0.47	0.04	0.23	0.11	0.04	0.41	0.08	0.15	0.04	0.41	0.08	0.19
Obj10	0.31	0.37	0.08	0.18	-0.04	0.47	0.06	0.25	-0.10	0.46	0.06	0.27	-0.10	0.05	0.41	-0.07	0.20	0.05	0.41	-0.08	0.22
Marg1	0.46	0.10	0.81	-0.03	-0.02	0.06	0.89	-0.02	-0.08	0.03	0.89	-0.03	-0.08	0.91	0.06	-0.08	-0.03	0.91	0.03	-0.08	-0.02
Marg2	0.38	-0.03	0.68	0.02	0.16	-0.09	0.80	0.02	0.14	-0.11	0.79	0.01	0.14	0.76	-0.08	0.14	0.02	0.76	-0.11	0.14	0.02
Marg3	0.34	0.06	0.60	-0.06	0.05	0.03	0.73	-0.08	0.02	0.01	0.73	-0.08	0.03	0.66	0.03	0.03	-0.07	0.67	0.01	0.03	-0.07
Marg4	0.41	0.16	0.56	-0.05	0.09	0.16	0.69	-0.10	0.07	0.14	0.69	-0.10	0.07	0.63	0.15	0.06	-0.09	0.63	0.13	0.07	-0.08
Marg5	0.38	0.20	0.52	-0.10	0.04	0.21	0.65	-0.15	0.01	0.20	0.65	-0.15	0.02	0.58	0.20	0.02	-0.13	0.59	0.18	0.03	-0.12
Marg6	0.28	0.03	0.47	0.19	-0.13	-0.03	0.61	0.32	-0.25	-0.05	0.61	0.31	-0.26	0.55	-0.02	-0.21	0.28	0.55	-0.04	-0.22	0.28
Marg7	0.23	-0.16	0.45	0.16	0.20	-0.24	0.57	0.22	0.23	-0.26	0.56	0.20	0.21	0.51	-0.21	0.19	0.19	0.50	-0.23	0.18	0.18
Strong1	0.18	0.04	0.00	0.40	0.20	0.00	-0.03	0.61	0.27	0.00	-0.04	0.61	0.24	-0.02	0.01	0.21	0.52	-0.04	0.01	0.19	0.52
Strong2	0.13	0.09	-0.04	0.40	0.02	0.06	-0.09	0.64	0.00	0.06	-0.10	0.64	-0.02	-0.07	0.06	0.00	0.53	-0.09	0.06	-0.02	0.53
Strong3	0.13	0.15	-0.06	0.38	-0.08	0.16	-0.10	0.60	-0.15	0.15	-0.10	0.61	-0.17	-0.08	0.15	-0.11	0.50	-0.09	0.15	-0.14	0.50
Strong4	0.17	-0.01	0.04	0.43	0.21	-0.06	0.01	0.66	0.28	-0.06	-0.01	0.65	0.25	0.01	-0.05	0.23	0.58	-0.01	-0.05	0.20	0.58
Strong5	0.19	0.07	0.15	0.33	-0.08	0.04	0.17	0.56	-0.18	0.03	0.17	0.56	-0.20	0.15	0.05	-0.13	0.44	0.14	0.04	-0.15	0.44
Angry1	0.22	0.18	0.03	-0.03	0.51	0.22	0.01	-0.13	0.75	0.23	-0.01	-0.12	0.76	0.01	0.21	0.66	-0.11	-0.01	0.21	0.67	-0.11
Angry2	0.16	-0.02	0.05	0.16	0.54	-0.03	0.03	0.13	0.77	-0.03	0.01	0.13	0.77	0.03	-0.03	0.68	0.13	0.00	-0.03	0.67	0.13
Angry3	0.13	-0.09	0.09	0.11	0.53	-0.11	0.10	0.06	0.77	-0.11	0.08	0.06	0.77	0.09	-0.11	0.69	0.06	0.06	-0.10	0.68	0.06

Figure 9.3: Comparison of path models for PCA and EFA

9.9 Practice Problems

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. In the *ReCentering Psych Stats: Psychometrics* OER, it would be ideal if you have selected a dataset you can utilize across the lessons. The least complex is to change the random seed in the research and rework the problem demonstrated in the lesson. The most complex is to use data of your own. In either case, please plan to:

- Properly format and prepare the data.
- Conduct diagnostic tests to determine the suitability of the data for PCA.
- Conducting tests to guide the decisions about number of components to extract.
- Conducting orthogonal and oblique extractions (at least two each with different numbers of components).
- Selecting one solution and preparing an APA style results section (with table and figure).
- Compare your results in light of any other psychometrics lessons where you have used this data (especially the **item analysis** and **PCA** lessons).

9.9.1 Problem #1: Play around with this simulation.

Copy the script for the simulation and then change (at least) one thing in the simulation to see how it impacts the results. If PAF is new to you, perhaps you just change the number in “set.seed(210921)” from 210921 to something else. Your results should parallel those obtained in the lecture, making it easier for you to check your work as you go.

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Conduct and interpret the three diagnostic tests to determine if PCA is appropriate as an analysis (KMO, Bartlett's, determinant).	5	_____
3. Determine how many components to extract (e.g., scree plot, eigenvalues, theory).	5	_____
4. Conduct an orthogonal extraction and rotation.	5	_____
5. Conduct an oblique extraction and rotation.	5	_____
6. Repeat the orthogonal and oblique extractions/rotations with a different number of specified factors.	5	_____
7. APA style results section with table and figure of one of the solutions.	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

9.9.2 Problem #2: Conduct a PCA with the Szymanski and Bissonette (?) research vignette that was used in prior lessons.

The second option involves utilizing one of the simulated datasets available in this OER. Szymanski and Bissonette's (?)Perceptions of the LGBTQ College Campus Climate Scale: Development and psychometric evaluation was used as the research vignette for the validity, reliability, and item analysis lessons. Although I switched vignettes, the Szymanski and Bissonette example is ready for PAF

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Conduct and interpret the three diagnostic tests to determine if PCA is appropriate as an analysis (KMO, Bartlett's, determinant).	5	_____
3. Determine how many components to extract (e.g., scree plot, eigenvalues, theory).	5	_____
4. Conduct an orthogonal extraction and rotation.	5	_____
5. Conduct an oblique extraction and rotation.	5	_____
6. Repeat the orthogonal and oblique extractions/rotations with a different number of specified factors.	5	_____
7. APA style results section with table and figure of one of the solutions.	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

9.9.3 Problem #3: Try something entirely new.

Using data for which you have permission and access (e.g., IRB approved data you have collected or from your lab; data you simulate from a published article; data from an open science repository; data from other chapters in this OER), complete PAF. The data should allow for at least two (ideally three) components/subscales.

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Check and, if needed, format data	5	_____
2. Conduct and interpret the three diagnostic tests to determine if PCA is appropriate as an analysis (KMO, Bartlett's, determinant).	5	_____
3. Determine how many components to extract (e.g., scree plot, eigenvalues, theory).	5	_____
4. Conduct an orthogonal extraction and rotation.	5	_____
5. Conduct an oblique extraction and rotation.	5	_____
6. Repeat the orthogonal and oblique extractions/rotations with a different number of specified factors.	5	_____
7. APA style results section with table and figure of one of the solutions.	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

Confirmatory Factor Analysis

Chapter 10

CFA: First Order Models

[Screencasted Lecture Link](#)

This is the first in our series on confirmatory factor analysis (CFA).

Our goal is:

- Comparison of CFA and EFA/PCA
- Identify issues in specifying models
- First order models – specification and running
 - unidimensional
 - multidimensional
- Interpreting output
- Comparing two versions (unidimensional, multidimensional) of a first-order model

10.1 Navigating this Lesson

This lesson is just over two hours. I would add another two hours to work through and digest the materials.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

10.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Compare and contrast EFA and CFA
- Identify the components of item-level variance in CFA
- Specify CFA measurement models

- Interpret fit indices (e.g., Chi-square, CFI, RMSEA)
- Interpret statistics used to compare two CFA models

10.1.2 Planning for Practice

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. The least complex is to change the random seed and rework the problem demonstrated in the lesson. The results *should* map onto the ones obtained in the lecture.

The second option comes from the “the back of the book” where a **chapter** contains simulated data for all of the examples worked in this volume. Any of these is available for CFA.

As a third option, you are welcome to use data to which you have access and is suitable for CFA. These could include other simulated data, data available through open science repositories, or your own data (presuming you have permission to use it).

The suggestion for practice spans this chapter and the **next**. From this assignment, you should plan to:

- Prepare the data frame for CFA.
- Specify and run unidimensional and single order (with correlated factors) models.
 - In the next chapter, you will add the specification, evaluation, and write-up of second-order and bifactor models.
- Narrate the adequacy of fit with χ^2 , CFI, RMSEA, SRMR
 - Write a mini-results section for each
- Compare model fit with $\chi^2\Delta$, AIC, and BIC.
- Write an APA style results sections with table(s) and figures.

10.1.3 Readings & Resources

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (when possible, linked) in the text with complete citations in the reference list.

Byrne, B. M. (2016). Structural equation modeling with AMOS: Basic concepts, applications, and programming (3rd ed.). Routledge. <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4556523>

- Chapter 1, Structural Equation Modeling: The basics
- Chapter 3, Application 1: Testing the Factorial Validity of a Theoretical Construct (First-Order CFA Model)
- Chapter 4, Application 2: Testing the Factorial Validity of a Measurement Scale (First-Order CFA Model)

Dekay, Nicole (2021). Quick Reference Guide: The statistics for psychometrics <https://www.humananalysts.com/quick-reference-guide-the-statistics-for-psychometrics>

Kline, R. (2016). Principles and practice of structural equation modeling (Fourth ed., Methodology in the social sciences). New York: The Guilford Press.

- Chapter 9: Specification and Identification of Confirmatory Factor Analysis Models
- Chapter 13: Analysis of Confirmatory Factor Analysis Models
- Chapter 12: Global Fit Testing

Rosseel, Y. (2019). The *lavaan* tutorial. Belgium: Department of Data Analysis, Ghent University. <http://lavaan.ugent.be/tutorial/tutorial.pdf>

- “The model syntax” pp. 3 - 4
- “A first example: confirmatory factor analysis (CFA)” pp. 4-8.

10.1.4 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(lavaan)){install.packages("lavaan")}
#if(!require(lavaanPlot)){install.packages("lavaanPlot")}
#if(!require(psych)){install.packages("psych")}
#if(!require(semTable)){install.packages("semTable")}
```

10.2 Two Broad Categories of Factor Analysis: Exploratory and Confirmatory

Kline (?) described confirmatory factor analysis as “exactly half that of SEM – the other half comes from regression analysis” (p. 189).

10.2.1 Common to Both Exploratory and Confirmatory Approaches

In both exploratory and confirmatory approaches, the variance of each indicator/item is divided into **common** and **unique** variance. When we assume that variance is 1.0, the common variance becomes the communality. If we have 8 items, we will have 8 communalities and this represents the common variance explained by the factors or components.

- **Common variance** is shared among the indicators and serves as a basis for observed covariances among them that depart, meaningfully, from zero. We generally assume that
 - common variance is due to the factors, and
 - there will be fewer factors than the number of indicators/items (after all, there is no point in retaining as many factors [explanatory entities] as there are entities to be explained [indicators/items])

- the proportion of total variance that is shared is the **communality** (estimated by h^2); if $h^2 = .70$, then 70% of the total indicator variance is common and potentially explained by the factors
- **Unique variance** consists of
 - **specific variance**: systematic variance that is not explained by any factor in the model
 - **random measurement error**
 - **method variance** is not pictured, but could be another source of unique variance
- In factor analysis, summing the communalities represents the total common variance (a portion of the total variance), but not the total variance.
 - Factor analysis, then, aligns well with classic test theory and classic approaches to understanding reliability (observed score = true score + error).
 - The inclusion of error is illustrated well in the classic illustrations of CFA and SEM where each item/indicator includes common variance (from the factor) and error variance.

Recall that principal components analysis (PCA is not factor analysis) one of the key distinctions is that all variance is common variance (there is no unique variance). Total common variance is equal to the total variance explained, which in turn is equal to the total variance.

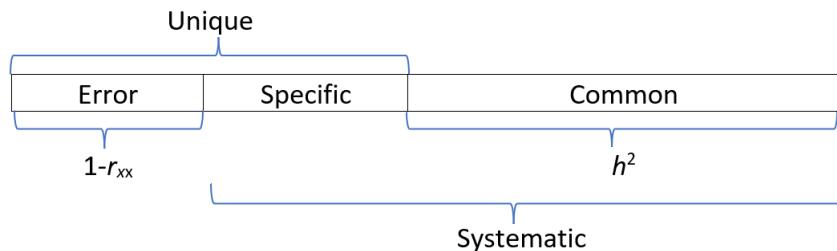


Figure 10.1: Figure illustrating the unique and common variance associated with a factor

10.2.2 Differences between EFA and CFA

- **A priori specification of the number of factors**
 - EFA requires no a priori specification; prior to extraction an EFA program will extract as many factors as indicators. Typically, in subsequent analyses, the researchers specifies how many factors to extract.
 - CFA requires researchers to specify the exact number of factors.
- **The degree of “exact correspondence” between indicators/items and factors/scales**
 - EFA is an **unrestricted measurement model** That is, indicators/items depend on (theoretically, measure) all factors. The direct effects from factors to indicators are *pattern coefficients*. Kline (?) says that most refer to these as *factor loadings* or just *loadings* but because he believes these terms are ambiguous, he refers to the direct effects as *pattern coefficients*. We assign them to factors based on their highest loadings (and

hopefully no cross-loadings). Depending on whether we select an orthogonal or oblique relationship, there may be an identified correlation between factors.

- CFA is a **restricted measurement model**. The researcher specifies the factor(s) on which each indicator/item(s) depends (recall, the causal direction in CFA is from factor to indicators/items.)
- **Identification status** The *identification* of a model has to do with whether it is theoretically possible for a computer to derive a unique set of model parameter estimates. Identification is related to model *degrees of freedom*; we will later explore under-, just-, and over-identified models. For now...
 - EFA models with multiple factors are *unidentified* because they will have more free parameters than observations. Thus, there is no unique set of statistical estimates for the multifactor EFA model, consequently this requires the rotation phase in EFA.
 - CFA models must be identified before they can be analyzed so there is only one unique set of parameter estimates. Correspondingly, there is no rotation phase in CFA.
- **Sharing variances**
 - In EFA the specific variance of each indicator is not shared with that of any other indicator.
 - In CFA, the researchers can specify if variance is shared between certain pairs of indicators (i.e., error covariances).

10.2.3 On the relationship between EFA and CFA

Kline (?) admonishes us to not overinterpret the labels “exploratory” and “confirmatory”. Why?

- EFA requires no a priori hypotheses about the relationship between indicators/items and factors, but researchers often expect to specify a predetermined number of factors.
- CFA is not strictly confirmatory. After initial runs, many researchers modify models and hypotheses.

CFA is not a verification or confirmation of EFA results for the same data and number of factors. Kline (?) does not recommend that researchers follow a model retained from EFA. Why?

- It is possible that the CFA model will be rejected. Oftentimes this is because the secondary coefficients (i.e., non-primary pattern coefficients) accounted for a significant proportion of variance in the model. When they are constrained to 0.0 in the CFA model, the model fit will suffer.
- If the CFA model is retained, then it is possible that both EFA and CFA capitalized on chance variation. Thus, if verification via CFA is desired, it should be evaluated through a replication sample.

10.3 Exploring a Standard CFA Model

The research vignette for today is a fairly standard CFA model.

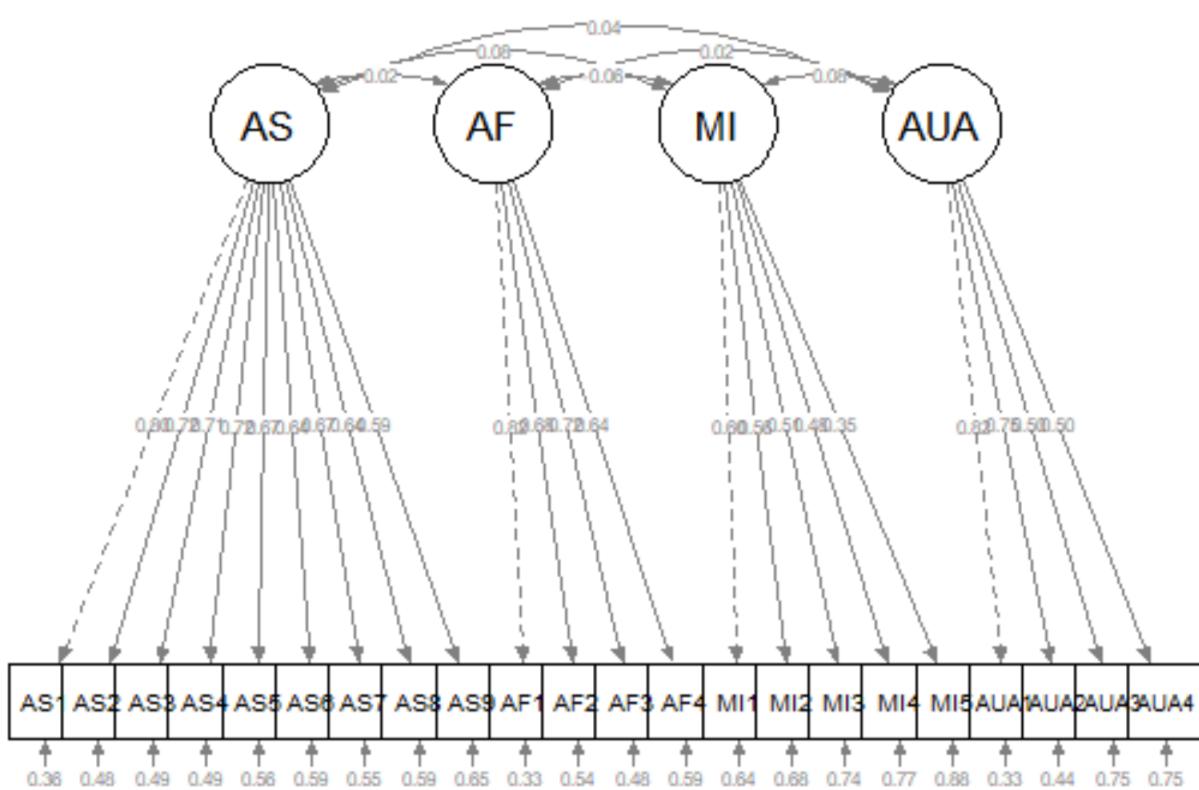


Figure 10.2: Image of the GRMSAAW represented as a standard CFA model

The image represents represents the hypothesis that $AS_1 - AS_9$, $AF_1 - AF_4$, $MI_1 - MI_5$, and $AUA_1 - AUA_4$ measure, respectively, the AS, AF, MI, and AUA factors, which are assumed to covary. Specifically,in this model:

1. Each indicator is continuous with two causes: $AS \rightarrow AS_1 \leftarrow E_1$
 - a single factor that the indicator is supposed to measure, and
 - all unique sources of influence represented by the error term
2. The error terms are independent of each other and of the factors
3. All associations are linear and the factors covary.
 - hence, the symbol for an unanalyzed association is a solid line (upgraded from the dashed one in the EFA)
4. Each item has a single *pattern coefficient* (i.e., often more casually termed as a “factor loading”)
 - All other potential pattern coefficients are set to “0.00.” These are *hard hypotheses* and are specified by their absence (i.e., not specified in the code or in the diagram).
5. *Structure coefficients* are the Pearson correlations between factors and continuous indicators. They reflect any source of association, causal or noncausal. Sometimes the association is an undirected, back-door path. There is no pattern coefficient for $AS_2 \leftrightarrow AF$. BUT, there is a connection from AS_2 to AF via the $AS \leftrightarrow AF$ covariance.
6. *Scaling constants* (aka *unit loading identification [ULI] constraints*) are necessary to scale the factors in a metric related to that of the explained (common) variance of the corresponding indicator, or *reference (marker) variable*. In the figure these are the dashed-line paths from $AS \rightarrow AS_1$, $AF \rightarrow AF_1$, $MI \rightarrow MI_1$ and $AUA \rightarrow AUA_1$.
 - Selecting the reference marker variable is usually arbitrary and selected by the computer program as the first (or last) variable in the code/path. So long as all the indicator variables of the same factor have equally reliable scores, this works satisfactorily.
 - Additional scaling constants are found for each of the errors and indicators.

10.3.1 Model Identification for CFA

SEM, in general, requires that all models be *identified*. Measurement models analyzed in CFA share this requirement, but identification is more straightforward than in other models.

Standard CFA models are sufficiently identified when:

1. A single factor model has at least three indicators, or
2. In a model with two or more factors, each factor has two or more indicators.
 - Note: It is better to have at least three to five indicators per factor to prevent technical problems with statistical identification.

Identification becomes much more complicated than this, but for today’s models this instruction is sufficient.

10.3.2 Selecting Indicators/Items for a Reflective Measurement

Reflective measurement is another term to describe the circumstance where latent variables are assumed to cause observed variables. Observed variables in reflective measurement are called *effect (reflective) indicators*.

- At least three for a unidimensional model; at least two per factor for a multidimensional model (but more is safer).
- The items/indicators should have reasonable internal consistency, they should correlate with each other, and correlate more with themselves than with items on other factors (if multidimensional).
- Negative correlations reduce the reliability of factor measurement, so they should be reverse coded prior to analysis.
- Do not be tempted to specify a factor with indicators that do not measure something. A common mistake is to create a “background” factor and include indicators such as gender, ethnicity, and level of education. *Just what is the predicted relationship between gender and ethnicity?*

10.4 CFA Workflow

Below is a screenshot of a CFA workflow. The original document is located in the [Github site](#) that hosts the ReCentering Psych Stats: Psychometrics OER.

Because the intended audience for the ReCentering Psych Stats OER is the scientist-practitioner-advocate, this lesson focuses on the workflow and decisions. As you might guess, the details of CFA can be quite complex and require more investigation and decision-making in models that pose more complexity or empirical challenges.

- Creating an items only dataframe where any items are scaled in the same direction (e.g., negatively worded items are reverse-scored).
- Determining a factor structure that is *identified*, that is
 - A single factor (unidimensional) model has at least three items/indicators
 - Multidimensional models have at least two items per factor
- Specify a series of models, these typically include
 - A unidimensional model (all items on a single factor)
 - A single order structure with correlated factors
 - A second order structure
 - A bifactor structure
- Evaluate model fit with a variety of indicators
 - factor loadings
 - fit indices
- Compare models
- In the event of poor model fit, investigate modification indices and consider respecification

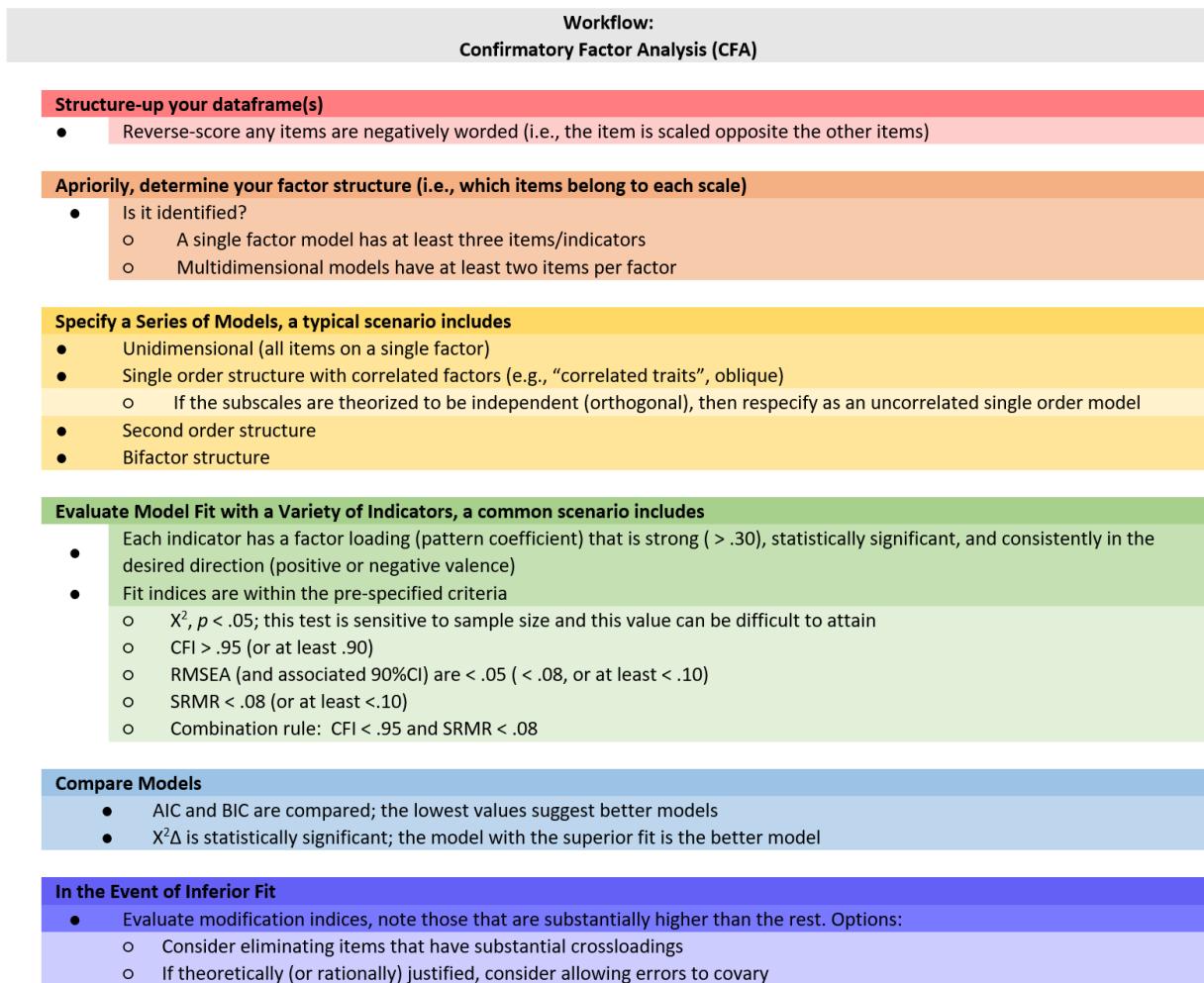


Figure 10.3: Image of a workflow for specifying and evaluating a confirmatory factor analytic model

- eliminating items
- changing factor membership
- allowing errors to covary

10.4.1 CFA in *lavaan* Requires Fluency with the Syntax

- It's really just regression
 - tilda (\sim , *is regressed on*) is regression operator
 - place DV (y) on left of operator
 - place IVs, separate by + on the right
- f is a latent variable (LV)
- Example: $y \sim f1 + f2 + x1 + x2$
- LVs must be *defined* by their manifest or latent indicators.
 - the special operator ($=\sim$, *is measured/defined by*) is used for this
 - Example: $f1 =\sim y1 + y2 + y3$
- Variances and covariances are specified with a double tilde operator ($\sim\sim$, *is correlated with*)
 - Example of variance: $y1 \sim\sim y1$ (the relationship with itself)
 - Example of covariance: $y1 \sim\sim y2$ (relationship with another variable)
 - Example of covariance of a factor: $f1 \sim\sim f2$

*Intercepts (~ 1) for observed and LVs are simple, intercept-only regression formulas + Example of variable intercept: $y1 \sim 1$ + Example of factor intercept: $f1 \sim 1$

A complete lavaan model is a combination of these formula types, enclosed between single quotation marks. Readibility of model syntax is improved by:

- splitting formulas over multiple lines
- using blank lines within single quote
- labeling with the hashtag

```
myModel <- '# regressions y1 + y2 ~ f1 + f2 + x1 + x2 f1 ~ f2 + f3 f2 ~ f3 + x1 + x2
```

```
# latent variable definitions
f1 =\sim y1 + y2 + y3
f2 =\sim y4 + y5 + y6
f3 =\sim y7 + y8 + y9 + y10

# variances and covariances
y1 ~~ y1
y2 ~~ y2
f1 ~~ f2
```

```
# intercepts
y1 ~ 1
fa ~ 1
```

10.4.2 Differing Factor Structures

All models worked in this lesson are *first-order* (or single-order) models; in the next lesson we extend to hierarchical and bifactor models. To provide an advanced cognitive organizer, let's take a look across the models.

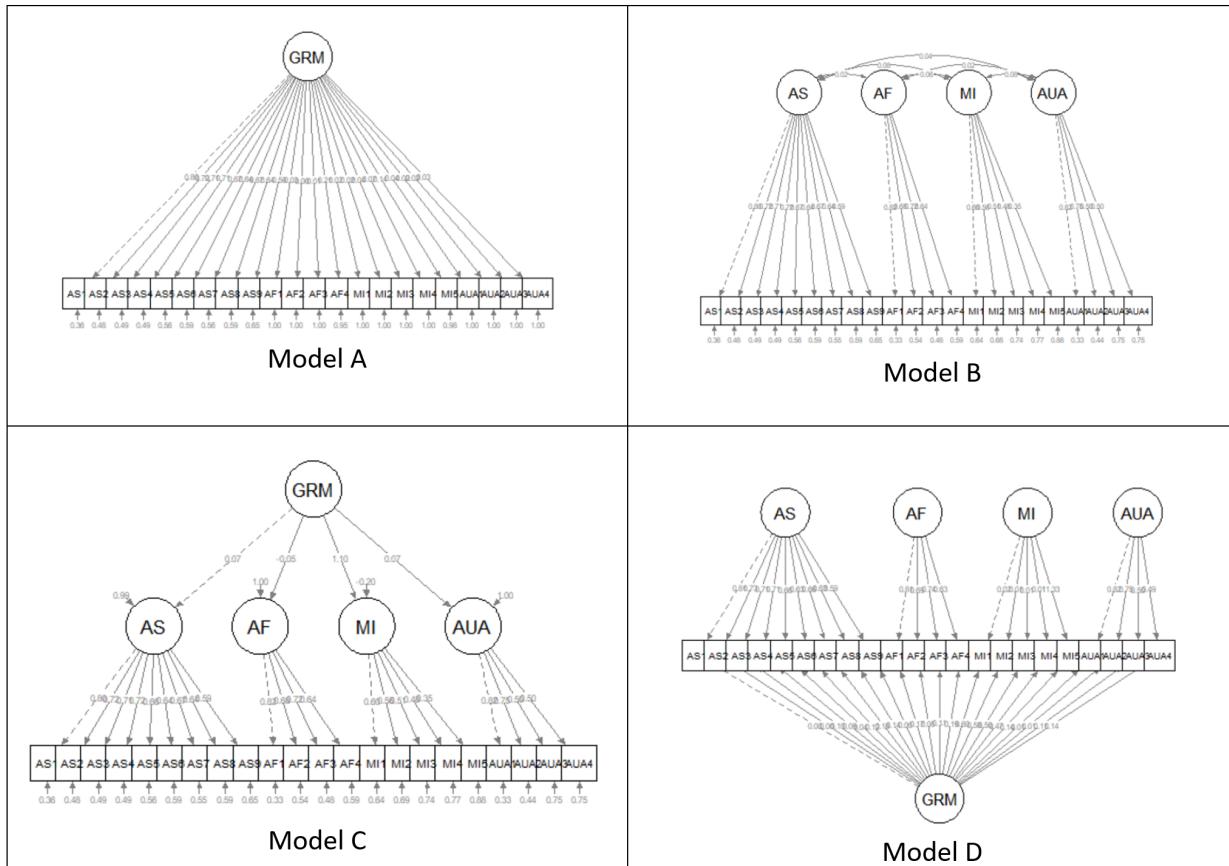


Figure 10.4: Image of first order (uncorrelated and correlated, second order, and bifactor structures)

Models A and B are first-order models. Note that all factors are on a single plane.

- Model A is undimensional, each item is influenced by a single common factor and a term that includes systematic and random error. Note that there is only one *systematic* source of variance for each item and it is from a single source.
- Model B is often referred to as a “correlated traits” model. Here, the larger construct is separated into distinct-yet-correlated elements. The variance of each item is assumed to be a weighted linear function of two or more common factors.

- Models C is a second-order factor structure. Rather than merely being correlated, factors are related because they share a common cause. In this model, the second order factor *explains* why three or more traits are correlated. Note that here is no direct relationship between the item and the target construct. Rather, the relationship between the second-order factor and each item is mediated through the primary factor (yes, an indirect effect!).
- Model D is a bifactor structure. Here each item loads on a general factor. This general factor (bottom row) reflects what is common among the items and represents the individual differences on the target dimension that a researcher is most interested in. Group factors (top row) are now specified as *orthogonal*. The group factors represent common factors measured by the items that explain item response variation not accounted for by the general factor. In some research scenarios, the group factors are termed “nuisance” dimensions. That is, that which they have in common interferes with measuring the primary target of interest.

10.5 Research Vignette

This lesson’s research vignette emerges from Keum et al’s Gendered Racial Microaggressions Scale for Asian American Women (GRMSAAW; (?)). The article reports on two separate studies that comprised the development, refinement, and psychometric evaluation of two, parallel, versions (stress appraisal, frequency) of scale. We simulate data from the final construction of the frequency version as the basis of the lecture. If the scale looks somewhat familiar it is because the authors used the Gendered Racial Microaggressions Scale for Black Women (?) as a model.

Keum et al. (?) reported support for a total scale score (22 items) and four subscales. Below, I list the four subscales, their number of items, and a single example item. At the outset, let me provide a content warning. For those who hold this particular identity (or related identities) the content in the items may be upsetting. In other lessons, I often provide a variable name that gives an indication of the primary content of the item. In the case of the GRMSAAW, I will simply provide an abbreviation of the subscale name and its respective item number. This will allow us to easily inspect the alignment of the item with its intended factor, and hopefully minimize discomfort. If you are not a member of this particular identity, I encourage you to learn about these microaggressions by reading the article in its entirety. Please do not ask members of this group to explain why these microaggressions are harmful or ask if they have encountered them.

There are 22 items on the GRMSAAW scale. The frequency scaling ranged included: 0(*never*), 1 (*rarely*), 2(*sometimes*), 3(*often*), 4(*very frequently*), and 5(*always*).

The four factors, number of items, and sample item are as follows:

- Ascribed Submissiveness
 - 9 items
 - “Others have been surprised when I disagree with them.”
 - Abbreviated in the simulated data as “AS#”
- Asian Fetishism
 - 4 items
 - “Others have treated me as if I am always open to sexual advances.”
 - Abbreviated in the simulated data as “AF#”

- Media Invalidation
 - 5 items
 - “I see AAW playing the same type of characters (e.g., Kung Fu woman, sidekick, mistress, tiger mom) in the media.”
 - Abbreviated in the simulated data as “MI#”
- Assumptions of Universal Appearance
 - 4 items
 - “Others have pointed out physical traits in AAW that do not look ‘Asian’”
 - Abbreviated in the simulated data as “UA#”

Below I walk through the data simulation. This is not an essential portion of the lesson, but I will lecture it in case you are interested. None of the items are negatively worded (relative to the other items), so there is no need to reverse-score any items.

```
#The GRMSAAW has two scales: frequency and stress appraisal. This simulation is for the frequency scale.
set.seed(210927)
GRMSAAWmat <- matrix(c(.83, .79, .75, .72, .70, .69, .69, .63, -.06, -.01, -.02, .21, -.03,
                        .07, -.03, -.06, -.02, .08, -.06, -.01, -.03, .13, .85, .76, .75, .70,
                        -.11, -.06, .04, .02, -.03, .04, .15, .08, -.03, -.10, .11, .13, -.13,
                        .03, .05, -.01, -.06, .04, .08, -.13, .03, .02, .07, .06, -.11, -.02, -.01),
                      nrow = 10, ncol = 10)
rownames(GRMSAAWmat) <- c("AS1", "AS2", "AS3", "AS4", "AS5", "AS6", "AS7", "AS8", "AS9", "AF1")
colnames(GRMSAAWmat) <- c("Submissiveness", "Fetishism", "Media", "Appearance") #component (subscale) names
GRMSAAWCorMat <- GRMSAAWmat %*% t(GRMSAAWmat) #create the correlation matrix via some matrix algebra
diag(GRMSAAWCorMat) <- 1
#GRMSAAWCorMat #prints the correlation matrix
GRMSAAW_M <- c(2.91, 3.3, 3.45, 2.85, 3.89, 3.11, 3.83, 3.07, 2.88, 3.3, 3.64)
GRMSAAW_SD <- c(1.21, 0.81, 1.34, 1.62, 1.89, 0.93, 1.01, 1.17, 1.22, 1.28, 1.42)
GRMSAAWCovMat <- GRMSAAW_SD %*% t(GRMSAAW_SD) * GRMSAAWCorMat #creates a covariance matrix (with correlations)
dfGRMSAAW <- as.data.frame(round(MASS::mvrnorm(n=304, mu = GRMSAAW_M, Sigma = GRMSAAWCovMat,
                                         empirical = TRUE)))
dfGRMSAAW[dfGRMSAAW>5] <- 5 #restricts the upperbound of all variables to be 5 or less
dfGRMSAAW[dfGRMSAAW<0] <- 0 #restricts the lowerbound of all variable to be 0 or greater

#Below is code if you would like an ID number for each case. Especially at first, the ID numbers are not unique.
#library(tidyverse)
#dfGRMSAAW <- dfGRMSAAW %>% dplyr::mutate(ID = row_number()) #add ID to each row
#dfGRMSAAW <- dfGRMSAAW %>% dplyr::select(ID, everything())#moving the ID number to the first column
```

Let's take a quick peek at the data to see if everthing looks correct.

```
psych::describe(dfGRMSAAW)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
AS1	1	304	2.90	1.22	3	2.91	1.48	0	5	5	-0.21	-0.48	0.07
AS2	2	304	3.28	0.84	3	3.27	1.48	1	5	4	0.06	-0.24	0.05
AS3	3	304	3.42	1.25	4	3.50	1.48	0	5	5	-0.45	-0.53	0.07

AS4	4 304 2.81 1.47	3	2.87 1.48	0 5	5 -0.22	-0.81 0.08
AS5	5 304 3.60 1.41	4	3.77 1.48	0 5	5 -0.68	-0.56 0.08
AS6	6 304 3.11 0.95	3	3.11 1.48	0 5	5 -0.13	-0.20 0.05
AS7	7 304 3.77 0.96	4	3.85 1.48	1 5	4 -0.55	-0.04 0.05
AS8	8 304 3.04 1.15	3	3.07 1.48	0 5	5 -0.28	-0.24 0.07
AS9	9 304 2.87 1.21	3	2.92 1.48	0 5	5 -0.30	-0.47 0.07
AF1	10 304 3.25 1.24	3	3.32 1.48	0 5	5 -0.30	-0.61 0.07
AF2	11 304 3.52 1.23	4	3.62 1.48	0 5	5 -0.52	-0.41 0.07
AF3	12 304 3.17 1.32	3	3.25 1.48	0 5	5 -0.32	-0.70 0.08
AF4	13 304 3.15 1.27	3	3.20 1.48	0 5	5 -0.23	-0.77 0.07
MI1	14 304 4.15 0.75	4	4.21 1.48	2 5	3 -0.49	-0.40 0.04
MI2	15 304 4.51 0.68	5	4.64 0.00	2 5	3 -1.24	0.96 0.04
MI3	16 304 4.47 0.72	5	4.60 0.00	2 5	3 -1.18	0.68 0.04
MI4	17 304 4.35 0.75	4	4.45 1.48	2 5	3 -0.89	0.13 0.04
MI5	18 304 4.61 0.63	5	4.72 0.00	2 5	3 -1.41	1.17 0.04
AUA1	19 304 4.24 0.84	4	4.34 1.48	1 5	4 -0.88	0.19 0.05
AUA2	20 304 4.38 0.80	5	4.51 0.00	1 5	4 -1.21	1.07 0.05
AUA3	21 304 4.27 0.86	4	4.39 1.48	1 5	4 -1.05	0.64 0.05
AUA4	22 304 4.24 0.86	4	4.36 1.48	2 5	3 -0.89	-0.12 0.05

The optional script below will let you save the simulated data to your computing environment as either a .csv file (think “Excel lite”) or .rds object (preserves any formatting you might do). If you save the .csv file and bring it back in, you will lose any formatting (e.g., ordered factors will be interpreted as character variables).

```
#write the simulated data as a .csv
#write.table(dfGRMSAAW, file="dfGRMSAAW.csv", sep=",", col.names=TRUE, row.names=FALSE)
#bring back the simulated dat from a .csv file
#dfGRMSAAW <- read.csv ("dfGRMSAAW.csv", header = TRUE)
```

An .rds file preserves all formatting to variables prior to the export and re-import. For the purpose of this chapter, you don’t need to do either. That is, you can re-simulate the data each time you work the problem.

```
#to save the df as an .rds (think "R object") file on your computer; it should save in the same directory
#saveRDS(dfGRMSAAW, "dfGRMSAAW.rds")
#bring back the simulated dat from an .rds file
#dfGRMSAAW <- readRDS("dfGRMSAAW.rds")
```

10.5.1 Modeling the GRMSAAW as Unidimensional

Let’s start simply, taking the GRMSAAW data and seeing about its fit as a unidimensional instrument. First evaluating multi-dimensional measures as unidimensional is a common practice. And there are two reasons:

- Operationally, it’s a check to see that data, script, and so forth. are all working.

- If you can't reject a single-factor model (e.g., if there is a strong support for such), then it makes little sense to evaluate models with more factors (?).

With a single factor model:

- GRMSAAW is a latent variable and can be named anything. We know this because it is followed by: =~
- All the items follow and are “added” with the plus sign
 - Don’t let this fool you...the assumption behind SEM/CFA is that the LV *causes* the score on the item/indicator. Recall, item/indicator scores are influenced by the LV and error.
- The entire model is enclosed in tic marks (‘ and ’)

```
grmsAAWmod1 <- 'GRMSAAW =~ AS1 + AS2 + AS3 + AS4 + AS5 + AS6 + AS7 + AS8 + AS9 + AF1 + AF2 + AF3 + AF4 + AF5'
```

```
[1] "GRMSAAW =~ AS1 + AS2 + AS3 + AS4 + AS5 + AS6 + AS7 + AS8 + AS9 + AF1 + AF2 + AF3 + AF4 + AF5"
```

The object representing the model is then included in the *lavaan::cfa()* along with the dataset.

We can ask for a summary of the object representing the results.

```
grmsAAW1fit <- lavaan::cfa (grmsAAWmod1, data = dfGRMSAAW)
lavaan::summary(grmsAAW1fit, fit.measures=TRUE, standardized=TRUE, rsquare = TRUE)
```

lavaan 0.6-9 ended normally after 28 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	44

Number of observations	304
------------------------	-----

Model Test User Model:

Test statistic	1004.136
Degrees of freedom	209
P-value (Chi-square)	0.000

Model Test Baseline Model:

Test statistic	2114.899
Degrees of freedom	231
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.578
Tucker-Lewis Index (TLI)	0.534

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-8833.514
Loglikelihood unrestricted model (H1)	-8331.446
Akaike (AIC)	17755.028
Bayesian (BIC)	17918.577
Sample-size adjusted Bayesian (BIC)	17779.032

Root Mean Square Error of Approximation:

RMSEA	0.112
90 Percent confidence interval - lower	0.105
90 Percent confidence interval - upper	0.119
P-value RMSEA <= 0.05	0.000

Standardized Root Mean Square Residual:

SRMR	0.124
------	-------

Parameter Estimates:

Standard errors	Standard
Information	Expected
Information saturated (h1) model	Structured

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
GRMSAAW =~						
AS1	1.000				0.972	0.799
AS2	0.625	0.047	13.208	0.000	0.607	0.722
AS3	0.911	0.070	12.969	0.000	0.886	0.711
AS4	1.082	0.083	13.048	0.000	1.052	0.715
AS5	0.966	0.081	11.978	0.000	0.939	0.665
AS6	0.626	0.055	11.402	0.000	0.608	0.638
AS7	0.655	0.055	11.980	0.000	0.637	0.665
AS8	0.757	0.066	11.450	0.000	0.736	0.640
AS9	0.738	0.070	10.491	0.000	0.717	0.594
AF1	-0.037	0.077	-0.487	0.626	-0.036	-0.030
AF2	0.006	0.077	0.080	0.937	0.006	0.005
AF3	-0.019	0.082	-0.235	0.814	-0.019	-0.014
AF4	0.280	0.079	3.567	0.000	0.272	0.214
MI1	0.017	0.047	0.367	0.714	0.017	0.022
MI2	0.014	0.042	0.323	0.747	0.013	0.020

MI3	0.027	0.045	0.610	0.542	0.027	0.037
MI4	0.054	0.046	1.167	0.243	0.053	0.071
MI5	0.092	0.039	2.370	0.018	0.089	0.143
AUA1	0.033	0.052	0.627	0.530	0.032	0.038
AUA2	0.016	0.050	0.326	0.745	0.016	0.020
AUA3	0.019	0.054	0.361	0.718	0.019	0.022
AUA4	0.030	0.054	0.565	0.572	0.030	0.034

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.AS1	0.535	0.055	9.739	0.000	0.535	0.361
.AS2	0.339	0.032	10.739	0.000	0.339	0.479
.AS3	0.768	0.071	10.835	0.000	0.768	0.494
.AS4	1.060	0.098	10.804	0.000	1.060	0.489
.AS5	1.109	0.099	11.171	0.000	1.109	0.557
.AS6	0.539	0.048	11.329	0.000	0.539	0.593
.AS7	0.510	0.046	11.170	0.000	0.510	0.557
.AS8	0.779	0.069	11.316	0.000	0.779	0.590
.AS9	0.946	0.082	11.537	0.000	0.946	0.648
.AF1	1.523	0.124	12.328	0.000	1.523	0.999
.AF2	1.519	0.123	12.329	0.000	1.519	1.000
.AF3	1.729	0.140	12.329	0.000	1.729	1.000
.AF4	1.539	0.126	12.259	0.000	1.539	0.954
.MI1	0.562	0.046	12.328	0.000	0.562	1.000
.MI2	0.460	0.037	12.328	0.000	0.460	1.000
.MI3	0.518	0.042	12.327	0.000	0.518	0.999
.MI4	0.552	0.045	12.322	0.000	0.552	0.995
.MI5	0.382	0.031	12.299	0.000	0.382	0.980
.AUA1	0.695	0.056	12.327	0.000	0.695	0.999
.AUA2	0.637	0.052	12.328	0.000	0.637	1.000
.AUA3	0.738	0.060	12.328	0.000	0.738	1.000
.AUA4	0.743	0.060	12.327	0.000	0.743	0.999
GRMSAAW	0.945	0.117	8.087	0.000	1.000	1.000

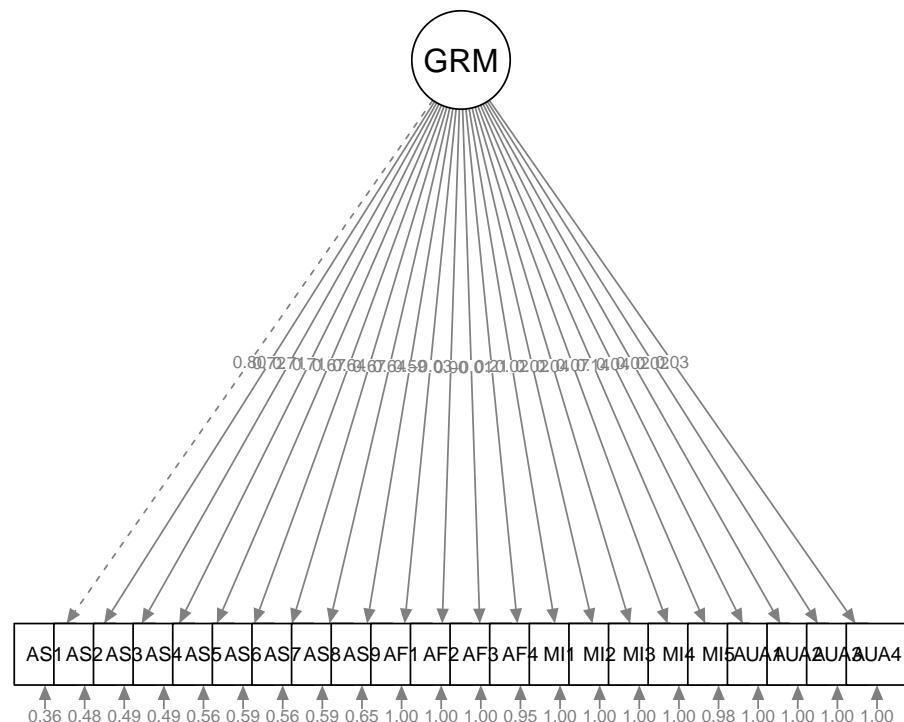
R-Square:

	Estimate
AS1	0.639
AS2	0.521
AS3	0.506
AS4	0.511
AS5	0.443
AS6	0.407
AS7	0.443
AS8	0.410
AS9	0.352
AF1	0.001
AF2	0.000
AF3	0.000

AF4	0.046
MI1	0.000
MI2	0.000
MI3	0.001
MI4	0.005
MI5	0.020
AUA1	0.001
AUA2	0.000
AUA3	0.000
AUA4	0.001

I find it helpful to immediately plot what we did. A quick look alerts me to errors.

```
semPlot::semPaths(grmsAAW1fit, layout = "tree", style = "lisrel", what = "col", whatLabels = "name")
```



10.5.1.1 Interpreting the Output

With a quick look at the plot, let's work through the results. Rosseel's (2019) *lavaan* tutorial is a useful resource in walking through the output.

The *header* is the first few lines of the information. It contains:

- the *lavaan* version number (0.6-9 that I'm using on 10/4/2021)
- maximum likelihood (ML) was used as the estimator
- confirmation that the specification converged normally after 28 iterations
- 304 cases were used in this analysis (would be less if some were skipped because of missing data)
- the model user test statistic, df, and corresponding p value: $\chi^2(209) = 1004.136, p < .001$

Fit statistics are included in the second section. They are only shown when the argument “fit.measures = TRUE” is in the script. Standardized values are not the default, they require the argument, “standardized = TRUE”. We'll come back to these shortly...

Parameter estimates is the last section.

For now we are interested in the Latent Variables section.

- *Estimate* contains the estimated or fixed parameter value for each model parameter;
- *Std. err* is the standard error for each estimated parameter;
- *Z-value* is the Wald statistic (the parameter divided by its SE)
- *P(>/z|)* is the p value for testing the null hypothesis that the parameter equals zero in the population
- *Std.lv* standardizes only the LVs
- *Std.all* both latent and observed variables are standardized; this is considered the “completely standardized solution”

Note that item AS1 might seem incomplete – there is only a 1.000 and a value for the Std.lv. Recall we used this to scale the single factor by fixing its value to 1.000. Coefficients that are fixed to 1.0 to scale a factor have no standard errors and therefore no significance test.

The SE and associated *p* values are associated with the unstandardized estimates. Intuitively, it is easiest for me to understand the relative magnitude of the pattern coefficients by looking at the *Std.all* column. We can see that the items associated with what we will soon define as the AS factor are all strong and positive. The remaining items have variable loadings with many of the being quite low, non-significant, and even negatively valenced.

Let's examine to the middle set metrics which assess *global fit*.

CFA falls into a *modeling* approach to evaluating results. While it provides some flexibility (we get away from the strict, NHST approach of $p < .05$) there is greater interpretive ambiguity.

Fit statistics tend to be clustered together based on their approach to summarizing the *goodness* or *badness* of fit.

10.5.1.2 Model Test *User Model*:

The chi-square statistic that evaluates the *exact-fit hypothesis* that there is no difference between the covariances predicted by the model, given the parameter estimates, and the population covariance matrix. Rejecting the hypothesis says that,

- the data contain covariance information that speak against the model, and
- the researcher should explain model-data discrepancies that exceed those expected by sampling error.

Traditional interpretation of the chi-square is an *accept-support test* where the null hypothesis represents the researchers' belief that the model is correct. This means that the absence of statistical significance ($p > .05$) that supports the model. This is backwards from our usual *reject-support test* approach.

The χ^2 is frequently criticized:

- *accept-support test* approaches are logically weaker because the failure to disprove an assertion (the exact-fit hypothesis) does not prove that the assertion is true;
- too small a sample size (low power) makes it more likely that the model will be retained;
- CFA/SEM, though, requires large samples and so the χ^2 is frequently statistically significant – which rejects the researchers' model;

Kline (?) recommends that we treat the χ^2 like a smoke alarm – if the alarm sounds, there may or may not be a fire (a serious model-data discrepancy), but we should treat the alarm seriously and further inspect issues of fit.

For our unidimensional GRMSAAW CFA $\chi^2(209) = 1004.136, p < .001$, this significant value is not what we want because it says that our specified model is different than the covariances in the model.

10.5.1.3 Model Test *Baseline Model*

This model is the *independence* model. That is, there is complete independence of all variables in the model (i.e., in which all correlations among variables are zero). This is the most restricted model. It is typical for chi-square values to be quite high (as it is in our example: 2114.899). On its own, this model is not useful to us. It is used, though, in comparisons of *incremental fit*.

10.5.1.4 Incremental Fit Indices (User versus Baseline Models)

Incremental fit indices ask the question, how much better is the fit of our specified model to the data than the baseline model (where it is assumed no relations between the variables).

The Comparative Fit Index (CFI) and Tucker-Lewis Index (TLI) are *goodness of fit* statistics, ranging from 0 to 1.0 where 1.0 is best.

CFI: compares the amount of departure from close fit for the researcher's model against that of the independence/baseline (null) model.

$$CFI = 1 - \frac{\hat{\Delta}_M}{\hat{\Delta}_B}$$

We can actually calculate this using the baseline and chi-square values from our own data:

```
1 - (1004.136/2114.899)
```

```
[1] 0.5252085
```

Where there is no departure from close fit, then CFI will equal 1.0. We interpret the value of the CFI as a percent of how much better the researcher's model is than the baseline model. While 58% sounds like an improvement – Hu and Bentler (1999) stated that “acceptable fit” is achieved when the $CFI \geq .95$ and $SRMR \leq .08$; the **combination rule**. It is important to note that later simulation studies have not supported those thresholds.

TLI: aka the **non-normed fit index (NNFI)** controls for df_M from the researcher's model and df_B from the baseline model. As such, it imposes a greater relative penalty for model complexity than the CFI. The TLI is a bit unstable in that the values can exceed 1.0.

Because the two measures are so related, only one should be reported (I typically see the CFI).

For our unidimensional GRMSAAW CFA, CFI = .578 and TLI = .534. While these predict around 58% better than the baseline/independence model, it does not come close to the standard of $\geq .95$.

I note that our hand calcuation of user and baseline models did not result in the exact CFI. I do not know why.

10.5.1.5 Loglikelihood and Information Criteria

The **Aikaike Information Criterion (AIC)** and the **Bayesian Information Criterion (BIC)** utilize an information theory approach to data analysis by combining statistical estimation and model selection into a single framework. The BIC augments the AIC by taking sample size into consideration.

The AIC and BIC are usually used to select among competing nonhierarchical models and are only used in comparison with each other. Thus our current values of 17755.028 (AIC) and 17918.577 (BIC) are meaningless on their own. The model with the smallest value of the predictive fit index is chosen as the one that is most likely to replicate. It means that this model has relatively better fit and fewer free parameters than competing models.

For our unidimensional GRMSAAW CFA we'll return to these values to compare a correlated, four-factor solution.

10.5.1.6 Root Mean Square Error of Approximation

The RMSEA is an absolute fit index scaled as a *badness-of-fit* statistic where a value of 0.00 is the best fit. The RMSEA favors models with more degrees of freedom and larger sample sizes. A unique aspect of the RMSEA is its 90% confidence interval.

While there is chatter/controversy about what constitutes an acceptable value, there is general consensus that $RMSEA \geq .10$ points to serious problems. An $RMSEA \leq .05$ is desired. Watching the upper bound of the confidence interval is important to see that it isn't sneaking into the danger zone.

For our unidimensional GRMSAAW CFA, $RMSEA = .112$, 90% CI(.105, .119). Unfortunately this value points to serious problems.

10.5.1.7 Standardized Root Mean Square Residual

The SRMR is an absolute fit index that is a *badness-of-fit* statistic (i.e., perfect model fit is when the value = 0.00 and increasingly higher values indicate the “badness”).

The SRMR is a standardized version of the **root mean square residual (RMR)**, which is a measure of the mean absolute covariance residual. Standardizing the value facilitates interpretation.

Poor fit is indicated when $SRMR \geq .10$.

Recall, Hu and Bentler's **combination rule** (which is somewhat contested) suggested that the SRMR be interpreted along with the CFI such that: $CFI \geq .95$ and $SRMR \leq .05$.

For our unidimensional GRMSAAW CFA, $SRMR = .124$. Not good.

Inspecting the residuals (we look for relatively large values) may help understand the source of poor fit, so let's do that.

```
lavaan::fitted(grmsAAW1fit)
```

	\$cov									
	AS1	AS2	AS3	AS4	AS5	AS6	AS7	AS8	AS9	AF1
AS1	1.480									
AS2	0.590	0.708								
AS3	0.861	0.538	1.552							
AS4	1.022	0.639	0.932	2.166						
AS5	0.912	0.570	0.832	0.987	1.990					
AS6	0.591	0.369	0.539	0.639	0.571	0.909				
AS7	0.619	0.387	0.564	0.670	0.598	0.387	0.916			
AS8	0.715	0.447	0.652	0.774	0.691	0.448	0.469	1.321		
AS9	0.697	0.436	0.636	0.754	0.673	0.436	0.457	0.528	1.461	
AF1	-0.035	-0.022	-0.032	-0.038	-0.034	-0.022	-0.023	-0.027	-0.026	1.525
AF2	0.006	0.004	0.005	0.006	0.006	0.004	0.004	0.004	0.004	0.000
AF3	-0.018	-0.011	-0.017	-0.020	-0.018	-0.011	-0.012	-0.014	-0.013	0.001
AF4	0.265	0.165	0.241	0.286	0.256	0.166	0.173	0.200	0.195	-0.010
MI1	0.016	0.010	0.015	0.018	0.016	0.010	0.011	0.012	0.012	-0.001
MI2	0.013	0.008	0.012	0.014	0.012	0.008	0.008	0.010	0.010	0.000
MI3	0.026	0.016	0.024	0.028	0.025	0.016	0.017	0.020	0.019	-0.001
MI4	0.051	0.032	0.047	0.055	0.049	0.032	0.034	0.039	0.038	-0.002
MI5	0.087	0.054	0.079	0.094	0.084	0.054	0.057	0.066	0.064	-0.003
AUA1	0.031	0.019	0.028	0.033	0.030	0.019	0.020	0.023	0.023	-0.001
AUA2	0.015	0.010	0.014	0.017	0.015	0.010	0.010	0.012	0.011	-0.001


```
#lavaan::residuals(grmsAAW1fit, type = "raw")
#lavaan::residuals(grmsAAW1fit, type = "standardized")

#will hashtag out for knitted file
lavaan::residuals(grmsAAW1fit, type = "cor")
```

```
$type
[1] "cor.bollen"

$cov
   AS1    AS2    AS3    AS4    AS5    AS6    AS7    AS8    AS9    AF1
AS1  0.000
AS2  0.020  0.000
AS3 -0.011 -0.022  0.000
AS4 -0.003  0.007  0.003  0.000
AS5 -0.002  0.013  0.000 -0.003  0.000
AS6 -0.018 -0.003  0.014  0.000 -0.012  0.000
AS7 -0.004 -0.007  0.031  0.017 -0.025  0.019  0.000
AS8  0.004 -0.018 -0.010  0.004  0.004  0.020  0.001  0.000
AS9  0.013  0.005  0.013 -0.025  0.026 -0.009 -0.031 -0.021  0.000
AF1  0.030 -0.009 -0.031 -0.018  0.029 -0.089 -0.081 -0.044  0.116  0.000
AF2 -0.003 -0.056 -0.036 -0.038  0.083 -0.061 -0.014  0.017  0.089  0.557
AF3  0.008 -0.043 -0.018 -0.004  0.045 -0.072 -0.031 -0.024  0.070  0.580
AF4  0.049 -0.015 -0.013 -0.019  0.041 -0.055 -0.068 -0.006  0.102  0.542
MI1 -0.069 -0.036  0.008  0.034 -0.016  0.022  0.062  0.044 -0.039 -0.023
MI2 -0.060 -0.064  0.001  0.023 -0.025  0.051  0.080  0.046 -0.016 -0.182
MI3 -0.061 -0.032  0.037  0.001 -0.014  0.029  0.044  0.049 -0.036 -0.111
MI4 -0.061  0.000  0.021 -0.018 -0.009  0.007  0.034  0.054  0.004 -0.047
MI5 -0.024 -0.031 -0.004  0.018 -0.036  0.016  0.009  0.081 -0.013  0.027
AUA1 0.003 -0.006 -0.013 -0.044  0.026  0.087 -0.091 -0.027  0.067  0.003
AUA2 0.024 -0.011 -0.019 -0.043  0.041  0.057 -0.073 -0.029  0.036  0.039
AUA3 -0.016  0.015  0.001 -0.045 -0.019  0.073 -0.065 -0.002  0.062  0.066
AUA4  0.037 -0.023 -0.024 -0.033 -0.013  0.017 -0.046  0.008  0.032  0.147
          AF2    AF3    AF4    MI1    MI2    MI3    MI4    MI5    AUA1    AUA2
AS1
AS2
AS3
AS4
AS5
AS6
AS7
AS8
AS9
AF1
AF2  0.000
AF3  0.512  0.000
AF4  0.403  0.466  0.000
```

```

MI1   0.138  0.187 -0.056  0.000
MI2  -0.050 -0.015 -0.188  0.332  0.000
MI3   0.023  0.015 -0.138  0.288  0.293  0.000
MI4   0.054  0.044 -0.052  0.305  0.273  0.228  0.000
MI5   0.020  0.030 -0.011  0.251  0.157  0.188  0.127  0.000
AUA1  0.006 -0.095 -0.011 -0.096  0.090  0.095  0.033 -0.013  0.000
AUA2  0.008 -0.063 -0.021 -0.058  0.088  0.076 -0.040 -0.070  0.618  0.000
AUA3  0.073  0.002 -0.009  0.027  0.103  0.137  0.026  0.038  0.417  0.347
AUA4  0.128  0.082  0.056  0.003  0.151  0.127  0.112 -0.022  0.384  0.381
      AUA3    AUA4
AS1
AS2
AS3
AS4
AS5
AS6
AS7
AS8
AS9
AF1
AF2
AF3
AF4
MI1
MI2
MI3
MI4
MI5
AUA1
AUA2
AUA3  0.000
AUA4  0.291  0.000

```

```
lavaan:::modindices(grmsAAW1fit)
```

	lhs	op	rhs	mi	epc	sepc.lv	sepc.all	sepc.nox
46	AS1	~~	AS2	1.135	0.033	0.033	0.078	0.078
47	AS1	~~	AS3	0.307	-0.026	-0.026	-0.040	-0.040
48	AS1	~~	AS4	0.024	-0.008	-0.008	-0.011	-0.011
49	AS1	~~	AS5	0.006	-0.004	-0.004	-0.005	-0.005
50	AS1	~~	AS6	0.661	-0.030	-0.030	-0.057	-0.057
51	AS1	~~	AS7	0.037	-0.007	-0.007	-0.014	-0.014
52	AS1	~~	AS8	0.036	0.009	0.009	0.013	0.013
53	AS1	~~	AS9	0.333	0.028	0.028	0.040	0.040
54	AS1	~~	AF1	0.945	0.057	0.057	0.063	0.063
55	AS1	~~	AF2	0.012	-0.006	-0.006	-0.007	-0.007
56	AS1	~~	AF3	0.070	0.016	0.016	0.017	0.017

57	AS1	~~	AF4	2.723	0.097	0.097	0.107	0.107
58	AS1	~~	MI1	5.109	-0.080	-0.080	-0.146	-0.146
59	AS1	~~	MI2	3.806	-0.062	-0.062	-0.126	-0.126
60	AS1	~~	MI3	3.975	-0.068	-0.068	-0.128	-0.128
61	AS1	~~	MI4	3.999	-0.070	-0.070	-0.129	-0.129
62	AS1	~~	MI5	0.653	-0.024	-0.024	-0.052	-0.052
63	AS1	~~	AUA1	0.013	0.004	0.004	0.007	0.007
64	AS1	~~	AUA2	0.627	0.030	0.030	0.051	0.051
65	AS1	~~	AUA3	0.273	-0.021	-0.021	-0.034	-0.034
66	AS1	~~	AUA4	1.461	0.049	0.049	0.078	0.078
67	AS2	~~	AS3	0.886	-0.033	-0.033	-0.064	-0.064
68	AS2	~~	AS4	0.098	0.013	0.013	0.021	0.021
69	AS2	~~	AS5	0.268	0.021	0.021	0.034	0.034
70	AS2	~~	AS6	0.015	-0.003	-0.003	-0.008	-0.008
71	AS2	~~	AS7	0.075	-0.008	-0.008	-0.018	-0.018
72	AS2	~~	AS8	0.445	-0.022	-0.022	-0.044	-0.044
73	AS2	~~	AS9	0.026	0.006	0.006	0.011	0.011
74	AS2	~~	AF1	0.057	-0.011	-0.011	-0.015	-0.015
75	AS2	~~	AF2	2.256	-0.066	-0.066	-0.092	-0.092
76	AS2	~~	AF3	1.326	-0.054	-0.054	-0.071	-0.071
77	AS2	~~	AF4	0.172	-0.018	-0.018	-0.026	-0.026
78	AS2	~~	MI1	0.953	-0.026	-0.026	-0.060	-0.060
79	AS2	~~	MI2	2.963	-0.042	-0.042	-0.106	-0.106
80	AS2	~~	MI3	0.747	-0.022	-0.022	-0.053	-0.053
81	AS2	~~	MI4	0.000	0.000	0.000	-0.001	-0.001
82	AS2	~~	MI5	0.717	-0.019	-0.019	-0.052	-0.052
83	AS2	~~	AUA1	0.026	-0.005	-0.005	-0.010	-0.010
84	AS2	~~	AUA2	0.096	-0.009	-0.009	-0.019	-0.019
85	AS2	~~	AUA3	0.167	0.013	0.013	0.025	0.025
86	AS2	~~	AUA4	0.396	-0.019	-0.019	-0.039	-0.039
87	AS3	~~	AS4	0.014	0.007	0.007	0.008	0.008
88	AS3	~~	AS5	0.000	0.001	0.001	0.001	0.001
89	AS3	~~	AS6	0.258	0.021	0.021	0.033	0.033
90	AS3	~~	AS7	1.376	0.048	0.048	0.077	0.077
91	AS3	~~	AS8	0.125	-0.018	-0.018	-0.023	-0.023
92	AS3	~~	AS9	0.209	0.025	0.025	0.029	0.029
93	AS3	~~	AF1	0.659	-0.054	-0.054	-0.050	-0.050
94	AS3	~~	AF2	0.929	-0.064	-0.064	-0.059	-0.059
95	AS3	~~	AF3	0.233	-0.034	-0.034	-0.030	-0.030
96	AS3	~~	AF4	0.120	-0.023	-0.023	-0.021	-0.021
97	AS3	~~	MI1	0.046	0.009	0.009	0.013	0.013
98	AS3	~~	MI2	0.001	0.001	0.001	0.002	0.002
99	AS3	~~	MI3	0.957	0.038	0.038	0.060	0.060
100	AS3	~~	MI4	0.313	0.022	0.022	0.034	0.034
101	AS3	~~	MI5	0.011	-0.004	-0.004	-0.007	-0.007
102	AS3	~~	AUA1	0.114	-0.015	-0.015	-0.021	-0.021
103	AS3	~~	AUA2	0.249	-0.021	-0.021	-0.031	-0.031
104	AS3	~~	AUA3	0.000	0.001	0.001	0.001	0.001

105	AS3	~~	AUA4	0.406	-0.029	-0.029	-0.039	-0.039
106	AS4	~~	AS5	0.014	-0.008	-0.008	-0.008	-0.008
107	AS4	~~	AS6	0.000	0.000	0.000	0.000	0.000
108	AS4	~~	AS7	0.408	0.031	0.031	0.042	0.042
109	AS4	~~	AS8	0.021	0.009	0.009	0.009	0.009
110	AS4	~~	AS9	0.772	-0.057	-0.057	-0.057	-0.057
111	AS4	~~	AF1	0.228	-0.037	-0.037	-0.029	-0.029
112	AS4	~~	AF2	1.036	-0.079	-0.079	-0.062	-0.062
113	AS4	~~	AF3	0.009	-0.008	-0.008	-0.006	-0.006
114	AS4	~~	AF4	0.261	-0.040	-0.040	-0.031	-0.031
115	AS4	~~	MI1	0.803	0.042	0.042	0.055	0.055
116	AS4	~~	MI2	0.377	0.026	0.026	0.038	0.038
117	AS4	~~	MI3	0.001	0.001	0.001	0.002	0.002
118	AS4	~~	MI4	0.242	-0.023	-0.023	-0.030	-0.030
119	AS4	~~	MI5	0.225	0.019	0.019	0.029	0.029
120	AS4	~~	AUA1	1.379	-0.062	-0.062	-0.072	-0.072
121	AS4	~~	AUA2	1.298	-0.057	-0.057	-0.070	-0.070
122	AS4	~~	AUA3	1.460	-0.065	-0.065	-0.074	-0.074
123	AS4	~~	AUA4	0.752	-0.047	-0.047	-0.053	-0.053
124	AS5	~~	AS6	0.162	-0.020	-0.020	-0.026	-0.026
125	AS5	~~	AS7	0.761	-0.042	-0.042	-0.056	-0.056
126	AS5	~~	AS8	0.016	0.007	0.007	0.008	0.008
127	AS5	~~	AS9	0.702	0.054	0.054	0.053	0.053
128	AS5	~~	AF1	0.502	0.055	0.055	0.043	0.043
129	AS5	~~	AF2	4.189	0.160	0.160	0.123	0.123
130	AS5	~~	AF3	1.241	0.093	0.093	0.067	0.067
131	AS5	~~	AF4	1.070	0.082	0.082	0.063	0.063
132	AS5	~~	MI1	0.163	-0.019	-0.019	-0.024	-0.024
133	AS5	~~	MI2	0.367	-0.026	-0.026	-0.036	-0.036
134	AS5	~~	MI3	0.112	-0.015	-0.015	-0.020	-0.020
135	AS5	~~	MI4	0.049	-0.010	-0.010	-0.013	-0.013
136	AS5	~~	MI5	0.802	-0.035	-0.035	-0.054	-0.054
137	AS5	~~	AUA1	0.403	0.034	0.034	0.038	0.038
138	AS5	~~	AUA2	1.014	0.051	0.051	0.061	0.061
139	AS5	~~	AUA3	0.221	-0.026	-0.026	-0.028	-0.028
140	AS5	~~	AUA4	0.099	-0.017	-0.017	-0.019	-0.019
141	AS6	~~	AS7	0.404	0.021	0.021	0.041	0.041
142	AS6	~~	AS8	0.417	0.026	0.026	0.041	0.041
143	AS6	~~	AS9	0.082	-0.013	-0.013	-0.018	-0.018
144	AS6	~~	AF1	4.431	-0.114	-0.114	-0.126	-0.126
145	AS6	~~	AF2	2.093	-0.078	-0.078	-0.087	-0.087
146	AS6	~~	AF3	2.897	-0.098	-0.098	-0.102	-0.102
147	AS6	~~	AF4	1.805	-0.073	-0.073	-0.081	-0.081
148	AS6	~~	MI1	0.269	0.017	0.017	0.031	0.031
149	AS6	~~	MI2	1.471	0.036	0.036	0.073	0.073
150	AS6	~~	MI3	0.475	0.022	0.022	0.041	0.041
151	AS6	~~	MI4	0.028	0.005	0.005	0.010	0.010
152	AS6	~~	MI5	0.146	0.010	0.010	0.023	0.023

153	AS6	~~	AUA1	4.206	0.075	0.075	0.123	0.123
154	AS6	~~	AUA2	1.793	0.047	0.047	0.080	0.080
155	AS6	~~	AUA3	2.997	0.065	0.065	0.104	0.104
156	AS6	~~	AUA4	0.163	0.015	0.015	0.024	0.024
157	AS7	~~	AS8	0.001	0.001	0.001	0.002	0.002
158	AS7	~~	AS9	0.994	-0.044	-0.044	-0.063	-0.063
159	AS7	~~	AF1	3.915	-0.105	-0.105	-0.119	-0.119
160	AS7	~~	AF2	0.118	-0.018	-0.018	-0.021	-0.021
161	AS7	~~	AF3	0.591	-0.044	-0.044	-0.046	-0.046
162	AS7	~~	AF4	2.957	-0.092	-0.092	-0.104	-0.104
163	AS7	~~	MI1	2.310	0.049	0.049	0.092	0.092
164	AS7	~~	MI2	3.887	0.058	0.058	0.119	0.119
165	AS7	~~	MI3	1.173	0.034	0.034	0.065	0.065
166	AS7	~~	MI4	0.692	0.027	0.027	0.050	0.050
167	AS7	~~	MI5	0.052	0.006	0.006	0.014	0.014
168	AS7	~~	AUA1	4.995	-0.080	-0.080	-0.135	-0.135
169	AS7	~~	AUA2	3.209	-0.062	-0.062	-0.108	-0.108
170	AS7	~~	AUA3	2.546	-0.059	-0.059	-0.096	-0.096
171	AS7	~~	AUA4	1.252	-0.041	-0.041	-0.067	-0.067
172	AS8	~~	AS9	0.426	-0.035	-0.035	-0.041	-0.041
173	AS8	~~	AF1	1.077	-0.068	-0.068	-0.062	-0.062
174	AS8	~~	AF2	0.165	0.026	0.026	0.024	0.024
175	AS8	~~	AF3	0.316	-0.039	-0.039	-0.034	-0.034
176	AS8	~~	AF4	0.022	-0.010	-0.010	-0.009	-0.009
177	AS8	~~	MI1	1.073	0.041	0.041	0.062	0.062
178	AS8	~~	MI2	1.180	0.039	0.039	0.065	0.065
179	AS8	~~	MI3	1.364	0.044	0.044	0.070	0.070
180	AS8	~~	MI4	1.650	0.050	0.050	0.077	0.077
181	AS8	~~	MI5	3.795	0.064	0.064	0.117	0.117
182	AS8	~~	AUA1	0.421	-0.029	-0.029	-0.039	-0.039
183	AS8	~~	AUA2	0.474	-0.029	-0.029	-0.041	-0.041
184	AS8	~~	AUA3	0.001	-0.002	-0.002	-0.002	-0.002
185	AS8	~~	AUA4	0.038	0.009	0.009	0.012	0.012
186	AS9	~~	AF1	6.807	0.186	0.186	0.155	0.155
187	AS9	~~	AF2	3.933	0.141	0.141	0.118	0.118
188	AS9	~~	AF3	2.468	0.119	0.119	0.093	0.093
189	AS9	~~	AF4	5.458	0.168	0.168	0.139	0.139
190	AS9	~~	MI1	0.761	-0.038	-0.038	-0.052	-0.052
191	AS9	~~	MI2	0.122	-0.014	-0.014	-0.021	-0.021
192	AS9	~~	MI3	0.642	-0.033	-0.033	-0.048	-0.048
193	AS9	~~	MI4	0.007	0.003	0.003	0.005	0.005
194	AS9	~~	MI5	0.082	-0.010	-0.010	-0.017	-0.017
195	AS9	~~	AUA1	2.237	0.072	0.072	0.089	0.089
196	AS9	~~	AUA2	0.636	0.037	0.037	0.047	0.047
197	AS9	~~	AUA3	1.925	0.069	0.069	0.082	0.082
198	AS9	~~	AUA4	0.506	0.035	0.035	0.042	0.042
199	AF1	~~	AF2	94.473	0.848	0.848	0.557	0.557
200	AF1	~~	AF3	102.426	0.942	0.942	0.580	0.580

201	AF1	~~	AF4	94.401	0.856	0.856	0.559	0.559
202	AF1	~~	MI1	0.161	-0.021	-0.021	-0.023	-0.023
203	AF1	~~	MI2	10.089	-0.153	-0.153	-0.182	-0.182
204	AF1	~~	MI3	3.733	-0.098	-0.098	-0.111	-0.111
205	AF1	~~	MI4	0.662	-0.043	-0.043	-0.047	-0.047
206	AF1	~~	MI5	0.231	0.021	0.021	0.028	0.028
207	AF1	~~	AUA1	0.002	0.003	0.003	0.003	0.003
208	AF1	~~	AUA2	0.471	0.039	0.039	0.039	0.039
209	AF1	~~	AUA3	1.314	0.070	0.070	0.066	0.066
210	AF1	~~	AUA4	6.586	0.157	0.157	0.147	0.147
211	AF2	~~	AF3	79.723	0.830	0.830	0.512	0.512
212	AF2	~~	AF4	52.153	0.635	0.635	0.415	0.415
213	AF2	~~	MI1	5.830	0.128	0.128	0.138	0.138
214	AF2	~~	MI2	0.749	-0.042	-0.042	-0.050	-0.050
215	AF2	~~	MI3	0.159	0.020	0.020	0.023	0.023
216	AF2	~~	MI4	0.881	0.049	0.049	0.054	0.054
217	AF2	~~	MI5	0.120	0.015	0.015	0.020	0.020
218	AF2	~~	AUA1	0.010	0.006	0.006	0.006	0.006
219	AF2	~~	AUA2	0.018	0.008	0.008	0.008	0.008
220	AF2	~~	AUA3	1.629	0.077	0.077	0.073	0.073
221	AF2	~~	AUA4	4.961	0.136	0.136	0.128	0.128
222	AF3	~~	AF4	69.665	0.783	0.783	0.480	0.480
223	AF3	~~	MI1	10.638	0.184	0.184	0.187	0.187
224	AF3	~~	MI2	0.070	-0.014	-0.014	-0.015	-0.015
225	AF3	~~	MI3	0.065	0.014	0.014	0.015	0.015
226	AF3	~~	MI4	0.582	0.043	0.043	0.044	0.044
227	AF3	~~	MI5	0.275	0.025	0.025	0.030	0.030
228	AF3	~~	AUA1	2.726	-0.104	-0.104	-0.095	-0.095
229	AF3	~~	AUA2	1.211	-0.066	-0.066	-0.063	-0.063
230	AF3	~~	AUA3	0.001	0.002	0.002	0.002	0.002
231	AF3	~~	AUA4	2.048	0.093	0.093	0.082	0.082
232	AF4	~~	MI1	1.002	-0.054	-0.054	-0.058	-0.058
233	AF4	~~	MI2	11.306	-0.163	-0.163	-0.193	-0.193
234	AF4	~~	MI3	6.110	-0.127	-0.127	-0.142	-0.142
235	AF4	~~	MI4	0.865	-0.049	-0.049	-0.054	-0.054
236	AF4	~~	MI5	0.039	-0.009	-0.009	-0.011	-0.011
237	AF4	~~	AUA1	0.036	-0.011	-0.011	-0.011	-0.011
238	AF4	~~	AUA2	0.142	-0.021	-0.021	-0.022	-0.022
239	AF4	~~	AUA3	0.024	-0.009	-0.009	-0.009	-0.009
240	AF4	~~	AUA4	1.002	0.062	0.062	0.058	0.058
241	MI1	~~	MI2	33.492	0.169	0.169	0.332	0.332
242	MI1	~~	MI3	25.185	0.155	0.155	0.288	0.288
243	MI1	~~	MI4	28.488	0.171	0.171	0.306	0.306
244	MI1	~~	MI5	19.558	0.118	0.118	0.254	0.254
245	MI1	~~	AUA1	2.792	-0.060	-0.060	-0.096	-0.096
246	MI1	~~	AUA2	1.038	-0.035	-0.035	-0.058	-0.058
247	MI1	~~	AUA3	0.227	0.018	0.018	0.027	0.027
248	MI1	~~	AUA4	0.003	0.002	0.002	0.003	0.003

249	MI2	~~	MI3	26.123	0.143	0.143	0.293	0.293
250	MI2	~~	MI4	22.744	0.138	0.138	0.274	0.274
251	MI2	~~	MI5	7.663	0.067	0.067	0.159	0.159
252	MI2	~~	AUA1	2.445	0.051	0.051	0.090	0.090
253	MI2	~~	AUA2	2.330	0.047	0.047	0.088	0.088
254	MI2	~~	AUA3	3.257	0.060	0.060	0.104	0.104
255	MI2	~~	AUA4	6.971	0.089	0.089	0.151	0.151
256	MI3	~~	MI4	15.904	0.122	0.122	0.229	0.229
257	MI3	~~	MI5	11.019	0.085	0.085	0.191	0.191
258	MI3	~~	AUA1	2.769	0.057	0.057	0.095	0.095
259	MI3	~~	AUA2	1.765	0.044	0.044	0.076	0.076
260	MI3	~~	AUA3	5.733	0.085	0.085	0.137	0.137
261	MI3	~~	AUA4	4.907	0.079	0.079	0.127	0.127
262	MI4	~~	MI5	5.073	0.059	0.059	0.129	0.129
263	MI4	~~	AUA1	0.338	0.021	0.021	0.033	0.033
264	MI4	~~	AUA2	0.501	-0.024	-0.024	-0.041	-0.041
265	MI4	~~	AUA3	0.205	0.017	0.017	0.026	0.026
266	MI4	~~	AUA4	3.874	0.072	0.072	0.113	0.113
267	MI5	~~	AUA1	0.052	-0.007	-0.007	-0.013	-0.013
268	MI5	~~	AUA2	1.533	-0.035	-0.035	-0.071	-0.071
269	MI5	~~	AUA3	0.457	0.021	0.021	0.039	0.039
270	MI5	~~	AUA4	0.150	-0.012	-0.012	-0.022	-0.022
271	AUA1	~~	AUA2	116.164	0.411	0.411	0.618	0.618
272	AUA1	~~	AUA3	53.048	0.299	0.299	0.418	0.418
273	AUA1	~~	AUA4	44.958	0.276	0.276	0.385	0.385
274	AUA2	~~	AUA3	36.536	0.238	0.238	0.347	0.347
275	AUA2	~~	AUA4	44.096	0.262	0.262	0.381	0.381
276	AUA3	~~	AUA4	25.874	0.216	0.216	0.292	0.292

Kline recommends evaluating the “cor” residuals. In our output, these seem to be the “cor.bollen” and are near the bottom. He recommends that residuals $> .10$ may be possible sources for misfit. He also indicated that patterns may be helpful (is there an item that has consistently high residuals).

Kline also cautions that there is no dependable or trustworthy connection between the size of the residual and the type or degree of model misspecification.

My first read of our results is that the items in the AS# factor were well-defined. I suspect that a multi-factor solution will improve the fit.

The *semTable* package can help us extract the values into a .csv file which will make it easier to create an APA style table. It takes some tinkering...

```
#library(semTable)
```

```
grmsAAW1table <- semTable::semTable(grmsAAW1fit, columns = c("eststars", "se", "p"), columnLabels = c("Estimate", "Standard Error", "P-value"))
```

```
#Can change "print.results" to TRUE if you want to see the (messy) output in the .rmd file (it
```

Cool, but it doesn't contain standardized estimates. One way to get them is to create an updated model with the standardized output:

```
grmsAAW1stdzd <- update (grmsAAW1fit, std.lv = TRUE, std.ov = TRUE, meanstructure = TRUE)
```

Now request both models in the semTable

```
grmsAAW1table <- semTable::semTable(list ("Ordinary" = grmsAAW1fit, "Standardized" = grmsAAW1s
```

```
#Can change "print.results" to TRUE if you want to see the (messy) output in the .rmd file (it
```

Troubleshooting If, while working with this function you get the error, “Error in file(file, ifelse(append,”a”, “w”)) : cannot open the connection” it’s because the .csv file that received your table is still open. R is just trying to write over it. A similar error happens when knitting, or updating any spreadsheet or word document.

APA Style Results from the Unidimensional model

Model testing. To evaluate the models we, we used confirmatory factor analysis (CFA) in the R package, *lavaan* (v.0-6.9) with maximum likelihood estimation. Our sample size was 304. We selected fit criteria for their capacity to assess different aspects of the statistical analysis. As is common among SEM researchers, we reported the Chi-square goodness of fit (χ^2). This evaluates the discrepancy between the unrestricted sample matrix and the restricted covariance matrix. Although the associated p value indicates adequate fit when the value is non-significant, it is widely recognized that large sample size can result in a statistically significant p value (?). The comparative fit index (CFI) is an incremental index, comparing the hypothesized model at least .90 and perhaps higher than .95 (?). The root mean square error of approximation (RMSEA) takes into account the error of approximation in the population and expresses it per degree of freedom. As such, the fit indicator considers the complexity of the model. Ideal values are equal to or less than .05, values less than .08 represent reasonable fit, and values between .08 and .10 represent mediocre fit. The standardized root mean residual is a standardized measure of the mean absolute covariance residual – the overall difference between the observed and predicted correlations. Values greater than .10 may indicate poor fit and inspection of residuals is then advised. Kline (?) advised researchers to be cautious when using these criteria as strict cut-offs. Elements such as sample size and model complexity should be considered when evaluating fit.

Our first model was unidimensional where each of the 24 items loaded onto a single factor representing overall, gendered racial microaggressions towards Asian American women. The Chi-square index was statistically significant ($\chi^2(209) = 1004.136, p < .001$) indicating likely misfit. The CFI value of .58 indicated poor fit. The RMSEA = .11 (90% CI [.11, .20]) suggested serious problems. The SRMR value of .12 exceeded the warning criteria of .10. The AIC and BIC values were 17755.028 and 17918.577, respectively, and will become useful in comparing subsequent models.

10.5.2 Modeling the GRMSAAW as a First-Order, 4-factor model

10.5.2.1 Specifying and Running the Model

As we know from the article, the GRMSAAW has four subscales. Therefore, let's respecify it as a first-order, four-factor model, allowing the factors to correlate.

Model identification is always a consideration. In a multi-dimensional model, each factor requires a minimum of two items/indicators. Our shortest scales are the AF and AUA scales, each with 4 items, so we are OK!

We will be using the `cfa()` function in lavaan. When we do this, it does three things by default:

1. The factor loading of the first indicator of a latent variable is fixed to 1.0; this fixes the scale of the LV
 2. Residual variances are added automatically.
 3. All exogenous LVs are correlated.
- If you are specifying an orthogonal model you will want to switch off the default behavior by including the statement: `auto.cov.lv.x=FALSE`

```
grmsAAW4mod <- 'AS =~ AS1 + AS2 + AS3 + AS4 + AS5 + AS6 + AS7 + AS8 + AS9
                  AF =~ AF1 + AF2 + AF3 + AF4
                  MI =~ MI1 + MI2 + MI3 + MI4 + MI5
                  AUA =~ AUA1 + AUA2 + AUA3 + AUA4'
```

```
[1] "AS =~ AS1 + AS2 + AS3 + AS4 + AS5 + AS6 + AS7 + AS8 + AS9\n" AF =~ AF1 + AF2 +
```

#This code is identical to the one we ran above -- in this code below, we are just clearly specifying the model

```
grmsAAW4mod <- 'AS =~ AS1 + AS2 + AS3 + AS4 + AS5 + AS6 + AS7 + AS8 + AS9
                  AF =~ AF1 + AF2 + AF3 + AF4
                  MI =~ MI1 + MI2 + MI3 + MI4 + MI5
                  AUA =~ AUA1 + AUA2 + AUA3 + AUA4'

#covariances in our oblique model
AS ~~ AF
AS ~~ MI
AS ~~ AUA
AF ~~ MI
AF ~~ AUA
MI ~~ AUA
```

```
grmsAAW4fit <- lavaan::cfa (grmsAAW4mod, data = dfGRMSAAW)
lavaan::summary(grmsAAW4fit, fit.measures=TRUE, standardized=TRUE, rsquare = TRUE)
```

lavaan 0.6-9 ended normally after 37 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	50
Number of observations	304

Model Test User Model:

Test statistic	220.858
Degrees of freedom	203
P-value (Chi-square)	0.186

Model Test Baseline Model:

Test statistic	2114.899
Degrees of freedom	231
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.991
Tucker-Lewis Index (TLI)	0.989

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-8441.875
Loglikelihood unrestricted model (H1)	-8331.446
Akaike (AIC)	16983.750
Bayesian (BIC)	17169.602
Sample-size adjusted Bayesian (BIC)	17011.027

Root Mean Square Error of Approximation:

RMSEA	0.017
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.031
P-value RMSEA <= 0.05	1.000

Standardized Root Mean Square Residual:

SRMR	0.058
------	-------

Parameter Estimates:

Standard errors	Standard
-----------------	----------

	Information	Expected				
	Information saturated (h1) model	Structured				
Latent Variables:						
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
AS =~						
AS1	1.000				0.971	0.799
AS2	0.626	0.047	13.211	0.000	0.608	0.723
AS3	0.912	0.070	12.953	0.000	0.886	0.711
AS4	1.084	0.083	13.047	0.000	1.053	0.716
AS5	0.966	0.081	11.955	0.000	0.938	0.665
AS6	0.626	0.055	11.389	0.000	0.608	0.638
AS7	0.658	0.055	12.006	0.000	0.639	0.667
AS8	0.755	0.066	11.393	0.000	0.734	0.638
AS9	0.735	0.071	10.427	0.000	0.714	0.591
AF =~						
AF1	1.000				1.014	0.821
AF2	0.824	0.075	10.935	0.000	0.836	0.678
AF3	0.932	0.081	11.487	0.000	0.945	0.719
AF4	0.802	0.077	10.369	0.000	0.814	0.641
MI =~						
MI1	1.000				0.449	0.599
MI2	0.848	0.145	5.847	0.000	0.381	0.561
MI3	0.812	0.145	5.595	0.000	0.365	0.506
MI4	0.797	0.147	5.439	0.000	0.358	0.481
MI5	0.491	0.112	4.395	0.000	0.220	0.353
AUA =~						
AUA1	1.000				0.682	0.818
AUA2	0.875	0.089	9.786	0.000	0.597	0.748
AUA3	0.634	0.083	7.619	0.000	0.432	0.503
AUA4	0.628	0.083	7.537	0.000	0.429	0.497
Covariances:						
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
AS ~~						
AF	0.017	0.066	0.262	0.794	0.017	0.017
MI	0.036	0.033	1.072	0.284	0.082	0.082
AUA	0.023	0.045	0.520	0.603	0.035	0.035
AF ~~						
MI	-0.028	0.036	-0.764	0.445	-0.060	-0.060
AUA	0.012	0.049	0.236	0.814	0.017	0.017
MI ~~						
AUA	0.024	0.025	0.960	0.337	0.077	0.077
Variances:						
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.AS1	0.536	0.055	9.727	0.000	0.536	0.362
.AS2	0.338	0.032	10.713	0.000	0.338	0.477

.AS3	0.767	0.071	10.819	0.000	0.767	0.494
.AS4	1.057	0.098	10.781	0.000	1.057	0.488
.AS5	1.110	0.099	11.162	0.000	1.110	0.558
.AS6	0.539	0.048	11.319	0.000	0.539	0.593
.AS7	0.508	0.046	11.147	0.000	0.508	0.554
.AS8	0.783	0.069	11.318	0.000	0.783	0.593
.AS9	0.950	0.082	11.540	0.000	0.950	0.651
.AF1	0.496	0.076	6.550	0.000	0.496	0.326
.AF2	0.821	0.083	9.934	0.000	0.821	0.540
.AF3	0.836	0.090	9.249	0.000	0.836	0.483
.AF4	0.951	0.091	10.395	0.000	0.951	0.589
.MI1	0.361	0.043	8.381	0.000	0.361	0.641
.MI2	0.315	0.035	9.074	0.000	0.315	0.685
.MI3	0.386	0.039	9.915	0.000	0.386	0.744
.MI4	0.427	0.042	10.233	0.000	0.427	0.769
.MI5	0.342	0.030	11.352	0.000	0.342	0.875
.AUA1	0.230	0.044	5.228	0.000	0.230	0.331
.AUA2	0.281	0.038	7.300	0.000	0.281	0.441
.AUA3	0.551	0.049	11.221	0.000	0.551	0.747
.AUA4	0.560	0.050	11.257	0.000	0.560	0.753
AS	0.944	0.117	8.073	0.000	1.000	1.000
AF	1.028	0.133	7.709	0.000	1.000	1.000
MI	0.202	0.047	4.279	0.000	1.000	1.000
AUA	0.465	0.067	6.992	0.000	1.000	1.000

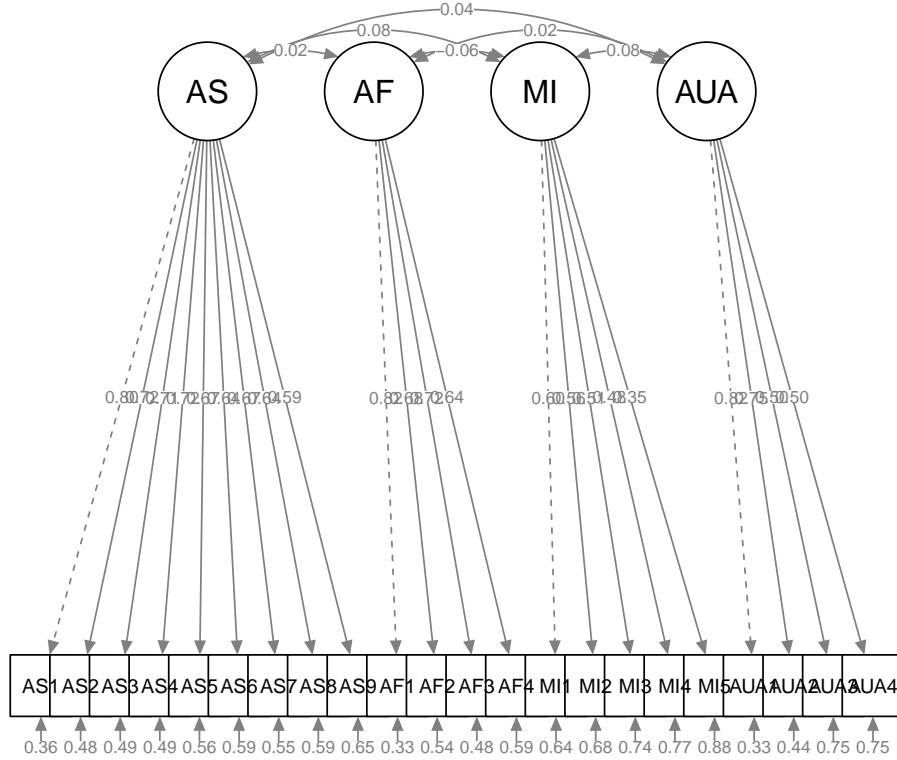
R-Square:

Estimate

AS1	0.638
AS2	0.523
AS3	0.506
AS4	0.512
AS5	0.442
AS6	0.407
AS7	0.446
AS8	0.407
AS9	0.349
AF1	0.674
AF2	0.460
AF3	0.517
AF4	0.411
MI1	0.359
MI2	0.315
MI3	0.256
MI4	0.231
MI5	0.125
AUA1	0.669
AUA2	0.559
AUA3	0.253

AUA4 0.247

```
semPlot::semPaths(grmsAAW4fit, layout = "tree", style = "lisrel", what = "col", whatLabels = "
```



The table

First an update to get the standardized results:

```
grmsAAW4stdzd <- update (grmsAAW4fit, std.lv = TRUE, std.ov = TRUE, meanstructure = TRUE)
```

```
grmsAAW4table <- semTable::semTable(list ("Ordinary" = grmsAAW4fit, "Standardized" = grmsAAW4stdzd))
#Can change "print.results" to TRUE if you want to see the (messy) output in the .rmd file (it
```

10.5.2.2 Interpretation

Our model converged, normally, with 37 iterations. The estimator was the lavaan default, maximum likelihood (ML). All 304 cases were used in the analysis.

I mapped our pattern coefficients into the GRMSAAW tables. Most pattern coefficients are strong, significant, and stably connected to their respective factor. The lowest factor loading was .220 (MI5).

A multidimensional factor structure also includes correlations/covariances between factors. We can see that the correlation (look at the Std.all column) shows the following correlations (none are statistically significant):

AF & AS: 0.017 AF & MI: -0.060 AF & AUA: 0.035 AS & MI: 0.082 AS & AUA: 0.035 MI & AUA: 0.077

For our multi-dimensional GRMSAAW4 CFA $\chi^2(203) = 220.858, p < .186$, this significant value is not what we want because it says that our specified model is not statistically significantly different than the covariances in the model. That is, our more parsimonious model is a reasonable explanation (simplification).

The CFI and TLI compare user (the 4-dimensional model we specified) and baseline (where no relations would exist between variables) models. These values will always be close together because the only difference is that the TLI imposes a penalty for any model complexity. The CFI seems to be more commonly reported and its value is 0.991. This means our model performed 99% better than a model with no relations. It well-exceeds the traditional cutoffs of .90 and the more strict cutoff of .95. The TLI imposes a greater relative penalty for model complexity, consequently it is a smidge lower at .989.

The RMSEA one of the *badness of fit*, absolute fit index, statistics where a value of 0.00 is the best fit. Our RMSEA = 0.017 (90%CI[.000, .031]). As a quick reminder, an there is general consensus that $RMSEA \leq .05$ is desired and an $RMSEA \geq .10$ points to serious problems. We watch the upper bound of the confidence interval to see that it isn't sneaking into the danger zone.

The SRMR is another absolute, *badness of fit* index (i.e., perfect model fit is when the value = 0.00 and increasingly higher values indicate the “badness”). The SRMR is a measure of the mean absolute covariance residual. Standardizing the value facilitates interpretation. Poor fit is indicated when $SRMR \geq .10$. The GRMSAAW SRMR = .058.

Recall, Hu and Bentler's **combination rule** (which is somewhat contested) suggested that the SRMR be interpreted along with the CFI such that: $CFI \geq .95$ and $SRMR \leq .08$.

For our unidimensional GRMSAAW CFA, the CFI = .99 and the SRMR = .058. We are close!

The AIC and BIC utilize an information theory approach to data analysis by combining statistical estimation and model selection into a single framework. The BIC augments the AIC by taking sample size into consideration. We can compare the values from our current model to the former one. The model with the smallest value of the predictive fit index is chosen as the one that is most likely to replicate. It means that this model has relatively better fit and fewer free parameters than competing models. We will do that in the next section.

Before moving to model comparison, it is a good practice for locating sources of misfit (we look for relatively large values) is to inspect the residuals, so let's do that.

```
lavaan::fitted(grmsAAW4fit)
```

```
$cov
```

AS1	AS2	AS3	AS4	AS5	AS6	AS7	AS8	AS9	AF1
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

AS1	1.480
AS2	0.591 0.708
AS3	0.861 0.539 1.552
AS4	1.023 0.641 0.933 2.166
AS5	0.911 0.571 0.832 0.988 1.990
AS6	0.591 0.370 0.539 0.641 0.571 0.909
AS7	0.620 0.389 0.566 0.673 0.599 0.389 0.916
AS8	0.713 0.446 0.650 0.773 0.688 0.446 0.469 1.321
AS9	0.694 0.435 0.633 0.752 0.670 0.435 0.456 0.524 1.461
AF1	0.017 0.011 0.016 0.019 0.017 0.011 0.011 0.013 0.013 1.525
AF2	0.014 0.009 0.013 0.015 0.014 0.009 0.009 0.011 0.010 0.847
AF3	0.016 0.010 0.015 0.017 0.016 0.010 0.011 0.012 0.012 0.959
AF4	0.014 0.009 0.013 0.015 0.013 0.009 0.009 0.010 0.010 0.825
MI1	0.036 0.022 0.033 0.039 0.034 0.022 0.023 0.027 0.026 -0.028
MI2	0.030 0.019 0.028 0.033 0.029 0.019 0.020 0.023 0.022 -0.023
MI3	0.029 0.018 0.026 0.031 0.028 0.018 0.019 0.022 0.021 -0.022
MI4	0.028 0.018 0.026 0.031 0.027 0.018 0.019 0.021 0.021 -0.022
MI5	0.017 0.011 0.016 0.019 0.017 0.011 0.012 0.013 0.013 -0.014
AUA1	0.023 0.015 0.021 0.025 0.023 0.015 0.015 0.018 0.017 0.012
AUA2	0.020 0.013 0.019 0.022 0.020 0.013 0.013 0.015 0.015 0.010
AUA3	0.015 0.009 0.014 0.016 0.014 0.009 0.010 0.011 0.011 0.007
AUA4	0.015 0.009 0.013 0.016 0.014 0.009 0.010 0.011 0.011 0.007
	AF2 AF3 AF4 MI1 MI2 MI3 MI4 MI5 AUA1 AUA2
AS1	
AS2	
AS3	
AS4	
AS5	
AS6	
AS7	
AS8	
AS9	
AF1	
AF2	1.519
AF3	0.790 1.729
AF4	0.680 0.769 1.613
MI1	-0.023 -0.026 -0.022 0.563
MI2	-0.019 -0.022 -0.019 0.171 0.460
MI3	-0.018 -0.021 -0.018 0.164 0.139 0.519
MI4	-0.018 -0.020 -0.018 0.161 0.136 0.131 0.555
MI5	-0.011 -0.013 -0.011 0.099 0.084 0.080 0.079 0.390
AUA1	0.010 0.011 0.009 0.024 0.020 0.019 0.019 0.012 0.696
AUA2	0.008 0.009 0.008 0.021 0.018 0.017 0.017 0.010 0.407 0.637
AUA3	0.006 0.007 0.006 0.015 0.013 0.012 0.012 0.007 0.295 0.258
AUA4	0.006 0.007 0.006 0.015 0.013 0.012 0.012 0.007 0.292 0.256
	AUA3 AUA4
AS1	
AS2	

```

AS3
AS4
AS5
AS6
AS7
AS8
AS9
AF1
AF2
AF3
AF4
MI1
MI2
MI3
MI4
MI5
AUA1
AUA2
AUA3 0.738
AUA4 0.185 0.743

```

```

#lavaan::residuals(grmsAAW4fit, type = "raw")
#lavaan::residuals(grmsAAW4fit, type = "standardized")
lavaan::residuals(grmsAAW4fit, type = "cor")

```

```

$type
[1] "cor.bollen"

$cov
   AS1    AS2    AS3    AS4    AS5    AS6    AS7    AS8    AS9    AF1
AS1  0.000
AS2  0.019  0.000
AS3 -0.010 -0.023  0.000
AS4 -0.003  0.006  0.002  0.000
AS5 -0.001  0.013  0.000 -0.004  0.000
AS6 -0.017 -0.004  0.014 -0.001 -0.012  0.000
AS7 -0.005 -0.009  0.029  0.014 -0.026  0.017  0.000
AS8  0.006 -0.017 -0.008  0.005  0.005  0.021  0.001  0.000
AS9  0.016  0.006  0.015 -0.024  0.028 -0.008 -0.031 -0.018  0.000
AF1 -0.005 -0.041 -0.062 -0.049  0.000 -0.117 -0.110 -0.072  0.090  0.000
AF2 -0.009 -0.061 -0.041 -0.043  0.079 -0.066 -0.019  0.013  0.084  0.000
AF3 -0.013 -0.062 -0.037 -0.023  0.028 -0.089 -0.049 -0.041  0.054 -0.010
AF4  0.212  0.132  0.132  0.127  0.176  0.074  0.067  0.124  0.222  0.010
MI1 -0.091 -0.055 -0.011  0.015 -0.034  0.005  0.044  0.027 -0.055  0.006
MI2 -0.081 -0.083 -0.018  0.004 -0.042  0.035  0.063  0.029 -0.031 -0.155
MI3 -0.065 -0.035  0.034 -0.002 -0.017  0.026  0.041  0.047 -0.038 -0.087
MI4 -0.036  0.022  0.043  0.004  0.012  0.027  0.055  0.074  0.022 -0.025

```



```
AUA2
AUA3 0.000
AUA4 0.042 0.000
```

```
#lavaan::modindices(grmsAAW4fit)
```

10.6 Model Comparison

We evaluated two models (i.e., a unidimensional model and four-factor correlated model), which one is better? While, we have the narrative comparison (and would create a table with the comparisons) where the four-dimensional fit values ($CFI = 0.99$, $RMSEA = 0.02$ (90%CI[.00, .03]), and $SRMR = .058$) outperformed the unidimensional ones ($CFI = 0.58$, $RMSEA = .11$ (90%CI[.11, .20])), and $SRMR = .12$). We can formally compare them with statistical comparisons.

Easy are AIC and BIC comparisons where “smaller value wins.”

AIC GRMSAAW1: 17755.028 AIC GRMSAAW4: 16983.750

BIC GRMSAAW1: 17918.577 BIC GRMSAAW4: 17169.602

In both cases, the smaller values are for the more complex, 4-dimensional model. The interpretation is that the model with the smaller AIC/BIC values is most likely to replicate.

Additionally, the **chi-square difference test**, χ^2_D can be used to compare nested models. Single-factor CFA models are nested under any other CFA model with two or more factors *for the same indicators*. This is because a one-factor model is a restricted version of any model with multiple factors. Our unidimensional GRMSAAW was nested under the 4-factor GRMSAAW model.

To calculate the chi-square difference test, we first grab the chi-square test values:

GRMSAAW1: $\chi^2(209) = 1004.136, p < .001$ GRMSAAW4: $\chi^2(203) = 220.858, p < .186$

Given both sets of results we calculate: $\chi^2(6) = 783.278, p < .05$ and determine that the two models are statistically significantly different. Given that the fit statistics are better for the single-order, correlated, four-factor model, we prefer that one.

How did I do that?

- Subtract the df
- Subtract the chi-square values
- Use a chi-square difference table to look up the chi-square critical value for a 6 df test
 - <https://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm>, or
 - use this code to look it up `qchisq(p, df, lower.tail=FALSE)`
 - the critical value for our test is 12.592
- We conclude that the two models are statistically significantly different; our 4-factor model is preferred.

```
209-203 #subtract df
```

```
[1] 6
```

```
1004.136 - 220.858 #subtract chi-square values
```

```
[1] 783.278
```

```
qchisq(.05, 6, lower.tail=FALSE)
```

```
[1] 12.59159
```

Of course, there is a function for something this easy:

```
lavaan:::lavTestLRT(grmsAAW1fit, grmsAAW4fit)
```

Chi-Squared Difference Test

	Df	AIC	BIC	Chisq	Pr(>Chisq)						
grmsAAW4fit	203	16984	17170	220.86							
grmsAAW1fit	209	17755	17919	1004.14	6 < 0.0000000000000022						
grmsAAW4fit				783.28							
grmsAAW1fit	***										

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

And we get the same result: $\chi^2(6) = 783.28, p < .001$

And now a table with estimates and fit indices from both models.

```
#All the requested data gets transferred over, but the pattern coefficients do not end up side by side
grmsAAWtables <- semTable::semTable(list("Single Dimension" = grmsAAW1fit, "Multidimensional" = grmsAAW4fit))
#Can change "print.results" to TRUE if you want to see the (messy) output in the .rmd file (it will be messy)
```

Let's try it with standardized output:

```
GRMSAAWstdzd <- semTable::semTable(list("Single Dimension" = grmsAAW1stdzd, "Multidimensional" = grmsAAW4stdzd))
#Can change "print.results" to TRUE if you want to see the (messy) output in the .rmd file (it will be messy)
```

(Placeholder, more to come!) APA Results Section:

Model testing. To evaluate the models we, we used confirmatory factor analysis (CFA) in the R package, *lavaan* (v.0.6-9) with maximum likelihood estimation. Our sample size was 304. We selected fit criteria for their capacity to assess different aspects of the statistical analysis. As is

common among SEM researchers, we reported the Chi-square goodness of fit (χ^2). This evaluates the discrepancy between the unrestricted sample matrix and the restricted covariance matrix. Although the associated p value indicates adequate fit when the value is non-significant, it is widely recognized that large sample size can result in a statistically significant p value (?). The comparative fit index (CFI) is an incremental index, comparing the hypothesized model at least .90 and perhaps higher than .95 (?). The root mean square error of approximation (RMSEA) takes into account the error of approximation in the population and expresses it per degree of freedom. As such, the fit indicator considers the complexity of the model. Ideal values are equal to or less than .05, values less than .08 represent reasonable fit, and values between .08 and .10 represent mediocre fit. The standardized root mean residual is a standardized measure of the mean absolute covariance residual – the overall difference between the observed and predicted correlations. Values greater than .10 may indicate poor fit and inspection of residuals is then advised. Because we were interested in comparing nested models we used the Chi-square difference test where a significant chi-square indicates statistically significant differences in models. Additionally we used Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC) that take model complexity and sample size into consideration. Models with lower values on each are considered to be superior. Kline (2016) advised researchers to be cautious when using these criteria as strict cut-offs. Elements such as sample size and model complexity should be considered when evaluating fit. Table 1 provides a side-by-side comparison of the resulting parameter estimates and fit statistics; Figures 1 and 2 provide a graphic representation of the models tested.

Our first model was unidimensional where each of the 22 items loaded onto a single factor representing overall gendered racial microaggressions for Asian American women. Standardized pattern coefficients ranged between -.030 and .799 and were not all statistically significant. The Chi-square index was statistically significant ($\chi^2(209) = 1004.136, p < .001$) indicating likely misfit. The CFI value of .58 indicated poor fit. The RMSEA = .11 (90% CI [.11, .20]) suggested serious problems. The SRMR value of .12 exceeded the warning criteria of .10. The AIC and BIC values were 17755.028 and 17918.577, respectively, and will become useful in comparing subsequent models.

Our second model was a single-order, multidimensional model where each of the 22 items loaded onto one of four factors. Standardized pattern coefficients ranged between .59 and .80 on the AF factor, between .64 and .82 on the AS factor, between .35 and .60 on the MI factor, and between .59 and .82 on the AUA factor. The Chi-square index was statistically significant ($\chi^2(203) = 220.858, p < .186$) indicating reasonable fit. The CFI value of .99 exceeded the recommendation of .95. The RMSEA = .017 (90% CI [.000, .031]) was satisfactory. The SRMR value of .058 remained below the warning criteria of .10. The AIC and BIC values were 16983.750 and 17169.602, respectively.

The Chi-square difference test ($\chi^2(6) = 783.28, p < .001$) was statistically significant and AIC and BIC values of the multidimensional value were lowest. Thus, we conclude the multidimensional model (i.e., the first-order, correlated factors model) is superior and acceptable for use in preliminary research and evaluation.

We will continue to create, evaluate, and compare models in the next lesson.

10.7 A concluding thought

Much like the children's game *Don't Break the Ice* we start with a full, saturated, matrix of sample data where every indicator/item is allowed to correlate/covary with every other.

As researchers, we specify a more parsimonious model where we fix some relations to zero and allow others to relate. In our GRMSAAW example, we allowed

- the AF items to relate via their relationship to the AF factor;
- the AS items to relate via their relationship to the AS factor;
- the MI items to relate via their relationship to the MI factor; and
- the AUA items to relate via their relationship to the AUA factor.
- we did not allow any of the items on any given factor to relate to the items on any other factor; these are *hard hypotheses* where we fix the relation to zero.

Our goal (especially via the chi-square test) is that we account for as much variance as possible through the specified relations that remain. Harkening to the *Don't Break the Ice* metaphor, we want the ice matrix to remain stable with as many ice cubes deleted as possible.



Source: <https://www.flickr.com/photos/arfsb/4407495674>

10.8 Practice Problems

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. The least complex is to change the random seed in the research and rework the problem demonstrated in the lesson. The most complex is to use data of your own. In either case, please plan to:

10.8.1 Problem #1: Play around with this simulation.

Copy the script for the simulation and then change (at least) one thing in the simulation to see how it impacts the results.

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Prepare data for CFA (items only df, reverse-scored)	5	_____
2. Specify and run a unidimensional model	5	_____
3. Narrate adequacy of fit with χ^2 , CFI, RMSEA, SRMR (write a mini-results section)	5	_____
4. Specify and run a single-order model with correlated factors	5	_____
5. Narrate adequacy of fit with χ^2 , CFI, RMSEA, SRMR (write a mini-results section)	5	_____
6. Compare model fit with $\chi^2\Delta$, AIC, BIC	5	_____
7. APA style results with table(s) and figure	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

10.8.2 Problem #2: Use simulated data from other lessons.

The second option comes from the “the back of the book” where a [chapter](#) contains simulated data for all of the examples worked in this volume. Any of these is available for CFA.

Assignment Component	Points Possible	Points Earned
1. Prepare data for CFA (items only df, reverse-scored)	5	_____
2. Specify and run a unidimensional model	5	_____
3. Narrate adequacy of fit with χ^2 , CFI, RMSEA, SRMR (write a mini-results section)	5	_____
4. Specify and run a single-order model with correlated factors	5	_____
5. Narrate adequacy of fit with χ^2 , CFI, RMSEA, SRMR (write a mini-results section)	5	_____
6. Compare model fit with $\chi^2\Delta$, AIC, BIC	5	_____
7. APA style results with table(s) and figure	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

10.8.3 Problem #3: Try something entirely new.

As a third option, you are welcome to use data to which you have access and is suitable for CFA. These could include other simulated data, data available through open science repositories, or your own data (presuming you have permission to use it). In either case, please plan to:

Using the lecture and workflow (chart) as a guide, please work through all the steps listed in the proposed assignment/grading rubric.

Assignment Component	Points Possible	Points Earned
1. Prepare data for CFA (items only df, reverse-scored)	5	_____
2. Specify and run a unidimensional model	5	_____
3. Narrate adequacy of fit with χ^2 , CFI, RMSEA, SRMR (write a mini-results section)	5	_____
4. Specify and run a single-order model with correlated factors	5	_____
5. Narrate adequacy of fit with χ^2 , CFI, RMSEA, SRMR (write a mini-results section)	5	_____
6. Compare model fit with $\chi^2\Delta$, AIC, BIC	5	_____
7. APA style results with table(s) and figure	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

Chapter 11

CFA: Hierarchical and Nested Models

[Screencasted Lecture Link](#)

This is the second lecture in our series on confirmatory factor analysis (CFA).

Our goal is:

- The systematic and sequential modeling and comparing of:
 - first-order structures (correlated v. uncorrelated factors)
 - second-order structures
 - bifactor structures
- Using modification indices to tweak each model's fit (e.g., by freeing error covariances)
- Determining and tracking the identification status of models, including:
 - nested/nesting models, how that impacts fit, and how comparisons are completed relative to nesting status
 - issues of equivalent models

11.1 Navigating this Lesson

The lecture is just under two hours. I would add another two-to-three hours to work through and digest the materials.

While the majority of R objects and data you will need are created within the R script that sources the chapter, occasionally there are some that cannot be created from within the R framework. Additionally, sometimes links fail. All original materials are provided at the [Github site](#) that hosts the book. More detailed guidelines for ways to access all these materials are provided in the OER's [introduction](#)

11.1.1 Learning Objectives

Focusing on this week's materials, make sure you can:

- Specify single order (correlated and uncorrelated), second order, and bifactor models.

- Interpret model adequacy and fit.
- Compare models on the basis of statistical criteria.
- Determine which (among models) is the nested model.
- Memorize which model (nested or nesting) will have better fit (without looking at the results).
- Determine whether or not models (or alterations to their specification) remain statistically identified.

11.1.2 Planning for Practice

In each of these lessons I provide suggestions for practice that allow you to select one or more problems that are graded in difficulty. The least complex is to change the random seed and rework the problem demonstrated in the lesson. The results *should* map onto the ones obtained in the lecture.

The second option comes from the “the back of the book” where a **chapter** contains simulated data for all of the examples worked in this volume. Any of these is available for CFA.

As a third option, you are welcome to use data to which you have access and is suitable for CFA. These could include other simulated data, data available through open science repositories, or your own data (presuming you have permission to use it).

The suggestion for practice spans the prior chapter and this one . For this combination assignment, you should plan to:

- Prepare the data frame for CFA.
- Specify and run unidimensional, single order (with correlated factors), second-order, and bifactor models.
- Narrate the adequacy of fit with χ^2 , CFI, RMSEA, SRMR
 - Write a mini-results section for each
- Compare model fit with $\chi^2\Delta$, AIC, and BIC.
- Write an APA style results sections with table(s) and figures.

In preparing this chapter, I drew heavily from the following resource(s). Other resources are cited (when possible, linked) in the text with complete citations in the reference list.

- Byrne, B. M. (2016). Application 3: Testing the Factorial Validity of Scores from a Measurement Scale (Second-Order CFA model). Chapter 5. In Structural Equation Modeling with AMOS: Basic Concepts, Applications, and Programming, Third Edition. Taylor & Francis Group. <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4556523>
- Dekay, Nicole (2021). Quick Reference Guide: The statistics for psychometrics <https://www.humanalysts.com/quick-reference-guide-the-statistics-for-psychometrics>
- Flora, D. B. (2020). Your Coefficient Alpha Is Probably Wrong, but Which Coefficient Omega Is Right? A Tutorial on Using R to Obtain Better Reliability Estimates. Advances in Methods and Practices in Psychological Science, 3(4), 484–501. <https://doi.org/10.1177/2515245920951747>

- Kline, R. (2016). Principles and practice of structural equation modeling (Fourth ed., Methodology in the social sciences). New York: The Guilford Press.
 - Chapter 6: Specification of Observed-Variable (Path Models)
 - Chapter 7: Identification of Observed-Variable (Path) Models *
 - Chapter 9: Specification and Identification of Confirmatory Factor Analysis Models
 - Chapter 13: Analysis of Confirmatory Factor Analysis Models
- Rosseel, Y. (2019). The *lavaan* tutorial. Belgium: Department of Data Analysis, Ghent University. <http://lavaan.ugent.be/tutorial/tutorial.pdf>

11.1.3 Packages

The packages used in this lesson are embedded in this code. When the hashtags are removed, the script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them.

```
#will install the package if not already installed
#if(!require(lavaan)){install.packages("lavaan")}
#if(!require(semPlot)){install.packages("semPlot")}
#if(!require(psych)){install.packages("psych")}
#if(!require(semTable)){install.packages("semTable")}
#if(!require(semTools)){install.packages("semTools")}
```

11.2 CFA Workflow

Below is a screenshot of a CFA workflow. The original document is located in the [Github site](#) that hosts the ReCentering Psych Stats: Psychometrics OER.

Because the intended audience for the ReCentering Psych Stats OER is the scientist-practitioner-advocate, this lesson focuses on the workflow and decisions. As you might guess, the details of CFA can be quite complex and require more investigation and decision-making in models that pose more complexity or empirical challenges.

- Creating an items only dataframe where any items are scaled in the same direction (e.g., negatively worded items are reverse-scored).
- Determining a factor structure that is *identified*, that is
 - A single factor (unidimensional) model has at least three items/indicators
 - Multidimensional models have at least two items per factor
- Specify a series of models, these typically include
 - A unidimensional model (all items on a single factor)
 - A single order structure with correlated factors
 - A second order structure
 - A bifactor structure

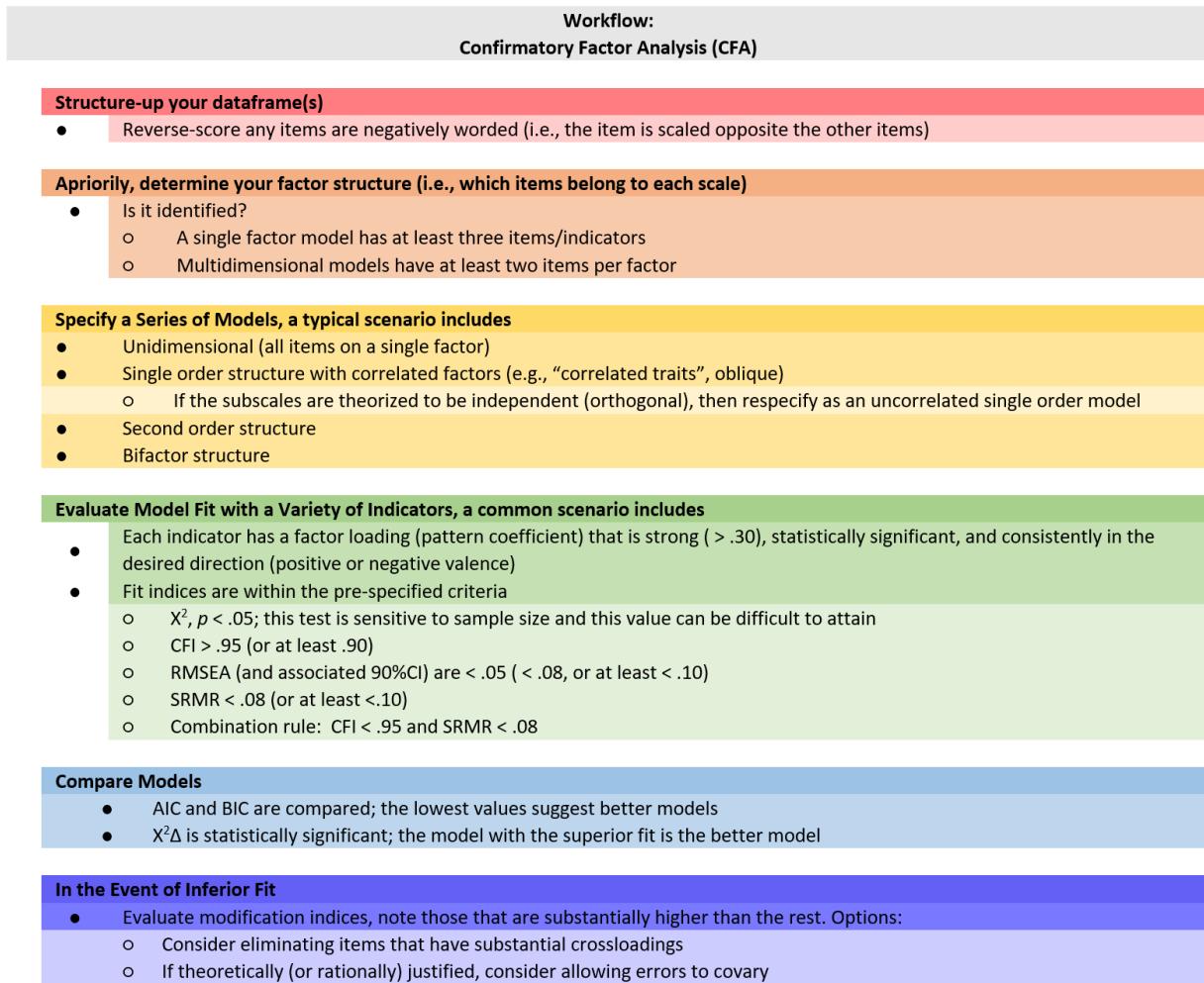


Figure 11.1: Image of a workflow for specifying and evaluating a confirmatory factor analytic model

- Evaluate model fit with a variety of indicators
 - factor loadings
 - fit indices
- Compare models
- In the event of poor model fit, investigate modification indices and consider respecification
 - eliminating items
 - changing factor membership
 - allowing errors to covary

11.3 Another Look at Varying Factor Structures

In this lecture we move into second-order and bifactor models, let's look again factor structures, considering unidimensional, first-order, and second-order variations.

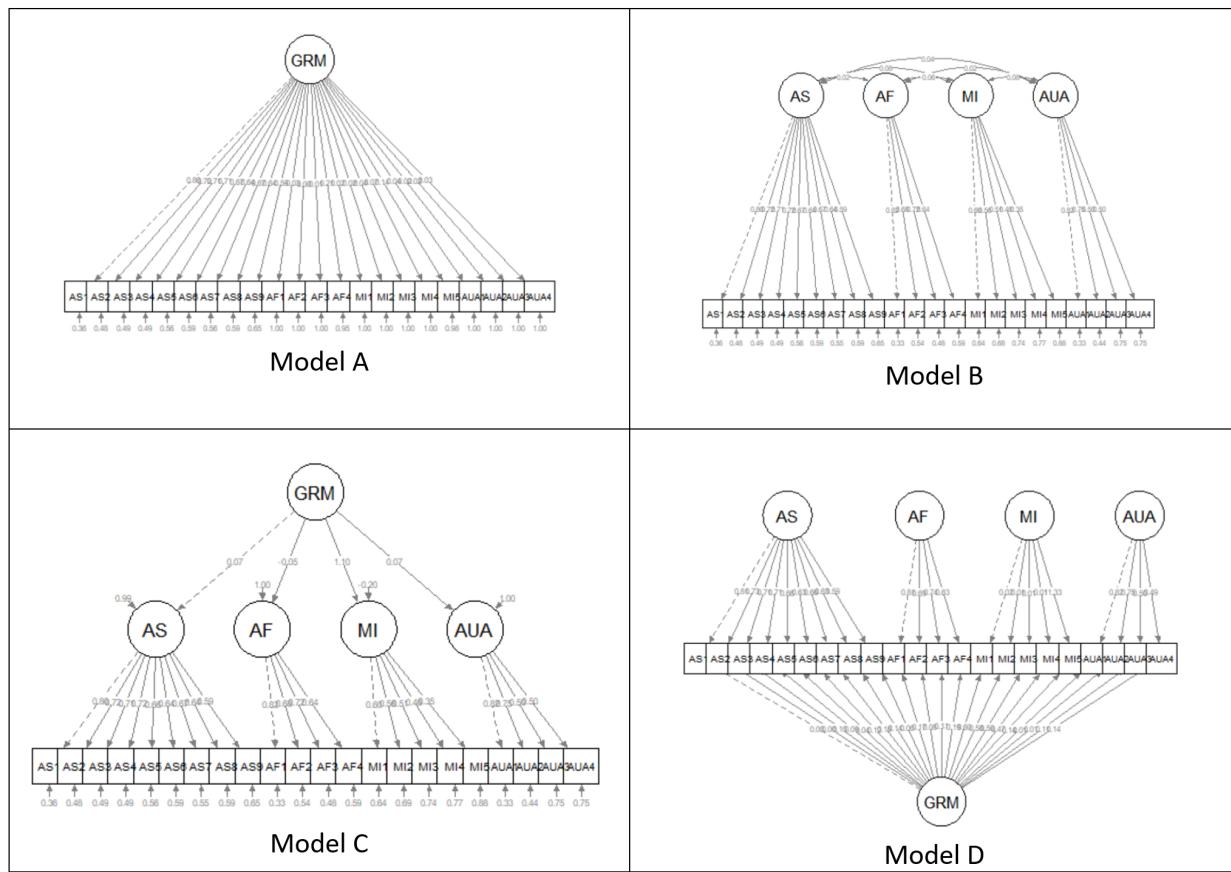


Figure 11.2: Image of first-order, second-order, and bifactor factor structures

Models A and B are **first-order models**. Note that all factors are on a single plane.

- Model A is unidimensional: each item is influenced by a single common factor, and defined by a single term that includes systematic and random error. Note that there is only one *systematic* source of variance for each item AND it is from a single source: F1.

- Model B is often referred to as a “correlated traits” model. Here, the larger construct is separated into distinct-yet-correlated elements. The variance of each item is assumed to be a weighted linear function of two or more common factors.
- Model C is a second-order factor structure. Rather than merely being correlated, factors are related because they share a common cause. In this model, the second-order factor *explains* why three or more traits are correlated. Note that there is no direct relationship between the item and the target construct. Rather, the relationship between the second-order factor and each item is mediated through the primary factor (yes, an indirect effect!).
- Model D is a bifactor structure. Here, each item loads on a general factor. This general factor (bottom row) reflects what is common among the items and represents the individual differences on the target dimension that a researcher is most interested in. Group factors (top row) are now specified as *orthogonal*. The group factors represent common factors measured by the items that explain item response variation not accounted for by the general factor. In some research scenarios, the group factors are termed “nuisance” dimensions. That is, that which they have in common interferes with measuring the primary target of interest.

11.4 Revisiting Model Identification

Model identification means it is *theoretically possible* for a statistical analysis to derive a unique estimate of every model parameter.

Theoretical emphasizes that identification is a property of the model and not the data; that is, it doesn’t matter if the sample size is 100 or 10,000.

CFA has the same general requirements for identification as other forms of SEM:

1. Every latent variable (including errors) must be scaled; and
2. Model degrees of freedom must be at least zero ($df_M \leq 0$) (aka “the counting rule”; this means that there must be at least as many observations as there are free parameters)

11.4.1 Identification Status

Underidentified (or underdetermined) models violate the counting rule because there are more free parameters than observations.

For example, solve this equation:

$$a + b = 6.$$

There are an infinite number of solutions: $4 + 2$, $3 + 3$, $2.5 + 3.5$...and so on to ∞ . When the computing algorithm tries to solve this problem, it will fail to converge.

The parallel scenario in an SEM/CFA model with more free parameters than observations would have *negative df*.

Just-identified (or just-determined) models have a single unique solution: ($df_M = 0$)

For example, for this set of equations: