

ReCentering Psych Stats: Multivariate Modeling

Lynette H. Bikos, PhD, ABPP

31 Oct 2023

Contents

1 Scrubbing	25
1.1 Navigating this Lesson	25
1.1.1 Learning Objectives	25
1.1.2 Planning for Practice	26
1.1.3 Readings & Resources	26
1.1.4 Packages	26
1.2 Workflow for Scrubbing	26
1.3 Research Vignette	28
1.4 Working the Problem	28
1.4.1 intRavenous Qualtrics	28
1.4.1.1 Option 1. Upload an .rds file	30
1.4.1.2 Option 2. Upload a .csv file	31
1.4.2 About the <i>Rate-a-Recent-Course Survey</i>	31
1.4.3 The Codebook	32
1.5 Scrubbing	33
1.5.1 Tools for Data Manipulation	33
1.5.2 Inclusion and Exclusion Criteria	34
1.5.3 Renaming Variables	37
1.5.4 Downsizing the Dataframe	38
1.6 Toward the APA Style Write-up	39
1.6.1 Method/Procedure	39
1.7 Practice Problems	39
1.7.1 Problem #1: Rework the Chapter Problem	39
1.7.2 Problem #2: Use the <i>Rate-a-Recent-Course Survey</i> , Choosing Different Variables	39
1.7.3 Problem #3: Other data	40

1.7.4	Grading Rubric	40
1.8	Bonus Track:	40
1.8.1	Importing data from an exported Qualtrics .csv file	40
2	Scoring	43
2.1	Navigating this Lesson	43
2.1.1	Learning Objectives	43
2.1.2	Planning for Practice	43
2.1.3	Readings & Resources	44
2.1.4	Packages	44
2.2	Workflow for Scoring	45
2.3	Research Vignette	45
2.4	On Missing Data	46
2.4.1	Data Loss Mechanisms	46
2.4.2	Diagnosing Missing Data Mechanisms	47
2.4.3	Managing Missing Data	48
2.4.4	Available Information Analysis (AIA)	49
2.5	Working the Problem	50
2.5.1	Variable Planning and Preparation	50
2.5.2	Missing Data Analysis: Whole df and Item level	56
2.5.3	Analyzing Missing Data Patterns	58
2.5.4	Can we identify the missing mechanisms?	59
2.6	Scoring	59
2.6.1	Reverse scoring	59
2.7	Missing Analysis: Scale level	61
2.8	Revisiting Missing Analysis at the Scale Level	63
2.8.1	Scale Level: Patterns of Missing Data	64
2.8.2	R-ready for Analysis	66
2.9	The APA Style Write-Up	67
2.10	Results	67
2.11	Practice Problems	68
2.11.1	Problem #1: Reworking the Chapter Problem	68
2.11.2	Problem #2: Use the <i>Rate-a-Recent-Course Survey</i> , Choosing Different Variables	68
2.11.3	Problem #3: Other data	68
2.11.4	Grading Rubric	68

CONTENTS	5
----------	---

3 Data Dx	71
3.1 Navigating this Lesson	71
3.1.1 Learning Objectives	71
3.1.2 Planning for Practice	72
3.1.3 Readings & Resources	72
3.1.4 Packages	72
3.2 Workflow for Preliminary Data Diagnostics	72
3.3 Research Vignette	73
3.4 Internal Consistency of Scales/Subscales	75
3.5 Distributional Characteristics of the Variables	79
3.5.1 Evaluating Univariate Normality	79
3.5.2 Pairs Panels	82
3.6 Evaluating Multivariate Normality	84
3.7 A Few Words on Transformations	87
3.8 The APA Style Write-Up	87
3.8.1 Data Diagnostics	87
3.9 A Quick Regression of our Research Vignette	89
3.9.1 Results	89
3.10 Practice Problems	91
3.10.1 Problem #1: Reworking the Chapter Problem	91
3.10.2 Problem #2: Use the <i>Rate-a-Recent-Course</i> Survey, Choosing Different Variables	92
3.10.3 Problem #3: Other data	92
3.10.4 Grading Rubric	92
3.11 Homeworked Example	92
3.11.1 Scrubbing	93
3.11.2 Scoring	95
3.11.3 Data Dx	103
3.11.4 Results	111
4 Multiple Imputation (A Brief Demo)	115
4.1 Navigating this Lesson	115
4.1.1 Learning Objectives	115
4.1.2 Planning for Practice	116

4.1.3	Readings & Resources	116
4.1.4	Packages	116
4.2	Workflow for Multiple Imputation	117
4.3	Research Vignette	117
4.4	Multiple Imputation – a Super Brief Review	119
4.4.1	Steps in Multiple Imputation	120
4.4.2	Statistical Approaches to Multiple Imputation	121
4.5	Working the Problem	121
4.5.1	Selecting and Formatting Variables	122
4.5.2	Creating Composite Variables	124
4.5.3	The Multiple Imputation	126
4.5.4	Creating Scale Scores	128
4.6	Multiple Regression with Multiply Imputed Data	129
4.7	Toward the APA Style Write-up	131
4.7.1	Method/Data Diagnostics	131
4.7.2	Results	132
4.8	Multiple imputation considerations	132
4.9	Practice Problems	133
4.9.1	Problem #1: Reworking the Chapter Problem	133
4.9.2	Problem #2: Use the <i>Rate-a-Recent-Course</i> Survey, Choosing Different Variables	133
4.9.3	Problem #3: Other data	133
4.9.4	Grading Rubric	134
4.10	Homeworked Example	134
4.10.1	Scrubbing	134
4.10.1.1	Format any variables that shouldn't be imputed in their raw form .	136
4.10.1.2	Multiply impute a minimum of 5 sets of data	137
4.10.1.3	Run a regression (for multiply imputed data) with at least three variables	140
4.10.1.4	APA style write-up of the multiple imputation section of data diagnostics	142
4.10.1.5	APA style write-up regression results	143

5 Simple Mediation	147
5.1 Navigating this Lesson	147
5.1.1 Learning Objectives	147
5.1.2 Planning for Practice	148
5.1.3 Readings & Resources	148
5.1.4 Packages	149
5.2 Estimating Indirect Effects (the analytic approach often termed <i>mediation</i>)	149
5.2.1 The definitional and conceptual	149
5.3 Workflow for Simple Mediation	151
5.4 Super Simple Mediation in <i>lavaan</i> : A focus on the mechanics	154
5.4.1 Simulate Fake Data	155
5.4.2 Specify Mediation Model	155
5.4.3 Interpret the Output	159
5.4.4 A Figure and Table	161
5.4.5 Results	165
5.5 Research Vignette	165
5.5.1 Data Simulation	166
5.5.2 Scrubbing, Scoring, and Data Diagnostics	171
5.5.3 Specify the Model in <i>lavaan</i>	173
5.5.4 Interpret the Output	177
5.5.5 A Figure and a Table	177
5.5.6 Results	180
5.6 Considering Covariates	180
5.6.1 A Figure and a Table	185
5.6.2 APA Style Write-up	188
5.7 STAY TUNED	188
5.8 Residual and Related Questions...	188
5.9 Practice Problems	190
5.9.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed	190
5.9.2 Problem #2: Rework the research vignette, but swap one or more variables .	190
5.9.3 Problem #3: Use other data that is available to you	190
5.9.4 Grading Rubric	190
5.10 Homeworked Example	191

5.10.1 Assign each variable to the X, Y, or M roles (ok but not required to include a covariate)	191
6 Complex Mediation	201
6.1 Navigating this Lesson	201
6.1.1 Learning Objectives	201
6.1.2 Planning for Practice	202
6.1.3 Readings & Resources	202
6.1.4 Packages	202
6.2 Complex Mediation	203
6.3 Workflow for Complex Mediation	203
6.4 Parallel Mediation	205
6.4.1 A Mechanical Example	206
6.4.1.1 Data Simulation	206
6.4.1.2 Specifying <i>lavaan</i> code	207
6.4.1.3 A note on indirect effects and confidence intervals	211
6.4.1.4 Figures and Tables	211
6.4.1.5 APA Style Writeup	214
6.4.2 Research Vignette	215
6.4.2.1 Data Simulation	216
6.4.3 Scrubbing, Scoring, and Data Diagnostics	219
6.4.3.1 Specifying the <i>lavaan</i> model	220
6.4.3.2 Table and Figure	222
6.4.3.3 APA Style Writeup	226
6.5 Serial Multiple Mediator Model	227
6.5.1 We stick with the Lewis et al. [2017] example, but modify it.	228
6.5.2 Specify the <i>lavaan</i> model	229
6.5.2.1 Table and Figure	230
6.5.3 APA Style Writeup	234
6.6 STAY TUNED	235
6.7 Troubleshooting and FAQs	235
6.8 Practice Problems	236
6.8.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed	237
6.8.2 Problem #2: Rework the research vignette, but swap one or more variables .	237

6.8.3	Problem #3: Use other data that is available to you	237
6.8.4	Grading Rubric	237
6.9	Homeworked Example	237
7	Simple Moderation in OLS and MLE	251
7.1	Navigating this Lesson	251
7.1.1	Learning Objectives	251
7.1.2	Planning for Practice	252
7.1.3	Readings & Resources	252
7.1.4	Packages	253
7.2	On <i>Modeling</i> : Introductory Comments on the simultaneously invisible and paradigm-shifting transition we are making	254
7.2.1	NHST versus modeling	254
7.2.2	Introducing: <i>The Model</i>	255
7.3	OLS to ML for Estimation	256
7.3.1	Ordinary least squares (OLS)	256
7.3.2	Maximum likelihood estimation (MLE): A brief orientation	257
7.3.3	OLS and MLE Comparison	258
7.3.4	Hayes and PROCESS (aka conditional process analysis)	259
7.4	Introducing the <i>lavaan</i> package	259
7.4.1	The FIML magic for which we have been waiting	260
7.5	Picking up with Moderation	262
7.6	Workflow for a Simple Moderation	264
7.6.1	Research Vignette	264
7.6.1.1	Data Simulation	265
7.6.2	Scrubbing, Scoring, and Data Diagnostics	268
7.7	Working the Simple Moderation with OLS and MLE	272
7.7.1	OLS with <i>lm()</i>	272
7.7.1.1	An APA Style Write-up of OLS results	276
7.7.2	MLE with <i>lavaan::sem()</i>	277
7.7.3	Tabling the data	285
7.7.4	APA Style Writeup	286
7.8	STAY TUNED	286
7.9	Residual and Related Questions...	287

7.10 Practice Problems	287
7.10.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed	287
7.10.2 Problem #2: Rework the research vignette, but swap one or more variables .	287
7.10.3 Problem #3: Use other data that is available to you	287
7.10.4 Grading Rubric	288
7.11 Bonus Track:	288
7.12 Homeworked Example	289
7.12.1 Note similarities and differences in the OLS results	303
8 Moderated Mediation	307
8.1 Navigating this Lesson	307
8.1.1 Learning Objectives	307
8.1.2 Planning for Practice	308
8.1.3 Readings & Resources	308
8.1.4 Packages	309
8.2 Conditional Process Analysis	310
8.2.1 The definitional and conceptual	310
8.3 Workflow for Moderated Mediation	312
8.4 Research Vignette	314
8.4.1 Simulating the data from the journal article	315
8.4.2 Scrubbing, Scoring, and Data Diagnostics	318
8.4.3 Quick peek at the data	319
8.5 Working the Moderated Mediation	320
8.5.1 Piecewise Assembly of the Moderated Mediation	321
8.5.1.1 Analysis #1: A simple moderation	321
8.5.1.2 Analysis #2: Another simple moderation	326
8.5.1.3 Analysis #3: A simple mediation	330
8.6 The Moderated Mediation: A Combined analysis	335
8.6.1 Specification in <i>lavaan</i>	336
8.6.2 Beginning the interpretation	344
8.6.3 Tabling the data	344
8.6.3.1 Conditional Indirect effects	345
8.6.3.2 Conditional Direct effect	346

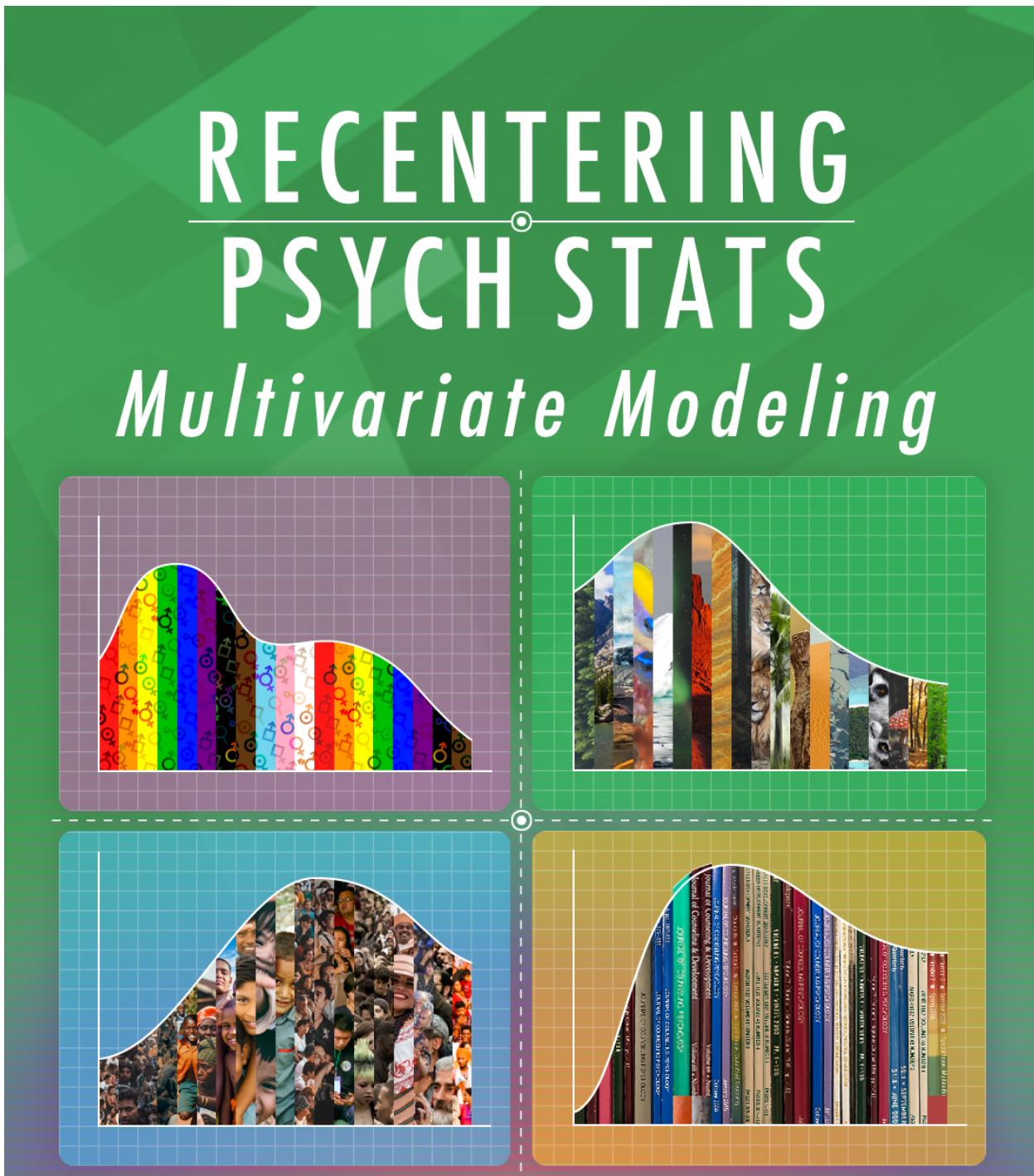
8.6.4	Model trimming	346
8.6.5	APA Style Write-up	346
8.7	STAY TUNED	348
8.8	Residual and Related Questions...	348
8.9	Practice Problems	350
8.9.1	Problem #1: Rework the research vignette as demonstrated, but change the random seed	350
8.9.2	Problem #2: Rework the research vignette, but swap one or more variables .	350
8.9.3	Problem #3: Use other data that is available to you	350
8.9.4	Grading Rubric	351
8.10	Homeworked Example 1: A moderation on the <i>a</i> path	351
8.11	Homeworked Example 2: A moderation on the <i>b</i> path	372
9	Establishing the Measurement Model	395
9.1	Navigating this Lesson	395
9.1.1	Learning Objectives	395
9.1.2	Planning for Practice	396
9.1.3	Readings & Resources	396
9.1.4	Packages	397
9.2	Introduction to Structural Equation Modeling (SEM)	397
9.3	Workflow for Evaluating a Structural Model	398
9.4	The Measurement Model: Specification and Evaluation	401
9.4.1	Degrees of Freedom and Model Identification	402
9.5	Research Vignette	403
9.5.1	Simulating the data from the journal article	404
9.6	Scrubbing, Scoring, and Data Diagnostics	407
9.7	Specifying the Measurement Model in <i>lavaan</i>	407
9.8	Interpreting the Output	425
9.8.1	Global Fit Indices	425
9.8.1.1	Model Test <i>User</i> Model:	426
9.8.1.2	Model Test <i>Baseline</i> Model	426
9.8.1.3	Incremental Fit Indices (Located in the <i>User versus Baseline Models</i> section)	427
9.8.1.4	Loglikelihood and Information Criteria	427
9.8.1.5	Root Mean Square Error of Approximation	427

9.8.1.6	Standardized Root Mean Square Residual	428
9.8.1.7	Factor Loadings	428
9.8.1.8	Adequacy of the Initial Measurement Model	428
9.9	Parceling	429
9.9.1	The Pros and Cons of Parceling	429
9.9.1.1	Philosophical Arguments	430
9.9.1.2	Psychometric Arguments	430
9.9.1.3	Modelers' Arguments	431
9.9.2	Practical Procedures for Parceling	431
9.9.2.1	Purposive Approaches to Parceling	431
9.9.2.2	Random Approaches to Parceling	432
9.10	Parceling with Subscale Scores	432
9.10.1	Measurement Model with Subscale Parcels	434
9.10.2	Measurement Model with Just-Identified Random Parcels	438
9.10.2.1	Evaluating the Randomly Identified Measurement Model	442
9.10.3	APA Style Write-up of the Results	446
9.11	Residual and Related Questions...	448
9.11.1	Wait! Why did we do this?	448
9.11.2	What if one of my variables only has one or two indicators?	448
9.11.3	What if I have missing data?	449
9.12	Practice Problems	449
9.12.1	Problem #1: Rework the research vignette as demonstrated, but change the random seed	449
9.12.2	Problem #2: Rework the research vignette, but swap one or more variables .	449
9.12.3	Problem #3: Try something entirely new.	449
9.12.4	Grading Rubric	450
9.13	Homeworked Example	450
10	Specifying and Evaluating the Structural Model	465
10.1	Navigating this Lesson	465
10.1.1	Learning Objectives	465
10.1.2	Planning for Practice	466
10.1.3	Readings & Resources	466
10.1.4	Packages	466

10.2 Evaluating Structural Models	467
10.3 Workflow for Evaluating a Structural Model	468
10.4 Research Vignette	470
10.4.1 Simulating the data from the journal article	470
10.5 Scrubbing, Scoring, and Data Diagnostics	473
10.6 Script for Specifying Models in <i>lavaan</i>	473
10.7 Quick Specification of the Measurement Model	474
10.8 The Structural Model: Specification and Evaluation	483
10.8.1 Model Identification	484
10.8.1.1 Model identification for the overall SEM	485
10.8.1.2 Model identification for the structural portion of the model	486
10.8.1.3 Specifying and Evaluating the Structural Model	487
10.8.1.4 Interpreting the Output	491
10.8.2 APA Style Write-up of the Results	497
10.9 What About Alternative Models?	498
10.9.1 Swapping the Mediator and the Outcome	499
10.9.2 REMS as a Predictor of Both CMI and PWB	506
10.9.3 What if we allowed PWB and CMI to co-vary?	513
10.9.4 Model comparisons	520
10.10 STAY TUNED	521
10.11 Practice Problems	521
10.11.1 Problem #1: Change the random seed	522
10.11.2 Problem #2: Swap one or more of the variables	522
10.11.3 Problem #3: Try something entirely new.	522
10.12 Homeworked Example	522
10.12.1 Conduct a formal comparison of <i>global</i> fit.	547
11 SEM: Model Respecifications	551
11.1 Navigating this Lesson	551
11.1.1 Learning Objectives	551
11.1.2 Planning for Practice	552
11.1.3 Readings & Resources	552
11.1.4 Packages	553
11.2 Respecifying Structural Models	553

11.2.1 Model Building	553
11.2.2 Model Trimming	554
11.2.3 Restating Approaches to Respecification	554
11.3 Workflow for Evaluating a Structural Model	555
11.4 Research Vignette	558
11.4.1 Simulating the data from the journal article	559
11.5 Scrubbing, Scoring, and Data Diagnostics	562
11.6 Script for Specifying Models in <i>lavaan</i>	562
11.7 Quick Specification of the Measurement Model	563
11.8 Specifying and Evaluating the Hypothesized Structural Model	574
11.8.1 Model Identification for the Hypothesized (Original) Model	574
11.8.1.1 Specifying and Evaluating the Structural Model	576
11.8.1.2 Interpreting the Output	583
11.9 Model Building	590
11.10 Model Trimming	601
11.10.0.1 Extreme Model Trimming	610
11.10.1 APA Style Write-up of the Results	619
11.11 STAY TUNED	621
11.12 Practice Problems	621
11.12.1 Problem #1: Change the random seed	621
11.12.2 Problem #2: Swap one or more of the variables	621
11.12.3 Problem #3: Try something entirely new.	621
11.13 Homeworked Example	622

BOOK COVER



at the GitHub repository:

- Formatted as an [html book](#) via GitHub Pages available
- As a [PDF](#)
- As an [ebook](#)
- As a [Word](#)

All materials used in creating this OER are available at its [GitHub repo](#).

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 31 Oct 2023. 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 [[Mallinckrodt et al., 2014](#)]. 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 [[2012](#)] 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 [[Green and Salkind, 2017](#)] 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 [[2016b](#)], “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 [[2019](#)] 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.

DATA PREP

Chapter 1

Scrubbing

[Screencasted Lecture Link](#)

The focus of this chapter is the process of starting with raw data and preparing it for multivariate analysis. To that end, we will address the conceptual considerations and practical steps in “scrubbing and scoring.”

A twist in this lesson is that I am asking you to contribute to the dataset that serves as the basis for the chapter and the practice problems. In the spirit of *open science*, this dataset is available to you and others for your own learning. Before continuing, please take 15-20 minutes to complete the survey titled, [Rate-a-Recent-Course: A ReCentering Psych Stats Exercise](#). The study is approved by the Institutional Review Board at Seattle Pacific University (SPUIRB# 202102011, no expiration). Details about the study, including an informed consent, are included at the link.

1.1 Navigating this Lesson

There is about 90 minutes of lecture. If you work through the materials with me it would be good to add another hour.

While the majority of R objects and data you will need are created within the R script that sources the chapter, there are a few 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](#)

1.1.1 Learning Objectives

Learning objectives from this lecture include the following:

- Import data from Qualtrics into R.
- Apply inclusion and exclusion criteria to a dataset.
- Rename variables.
- Create a smaller dataframe with variables appropriate for testing a specific statistical model.
- Use critical data manipulation functions from the *tidyverse* (and *dplyr*) in particular such as *filter()*, *select()*, and *mutate()* to prepare variables.
- Articulate the initial steps in a workflow for scrubbing and scoring data.

1.1.2 Planning for Practice

The suggestions for practice will start with this chapter and continue in the next two chapters (Scoring, Data Dx). Using Parent's [2013] AIA (available item analysis) approach to managing missing data, you will scrub-and-score a raw dataset. Options of graded complexity could include:

- Repeating the steps in the chapter with the most recent data from the Rate-A-Recent-Course survey; differences will be in the number of people who have completed the survey since the chapter was written.
- Use the dataset that is the source of the chapter, but score a different set of items that you choose.
- Begin with raw data to which you have access.

1.1.3 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.

- Parent, M. C. (2013). Handling item-level missing data: Simpler is just as good. *The Counseling Psychologist*, 41(4), 568–600. <https://doi.org/10.1177/0011000012445176>
 - The purpose of Parent's article was to argue that complex and resource-intensive procedures like multiple imputation are unnecessary. Following a simulation that supports his claims, Parent provides some guidelines to follow for the AIA approach.
- Kline, R. B. (2016). Data preparation and psychometrics review. In Principles and Practice of Structural Equation Modeling, Fourth Edition. Guilford Publications. <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4000663>
 - Kline's chapter is my “go-to” for making decisions about preparing data for analysis.

1.1.4 Packages

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')}
# if(!require(tidyverse)){install.packages('tidyverse')}
```

1.2 Workflow for Scrubbing

The same workflow guides us through the Scrubbing, Scoring, and Data Dx chapters. In this lesson we focus on downloading data from Qualtrics and determining which cases can be retained for analysis based on inclusion and exclusion criteria.

Here is a narration of the figure:

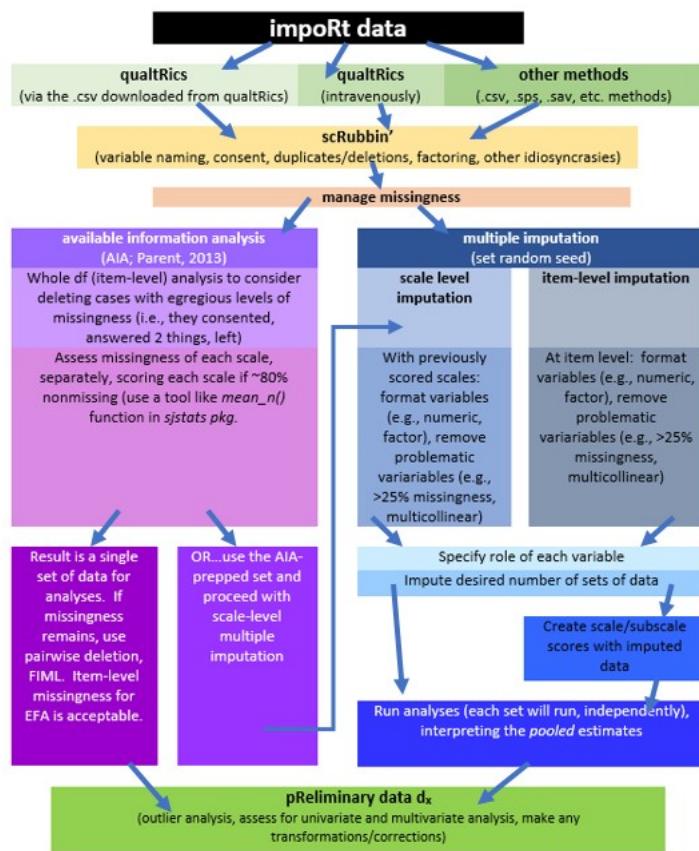


Figure 1.1: An image of a workflow for scrubbing and scoring data.

1. The workflow begins by importing data into R. Most lessons in this series involve simulated data that are created directly in R. Alternatively, data could be:
 - imported “intRavenously” through programs such as Qualtrics,
 - exported from programs such as Qualtrics to another program (e.g., .xlsx, .csv),
 - imported in other forms (e.g., .csv, .sps, .sav).
2. Scrubbing data by
 - variable naming,
 - specifying variable characteristics such as factoring,
 - ensuring that included participants consented to participation,
 - determining and executing the inclusion and exclusion criteria.
3. Conduct preliminary data diagnostics such as
 - outlier analysis
 - assessing for univariate and multivariate analysis
 - making transformations and/or corrections
4. Managing missingness by one of two routes
 - Available information analysis [Parent, 2013] at either the item-level or scale level. The result is a single set of data for analysis. If missingness remains, options include pairwise deletion, listwise deletion, or specifying FIML (when available). Another option is to use multiple imputation.
 - Multiple imputation at either scale level or item-level

1.3 Research Vignette

To provide first-hand experience as both the respondent and analyst for the same set of data, you were asked to complete a survey titled, [Rate-a-Recent-Course: A ReCentering Psych Stats Exercise](#). If you haven’t yet completed it, please consider doing so, now. In order to reduce the potential threats to validity by providing background information about the survey, I will wait to describe it until later in the chapter.

The survey is administered in Qualtrics. In the chapter I teach two ways to import Qualtrics data into R. We will then use the data to work through the steps identified in the workflow.

1.4 Working the Problem

1.4.1 intRavenous Qualtrics

I will demonstrate 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. The surveys were designed to not capture personally identifying information.

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**. To retrieve these:

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

The **root_url** 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 or upload to a data repository, 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.

```
# You only need to run this ONCE to draw from the same Qualtrics
# account. If you change Qualtrics accounts you will need to get a
# different token.

# 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 <- qualtrics::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 account
# (i.e., maybe this one instead of your own). Go back and change your
# API token (it saves your old one). Changing the API likely requires
# a restart of R.
```

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 responses
#'convert', when TRUE, attempts to convert certain question types to the 'proper' data type in R
#'force_request', when TRUE, always downloads the survey from the API instead of from a temporary URL
# 'import_id', when TRUE includes the unique Qualtrics-assigned ID;
# since I have provided labels, I want false

# QTRX_df <- qualtrics::fetch_survey(surveyID = 'SV_b2cClqAllGQ6nLU',
# time_zone = NULL, verbose = FALSE, label=FALSE, convert=FALSE,
# force_request = TRUE, import_id = FALSE)

# useLocalTime = TRUE,
```

It is possible to import Qualtrics data that has been downloaded from Qualtrics as a .csv. I demo this in the Bonus Reel at the end of this lesson.

In prior versions of this chapter I allowed the chapter to automatically update with “all the new data” each time the OER was re-rendered/built. Because I think this caused confusion, I have decided to save the data in both .csv and .rds versions, then clear my environment, upload the .rds (my personal favorite format) version, and demonstrate the scrubbing techniques with that data. If you continue with data you just downloaded from Qualtrics, you will get different answers than are in the lesson. While I think that continuing with the most current data set is a viable option for a practice problem, it could be confusing. Rather, follow one of the two options below to upload .csv or .rds versions of the data I used in the lesson.

1.4.1.1 Option 1. Upload an .rds file

Because .rds files will retain any formatting information we provide about variables, I like using them. The downside is that you cannot simply open and view them outside of the R environment. Here is the code I used to produce the .rds version of the file. If you want to obtain the same results as I report in the chapter, do NOT run it again.

```
# to save the df as an .rds (think 'R object') file on your computer;
# it should save in the same file as the .rmd file you are working
# with saveRDS(QTRX_df, 'QTRX_df230902.rds')
```

Rather, head to the [MultivModel GitHub](#) site and download the *QTRX_df230902b.rds* file. Place it in the same folder as the .rmd you are using and run the code below. And actually, I further re-named the file that you will retrieve so that it won’t be over-written.*

```
QTRX_df <- readRDS("QTRX_df230902b.rds")
```

Occasionally, I have had a student for whom the .rds files don’t seem to work. Uploading a .csv file is an option.

1.4.1.2 Option 2. Upload a .csv file

Simply for your information, here is the code I used to produce the .csv version of the file. If you want to obtain the same results as I report in the chapter, do NOT run it again.

```
# write the simulated data as a .csv write.table(QTRX_df,
# file='QTRX_df230902.csv', sep=',', col.names=TRUE, row.names=FALSE)
```

Rather, head to the [MultivModel GitHub](#) site and download the *QTRX_df230902b.csv* file. Place it in the same folder as the .rmd you are using and run the code below. *And actually, I further re-named the file that you will retrieve so that it won't be over-written.*

```
# bring back the simulated dat from a .csv file QTRX_df <-
# read.csv('QTRX_df230902b.csv', header = TRUE)
```

You need not do both. That is, either download-and-import either the .rds or .csv file.

1.4.2 About the *Rate-a-Recent-Course Survey*

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.

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” [Lewis and Shah, 2019] this is likely an oversimplification. In fact, an increase in diverse representation without attention to interacting factors can increase hostility on campus [Hurtado, 2007]. Thus, we included 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.

Our 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; Iacovino and James [2016] have noted that White students perceive campus more favorably than Black student counterparts. We made this decision to protect the identity of the respondent. As you will see when we download 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) or course evaluation data 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. 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. [Mosley et al., 2021, 2020, Singh, 2020]. 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* [Adames et al., 2021, Capielo Rosario et al., 2019]. Another possibility is that the limited options may erase, or make invisible, other identities. At the time that I am writing the first draft of this chapter, the murder of six Asian American women in Atlanta has 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% [noa, a]. These incidents have occurred not only in cities, but in our neighborhoods and on our campus [Kim, 2021b,a, noa, c]. 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.

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 or harmful comments.

1.4.3 The Codebook

In order to scrub-and-score a survey, it is critical to know about its content, scoring directions for scales/subscales, and its design. A more complete description of the survey design elements is (or will be) available in the *Recentering Psych Stats: Psychometric OER*. The review in this chapter provides just-enough information to allow us to make decisions about which items to retain and how to score them. When they are well-written, information in the **IRB application** and **pre-registration** can be helpful in the scrubbing and scoring process.

Let’s look “live” at the survey. In Qualtrics it is possible to *print* a PDF that looks very similar to its presentation when someone is taking it. You can access that static version [here](#).

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.

It is almost impossible to give this lecture without some reference to Qualtrics and the features used in Qualtrics. An import of raw data from Qualtrics into R can be nightmare in that the Qualtrics-assigned variable names are numbers (e.g., QID1, QID2) – but often out of order because the number is assigned when the question is first created. If the survey is reordered, the numbers get out of sequence.

Similarly, values for Likert-type scales can also get out of order if the scale anchors are revised (which is common to do).

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.

The Rate-a-Recent-Course survey was written using some special features in Qualtrics. These include

- Display logic
 - Items that are U.S.-centric are only shown if the respondent is taking a course from an institution in the U.S. is a student in the U.S.
- Loop and merge
 - Because course may have multiple instructional staff, the information asking about demographic characteristics of the instructors is repeated according to the number input by the respondent
- Random presentation of the 30 items asking about campus climate for the five groups of students
 - Although this might increase the cognitive load of the survey, this helps “spread out” missingness for respondents who might tire of the survey and stop early
- Rank ordering of the institutional level (department, school/faculty, campus/university) to which the respondent feels most connected

Looking at the QTRX_df, *StartDate* thru *UserLanguage* are metadata created by Qualtrics. The remaining variables and associated value labels are in the [codebook](#).

1.5 Scrubbing

With a look at our survey, codebook, and imported data, we now get to the business of scrubbing (deleting those who did not give consent, deleting previews, etc.). This level of “scrubbing” precedes the more formal detection of outliers.

1.5.1 Tools for Data Manipulation

The next stages will provide some experience manipulating data with **dplyr** from the **tidyverse**.

The **tidyverse** is a system of packages (i.e., when you download the tidyverse, you download all its packages/members) for data manipulation, exploration and visualization. The packages in the tidyverse share a common design philosophy. These were mostly developed by Hadley Wickham,

but more recently, more designers are contributing to them. Tidyverse packages are intended to make statisticians and data scientists more productive by guiding them through workflows that facilitate communication and result in reproducible work products. Fundamentally, the tidyverse is about the connections between the tools that make the workflow possible. Critical packages in the tidyverse include:

- **dplyr**: data manipulation: mutate, select, filter, summarize, arrange
- **ggplot2**: extravagant graphing
- **tibble**: a *tibble* is a dataframe that provides the user with more (and less) control over the data.
- **readr**: gives access to “rectangular data” like .csv and tables
- **tidyverse**: tidy data is where each variable is a column, each observation is a row, each value is a cell (duh). **tidyverse**’s contributions are gather(wide to long) and spread(long to wide) as well as separate, extract, unite.
- **purrr**: facilitates working with functions and vectors. For example, if you write a function, using purrr may help you replace loops with code that is more efficient and intuitive.

The tidyverse is ever-evolving – so check frequently for updates and troubleshooting.

A handy cheatsheet for data transformation is found [here](#).

1.5.2 Inclusion and Exclusion Criteria

For me, the first pass at scrubbing is to eliminate the obvious. In our case this includes *previews* and respondents who did not consent to continue. Previews are the researcher-initiated responses usually designed to proofread or troubleshoot survey problems. There could be other first-pass-deletions, such as selecting response between certain dates.

I think these first-pass deletions, especially the ones around consent, are important to do as soon as possible. Otherwise, we might delete some of the variables (e.g., timestamps, consent documentation, preview status) and neglect to delete these cases later in the process.

We are here in the workflow:

We can either update the existing df (by using the same object), or creating a new df from the old. Either works. In my early years, I tended to create lots of new objects. As I have gained confidence in myself and in R, I’m inclined to update the existing df. Why? Because unless you write the object as an outfile (using the same name for the object as for the filename – which I do not recommend), the object used in R does not change the source of the dat. Therefore, it is easy to correct early code and it keeps the global environment less cluttered.

In this particular survey, the majority of respondents will take the survey because they clicked an *anonymous* link provided by Qualtrics. Another Qualtrics distribution method is e-mail. At the time of this writing, we have not recruited by e-mail, but it is possible we could do so in the future. What we should not include, though, are *previews*. These are the times when the researcher is self-piloting the survey to look for errors and to troubleshoot.

```
# the filter command is used when we are making inclusion/exclusion
# decisions about rows != means do not include cases with 'preview'
```

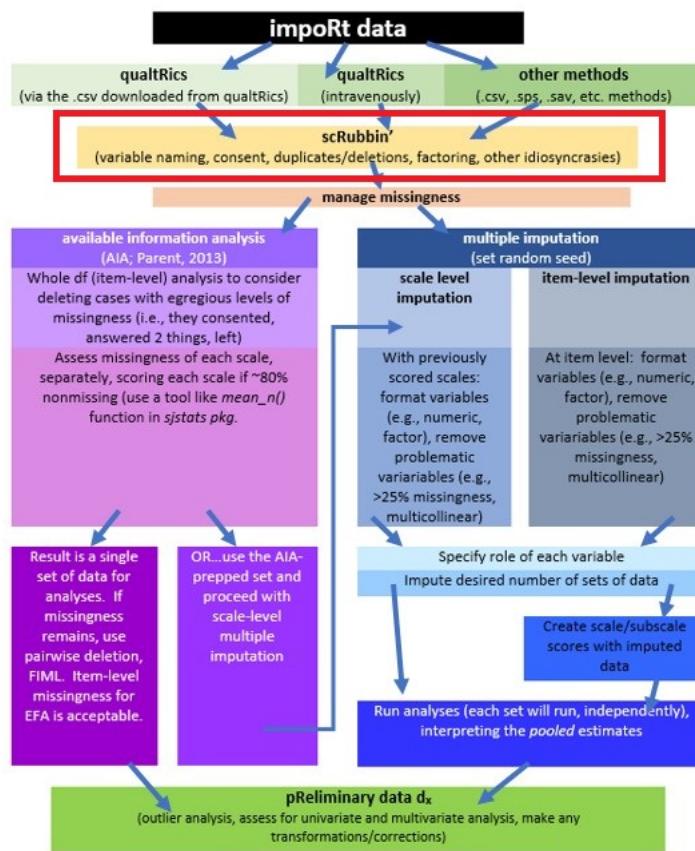


Figure 1.2: An image of a workflow for scrubbing and scoring data.

```

QTRX_df <- dplyr::filter(QTRX_df, DistributionChannel != "preview")

# FYI, another way that doesn't use tidyverse, but gets the same
# result QTRX_df <- QTRX_df[!QTRX_df$DistributionChannel ==
# 'preview',]

```

APA Style, and in particular the Journal Article Reporting Standards (JARS) for quantitative research specify that we should report the frequency or percentages of missing data. We would start our counting *after* eliminating the previews.

```

# I created an object that lists how many rows/cases remain. I used
# inline text below to update the text with the new number
nrow(QTRX_df)

```

[1] 107

CAPTURING RESULTS FOR WRITING IT UP:

Data screening suggested that 107 individuals opened the survey link.

Next let's filter in only those who consented to take the survey. Because Qualtrics discontinued the survey for everyone who did not consent, we do not have to worry that their data is unintentionally included, but it can be useful to mention the number of non-consenters in the summary of missing data.

```

# == are used
QTRX_df <- dplyr::filter(QTRX_df, Consent == 1)
nrow(QTRX_df)

```

[1] 83

CAPTURING RESULTS FOR WRITING IT UP:

Data screening suggested that 107 individuals opened the survey link. Of those, 83 granted consent and proceeded into the survey items.

In this particular study, the categories used to collect information about race/ethnicity were U.S.-centric. Thus, they were only shown if the respondent indicated that the course being rated was taught by an institution in the U.S. Therefore, an additional inclusion criteria for this specific research model should be that the course was taught in the U.S.

```

QTRX_df <- dplyr::filter(QTRX_df, USinst == 0)
nrow(QTRX_df)

```

```
[1] 69
```

CAPTURING RESULTS FOR WRITING IT UP:

Data screening suggested that 107 individuals opened the survey link. Of those, 83 granted consent and proceeded into the survey items. A further inclusion criteria was that the course was taught in the U.S; 69 met this criteria.

1.5.3 Renaming Variables

Even though we renamed the variables in Qualtrics, the loop-and-merge variables were auto-renamed such that they each started with a number. I cannot see how to rename these from inside Qualtrics. A potential problem is that, in R, when variable names start with numbers, they need to be surrounded with single quotation marks. I find it easier to rename them now. I used “i” to start the variable name to represent “instructor.”

The form of the `rename()` function is this: `df_named <- rename(df_raw, NewName1 = OldName1)`

```
QTRX_df <- dplyr::rename(QTRX_df, iRace1 = "1_iRace", iRace2 = "2_iRace",
                           iRace3 = "3_iRace", iRace4 = "4_iRace", iRace5 = "5_iRace", iRace6 = "6_iRace",
                           iRace7 = "7_iRace", iRace8 = "8_iRace", iRace9 = "9_iRace", iRace10 = "10_iRace")
```

Also in Qualtrics, it was not possible to rename the variable (formatted with sliders) that asked respondents to estimate the proportion of classmates in each race-based category. Using the code-book, we can do this now. I will use “cm” to precede each variable name to represent “classmates.”

```
QTRX_df <- dplyr::rename(QTRX_df, cmBiMulti = Race_10, cmBlack = Race_1,
                           cmNBPoC = Race_7, cmWhite = Race_8, cmUnsure = Race_2)
```

Let’s also create an ID variable (different from the lengthy Qualtrics-issued ID) and then move it to the front of the distribution.

```
# Opening the tidyverse so that I can use pipes
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.2     v tidyr    1.3.0
v purrr     1.0.1
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```

QTRX_df <- QTRX_df %>%
  dplyr::mutate(ID = row_number())

# moving the ID number to the first column; requires
QTRX_df <- QTRX_df %>%
  dplyr::select(ID, everything())

```

1.5.4 Downsizing the Dataframe

Although researchers may differ in their approach, my tendency is to downsize the df to the variables I will be using in my study. These could include variables in the model, demographic variables, and potentially auxiliary variables (i.e., variables not in the model, but that might be used in the case of multiple imputation).

This particular survey did not collect demographic information, so that will not be used. The model that I will demonstrate in this research vignette examines the the respondent's perceived campus climate for students who are Black, predicted by the the respondent's own campus belonging, and also the *structural diversity* [Lewis and Shah, 2019] proportions of Black students in the classroom and BIPOC (Black, Indigenous, and people of color) instructional staff.

I would like to assess the model by having the instructional staff variable to be the %Black instructional staff. At the time that this lecture is being prepared, there is not sufficient Black representation in the staff to model this.

The `select()` function can let us list the variables we want to retain.

```

# You can use the ':' to include all variables from the first to last
# variable in any sequence; I could have written this more
# efficiently. I just like to 'see' my scales and clusters of
# variables.

Model_df <- (dplyr::select(QTRX_df, ID, iRace1, iRace2, iRace3, iRace4,
  iRace5, iRace6, iRace7, iRace8, iRace9, iRace10, cmBiMulti, cmBlack,
  cmNBPoC, cmWhite, cmUnsure, Belong_1:Belong_3, Blst_1:Blst_6))

```

It can be helpful to save outfile of progress as we go along. Here I save this raw file. I will demonstrate how to save both .rds and .csv files.

```

# to save the df as an .rds (think 'R object') file on your computer;
# it should save in the same file as the .rmd file you are working
# with saveRDS(Model_df, 'BlackStntsModel230902.rds') code to import
# that model we just saved Model_df <-
# readRDS('BlackStntsModel230902.rds')

# write the simulated data as a .csv write.table(Model_df,
# file='BlackStntsModel230902.csv', sep=',', col.names=TRUE,
# row.names=FALSE) bring back the simulated data from a .csv file
# Model_df <- read.csv('BlackStntsModel230902.csv', header = TRUE)

```

1.6 Toward the APA Style Write-up

Because we have been capturing the results as we have worked the problem, our results section is easy to assemble.

1.6.1 Method/Procedure

Data screening suggested that 107 individuals opened the survey link. Of those, 83 granted consent and proceeded into the survey items. A further inclusion criteria was that the course was taught in the U.S; 69 met this criteria.

1.7 Practice Problems

Starting with this chapter, the practice problems for this and the next two chapters (i.e., Scoring, Data Dx) are intended to be completed in a sequence. Whatever practice option(s) you choose, please

- Use raw data that has some missingness (as a last resort you could manually delete some cells),
- Includes at least 3 independent/predictor variables
 - these could be categorically or continuously scaled
 - at least one variable should require scoring.
- Include at least 1 dependent variable
 - at this point in your learning it should be continuously scaled

The three problems below are listed in the order of graded complexity. If you are just getting started, you may wish to start with the first problem. If you are more confident, choose the second or third option. You will likely encounter challenges that were not covered in this chapter. Search for and try out solutions, knowing that there are multiple paths through the analysis.

1.7.1 Problem #1: Rework the Chapter Problem

Because the *Rate-a-Recent-Course* survey remains open, it is quite likely that there will be more participants who have taken the survey since this chapter was last updated. If not – please encourage a peer to take it. Even one additional response will change the results. This practice problem encourages you to rework the chapter, as written, with the updated data from the survey.

1.7.2 Problem #2: Use the *Rate-a-Recent-Course* Survey, Choosing Different Variables

Before starting this option, choose a minimum of three variables from the *Rate-a-Recent-Course* survey to include in a simple statistical model. Work through the chapter making decisions that

are consistent with the research model you have proposed. There will likely be differences at several points in the process. For example, you may wish to include (not exclude) data where the rated-course was offered by an institution outside the U.S. Different decisions may involve an internet search for the R script you will need as you decide on inclusion and exclusion criteria.

1.7.3 Problem #3: Other data

Using raw data for which you have access, use the chapter as a rough guide. Your data will likely have unique characteristics that may involved searching for solutions beyond this chapter/OER.

1.7.4 Grading Rubric

Regardless which option(s) you chose, use the elements in the grading rubric to guide you through the practice.

Assignment Component	Points Possible	Points Earned
1. Specify a research model that includes three predictor variables (continuously or categorically scaled) and one dependent (continuously scaled) variable	5	_____
2. Import data	5	_____
3. Include only those who consented*	5	_____
4. Apply exclusionary criteria *	5	_____
5. Rename variables to be sensible and systematic *	5	_____
6. Downsize the dataframe to the variables of interest	5	_____
7. Provide an APA style write-up of these preliminary steps	5	_____
8. Explanation to grader	5	_____
Totals	40	_____

* If your dataset does not require these steps, please provide example code that uses variables in your dataset. For example, for the inclusion or exclusion criteria, provide an example of how to filter in (or out) any variable on the basis of one of the response options. Once demonstrated, hashtag it out and rerun your script with those commands excluded.

A *homeworked example* for the Scrubbing, Scoring, and DataDx lessons (combined) follows the [Data Dx](#) lesson.

1.8 Bonus Track:

1.8.1 Importing 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

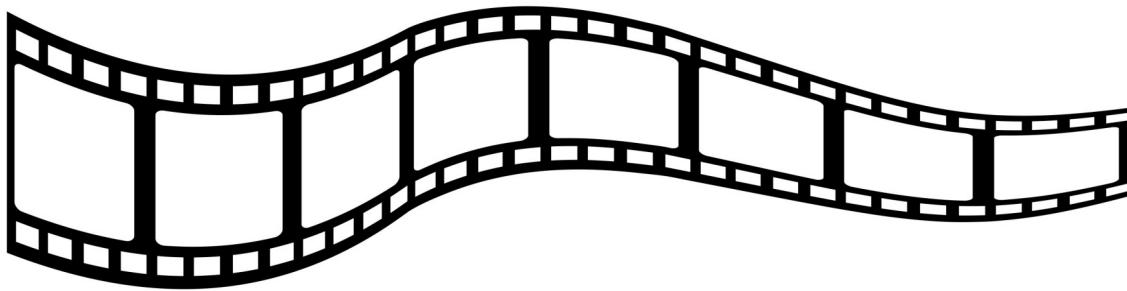


Figure 1.3: Image of a filmstrip

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 idiosyncracy 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_zone = NULL, legacy = FALSE)
```

```
-- 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 = col_number(),
  `^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.
```

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

Chapter 2

Scoring

[Screencasted Lecture Link](#)

The focus of this chapter is to continue the process of scrubbing-and-scoring. We continue with the raw data we downloaded and prepared in the prior chapter. In this chapter we analyze and manage missingness, score scales/subscales, and represent our work with an APA-style write-up. To that end, we will address the conceptual considerations and practical steps in this process.

2.1 Navigating this Lesson

There is about 1 hour and 20 minutes of lecture. If you work through the materials with me it would be good to add another hour.

While the majority of R objects and data you will need are created within the R script that sources the chapter, there are a few 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

Learning objectives from this lecture include the following:

- Recognize the key components of data loss mechanisms (MCAR, MAR, MNAR), including how to diagnose MCAR.
- Interpret missingness figures produced by packages such as *mice*.
- Articulate a workflow for scrubbing and scoring data.
- Use critical data manipulation functions from *dplyr* including *filter()*, *select()*, and *mutate()* to prepare variables.
- Interpret code related to missingness (i.e., “is.na”, “!is.na”) and the pipe (%>%)

2.1.2 Planning for Practice

The suggestions for practice continue from the prior chapter. The assignment in the prior chapter involved downloading a dataset from Qualtrics and the “scrubbing” it on the basis of inclusion

and exclusion criteria. Using that same data, the practice suggestions in this chapter will continue to use Parent's [2013] AIA approach to managing missing data, to score the variables of interest. Options of graded complexity could include:

- Repeating the steps in the chapter with the most recent data from the Rate-A-Recent-Course survey; differences will be in the number of people who have completed the survey since the chapter was written.
- Use the dataset that is the source of the chapter, but score a different set of items that you choose.
- Begin with raw data to which you have access.

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.

- Enders, C. K. (2010). Applied missing data analysis (2010-13190-000). Guilford Press.
 - Enders' text continues to be the comprehensive “go-to” source for examining and managing missing data.
- Kline, R. B. (2016). Data preparation and psychometrics review. In Principles and Practice of Structural Equation Modeling, Fourth Edition. Guilford Publications. <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4000663>
 - Kline’s chapter is my “go-to” for making decisions about preparing data for analysis.
- Parent, M. C. (2013). Handling item-level missing data: Simpler is just as good. *The Counseling Psychologist*, 41(4), 568–600. <https://doi.org/10.1177/0011100012445176>
 - The purpose of Parent’s article was to argue that complex and resource-intensive procedures like multiple imputation are unnecessary. Following a simulation that supports his claims, Parent provides some guidelines to follow for the AIA approach.

2.1.4 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.

```
# if(!require(tidyverse)){install.packages('tidyverse')}
# if(!require(psych)){install.packages('psych')}
# if(!require(mice)){install.packages('mice')}
# if(!require(sjstats)){install.packages('sjstats')}
# if(!require(formattable)){install.packages('formattable')}
```

2.2 Workflow for Scoring

The following is a proposed workflow for preparing data for analysis.

The same workflow guides us through the Scrubbing, Scoring, and Data Dx chapters. At this stage in the chapter we are still scrubbing as we work through the item-level and whole-level portions of the AIA (left side) of the chart.

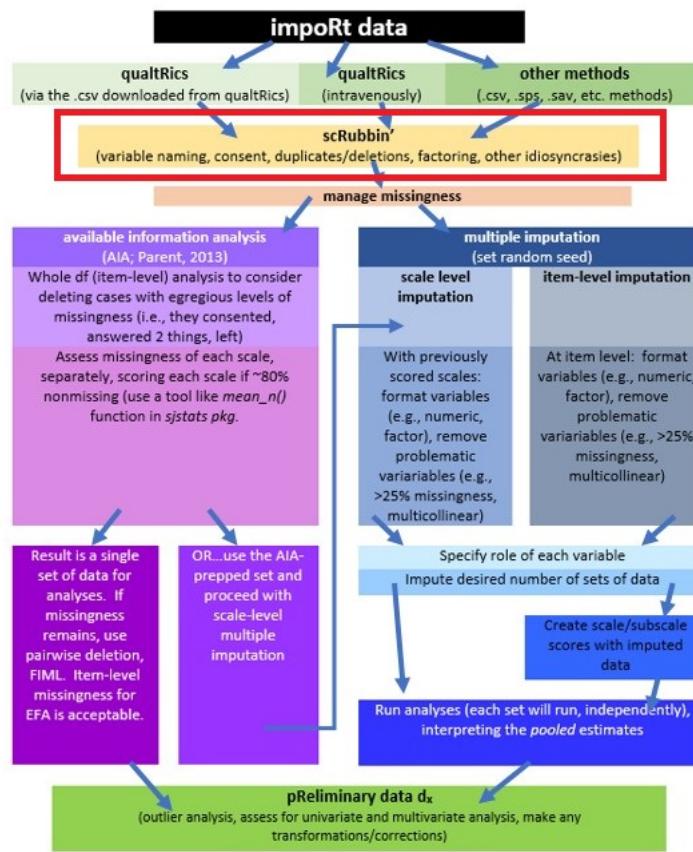


Figure 2.1: An image of our stage in the workflow for scrubbing and scoring data.

2.3 Research Vignette

The research vignette comes from the survey titled, [Rate-a-Recent-Course: A ReCentering Psych Stats Exercise](#) and is explained in the prior chapter. In the prior chapter we conducted super-preliminary scrubbing of variables that will allow us to examine the respondent's perceived campus climate for students who are Black, predicted by the the respondent's own campus belonging, and also the *structural diversity* proportions of Black students in the classroom and the BIPOC instructional staff. At present, I see this as a parallel mediation. That is, the perceived campus climate for Black students will be predicted by the respondent's sense of belonging, through the proportion of Black classmates and BIPOC (Black, Indigenous, and people of color)instructional staff.

I would like to assess the model by having the instructional staff variable to be the percent of Black instructional staff. At the time that this lecture is being prepared, there is insufficient representation of Black faculty to model this.

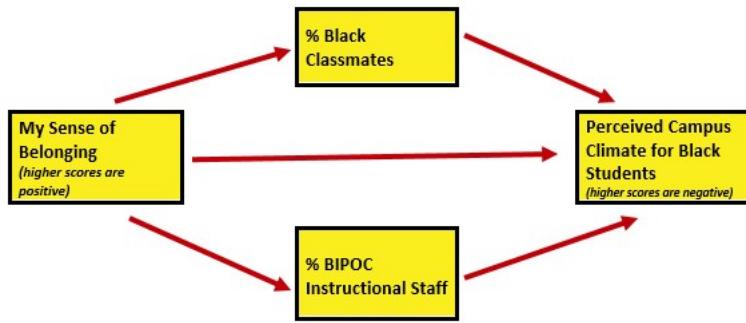


Figure 2.2: An image of the statistical model for which we are preparing data.

First, though, let's take a more conceptual look at issues regarding missing data. We'll come back to details of the survey as we work with it.

2.4 On Missing Data

On the topic of missing data, we follow the traditions in most textbooks. We start by considering *data loss mechanisms* and options for *managing missingness*.

Although the workflow I recommend is fairly straightforward, the topic is not. Quantitative psychologist have produced volumes of research that supports and refutes all of these issues in detail. An in-depth review of this is found in Enders' [2010] text.

2.4.1 Data Loss Mechanisms

We generally classify missingness in data in three different ways [Kline, 2016b, Parent, 2013]:

Missing completely at random (MCAR) is the ideal case (and often unrealistic in actual data). For variable Y this means that

- Missingness is due to a factor(s) completely unrelated to the missing data. Stated another way:
 - Missing observations differ from the observed scores only by chance; that is, whether scores on Y are missing or not missing is unrelated to Y itself
- The presence versus absence of data on Y is unrelated to all other variables in the dataset. That is, the nonmissing data are just a random sample of scores that the researcher would have analyzed had the data been complete. We might think of it as *haphazard* missing.
 - A respondent is interrupted, looks up, looks down, and skips an item.
 - A computer glitch causes spotty missingness – unrelated to any particular variable.

MCAR is the ideal state because results from it should not be biased as a function of the missingness.

Missing at random (MAR) missing data arise from a process that is both measured and predictable in a particular sample. *Admittedly the use of “random” in this term is odd, because, by definition, the missingness is not random.*

Restated:

1. Missingness on Y is unrelated to Y itself, but
2. Missingness is on Y is correlated with other variables in the data set.

Example: Men are less likely to respond to questions about mental health than women, but among men, the probability of responding is unrelated to their true mental health status.

Kline [2016b] indicated that information loss due to MAR is potentially recoverable through imputation where missing scores are replaced by predicted scores. The predicted scores are generated from other variables in the data set that predict missingness on Y. If the strength of that prediction is reasonably strong, then results on Y after imputation may be relatively unbiased. In this sense, the MAR pattern is described as *ignorable* with regard to potential bias. Two types of variables can be used to predict the missing data

1. variables that are in the prediction equation, and
2. *auxiliary* variables (i.e., variables in the dataset that are not in the prediction equation).

Parent [2013] noted that multiple imputation and expectation maximization have frequently been used to manage missingness in MAR circumstances.

Missing not at random (MNAR) is when the presence versus absence of scores on Y depend on Y itself. This is *non-ignorable*.

For example, if a patient drops out of a medical RCT because there are unpleasant side effects from the treatment, this discomfort is not measured, but the data is missing due to a process that is unknown in a particular data set. Results based on *complete cases only* can be severely biased when the data loss pattern is MNAR. That is, a treatment may look more beneficial than it really is if data from patients who were unable to tolerate the treatment are lost.

Parent [2013] described MNAR a little differently – but emphasized that the systematic missingness would be related to a variable outside the dataset. Parent provided the example of items written in a manner that may be inappropriate for some participants (e.g., asking women about a relationship with their boyfriend/husband, when the woman might be in same gender relationship). If there were not demographic items that could identify the bias, this would be MNAR. Parent strongly advises researchers to carefully proofread and pilot surveys to avoid MNAR circumstances.

Kline [2016b] noted that the choice of the method to deal with the incomplete records can make a difference in the results, and should be made carefully.

2.4.2 Diagnosing Missing Data Mechanisms

The bad news is that we never really know (with certainty) the type of missing data mechanism in our data. The following tools can help understand the mechanisms that contribute to missingness.

- Missing data analyses often includes correlations that could predict missingness.
- Little and Rubin [2002] proposed a multivariate statistical test of the MCAR assumption that simultaneously compares complete versus incomplete cases on Y across all other variables. If this comparison is significant, then the MCAR hypothesis is rejected.
 - To restate: we want a non-significant result; and we use the sometimes-backwards-sounding NHST (null hypothesis significance testing) language, “MCAR cannot be rejected.”
- MCAR can also be examined through a series of t tests of the cases that have missing scores on Y with cases that have complete records on other variables. Unfortunately, sample sizes contribute to problems with interpretation. With low samples, they are underpowered; in large samples they can flag trivial differences.

If MCAR is rejected, we are never sure whether the data loss mechanism is MAR or MNAR. There is no magical statistical “fix.” Kline [2016b] wrote, “About the best that can be done is to understand the nature of the underlying data loss pattern and accordingly modify your interpretation of the results” (p. 85).

2.4.3 Managing Missing Data

There are a number of approaches to managing missing data. Here is a summary of the ones most commonly used.

- **Listwise deletion** (aka, Complete Case Analysis) If there is a missing score on any variable, that case is excluded from **all** analyses.
- **Pairwise deletion** Cases are excluded only if they have missing data on variables involved in a particular analysis. AIA is a variant of pair-wise deletion, but it preserves as much data as possible with person-mean imputation at the scale level.
- **Mean/median substitution** Mean/median substitution replaces missing values with the mean/median of that particular variable. While this preserves the mean of the dataset, it can cause bias by decreasing variance. For example, if you have a column that has substantial of missingness and you replace each value with the same, fixed, mean, the variability of that variable has just been reduced. A variation on this is a **group-mean substitution** where the missing score in a particular group (e.g., women) is replaced by the group mean.
- **Full information maximum likelihood (FIML)** A *model-based method* that takes the researcher’s model as the starting point. The procedure partitions the cases in a raw data file into subsets, each with the same pattern of missing observations, including none (complete cases). Statistical information (e.g., means, variances) is extracted from each subset so all case are retained in the analysis. Parameters for the researcher’s model are estimated after combining all available information over the subsets of cases.
- **Multiple imputation** A *data based method* that works with the whole raw data file (not just with the observed variables that comprise the researcher’s model). Multiple imputation assumes that data are MAR (remember, MCAR is the more prestigious one). This means that researchers assume that missing values can be replaced by predictions derived from the observable portion fo the dataset.

- Multiple datasets (often 5 to 20) are created where missing values are replaced via a randomized process (so the same missing value [item 4 for person A] will likely have different values for each dataset).
- The desired analysis(es) is conducted simultaneously/separately for each of the imputed sets (so if you imputed 5 sets and wanted a linear regression, you get 5 linear regressions).
- A *pooled analysis* uses the point estimates and the standard errors to provide a single result that represents the analysis.

2.4.4 Available Information Analysis (AIA)

Parent [2013] has created a set of recommendations that help us create a streamlined workflow for managing missing data. After evaluating three approaches to managing missingness (AIA, mean substitution, and multiple imputation) Parent concluded that in datasets with (a) low levels of missingness, (b) a reasonable sample size, and (c) adequate internal reliability of measures, these approaches had similar results.

Further, in simulation studies where there was (a) low sample size ($n = 50$), (b) weak associations among items, and (c) a small number of missing items, AIA was equivalent to multiple imputation. Even in cases where the data conditions were the “best” (i.e., $N = 200$, moderate correlations, at least 10 items), even 10% missingness (overall) did not produce notable difference among the methods. That is, means, standard errors, and alphas were similar across the methods (AIA, mean substitution, multiple imputation).

AIA is an older method of handling missing data that, as its name suggests, uses the *available data* for analysis and excludes missing data points only for analyses in which the missing data point would be directly involved. This means

- In the case of research that uses multiple item scales, and analysis takes place at the scale level
 - AIA is used to generate **mean** scores for the scale using the available data without substituting or imputing values;
 - This method generally produces a fairly complete set of scale-level data where
 - * pairwise deletion (the whole row/case/person is skipped) can be used where there will be multiple analyses using statistics (e.g., correlations, t-tests, ANOVA) where missingness is not permitted
 - * FIML can be specified in path analysis and CFA/SEM (where item-level data is required), and
 - * some statistics, such as principal components analysis and principal axis factoring (item-level analyses) permit missing data,
 - Of course, the researcher could still impute data, but why...

Parent's [2013] recommendations:

- Scale scores should be first calculated as a *mean* (average) not a sum. Why?
 - Calculating a “sum” from available data will result in automatically lower scores in cases where there is missingness.

- If a sum is required (i.e., because you want to interpret some clinical level of something), calculate the mean first, do the analyses, then transform the results back into the whole-scale equivalent (multiply the mean by the number of items) for any interpretation.
- For R script, do not write the script ($[item1 + item2 + item3]/3$) because this will return an empty entry for participants missing data (same problem as if you were to use sum). There are several functions for properly computing a mean; I will demo the *mean_n()* function from *sjstats* package because it allows us to simultaneously specify the tolerance level (next item).
- Determine your *tolerance* for missingness (20% seems to be common, although you could also look for guidance in the test manual/article). Then
 - Run a “percent missingness” check on the level of analysis (i.e., total score, scale, or subscale) you are using. If you are using a total scale score, then check to see what percent is missing across all the items in the whole scale. In contrast, if you are looking at subscales, run the percent missing at that level.
 - Parent [2013] advised that the tolerance levels should be made mindfully. A four-item scale with one item missing, won’t meet the 80% threshold, so it may make sense to set a 75% threshold for this scale.
- “Clearly and concisely detail the level of missingness” in papers [Parent, 2013, p. 595]. This includes
 - tolerance level for missing data by scale or subscale (e.g., 80% or 75%)
 - the number of missing values out of all data points on that scale for all participants and the maximum by participant (e.g., “For Scale X, a total of # missing data points out of ### were observed with no participant missing more than a single point.”)
 - verify a manual inspection of missing data for obvious patterns (e.g., abnormally high missing rates for only one or two items). This can be accomplished by requesting frequency output for the items and checking the nonmissing data points for each scale, ensuring there are no abnormal spikes in missingness (looking for MNAR).
- Curiously, Parent [2013] does not recommend that we run all the diagnostic tests. However, because recent reviewers have required them of me, I will demonstrate a series of them.
- Reducing missingness starts at the survey design – make sure that all people can answer all items (i.e., relationship-related items may contain heterosexist assumptions...which would result in an MNAR circumstance)

Very practically speaking, Parent’s [2013] recommendations follow us through the entire data analysis process.

2.5 Working the Problem

2.5.1 Variable Planning and Preparation

In the **Scrubbing lesson** we imported the data from Qualtrics and applied the broadest levels of inclusion (e.g., the course rated was offered from an institution in the U.S., the respondent consented to participation) and exclusion (e.g., the survey was not a preview). We then downsized the survey

to include the variables we will use in our statistical model. We then saved the data in .csv and .rds file.

Presuming that you are working along with me in an .rmd file and have placed that file in the same folder as this .rmd file, the following code should read the data into your environment.

I use *different* names for the object/df in my R environment than I use for the filename that holds the data on my computer. Why? I don't want to accidentally overwrite this precious "source" of data.

```
# scrub_df <- read.csv ('BlackStntsModel230902.csv', head = TRUE, sep
# = ',')
scrub_df <- readRDS("BlackStntsModel230902.rds")
str(scrub_df)

## Classes 'tbl_df', 'tbl' and 'data.frame': 69 obs. of 25 variables:
## $ ID      : int 1 2 3 4 5 6 7 8 9 10 ...
## $ iRace1   : num 3 3 3 3 1 3 3 3 1 0 ...
## ..- attr(*, "label")= Named chr "1 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "1_iRace"
## $ iRace2   : num 1 NA 1 1 NA NA 3 NA NA 0 ...
## ..- attr(*, "label")= Named chr "2 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "2_iRace"
## $ iRace3   : num 3 NA NA 3 NA NA NA NA NA 3 ...
## ..- attr(*, "label")= Named chr "3 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "3_iRace"
## $ iRace4   : num NA NA NA NA NA NA NA NA NA 3 ...
## ..- attr(*, "label")= Named chr "4 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "4_iRace"
## $ iRace5   : logi NA NA NA NA NA NA NA ...
## ..- attr(*, "label")= Named chr "5 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "5_iRace"
## $ iRace6   : logi NA NA NA NA NA NA NA ...
## ..- attr(*, "label")= Named chr "6 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "6_iRace"
## $ iRace7   : logi NA NA NA NA NA NA NA ...
## ..- attr(*, "label")= Named chr "7 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "7_iRace"
## $ iRace8   : logi NA NA NA NA NA NA NA ...
## ..- attr(*, "label")= Named chr "8 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "8_iRace"
## $ iRace9   : logi NA NA NA NA NA NA NA ...
## ..- attr(*, "label")= Named chr "9 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "9_iRace"
## $ iRace10  : logi NA NA NA NA NA NA NA ...
## ..- attr(*, "label")= Named chr "10 - From your perspective as a student, which of the following ...
## ...- attr(*, "names")= chr "10_iRace"
## $ cmBiMulti: num 0 0 0 2 5 15 0 0 0 7 ...
```

```

## ..- attr(*, "label")= Named chr "Regarding race, what proportion of students were from each race group"
## ... - attr(*, "names")= chr "Race_10"
## $ cmBlack : num 0 5 10 6 5 20 0 0 0 4 ...
## ..- attr(*, "label")= Named chr "Regarding race, what proportion of students were from each race group"
## ... - attr(*, "names")= chr "Race_1"
## $ cmNBPoC : num 39 10 30 19 10 30 40 5 30 13 ...
## ..- attr(*, "label")= Named chr "Regarding race, what proportion of students were from each race group"
## ... - attr(*, "names")= chr "Race_7"
## $ cmWhite : num 61 85 60 73 80 35 60 90 70 73 ...
## ..- attr(*, "label")= Named chr "Regarding race, what proportion of students were from each race group"
## ... - attr(*, "names")= chr "Race_8"
## $ cmUnsure : num 0 0 0 0 0 0 0 5 0 3 ...
## ..- attr(*, "label")= Named chr "Regarding race, what proportion of students were from each race group"
## ... - attr(*, "names")= chr "Race_2"
## $ Belong_1 : num 6 4 NA 5 4 5 6 7 6 3 ...
## ..- attr(*, "label")= Named chr "Please indicate the degree to which you agree with the following statement"
## ... - attr(*, "names")= chr "Belong_1"
## $ Belong_2 : num 6 4 3 3 4 6 6 7 6 3 ...
## ..- attr(*, "label")= Named chr "Please indicate the degree to which you agree with the following statement"
## ... - attr(*, "names")= chr "Belong_2"
## $ Belong_3 : num 7 6 NA 2 4 5 5 7 6 3 ...
## ..- attr(*, "label")= Named chr "Please indicate the degree to which you agree with the following statement"
## ... - attr(*, "names")= chr "Belong_3"
## $ Blst_1 : num 5 6 NA 2 6 5 5 5 5 3 ...
## ..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
## ... - attr(*, "names")= chr "Blst_1"
## $ Blst_2 : num 3 6 5 2 1 1 4 4 3 5 ...
## ..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
## ... - attr(*, "names")= chr "Blst_2"
## $ Blst_3 : num 5 2 2 2 1 1 4 3 1 2 ...
## ..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
## ... - attr(*, "names")= chr "Blst_3"
## $ Blst_4 : num 2 2 2 2 1 2 4 3 2 3 ...
## ..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
## ... - attr(*, "names")= chr "Blst_4"
## $ Blst_5 : num 2 4 NA 2 1 1 4 4 1 3 ...
## ..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
## ... - attr(*, "names")= chr "Blst_5"
## $ Blst_6 : num 2 1 2 2 1 2 4 3 2 3 ...
## ..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
## ... - attr(*, "names")= chr "Blst_6"
## - attr(*, "column_map")=Classes 'tbl_df', 'tbl' and 'data.frame': 182 obs. of 7 variables
##   ..$ qname      : chr [1:182] "StartDate" "EndDate" "Status" "Progress" ...
##   ..$ description: chr [1:182] "Start Date" "End Date" "Response Type" "Progress" ...
##   ..$ main       : chr [1:182] "Start Date" "End Date" "Response Type" "Progress" ...
##   ..$ sub        : chr [1:182] "" "" "" "" ...
##   ..$ ImportId   : chr [1:182] "startDate" "endDate" "status" "progress" ...
##   ..$ timeZone   : chr [1:182] "America/Los_Angeles" "America/Los_Angeles" NA NA ...

```

```
## ..$ choiceId : chr [1:182] NA NA NA NA ...
```

Let's think about how the variables in our model should be measured:

- DV: Campus Climate for Black Students (as perceived by the respondent)
 - mean score of the 6 items on that scale (higher scores indicate a climate characterized by hostility, nonresponsiveness, and stigma)
 - 1 item needs to be reverse-coded
 - this scale was adapted from the LGBT Campus Climate Scale [[Szymanski and Bissonette, 2020](#)]
- IV: Belonging
 - mean score for the 3 items on that scale (higher scores indicate a greater sense of belonging)
 - this scale is taken from the Sense of Belonging subscale from the Perceived Cohesion Scale [[Bollen and Hoyle, 1990](#)]
- Proportion of classmates who are Black
 - a single item
- Proportion of instructional staff who are BIPOC
 - must be calculated from each of the single items for each instructor

To summarize, the Campus Climate and Belonging scales are traditional in the sense that they have items that we sum. The variable representing proportion of classmates who are Black is a single item. The variable representing the proportion of instructional staff who are BIPOC must be calculated in a manner that takes into consideration the there may be multiple instructors. The survey allowed a respondent to name up to 10 instructors.

```
str(scrub_df$iRace1)
```

```
## num [1:69] 3 3 3 3 1 3 3 3 1 0 ...
## - attr(*, "label")= Named chr "1 - From your perspective as a student, which of the follow
##   ..- attr(*, "names")= chr "1_iRace"
```

Looking at the structure of our data, the iRace(1 thru 10) variables are in “int” or integer format. This means that they are represented as whole numbers. We need them to be represented as factors. R handles factors represented as words well. Therefore, let's use our codebook to reformat this variable as a an ordered factor, with words instead of numbers.

Qualtrics imports many of the categorical variables as numbers. R often reads them numerically (integers or numbers). If they are directly converted to factors, R will sometimes collapse across missing numbers. In this example, if there is a race that is not represented (e.g., 2 for BiMulti), when the numbers are changed to factors, R will assume they are ordered and there is a consecutive series of numbers (0,1,2,3,4). If a number in the sequence is missing (0,1,3,4) and labels are applied, it will collapse across the numbers and the labels you think are attached to each number are not.

Therefore, it is ESSENTIAL to check (again and again ad nauseum) to ensure that your variables are recoding in a manner you understand.

One way to avoid this is to use the code below to identify the levels and the labels. When they are in order, they align and don't "skip" numbers. To quadruple check our work, we will recode into a new variable "tRace#" for "teacher" Race.

```
scrub_df$tRace1 = factor(scrub_df$iRace1, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace2 = factor(scrub_df$iRace2, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace3 = factor(scrub_df$iRace3, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace4 = factor(scrub_df$iRace4, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace5 = factor(scrub_df$iRace5, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace6 = factor(scrub_df$iRace6, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace7 = factor(scrub_df$iRace7, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace8 = factor(scrub_df$iRace8, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace9 = factor(scrub_df$iRace9, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
scrub_df$tRace10 = factor(scrub_df$iRace10, levels = c(0, 1, 2, 3, 4),
  labels = c("Black", "nBpoc", "BiMulti", "White", "NotNotice"))
```

Let's check the structure to see if they are factors.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyrr    1.3.0
## v purrr    1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (http://conflicted.r-lib.org/) to force all conflicts to be
```

```
glimpse(scrub_df)
```

```
## Rows: 69  
## Columns: 35
```

```

## $ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1~
## $ iRace1  <dbl> 3, 3, 3, 3, 1, 3, 3, 1, 0, 2, 1, 1, 1, 3, 3, 3, 1, 3, 3, ~
## $ iRace2  <dbl> 1, NA, 1, 1, NA, NA, 3, NA, NA, 0, NA, NA, 3, NA, 3, 3, NA, ~
## $ iRace3  <dbl> 3, NA, NA, 3, NA, NA, NA, NA, NA, 3, NA, NA, NA, NA, 3, 1, N~
## $ iRace4  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 3, NA, NA, NA, NA, NA, 3~
## $ iRace5  <lgl> NA, ~
## $ iRace6  <lgl> NA, ~
## $ iRace7  <lgl> NA, ~
## $ iRace8  <lgl> NA, ~
## $ iRace9  <lgl> NA, ~
## $ iRace10 <lgl> NA, ~
## $ cmBiMulti <dbl> 0, 0, 0, 2, 5, 15, 0, 0, 0, 7, 0, 0, 20, 0, 9, 12, 0, 6, 6, ~
## $ cmBlack   <dbl> 0, 5, 10, 6, 5, 20, 0, 0, 0, 4, 0, 7, 0, 6, 9, 1, 21, 5, 6, ~
## $ cmNBPoC   <dbl> 39, 10, 30, 19, 10, 30, 40, 5, 30, 13, 80, 19, 0, 19, 15, 22~
## $ cmWhite   <dbl> 61, 85, 60, 73, 80, 35, 60, 90, 70, 73, 10, 74, 80, 0, 67, 5~
## $ cmUnsure  <dbl> 0, 0, 0, 0, 0, 0, 5, 0, 3, 10, 0, 0, 75, 0, 14, 0, 5, 0, ~
## $ Belong_1  <dbl> 6, 4, NA, 5, 4, 5, 6, 7, 6, 3, 6, 6, 3, 4, 3, 3, 4, 5, 1, 2, ~
## $ Belong_2  <dbl> 6, 4, 3, 3, 4, 6, 6, 7, 6, 3, 6, 6, 5, 4, 3, 3, 4, 6, 1, 2, ~
## $ Belong_3  <dbl> 7, 6, NA, 2, 4, 5, 5, 7, 6, 3, 5, 6, 4, 4, 3, 2, 4, 5, 1, 1, ~
## $ Blst_1    <dbl> 5, 6, NA, 2, 6, 5, 5, 5, 5, 3, NA, 4, 5, 6, 3, 4, 6, 4, 4, 4~
## $ Blst_2    <dbl> 3, 6, 5, 2, 1, 1, 4, 4, 3, 5, NA, 5, 1, 1, 3, 2, 1, 2, 5, 3, ~
## $ Blst_3    <dbl> 5, 2, 2, 2, 1, 1, 4, 3, 1, 2, 2, 1, 1, 1, 3, 2, 6, 2, 2, 2, ~
## $ Blst_4    <dbl> 2, 2, 2, 2, 1, 2, 4, 3, 2, 3, NA, 4, 3, 1, 3, 2, 1, 3, 2, 1, ~
## $ Blst_5    <dbl> 2, 4, NA, 2, 1, 1, 4, 4, 1, 3, 2, 2, 1, 1, 3, 2, 1, 2, 2, 1, ~
## $ Blst_6    <dbl> 2, 1, 2, 2, 1, 2, 4, 3, 2, 3, NA, 2, 1, 1, 3, 2, 2, 3, 2, 1, ~
## $ tRace1   <fct> White, White, White, White, nBpoc, White, White, White, nBpo~
## $ tRace2   <fct> nBpoc, NA, nBpoc, nBpoc, NA, NA, White, NA, NA, Black, NA, N~
## $ tRace3   <fct> White, NA, NA, White, NA, NA, NA, NA, White, NA, NA, NA, ~
## $ tRace4   <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, White, NA, NA, NA, NA, N~
## $ tRace5   <fct> NA, ~
## $ tRace6   <fct> NA, ~
## $ tRace7   <fct> NA, ~
## $ tRace8   <fct> NA, ~
## $ tRace9   <fct> NA, ~
## $ tRace10  <fct> NA, ~

```

Calculating the proportion of the BIPOC instructional staff could likely be accomplished a number of ways. My searching for solutions resulted in this. Hopefully it's a fair balance between intuitive and elegant coding. First, I created code that

- created a new variable (count.BIPOC) by
 - summing across the tRace1 through tRace10 variables,
 - assigning a count of “1” each time the factor value was Black, nBpoc, or BiMulti

```

scrub_df$count.BIPOC <- apply(scrub_df[c("tRace1", "tRace2", "tRace3",
                                         "tRace4", "tRace5", "tRace6", "tRace7", "tRace8", "tRace9", "tRace10")],
                                1, function(x) sum(x %in% c("Black", "nBpoc", "BiMulti")))

```

Next, I created a variable that counted the number of non-missing values across the tRace1 through tRace10 variables.

```
scrub_df$count.nMiss <- apply(scrub_df[c("tRace1", "tRace2", "tRace3",
  "tRace4", "tRace5", "tRace6", "tRace7", "tRace8", "tRace9", "tRace10")],
  1, function(x) sum(!is.na(x)))
```

Now to calculate the proportion of BIPOC instructional faculty for each case.

```
scrub_df$iBIPOC_pr = scrub_df$count.BIPOC/scrub_df$count.nMiss
```

2.5.2 Missing Data Analysis: Whole df and Item level

In understanding missingness across the dataset, I think it is important to analyze and manage it, iteratively. We will start with a view of the whole df-level missingness. Subsequently, and consistent with the available information analysis [AIA; Parent [2013]] approach, we will score the scales and then look again at missingness, using the new information to update our decisions about how to manage it.

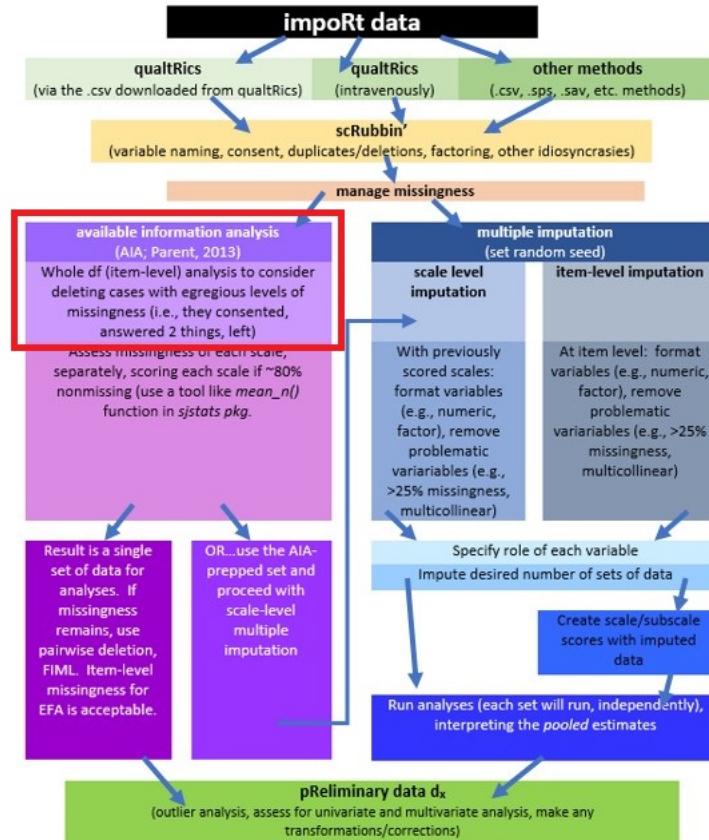


Figure 2.3: An image of our stage in the workflow for scrubbing and scoring data.

Because we just created a host of new variables in creating the *prop_BIPOC* variable, let's downsize the df so that the calculations are sensible.

```
scrub_df <- (select(scrub_df, ID, iBIPOC_pr, cmBlack, Belong_1:Belong_3,
  Blst_1:Blst_6))
```

With a couple of calculations, we create a proportion of item-level missingness.

In this chunk I first calculate the number of missing (nmiss)

```
library(tidyverse)
#Calculating number and proportion of item-level missingness
scrub_df$nmiss <- scrub_df%>%
  select(iBIPOC_pr:Blst_6) %>% #the colon allows us to include all variables between the two
  is.na %>%
  rowSums

scrub_df<- scrub_df%>%
  dplyr::mutate(prop_miss = (nmiss/11)*100) #11 is the number of variables included in calcula
```

We can grab the descriptives for the *prop_miss* variable to begin to understand our data. I will create an object from it so I can use it with inline

```
psych::describe(scrub_df$prop_miss)
```

```
##      vars   n  mean      sd median trimmed mad min max range skew kurtosis    se
## X1      1 69 7.77 22.61       0     1.59    0    0 100    100 3.04      8.19 2.72
```

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%.

At the time that I am lecturing this, we do have some rather egregious missingness. At this point I will write code to eliminate cases with $\geq 90\%$.

```
scrub_df <- dplyr::filter(scrub_df, prop_miss <= 90) #update df to have only those with at le
```

To analyze missingness at this level, we need a df that has only the variables of interest. That is, variables like *ID* and the *prop_miss* and *nmiss* variables we created will interfere with an accurate assessment of missingness. I will update our df to eliminate these.

```
# further update to exclude the n_miss and prop_miss variables
scrub_df <- scrub_df %>%
  dplyr::select(-c(ID, nmiss, prop_miss))
```

Missing data analysis commonly looks at proportions by:

- the entire df
- rows/cases/people

```
# what proportion of cells missing across entire dataset
formattable::percent(mean(is.na(scrub_df)))
```

```
## [1] 3.86%
```

```
# what proportion of cases (rows) are complete (nonmissing)
formattable::percent(mean(complete.cases(scrub_df)))
```

```
## [1] 87.88%
```

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

2.5.3 Analyzing Missing Data Patterns

One approach to analyzing missing data is to assess patterns of missingness.

Several R packages are popularly used for conducting such analyses. In the *mice* package, *md.pattern()* function provides a matrix with the number of columns + 1, in which each row corresponds to a missing data pattern (1 = observed, 0 = missing).

Rows and columns are sorted in increasing amounts of missing information.

The last column and row contain row and column counts, respectively.

```
mice_out <- mice::md.pattern(scrub_df, plot = TRUE, rotate.names = TRUE)
mice_out
write.csv(mice_out, file = "mice_out.csv") #optional to write it to a .csv file
```

The table lets us examine each missing pattern and see which variable(s) is/are missing. The output is in the form of a table that indicates the frequency of each pattern of missingness. Because I haven't (yet) figured out how to pipe objects from this table into the chapter, this text may differ from the patterns in the current data frame.

Each row in the table represents a different pattern of missingness. At the time of writing, there are 8 patterns of missing data. The patterns are listed in descending order of the least amount of missingness. The most common pattern (58 cases, top row) is one with no missing data. One case is missing one cell – one item assessing the campus climate for Black students, and so forth.

2.5.4 Can we identify the missing mechanisms?

To date, we do not have statistical tools that can accurately diagnose our patterns of missingness. You may have heard that “Little’s MCAR” is a helpful tool. Unfortunately, as Enders [2010] has noted, the tool is problematic. Perhaps the most significant one is that under the null hypothesis, a statistically significant test indicates that the missing data are MAR (missing at random) or MNAR (missing not at random); a non-significant test indicates the data are MCAR (missing completely at random) or MNAR. Consequently, regardless of the result, an MNAR circumstance cannot be ruled out. Correspondingly, the Little’s MCAR test has disappeared from the more reliable R packages that assess missingness.

Enders [2010] *Applied Missing Data Analysis* text does provide a set of [figures](#) (page 3) that illustrate common missing data patterns. Comparing these to the figure produced with *mice::mdpattern* our data looks somewhat monotonic – that is, as individuals completed the survey, they began to experience test fatigue and simply stopped responding. Diagnosing monotonicity requires that the variables in the dataset must be in the order in which the students completed them. If the variables have been re-ordered or if the surveys were presented to students in a randomized order, then more data manipulation would be required before attributing missingness to test fatigue.

Survey programs like Qualtrics offer the randomization of items within blocks (or blocks themselves). This can help distribute missingness caused by test fatigue so that more cases can be retained.

2.6 Scoring

So let’s get to work to score up the measures for our analysis. Each step of this should involve careful cross-checking with the [codebook](#).

2.6.1 Reverse scoring

As we discovered previously, in the scale that assesses campus climate (higher scores reflect a more negative climate) one of our items (Blst_1, “My *institution* provides a supportive environment for Black students.”) requires reverse-coding.

To rescore:

- Create a *new* variable (this is essential) that is designated as the reversed item. We might put a the letter “r” (for reverse scoring) at the beginning or end: rBlst_1 or Blst_1r. It does not matter; just be consistent.
 - We don’t reverse score into the same variable because when you rerun the script, it just re-reverses the reversed score...into infinity. It’s very easy to lose your place.
- The reversal is an *equation* where you subtract the value in the item from the range/scaling + 1. For the our three items we subtract each item’s value from 8.

```
scrub_df <- scrub_df %>%
  dplyr::mutate(rBlst_1 = 8 - Blst_1) #if you had multiple items, you could add a pipe (%>%)
```

Per Parent [2013] we will analyze missingness for each scale, separately.

- We will calculate scale scores on each scale separately when 80% (roughly) of the data is present.
 - this is somewhat arbitrary, on 4 item scales, I would choose 75% (to allow one to be missing)
 - on the 3 item scale, I will allow one item to be missing (65%)
- After calculating the scale scores, we will return to analyzing the missingness, looking at the whole df

The *mean_n()* function of *sjstats* package has allows you to specify how many items (whole number) or what percentage of items should be present in order to get the mean. First, though, we should identify the variables (properly formatted, if rescoreing was needed) that should be included in the calculation of each scale and subscale.

In our case, the scale assessing belonging [Bollen and Hoyle, 1990, Hurtado and Carter, 1997] involves three items with no reversals. Our campus climate scale was adapted from Szymanski et al.'s LGBTQ College Campus Climate Scale [Szymanski and Bissonette, 2020]. While it has not been psychometrically evaluated for the purpose for which I am using it, I will follow the scoring structure in the journal article that introduces the measure. Specifically, the factor structure permits a total scale score and two subscales representing the college response and stigma.

```
# Making the list of variables
Belonging_vars <- c("Belong_1", "Belong_2", "Belong_3")
ResponseBL_vars <- c("rBlst_1", "Blst_4", "Blst_6")
StigmaBL_vars <- c("Blst_2", "Blst_3", "Blst_5")
ClimateBL_vars <- c("rBlst_1", "Blst_4", "Blst_6", "Blst_2", "Blst_3",
                     "Blst_5")

# Creating the new variables
scrub_df$Belonging <- sjstats::mean_n(scrub_df[, Belonging_vars], 0.65)
scrub_df$ResponseBL <- sjstats::mean_n(scrub_df[, ResponseBL_vars], 0.8)
scrub_df$StigmaBL <- sjstats::mean_n(scrub_df[, StigmaBL_vars], 0.8)
scrub_df$ClimateBL <- sjstats::mean_n(scrub_df[, ClimateBL_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfLewis$Belonging <-
# sjstats::mean_n(dfLewis[, ..Belonging_vars], 0.80)
```

Later it will be helpful to have a df with the item and scale-level variables. It will also be helpful if there is an ID for each case.

```
scrub_df <- scrub_df %>%
  dplyr::mutate(ID = row_number())
```

```
# moving the ID number to the first column; requires
scrub_df <- scrub_df %>%
  dplyr::select(ID, everything())
```

Let's save our *scrub_df* data for this and write it as an outfile. I will save it in both .rds and .csv formats so that you can use either one.

```
write.table(scrub_df, file = "BlStItmsScrs230902.csv", sep = ",",
            row.names = FALSE)
saveRDS(scrub_df, "BlStItmsScrs230902.rds")
```

2.7 Missing Analysis: Scale level

Let's return to analyzing the missingness, this time including the *scale level* variables (without the individual items) that will be in our statistical model(s).

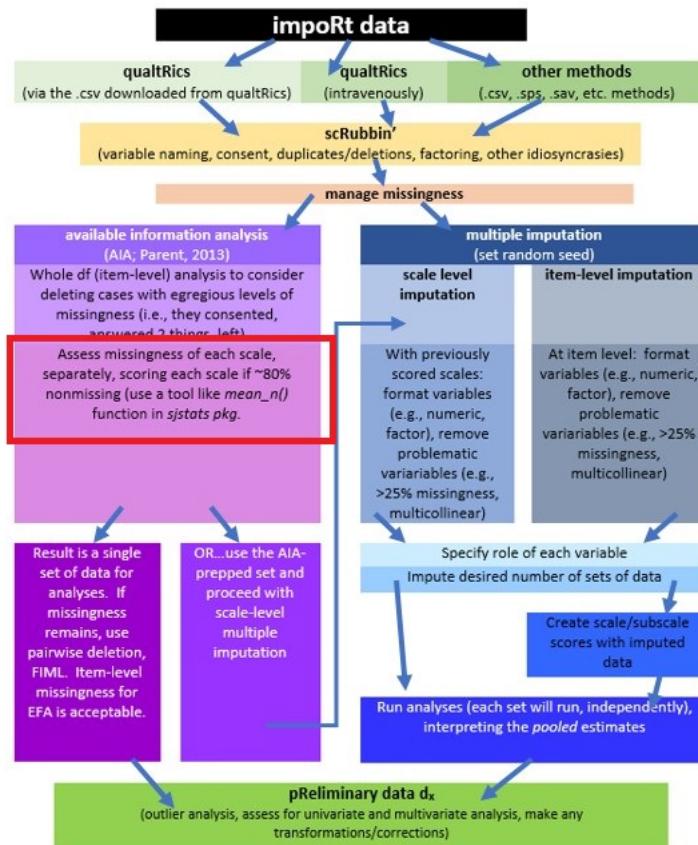


Figure 2.4: An image of our stage in the workflow for scrubbing and scoring data.

First let's get the df down to the variables we want to retain:

```

scored <- dplyr::select(scrub_df, iBIPOC_pr, cmBlack, Belonging, ResponseBL,
    StigmaBL, ClimateBL)
ScoredCaseMiss <- nrow(scored) #I produced this object for the sole purpose of feeding the nu
ScoredCaseMiss

## [1] 66

```

Before we start our formal analysis of missingness at the scale level, let's continue to scrub by eliminating cases that will have too much missingness. In the script below we create a variable that counts the number of missing variables and then creates a proportion by dividing it by the number of total variables.

Using the *describe()* function from the *psych* package, we can investigate this variable.

```

# Create a variable (n_miss) that counts the number missing
scored$n_miss <- scored %>%
    dplyr::select(iBIPOC_pr:ClimateBL) %>%
    is.na %>%
    rowSums

# Create a proportion missing by dividing n_miss by the total number
# of variables (6) Pipe to sort in order of descending frequency to
# get a sense of the missingness
scored <- scored %>%
    dplyr::mutate(prop_miss = (n_miss/6) * 100) %>%
    arrange(desc(n_miss))

psych::describe(scored$prop_miss)

##      vars   n  mean      sd median trimmed  mad  min   max range skew kurtosis     se
## X1     1 66  3.79 12.33       0    0.31    0   0 66.67 66.67 3.44     11.77 1.52

```

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

Across the 66 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%.

We need to decide what is our retention threshhold. Twenty percent seems to be a general rule of thumb. Let's delete all cases with missingness at 20% or greater.

```

# update df to have only those with at least 20% of complete data
# (this is an arbitrary decision)
scored <- dplyr::filter(scored, prop_miss <= 20)

# the variable selection just lops off the proportion missing
scored <- (select(scored, iBIP0C_pr:ClimateBL))

# this produces the number of cases retained
nrow(scored)

## [1] 61

```

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

Across the 66 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 61 cases.

With a decision about the number of cases we are going to include, we can continue to analyze missingness.

2.8 Revisiting Missing Analysis at the Scale Level

We work with a df that includes only the variables in our model. In our case this is easy. In other cases (i.e., maybe there is an ID number) it might be good to create a subset just for this analysis.

Again, we look at missingness as the proportion of

- individual cells across the scored dataset, and
- rows/cases with nonmissing data

```

# percent missing across df
formattable::percent(mean(is.na(scored)))

## [1] 0.55%

# percent of rows with nonmissing data
formattable::percent(mean(complete.cases(scored)))

## [1] 96.72%

```

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

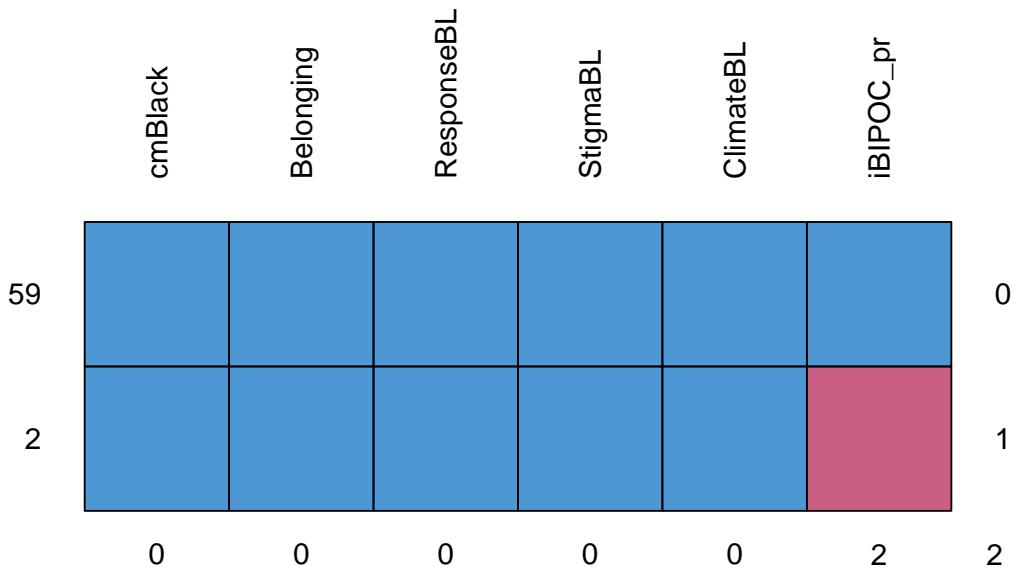
Across the 66 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 61 cases. In this dataset we had less than 1% (0.55%) missing across the df; 97% of the rows had nonmissing data.

Let's look again at missing patterns and mechanisms.

2.8.1 Scale Level: Patterns of Missing Data

Returning to the *mice* package, we can use the *md.pattern()* function to examine a matrix with the number of columns + 1 in which each row corresponds to a missing data pattern (1 = observed, 0 = missing). The rows and columns are sorted in increasing amounts of missing information. The last column and row contain row and column counts, respectively.

```
mice_ScaleLvl <- mice::md.pattern(scored, plot = TRUE, rotate.names = TRUE)
```



```
mice_ScaleLvl
```

```
##      cmBlack Belonging ResponseBL StigmaBL ClimateBL iBIPOC_pr
## 59      1        1        1        1        1        1 0
## 2       1        1        1        1        1        0 1
##      0        0        0        0        0        2 2
```

At the scale-level, this is much easier to interpret. There are 2 rows of data because there are only 2 patterns of missingness. The most common pattern is non-missing data ($n = 59$).

If our statistical choice uses listwise deletion (i.e., the case is eliminated if one or more variables in the model has missing data), our sample size will be 59. As we will learn in later chapters, there are alternatives (i.e., specifying a FIML option in analyses that use maximum likelihood estimators) that can use all of the cases – even those with missing data.

2.8.2 R-ready for Analysis

At this stage the data is ready for analysis (data diagnostics). With the AIA approach [Parent, 2013] the following preliminary analyses would involve pairwise deletion (i.e., the row/case is dropped for that analysis, but included for all others):

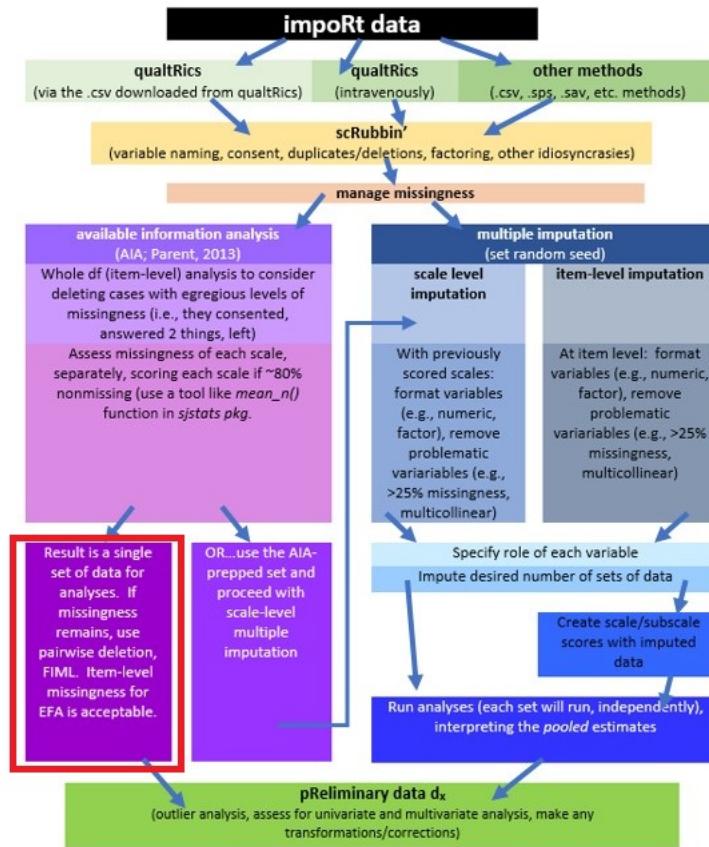


Figure 2.5: An image of our stage in the workflow for scrubbing and scoring data.

- data diagnostics
 - psychometric properties of scales, such as alpha coefficients
 - assessing assumptions such as univariate and multivariate normality, outliers, etc.
- preliminary analyses
 - descriptives (means/standard deviations, frequencies)
 - correlation matrices

AIA can also be used with primary analyses. Examples of how to manage missingness include:

- ANOVA/regression models
 - if completed with ordinary least squares, pairwise deletion would be utilized

- SEM/CFA models with observed, latent, or hybrid models
 - if FIML (we'll discuss later) is specified, all cases are used, even when there is missingness
- EFA models
 - these can handle item-level missingness
- Hierarchical linear modeling/multilevel modeling/mixed effects modeling
 - While all data needs to be present for a given cluster/wave, it is permissible to have varying numbers of clusters/waves per case

2.9 The APA Style Write-Up

2.10 Results

All analyses were completed in R Studio (v. RStudio 2023.06.1+524 “Mountain Hydrangea”) with R (v. 4.3.1).

Missing Data Analysis and Treatment of Missing Data

Available item analysis (AIA; [Parent, 2013]) is a strategy for managing missing data that uses available data for analysis and excludes cases with missing data points only for analyses in which the data points would be directly involved. Parent (2013) suggested that AIA is equivalent to more complex methods (e.g., multiple imputation) across a number of variations of sample size, magnitude of associations among items, and degree of missingness. Thus, we utilized Parent's recommendations to guide our approach to managing missing data. Missing data analyses were conducted with tools in base R as well as the R packages, *psych* (v. 2.3.6) and *mice* (v. 3.16.0).

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 67%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data. At this stage in the analysis, we allowed all cases with less than 90% missing to continue to the scoring stage. Guided by Parent's [2013] AIA approach, scales with three items were scored if at least two items were non-missing; the scale with four items was scored if it at least three non-missing items; and the scale with six items was scored if it had at least five non-missing items.

Across the 66 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 61 cases. In this dataset we had less than 1% (0.55%) missing across the data set; 97% of the rows had nonmissing data.

2.11 Practice Problems

The three problems described below are designed to be continuations from the previous chapter (Scrubbing). You will likely encounter challenges that were not covered in this chapter. Search for and try out solutions, knowing that there are multiple paths through the analysis. The overall notion of the suggestions for practice are to (a) properly format three variables, (b) evaluate item-level missingness, (c) score any scales, (c) evaluate scale-level missingness, (d) provide an APA-style write-up, and (e) explain it to someone.

2.11.1 Problem #1: Reworking the Chapter Problem

If you chose this option in the prior chapter, you imported the data from Qualtrics, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, and wrote up the preliminary results.

2.11.2 Problem #2: Use the *Rate-a-Recent-Course Survey*, Choosing Different Variables

If you chose this option in the prior chapter, you chose a minimum of three variables from the *Rate-a-Recent-Course* survey to include in a simple statistical model. You imported the dat from Qualtrics, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest and wrote up the preliminary results.

2.11.3 Problem #3: Other data

If you chose this option in the prior chapter, you used raw data that was available to you. You imported it into R, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, and wrote up the preliminary results.

2.11.4 Grading Rubric

Assignment Component	Points Possible	Points Earned
1. Proper formatting of the items(s) in your first predictor variable	5	_____
2. Proper formatting of the items(s) in your second predictor variable	5	_____
3. Proper formatting of the items(s) your third predictor variable	5	_____
4. Proper formatting of your dependent variable	5	_____
4. Evaluate and interpret item-level missingness	5	_____
5. Score any scales/subscales	5	_____
7. Represent your work in an APA-style write-up (added to the writeup in the previous chapter)	5	_____
8. Explanation to grader	5	_____
Totals	45	_____

A *homeworked example* for the Scrubbing, Scoring, and DataDx lessons (combined) follows the [Data Dx](#) lesson.

Chapter 3

Data Dx

[Screencasted Lecture Link](#)

The focus of this chapter is *data diagnostics*. We are asking the question, “Does the data have the appropriate characteristics for the analysis we want to perform?” Some statistics are more robust than others to violations of the assumptions about the characteristics of the data. None-the-less, we must report these characteristics when we disseminate the results.

3.1 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, there are a few 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

Learning objectives from this lecture include the following:

- Conduct and interpret critical data diagnostics, including
 - alpha coefficients
 - skew
 - kurtosis
- Assess univariate and multivariate normality
- Identify options for managing outliers and skewed data
- Articulate a workflow for data preparation, including scrubbing, scoring, and data diagnostics

3.1.2 Planning for Practice

The suggestions from practice are a continuation from the two prior chapters. If you have completed one or more of those assignments, you should have started with a raw dataset and then scrubbed and scored it. This chapter will involve running basic data diagnostics. Options of graded complexity could include:

- Repeating the steps in the chapter with the most recent data from the Rate-A-Recent-Course survey; differences will be in the number of people who have completed the survey since the chapter was written.
- Use the dataset that is the source of the chapter, but score a different set of items that you choose.
- Begin with raw data to which you have access.

3.1.3 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.

- Parent, M. C. (2013). Handling item-level missing data: Simpler is just as good. *The Counseling Psychologist*, 41(4), 568–600. <https://doi.org/10.1177/001100012445176>
 - The purpose of Parent’s article was to argue that complex and resource-intensive procedures like multiple imputation are unnecessary. Following a simulation that supports his claims, Parent provides some guidelines to follow for the AIA approach.
- Kline, R. B. (2016). Data preparation and psychometrics review. In *Principles and Practice of Structural Equation Modeling*, Fourth Edition. Guilford Publications. <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4000663>
 - Kline’s chapter is my “go-to” for making decisions about preparing data for analysis.

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.

```
# if(!require(tidyverse)){install.packages('tidyverse')} #this
# includes dplyr if(!require(psych)){install.packages('psych')}
# if(!require(apaTables)){install.packages('apaTables')}
```

3.2 Workflow for Preliminary Data Diagnostics

The same workflow guides us through the Scrubbing, Scoring, and Data Dx chapters. At this stage we have

- imported our raw data from Qualtrics,
- scrubbed the data by applying our inclusion and exclusion criteria, and
- used Parent's available information approach [AIA; -Parent [2013]] for determining the acceptable amount of missingness for each scale, and
- prepared variables and scored them.

We are now ready to engage in data diagnostics for the statistical model we will test.

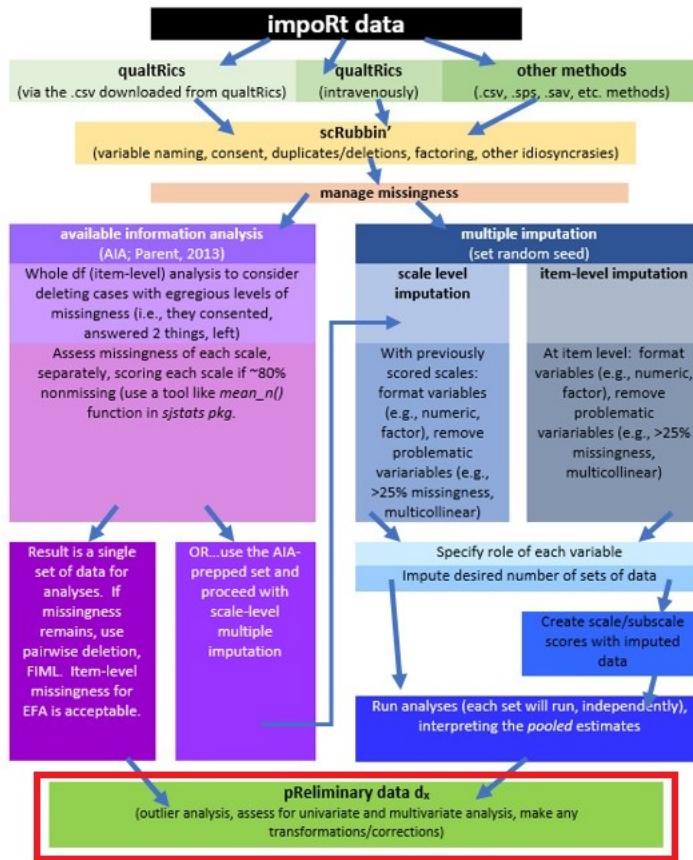


Figure 3.1: An image of our stage in the workflow for scrubbing and scoring data.

3.3 Research Vignette

The research vignette comes from the survey titled, [Rate-a-Recent-Course: A ReCentering Psych Stats Exercise](#) and is explained in the [scrubbing chapter](#). In the scoring chapter we prepared four variables for analysis. Details for these are in our [codebook](#).

Variable recap:

- Perceived Campus Climate for Black Students includes 6 items, one of which was reverse scored. This scale was adapted from Szymanski et al.'s [2020] Campus Climate for LGBTQ students. It has not been evaluated for use with other groups. The Szymanski et al. analysis

suggested that it could be used as a total scale score, or divided into three items each that assess

- College response to LGBTQ students (items 6, 4, 1)
- LGBTQ stigma (items 3, 2, 5)
- Sense of Belonging includes 3 items. This is a subscale from Bollen and Hoyle's [1990] Perceived Cohesion Scale. There are no items on this scale that require reversing.
- Percent of Black classmates is a single item that asked respondents to estimate the proportion of students in various racial categories
- Percent of BIPOC instructional staff, similarly, asked respondents to identify the racial category of each member of their instructional staff

As we noted in the [scrubbing chapter](#), our design has notable limitations. Briefly, (a) owing to the open source aspect of the data we do not ask about the demographic characteristics of the respondent; (b) the items that ask respondents to *guess* the identities of the instructional staff and to place them in broad categories, (c) we do not provide a “write-in” a response. We made these decisions after extensive conversation with stakeholders. The primary reason for these decisions was to prevent potential harm (a) to respondents who could be identified if/when the revealed private information in this open-source survey, and (b) trolls who would write inappropriate or harmful comments.

As I think about “how these variables go together” (which is often where I start in planning a study), imagine a parallel mediation. That is the perception of campus climate for Black students would be predicted by the respondent’s sense of belonging, mediated in separate paths through the proportion of classmates who are Black and the proportion of BIPOC instructional staff.

I would like to assess the model by having the instructional staff variable to be the %Black instructional staff. At the time that this lecture is being prepared, there is not sufficient Black representation in the staff to model this.

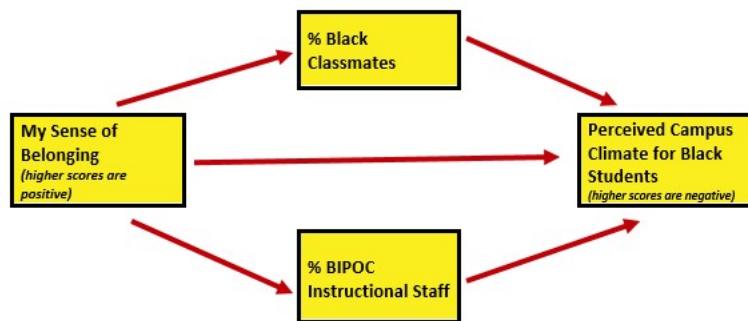


Figure 3.2: An image of the statistical model for which we are preparing data.

I will finish up this chapter by conducting a regression. Because parallel mediation can be complicated (I teach it in a later chapter), I will demonstrate use of our prepared variables with a simple multiple regression.

First, though, let’s take a more conceptual look at issues regarding missing data. We’ll come back to details of the survey as we work with it.

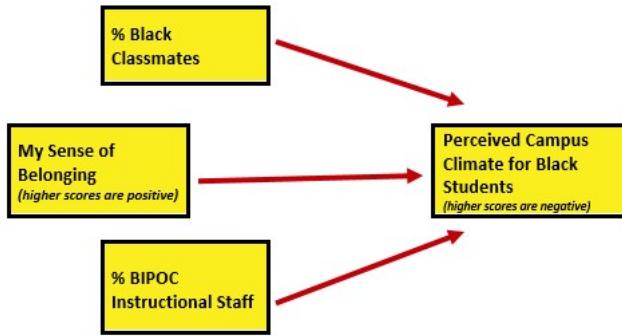


Figure 3.3: An image of the statistical model for which we are preparing data.

3.4 Internal Consistency of Scales/Subscales

Alpha coefficients are *reliability coefficients* that assess the *internal consistency* of an instrument. It asks, “For each person, are responses *consistently* high, or medium, or low?” To the degree that they are (meaning there are high inter-item correlations), the internal consistency coefficient will be high. We want values $>.80$. There are numerous problems with alpha coefficients. The biggest one is that they are influenced by sample size – longer scales have higher alpha coefficients [Cortina, 1993]. Fourteen seems to be a magic number where we begin to not trust the high alpha coefficient. I address this more thoroughly – offering an alternative – in psychometrics. While there is much criticism about the usefulness of the alpha coefficient [Sijtsma, 2009], researchers continue to use the alpha coefficient as an indicator of the internal consistency of scales that consist of multiple items and contain several variables.

We need item level data to compute an alpha coefficient. The easiest way to get an alpha coefficient is to feed the *alpha()* function (*psych* package) a concatenated list of items (with any items already reverse-scored). There should be no extra items. In the [scoring chapter](#) we already reverse-coded the single item in the campus climate scale, so we are ready to calculate alphas.

The df from which I am pulling data was created and written as an outfile in the [scoring chapter](#). You may also download the file from the [Github site](#) that hosts the chapter. Be sure to place the file in the same folder as the .rmd file. This particular df has item-level data. I am working with the .rds file. In case this is problematic for you, I have also provided code to import a .csv version of the file.

```
item_scores_df <- readRDS("BlStItmsScrs230902.rds")
# item_scores_df <- read.csv('BlStItmsScrs230902.csv', header = TRUE)
```

Within the *psych::alpha* function we can retrieve alpha coefficients for the specific variables of interest by imbedding a concatenated list. *A priori*, we are planning to use the campus climate scale as a total score. However, we’ll go ahead and also calculate alpha coefficients for the subscales because (a) it’s good practice and (b) if the alpha is low, a *reason* might show up in one of the subscales.

```
# alpha for the belonging scale
psych::alpha(item_scores_df[c("Belong_1", "Belong_2", "Belong_3")])
```

Reliability analysis
Call: psych::alpha(x = item_scores_df[c("Belong_1", "Belong_2", "Belong_3")])

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd	median_r
	0.95	0.95	0.93	0.87	21	0.0099	4	1.5	0.88

95% confidence boundaries
lower alpha upper
Feldt 0.93 0.95 0.97
Duhachek 0.93 0.95 0.97

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r	med.r
Belong_1	0.94	0.94	0.88	0.88	15	0.016	NA	0.88	
Belong_2	0.92	0.92	0.85	0.85	11	0.020	NA	0.85	
Belong_3	0.94	0.94	0.89	0.89	16	0.015	NA	0.89	

Item statistics
n raw.r std.r r.cor r.drop mean sd
Belong_1 64 0.95 0.95 0.92 0.90 4.1 1.5
Belong_2 65 0.96 0.96 0.94 0.92 4.1 1.6
Belong_3 64 0.95 0.95 0.91 0.89 3.8 1.5

Non missing response frequency for each item
1 2 3 4 5 6 7 miss
Belong_1 0.02 0.14 0.23 0.17 0.22 0.17 0.05 0.03
Belong_2 0.03 0.14 0.22 0.22 0.15 0.20 0.05 0.02
Belong_3 0.05 0.19 0.19 0.23 0.20 0.09 0.05 0.03

For each scale I will capture a statement for the APA style write-up. Because these values are typically reported with each measure (and not in the preliminary results), I won't create a cumulative write-up.

Cronbach's alpha for the belonging scale was 0.95.

```
# alpha for the campus climate for Black students scale
psych::alpha(item_scores_df[c("rBlst_1", "Blst_2", "Blst_3", "Blst_4",
"Blst_5", "Blst_6")])
```

Reliability analysis
Call: psych::alpha(x = item_scores_df[c("rBlst_1", "Blst_2", "Blst_3",

```

"Blst_4", "Blst_5", "Blst_6")])

raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
  0.85      0.87     0.87      0.52 6.5 0.03  2.5 1.1      0.52

 95% confidence boundaries
    lower alpha upper
Feldt      0.78  0.85  0.90
Duhachek   0.79  0.85  0.91

Reliability if an item is dropped:
  raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
rBlst_1      0.85      0.87     0.87      0.57 6.5 0.031 0.029  0.57
Blst_2       0.87      0.88     0.87      0.59 7.1 0.026 0.019  0.56
Blst_3       0.83      0.85     0.85      0.54 5.8 0.034 0.029  0.50
Blst_4       0.80      0.82     0.82      0.48 4.6 0.041 0.027  0.48
Blst_5       0.79      0.81     0.81      0.46 4.3 0.042 0.024  0.47
Blst_6       0.80      0.82     0.81      0.48 4.6 0.040 0.021  0.50

Item statistics
  n raw.r std.r r.cor r.drop mean   sd
rBlst_1 60  0.69  0.67  0.56   0.52 3.4 1.6
Blst_2  64  0.68  0.62  0.51   0.46 3.0 1.8
Blst_3  63  0.71  0.74  0.66   0.59 2.0 1.2
Blst_4  62  0.85  0.86  0.84   0.77 2.5 1.3
Blst_5  63  0.89  0.89  0.89   0.82 2.0 1.2
Blst_6  63  0.83  0.86  0.86   0.77 2.1 1.3

Non missing response frequency for each item
  1   2   3   4   5   6   7 miss
rBlst_1 0.10 0.23 0.20 0.25 0.08 0.10 0.03 0.09
Blst_2  0.33 0.16 0.09 0.17 0.16 0.06 0.03 0.03
Blst_3  0.44 0.33 0.06 0.11 0.03 0.02 0.00 0.05
Blst_4  0.27 0.34 0.15 0.18 0.05 0.00 0.02 0.06
Blst_5  0.46 0.30 0.05 0.14 0.05 0.00 0.00 0.05
Blst_6  0.38 0.35 0.11 0.08 0.06 0.02 0.00 0.05

```

Cronbach's alpha for the campus climate scale was 0.87.

Since this value is $\geq .80$, it is within the realm of acceptability. Let's go ahead, though, and examine its subscales.

```

# alpha for the stigma scale of the campus climate for Black students
# scale
psych::alpha(item_scores_df[c("Blst_3", "Blst_2", "Blst_5")])

```

```

Reliability analysis
Call: psych::alpha(x = item_scores_df[c("Blst_3", "Blst_2", "Blst_5")])

  raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
  0.69      0.73      0.69      0.47 2.7 0.065  2.3 1.2      0.54

  95% confidence boundaries
    lower alpha upper
Feldt     0.54  0.69  0.80
Duhachek  0.57  0.69  0.82

Reliability if an item is dropped:
  raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r med.r
Blst_3     0.67      0.70      0.54      0.54 2.35  0.074   NA  0.54
Blst_2     0.75      0.75      0.60      0.60 3.03  0.061   NA  0.60
Blst_5     0.41      0.43      0.28      0.28 0.76  0.135   NA  0.28

Item statistics
  n raw.r std.r r.cor r.drop mean   sd
Blst_3 63  0.72  0.78  0.62  0.46   2 1.2
Blst_2 64  0.82  0.75  0.55  0.46   3 1.8
Blst_5 63  0.87  0.89  0.83  0.70   2 1.2

Non missing response frequency for each item
  1   2   3   4   5   6   7 miss
Blst_3 0.44 0.33 0.06 0.11 0.03 0.02 0.00 0.05
Blst_2 0.33 0.16 0.09 0.17 0.16 0.06 0.03 0.03
Blst_5 0.46 0.30 0.05 0.14 0.05 0.00 0.00 0.05

```

Cronbach's alpha for the campus climate stigma subscale was 0.73.

```

# alpha for the campus responsiveness scale of the campus climate for
# Black students scale
psych::alpha(item_scores_df[c("rBlst_1", "Blst_4", "Blst_6")])

```

```

Reliability analysis
Call: psych::alpha(x = item_scores_df[c("rBlst_1", "Blst_4", "Blst_6")])

  raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
  0.79      0.81      0.76      0.58 4.2 0.045  2.7 1.2      0.52

  95% confidence boundaries
    lower alpha upper
Feldt     0.69  0.79  0.87
Duhachek  0.71  0.79  0.88

```

```

Reliability if an item is dropped:
    raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
rBlst_1      0.86      0.86     0.75      0.75 6.0     0.035    NA   0.75
Blst_4       0.64      0.65     0.48      0.48 1.8     0.087    NA   0.48
Blst_6       0.68      0.68     0.52      0.52 2.1     0.078    NA   0.52

Item statistics
    n raw.r std.r r.cor r.drop mean   sd
rBlst_1 60  0.81  0.78  0.58  0.53  3.4 1.6
Blst_4   62  0.88  0.89  0.84  0.72  2.5 1.3
Blst_6   63  0.85  0.87  0.81  0.69  2.1 1.3

Non missing response frequency for each item
    1   2   3   4   5   6   7 miss
rBlst_1 0.10 0.23 0.20 0.25 0.08 0.10 0.03 0.09
Blst_4   0.27 0.34 0.15 0.18 0.05 0.00 0.02 0.06
Blst_6   0.38 0.35 0.11 0.08 0.06 0.02 0.00 0.05

```

Cronbach's alpha for the campus climate responsiveness subscale was 0.80. Between the two subscales, it looks as if the responsiveness subscale is more internally consistent.

3.5 Distributional Characteristics of the Variables

3.5.1 Evaluating Univariate Normality

Statistics like ANOVA and regression each have a set of assumptions about the distributional characteristics of the data. In most of the chapters in this OER we review those assumptions and how to evaluate them. Common across many statistics is the requirement of univariate and multivariate normality. Let's take a look at the variables we will use in our analysis and assess those.

We can continue to work from the df we uploaded at the beginning of the chapter to do this work. Let's take a quick peek. This df has the item-level data (we used it for the alpha coefficients); the scale and subscale scores; and the two items that assess proportion of instructional staff that are BIPOC and proportion of classmates that are BIPOC.

The `str()` function let's us look at the variable format/measurement level of each variable.

```
str(item_scores_df)
```

```

Classes 'tbl_df', 'tbl' and 'data.frame':  66 obs. of  17 variables:
 $ ID        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ iBIPOC_pr : num  0.333 0 0.5 0.333 1 ...
 $ cmBlack   : num  0 5 10 6 5 20 0 0 0 4 ...
 ..- attr(*, "label")= Named chr "Regarding race, what proportion of students were from each "
 ...- attr(*, "names")= chr "Race_1"

```

```

$ Belong_1 : num  6 4 NA 5 4 5 6 7 6 3 ...
..- attr(*, "label")= Named chr "Please indicate the degree to which you agree with the following statement"
.. ..- attr(*, "names")= chr "Belong_1"
$ Belong_2 : num  6 4 3 3 4 6 6 7 6 3 ...
..- attr(*, "label")= Named chr "Please indicate the degree to which you agree with the following statement"
.. ..- attr(*, "names")= chr "Belong_2"
$ Belong_3 : num  7 6 NA 2 4 5 5 7 6 3 ...
..- attr(*, "label")= Named chr "Please indicate the degree to which you agree with the following statement"
.. ..- attr(*, "names")= chr "Belong_3"
$ Blst_1   : num  5 6 NA 2 6 5 5 5 5 3 ...
..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
.. ..- attr(*, "names")= chr "Blst_1"
$ Blst_2   : num  3 6 5 2 1 1 4 4 3 5 ...
..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
.. ..- attr(*, "names")= chr "Blst_2"
$ Blst_3   : num  5 2 2 2 1 1 4 3 1 2 ...
..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
.. ..- attr(*, "names")= chr "Blst_3"
$ Blst_4   : num  2 2 2 2 1 2 4 3 2 3 ...
..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
.. ..- attr(*, "names")= chr "Blst_4"
$ Blst_5   : num  2 4 NA 2 1 1 4 4 1 3 ...
..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
.. ..- attr(*, "names")= chr "Blst_5"
$ Blst_6   : num  2 1 2 2 1 2 4 3 2 3 ...
..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
.. ..- attr(*, "names")= chr "Blst_6"
$ rBlst_1  : num  3 2 NA 6 2 3 3 3 3 5 ...
..- attr(*, "label")= Named chr "Each item below asks you to rate elements of campus climate"
.. ..- attr(*, "names")= chr "Blst_1"
$ Belonging : num  6.33 4.67 NA 3.33 4 5.33 5.67 7 6 3 ...
$ ResponseBL: num  2.33 1.67 2 3.33 1.33 2.33 3.67 3 2.33 3.67 ...
$ StigmaBL  : num  3.33 4 3.5 2 1 1 4 3.67 1.67 3.33 ...
$ ClimateBL : num  2.83 2.83 NA 2.67 1.17 1.67 3.83 3.33 2 3.5 ...
- attr(*, "column_map")=Classes 'tbl_df', 'tbl' and 'data.frame': 182 obs. of 7 variables:
..$ qname      : chr [1:182] "StartDate" "EndDate" "Status" "Progress" ...
..$ description: chr [1:182] "Start Date" "End Date" "Response Type" "Progress" ...
..$ main       : chr [1:182] "Start Date" "End Date" "Response Type" "Progress" ...
..$ sub        : chr [1:182] "" "" "" ...
..$ ImportId   : chr [1:182] "startDate" "endDate" "status" "progress" ...
..$ timeZone   : chr [1:182] "America/Los_Angeles" "America/Los_Angeles" NA NA ...
..$ choiceId   : chr [1:182] NA NA NA NA ...

```

The difference between “int” (integer) and “num” (numerical) is that integers are limited to whole numbers. For the statistics used in this lesson, both are acceptable formats for the variables.

```
# the script may look a little complicated; I could have simply
# written: describe(item_scores_df) because I only wanted only a few
# variables, I provided them in a concatenated: list [c('iBIPOC_pr',
# 'cmBlack', 'Belonging', 'ClimateBL')] I used type =1 so that we can
# interpret skew and kurtosis along Kline's recommendations I created
# an object from the descriptive results, this can be used to export
# the results for easier table making or manipulation outside of R

descriptives <- psych::describe(item_scores_df[c("iBIPOC_pr", "cmBlack",
    "Belonging", "ClimateBL")], type = 1)
# When we capture results in an object, we need to write it below so
# the results will display
descriptives
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis
iBIPOC_pr	1	64	0.35	0.39	0.25	0.32	0.37	0	1.00	1.00	0.64	-1.05
cmBlack	2	66	8.20	8.02	5.50	7.24	8.15	0	30.00	30.00	0.95	0.05
Belonging	3	64	4.03	1.47	4.00	4.03	1.48	1	7.00	6.00	0.03	-0.76
ClimateBL	4	61	2.48	1.09	2.33	2.41	0.99	1	5.67	4.67	0.56	0.04
		se										
iBIPOC_pr		0.05										
cmBlack		0.99										
Belonging		0.18										
ClimateBL		0.14										

```
# this can be useful if you wish to manually format the data for an
# APA style table
write.csv(descriptives, file = "DataDx_descripts.csv")
```

Skew and kurtosis are one way to evaluate whether or not data are normally distributed. When we use the “type=1” argument, the skew and kurtosis indices in the *psych* package can be interpreted according to Kline’s [2016a] guidelines. Regarding skew, values greater than the absolute value of 3.0 are generally considered “severely skewed.” Regarding kurtosis, “severely kurtotic” is argued to be anywhere greater 8 to 20. Kline recommended using a conservative threshold of the absolute value of 10. The skew and kurtosis values for our variables fall well below these thresholds.

We can also apply the Shapiro-Wilk test of normality to each of our variables. When the *p* value is < .05, the variable’s distribution is deviates from a normal distribution to a degree that is statistically significant. Below, the plotting of the histogram with a normal curve superimposed shows how the distribution approximates one that is normal.

```
# The shapiro-test is in base R; it's specification is simple:
# shapiro.test(df$variable) I added the object (and had to list it
# below) so I can use the inline text function
shapiro.test(item_scores_df$cmBlack)
```

```
Shapiro-Wilk normality test

data: item_scores_df$cmBlack
W = 0.87796, p-value = 0.000009899

shapiro.test(item_scores_df$iBIPOC_pr)
```

```
Shapiro-Wilk normality test

data: item_scores_df$iBIPOC_pr
W = 0.78725, p-value = 0.00000003181

shapiro.test(item_scores_df$Belonging)
```

```
Shapiro-Wilk normality test

data: item_scores_df$Belonging
W = 0.97262, p-value = 0.1654

shapiro.test(item_scores_df$ClimateBL)
```

```
Shapiro-Wilk normality test

data: item_scores_df$ClimateBL
W = 0.95102, p-value = 0.01613
```

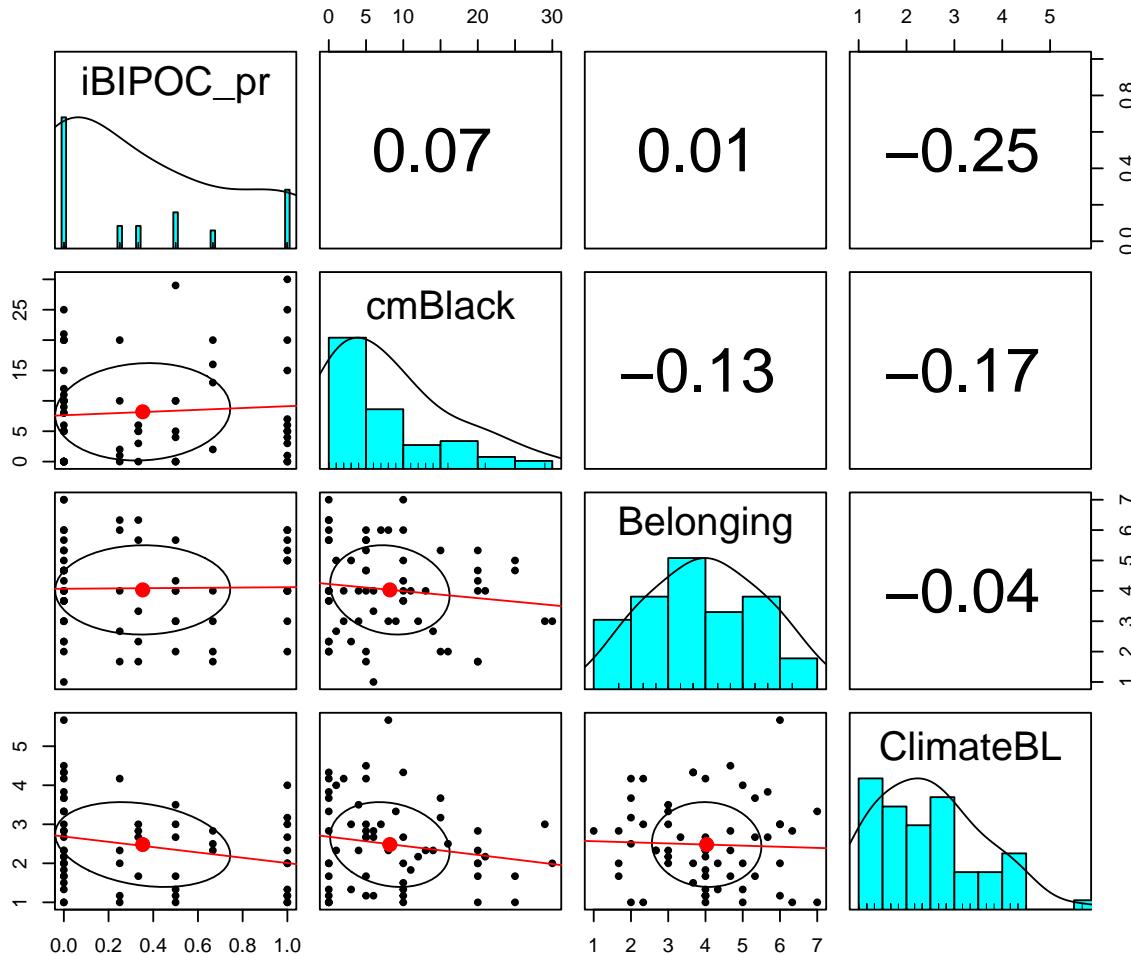
3.5.2 Pairs Panels

As we work our way from univariate to multivariate inspection of our data, let's take a look at the bivariate relations.

The *pairs.panels()* function from the *psych* package is useful for showing the relationship between variables (probably no more than 10) in a model.

- The lower half is a scatterplot between the two variables with a regression line (red) and mean (dot).
- The diagonal is a histogram of each variable.
- The upper half of is the correlation coefficient between the two variables.

```
psych::pairs.panels(item_scores_df[c("iBIPOC_pr", "cmBlack", "Belonging",
  "ClimateBL")], stars = TRUE, lm = TRUE)
```



The histograms displayed in the diagonal graph for us what we learned from the Shapiro Wilk's test of normality. We can clearly see the non-normal distribution in the iBIPOC_pr and cmBlack variables.

CUMULATIVE CAPTURE FOR THE APA STYLE WRITE-UP:

Regarding the distributional characteristics of the data, skew and kurtosis values of the variables fell below the values of 3 (skew) and 10 (kurtosis) that Kline suggests are concerning [2016b]. Results of the Shapiro-Wilk test of normality indicate that our variables assessing the proportion of classmates who are Black ($W = 0.878, p < 0.001$) and the proportion of BIPOC instructional staff ($W = 0.787, p < 0.001$) are statistically significantly different than a normal distribution. Similarly the scale assessing the respondent's perception of campus climate for Black students ($W = 0.951, p = 0.016$) differed significantly from a normal distribution. In all three cases the skew

values and histograms suggested a somewhat positive skew. That is, there were predominantly low proportions of instructional staff who are BIPOC and classmates who are Black, and the perceptions of campus climate for Black students was evaluated somewhat favorably. The scales assessing the respondent's belonging ($0.973, p = 0.165$) did not differ significantly from a normal distribution.

What would we do in the case of a univariate outlier? I find Kline's [2016b] chapter on data preparation and management to be extremely useful. He provides ideas for more complex analysis of both univariate and multivariate normality and provides suggestions that range from recoding an extreme value to the next most extreme that is within three standard deviations of the mean to more complicated transformations. First, though we need to further examine the relationships between variables. We do that, next.

3.6 Evaluating Multivariate Normality

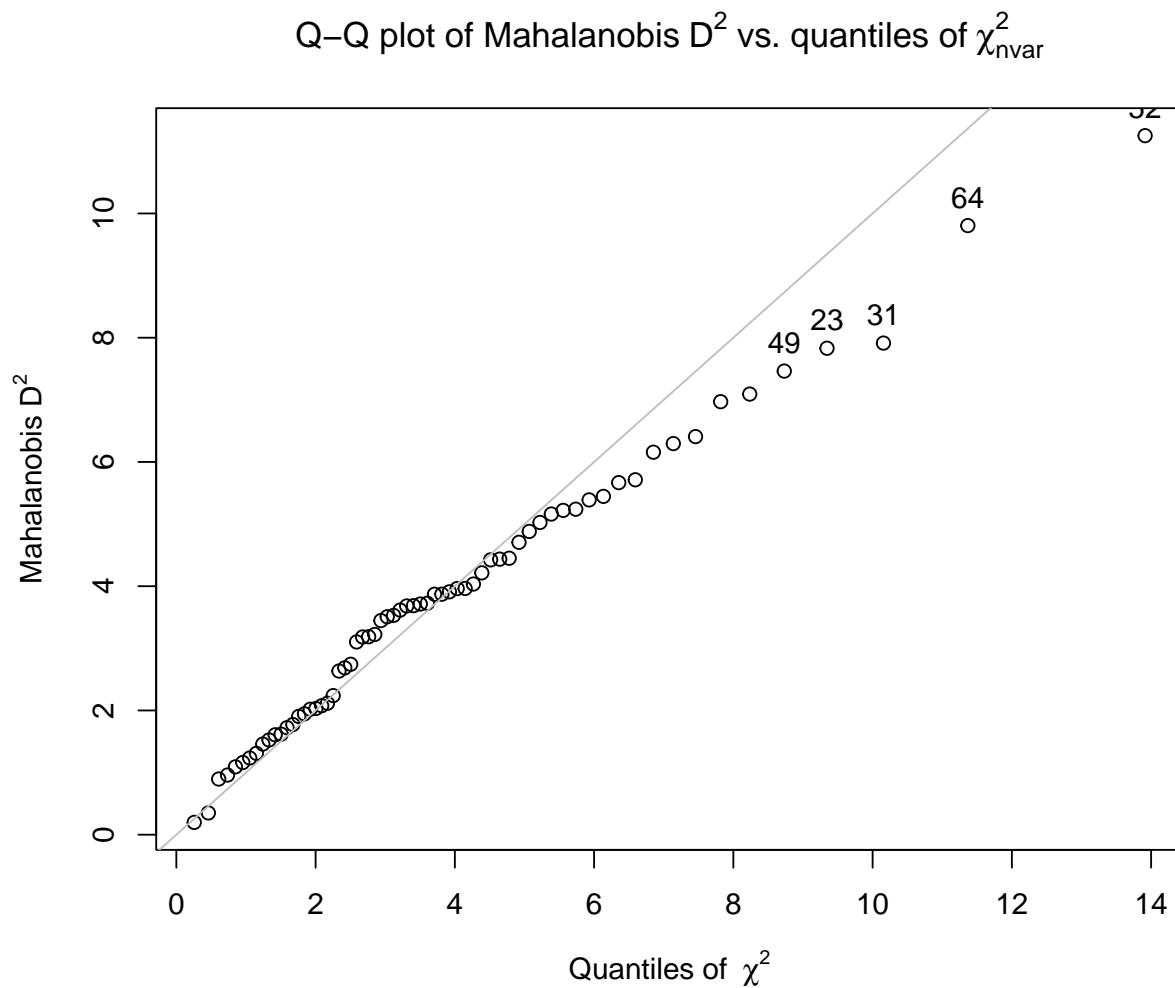
Multivariate outliers have extreme scores on two or more variables, or a pattern of scores that is atypical. For example, a case may have scores between two and three standard deviations above the mean on all variables, even though no case would be extreme. A common method of multivariate outlier detection is the **Mahalanobis distance** (D_M^2). This indicates the distance in variance units between the profile of scores for that case and the vector of sample means, or **centroid**, correcting for intercorrelations.

The *outlier()* function from the *psych* package tells us how far each datapoint is from the multivariate centroid of the data. That is, find the squared Mahalanobis distance for each data point and compare it to the expected values of χ^2 . The *outlier()* protocol also produces a Q-Q (quantile-quantile) plot with the n most extreme data points labeled.

The code below appends the Mahalanobis values to the dataframe. It is easy, then, to identify, sort, and examine the most extreme values (relative to the rest of the data in their case/row) to make decisions about their retention or adjustment.

Numeric variables are required in the of the calculation of the Mahalanobis.

```
item_scores_df$Mahal <- psych::outlier(item_scores_df[c("iBIPOC_pr", "cmBlack",
  "Belonging", "ClimateBL")])
```



Q-Q plots take your sample data, sort it in ascending order, and then plot them versus quantiles (the number varies; you can see it on the X axis) calculated from a theoretical distribution. The number of quantiles is selected to match the size of your sample data. While Normal Q-Q Plots are the ones most often used in practice due to so many statistical methods assuming normality, Q-Q Plots can actually be created for any distribution. To the degree that the plotted line stays on the straight line (representing the theoretical normal distribution), the data is multivariate normally distributed.

It is possible, then to analyze the Mahalanobis distance values.

```
psych::describe(item_scores_df$Mahal)
```

vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	66	3.81	2.24	3.68	3.62	2.36	0.2	11.25	11.05	0.86	0.82

Using this information we can determine cases that have a Mahalanobis distance values that exceeds three standard deviations around the median. In fact, we can have these noted in a column in the dataframe.

```

# creates a variable indicating TRUE or FALSE if an item is an
# outlier
item_scores_df$MOutlier <- dplyr::if_else(item_scores_df$Mahal > (median(item_scores_df$Mahal)
  (3 * sd(item_scores_df$Mahal))), TRUE, FALSE)

# shows us the first 6 rows of the data so we can see the new
# variables (Mahal, MOutlier)
head(item_scores_df)

# A tibble: 6 x 19
  ID iBIPOC_pr cmBlack Belong_1 Belong_2 Belong_3 Blst_1 Blst_2 Blst_3 Blst_4
  <int>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1     1      0.333      0       6       6       7       5       3       5       2
2     2      0.000      5       4       4       6       6       6       2       2
3     3      0.500     10      NA       3      NA      NA       5       2       2
4     4      0.333      6       5       3       2       2       2       2       2
5     5      1.000      5       4       4       4       6       1       1       1
6     6      0.000     20       5       6       5       5       1       1       2
# i 9 more variables: Blst_5 <dbl>, Blst_6 <dbl>, rBlst_1 <dbl>,
#   Belonging <dbl>, ResponseBL <dbl>, StigmaBL <dbl>, ClimateBL <dbl>,
#   Mahal <dbl>, MOutlier <lgl>

library(tidyverse)

-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become

# counts frequency TRUE and FALSE indicating outlier or not
OutlierCount <- item_scores_df %>%
  dplyr::count(MOutlier)

# calculating how many outliers a slightly different way
nrow(item_scores_df) - OutlierCount

MOutlier   n
1          66   1
2          65  65

```

When we identify outliers we often ask if we should delete them or transform the data. A general rule of thumb is to look for “jumps” in the Mahalanobis distance values. If they are progressing steadily and there is no “jump,” researchers will often retain the outliers.

CUMULATIVE CAPTURE FOR THE APA STYLE WRITE-UP:

We evaluated multivariate normality with the Mahalanobis distance test. Specifically, we used the *psych::outlier()* function and included all continuous variables in the calculation. Our visual inspection of the Q-Q plot suggested that the plotted line strayed from the straight line as the quantiles increased. Additionally, we appended the Mahalanobis distance scores as a variable to the data. Analyzing this variable, we found that 1 case exceed three standard deviations beyond the median. Given that the Mahalanobis distance values increased in a consistent manner (i.e., no extreme “jumps”) we retained all cases.

3.7 A Few Words on Transformations

To quote from Kline [2016b], “Before applying a normalizing transformation, you should think about the variables of interest and whether the expectation of normality is reasonable.” (p. 77)

At this point in history, the non-normal distribution of the proportions of classmates who are Black and instructional staff who are BIPOC are accurate representations in higher education. Kline [2016b] has noted that transforming an inherently non-normal variable to force a normal distribution may fundamentally alter it such that the variable of interest is not actually studied. Kline’s chapter reviews some options for applying corrections to outliers. Additionally, the chapter describes a variety of normalizing transformations.

On a personal note, while I will use standardized scores (a linear transformation) if it improves interpretation and center variables around a meaningful intercept, I tend to resist the transformation of data without a really compelling reason. Why? It’s complicated and can make interpretation difficult.

3.8 The APA Style Write-Up

This results section will draw from the three lessons on scrubbing, scoring, and data diagnostics.:.

3.8.1 Data Diagnostics

Data screening suggested that 107 individuals opened the survey link. Of those, 83 granted consent and proceeded into the survey items. A further inclusion criteria was that the course was taught in the U.S; 69 met this criteria.

Available item analysis (AIA; [Parent, 2013]) is a strategy for managing missing data that uses available data for analysis and excludes cases with missing

data points only for analyses in which the data points would be directly involved. Parent (2013) suggested that AIA is equivalent to more complex methods (e.g., multiple imputation) across a number of variations of sample size, magnitude of associations among items, and degree of missingness. Thus, we utilized Parent’s recommendations to guide our approach to managing missing data. Missing data analyses were conducted with tools in base R as well as the R packages, *psych* (v. 2.3.6) and *mice* (v. 3.16.0).

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 67%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data. At this stage in the analysis, we allowed all cases with less than 90% missing to continue to the scoring stage. Guided by Parent’s [2013] AIA approach, scales with three items were scored if at least two items were non-missing; the scale with four items was scored if it at least three non-missing items; and the scale with six items was scored if it had at least five non-missing items.

Across the 66 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 61 cases. In this dataset we had less than 1% (0.55%) missing across the df; 97% of the rows had nonmissing data.

Regarding the distributional characteristics of the data, skew and kurtosis values of the variables fell below the values of 3 (skew) and 10 (kurtosis) that Kline suggests are concerning [2016b]. Results of the Shapiro-Wilk test of normality indicate that our variables assessing the proportion of classmates who are Black ($W = 0.878, p < 0.001$) and the proportion of BIPOC instructional staff($W = 0.787, p < 0.001$) are statistically significantly different than a normal distribution. The scales assessing the respondent’s belonging (0.973, $p = 0.165$) and the respondent’s perception of campus climate for Black students ($W = 0.951, p = 0.016$) did not differ differently from a normal distribution.

We evaluated multivariate normality with the Mahalanobis distance test. Specifically, we used the *psych::outlier()* function and included all continuous variables in the calculation. Our visual inspection of the Q-Q plot suggested that the plotted line strayed from the straight line as the quantiles increased. Additionally, we appended the Mahalanobis distance scores as a variable to the data. Analyzing this variable, we found that 1 case exceed three standard deviations beyond the median. Given that the Mahalanobis distance values increased in a consistent manner (i.e., no extreme “jumps”) we retained all cases.

Given that our sample sizes were reasonable for the planned analyses and the degree of missingness was low, we used pairwise deletion in our multiple regression analysis.

3.9 A Quick Regression of our Research Vignette

With some confidence that our scrubbed-and-scored variables are appropriate for analysis, let me conduct the super quick regression that is our research vignette.

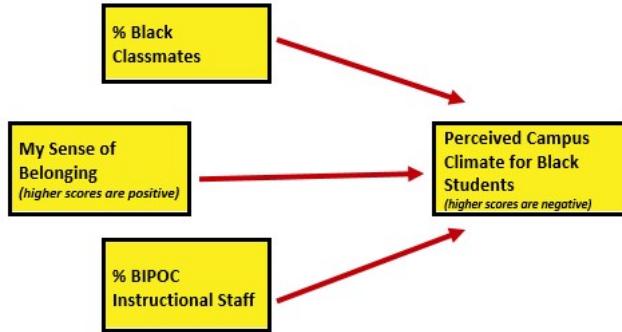


Figure 3.4: An image of the statistical model for which we are preparing data.

```
Climate_fit <- lm(ClimateBL ~ Belonging + cmBlack + iBIPOC_pr, data = item_scores_df)
summary(Climate_fit)
```

Call:

```
lm(formula = ClimateBL ~ Belonging + cmBlack + iBIPOC_pr, data = item_scores_df)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.86732	-0.80535	0.02355	0.70459	3.02003

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.90791	0.46653	6.233	0.0000000674 ***
Belonging	-0.01742	0.09643	-0.181	0.857
cmBlack	-0.01918	0.01717	-1.117	0.269
iBIPOC_pr	-0.64125	0.35701	-1.796	0.078 .

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

Residual standard error: 1.066 on 55 degrees of freedom

(7 observations deleted due to missingness)

Multiple R-squared: 0.08212, Adjusted R-squared: 0.03206

F-statistic: 1.64 on 3 and 55 DF, p-value: 0.1906

3.9.1 Results

Results of a multiple regression predicting the respondents' perceptions of campus climate for Black students indicated that neither con-

butions of the respondents' personal belonging ($B = -0.017, p = -.857$), the proportion of BIPOC instructional staff ($B = 0.641, p = 0.078$), nor the proportion of Black classmates ($B = -0.019, p = 0.269$) led to statistically significant changes in perceptions of campus climate for Black students. The model accounted for only 8% of the variance and was not statistically significant ($p = 0.191$). Means, standard deviations, and correlations among variables are presented in Table 1; results of the regression model are presented in Table 2.

```
apaTables::apa.cor.table(item_scores_df[c("iBIPOC_pr", "cmBlack", "Belonging",
                                         "ClimateBL")], table.number = 1, show.sig.stars = TRUE, filename = "Table1_M_SDs_r_DataDx.rmd")
```

Table 1

Means, standard deviations, and correlations with confidence intervals

Variable	M	SD	1	2	3
1. iBIPOC_pr	0.35	0.39			
2. cmBlack	8.20	8.02	.07 [-.18, .31]		
3. Belonging	4.03	1.47	.01 [-.24, .26]	-.13 [-.36, .12]	
4. ClimateBL	2.48	1.09	-.25 [-.47, .01]	-.17 [-.41, .08]	-.04 [-.29, .22]

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 $p < .01$.

```
library(apaTables)
apaTables::apa.reg.table(Climate_fit, table.number = 2, filename = "Climate_table.doc")
```

Table 2

Regression results using ClimateBL as the criterion

Predictor	b	b_95%_CI	beta	beta_95%_CI	sr2	sr2_95%_CI	r
(Intercept)	2.91**	[1.97, 3.84]					
Belonging	-0.02	[-0.21, 0.18]	-0.02	[-0.28, 0.24]	.00	[-.01, .01]	-.00
cmBlack	-0.02	[-0.05, 0.02]	-0.15	[-0.41, 0.12]	.02	[-.05, .09]	-.17
iBIPOC_pr	-0.64	[-1.36, 0.07]	-0.23	[-0.49, 0.03]	.05	[-.06, .16]	-.25

Fit

R2 = .082
95% CI [.00, .20]

Note. A significant b-weight indicates the beta-weight and semi-partial correlation are also significant. b represents unstandardized regression weights. beta indicates the standardized regression weight. sr2 represents the semi-partial correlation squared. r represents the zero-order correlation. Square brackets are used to enclose the lower and upper limits of a confidence interval. * indicates p < .05. ** indicates p < .01.

3.10 Practice Problems

The three problems described below are designed to be continuations from the Scrubbing and Scoring lessons. You will likely encounter challenges that were not covered in this chapter. Search for and try out solutions, knowing that there are multiple paths through the analysis. The overall notion of the suggestions for practice are to (a) calculate alpha coefficients for the scales, (b) evaluate univariate and multivariate normality, (c) create an APA-style write-up appropriate for a data diagnostics subsection of the results, and (d) run a “quickie” regression, ANOVA, or similar analysis.

3.10.1 Problem #1: Reworking the Chapter Problem

If you chose this option in the prior chapters, you imported the data from Qualtrics, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, properly formatted the variables, interpreted item-level missingness, scored the scales/subscales, interpreted scale-level missingness, and wrote up the results. Please continue with the remaining tasks.

3.10.2 Problem #2: Use the *Rate-a-Recent-Course* Survey, Choosing Different Variables

If you chose this option in the prior chapter, you chose a minimum of three variables (different from those in the chapter) from the *Rate-a-Recent-Course* survey to include in a simple statistical model. You imported the data from Qualtrics, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, properly formatted the variables, interpreted item-level missingness, scored the scales/subscales, interpreted scale-level missingness, and wrote up the results. Please continue with the remaining tasks.

3.10.3 Problem #3: Other data

If you chose this option in the prior chapter, you used raw data that was available to you. You imported it into R, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, properly formatted the variables, interpreted item-level missingness, scored the scales/subscales, interpreted scale-level missingness, and wrote up the results. Please continue with the remaining tasks.

3.10.4 Grading Rubric

Assignment Component		
1. Calculate alpha coefficients for scales/subscales.	5	_____
2. Evaluate univariate normality (skew, kurtosis, Shapiro-Wilks).	5	_____
3. Evaluate multivariate normality (Mahalanobis test)	5	_____
4. Represent your work in an APA-style write-up (added to the writeup in the previous chapter)	5	_____
5. Conduct a quick analysis (e.g., regression, ANOVA) including at least three predictor variables	5	_____
6. Explanation to grader	5	_____
Totals	30	_____

3.11 Homeworked Example

Screencast Link

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

Although the lessons focused on preparing data for analyses were presented in smaller sections, this homeworked example combines the suggestions for practice from the [Scrubbing](#), [Scoring](#), and [Data Dx](#) lessons. My hope is that this cumulative presentation is a closer approximation of what researchers need for their research projects.

These lessons were created to prepare a set of data to analyze a specific research model. Consequently, the model should be known and described at the beginning.

3.11.1 Scrubbing

Specify a research model

A further requirement was that the model should include three predictor variables (continuously or categorically scaled) and one dependent (continuously scaled) variable.

I am hypothesizing that socially responsive pedagogy (my dependent variable) will increase as a function of:

- the transition from SPSS (0) to R(1),
- the transition from a pre-centered (0) to re-centered (1) curriculum, and
- higher evaluations of traditional pedagogy

Because this data is nested within the person (i.e., students can contribute up to three course evaluations over the ANOVA, multivariate, and psychometrics courses) proper analysis would require a statistic (e.g., multilevel modeling) that would address the dependency in the data. Therefore, I will include only those students who are taking the multivariate modeling class.

If you wanted to use this example and dataset as a basis for a homework assignment, you could create a different subset of data. I worked the example for students taking the multivariate modeling class. You could choose ANOVA or psychometrics. You could also choose a different combinations of variables.

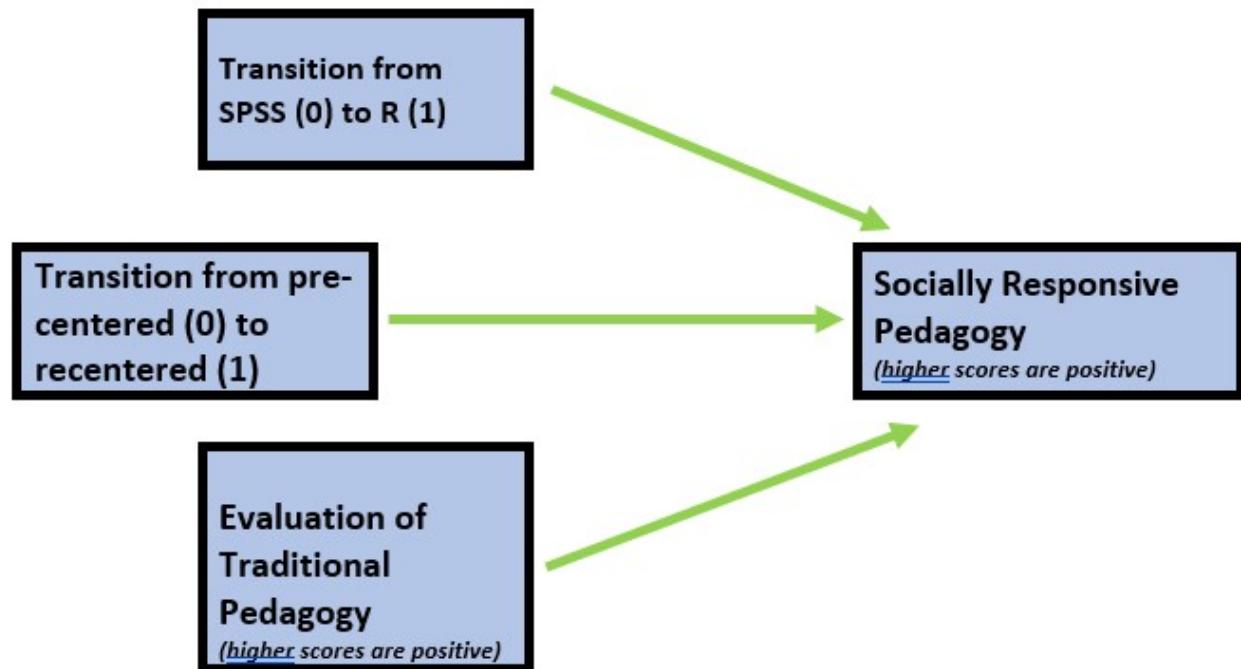


Figure 3.5: An image of our the prediction model for the homeworked example.

Import data

```
raw <- readRDS("ReC.rds")
nrow(raw)
```

[1] 310

Include only those who consented

Because this data is publicly posted on the Open Science Framework, it was necessary for me to already exclude those individuals. This data was unique in that students could freely write some version of “Opt out.” My original code included a handful of versions, but here was the basic form:

```
# testing to see if my code worked raw <- dplyr::filter (raw,
# SPFC.Decolonize.Opt.Out != 'Okay')
raw <- dplyr::filter(raw, SPFC.Decolonize.Opt.Out != "Opt Out")
```

Apply exclusionary criteria

I want to exclude students’ responses for the ANOVA and psychometrics courses.

```
raw <- (dplyr::filter(raw, Course == "Multivariate"))
```

At this point, these my only inclusion/exclusion criteria. I can determine how many students (who consented) completed any portion of the survey.

```
nrow(raw)
```

[1] 84

Rename variables to be sensible and systematic

Because this dataset is already on the OSF, the variables are sensibly named. However, I don’t like “SPFC.Decolonize.Opt.Out”. I will change it to simply “OptOut.”

```
raw <- dplyr::rename(raw, OptOut = "SPFC.Decolonize.Opt.Out")
```

It would have made more sense to do this before I used this variable in the calculations.

Downsize the dataframe to the variables of interest

I will need to include:

- deID
- StatsPkg
- Centering
- Items included in the traditional pedagogy scale: ClearResponsibilities, EffectiveAnswers, Feedback, ClearOrganization, ClearPresentation
- Items included in the socially responsive pedagogy scale: InclusvClassrm, EquitableEval, MultPerspectives, DEIntegration

```
scrub_df <- (dplyr::select(raw, deID, StatsPkg, Centering, ClearResponsibilities,
                           EffectiveAnswers, Feedback, ClearOrganization, ClearPresentation, InclusvClassrm,
                           EquitableEval, MultPerspectives, DEIntegration))
```

Provide an APA style write-up of these preliminary steps

This is a secondary analysis of data involved in a more comprehensive dataset that included students taking multiple statistics courses ($N = 310$). Having retrieved this data from a repository in the Open Science Framework, only those who consented to participation in the study were included. Data used in these analyses were 84 students who completed the multivariate class.

3.11.2 Scoring

Proper formatting of the item(s) in your first predictor variable

StatsPkg is a dichotomous variable. It should be structured as a factor with two ordered levels: SPSS, R

Because I am using the .rds form of the data from the OSF, this variable retains the former structure I assigned to it. If I needed to write the code, I would do this:

```
scrub_df$StatsPkg <- factor(scrub_df$StatsPkg, levels = c("SPSS", "R"))
str(scrub_df$StatsPkg)
```

Factor w/ 2 levels "SPSS","R": 2 2 2 2 2 2 2 2 2 ...

Proper formatting of item(s) in your second predictor variable

Similarly, Centering is a dichotomous variable. It should be structured as a factor with two ordered levels: Pre, Re.

Because I am using the .rds form of the data from the OSF, this variable retains the former structure I assigned to it. If I needed to write the code, I would do this:

```
scrub_df$Centering <- factor(scrub_df$Centering, levels = c("Pre", "Re"))
str(scrub_df$Centering)
```

Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 2 ...

Proper formatting of the item(s) in your third predictor variable

Proper formatting of the item(s) in your dependent variable

The third predictor variable is traditional pedagogy. The dependent variable is socially responsive pedagogy. The items that will be used in the scale scores for both of these variables are all continuously scaled and should be identified as “int” or “num.” None of the items need to be reverse-scored.

```
str(scrub_df)
```

```
Classes 'data.table' and 'data.frame': 84 obs. of 12 variables:
 $ deID           : int 11 12 13 14 15 16 17 18 35 19 ...
 $ StatsPkg       : Factor w/ 2 levels "SPSS","R": 2 2 2 2 2 2 2 2 2 2 ...
 $ Centering      : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 2 ...
 $ ClearResponsibilities: int 4 5 5 5 4 3 5 5 3 5 ...
 $ EffectiveAnswers: int 4 5 5 4 4 3 5 5 4 4 ...
 $ Feedback        : int 4 5 4 4 5 4 5 4 4 5 ...
 $ ClearOrganization: int 3 5 5 4 4 3 5 5 4 5 ...
 $ ClearPresentation: int 4 5 5 3 4 2 5 4 5 5 ...
 $ InclusvClassrm  : int 5 5 5 5 5 4 5 5 5 5 ...
 $ EquitableEval   : int 4 5 5 5 4 4 5 4 5 5 ...
 $ MultPerspectives: int 4 5 5 5 5 5 5 4 5 5 ...
 $ DEIintegration  : int 5 5 5 5 5 5 5 5 5 5 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Evaluate and interpret item-level missingness

The *scrub_df* is already downsized to include the item-level raw variables and the ID variable. We can continue using it.

I will create a “proportion missing” variable.

In this chunk I first calculate the number of missing (nmiss)

```
library(tidyverse)#needed because the script has pipes

#Calculating number and proportion of item-level missingness
scrub_df$nmiss <- scrub_df%>%
  dplyr::select(StatsPkg:DEIintegration) %>% #the colon allows us to include all variables b
  is.na %>%
```

```

rowSums

scrub_df<- scrub_df%>%
  dplyr::mutate(prop_miss = (nmiss/11)*100) #11 is the number of variables included in calculation

```

We can grab the descriptives for the `prop_miss` variable to begin to understand our data. I will create an object from it so I can use it with inline

```
psych::describe(scrub_df$prop_miss)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	84	2.38	6.17	0	0.94	0	0	36.36	36.36	3.29	12.33	0.67

Because I want to use the AIA approach to scoring, I'm not willing to filter out any cases yet. If I wanted to eliminate cases with egregious missing (i.e., like 90%), here is the code I would use:

```
scrub_df <- dplyr::filter(scrub_df, prop_miss <= 90) #update df to have only those with at least 10% missing
```

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 36%.

To analyze missingness at the item level, we need a df that has only the variables of interest. That is, variables like `ID` and the `prop_miss` and `nmiss` variables we created will interfere with an accurate assessment of missingness. I will update our df to eliminate these.

```
# further update to exclude the n_miss and prop_miss variables
ItemMiss_df <- scrub_df %>%
  dplyr::select(-c(deID, nmiss, prop_miss))
```

Missing data analysis commonly looks at proportions by:

- the entire df
- rows/cases/people

```
# what proportion of cells missing across entire dataset
formattable::percent(mean(is.na(ItemMiss_df)))
```

[1] 2.38%

```
# what proportion of cases (rows) are complete (nonmissing)
formattable::percent(mean(complete.cases(ItemMiss_df)))
```

[1] 82.14%

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 36%. Across the dataset, 2.38% of cells had missing data and 82.14% of cases had nonmissing data.

We can further explore patterns of missingness with *mice.md.pattern*.

```
mice:::md.pattern(ItemMiss_df, plot = TRUE, rotate.names = TRUE)
```

There are 6 missingness patterns. The most common ($n = 69$) have no missingness. There are 11 students missing the DEIntegration item (on the traditional pedagogy scale). This item may have been a later addition to the Canvas course evaluations.

Comparing this to Enders' [2010] [prototypical patterns of missingness](#) (page 3), the *mice* output represents the monotonic pattern often caused by test fatigue. That is, once a student stopped responding, they didn't continue with the rest of the evaluation. That said, this was true of only 4 students (1 each pattern). A quick reminder – diagnosing monotonicity requires that the variables in the *mice.mdpattern* figures were presented to the research participant in that order.

Score any scales/subscales

Traditional pedagogy is a predictor variable that needs to be created by calculating the mean if at least 75% of the items are non-missing. None of the items need to be reverse-scored. I will return to working with the *scrub_df* data.

```
# this seems to work when I build the book, but not in 'working the
# problem' TradPed_vars <- c('ClearResponsibilities',
# 'EffectiveAnswers', 'Feedback',
# 'ClearOrganization', 'ClearPresentation') scrub_df$TradPed <-
# sjstats::mean_n(scrub_df[, TradPed_vars], .75)

# this seems to work when I 'work the problem' (but not when I build
# the book) the difference is the two dots before the last SRPed_vars
TradPed_vars <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback",
"ClearOrganization", "ClearPresentation")
scrub_df$TradPed <- sjstats::mean_n(scrub_df[, TradPed_vars], 0.75)
```

The dependent variable is socially responsive pedagogy. It needs to be created by calculating the mean if at least 75% of the items are non-missing. None of the items need to be reverse-scored.

```
# this seems to work when I build the book, but not in 'working the
# problem' SRPed_vars <- c('InclusvClassrm', 'EquitableEval',
# 'MultPerspectives', 'DEIintegration') scrub_df$SRPed <-
# sjstats::mean_n(scrub_df[, SRPed_vars], .75)

# this seems to work when I 'work the problem' (but not when I build
# the book) the difference is the two dots before the last SRPed_vars
SRPed_vars <- c("InclusvClassrm", "EquitableEval", "MultPerspectives",
  "DEIintegration")
scrub_df$SRPed <- sjstats::mean_n(scrub_df[, SRPed_vars], 0.75)
```

Evaluate and interpret scale-level missingness

To evaluate scale level missingness, let's create a df with the focal variables.

```
scored <- dplyr::select(scrub_df, StatsPkg, Centering, TradPed, SRPed)
ScoredCaseMiss <- nrow(scored) #I produced this object for the sole purpose of feeding the nu
ScoredCaseMiss
```

[1] 84

Before we start our formal analysis of missingness at the scale level, let's continue to scrub by eliminating cases that will have too much missingness. In the script below we create a variable that counts the number of missing variables and then creates a proportion by dividing it by the number of total variables.

Using the *describe()* function from the *psych* package, we can investigate this variable.

```
library(tidyverse)
# Create a variable (n_miss) that counts the number missing
scored$n_miss <- scored %>%
  is.na %>%
  rowSums

# Create a proportion missing by dividing n_miss by the total number
# of variables (6) Pipe to sort in order of descending frequency to
# get a sense of the missingness
scored <- scored %>%
  mutate(prop_miss = (n_miss/6) * 100) %>%
  arrange(desc(n_miss))

psych::describe(scored$prop_miss)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	84	0.79	4.41	0	0	0	0	33.33	33.33	5.89	36.31	0.48

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 36%. Across the dataset, 2.38% of cells had missing data and 82.14% of cases had nonmissing data.

Across the 84 cases for which the scoring protocol was applied, missingness ranged from 0 to 33%.

We need to decide what is our retention threshold. Twenty percent seems to be a general rule of thumb. Let's delete all cases with missingness at 20% or greater.

```
# update df to have only those with at least 20% of complete data
# (this is an arbitrary decision)
scored <- dplyr::filter(scored, prop_miss <= 20)

# the variable selection just lops off the proportion missing
scored <- (select(scored, StatsPkg:SRPed))

# this produces the number of cases retained
nrow(scored)
```

[1] 83

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

Across the 84 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 83 cases.

Now, at the scale level, we look at missingness as the proportion of

- individual cells across the scored dataset, and
- rows/cases with nonmissing data

```
# percent missing across df
formattable::percent(mean(is.na(scored)))
```

[1] 0.60%

```
# percent of rows with nonmissing data  
formattable::percent(mean(complete.cases(scored)))
```

```
[1] 97.59%
```

CUMULATIVE CAPTURE FOR WRITING IT UP:

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

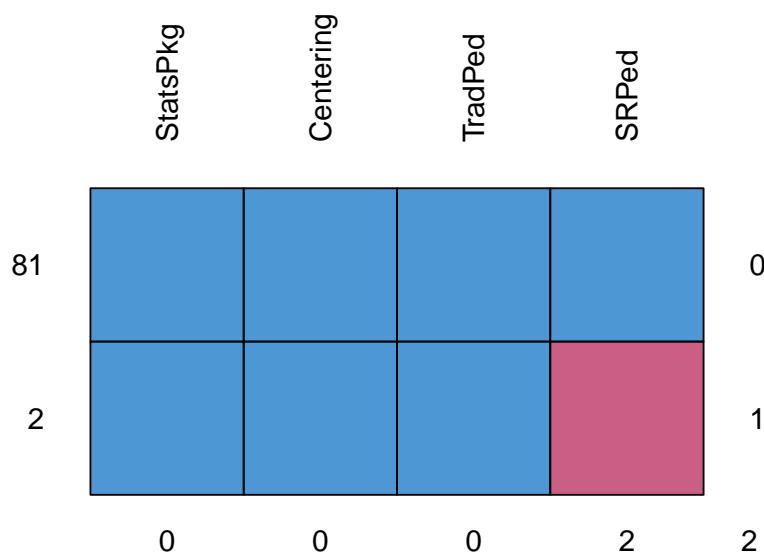
Across the 84 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 83 cases. In this dataset we had less than 1% (0.60%) missing across the df; 98% of the rows had nonmissing data.

Let's look again at missing patterns and mechanisms.

Returning to the *mice* package, we can use the *md.pattern()* function to examine a matrix with the number of columns 1 in which each row corresponds to a missing data pattern (0 = observed, 0 = missing). The rows and columns are sorted in increasing amounts of missing information. The last column and row contain row and column counts, respectively.

The corresponding figure shows non-missing data in blue; missing data in red.

```
mice_ScaleLvl <- mice::md.pattern(scored, plot = TRUE, rotate.names = TRUE)
```



```
mice_ScaleLvl
```

	StatsPkg	Centering	TradPed	SRPed
81	1	1	1	1 0
2	1	1	1	0 1
	0	0	0	2 2

There are 2 rows of data because there are only 2 patterns of missingness. The most common pattern is non-missing data ($n = 81$). Two cases are missing the SRPed variable. If our statistical choice uses listwise deletion (i.e., the case is eliminated if one or more variables in the model has missing data), our sample size will be 79. As we will learn in later chapters, there are alternatives (i.e., specifying a FIML option in analyses that use maximum likelihood estimators) that can use all of the cases – even those with missing data.

Represent your work in an APA-style write-up (added to the writeup in the previous chapter)

Available item analysis (AIA; [Parent, 2013]) is a strategy for managing missing data that uses available data for analysis and excludes cases with missing data points only for analyses in which the data points would be directly involved. Parent (2013) suggested that AIA is equivalent to more complex methods (e.g., multiple imputation) across a number of variations of sample size, magnitude of associations among items, and degree of missingness. Thus, we utilized Parent's recommendations to guide our approach to managing missing data. Missing data analyses were conducted with tools in base R as well as the R packages, *psych* (v. 2.3.6) and *mice* (v. 3.16.0).

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

Across the 84 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 83 cases. In this dataset we had less than 1% (0.60%) missing across the df; 98% of the rows had nonmissing data.

3.11.3 Data Dx

Calculate alpha coefficients for scales/subscales

To calculate the alpha coefficients, we need item-level data. We will return to *scrub_df* that contains the item-level data.

```
# alpha for the traditional pedagogy scale
psych::alpha(scrub_df[c("ClearResponsibilities", "EffectiveAnswers", "Feedback",
  "ClearOrganization", "ClearPresentation")])
```

```
Reliability analysis
Call: psych::alpha(x = scrub_df[c("ClearResponsibilities", "EffectiveAnswers",
  "Feedback", "ClearOrganization", "ClearPresentation")])

raw_alpha std.alpha G6(smc) average_r S/N    ase mean      sd median_r
  0.87        0.88     0.87      0.59 7.2 0.022   4.3 0.72      0.58

95% confidence boundaries
      lower alpha upper
Feldt     0.83  0.87  0.91
Duhachek 0.83  0.87  0.92
```

Reliability if an item is dropped:

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha	se	var.r
ClearResponsibilities	0.84	0.84	0.82	0.57	5.3	0.029	0.0110	
EffectiveAnswers	0.84	0.84	0.81	0.57	5.2	0.029	0.0088	
Feedback	0.87	0.87	0.86	0.64	7.0	0.023	0.0053	
ClearOrganization	0.86	0.86	0.83	0.60	6.1	0.025	0.0067	
ClearPresentation	0.83	0.84	0.81	0.57	5.3	0.030	0.0074	
	med.r							
ClearResponsibilities	0.55							
EffectiveAnswers	0.58							
Feedback	0.63							
ClearOrganization	0.59							
ClearPresentation	0.57							

Item statistics

	n	raw.r	std.r	r.cor	r.drop	mean	sd
ClearResponsibilities	83	0.85	0.85	0.80	0.74	4.5	0.87
EffectiveAnswers	84	0.84	0.85	0.82	0.76	4.4	0.79
Feedback	82	0.74	0.75	0.65	0.60	4.3	0.81
ClearOrganization	84	0.82	0.80	0.74	0.68	4.1	1.04
ClearPresentation	84	0.85	0.85	0.81	0.76	4.2	0.87

Non missing response frequency for each item

	1	2	3	4	5	miss
ClearResponsibilities	0.01	0.05	0.04	0.27	0.64	0.01
EffectiveAnswers	0.02	0.00	0.05	0.40	0.52	0.00
Feedback	0.01	0.01	0.11	0.38	0.49	0.02
ClearOrganization	0.04	0.07	0.07	0.43	0.39	0.00
ClearPresentation	0.01	0.06	0.04	0.46	0.43	0.00

Cronbach's alpha for the traditional pedagogy scale was 0.88.

```
# alpha for the traditional pedagogy scale
psych::alpha(scrub_df[c("InclusvClassrm", "EquitableEval", "DEIintegration",
  "DEIIntegration")])
```

Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done

```
In smc, smcs < 0 were set to .0
In smc, smcs < 0 were set to .0
In smc, smcs < 0 were set to .0
In smc, smcs < 0 were set to .0
```

Reliability analysis

```
Call: psych::alpha(x = scrub_df[c("InclusvClassrm", "EquitableEval",
```

```

"DEIintegration", "DEIintegration")])

raw_alpha std.alpha G6(smc) average_r S/N    ase mean    sd median_r
      0.85      0.85      0.7      0.58 5.6 0.025  4.5 0.62      0.55

95% confidence boundaries
      lower alpha upper
Feldt     0.79  0.85  0.9
Duhachek  0.80  0.85  0.9

Reliability if an item is dropped:
      raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r
InclusvClassrm      0.84      0.83      0.58      0.61 4.8  0.027 0.1115
EquitableEval       0.88      0.88      0.63      0.71 7.3  0.025 0.0640
DEIintegration      0.74      0.75      0.68      0.50 3.1  0.046 0.0054
DEIintegration.1    0.74      0.75      0.68      0.50 3.1  0.046 0.0054
      med.r
InclusvClassrm     0.42
EquitableEval      0.56
DEIintegration     0.53
DEIintegration.1   0.53

Item statistics
      n raw.r std.r r.cor r.drop mean    sd
InclusvClassrm    80  0.85  0.80  0.75  0.62  4.6  0.72
EquitableEval     84  0.71  0.72  0.60  0.51  4.7  0.50
DEIintegration    70  0.96  0.90  0.71  0.85  4.5  0.79
DEIintegration.1  70  0.96  0.90  0.71  0.85  4.5  0.79

Non missing response frequency for each item
      1    3    4    5 miss
InclusvClassrm   0.01 0.06 0.21 0.71 0.05
EquitableEval    0.00 0.01 0.32 0.67 0.00
DEIintegration   0.00 0.19 0.17 0.64 0.17
DEIintegration.1 0.00 0.19 0.17 0.64 0.17

```

Cronbach's alpha for the socially responsive pedagogy scale was 0.85.

Both of these are above the recommended value of 0.80.

Evaluate univariate normality (skew, kurtosis, Shapiro-Wilks)

We can inspect univariate normality by examining the skew and kurtosis values of the continuously scored variables.

```
psych::describe(scored, type = 1)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis
StatsPkg*	1	83	1.73	0.44	2.00	1.79	0.00	1.00	2	1.00	-1.06	-0.87
Centering*	2	83	1.36	0.48	1.00	1.33	0.00	1.00	2	1.00	0.58	-1.67
TradPed	3	83	4.29	0.72	4.40	4.40	0.59	1.20	5	3.80	-1.75	4.49
SRPed	4	81	4.51	0.58	4.75	4.60	0.37	2.33	5	2.67	-1.19	1.30
			se									
StatsPkg*			0.05									
Centering*			0.05									
TradPed			0.08									
SRPed			0.06									

When we use the “type=1” argument, the skew and kurtosis indices in the *psych* package can be interpreted according to Kline’s [2016a] guidelines.

Regarding the distributional characteristics of the data, skew and kurtosis values for our continuously scaled variables fall below the thresholds of concern (i.e., absolute value of 3 for skew; absolute value of 10 for kurtosis) identified by Kline [2016a].

Still at the univariate level, we can apply the Shapiro-Wilk test of normality to each of our continuously scaled variables. When the *p* value is < .05, the variable’s distribution is deviates from a normal distribution to a degree that is statistically significant. Below, the plotting of the histogram with a normal curve superimposed shows how the distribution approximates one that is normal.

```
# The shapiro-test is in base R; it's specification is simple:
# shapiro.test(df$variable) I added the object (and had to list it
# below) so I can use the inline text function
shapiro.test(scored$TradPed)
```

```
Shapiro-Wilk normality test

data: scored$TradPed
W = 0.83046, p-value = 0.0000000245

shapiro.test(scored$SRPed)
```

```
Shapiro-Wilk normality test

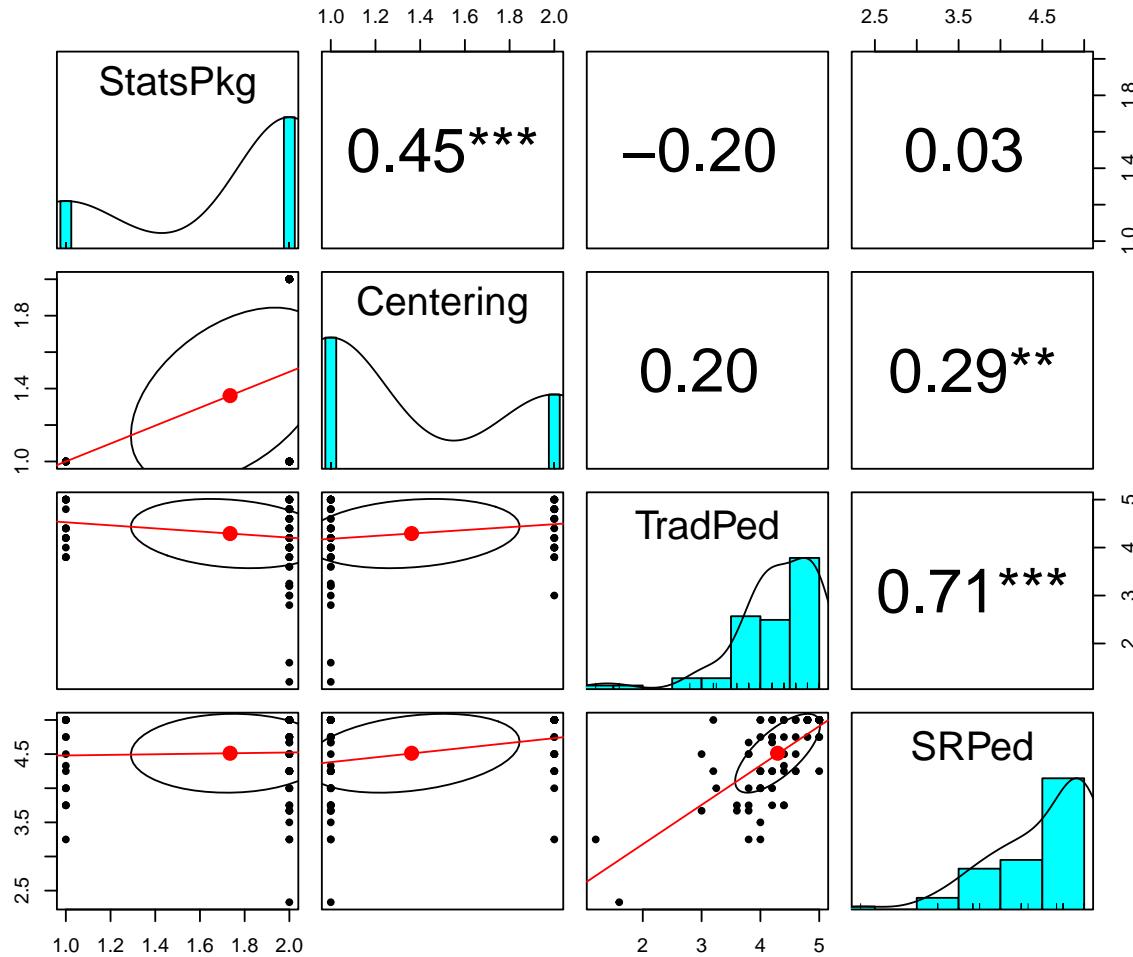
data: scored$SRPed
W = 0.81782, p-value = 0.0000000134
```

Both variable differ from a normal distribution in a statistically significant way.

- For the traditional pedagogy variable, $W = 0.830, p < 0.001$
- for the socially responsive pedagogy variable, $0.818, p < 0.001$

Obtaining a quick `psych::pairs.panel` can provide a quick glimpse of the distribution.

```
psych::pairs.panels(scored, stars = TRUE, lm = TRUE)
```



CUMULATIVE CAPTURE FOR THE APA STYLE WRITE-UP:

Regarding the distributional characteristics of the data, skew and kurtosis values of the variables fell below the values of 3 (skew) and 10 (kurtosis) that Kline suggests are concerning [2016b]. Results of the Shapiro-Wilk test of normality indicate that our variables assessing the traditional pedagogy ($W = 0.830, p < 0.001$) and socially responsive pedagogy ($0.818, p < 0.001$) are statistically significantly different than a normal distribution. Inspection

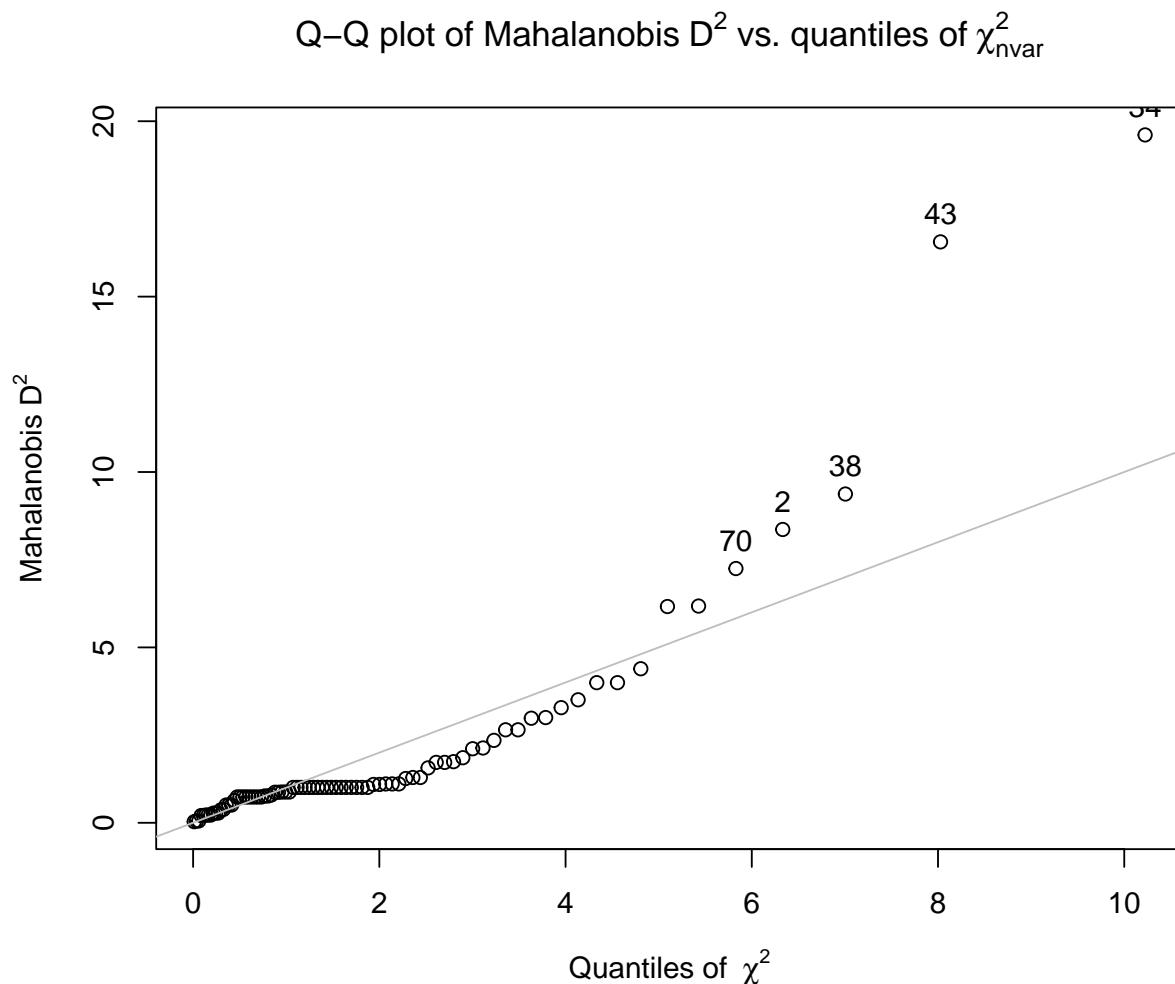
of distributions of the variables indicated that both course evaluation variables were negatively skewed, with a large proportion of high scores.

Evaluate multivariate normality (Mahalanobis test)

In more complex models, multivariate normality is probably a more useful analysis. Although I am teaching this evaluation in advance of the formal analysis, as demonstrated in many of [ReCentering Psych Stats ANOVA chapters](#), this can also be assessed by examining the distribution of residuals after the analysis is complete.

Multivariate normality can be assessed with the continuously scaled variables. The code below includes the only two continuously scaled variables. The code simultaneously (a) appends the df with a Mahalanobis value and (b) creates a QQ plot. Dots that stray from the line are the scores that are contributing to multivariate non-normality.

```
scored$Mahal <- psych::outlier(scored[c("TradPed", "SRPed")])
```



We can analyze the distributional characteristics of the Mahalanobis values with *psych::describe*. It is possible, then to analyze the Mahalanobis distance values.

```
psych::describe(scored$Mahal)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	83	1.97	3.12	1.01	1.27	0.42	0.03	19.61	19.58	3.75	15.87	0.34

Using this information we can determine cases that have a Mahalanobis distance values that exceeds three standard deviations around the median. In fact, we can have these noted in a column in the dataframe.

```
# creates a variable indicating TRUE or FALSE if an item is an
# outlier
scored$MOutlier <- dplyr::if_else(scored$Mahal > (median(scored$Mahal) +
  (3 * sd(scored$Mahal))), TRUE, FALSE)
```

```
# shows us the first 6 rows of the data so we can see the new
# variables (Mahal, MOutlier)
head(scored)
```

	StatsPkg	Centering	TradPed	SRPed	Mahal	MOutlier
1	SPSS	Pre	4.2	NA	0.0319020	FALSE
2	R	Pre	2.8	NA	8.3615550	FALSE
3	R	Re	3.8	4.5	0.8702516	FALSE
4	R	Re	5.0	5.0	1.0087776	FALSE
5	R	Re	4.8	5.0	0.7363631	FALSE
6	R	Re	4.0	5.0	2.6509906	FALSE

```
library(tidyverse)
# counts frequency TRUE and FALSE indicating outlier or not
OutlierCount <- scored %>%
  dplyr::count(MOutlier)

# calculating how many outliers a slightly different way
nrow(scored) - OutlierCount
```

	MOutlier	n
1	83	2
2	82	81

When we identify outliers we often ask if we should delete them or transform the data. A general rule of thumb is to look for “jumps” in the Mahalanobis distance values. If they are progressing steadily and there is no “jump,” researchers will often retain the outliers.

In this case, I do see a jump. When I sort the df on Mahal values, the jump from 9.37 to 16.56 is much different than the more gradual increase in values that precedes it. Therefore, I think I will delete cases with Mahalanobis values greater than 10 (a number I “just picked”).

```
scored <- dplyr::filter(scored, Mahal < "10")
```

We evaluated multivariate normality with the Mahalanobis distance test. Specifically, we used the *psych::outlier()* function and included both continuous variables in the calculation. Our visual inspection of the Q-Q plot suggested that the plotted line strayed from the straight line as the quantiles increased. Additionally, we appended the Mahalanobis distance scores as a variable to the data. Analyzing this variable, we found that 2 cases exceed three standard deviations beyond the median. Because there was a substantial “jump” between the non-outliers and these two variables we chose to delete them.

Represent your work in an APA-style write-up (added to the writeup in the previous chapter)

This is a secondary analysis of data involved in a more comprehensive dataset that included students taking multiple statistics courses ($N = 310$). Having retrieved this data from a repository in the Open Science Framework, only those who consented to participation in the study were included. Data used in these analyses were 84 students who completed the multivariate clas.

Available item analysis (AIA; [Parent, 2013]) is a strategy for managing missing data that uses available data for analysis and excludes cases with missing data points only for analyses in which the data points would be directly involved. Parent (2013) suggested that AIA is equivalent to more complex methods (e.g., multiple imputation) across a number of variations of sample size, magnitude of associations among items, and degree of missingness. Thus, we utilized Parent’s recommendations to guide our approach to managing missing data. Missing data analyses were conducted with tools in base R as well as the R packages, *psych* (v. 2.3.6) and *mice* (v. 3.16.0).

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data.

Across the 84 cases for which the scoring protocol was applied, missingness ranged from 0 to 67%. After eliminating cases with greater than 20% missing, the dataset analyzed included 83 cases. In this dataset we had less than 1% (0.60%) missing across the df; 98% of the rows had nonmissing data.

Regarding the distributional characteristics of the data, skew and kurtosis values of the variables fell below the values of 3 (skew) and 10 (kurtosis) that Kline suggests are concerning [2016b]. Results of the Shapiro-Wilk test of normality indicate that our variables assessing the traditional pedagogy ($W = 0.830, p < 0.001$) and socially responsive pedagogy ($0.818, p < 0.001$)

are statistically significantly different than a normal distribution. Inspection of distributions of the variables indicated that both course evaluation variables were negatively skewed, with a large proportion of high scores.

We evaluated multivariate normality with the Mahalanobis distance test. Specifically, we used the *psych::outlier()* function and included both continuous variables in the calculation. Our visual inspection of the Q-Q plot suggested that the plotted line strayed from the straight line as the quantiles increased. Additionally, we appended the Mahalanobis distance scores as a variable to the data. Analyzing this variable, we found that 2 cases exceed three standard deviations beyond the median. Because there was a substantial “jump” between the non-outliers and these two variables we chose to delete them.

Conduct a quick analysis (e.g., regression, ANOVA) including at least three variables

```
SRPed_fit <- lm(SRPed ~ StatsPkg + Centering + TradPed, data = scored)
summary(SRPed_fit)
```

Call:

```
lm(formula = SRPed ~ StatsPkg + Centering + TradPed, data = scored)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.56099	-0.14406	0.01551	0.10594	0.46498

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.46330	0.34441	4.249	0.000077464849487 ***
StatsPkgR	0.13251	0.08056	1.645	0.105
CenteringRe	0.05666	0.07423	0.763	0.448
TradPed	0.68663	0.07365	9.323	0.000000000000332 ***

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

Residual standard error: 0.2433 on 59 degrees of freedom

(1 observation deleted due to missingness)

Multiple R-squared: 0.6167, Adjusted R-squared: 0.5972

F-statistic: 31.64 on 3 and 59 DF, p-value: 0.000000000002547

3.11.4 Results

Results of a multiple regression predicting the socially responsive course evaluation ratings indicated that neither the transition from SPSS to R ($B =$

$0.133, p = 0.105$) nor the transition to an explicitly recentered curriculum ($B = 0.057, p = 0.448$) led to statistically significant differences. In contrast, traditional pedagogy had a $\beta = 0.686$, $p < 0.001$). The model accounted for 62% of the variance and was statistically significant ($p, 0.001$). Means, standard deviations, and correlations among variables are presented in Table 1; results of the regression model are presented in Table 2.

```
apaTables::apa.cor.table(scored[c("SRPed", "StatsPkg", "Centering", "TradPed")],  
  table.number = 1, show.sig.stars = TRUE, filename = "Table1__DataDx_HW.doc")
```

Table 1

Means, standard deviations, and correlations with confidence intervals

Variable	M	SD	1
1. SRPed	4.69	0.38	
2. TradPed	4.53	0.43	.76** [.63, .85]

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 $p < .01$.

```
apaTables::apa.reg.table(SRPed_fit, table.number = 2, filename = "SRPed_table.doc")
```

Table 2

Regression results using SRPed as the criterion

Predictor	b	b_95%_CI	sr2	sr2_95%_CI	Fit
(Intercept)	1.46**	[0.77, 2.15]			
StatsPkgR	0.13	[-0.03, 0.29]	.02	[-.02, .06]	
CenteringRe	0.06	[-0.09, 0.21]	.00	[-.02, .02]	
TradPed	0.69**	[0.54, 0.83]	.56	[.40, .73]	R2 = .617**

95% CI [.43,.70]

Note. A significant b-weight indicates the semi-partial correlation is also significant.
b represents unstandardized regression weights.

sr2 represents the semi-partial correlation squared.

Square brackets are used to enclose the lower and upper limits of a confidence interval.

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

Chapter 4

Multiple Imputation (A Brief Demo)

[Screencasted Lecture Link](#)

Multiple imputation is a tool for managing missing data that works with the whole raw data file to impute values for missing data for *multiple sets* (e.g., 5-20) of the raw data. Those multiple sets are considered together in analyses (such as regression) and interpretation is made on the pooled results. Much has been written about multiple imputation and, if used, should be done with many considerations. This chapter is intended as a brief introduction. In this chapter, I demonstrate the use of multiple imputation with the data from the [Rate-a-Recent-Course: A ReCentering Psych Stats Exercise](#) that has served as the research vignette for the first few chapters of this OER.

4.1 Navigating this Lesson

There is about one hour of lecture. If you work through the materials with me it would be good to add another hour (to an 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, there are a few 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

Learning objectives from this lecture include the following:

- Describe circumstances under which multiple imputation would be appropriate
- List and define the stages in multiple imputation.
- Apply multiple imputation to a dataset that has missingness
- Interpret results from a simple regression that uses multiple imputation
- Articulate how multiple imputation fits into the workflow for scrubbing and scoring data.
- Write up the results of an the process of imputation from raw data through analyzing a simple regression (or similar) analysis.

4.1.2 Planning for Practice

The suggestions for practice are a continuation from the three prior chapters. If you have completed one or more of those assignments, you should have worked through the steps in preparing a data set and evaluating its appropriateness for the planned, statistical, analysis. This chapter takes a deviation from the AIA [Parent, 2013] approach that was the focus of the first few chapters in that we used multiple imputation as the approach for managing missingness. Options, of graded complexity, for practice include:

- Repeating the steps in the chapter with the most recent data from the Rate-A-Recent-Course survey; differences will be in the number of people who have completed the survey since the chapter was written.
- Use the dataset that is the source of the chapter, but score a different set of items that you choose.
- Begin with raw data to which you have access.

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.

- Enders, C. K. (2017). Multiple imputation as a flexible tool for missing data handling in clinical research. *Behaviour Research and Therapy*, 98, 4–18.
 - Craig Enders is a leading expert in the analysis and management of missing data. This article is useful in describing multiple imputation as a method for managing missingness.
- Katitas, A. (2019). Getting Started with Multiple Imputation in R. University of Virginia Library: Research Data Services + Sciences. <https://library.virginia.edu/data/articles/getting-started-with-multiple-imputation-in-r>
 - Tutorial for conducting multiple imputation in R.
- Kline Ch4, Data Preparation & Psychometrics Review (pp. 72/Outliers - 88/Modern Methods)
- Kline's chapter is my “go-to” for making decisions about preparing data for analysis.

4.1.4 Packages

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

```

}
if (!require(dplyr)) {
  install.packages("dplyr")
}
if (!require(mice)) {
  install.packages("mice")
}

```

4.2 Workflow for Multiple Imputation

The following is a proposed workflow for preparing data for analysis.

In this lecture we are working on the right side of the flowchart in the multiple imputation (blue) section. Within it, there are two options, each with a slightly different set of options.

- imputing at the item level
 - in this case, scales/subscales are scored after the item-level imputation
- imputating at the scale level
 - in this case, scales/subscales are scored prior to the imputation; likely using some of the same criteria as identified in the scoring chapter (i.e., scoring if 75-80% of data are non-missing). Multiple imputation, then, is used to estimate the remaining, missing values.

Whichever approach is used, the imputed variables (multiple sets) are used in a *pooled analysis* and results are interpreted from that analysis.

4.3 Research Vignette

The research vignette comes from the survey titled, [Rate-a-Recent-Course: A ReCentering Psych Stats Exercise](#) and is explained in the [scrubbing chapter](#). In the [scoring chapter](#) we prepared four variables for analysis. In the [data diagnostics chapter](#) we assessed the quality of the variables and conducted the multiple regression described below. Details for these are in our [codebook](#).

Let's quickly review the variables in our model:

- Perceived Campus Climate for Black Students includes 6 items, one of which was reverse scored. This scale was adapted from Szymanski et al.'s [2020] Campus Climate for LGBTQ students. It has not been evaluated for use with other groups. The Szymanski et al. analysis suggested that it could be used as a total scale score, or divided into three items each that assess
 - College response to LGBTQ students (items 6, 4, 1)
 - LGBTQ stigma (items 3, 2, 5)

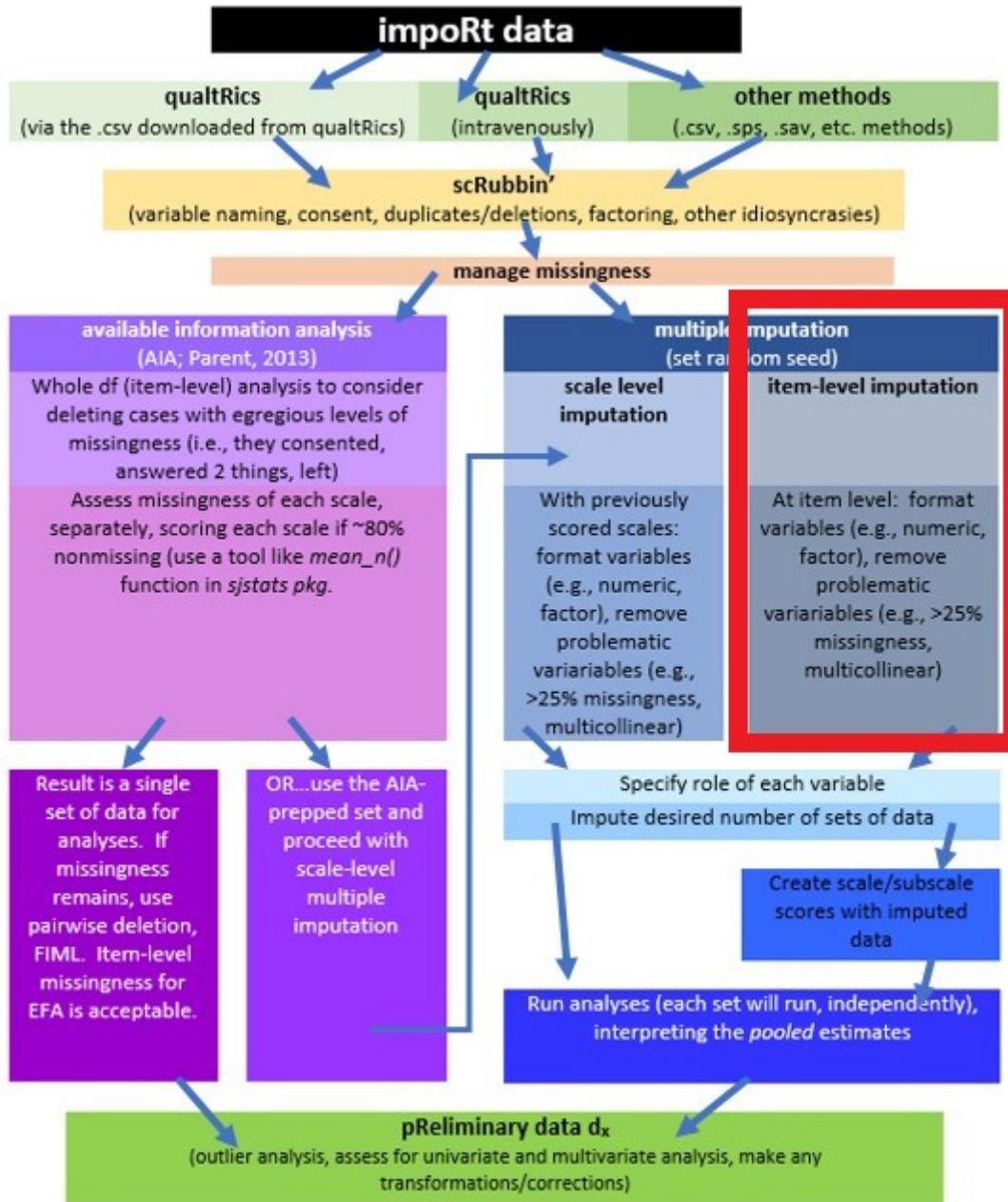


Figure 4.1: An image of a workflow for scrubbing and scoring data.

- Sense of Belonging includes 3 items. This is a subscale from Bollen and Hoyle's [1990] Perceived Cohesion Scale. There are no items on this scale that require reversing.
- Percent of Black classmates is a single item that asked respondents to estimate the proportion of students in various racial categories
- Percent of BIPOC instructional staff, similarly, asked respondents to identify the racial category of each member of their instructional staff

As we noted in the [scrubbing chapter](#), our design has notable limitations. Briefly, (a) owing to the open source aspect of the data we do not ask about the demographic characteristics of the respondent; (b) the items that ask respondents to *guess* the identities of the instructional staff and to place them in broad categories, (c) we do not provide a “write-in” a response. We made these decisions after extensive conversation with stakeholders. The primary reason for these decisions was to prevent potential harm (a) to respondents who could be identified if/when the revealed private information in this open-source survey, and (b) trolls who would write inappropriate or harmful comments.

As I think about “how these variables go together” (which is often where I start in planning a study), I suspect parallel mediation. That is the perception of campus climate for Black students would be predicted by the respondent’s sense of belonging, mediated in separate paths through the proportion of classmates who are Black and the proportion of BIPOC instructional staff.

I would like to assess the model by having the instructional staff variable to be the %Black instructional staff. At the time that this lecture is being prepared, there is not sufficient Black representation in the instructional staff to model this.

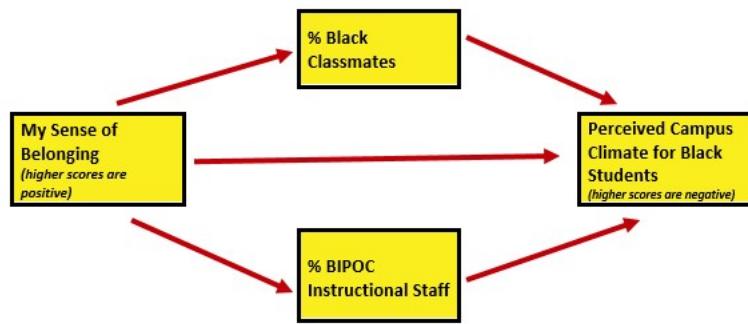


Figure 4.2: An image of the statistical model for which we are preparing data.

As in the [data diagnostic chapter](#), I will conclude this chapter by conducting a statistical analysis with the multiply imputed data. Because parallel mediation can be complicated (I teach it in a later chapter), I will demonstrate use of our prepared variables with a simple multiple regression.

4.4 Multiple Imputation – a Super Brief Review

Multiple imputation is complex. Numerous quantitative psychologists had critiqued it and provided numerous cautions and guidelines for its use [[Enders, 2010, 2017](#), [Little et al., 2008](#), [Little and Rubin, 2002](#)]. In brief,

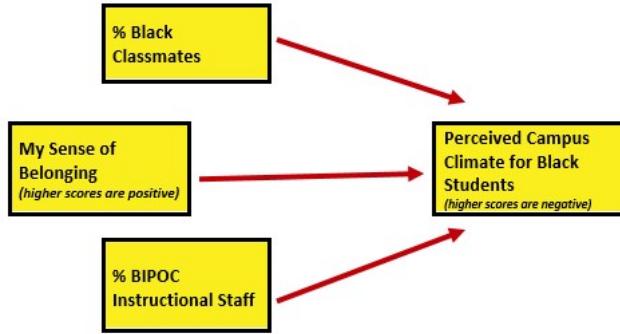


Figure 4.3: An image of the statistical model for which we are preparing data.

4.4.1 Steps in Multiple Imputation

- Multiple imputation starts with a raw data file.
 - Multiple imputation assumes that data are MAR (remember, MCAR is the more prestigious one). This means that researchers assume that missing values can be replaced by predictions derived from the observable portion of the dataset.
- Multiple datasets (often 5 to 20) are created where missing values are replaced via a randomized process (so the same missing value [item 4 for person A] will likely have different values for each dataset).
- The desired analysis is conducted simultaneously/separately for each of the imputed sets (so if you imputed 5 sets and wanted a linear regression, you get 5 linear regressions).
- A *pooled analysis* uses the point estimates and the standard errors to provide a single result that represents the analysis.

In a web-hosted guide from the University of Virginia Library, Katitas [2019] provided a user-friendly review and example of using tools in R in a multiple imputation. Katitas' figure is a useful conceptual tool in understanding how multiple imputation works. *This figure is my recreation of Katitas' original.*

- the dataframe with missing data is the single place we start
- we intervene with a package like `mice()` to
- impute multiple sets of data (filling in the missing variables with different values that are a product of their conditional distribution and an element of “random”);
 - “mids” (“multiply imputed dataset”) is an object class where the completed datasets are stored.
- the “with_mids” command allows OLS regression to be run, as many times as we have imputed datasets (in this figure, 3X). It produces different regression coefficients for each dataset

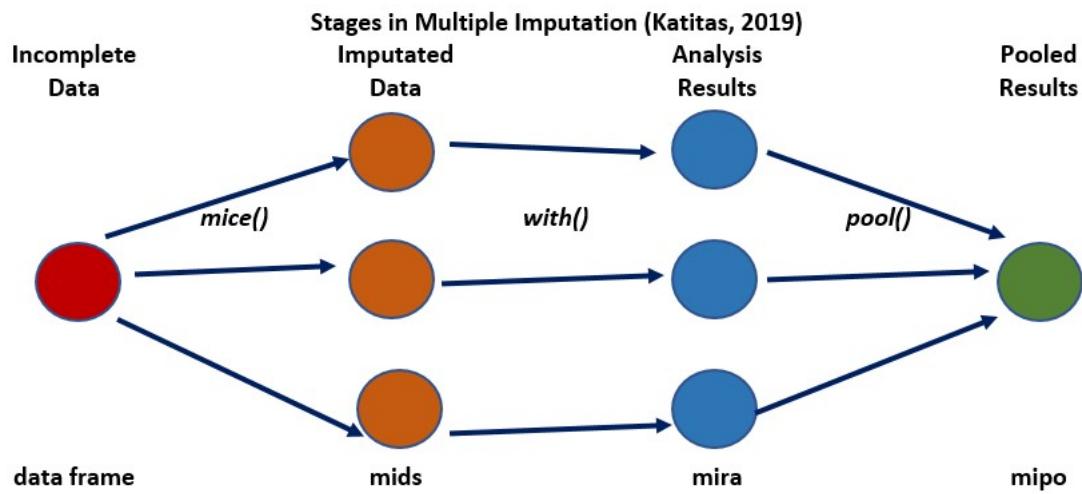


Figure 4.4: An image adapted from the Katitas multiple imputation guide showing the four stages of multiple imputation.

- the “pool” command pools together the multiple coefficients taking into consideration the value of the coefficients, the standard errors, and the variance of the missing value parameter across the samples.

4.4.2 Statistical Approaches to Multiple Imputation

Joint multivariate normal distribution multiple imputation assumes that the observed data follow a multivariate normal distribution. The algorithm used draws from this assumed distribution. A drawback is that if the data do not follow a multivariate normal distribution, the imputed values are incorrect. *Amelia* and *norm* packages use this approach.

Conditional multiple imputation is an iterative procedure, modeling the conditional distribution of a certain variable given the other variables. In this way the distribution is assumed for each variable, rather than for the entire dataset. *mice* uses this approach.

mice: multivariate imputation by chained equations

4.5 Working the Problem

Katitas [2019] claims that it is best to impute the data in its rawest form possible because any change would be taking it away from its original distribution. There are debates about how many variables to include in an imputation. Some authors would suggest that researchers include everything that was collected. Others (like me) will trim the dataset to include (a) the variables included in the model, plus (b) auxiliary variables (i.e., variables not in the model, but that are sufficiently non-missing and will provide additional information to the data).

In our case we will want:

Item for the variables represented in our model

- the item level responses to the scales/subscales
 - respondents' sense of belonging to campus (3 items)
 - respondents' rating of campus climate for Black students (6 items)
- proportion of BIPOC instructional staff
- proportion of classmates who are Black

Auxiliary variables – let's choose four. One will be the format of the course. Three items will be from the course evaluation.

- format, whether the course was taught in-person, a blend, or virtual
- cEval_1, “Course material was presented clearly”
- cEval_13, “Where applicable, issues were considered from multiple perspectives”
- cEval_19, “My understanding of the subject matter increased over the span of the course”

4.5.1 Selecting and Formatting Variables

There are some guidelines for selecting and formatting variables for imputation.

- Variables should be in their *most natural* state
- Redundant or too highly correlated variables should not be included
 - If you reverse coded a variable (we haven't yet), that's ok, but if you have already reverse-coded, then exclude the original variable
 - Redundant variables (or multicollinear variables) may cause the multiple imputation process to cease
 - Violation of this also provides clues for troubleshooting
- Exclude variables with more than 25% missing

To make this as realistic as possible. Let's start with our very raw data. The [Scrubbing chapter](#) provides greater detail on importing data directly from Qualtrics. If you have worked the lessons, consecutively, you know that data can be added to this survey at any time. So that the values in the chapter are consistent, I will use the datafiles that I immediately saved when I conducted the analysis at the time I last updated the chapter.

Please download the .rds or .csv file from [MultivModel GitHub](#) site. Please the file in the same folder as your .rmd file. As always, I prefer working with .rds files.

```
QTRX_df2 <- readRDS("QTRX_df230902b.rds")
# QTRX_df <- read.csv('QTRX_df230902b.csv', header = TRUE)
```

Next, I apply inclusion/exclusion criteria. As described in the [Scrubbing chapter](#) this includes:

- excluding all *previews*

- including only those who consented
- including only those whose rated course was offered by a U.S. institution

```
library(tidyverse)
QTRX_df2 <- dplyr::filter(QTRX_df2, DistributionChannel != "preview")
QTRX_df2 <- dplyr::filter(QTRX_df2, Consent == 1)
QTRX_df2 <- dplyr::filter(QTRX_df2, USInst == 0)
```

Preparing the data also meant renaming some variables that started with numbers (a hassle in R). I also renamed variables on the Campus Climate scale so that we know to which subscale they belong.

```
# renaming variables that started with numbers
QTRX_df2 <- dplyr::rename(QTRX_df2, iRace1 = "1_iRace", iRace2 = "2_iRace",
                           iRace3 = "3_iRace", iRace4 = "4_iRace", iRace5 = "5_iRace", iRace6 = "6_iRace",
                           iRace7 = "7_iRace", iRace8 = "8_iRace", iRace9 = "9_iRace", iRace10 = "10_iRace")
# renaming variables from the identification of classmates
QTRX_df2 <- dplyr::rename(QTRX_df2, cmBiMulti = Race_10, cmBlack = Race_1,
                           cmNBPoC = Race_7, cmWhite = Race_8, cmUnsure = Race_2)
```

The Qualtrics download does not include an ID number. Because new variables are always appended to the end of the df, we also include code to make this the first column.

```
QTRX_df2 <- QTRX_df2 %>%
  dplyr::mutate(ID = row_number())
# moving the ID number to the first column; requires
QTRX_df2 <- QTRX_df2 %>%
  dplyr::select(ID, everything())
```

Because this huge df is cumbersome to work with, let's downsize it to be closer to the size we will work with in the imputation

```
mimp_df <- dplyr::select(QTRX_df2, ID, iRace1, iRace2, iRace3, iRace4,
                           iRace5, iRace6, iRace7, iRace8, iRace9, iRace10, cmBiMulti, cmBlack,
                           cmNBPoC, cmWhite, cmUnsure, Belong_1:Belong_3, Blst_1:Blst_6, cEval_1,
                           cEval_13, cEval_19, format)
# glimpse(mimp_df)
head(mimp_df)
```

```
# A tibble: 6 x 29
  ID iRace1 iRace2 iRace3 iRace4 iRace5 iRace6 iRace7 iRace8 iRace9 iRace10
  <int> <dbl> <dbl> <dbl> <dbl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl>
1     1      3      1      3     NA  NA     NA     NA     NA  NA     NA
2     2      3     NA     NA     NA  NA     NA     NA     NA  NA     NA
3     3      3      1     NA     NA  NA  NA     NA     NA  NA     NA
4     4      3      1      3     NA  NA     NA     NA     NA  NA     NA
```

```

5      5      1      NA      NA      NA NA      NA      NA      NA      NA
6      6      3      NA      NA      NA NA      NA      NA      NA      NA
# i 18 more variables: cmBiMulti <dbl>, cmBlack <dbl>, cmNBPoC <dbl>,
#   cmWhite <dbl>, cmUnsure <dbl>, Belong_1 <dbl>, Belong_2 <dbl>,
#   Belong_3 <dbl>, Blst_1 <dbl>, Blst_2 <dbl>, Blst_3 <dbl>, Blst_4 <dbl>,
#   Blst_5 <dbl>, Blst_6 <dbl>, cEval_1 <dbl>, cEval_13 <dbl>, cEval_19 <dbl>,
#   format <dbl>

```

4.5.2 Creating Composite Variables

Qualtrics imports many of the categorical variables as numbers. R often reads them numerically (integers or numbers). If they are directly converted to factors, R will sometimes collapse. In this example, if there is a race that is not represented (e.g., 2 for BiMulti), when the numbers are changed to factors, R will assume it's ordered and will change up the numbers. Therefore, it is ESSENTIAL to check (again and again ad nauseum) to ensure that your variables are recoding in a manner you understand.

```

mimp_df$iRace1 = factor(mimp_df$iRace1, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace2 = factor(mimp_df$iRace2, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace3 = factor(mimp_df$iRace3, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace4 = factor(mimp_df$iRace4, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace5 = factor(mimp_df$iRace5, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace6 = factor(mimp_df$iRace6, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace7 = factor(mimp_df$iRace7, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace8 = factor(mimp_df$iRace8, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace9 = factor(mimp_df$iRace9, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))
mimp_df$iRace10 = factor(mimp_df$iRace10, levels = c(0, 1, 2, 3, 4), labels = c("Black",
  "nBpoc", "BiMulti", "White", "NotNotice"))

head(mimp_df)

```

This is a quick recap of how we calculated the proportion of instructional staff who are BIPOC.

```

# creating a count of BIPOC faculty identified by each respondent
mimp_df$count.BIPOC <- apply(mimp_df[c("iRace1", "iRace2", "iRace3", "iRace4",
  "iRace5", "iRace6", "iRace7", "iRace8", "iRace9", "iRace10")], 1, function(x) sum(x %in%
  c("Black", "nBpoc", "BiMulti")))

```

```
# creating a count of all instructional faculty identified by each
# respondent
mimp_df$count.nMiss <- apply(mimp_df[c("iRace1", "iRace2", "iRace3", "iRace4",
  "iRace5", "iRace6", "iRace7", "iRace8", "iRace9", "iRace10")], 1, function(x) sum(!is.na(x))

# calculating the proportion of BIPOC faculty with the counts above
mimp_df$iBIPOC_pr = mimp_df$count.BIPOC/mimp_df$count.nMiss
```

I have included another variable, *format* that we will use as auxiliary variable. As written, these are the following meanings:

1. In-person (all persons are attending in person)
2. In person (some students are attending remotely)
3. Blended: some sessions in person and some sessions online/virtual
4. Online or virtual
5. Other

Let's recode it to have three categories:

0. 100% in-person (1)
1. Some sort of blend/mix (2, 3)
2. 100% online/virtual (4) NA. Other (5)

```
# we can assign more than one value to the same factor by repeating
# the label
mimp_df$format = factor(mimp_df$format, levels = c(1, 2, 3, 4, 5), labels = c("InPerson",
  "Blend", "Blend", "Online", is.na(5)))
```

Let's trim the df again to just include the variables we need in the imputation.

```
mimp_df <- select(mimp_df, ID, iBIPOC_pr, cmBlack, Belong_1:Belong_3, Blst_1:Blst_6,
  cEval_1, cEval_13, cEval_19, format)
```

Recall one of the guidelines was to remove variables with more than 25% missing. This code calculates the proportion missing from our variables and places them in rank order.

```
p_missing <- unlist(lapply(mimp_df, function(x) sum(is.na(x))))/nrow(mimp_df)
sort(p_missing[p_missing > 0], decreasing = TRUE)
```

	Blst_1	Blst_4	Blst_3	Blst_5	Blst_6	Belong_1	Belong_3
0.13043478	0.10144928	0.08695652	0.08695652	0.08695652	0.07246377	0.07246377	
Blst_2	Belong_2	cEval_1	cEval_19	iBIPOC_pr	cmBlack	cEval_13	
0.07246377	0.05797101	0.05797101	0.05797101	0.04347826	0.04347826	0.04347826	

Luckily, none of our variables have more than 25% missing. If we did have a variable with more than 25% missing, we would have to consider what to do about it.

Later we learn that we should eliminate case with greater than 50% missingness. Let's write code for that, now.

```
#Calculating number and proportion of item-level missingness
mimp_df$nmiss <- mimp_df %>%
  dplyr::select(iBIPOC_pr:format) %>% #the colon allows us to include all variables between
  is.na %>%
  rowSums

mimp_df<- mimp_df %>%
  dplyr::mutate(prop_miss = (nmiss/15)*100) #11 is the number of variables included in calculation

mimp_df <- dplyr::filter(mimp_df, prop_miss <= 50) #update df to have only those with at least
```

Once again, trim the df to include only the data to be included in the imputation

```
mimp_df <- select(mimp_df, ID, iBIPOC_pr, cmBlack, Belong_1:Belong_3, Blst_1:Blst_6,
  cEval_1, cEval_13, cEval_19, format)
```

4.5.3 The Multiple Imputation

Because multiple imputation is a *random* process, if we all want the same answers we need to set a *random seed*.

```
set.seed(210404) #you can pick any number you want, today I'm using today's timestamp
```

The program we will use is *mice*. *mice* assumes that each variable has a distribution and it imputes missing variables according to that distribution.

This means we need to correctly specify each variable's format/role. *mice* will automatically choose a distribution (think "format") for each variable; we can override this by changing the methods' characteristics.

The following code sets up the structure for the imputation. I'm not an expert at this – just following the Katitas example.

```
library(mice)
# runs the mice code with 0 iterations
imp <- mice(mimp_df, maxit = 0)
# Extract predictor Matrix and methods of imputation
predM = imp$predictorMatrix
meth = imp$method
```

Here we code what format/role each variable should be.

```

# These variables are left in the dataset, but setting them = 0 means
# they are not used as predictors. We want our ID to be retained in
# the df. There's nothing missing from it, and we don't want it used
# as a predictor, so it will just hang out.
predM[, c("ID")] = 0

# If you like, view the first few rows of the predictor matrix
# head(predM)

# We don't have any ordered categorical variables, but if we did we
# would follow this format poly <- c('Var1', 'Var2')

# We don't have any dichotomous variables, but if we did we would
# follow this format log <- c('Var3', 'Var4')

# Unordered categorical variables (nominal variables), but if we did
# we would follow this format
poly2 <- c("format")

# Turn their methods matrix into the specified imputation models
# Remove the hashtag if you have any of these variables meth[poly] =
# 'polr' meth[log] = 'logreg'
meth[poly2] = "polyreg"

meth

```

ID	iBIPOC_pr	cmBlack	Belong_1	Belong_2	Belong_3	Blst_1	Blst_2
"	"pmm"	"	"pmm"	"	"pmm"	"pmm"	"pmm"
Blst_3	Blst_4	Blst_5	Blst_6	cEval_1	cEval_13	cEval_19	format
"pmm"	"pmm"	"pmm"	"pmm"	"pmm"	"	"pmm"	"polyreg"

This list (meth) contains all our variables; “pmm” is the default and is the “predictive mean matching” process used. We see that format (an unordered categorical variable) is noted as “polyreg.” If we had used other categorical variables (ordered/poly, dichotomous/log), we would have seen those designations, instead. If there is “ ” underneath it means the data is complete.

Our variables of interest are now configured to be imputed with the imputation method we specified. Empty cells in the method matrix mean that those variables aren’t going to be imputed.

If a variable has no missing values, it is automatically set to be empty. We can also manually set variables to not be imputed with the *meth[variable]=“ ”* command.

The code below begins the imputation process. We are asking for 5 datasets. If you have many cases and many variables, this can take awhile. How many imputations? Recommendations have ranged as low as five to several hundred.

```

# With this command, we tell mice to impute the mimp_df data, create
# 5 datasets, use predM as the predictor matrix and don't print the

```

```
# imputation process. If you would like to see the process (or if
# the process is failing to execute) set print as TRUE; seeing where
# the execution halts can point to problematic variables (more notes
# at end of lecture)

imp2 <- mice(mimp_df, maxit = 5, predictorMatrix = predM, method = meth,
              print = FALSE)
```

We need to create a “long file” that stacks all the imputed data. Looking at the df in R Studio shows us that when imp = 0 (the pre-imputed data), there is still missingness. As we scroll through the remaining imputations, there are no NA cells.

```
# First, turn the datasets into long format This procedure is, best I
# can tell, unique to mice and wouldn't work for repeated measures
# designs
mimp_long <- mice::complete(imp2, action = "long", include = TRUE)
```

If we look at it, we can see 6 sets of data. If the *ID* variable is sorted we see that:

- .imp = 0 is the unimputed set; there are still missing values
- .imp = 1, 2, 3, or 5 has no missing values for the variables we included in the imputation

With the code below we can see the proportion of missingness for each variable (that has missing data), sorted from highest to lowest.

```
p_missing_mimp_long <- unlist(lapply(mimp_long, function(x) sum(is.na(x))))/nrow(mimp_long)
sort(p_missing_mimp_long[p_missing_mimp_long > 0], decreasing = TRUE) #check to see if this w
```

Blst_1	Blst_4	iBIPOC_pr	Blst_3	Blst_5	Blst_6
0.012820513	0.007692308	0.005128205	0.005128205	0.005128205	0.005128205
Belong_1	Belong_3	Blst_2	cEval_1	cEval_19	
0.002564103	0.002564103	0.002564103	0.002564103	0.002564103	

4.5.4 Creating Scale Scores

Because our imputation was item-level, we need to score the variables with scales/subscales. As demonstrated more completely in the [Scoring chapter](#), this required reversing one item in the campus climate scale:

```
mimp_long <- mimp_long %>%
  mutate(rBlst_1 = 8 - Blst_1) #if you had multiple items, you could add a pipe (%>%) at th
```

Below is the scoring protocol we used in the AIA protocol for scoring. Although the protocol below functionally says, “Create a mean score if (65-80)% is non-missing, for the imputed version, it doesn’t harm anything to leave this because there is no missing data.

```
# Making the list of variables
Belonging_vars <- c("Belong_1", "Belong_2", "Belong_3")
ResponseBL_vars <- c("rBlst_1", "Blst_4", "Blst_6")
StigmaBL_vars <- c("Blst_2", "Blst_3", "Blst_5")
ClimateBL_vars <- c("rBlst_1", "Blst_4", "Blst_6", "Blst_2", "Blst_3",
"Blst_5")

# Creating the new variables
mimp_long$Belonging <- sjstats::mean_n(mimp_long[, Belonging_vars], 0.65)
mimp_long$ResponseBL <- sjstats::mean_n(mimp_long[, ResponseBL_vars], 0.8)
mimp_long$StigmaBL <- sjstats::mean_n(mimp_long[, StigmaBL_vars], 0.8)
mimp_long$ClimateBL <- sjstats::mean_n(mimp_long[, ClimateBL_vars], 0.8)
```

4.6 Multiple Regression with Multiply Imputed Data

For a refresher, here was the script when we used the AIA approach for managing missingness:

```
Climate_fit <- lm(ClimateBL ~ Belonging + cmBlack + iBIPOC_pr, data = item_scores_df)
summary(Climate_fit)
```

In order for the regression to use multiply imputed data, it must be a “mids” (multiply imputed data sets) type

```
# Convert to mids type - mice can work with this type
mimp_mids <- as.mids(mimp_long)
```

Here's what we do with imputed data:

```
fitimp <- with(mimp_mids, lm(ClimateBL ~ Belonging + cmBlack + iBIPOC_pr))
```

In this process, 5 individual, OLS, regressions are being conducted and the results being pooled into this single set.

```
# to get the 5, individual imputations
summary(fitimp)
```

term	estimate	std.error	statistic	p.value	nobs
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
1 (Intercept)	3.02	0.435	6.95	0.00000000283	65
2 Belonging	-0.0311	0.0897	-0.346	0.730	65
3 cmBlack	-0.0206	0.0165	-1.25	0.215	65
4 iBIPOC_pr	-0.663	0.339	-1.95	0.0552	65
5 (Intercept)	3.02	0.446	6.77	0.00000000578	65

6	Belonging	-0.0349	0.0907	-0.385	0.702	65
7	cmBlack	-0.0234	0.0166	-1.41	0.165	65
8	iBIPOC_pr	-0.470	0.329	-1.43	0.158	65
9	(Intercept)	3.01	0.450	6.70	0.00000000744	65
10	Belonging	-0.0349	0.0915	-0.381	0.704	65
11	cmBlack	-0.0222	0.0167	-1.33	0.187	65
12	iBIPOC_pr	-0.485	0.330	-1.47	0.147	65
13	(Intercept)	2.95	0.448	6.57	0.0000000127	65
14	Belonging	-0.0152	0.0920	-0.165	0.870	65
15	cmBlack	-0.0216	0.0168	-1.29	0.203	65
16	iBIPOC_pr	-0.558	0.343	-1.62	0.110	65
17	(Intercept)	3.00	0.452	6.64	0.00000000963	65
18	Belonging	-0.0311	0.0921	-0.337	0.737	65
19	cmBlack	-0.0214	0.0168	-1.28	0.207	65
20	iBIPOC_pr	-0.531	0.337	-1.57	0.121	65

```
pool(fitimp)
```

	Class:	mipo	m = 5	term	m	estimate	ubar	b	t	dfcom
1	(Intercept)	5	2.99980658	0.1990231323	0.000999315858	0.2002223113				61
2	Belonging	5	-0.02940746	0.0083160161	0.000067017541	0.0083964371				61
3	cmBlack	5	-0.02184241	0.0002777582	0.000001056566	0.0002790261				61
4	iBIPOC_pr	5	-0.54138195	0.1128248694	0.005817914953	0.1198063673				61
				df	riv	lambda	fmi			
1	58.70890	0.006025325	0.005989238	0.03820536						
2	58.44929	0.009670622	0.009577997	0.04181342						
3	58.80737	0.004564689	0.004543947	0.03675551						
4	53.13966	0.061879070	0.058273179	0.09182261						

```
summary(pool(fitimp))
```

	term	estimate	std.error	statistic	df	p.value
1	(Intercept)	2.99980658	0.44746208	6.7040465	58.70890	0.000000008735881
2	Belonging	-0.02940746	0.09163207	-0.3209298	58.44929	0.749408305666738
3	cmBlack	-0.02184241	0.01670407	-1.3076097	58.80737	0.196094825405891
4	iBIPOC_pr	-0.54138195	0.34613056	-1.5640975	53.13966	0.123730969370680

Results of a multiple regression predicting the respondents' perceptions of campus climate for Black students indicated that neither contributions of the respondents' personal belonging ($B = -0.029, p = 0.749$), the proportion of BIPOC instructional staff ($B = -0.541, p = 0.124$), nor proportion of Black classmates ($B = -0.022, p = 0.196$) led to statistically significant changes in perceptions of campus climate for Black students. Results are presented in Table X.

4.7 Toward the APA Style Write-up

4.7.1 Method/Data Diagnostics

Data screening suggested that 107 individuals opened the survey link. Of those, 83 granted consent and proceeded into the survey items. A further inclusion criteria was that the course was taught in the U.S; 69 met this criteria.

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 67%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data. At this stage in the analysis, we allowed all cases with fewer than 50% missing to be included the multiple imputation [Katitas, 2019].

Regarding the distributional characteristics of the data, skew and kurtosis values of the variables fell below the values of 3 (skew) and 10 (kurtosis) that Kline suggests are concerning [2016b]. Results of the Shapiro-Wilk test of normality indicate that our variables assessing the proportion of classmates who are Black ($W = 0.878, p < 0.001$) and the proportion of BIPOC instructional staff($W = 0.787, p < 0.001$) are statistically significantly different than a normal distribution. The scales assessing the respondent's belonging (0.973, $p = 0.165$) and the respondent's perception of campus climate for Black students ($W = 0.951, p = 0.016$) did not differ differently from a normal distribution.

We evaluated multivariate normality with the Mahalanobis distance test. Specifically, we used the *psych::outlier()* function and included all continuous variables in the calculation. Our visual inspection of the Q-Q plot suggested that the plotted line strayed from the straight line as the quantiles increased. Additionally, we appended the Mahalanobis distance scores as a variable to the data. Analyzing this variable, we found that 1 case exceed three standard deviations beyond the median. Given that the Mahalanobis distance values increased in a consistent manner (i.e., no extreme “jumps”) we retained all cases.

We managed missing data with multiple imputation [Enders, 2017, Katitas, 2019]. We imputed five sets of data with the R package, *mice* (v. 3.13) – a program that utilizes conditional multiple imputation. The imputation included the item-level variables that comprised our scales, the variables that represented proportion of BIPOC instructional staff and proportion of Black classmates, as well as four auxiliary variables (three variables from the course evaluation and the format [in-person/blended/virtual] of the class).

4.7.2 Results

Results of a multiple regression predicting the respondents' perceptions of campus climate for Black students indicated that neither contributions of the respondents' personal belonging ($B = -0.029, p = 0.749$), *the proportion of BIPOC instructional staff* ($B = -0.541, p = 0.124$), nor proportion of Black classmates ($B = -0.022, p = 0.196$) led to statistically significant changes in perceptions of campus climate for Black students. Results are presented in Table X.

Some notes about this write-up

- I went ahead and used the data diagnostics that we did in the AIA method. It feels to me like these should be calculated with the multiply imputed data (i.e., 5 sets, with pooled estimates and standard errors), but I do not see that modeled – anywhere in R.
- Note the similarities with the AIA write-up.

4.8 Multiple imputation considerations

- Character vectors (i.e., values that are represented with words) can be problematic. If they are causing trouble, consider
 - recode into factors,
 - keep it in the df, but exclude it from the imputation protocol,
 - our “format” variable was an ordered factor (i.e., each term was associated with a value), so I think that helped us avoid problems
- Variables with really high (like 50% or more) proportions of missingness should be excluded.
- Variables that are highly correlated or redundant (even if inverse) will halt the execution. If you set `print=TRUE` you will see where the algorithm is having difficulty because it will halt at that variable.
- Variables with non-missing values can be problematic. If they are problematic, just exclude them from the process. *Width (columns/variables) versus length (rows/cases). You must have more rows/cases than columns/variables. It is difficult to say how many. If this is a problem:
 - Consider scoring scales first with AIA, then impute with whole scales.
 - Divide the df in halves or thirds, impute separately, then join with the ID numbers.
 - There should be auxiliary variables in each. *Item-level imputation is its “whole big thing” with multiple, complex considerations. There are tremendous resources
 - Enders [BLIMP](#) app is free and works with R
 - Little’s [2002] article
- How many imputations? Controversial and has changed over the years.
 - Practical concern: the more you request, the longer it will take in R, this demo was 5
 - For a number of years there was a push for 20, but I’ve also seen recommendations for 100s.

- Check examples of imputed studies in your disciplinary specialty/journals.
- There are lots of discussions and debates about
 - allowing for fractional/decimal responses (a 3.5 on 1 to 4 scaling; or a 0.75 on a dichotomous variable such as male/female)
 - out-of-bounds estimates (what if you get a 7 on 1 to 4 scaling?)

4.9 Practice Problems

The three problems described below are designed to be continuations from the previous chapters. You will likely encounter challenges that were not covered in this chapter. Search for and try out solutions, knowing that there are multiple paths through the analysis. In addition to the scrubbing, scoring, and data diagnostic skills learned in the prior lessons, the overall notion of the suggestions for practice are to (a) multiply impute a minimum of 5 sets of data, (b) repeat the regression (attempted in the Data Dx chapter), (c) create APA style write-ups of the multiple imputation method and regression results, and (d) explain it to someone.

4.9.1 Problem #1: Reworking the Chapter Problem

If you chose this option in the prior chapters, you imported the data from Qualtrics, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, properly formatted the variables, interpreted item-level missingness, scored the scales/subscales, interpreted scale-level missingness, and wrote up the results. Please continue with the remaining tasks.

4.9.2 Problem #2: Use the *Rate-a-Recent-Course* Survey, Choosing Different Variables

If you chose this option in the prior chapter, you chose a minimum of three variables from the *Rate-a-Recent-Course* survey to include in a simple statistical model. You imported the data from Qualtrics, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, properly formatted the variables, interpreted item-level missingness, scored the scales/subscales, interpreted scale-level missingness, and wrote up the results. Please continue with the remaining tasks.

4.9.3 Problem #3: Other data

If you chose this option in the prior chapter, you used raw data that was available to you. You imported it into R, applied inclusion/exclusion criteria, renamed variables, downsized the df to the variables of interest, properly formatted the variables, interpreted item-level missingness, scored the scales/subscales, interpreted scale-level missingness, and wrote up the results. Please continue with the remaining tasks.

4.9.4 Grading Rubric

Assignment Component	Points Possible	Points Earned
1. Specify a research model with three predictor variables (continuously or categorically scaled) and one dependent (continuously scaled) variable.	5	_____
2. Import the raw data	5	_____
3. Apply inclusionary/exclusionary criteria	5	_____
4. Format any variables that shouldn't be imputed in their raw form	5	_____
5. Multiply impute a minimum of 5 sets of data	5	_____
6. Run a regression (for multiply imputed data) with at least three variables	5	_____
7. APA style write-up of the multiple imputation section of data diagnostics	5	_____
8. APA style write-up regression results	5	_____
9. Explanation to grader	5	_____
Totals	45	_____

4.10 Homeworked Example

Screencast Link

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

Although the lessons focused on preparing data for analyses were presented in smaller sections, this homeworked example combines the suggestions for practice from the [Scrubbing](#), [Scoring](#), [Data Dx](#) because they are also used when missing data is managed with multiple imputation. My hope is that is cumulative presentation is a closer approximation of what researchers need for their research projects.

These lessons were created to prepare a set of data to analyze a specific research model. Consequently, the model should be known and described at the beginning.

4.10.1 Scrubbing

Specify a research model

A further assignment requirement was that the model should include three predictor variables (continuously or categorically scaled) and one dependent (continuously scaled) variable.

As in the homeworked example for the Data Dx lesson, I am hypothesizing that socially responsive pedagogy (my dependent variable) will increase as a function of:

- the transition from SPSS (0) to R(1),
- the transition from a pre-centered (0) to re-centered (1) curriculum, and
- higher evaluations of traditional pedagogy

Because this data is nested within the person (i.e., students can contribute up to three course evaluations over the ANOVA, multivariate, and psychometrics courses) proper analysis would require a statistic (e.g., multilevel modeling) that would address the dependency in the data. Therefore, I will include only those students who are taking the multivariate modeling class.

While it is possible to conduct multiple imputation at the scale level, we will do so at the item-level (i.e., before we compute the scale scores).

If you wanted to use this example and dataset as a basis for a homework assignment, you could create a different subset of data. I worked the example for students taking the multivariate modeling class. You could choose ANOVA or psychometrics. You could also choose a different combinations of variables.

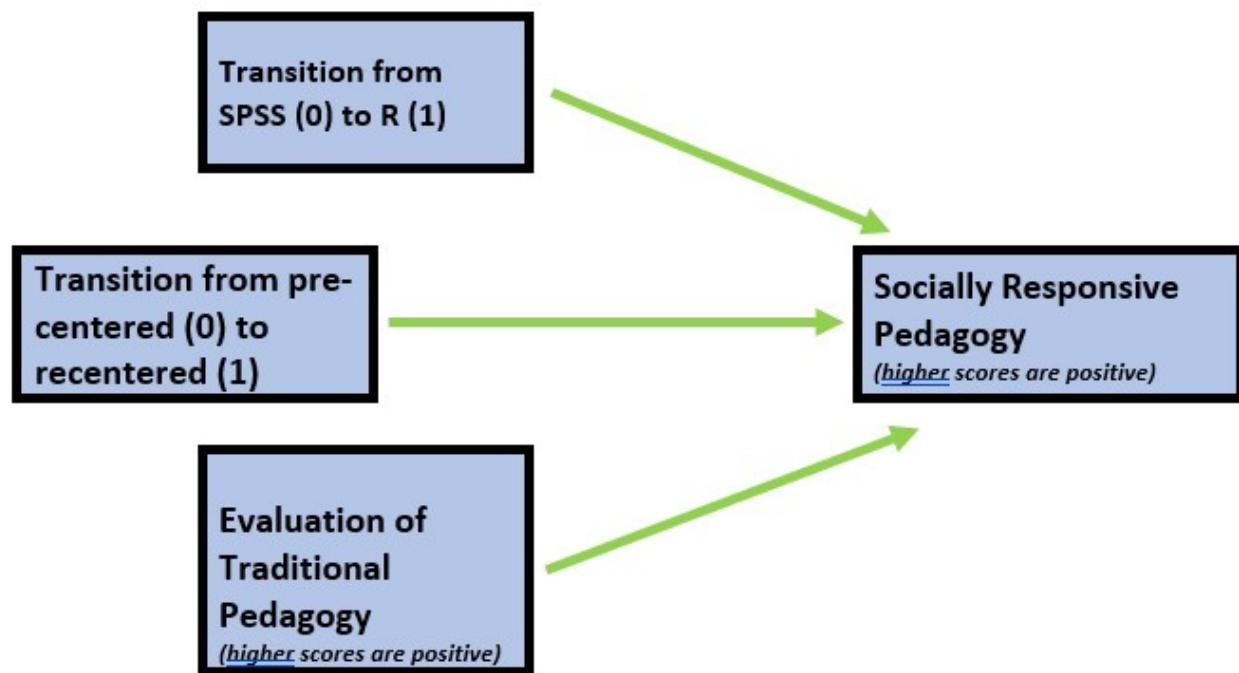


Figure 4.5: An image of our the prediction model for the homeworked example.

Import data

```
raw <- readRDS("ReC.rds")
nrow(raw)
```

[1] 310

Apply inclusionary/exclusionary criteria

Because this data is publicly posted on the Open Science Framework, it was necessary for me to already exclude those individuals. This data was unique in that students could freely write some version of “Opt out.” My original code included a handful of versions, but here was the basic form:

```
# testing to see if my code worked raw <- dplyr::filter (raw,
# SPFC.Decolonize.Opt.Out != 'Okay')
raw <- dplyr::filter(raw, SPFC.Decolonize.Opt.Out != "Opt Out")
```

I want to exclude students’ responses for the ANOVA and psychometrics courses.

```
raw <- dplyr::filter(raw, Course == "Multivariate")
```

At this point, these my only inclusion/exclusion criteria. I can determine how many students (who consented) completed any portion of the survey.

```
nrow(raw)
```

```
[1] 84
```

4.10.1.1 Format any variables that shouldn’t be imputed in their raw form

Let’s first create a df with the item-level variables that will fuel our model.

In addition to the variables in our model, we will include four auxiliary variables. These include Dept (Department: Clinical or Industrial-Organizational) and four additional course evaluation items: OvInstructor, MyContribution, IncrInterest, IncrUnderstanding.

Let’s check the structure to be certain that *StatsPkg* (SPSS, R) and *Centered* (Pre, Re) are ordered factors. We also want the course evaluation items to be integer (or numerical).

```
mimp_df <- dplyr::select(raw, deID, StatsPkg, Centering, ClearResponsibilities,
  EffectiveAnswers, Feedback, ClearOrganization, ClearPresentation, InclusvClassrm,
  EquitableEval, MultPerspectives, DEIIntegration, Dept, OvInstructor,
  MyContribution, IncrInterest, IncrUnderstanding)
str(mimp_df)
```

```
Classes 'data.table' and 'data.frame': 84 obs. of 17 variables:
 $ deID           : int 11 12 13 14 15 16 17 18 35 19 ...
 $ StatsPkg       : Factor w/ 2 levels "SPSS","R": 2 2 2 2 2 2 2 2 2 2 ...
 $ Centering      : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 2 ...
 $ ClearResponsibilities: int 4 5 5 5 4 3 5 5 3 5 ...
 $ EffectiveAnswers : int 4 5 5 4 4 3 5 5 4 4 ...
 $ Feedback        : int 4 5 4 4 5 4 5 4 4 5 ...
 $ ClearOrganization: int 3 5 5 4 4 3 5 5 4 5 ...
```

```
$ ClearPresentation      : int  4 5 5 3 4 2 5 4 5 5 ...
$ InclusvClassrm       : int  5 5 5 5 4 5 5 5 5 ...
$ EquitableEval         : int  4 5 5 5 4 4 5 4 5 5 ...
$ MultPerspectives      : int  4 5 5 5 5 5 5 4 5 5 ...
$ DEIintegration        : int  5 5 5 5 5 5 5 5 5 5 ...
$ Dept                  : chr  "CPY" "CPY" "CPY" "CPY" ...
$ OvInstructor          : int  3 5 5 3 5 2 5 4 5 5 ...
$ MyContribution        : int  4 5 4 3 4 3 5 4 4 5 ...
$ IncrInterest          : int  4 5 4 3 4 3 5 4 5 4 ...
$ IncrUnderstanding     : int  4 5 5 3 4 3 5 4 5 5 ...
- attr(*, ".internal.selfref")=<externalptr>
```

```
mimp_df$Dept <- factor(mimp_df$Dept, levels = c("CPY", "ORG"))
str(mimp_df$Dept)
```

Factor w/ 2 levels "CPY","ORG": 1 1 1 1 1 1 1 1 1 1 ...

We should eliminate case with greater than 50% missingness.

```
library(tidyverse)
#Calculating number and proportion of item-level missingness
mimp_df$nmiss <- mimp_df%>%
  dplyr::select(StatsPkg:IncrUnderstanding) %>% #the colon allows us to include all variables
  is.na %>%
  rowSums

mimp_df<- mimp_df%>%
  dplyr::mutate(prop_miss = (nmiss/13)*100) #11 is the number of variables included in calculation

mimp_df <- filter(mimp_df, prop_miss <= 50) #update df to have only those with at least 50% o
```

Once again, trim the df to include only the data to be included in the imputation

```
mimp_df <- dplyr::select(mimp_df, deID, StatsPkg, Centering,ClearResponsibilities, EffectiveA
```

4.10.1.2 Multiply impute a minimum of 5 sets of data

Because multiple imputation is a *random* process, if we all want the same answers we need to set a *random seed*.

```
set.seed(2309034) #you can pick any number you want, today I'm using today's timestamp
```

The program we will use is *mice*. *mice* assumes that each variable has a distribution and it imputes missing variables according to that distribution.

This means we need to correctly specify each variable's format/role. *mice* will automatically choose a distribution (think “format”) for each variable; we can override this by changing the methods' characteristics.

The following code sets up the structure for the imputation. This follows the Katitas example.

```
library(mice)
# runs the mice code with 0 iterations
imp <- mice(mimp_df, maxit = 0)
# Extract predictor Matrix and methods of imputation
predM = imp$predictorMatrix
meth = imp$method
log = imp$log
```

Here we code what format/role each variable should be.

```
# These variables are left in the dataset, but setting them = 0 means
# they are not used as predictors. We want our ID to be retained in
# the df. There's nothing missing from it, and we don't want it used
# as a predictor, so it will just hang out.
predM[, c("deID")] = 0

# If you like, view the first few rows of the predictor matrix
# head(predM)

# We don't have any ordered categorical variables, but if we did we
# would follow this format poly <- c('Var1', 'Var2')

# We have three dichotomous variables
log <- c("StatsPkg", "Centering", "Dept")

# Unordered categorical variables (nominal variables), but if we did
# we would follow this format poly2 <- c('format')

# Turn their methods matrix into the specified imputation models
# Remove the hashtag if you have any of these variables meth[poly] =
# 'polr'
meth[log] = "logreg"
# meth[poly2] = 'polyreg'

meth
```

	StatsPkg	Centering
deID	"logreg"	"logreg"
ClearResponsibilities	EffectiveAnswers	Feedback
"pmm"	"	"pmm"
ClearOrganization	ClearPresentation	InclusvClassrm
""	""	"pmm"

EquitableEval	MultPerspectives	DEIintegration
" "	"pmm"	"pmm"
Dept	OvInstructor	MyContribution
"logreg"	" "	" "
IncrInterest	IncrUnderstanding	" "
"pmm"		

This list (*meth*) contains all our variables; “pmm” is the default and is the “predictive mean matching” process used. We see that *StatsPkg* and *Centering* are noted as “logreg.” This is because they are dichotomous variables. If there is “ ” underneath it means the data is complete. The data will be used in imputing other data, but none of that data will be imputed.

Our variables of interest are now configured to be imputed with the imputation method we specified. Empty cells in the method matrix mean that those variables aren’t going to be imputed.

If a variable has no missing values, it is automatically set to be empty. We can also manually set variables to not be imputed with the *meth[variable] = “ ”* command.

The code below begins the imputation process. We are asking for 5 datasets. If you have many cases and many variables, this can take awhile. How many imputations? Recommendations have ranged as low as five to several hundred.

```
# With this command, we tell mice to impute the anesimpor2 data,
# create 5vudatasets, use predM as the predictor matrix and don't
# print the imputation process. If you would like to see the process
# (or if the process is failing to execute) set print as TRUE; seeing
# where the execution halts can point to problematic variables (more
# notes at end of lecture)

imp2 <- mice(mimp_df, maxit = 5, predictorMatrix = predM, method = meth,
  log = log, print = FALSE)
```

We need to create a “long file” that stacks all the imputed data. Looking at the df in R Studio shows us that when *imp* = 0 (the pre-imputed data), there is still missingness. As we scroll through the remaining imputations, there are no NA cells.

```
# First, turn the datasets into long format This procedure is, best I
# can tell, unique to mice and wouldn't work for repeated measures
# designs
mimp_long <- mice::complete(imp2, action = "long", include = TRUE)
```

If we look at it, we can see 6 sets of data. If the *deID* variable is sorted we see that:

- *.imp* = 0 is the unimputed set; there are still missing values
- *.imp* = 1, 2, 3, or 5 has no missing values for the variables we included in the imputation

With the code below we can see the proportion of missingness for each variable (that has missing data), sorted from highest to lowest.

```
p_missing_mimp_long <- unlist(lapply(mimp_long, function(x) sum(is.na(x))))/nrow(mimp_long)
sort(p_missing_mimp_long[p_missing_mimp_long > 0], decreasing = TRUE) #check to see if this works
```

DEIIntegration	InclusvClassrm	Feedback
0.027777778	0.007936508	0.003968254

ClearResponsibilities	MultPerspectives	IncrInterest
0.001984127	0.001984127	0.001984127

Because our imputation was item-level, we need to score the variables with scales/subscales.

Traditional pedagogy is a predictor variable that needs to be created by calculating the mean if at least 75% of the items are non-missing. None of the items need to be reverse-scored. I will return to working with the *scrub_df* data.

```
# this seems to work when I build the book, but not in 'working the
# problem'
TradPed_vars <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback",
  "ClearOrganization", "ClearPresentation")
# mimp_long$TradPed <- sjstats::mean_n(mimp_long[, TradPed_vars],
# .75)

# this seems to work when I 'work the problem' (but not when I build
# the book) the difference is the two dots before the last SRPed_vars
mimp_long$TradPed <- sjstats::mean_n(mimp_long[, TradPed_vars], 0.75)
```

The dependent variable is socially responsive pedagogy. It needs to be created by calculating the mean if at least 75% of the items are non-missing. None of the items need to be reverse-scored.

```
# this seems to work when I build the book, but not in 'working the
# problem' SRPed_vars <- c('InclusvClassrm', 'EquitableEval',
# 'MultPerspectives', 'DEIIntegration') mimp_long$SRPed <-
# sjstats::mean_n(mimp_long[, SRPed_vars], .75)

# this seems to work when I 'work the problem' (but not when I build
# the book) the difference is the two dots before the last SRPed_vars
SRPed_vars <- c("InclusvClassrm", "EquitableEval", "MultPerspectives",
  "DEIIntegration")
mimp_long$SRPed <- sjstats::mean_n(mimp_long[, SRPed_vars], 0.75)
```

4.10.1.3 Run a regression (for multiply imputed data) with at least three variables

For comparison, here was the script when we used the AIA approach for managing missingness:

```
SRPed_fit <- lm(SRPed ~ StatsPkg + Centering + TradPed, data = scored)
```

In order for the regression to use multiply imputed data, it must be a “mids” (multiply imputed data sets) type

```
# Convert to mids type - mice can work with this type
mimp_mids <- as.mids(mimp_long)
```

Here's what we do with imputed data:

```
fitimp <- with(mimp_mids, lm(SRPed ~ StatsPkg + Centering + TradPed))
```

In this process, 5 individual, OLS, regressions are being conducted and the results being pooled into this single set.

```
# to get the 5, individual imputations
summary(fitimp)
```

	term	estimate	std.error	statistic	p.value	nobs
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
1	(Intercept)	1.90	0.310	6.13	3.13e- 8	84
2	StatsPkgR	0.187	0.118	1.59	1.16e- 1	84
3	CenteringRe	0.117	0.108	1.09	2.79e- 1	84
4	TradPed	0.565	0.0659	8.56	6.30e-13	84
5	(Intercept)	1.94	0.314	6.17	2.62e- 8	84
6	StatsPkgR	0.191	0.119	1.60	1.13e- 1	84
7	CenteringRe	0.110	0.109	1.01	3.17e- 1	84
8	TradPed	0.557	0.0667	8.36	1.63e-12	84
9	(Intercept)	1.96	0.313	6.26	1.80e- 8	84
10	StatsPkgR	0.178	0.119	1.50	1.38e- 1	84
11	CenteringRe	0.111	0.109	1.02	3.10e- 1	84
12	TradPed	0.555	0.0665	8.35	1.69e-12	84
13	(Intercept)	2.03	0.325	6.24	1.98e- 8	84
14	StatsPkgR	0.185	0.123	1.50	1.38e- 1	84
15	CenteringRe	0.104	0.113	0.918	3.62e- 1	84
16	TradPed	0.539	0.0691	7.80	1.95e-11	84
17	(Intercept)	1.91	0.306	6.26	1.77e- 8	84
18	StatsPkgR	0.158	0.116	1.36	1.78e- 1	84
19	CenteringRe	0.117	0.107	1.10	2.76e- 1	84
20	TradPed	0.567	0.0649	8.73	2.93e-13	84

```
summary(pool(fitimp))
```

	term	estimate	std.error	statistic	df	p.value
1	(Intercept)	1.9480744	0.31833535	6.119567	74.55114	0.000000040039269753
2	StatsPkgR	0.1798400	0.11996611	1.499090	76.64577	0.137957984459613908
3	CenteringRe	0.1117906	0.10918108	1.023901	77.81162	0.309054914517060075
4	TradPed	0.5564494	0.06768356	8.221338	74.26455	0.000000000004825124

Results of a multiple regression predicting the socially responsive course evaluation ratings indicated that neither the transition from SPSS to R ($B = 0.178, p = 0.135$) nor the transition to an explicitly recentered curriculum ($B = 0.116, p = 0.285$) led to statistically significant differences. In contrast, traditional pedagogy had a $B = 0.571, p < 0.001$). Results of the regression model are presented in Table 2.

4.10.1.4 APA style write-up of the multiple imputation section of data diagnostics

My write-up draws from some of the results we obtained in the homeworked example at the end of the **Data Dx** chapter.

This is a secondary analysis of data involved in a more comprehensive dataset that included students taking multiple statistics courses ($N = 310$). Having retrieved this data from a repository in the Open Science Framework, only those who consented to participation in the study were included. Data used in these analyses were 84 students who completed the multivariate clas.

Across cases that were deemed eligible on the basis of the inclusion/exclusion criteria, missingness ranged from 0 to 100%. Across the dataset, 3.86% of cells had missing data and 87.88% of cases had nonmissing data. At this stage in the analysis, missingness for all cases did not exceed 50% [Katitas, 2019] and they were all included in the multiple imputation .

Regarding the distributional characteristics of the data, skew and kurtosis values of the variables fell below the values of 3 (skew) and 10 (kurtosis) that Kline suggests are concerning [2016b]. Results of the Shapiro-Wilk test of normality indicate that our variables assessing the traditional pedagogy ($W = 0.830, p < 0.001$) and socially responsive pedagogy (0.818, $p < 0.001$) are statistically significantly different than a normal distribution. Inspection of distributions of the variables indicated that both course evaluation variables were negatively skewed, with a large proportion of high scores.

We evaluated multivariate normality with the Mahalanobis distance test. Specifically, we used the *psych::outlier()* function and included both continuous variables in the calculation. Our visual inspection of the Q-Q plot suggested that the plotted line strayed from the straight line as the quantiles increased. Additionally, we appended the Mahalanobis distance scores as a variable to the data. Analyzing this variable, we found that 2 cases exceed three standard deviations beyond the median.

We managed missing data with multiple imputation [Enders, 2017, Katitas, 2019]. We imputed five sets of data with the R package, *mice* (v. 3.13) – a program that utilizes conditional multiple imputation. The imputation included the 9 item-level variables that comprised our scales and the dichotomous variable representing traditional pedagogy and socially responsive pedagogy. We also included five auxiliary variables (four variables from the course

evaluation and the whether the student was from the Clinical or Industrial-Organizational Psychology program).

4.10.1.5 APA style write-up regression results

Results of a multiple regression predicting the socially responsive course evaluation ratings indicated that neither the transition from SPSS to R ($B = 0.178, p = 0.135$) nor the transition to an explicitly recentered curriculum ($B = 0.116, p = 0.285$) led to statistically significant differences. In contrast, traditional pedagogy had a $B = 0.571, p < 0.001$). Results of the regression model are presented in Table 2.

As in the lesson itself, I used the data diagnostics that we did in the AIA method. It feels to me like these should be calculated with the multiply imputed data (i.e., 5 sets, with pooled estimates and standard errors), but I do not see that modeled – anywhere in tutorials I consulted.

MEDIATION

Chapter 5

Simple Mediation

[Screencasted Lecture Link](#)

The focus of this lecture is to estimate indirect effects (aka “mediation”). We examine the logic/design required to support the argument that *mediation* is the *mechanism* that explains the $X \rightarrow Y$ relationship. We also work three examples (one with covariates).

At the outset, please note that although I rely heavily on Hayes [2018] text and materials, I am using the R package *lavaan* in these chapters. In recent years, Hayes has introduced a [PROCESS macro for R](#). Because I am not yet up-to-speed on using this macro (it is not a typical R package) and because ReCentering Psych Stats uses *lavaan* for confirmatory factor analysis and structural equation modeling, I have chosen to utilize the *lavaan* package. A substantial difference is that the PROCESS macros use ordinary least squares and *lavaan* uses maximum likelihood estimators.

5.1 Navigating this Lesson

There is about 1 hour and 10 minutes of lecture. If you work through the materials with me it would be plan for an additional 1.5 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](#)

5.1.1 Learning Objectives

Learning objectives from this lecture include the following:

- Define mediation and indirect effect.
- Distinguish the role of a mediating variable from independent variables, covariates, and moderators.
- Identify the conditions upon which there can be justification to support the presence of a mediated effect.

- Articulate the arguments for and against using the term, “mediation.”
- Using the R package *lavaan*,
 - specify a model with indirect effects,
 - identify and interpret B weights, *p* values, and *CIs* for total, direct, and indirect effects,
 - calculate the total effects of X and M on Y,
 - identify the proportion of variance accounted for in predicting M and Y.
- Hand calculate the values of an indirect, direct, and total effects from statistical output or a figure (just the *B* or β , not the significance level)

5.1.2 Planning for Practice

The following suggestions for practice will involve specifying, testing, and interpreting a model with a single indirect effect (mediator).

- Rework the problem in the chapter by changing the random seed in the code that simulates the data. This should provide minor changes to the data, but the results will likely be very similar.
- There are a number of variables in the dataset and there were a handful of simple mediations conducted in the journal article that sources the research vignette. Swap out one or more variables in the model of simple mediation and compare your solution to the one in the chapter and/or the research article.
- Conduct a simple mediation with data to which you have access. This could include data you simulate on your own or from a published article.

5.1.3 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.

- Hayes, A. F. (2022). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. New York, NY: Guilford Press.
 - **Chapter 3, The simple mediation model:** Hayes’ text is another great example of a teaching tool that is accessible at both procedural and conceptual levels. I especially appreciate his attention to the controversies (even those directed toward his work). We deviate from his text in that we are not using the PROCESS macro...and I’ll address those concerns in the lecture.
 - **Chapter 4, Causal steps, scaling, confounding, and causal order:** A great chapter that addresses “What happened to Baron & Kenny”; partial v complete mediation; and conditions required for claims of causality. Procedurally, our focus in this chapter is on the role of covariates.
 - **Appendix A: Using PROCESS:** An essential tool for PROCESS users because, even when we are in the R environment, this is the “idea book.” That is, the place where all the path models are presented in figures.
- Kim, P. Y., Kendall, D. L., & Cheon, H.-S. (2017). Racial microaggressions, cultural mistrust, and mental health outcomes among Asian American college students. *American Journal of Orthopsychiatry*, 87(6), 663–670. <https://doi-org.ezproxy.spu.edu/10.1037/ort0000203>

5.1.4 Packages

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

5.2 Estimating Indirect Effects (the analytic approach often termed *mediation*)

5.2.1 The definitional and conceptual

As in Hayes text [2018], we will differentiate between *moderation* and *mediation*. *Conditional process analysis* involves both! With each of these, we are seeking to understand the *mechanism* at work that leads to the relationship (be it correlational, predictive, or causal)

Even though this process has sometimes been termed *causal modeling*, Hayes argues that his *statistical approach* is not claiming to determine *cause*; that is really left to the argument of the research design.

Moderation (a review):

- Answers questions of *when* or *for whom* and is often the source of the answer, *it depends*.
- Think of our *interaction* effects in ANOVA and regression
- The effect of X on some variable Y is moderated by W if its size, sign, or strength depends on, or can be predicted, by W. Then we can say, “W is a *moderator* of X’s effect on Y” or “W and X *interact* in their influence on Y.”
- The image below illustrates moderation with *conceptual* and *statistical* diagrams. Note that three predictors (IV, DV, their interaction) point to the DV.

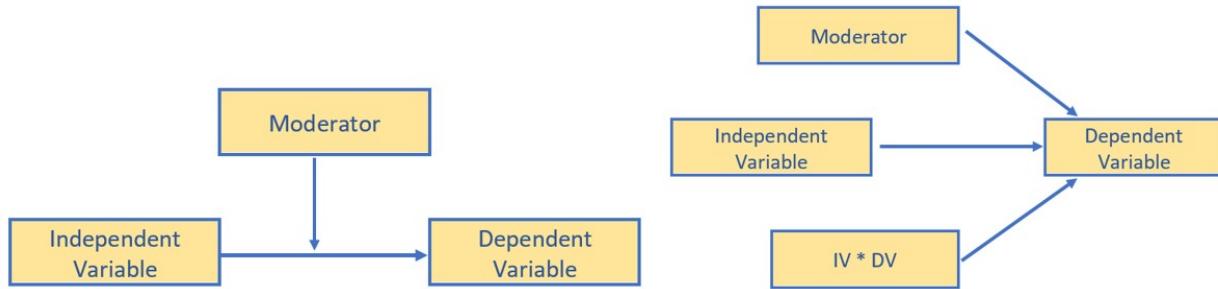


Figure 5.1: Image of Hayes's style conceptual and statistical diagrams of a simple moderation

The classic plot of moderation results is often the best way to detect that an interaction was included in the analysis and helps understand the *conditional* (e.g., for whom, under what conditions) nature of the analysis.

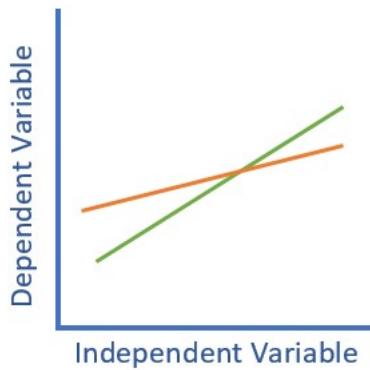


Figure 5.2: Image of classic interaction graph that illustrates a moderated effect. The IV is on the X axis, DV on the Y axis, and two intersecting lines represent the differential/moderated effect of the IV on the DV by the moderator

Mediation:

- Answers questions of *how* (I also think *through* and *via* to describe the proposed mediating mechanism)
- Paths in a mediation model are *direct* (X does not pass through M on its way to Y) and *indirect* (X passes through M on its way to Y). Once we get into the statistics, we will also be focused on *total* effects.
- Hayes thinks in terms of *antecedent* and *consequent* variables. In a 3-variable, simple mediation, X and M are the antecedent variables; X and M are the consequent variables.
- There is substantial debate and controversy about whether we can say “the effect of X on Y is *mediated* through M” or whether we should say, “There is a statistically significant indirect effect of X on Y thru M.” Hayes comes down on the “use mediation language” side of the debate.

- In sum, a simple mediation model is any causal system in which at least one causal antecedent X variable is proposed as influencing an outcome Y through a single intervening variable, M. In such a model there are two pathways by which X can influence Y.
- The figure below doubles as both the conceptual and statistical diagram of evaluating a simple mediation – a simple indirect effect.

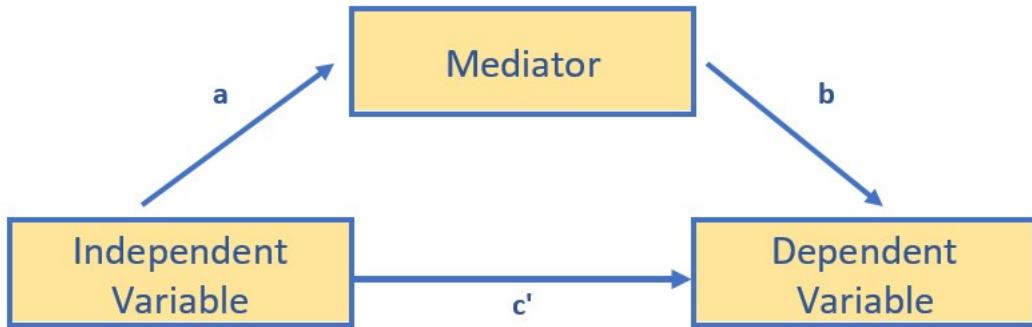


Figure 5.3: Image of Hayes's style conceptual diagram of a simple moderation

Conditional process analysis:

- Used when the research goal is to understand the boundary conditions of the mechanism(s) by which a variable transmits its effect on another.
- Typically, simultaneously, assesses the influence of mediating (indirect effects) and moderating (interactional effects) in a model-building fashion.
- In a conditional process model, the moderator(s) may be hypothesized to influence one or more of the paths.

We will work toward building a conditional process model, a moderated mediation, over the next several chapters.

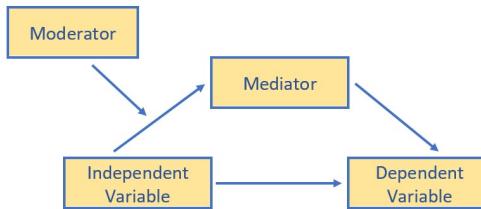


Figure 5.4: Image of conditional process analysis model where the moderator is hypothesized to change the a path; the path between the IV and mediator

5.3 Workflow for Simple Mediation

The following is a proposed workflow for conducting a simple mediation.

Conducting a simple mediation involves the following steps:

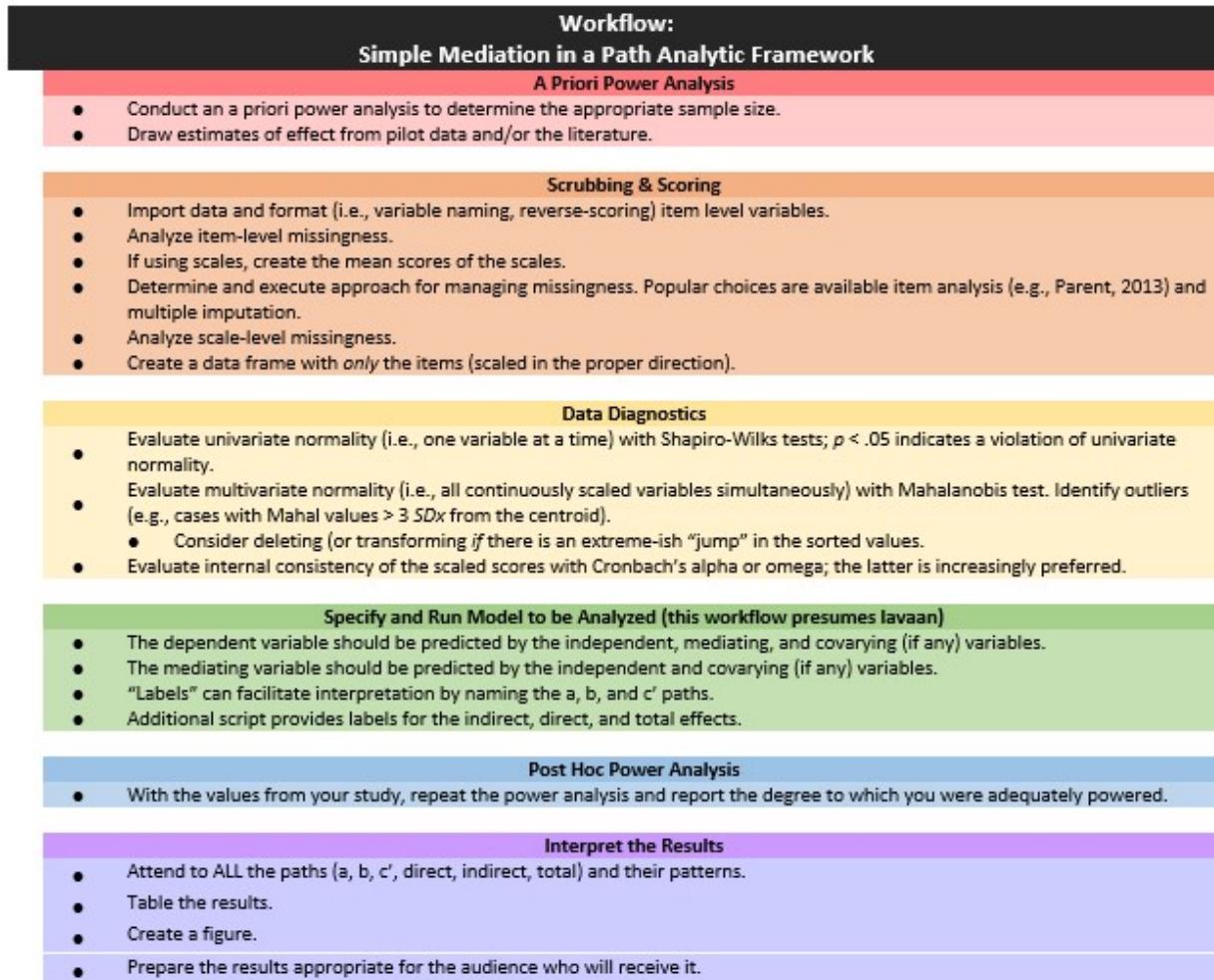


Figure 5.5: A colorful image of a workflow for the simple mediation

1. Conducting an a priori power analysis to determine the appropriate sample size.
 - This will require estimates of effect that are drawn from pilot data, the literature, or both.
2. Scrubbing and scoring the data.
 - Guidelines for such are presented in the respective lessons.
3. Conducting data diagnostics, this includes:
 - item and scale level missingness,
 - internal consistency coefficients (e.g., alphas or omegas) for scale scores,
 - univariate and multivariate normality
4. Specifying and running the model (this lesson presumes it will with the R package, *lavaan*).
 - The dependent variable should be predicted by the independent, mediating, and covarying (if any) variables.
 - “Labels” can facilitate interpretation by naming the a, b, and c’ paths. +Additional script provides labels for the indirect, direct, and total effects.
5. Conducting a post hoc power analysis.
 - Informed by your own results, you can see if you were adequately powered to detect a statistically significant effect, if, in fact, one exists.
6. Interpret and report the results.
 - Interpret ALL the paths and their patterns.
 - Create a table and figure.
 - Prepare the results in a manner that is useful to your audience.

In addition to the workflow through the statistical problem, the very traditional and classic figure below is useful in understanding the logic beneath mediation as the explanatory mechanism.

The top figure represents the bivariate relationship between the independent and dependent variable. The result of a simple linear regression (one predictor) represent the *total* effect of the IV on the DV. We can calculate this by simply regressing the DV onto the IV. The resulting *B* weight is known as the *c* path. A bivariate correlation coefficient results in the same value – only it is standardized (so would be the same as the β weight).

The lower figure represents that the relationship between the IV and DV is *mediated* by a third variable. We assign three labels to the paths: *a*, between the IV and mediator; *b*, between the mediator and DV; and *c'* (*c* prime) between the IV and DV.

Although Hayes makes a compelling case that we can claim “mediation” when there is a statistically significant indirect effect [2018], traditionally, a mediated relationship is supported when the value of *c'* is statistically significantly lower than *c*. When this occurs, then know that the mediator is sharing some of the variance (and therefore acting as a *conduit*) in the prediction of the DV.

You might already be imagining potential challenges to this model. For example, which variable should be the IV and which one should be the mediator? Can we switch them? You can – and you will likely have very similar (if not identical) results. Good research design is what provides support for suggesting that mediation is the proper, causal, mechanism regarding the relationship between

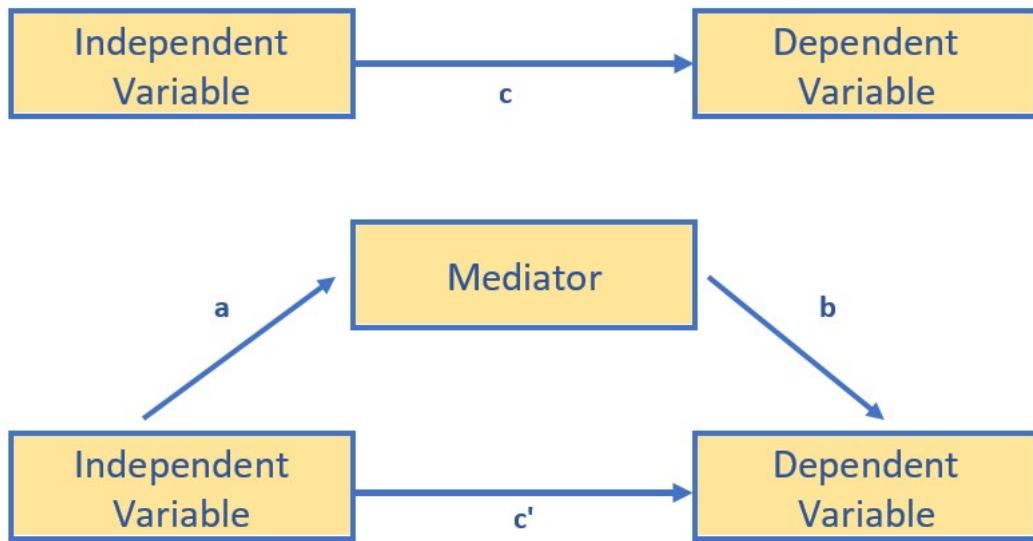


Figure 5.6: Image of conditional process analysis model where the mediator is hypothesized to change the a path; the path between the IV and mediator

the IV and DV. An excellent review of the challenges of establishing a robust mediation model is provided by Kline [2015], where he suggests the following as the minimally required elements of a mediation design:

- the IV is an experimental variable with random assignment to conditions;
- the mediator is an individual difference variable that is not manipulated and is measured at a later time; and
- the DV is measured at a third occasion

These criteria are in addition to the rather standard criteria for establishing causality [see Stone-Romero and Rosopa, 2010, for a review]:

- temporal precedence,
- statistical covariation, and
- ruling out plausible rival hypotheses.

Some journals take this very seriously. In fact [FAQs](#) in the Journal of Vocational Behavior make it clear that they will very rarely publish a “mediation manuscript” unless it has a minimum of three waves.

Working through a mediation will help operationalize these concepts.

5.4 Super Simple Mediation in *lavaan*: A focus on the mechanics

The lavaan tutorial [[Rosseel, 2020](#)] provides a helpful model of how writing code to estimate an indirect effect. Using the lavaan tutorial as our guide, let’s start with just a set of fake data with

variable names that represent X (predictor, IV, antecedent), M (mediator, atencedent, consequent), and Y (outcome, DV, consequent).

5.4.1 Simulate Fake Data

The code below is asking to create a dataset with a sample size of 100. The dataset has 3 variables, conveniently named X (predictor, antecedent, IV), M (mediator), and Y (outome, consequent, DV). The R code asks for random selection of numbers with a normal distribution. You can see that the M variable will be related to the X variable by + .5; and the Y variable will be related to the M variable by + .7. This rather ensures a statistically significant indirect effect.

```
set.seed(230916)
X <- rnorm(100)
M <- 0.5 * X + rnorm(100)
Y <- 0.7 * M + rnorm(100)
Data <- data.frame(X = X, Y = Y, M = M)
```

5.4.2 Specify Mediation Model

The package we are using is *lavaan*. Hayes' model is *path analysis*, which can be a form of structural equation modeling. As a quick reminder, in SPSS, PROCESS is limited to ordinary least squares regression. We will use maximum likliehood estimators for the Hayes/PROCESS examples, but *lavaan* can take us further than PROCESS because

- We can (and, in later chapters, will) do latent variable modeling.
- We can have more specificity and flexibility than the prescribed PROCESS models allow. I say this with all due respect to Hayes – there is also a good deal of flexibility to be able to add multiple mediators and covariates within most of the Hayes' prescribed models.

Hayes text is still a great place to start because the conceptual and procedural information is clear and transferable to the R environment.

Our atheoretical dataset makes it easy to identify which variable belongs in each role (X,Y,M). When specifying the paths in lavaan, here's what to keep in mind:

- Name your model/object (below is X, “<-” means “is defined by”)
- The model exists between 2 single quotation marks (the odd looking ' and ' at the beginning and end).
- The # of regression equations you need depends on the # of variables that have arrows pointing to them. In a simple mediation, there are 3 variables with 2 variables having arrows pointing to them – need 2 regression equations:
 - one for the Mediator
 - one for the DV (Y)
- Operator for a regression analysis is the (tilde, ~)
- DV goes on left

- In first equation we regress both the X and M onto Y
- In second equation we regress M onto X
- The asterisk (*) is a handy tool to label variables (don't confuse it as defining an interaction); this labeling as a, b, and c_p (in traditional mediation, the total effect is labeled with a and the direct effect is c'[c prime], but the script won't allow and extra single quotation mark, hence c_p) is super helpful in interpreting the output
- The indirect effect is created by multiplying the a and b paths.
- The “:=” sign is used when creating a new variable that is a function of variables in the model, but not in the dataset (i.e., the a and b path).

After specifying the model, we create an object that holds our results from the SEM. To obtain all the results from our of indirect effects, we also need to print a summary of the fit statistics, standardized estimates, r-squared, and confidence intervals.

Other authors will write the model code more sensibly, predicting the mediator first, and then the Y variable. However, I found that by doing it this way, the semPlot produces a more sensible figure.

Also, because we set a random seed, you should get the same results, but if it differs a little, don't panic. Also, in Hayes text the direct path from X to Y is c' (“c prime”; where as c is reserved for the total effect of X on Y).

Let's run the whole model.

```
model <- "
  Y ~ b*M + c_p*X
  M ~ a*X

  indirect := a*b
  direct   := c_p
  total_c  := c_p + (a*b)

  "
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
fit <- lavaan::sem(model, data = Data, se = "bootstrap", missing = "fiml")
FDsummary <- lavaan::summary(fit, standardized = T, rsq = T, fit = TRUE,
  ci = TRUE)
FD_ParamEsts <- lavaan::parameterEstimates(fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
FDsummary

## lavaan 0.6.16 ended normally after 1 iteration
##
##   Estimator                               ML
##   Optimization method                     NLMINB
##   Number of model parameters                   7
##   Number of observations                  100
```

```

## Number of missing patterns 1
##
## Model Test User Model:
##
## Test statistic 0.000
## Degrees of freedom 0
##
## Model Test Baseline Model:
##
## Test statistic 66.380
## Degrees of freedom 3
## P-value 0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI) 1.000
## Tucker-Lewis Index (TLI) 1.000
##
## Robust Comparative Fit Index (CFI) 1.000
## Robust Tucker-Lewis Index (TLI) 1.000
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0) -279.032
## Loglikelihood unrestricted model (H1) -279.032
##
## Akaike (AIC) 572.064
## Bayesian (BIC) 590.301
## Sample-size adjusted Bayesian (SABIC) 568.193
##
## Root Mean Square Error of Approximation:
##
## RMSEA 0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.000
## P-value H_0: RMSEA <= 0.050 NA
## P-value H_0: RMSEA >= 0.080 NA
##
## Robust RMSEA 0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.000
## P-value H_0: Robust RMSEA <= 0.050 NA
## P-value H_0: Robust RMSEA >= 0.080 NA
##
## Standardized Root Mean Square Residual:
##
## SRMR 0.000
##

```

```

## Parameter Estimates:
##
## Standard errors                                Bootstrap
## Number of requested bootstrap draws           1000
## Number of successful bootstrap draws          1000
##
## Regressions:
##                         Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## Y ~
##   M          (b)    0.708   0.085   8.360   0.000   0.537   0.869
##   X          (c_p) -0.107   0.112  -0.954   0.340  -0.327   0.114
## M ~
##   X          (a)    0.513   0.097   5.278   0.000   0.334   0.708
## Std.lv Std.all
##
##   0.708    0.639
##  -0.107   -0.080
##
##   0.513    0.426
##
## Intercepts:
##                         Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .Y              -0.022   0.097  -0.224   0.822  -0.212   0.179
## .M              -0.031   0.097  -0.320   0.749  -0.232   0.143
## Std.lv Std.all
##  -0.022   -0.018
##  -0.031   -0.028
##
## Variances:
##                         Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .Y              0.927   0.127   7.315   0.000   0.669   1.160
## .M              0.981   0.128   7.636   0.000   0.716   1.229
## Std.lv Std.all
##   0.927    0.629
##   0.981    0.818
##
## R-Square:
##                         Estimate
## Y                0.371
## M                0.182
##
## Defined Parameters:
##                         Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## indirect          0.363   0.084   4.328   0.000   0.216   0.543
## direct           -0.107   0.112  -0.953   0.340  -0.327   0.114
## total_c          0.257   0.120   2.132   0.033   0.024   0.507
## Std.lv Std.all
##   0.363    0.272

```

```
##   -0.107  -0.080
##   0.257   0.192
```

FD_ParamEsts

##	lhs	op	rhs	label	est	se	z	pvalue	ci.lower	ci.upper
## 1	Y	~	M	b	0.708	0.085	8.360	0.000	0.541	0.871
## 2	Y	~	X	c_p	-0.107	0.112	-0.954	0.340	-0.326	0.120
## 3	M	~	X	a	0.513	0.097	5.278	0.000	0.332	0.705
## 4	Y	~~	Y		0.927	0.127	7.315	0.000	0.713	1.252
## 5	M	~~	M		0.981	0.128	7.636	0.000	0.766	1.282
## 6	X	~~	X		0.827	0.000	NA	NA	0.827	0.827
## 7	Y	~1			-0.022	0.097	-0.224	0.822	-0.218	0.174
## 8	M	~1			-0.031	0.097	-0.320	0.749	-0.210	0.183
## 9	X	~1			-0.005	0.000	NA	NA	-0.005	-0.005
## 10	indirect	:=	a*b	indirect	0.363	0.084	4.328	0.000	0.224	0.557
## 11	direct	:=	c_p	direct	-0.107	0.112	-0.953	0.340	-0.326	0.120
## 12	total_c	:=	c_p+(a*b)	total_c	0.257	0.120	2.132	0.033	0.029	0.517
##	std.lv	std.all	std.nox							
## 1	0.708	0.639	0.639							
## 2	-0.107	-0.080	-0.088							
## 3	0.513	0.426	0.469							
## 4	0.927	0.629	0.629							
## 5	0.981	0.818	0.818							
## 6	0.827	1.000	0.827							
## 7	-0.022	-0.018	-0.018							
## 8	-0.031	-0.028	-0.028							
## 9	-0.005	-0.005	-0.005							
## 10	0.363	0.272	0.299							
## 11	-0.107	-0.080	-0.088							
## 12	0.257	0.192	0.211							

5.4.3 Interpret the Output

Note that in the script we ask (and get) two sets of parameter estimates. The second set (in the really nice dataframe) includes bootstrapped, bias-corrected confidence intervals. Bias-corrected confidence intervals have the advantage of being more powerful and bias-free. Note, though, that when the CI crosses 0, the effect is NS.

So let's look at this step-by-step.

- Overall, our model accounted for 37% of the variance in the IV and 18% of the variance in the mediator.
- a path = $B = 0.513, p < 0.001$
- b path = $0.708, p < 0.001$
- the indirect effect is a product of the a and b paths ($0.513 * 0.708 = 0.363$); while we don't hand calculate it's significance, we see that it is $p < 0.001$.

- the direct effect (c' , c prime, or c_p) is the isolated effect of X on Y when including M as a predictor. We hope this value is *lower* than the total effect because this means that including M shared some of the variance in predicting Y : $c' = -0.107, p = 0.340$, and it is no longer significant.
- we also see the total effect; this value is
 - identical to the value of simply predicting Y on X (with no M in the model)
 - the value of $a(b) + c_p$: $(0.513 * 0.708) + (-0.107) = 0.257; (p = 0.033)$

Here's a demonstration that the total effect is, simply, predicting Y from X :

```
fitXY <- lm(Y ~ X, data = Data)
summary(fitXY)
```

```
##
## Call:
## lm(formula = Y ~ X, data = Data)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -2.36350 -0.90598 -0.07158  0.74879  2.52079
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.04374   0.12035 -0.363   0.7171
## X            0.25668   0.13237  1.939   0.0554 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.204 on 98 degrees of freedom
## Multiple R-squared:  0.03695,    Adjusted R-squared:  0.02712
## F-statistic:  3.76 on 1 and 98 DF,  p-value: 0.05537
```

In a simple model such as this, it is also the same value as the bivariate correlation. The only trick is that the bivariate correlation produces a standardized result; so it would be the β .

```
library(psych)
XY_r <- corr.test(Data[c("Y", "X")])
XY_r

## Call:corr.test(x = Data[c("Y", "X")])
## Correlation matrix
##      Y   X
## Y  1.00 0.19
## X  0.19 1.00
## Sample Size
## [1] 100
```

```
## Probability values (Entries above the diagonal are adjusted for multiple tests.)  
##      Y      X  
## Y  0.00  0.06  
## X  0.06  0.00  
##  
## To see confidence intervals of the correlations, print with the short=FALSE option
```

5.4.4 A Figure and Table

We can use the package [tidySEM](#) to create a figure that includes the values on the path.

Here's what the base package gets us

```
# only worked when I used the library to turn on all these pkgs  
library(lavaan)
```

```
## This is lavaan 0.6-16  
## lavaan is FREE software! Please report any bugs.
```

```
##  
## Attaching package: 'lavaan'
```

```
## The following object is masked from 'package:psych':  
##  
##      cor2cov
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##      filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##      intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
##  
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':  
##  
##     %+%, alpha
```

```
library(tidySEM)
```

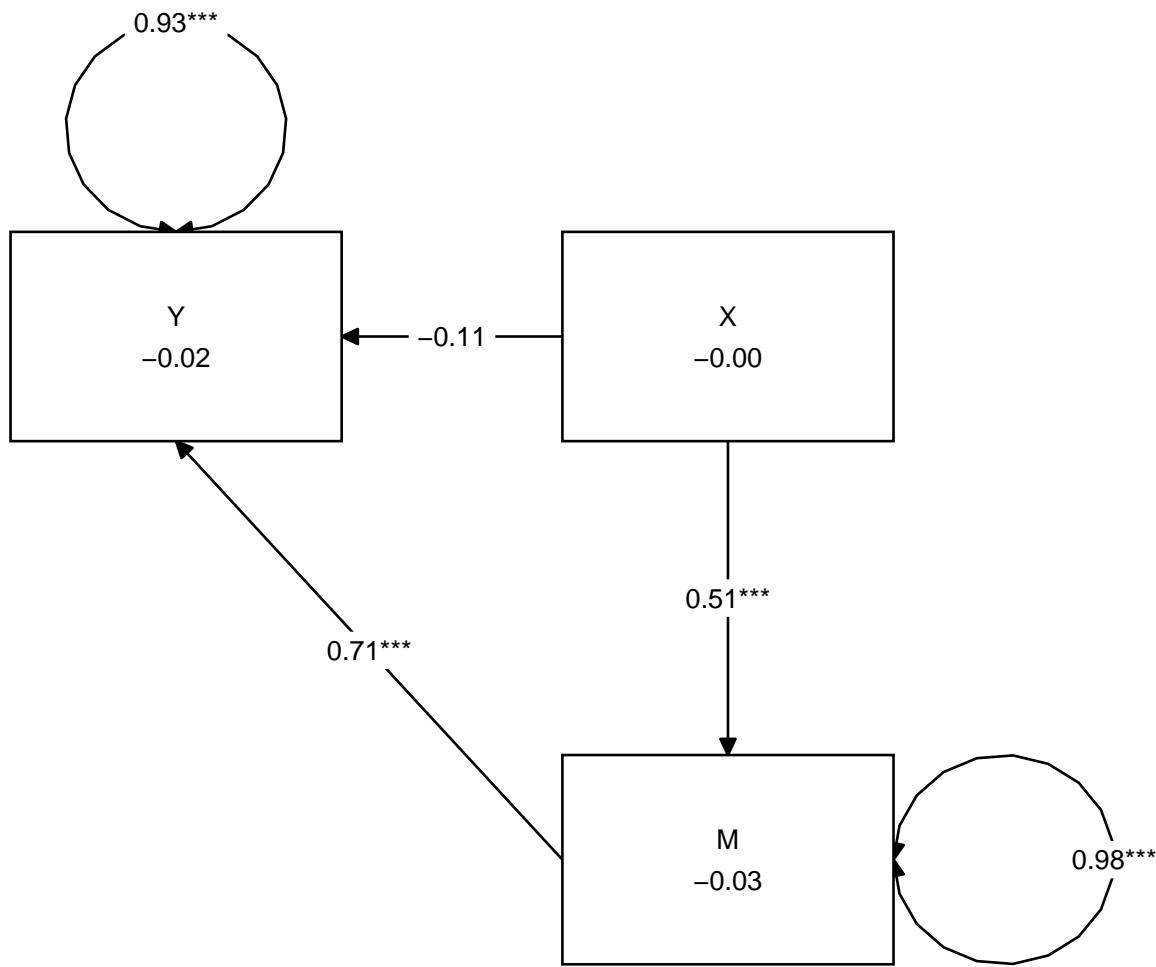
```
## Loading required package: OpenMx
```

```
##  
## Attaching package: 'OpenMx'
```

```
## The following object is masked from 'package:psych':  
##  
##     tr
```

```
## Registered S3 method overwritten by 'tidySEM':  
##   method      from  
##   predict.MxModel  OpenMx
```

```
tidySEM::graph_sem(model = fit)
```



Hayes has great examples of APA style tables that have become the standard way to communicate results. I haven't yet found a package that will turn this output into a journal-ready table, however with a little tinkering, we can approximate one of the standard tables. This code lets us understand the label names and how they are mapped

```
tidySEM::get_layout(fit)
```

```
##      [,1] [,2] [,3]
## [1,] "Y"   "M"   "X"
## attr(),"class")
## [1] "layout_matrix" "matrix"       "array"
```

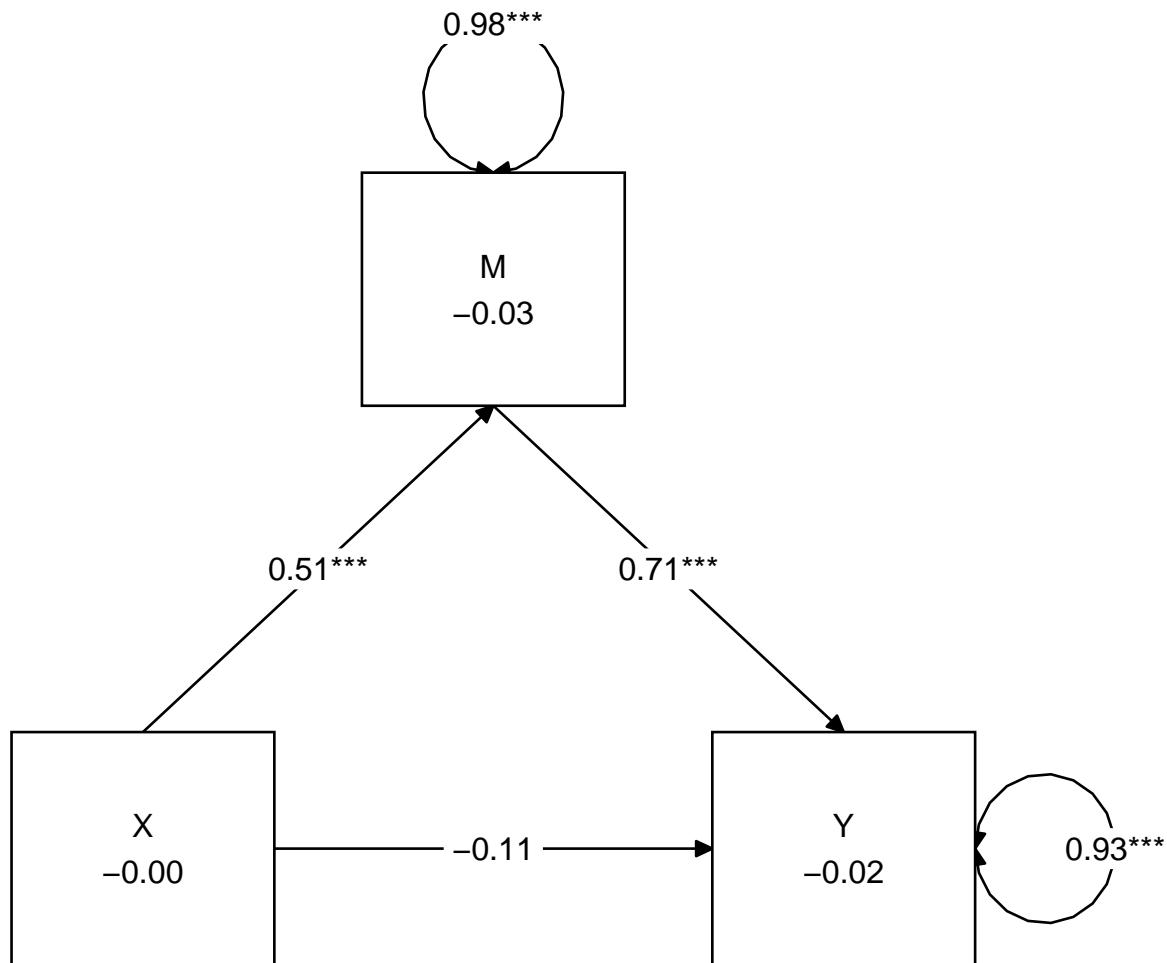
We can write code to remap them

```
med_map <- tidySEM::get_layout("", "M", "", "X", "", "Y", rows = 2)
med_map
```

```
##      [,1] [,2] [,3]
## [1,] ""   "M"  ""
## [2,] "X"  ""   "Y"
## attr(),"class")
## [1] "layout_matrix" "matrix"           "array"
```

We run again with our map and BOOM! Still needs tinkering for gorgeous, but hey!

```
tidySEM::graph_sem(fit, layout = med_map, rect_width = 1.5, rect_height = 1.25,
spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



To assist in table preparation, it is possible to export the results to a .csv file that can be manipulated in Excel, Microsoft Word, or other program to prepare an APA style table.

```
write.csv(FD_ParamEsts, file = "FakeDataOUT.csv")
```

Check with your discipline's journals to see how results of mediations are reported. Here's a version that I like.

Table 1

Model Coefficients Assessing M as a Mediator Between X and Y

Mediator (M)					Dependent Variable (Y)				
Antecedent constant	path i_M	B 0.031	SE 0.097	p 0.749	path i_Y	B -0.022	SE 0.097	p 0.822	
Independent (X)	a	0.513	0.097	< 0.001	c'	-0.107	0.112	0.340	
Mediator (M)					b	0.708	0.085	< 0.0	
		$R^2 = 18\%$				$R^2 = 37\%$			

Note. The value of the indirect effect was $B = 0.363$, $SE = 0.084$, $p < 0.001$, $95CI(0.224, 0.557)$

5.4.5 Results

A simple mediation model examined the degree to which M mediated the relation of X on Y. Using the *lavaan* package (v 0.6-16) in R, coefficients for each path, the indirect effect, and total effects were calculated. These values are presented in Table 1 and illustrated in Figure 1. Results suggested that 18% of the variance in M and 37% of the variance in Y were accounted for in the model. The indirect effect ($B = 0.363$, $SE = 0.084$, $p < 0.001$) was statistically significant; the direct effect ($B = -0.107$, $SE = 0.112$, $p = 0.340$) was not. Comparing the nonsignificant direct effect to the statistically significant total effect ($B = 0.257$, $SE = 0.120$, $p = 0.033$) is consistent with the notion that the effect of X on Y is explained through M.

5.5 Research Vignette

The research vignette comes from the Kim, Kendall, and Cheon's [2017], "Racial Microaggressions, Cultural Mistrust, and Mental Health Outcomes Among Asian American College Students." Participants were 156 Asian American undergraduate students in the Pacific Northwest. The researchers posited the a priori hypothesis that cultural mistrust would mediate the relationship between racial microaggressions and two sets of outcomes: mental health (e.g., depression, anxiety, well-being) and help-seeking.

Variables used in the study included:

- **REMS:** Racial and Ethnic Microaggressions Scale (Nadal, 2011). The scale includes 45 items on a 2-point scale where 0 indicates no experience of a microaggressive event and 1 indicates it was experienced at least once within the past six months. Higher scores indicate more experience of microaggressions.

- **CMI:** Cultural Mistrust Inventory (Terrell & Terrell, 1981). This scale was adapted to assess cultural mistrust harbored among Asian Americans toward individuals from the mainstream U.S. culture (e.g., Whites). The CMI includes 47 items on a 7-point scale where higher scores indicate a higher degree of cultural mistrust.
- **ANX, DEP, PWB:** Subscales of the Mental Health Inventory (Veit & Ware, 1983) that assess the mental health outcomes of anxiety (9 items), depression (4 items), and psychological well-being (14 items). Higher scores (on a 6 point scale) indicate stronger endorsement of the mental health outcome being assessed.
- **HlpSkg:** The Attitudes Toward Seeking Professional Psychological Help – Short Form (Fischer & Farina, 1995) includes 10 items on a 4-point scale (0 = disagree, 3 = agree) where higher scores indicate more favorable attitudes toward help seeking.

5.5.1 Data Simulation

We used the *lavaan::simulateData* function for the simulation. If you have taken psychometrics, you may recognize the code as one that creates latent variables from item-level data. In trying to be as authentic as possible, we retrieved factor loadings from psychometrically oriented articles that evaluated the measures [Nadal, 2011, Veit and Ware, 1983]. For all others we specified a factor loading of 0.80. We then approximated the *measurement model* by specifying the correlations between the latent variable. We sourced these from the correlation matrix from the research vignette [Kim et al., 2017]. The process created data with multiple decimals and values that exceeded the boundaries of the variables. For example, in all scales there were negative values. Therefore, the final element of the simulation was a linear transformation that rescaled the variables back to the range described in the journal article and rounding the values to integer (i.e., with no decimal places).

```
# Entering the intercorrelations, means, and standard deviations from
# the journal article
Kim_generating_model <- "
  ##measurement model
  REMS =~ .82*Inf32 + .75*Inf38 + .74*Inf21 + .72*Inf17 + .69*Inf9 + .61*Inf36 + .51*In
  CMI =~ .8*cmi1 + .8*cmi2 + .8*cmi3 + .8*cmi4 + .8*cmi5 + .8*cmi6 + .8*cmi7 + .8*cmi8 +
  ANX =~ .80*Anx1 + .80*Anx2 + .77*Anx3 + .74*Anx4 + .74*Anx5 + .69*Anx6 + .69*Anx7 + .
  DEP =~ .74*Dep1 + .83*Dep2 + .82*Dep3 + .74*Dep4
  PWB =~ .83*pwb1 + .72*pwb2 + .67*pwb3 + .79*pwb4 + .77*pwb5 + .75*pwb6 + .74*pwb7 + .7
  HlpSkg =~ .8*hlpstk1 + .8*hlpstk2 + .8*hlpstk3 + .8*hlpstk4 + .8*hlpstk5 + .8*hlpstk6

  # Means
  REMS ~ 0.34*1
  CMI ~ 3*1
  ANX ~ 2.98*1
  DEP ~ 2.36*1
  PWB ~ 3.5*1
  HlpSkg ~ 1.64*1
```

```

# Correlations (ha!)
REMS ~ 0.58*CMI
REMS ~ 0.26*ANX
REMS ~ 0.34*DEP
REMS ~ -0.25*PWB
REMS ~ -0.02*HlpSkg
CMI ~ 0.12*ANX
CMI ~ 0.19*DEP
CMI ~ -0.28*PWB
CMI ~ 0*HlpSkg
ANX ~ 0.66*DEP
ANX ~ -0.55*PWB
ANX ~ 0.07*HlpSkg
DEP ~ -0.66*PWB
DEP ~ 0.05*HlpSkg
PWB ~ 0.08*HlpSkg
""

set.seed(230916)
dfKim <- lavaan::simulateData(model = Kim_generating_model, model.type = "sem",
  meanstructure = T, sample.nobs = 156, standardized = FALSE)
library(tidyverse)
# Kim_df_latent <- Kim_df_latent %>% round(0) %>% abs()

dfKim$Inf32 <- scales::rescale(dfKim$Inf32, c(0, 1))
dfKim$Inf38 <- scales::rescale(dfKim$Inf38, c(0, 1))
dfKim$Inf21 <- scales::rescale(dfKim$Inf21, c(0, 1))
dfKim$Inf17 <- scales::rescale(dfKim$Inf17, c(0, 1))
dfKim$Inf9 <- scales::rescale(dfKim$Inf9, c(0, 1))
dfKim$Inf36 <- scales::rescale(dfKim$Inf36, c(0, 1))
dfKim$Inf5 <- scales::rescale(dfKim$Inf5, c(0, 1))
dfKim$Inf22 <- scales::rescale(dfKim$Inf22, c(0, 1))
dfKim$SClass6 <- scales::rescale(dfKim$SClass6, c(0, 1))
dfKim$SClass31 <- scales::rescale(dfKim$SClass31, c(0, 1))
dfKim$SClass8 <- scales::rescale(dfKim$SClass8, c(0, 1))
dfKim$SClass40 <- scales::rescale(dfKim$SClass40, c(0, 1))
dfKim$SClass2 <- scales::rescale(dfKim$SClass2, c(0, 1))
dfKim$SClass34 <- scales::rescale(dfKim$SClass34, c(0, 1))
dfKim$SClass11 <- scales::rescale(dfKim$SClass11, c(0, 1))
dfKim$mInv27 <- scales::rescale(dfKim$mInv27, c(0, 1))
dfKim$mInv30 <- scales::rescale(dfKim$mInv30, c(0, 1))
dfKim$mInv39 <- scales::rescale(dfKim$mInv39, c(0, 1))
dfKim$mInv7 <- scales::rescale(dfKim$mInv7, c(0, 1))
dfKim$mInv26 <- scales::rescale(dfKim$mInv26, c(0, 1))
dfKim$mInv33 <- scales::rescale(dfKim$mInv33, c(0, 1))
dfKim$mInv4 <- scales::rescale(dfKim$mInv4, c(0, 1))
dfKim$mInv14 <- scales::rescale(dfKim$mInv14, c(0, 1))

```

```
dfKim$mInv10 <- scales::rescale(dfKim$mInv10, c(0, 1))
dfKim$Exot3 <- scales::rescale(dfKim$Exot3, c(0, 1))
dfKim$Exot29 <- scales::rescale(dfKim$Exot29, c(0, 1))
dfKim$Exot45 <- scales::rescale(dfKim$Exot45, c(0, 1))
dfKim$Exot35 <- scales::rescale(dfKim$Exot35, c(0, 1))
dfKim$Exot42 <- scales::rescale(dfKim$Exot42, c(0, 1))
dfKim$Exot23 <- scales::rescale(dfKim$Exot23, c(0, 1))
dfKim$Exot13 <- scales::rescale(dfKim$Exot13, c(0, 1))
dfKim$Exot20 <- scales::rescale(dfKim$Exot20, c(0, 1))
dfKim$Exot43 <- scales::rescale(dfKim$Exot43, c(0, 1))
dfKim$mEnv37 <- scales::rescale(dfKim$mEnv37, c(0, 1))
dfKim$mEnv24 <- scales::rescale(dfKim$mEnv24, c(0, 1))
dfKim$mEnv19 <- scales::rescale(dfKim$mEnv19, c(0, 1))
dfKim$mEnv28 <- scales::rescale(dfKim$mEnv28, c(0, 1))
dfKim$mEnv18 <- scales::rescale(dfKim$mEnv18, c(0, 1))
dfKim$mEnv41 <- scales::rescale(dfKim$mEnv41, c(0, 1))
dfKim$mEnv12 <- scales::rescale(dfKim$mEnv12, c(0, 1))
dfKim$mWork25 <- scales::rescale(dfKim$mWork25, c(0, 1))
dfKim$mWork15 <- scales::rescale(dfKim$mWork15, c(0, 1))
dfKim$mWork1 <- scales::rescale(dfKim$mWork1, c(0, 1))
dfKim$mWork16 <- scales::rescale(dfKim$mWork16, c(0, 1))
dfKim$mWork44 <- scales::rescale(dfKim$mWork44, c(0, 1))

dfKim$cmi1 <- scales::rescale(dfKim$cmi1, c(1, 7))
dfKim$cmi2 <- scales::rescale(dfKim$cmi2, c(1, 7))
dfKim$cmi3 <- scales::rescale(dfKim$cmi3, c(1, 7))
dfKim$cmi4 <- scales::rescale(dfKim$cmi4, c(1, 7))
dfKim$cmi5 <- scales::rescale(dfKim$cmi5, c(1, 7))
dfKim$cmi6 <- scales::rescale(dfKim$cmi6, c(1, 7))
dfKim$cmi7 <- scales::rescale(dfKim$cmi7, c(1, 7))
dfKim$cmi8 <- scales::rescale(dfKim$cmi8, c(1, 7))
dfKim$cmi9 <- scales::rescale(dfKim$cmi9, c(1, 7))
dfKim$cmi10 <- scales::rescale(dfKim$cmi10, c(1, 7))
dfKim$cmi11 <- scales::rescale(dfKim$cmi11, c(1, 7))
dfKim$cmi12 <- scales::rescale(dfKim$cmi12, c(1, 7))
dfKim$cmi13 <- scales::rescale(dfKim$cmi13, c(1, 7))
dfKim$cmi14 <- scales::rescale(dfKim$cmi14, c(1, 7))
dfKim$cmi15 <- scales::rescale(dfKim$cmi15, c(1, 7))
dfKim$cmi16 <- scales::rescale(dfKim$cmi16, c(1, 7))
dfKim$cmi17 <- scales::rescale(dfKim$cmi17, c(1, 7))
dfKim$cmi18 <- scales::rescale(dfKim$cmi18, c(1, 7))
dfKim$cmi19 <- scales::rescale(dfKim$cmi19, c(1, 7))
dfKim$cmi20 <- scales::rescale(dfKim$cmi20, c(1, 7))
dfKim$cmi21 <- scales::rescale(dfKim$cmi21, c(1, 7))
dfKim$cmi22 <- scales::rescale(dfKim$cmi22, c(1, 7))
dfKim$cmi23 <- scales::rescale(dfKim$cmi23, c(1, 7))
dfKim$cmi24 <- scales::rescale(dfKim$cmi24, c(1, 7))
```

```
dfKim$cmi25 <- scales::rescale(dfKim$cmi25, c(1, 7))
dfKim$cmi26 <- scales::rescale(dfKim$cmi26, c(1, 7))
dfKim$cmi27 <- scales::rescale(dfKim$cmi27, c(1, 7))
dfKim$cmi28 <- scales::rescale(dfKim$cmi28, c(1, 7))
dfKim$cmi29 <- scales::rescale(dfKim$cmi29, c(1, 7))
dfKim$cmi30 <- scales::rescale(dfKim$cmi30, c(1, 7))
dfKim$cmi31 <- scales::rescale(dfKim$cmi31, c(1, 7))
dfKim$cmi32 <- scales::rescale(dfKim$cmi32, c(1, 7))
dfKim$cmi33 <- scales::rescale(dfKim$cmi33, c(1, 7))
dfKim$cmi34 <- scales::rescale(dfKim$cmi34, c(1, 7))
dfKim$cmi35 <- scales::rescale(dfKim$cmi35, c(1, 7))
dfKim$cmi36 <- scales::rescale(dfKim$cmi36, c(1, 7))
dfKim$cmi37 <- scales::rescale(dfKim$cmi37, c(1, 7))
dfKim$cmi38 <- scales::rescale(dfKim$cmi38, c(1, 7))
dfKim$cmi39 <- scales::rescale(dfKim$cmi39, c(1, 7))
dfKim$cmi40 <- scales::rescale(dfKim$cmi40, c(1, 7))
dfKim$cmi41 <- scales::rescale(dfKim$cmi41, c(1, 7))
dfKim$cmi42 <- scales::rescale(dfKim$cmi42, c(1, 7))
dfKim$cmi43 <- scales::rescale(dfKim$cmi43, c(1, 7))
dfKim$cmi44 <- scales::rescale(dfKim$cmi44, c(1, 7))
dfKim$cmi45 <- scales::rescale(dfKim$cmi45, c(1, 7))
dfKim$cmi46 <- scales::rescale(dfKim$cmi46, c(1, 7))
dfKim$cmi47 <- scales::rescale(dfKim$cmi47, c(1, 7))

dfKim$Anx1 <- scales::rescale(dfKim$Anx1, c(1, 5))
dfKim$Anx2 <- scales::rescale(dfKim$Anx2, c(1, 5))
dfKim$Anx3 <- scales::rescale(dfKim$Anx3, c(1, 5))
dfKim$Anx4 <- scales::rescale(dfKim$Anx4, c(1, 5))
dfKim$Anx5 <- scales::rescale(dfKim$Anx5, c(1, 5))
dfKim$Anx6 <- scales::rescale(dfKim$Anx6, c(1, 5))
dfKim$Anx7 <- scales::rescale(dfKim$Anx7, c(1, 5))
dfKim$Anx8 <- scales::rescale(dfKim$Anx8, c(1, 5))
dfKim$Anx9 <- scales::rescale(dfKim$Anx9, c(1, 5))

dfKim$Dep1 <- scales::rescale(dfKim$Dep1, c(1, 5))
dfKim$Dep2 <- scales::rescale(dfKim$Dep2, c(1, 5))
dfKim$Dep3 <- scales::rescale(dfKim$Dep3, c(1, 5))
dfKim$Dep4 <- scales::rescale(dfKim$Dep4, c(1, 5))

dfKim$pwb1 <- scales::rescale(dfKim$pwb1, c(1, 5))
dfKim$pwb2 <- scales::rescale(dfKim$pwb2, c(1, 5))
dfKim$pwb3 <- scales::rescale(dfKim$pwb3, c(1, 5))
dfKim$pwb4 <- scales::rescale(dfKim$pwb4, c(1, 5))
dfKim$pwb5 <- scales::rescale(dfKim$pwb5, c(1, 5))
dfKim$pwb6 <- scales::rescale(dfKim$pwb6, c(1, 5))
dfKim$pwb7 <- scales::rescale(dfKim$pwb7, c(1, 5))
dfKim$pwb8 <- scales::rescale(dfKim$pwb8, c(1, 5))
```

```

dfKim$pwb9 <- scales::rescale(dfKim$pwb9, c(1, 5))
dfKim$pwb10 <- scales::rescale(dfKim$pwb10, c(1, 5))
dfKim$pwb11 <- scales::rescale(dfKim$pwb11, c(1, 5))

dfKim$hlpstk1 <- scales::rescale(dfKim$hlpstk1, c(0, 3))
dfKim$hlpstk2 <- scales::rescale(dfKim$hlpstk2, c(0, 3))
dfKim$hlpstk3 <- scales::rescale(dfKim$hlpstk3, c(0, 3))
dfKim$hlpstk4 <- scales::rescale(dfKim$hlpstk4, c(0, 3))
dfKim$hlpstk5 <- scales::rescale(dfKim$hlpstk5, c(0, 3))
dfKim$hlpstk6 <- scales::rescale(dfKim$hlpstk6, c(0, 3))
dfKim$hlpstk7 <- scales::rescale(dfKim$hlpstk7, c(0, 3))
dfKim$hlpstk8 <- scales::rescale(dfKim$hlpstk8, c(0, 3))
dfKim$hlpstk9 <- scales::rescale(dfKim$hlpstk9, c(0, 3))
dfKim$hlpstk10 <- scales::rescale(dfKim$hlpstk10, c(0, 3))

# psych::describe(dfKim)

library(tidyverse)
dfKim <- dfKim %>%
  round(0)

# I tested the rescaling the correlation between original and
# rescaled variables is 1.0 Kim_df_latent$INF32 <-
# scales::rescale(Kim_df_latent$Inf32, c(0, 1))
# cor.test(Kim_df_latent$Inf32, Kim_df_latent$INF32,
# method='pearson')

# Checking our work against the original correlation matrix
# round(cor(Kim_df),3)

```

The script below allows you to store the simulated data as a file on your computer. This is optional – the entire lesson can be worked with the simulated data.

If you prefer the .rds format, use this script (remove the hashtags). The .rds format has the advantage of preserving any formatting of variables. A disadvantage is that you cannot open these files outside of the R environment.

Script to save the data to your computer as an .rds file.

```
#saveRDS(dfKim, 'dfKim.rds')
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfKim<- readRDS('dfKim.rds')
```

If you prefer the .csv format (think “Excel lite”) use this script (remove the hashtags). An advantage of the .csv format is that you can open the data outside of the R environment. A disadvantage is that it may not retain any formatting of variables

Script to save the data to your computer as a .csv file.

```
#write.table(dfKim, file = 'dfKim.csv', sep = ',', col.names=TRUE, row.names=FALSE)
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfKim<- read.csv ('dfKim.csv', header = TRUE)
```

5.5.2 Scrubbing, Scoring, and Data Diagnostics

Because the focus of this lesson is on simple mediation, we have used simulated data. If this were real, raw, data, it would be important to [scrub](#), [score](#), and conduct [data diagnostics](#) to evaluate the suitability of the data for the proposed analyses.

Because we are working with item level data we first need to score the scales used in the researcher's model/. Because we are using simulated data and the authors already reverse coded any items requiring recoding, we can omit that step.

As described in the [Scoring](#) chapter, we can calculate mean scores of these variables by first creating concatenated lists of variable names. Next we apply the `sjstats::mean_n` function to obtain mean scores when a given percentage (we'll specify 80%) of variables are non-missing. We simulated a set of data that does not have missingness, none-the-less, this specification is useful in real-world settings.

```
PWB_vars <- c("pwb1", "pwb2", "pwb3", "pwb4", "pwb5", "pwb6", "pwb7", "pwb8",
  "pwb9", "pwb10")
ANX_vars <- c("Anx1", "Anx2", "Anx3", "Anx4", "Anx5", "Anx6", "Anx7", "Anx8",
  "Anx9")
CMI_vars <- c("cmi1", "cmi2", "cmi3", "cmi4", "cmi5", "cmi6", "cmi7", "cmi8",
  "cmi9", "cmi10", "cmi11", "cmi12", "cmi13", "cmi14", "cmi15", "cmi16",
  "cmi17", "cmi18", "cmi19", "cmi20", "cmi21", "cmi22", "cmi23", "cmi24",
  "cmi25", "cmi26", "cmi27", "cmi28", "cmi29", "cmi30", "cmi31", "cmi32",
  "cmi33", "cmi34", "cmi35", "cmi36", "cmi37", "cmi38", "cmi39", "cmi40",
  "cmi41", "cmi42", "cmi43", "cmi44", "cmi45", "cmi46", "cmi47")
REMS_vars <- c("Inf32", "Inf38", "Inf21", "Inf17", "Inf9", "Inf36", "Inf5",
  "Inf22", "SClass6", "SClass31", "SClass8", "SClass40", "SClass2", "SClass34",
  "SClass11", "mInv27", "mInv30", "mInv39", "mInv7", "mInv26", "mInv33",
  "mInv4", "mInv14", "mInv10", "Exot3", "Exot29", "Exot45", "Exot35",
  "Exot42", "Exot23", "Exot13", "Exot20", "Exot43", "mEnv37", "mEnv24",
  "mEnv19", "mEnv28", "mEnv18", "mEnv41", "mEnv12", "mWork25", "mWork15",
  "mWork1", "mWork16", "mWork44")

dfKim$PWB <- sjstats::mean_n(dfKim[, PWB_vars], 0.8)
dfKim$ANX <- sjstats::mean_n(dfKim[, ANX_vars], 0.8)
dfKim$CMI <- sjstats::mean_n(dfKim[, CMI_vars], 0.8)
dfKim$REMS <- sjstats::mean_n(dfKim[, REMS_vars], 0.8)
```

```
# If the scoring code above does not work for you, try the format
# below which involves inserting periods in front of the variable
# list. One example is provided. dfLewis$GRMS <-
# sjstats::mean_n(dfLewis[, ..GRMS_vars], 0.80)
```

Now that we have scored our data, let's trim the variables to just those we need.

```
dfModel <- dplyr::select(dfKim, PWB, ANX, CMI, REMS)
```

Let's check a table of means, standards, and correlations to see if they align with the published article.

```
DescriptivesTable <- apaTables::apa.cor.table(dfModel, table.number = 1,
                                             show.sig.stars = TRUE, landscape = TRUE, filename = NA)
print(DescriptivesTable)
```

```
##
##
## Table 1
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##      Variable M     SD    1          2          3
##      1. PWB   3.09 0.45
##
##      2. ANX   2.82 0.57 -.50**    [-.61, -.37]
##
##      3. CMI   3.94 0.77 -.49**    .43**    [-.60, -.36] [.30, .55]
##
##      4. REMS  0.51 0.29 -.47**    .58**    .58**    [-.59, -.34] [.47, .68] [.47, .68]
##
##
## 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.
##
```

While the patterns are similar, we can see some differences. This means that our simulated results are likely to have some difference than the results in the published article.

Comparison	Article	Simulation
PWB mean	3.50	3.09
ANX mean	2.98	2.82
CMI mean	3.00	3.94
REM mean	.34	.51
PWB ~ ANX	-0.55***	-0.50**
PWB ~ CMI	-0.28***	-0.49**
PWB ~ REMS	-0.25**	-0.47**
ANX ~ CMI	0.12	0.43**
ANX ~ REMS	0.26**	0.58**
CMI ~ REMS	0.59***	0.58**

There are a number of reasons I love the Kim et al. [2017] manuscript. One is that their approach was openly one that tested *alternate models*. Byrne [2016b] credits Joreskog [Joreskog, 1993] with classifying the researcher's model testing approach in three ways. If a researcher uses a *strictly confirmatory* approach, they only test the proposed model and then accept or reject it without further alteration. While this is the tradition of null hypothesis significance testing (NHST), it contributes to the “file drawer problem” of unpublished, non-significant, findings. Additionally, the data are them discarded – potentially losing valuable resource. The *alternative models* approach is to propose a handful of competing models before beginning the analysis and then evaluating to see if one model is superior to the other. The third option is *model generating*. In this case the researcher begins with a theoretically proposed model. In the presence of poor fit, the researcher seeks to identify the source of misfit – respecifying it to best represent the sample data. The researcher must use caution to produce a model that fits well and is meaningful.

Several of the Kim et al. [2017] models were non-significant. To demonstrate a model that is statistically significant, I will test the hypothesis that racial microaggressions (REMS, the X variable) influence depression (DEP, the Y variable) through cultural mistrust (CMI, the M variable).

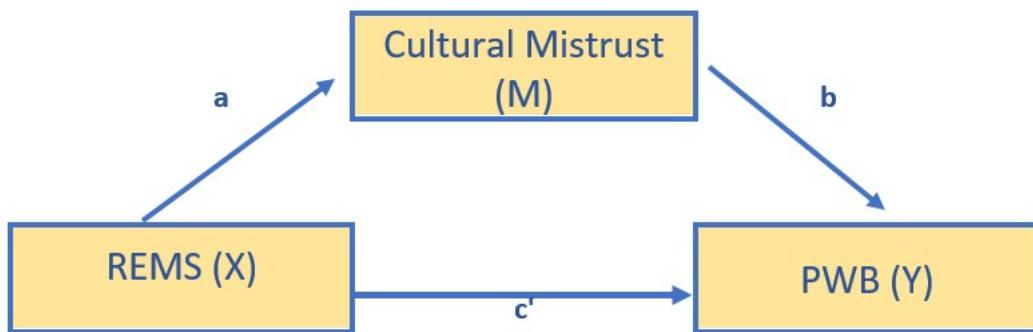


Figure 5.7: Image of the simple mediation model from Kim et al.

5.5.3 Specify the Model in *lavaan*

I am a big fan of “copying the model.” That is, I find *code that works* as a starting point. In specifying my model I used the simple mediation template from above. I

- replaced the Y, X, and M with variables names
- replacing the name of the df
- updated the object names (so I could use them in the same .rmd file)

```
modKim <- "
    PWB ~ b*CMI + c_p*REMS
    CMI ~a*REMS

    indirect := a*b
    direct   := c_p
    total_c  := c_p + (a*b)
    "
```

```
set.seed(230916) #necessary for reproducible results since lavaan introduces randomness in the
Kim_fit <- lavaan::sem(modKim, data = dfModel, se = "bootstrap", missing = "fiml")
```

```
Kim_summary <- summary(Kim_fit, standardized = T, rsq = T, fit = TRUE,
                        ci = TRUE)
Kim_ParamEsts <- parameterEstimates(Kim_fit, boot.ci.type = "bca.simple",
                                      standardized = TRUE)
Kim_summary
```

```
## lavaan 0.6.16 ended normally after 1 iteration
##
##             Estimator               ML
## Optimization method           NLMINB
## Number of model parameters          7
##
## Number of observations        156
## Number of missing patterns         1
##
## Model Test User Model:
##
##             Test statistic      0.000
## Degrees of freedom                  0
##
## Model Test Baseline Model:
##
##             Test statistic     119.320
## Degrees of freedom                   3
## P-value                           0.000
##
## User Model versus Baseline Model:
##
##             Comparative Fit Index (CFI)      1.000
## Tucker-Lewis Index (TLI)            1.000
##
```

```

## Robust Comparative Fit Index (CFI)           1.000
## Robust Tucker-Lewis Index (TLI)            1.000
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)          -218.515
## Loglikelihood unrestricted model (H1) -218.515
##
## Akaike (AIC)                          451.030
## Bayesian (BIC)                         472.379
## Sample-size adjusted Bayesian (SABIC)  450.222
##
## Root Mean Square Error of Approximation:
##
## RMSEA                               0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.000
## P-value H_0: RMSEA <= 0.050          NA
## P-value H_0: RMSEA >= 0.080          NA
##
## Robust RMSEA                         0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.000
## P-value H_0: Robust RMSEA <= 0.050   NA
## P-value H_0: Robust RMSEA >= 0.080   NA
##
## Standardized Root Mean Square Residual:
##
## SRMR                                0.000
##
## Parameter Estimates:
##
## Standard errors                      Bootstrap
## Number of requested bootstrap draws  1000
## Number of successful bootstrap draws 1000
##
## Regressions:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## PWB ~
##   CMI      (b)   -0.189   0.052  -3.640   0.000  -0.291  -0.088
##   REMS    (c_p)  -0.453   0.139  -3.260   0.001  -0.740  -0.194
## CMI ~
##   REMS      (a)   1.576   0.177   8.920   0.000   1.199   1.938
##   Std.lv  Std.all
##   -0.189  -0.323
##   -0.453  -0.286
##

```

```

##      1.576    0.584
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .PWB          4.066   0.177  22.934  0.000   3.733   4.416
## .CMI          3.141   0.104  30.276  0.000   2.933   3.364
## Std.lv  Std.all
##    4.066    9.004
##    3.141   4.072
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .PWB          0.144   0.017   8.248  0.000   0.109   0.176
## .CMI          0.392   0.041   9.557  0.000   0.312   0.473
## Std.lv  Std.all
##    0.144    0.706
##    0.392   0.659
##
## R-Square:
##              Estimate
## PWB          0.294
## CMI          0.341
##
## Defined Parameters:
##              Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## indirect       -0.298   0.092  -3.251  0.001  -0.488  -0.138
## direct         -0.453   0.139  -3.259  0.001  -0.740  -0.194
## total_c        -0.750   0.112  -6.703  0.000  -0.961  -0.524
## Std.lv  Std.all
## -0.298   -0.188
## -0.453   -0.286
## -0.750   -0.475

```

Kim_ParamEsts

	lhs	op	rhs	label	est	se	z	pvalue	ci.lower	ci.upper
## 1	PWB	~	CMI	b	-0.189	0.052	-3.640	0.000	-0.284	-0.081
## 2	PWB	~	REMS	c_p	-0.453	0.139	-3.260	0.001	-0.754	-0.207
## 3	CMI	~	REMS	a	1.576	0.177	8.920	0.000	1.196	1.937
## 4	PWB	~~	PWB		0.144	0.017	8.248	0.000	0.115	0.190
## 5	CMI	~~	CMI		0.392	0.041	9.557	0.000	0.320	0.487
## 6	REMS	~~	REMS		0.082	0.000	NA	NA	0.082	0.082
## 7	PWB	~1			4.066	0.177	22.934	0.000	3.722	4.389
## 8	CMI	~1			3.141	0.104	30.276	0.000	2.941	3.367
## 9	REMS	~1			0.507	0.000	NA	NA	0.507	0.507
## 10	indirect	:=	a*b	indirect	-0.298	0.092	-3.251	0.001	-0.485	-0.131
## 11	direct	:=	c_p	direct	-0.453	0.139	-3.259	0.001	-0.754	-0.207

```

## 12 total_c := c_p+(a*b)  total_c -0.750 0.112 -6.703  0.000   -0.951   -0.518
##      std.lv std.all std.nox
## 1  -0.189  -0.323  -0.323
## 2  -0.453  -0.286  -1.002
## 3   1.576   0.584   2.043
## 4   0.144   0.706   0.706
## 5   0.392   0.659   0.659
## 6   0.082   1.000   0.082
## 7   4.066   9.004   9.004
## 8   3.141   4.072   4.072
## 9   0.507   1.775   0.507
## 10 -0.298  -0.188  -0.659
## 11 -0.453  -0.286  -1.002
## 12 -0.750  -0.475  -1.662

```

5.5.4 Interpret the Output

- Overall, our model accounted for 29% of the variance in the independent variable, well-being, and 34% of the variance in the mediator, cultural mistrust.
- a path: $B = 1.576, p < 0.001$
- b path: $B = -0.189, p < 0.001$
- the indirect effect is a product of the a and b paths: $B = -0.298, p = 0.001$.
- The bias-corrected bootstrapped confidence intervals can sometimes be more lenient than p values; it is important they don't cross zero ($95CI = -0.485, -0.131$). If 0.00 is included in the confidence interval, then we cannot be confident that the estimate is not, itself, zero.
- the direct effect (c' , c prime, or c_p) is the isolated effect of X on Y when including M. We hope this value is lower than the total effect because it would mean that including M shared some of the variance in predicting Y. In our case the value for c' is: $B = -0.453, p = 0.001$. Unfortunately, they are significant and they are not markedly different from the total effect ($B = -0.750, p < 0.001$).
- As a reminder, the total effect is is
- identical to the value of simply predicting Y on X (with no M it the model)
- the value of $a(b) + c_p$: $(1.576 * -0.189) + (-0.453) = -0.750; p < 0.001$

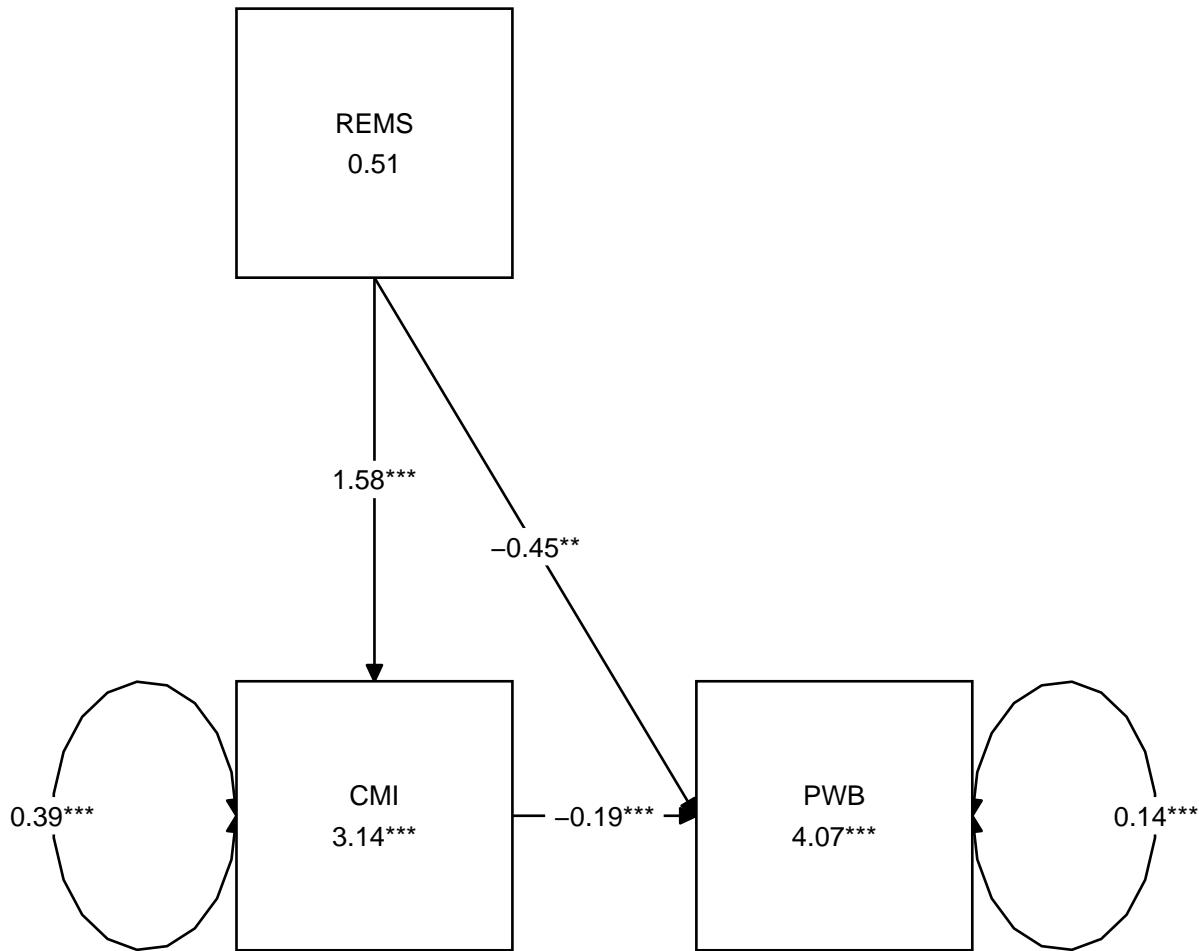
5.5.5 A Figure and a Table

I make it a practice to immediately plot what I did. Because the plotting packages use our models, this can be a helpful self-check of our work.

```

# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = Kim_fit)

```



Hayes has great examples of APA style tables that have become the standard way to communicate results. I haven't yet found a package that will turn this output into a journal-ready table, however with a little tinkering, we can approximate one of the standard tables. This code lets us understand the label names and how they are mapped

```
tidySEM::get_layout(Kim_fit)
```

```
##      [,1]  [,2]
## [1,] NA   "REMS"
## [2,] "PWB" "CMI"
## attr(,"class")
## [1] "layout_matrix" "matrix"        "array"
```

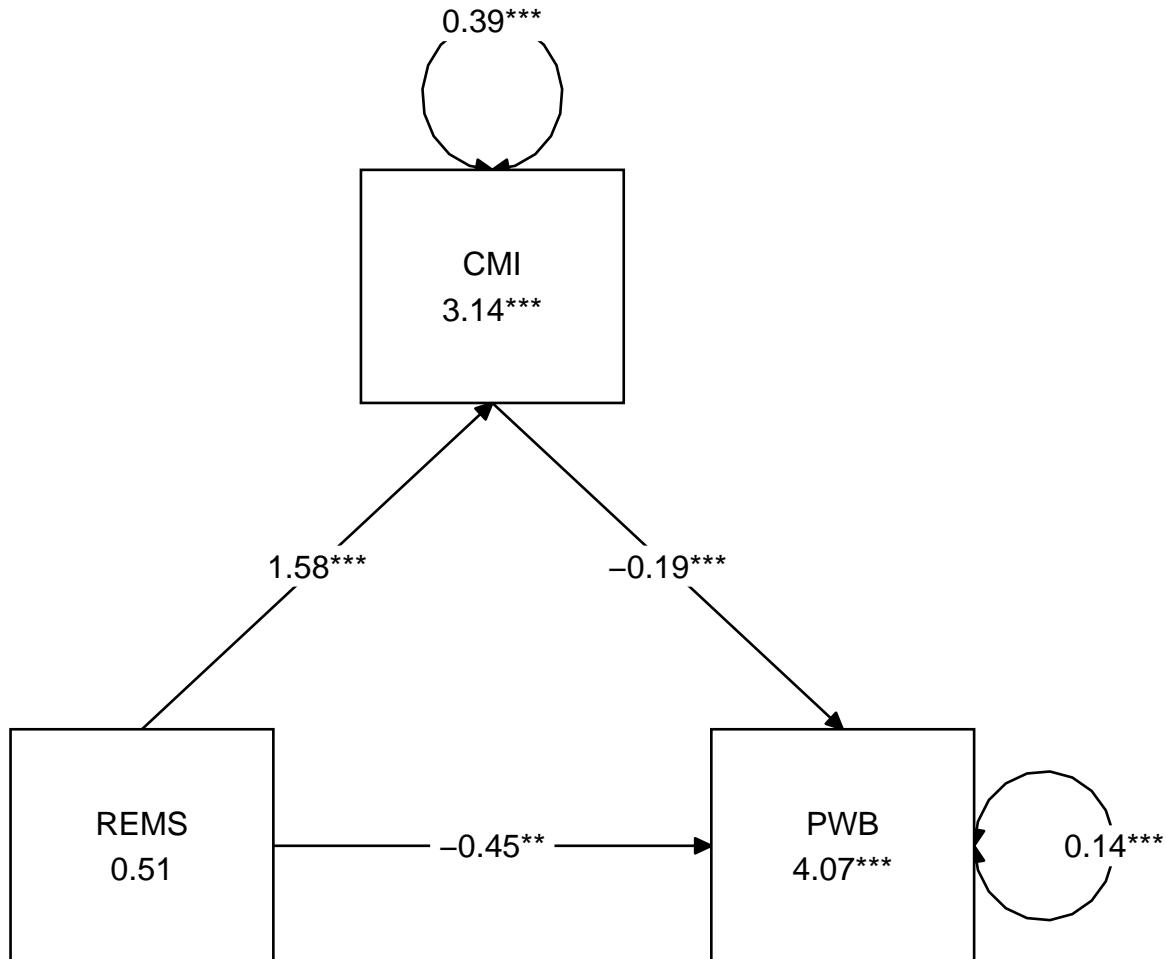
We can write code to remap them

```
med_map2 <- tidySEM::get_layout("", "CMI", "", "REMS", "", "PWB", rows = 2)
med_map2
```

```
##      [,1]  [,2]  [,3]
## [1,] ""    "CMI" ""
## [2,] "REMS" ""    "PWB"
## attr("class")
## [1] "layout_matrix" "matrix"      "array"
```

We run again with our map and BOOM! Still needs tinkering for gorgeous, but hey!

```
tidySEM::graph_sem(Kim_fit, layout = med_map2, rect_width = 1.5, rect_height = 1.25,
                    spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



We can use simple code from base R to write the results to a .csv file. This makes it easier to create a table for presenting the results.

```
write.csv(Kim_ParamEsts, file = "KimSimpleMed.csv")
```

Here's how I might organize the data.

Table 2

Model Coefficients Assessing Cultural Mistrust as a Mediator Between Racial Microaggressions and Well-Being

		Cultural Mistrust (M)				Well-Being (Y)			
Antecedent	path	B	SE	p	path	B	SE	p	
constant	i_M	3.1419	0.103	< 0.001	i_Y	4.066	0.177	< 0.001	
REMS (X)	a	1.576	0.184	< 0.001	c'	-0.453	0.139	0.001	
CMI (M)					b	-0.189	0.052	< 0.001	

$R^2 = 34\%$	$R^2 = 29\%$
--------------	--------------

Note. The value of the indirect effect was $B = -.298, SE = 0.092, p = 0.001, 95CI(-0.485, -0.131)$.

5.5.6 Results

A simple mediation model examined the degree to which cultural mistrust mediated the relation of racial microaggressions on well-being. Using the *lavaan* package (v 0.6-16) in R, coefficients for each path, the indirect effect, and total effects were calculated. These values are presented in Table 2 and illustrated in Figure 2. Results suggested that racial/ethnic microaggressions had statistically significant effects on both cultural mistrust ($B = 1.576, p < 0.001$) and well-being ($B = -0.453, p = 0.001$). Further, the indirect effect from our simulated data was statistically significant ($B = -.298, SE = 0.092, p = 0.001, 95CI[-0.485, -0.131]$). Results suggested that 34% of the variance in cultural mistrust and 29% of the variance in well-being were accounted for by the model.

5.6 Considering Covariates

Hayes Chapter 4 [2018] considers the role of covariates (e.g., other variables that could account for some of the variance in the model). When previous research (or commonsense, or detractors) suggest you should include them it is advisable to do so. If they are non-significant and/or your variables continue to explain variance over-and-above their contribution, then you have gained ground in ruling out plausible rival hypotheses and are adding to causal evidence.

Covariates are relatively easy to specify in *lavaan*. I tend to look at my figure and “see where the arrows go.” Those translate readily to the equations we write in the *lavaan* code.

Let’s say we are concerned that anxiety covaries with cultural mistrust and well-being. We’ll add it as a covariate to both.

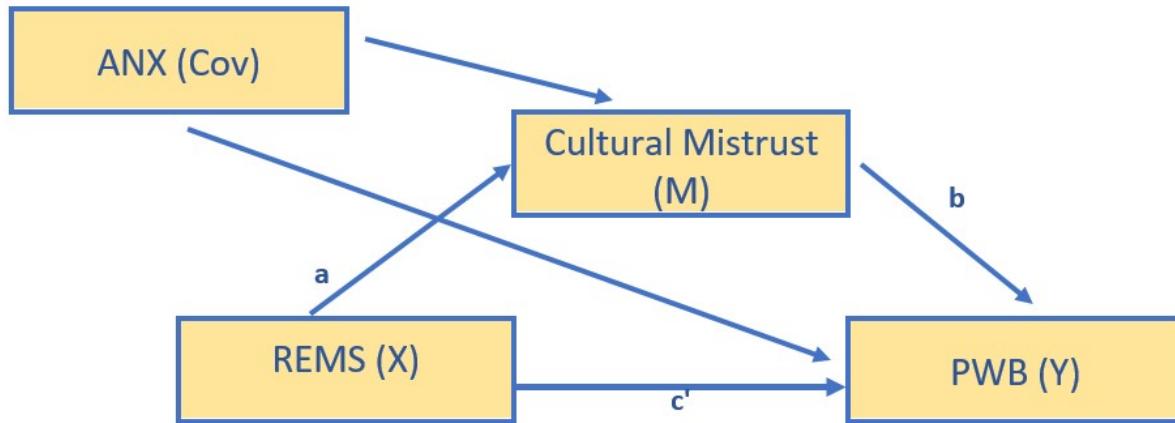


Figure 5.8: Image of the simple mediation model from Kim et al.

```

Kim_fit_covs <- "
  PWB ~ b*CMI + c_p*REMS
  CMI ~ a*REMS
  CMI ~ covM*ANX
  PWB ~ covY*ANX

  indirect := a*b
  direct := c_p
  total_c := c_p + (a*b)

  "
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
Kim_fit_covs <- lavaan::sem(Kim_fit_covs, data = dfKim, se = "bootstrap",
  missing = "fiml")
Kcov_sum <- lavaan::summary(Kim_fit_covs, standardized = T, rsq = T, fit = TRUE,
  ci = TRUE)
Kcov_ParEsts <- lavaan::parameterEstimates(Kim_fit_covs, boot.ci.type = "bca.simple",
  standardized = TRUE)
Kcov_sum

## lavaan 0.6.16 ended normally after 1 iteration
##                                         ML
##   Estimator                           NLMINB
##   Optimization method
##   Number of model parameters          9
  
```

```

## 
##   Number of observations           156
##   Number of missing patterns       1
##
## Model Test User Model:
##
##   Test statistic                  0.000
##   Degrees of freedom              0
##
## Model Test Baseline Model:
##
##   Test statistic                 136.009
##   Degrees of freedom              5
##   P-value                         0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)     1.000
##   Tucker-Lewis Index (TLI)        1.000
##
##   Robust Comparative Fit Index (CFI) 1.000
##   Robust Tucker-Lewis Index (TLI) 1.000
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)    -210.170
##   Loglikelihood unrestricted model (H1) -210.170
##
##   Akaike (AIC)                    438.341
##   Bayesian (BIC)                  465.789
##   Sample-size adjusted Bayesian (SABIC) 437.301
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                           0.000
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.000
##   P-value H_0: RMSEA <= 0.050          NA
##   P-value H_0: RMSEA >= 0.080          NA
##
##   Robust RMSEA                     0.000
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.000
##   P-value H_0: Robust RMSEA <= 0.050    NA
##   P-value H_0: Robust RMSEA >= 0.080    NA
##
## Standardized Root Mean Square Residual:
##

```

```

##      SRMR          0.000
##
## Parameter Estimates:
##
## Standard errors                                Bootstrap
## Number of requested bootstrap draws           1000
## Number of successful bootstrap draws          1000
##
## Regressions:
##              Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## PWB ~
##   CMI      (b) -0.163  0.051 -3.212  0.001 -0.263 -0.068
##   REMS     (c_p) -0.219  0.149 -1.474  0.140 -0.519  0.071
## CMI ~
##   REMS     (a)  1.349  0.191  7.045  0.000  0.956  1.707
##   ANX      (covM) 0.198  0.096  2.067  0.039  0.009  0.379
## PWB ~
##   ANX      (covY) -0.238  0.061 -3.910  0.000 -0.349 -0.109
## Std.lv  Std.all
##
## -0.163  -0.279
## -0.219  -0.139
##
##  1.349   0.500
##  0.198   0.145
##
## -0.238  -0.299
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .PWB        4.521   0.209 21.595  0.000  4.111  4.936
## .CMI        2.697   0.245 11.004  0.000  2.236  3.182
## Std.lv  Std.all
##  4.521   10.011
##  2.697   3.497
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .PWB        0.132   0.016  8.210  0.000  0.098  0.161
## .CMI        0.384   0.040  9.708  0.000  0.304  0.461
## Std.lv  Std.all
##  0.132   0.648
##  0.384   0.645
##
## R-Square:
##              Estimate
## PWB          0.352
## CMI          0.355

```

```
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## indirect                  -0.220   0.076 -2.888  0.004  -0.377  -0.082
## direct                   -0.219   0.149 -1.473  0.141  -0.519   0.071
## total_c                  -0.440   0.122 -3.612  0.000  -0.675  -0.209
## Std.lv Std.all
## -0.220  -0.139
## -0.219  -0.139
## -0.440  -0.278
```

Kcov_ParEsts

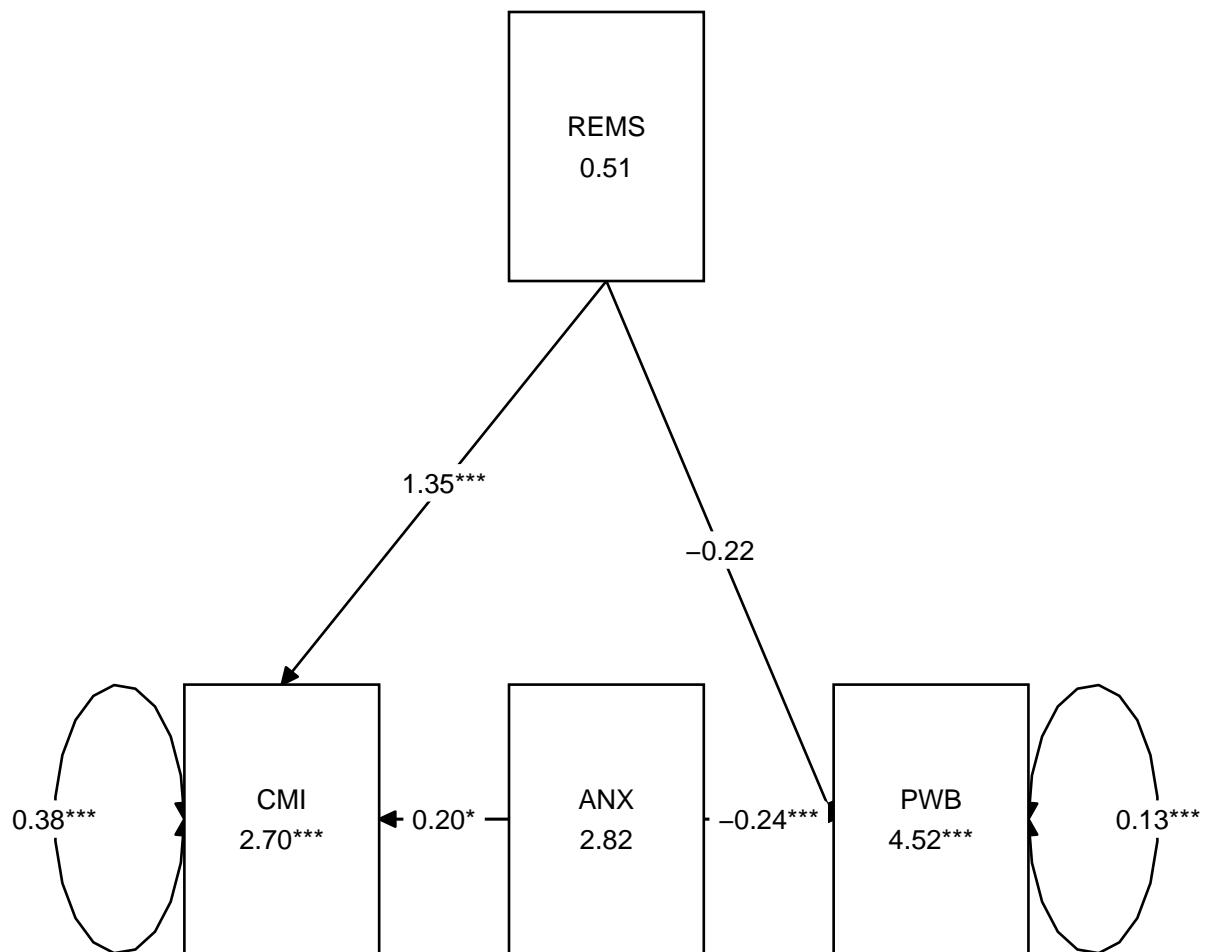
	lhs	op	rhs	label	est	se	z	pvalue	ci.lower	ci.upper
## 1	PWB	~	CMI	b	-0.163	0.051	-3.212	0.001	-0.257	-0.056
## 2	PWB	~	REMS	c_p	-0.219	0.149	-1.474	0.140	-0.528	0.062
## 3	CMI	~	REMS	a	1.349	0.191	7.045	0.000	0.910	1.673
## 4	CMI	~	ANX	covM	0.198	0.096	2.067	0.039	0.009	0.377
## 5	PWB	~	ANX	covY	-0.238	0.061	-3.910	0.000	-0.353	-0.110
## 6	PWB	~~	PWB		0.132	0.016	8.210	0.000	0.107	0.169
## 7	CMI	~~	CMI		0.384	0.040	9.708	0.000	0.320	0.479
## 8	REMS	~~	REMS		0.082	0.000	NA	NA	0.082	0.082
## 9	REMS	~~	ANX		0.094	0.000	NA	NA	0.094	0.094
## 10	ANX	~~	ANX		0.320	0.000	NA	NA	0.320	0.320
## 11	PWB	~1			4.521	0.209	21.595	0.000	4.114	4.941
## 12	CMI	~1			2.697	0.245	11.004	0.000	2.232	3.180
## 13	REMS	~1			0.507	0.000	NA	NA	0.507	0.507
## 14	ANX	~1			2.824	0.000	NA	NA	2.824	2.824
## 15	indirect :=		a*b	indirect	-0.220	0.076	-2.888	0.004	-0.385	-0.085
## 16	direct :=		c_p	direct	-0.219	0.149	-1.473	0.141	-0.528	0.062
## 17	total_c :=	c_p+(a*b)		total_c	-0.440	0.122	-3.612	0.000	-0.673	-0.206
	std.lv	std.all	std.nox							
## 1	-0.163	-0.279	-0.279							
## 2	-0.219	-0.139	-0.485							
## 3	1.349	0.500	1.749							
## 4	0.198	0.145	0.256							
## 5	-0.238	-0.299	-0.528							
## 6	0.132	0.648	0.648							
## 7	0.384	0.645	0.645							
## 8	0.082	1.000	0.082							
## 9	0.094	0.580	0.094							
## 10	0.320	1.000	0.320							
## 11	4.521	10.011	10.011							
## 12	2.697	3.497	3.497							
## 13	0.507	1.775	0.507							
## 14	2.824	4.995	2.824							
## 15	-0.220	-0.139	-0.488							

```
## 16 -0.219 -0.139 -0.485
## 17 -0.440 -0.278 -0.974
```

5.6.1 A Figure and a Table

Let's look at a figure to see if we did what we think we did. And to also get a graphic representation of our results.

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = Kim_fit_covs)
```



```
tidySEM::get_layout(Kim_fit_covs)

##      [,1]  [,2]  [,3]
## [1,] NA    "REMS" NA
## [2,] "CMI" "ANX"  "PWB"
## attr(,"class")
## [1] "layout_matrix" "matrix"      "array"
```

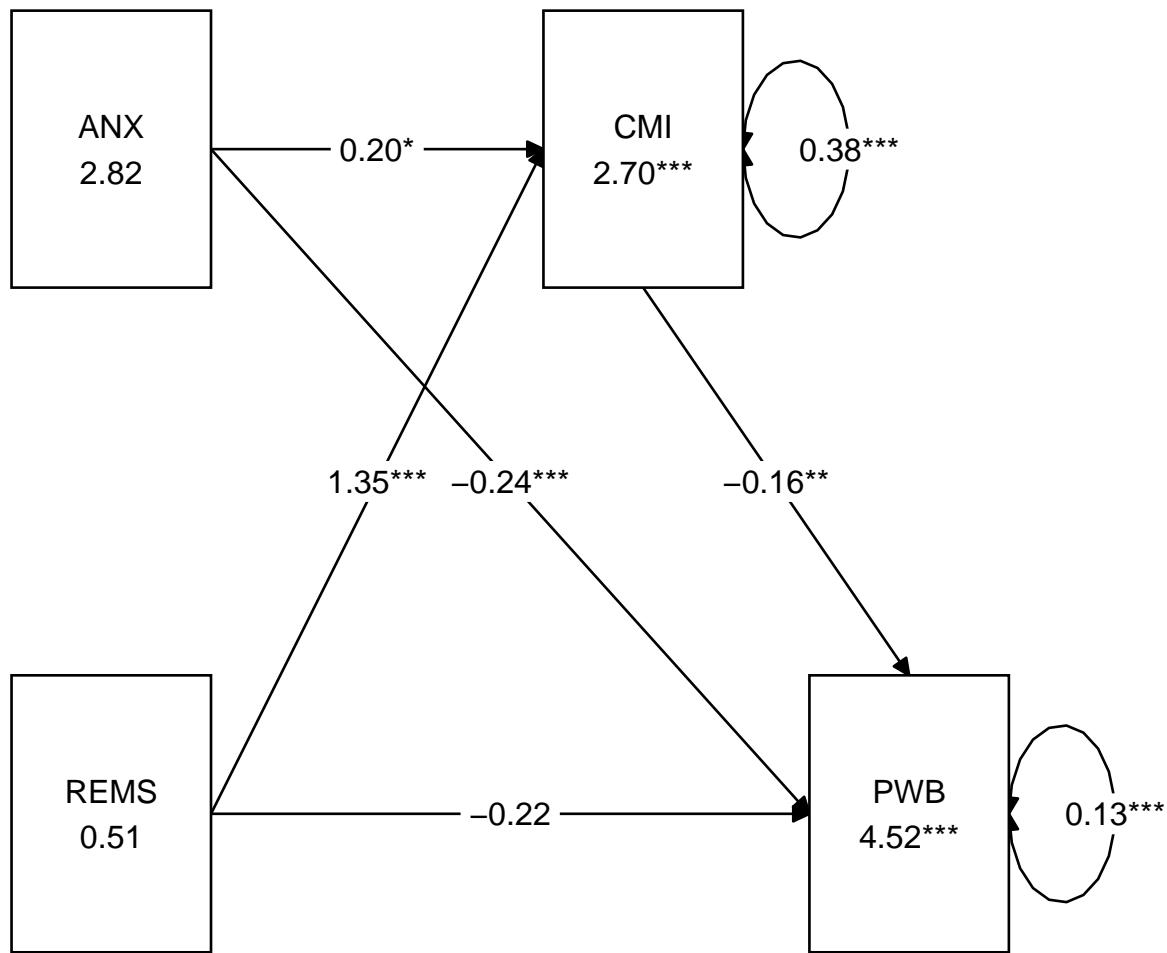
We can write code to remap them

```
med_map3 <- tidySEM::get_layout(
  "ANX", "", "CMI", "",
  "REMS", "", "", "PWB", rows=2)
med_map3
```

```
##      [,1]  [,2]  [,3]  [,4]
## [1,] "ANX"  ""    "CMI"  ""
## [2,] "REMS"  ""    ""     "PWB"
## attr(,"class")
## [1] "layout_matrix" "matrix"      "array"
```

We run again with our map and BOOM! Still needs tinkering for gorgeous, but hey!

```
tidySEM::graph_sem(Kim_fit_covs, layout = med_map3, rect_width = 1.5, rect_height = 1.25,
  spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



Below is code to create an outfile that could help with creating a table in a word document or spreadsheet. There will be output that is produced with SEM models that won't be relevant for this project.

```
write.csv(Kcov_ParEsts, file = "KimMedCov.csv")
```

Table 3

Model Coefficients Assessing Cultural Mistrust as a Mediator Between Racial Microaggressions and Well-Being

Cultural Mistrust (M)					Well-Being (Y)				
Antecedent	path	B	SE	p	path	B	SE	p	
Antecedent	path	B	SE	p	path	B	SE	p	

constant	i_M	2.697	0.245	<0.001	i_Y	4.521	0.209	<0.001
REMS (X)	a	1.349	0.191	<0.001	c'	-0.219	0.149	0.140
CMI (M)					b	-0.163	0.051	0.001
ANX (Cov)		0.198	0.096	0.039		-0.238	0.061	<0.001

$R^2 = 36\%$

$R^2 = 35\%$

Note. The value of the indirect effect was
 $B = -0.220, SE = 0.076, p = 0.004, 95CI(-0.385, -0.085)$.

5.6.2 APA Style Write-up

There are varying models for reporting the results of mediation. The Kim et al. [Kim et al., 2017] writeup is a great example. Rather than copying it directly, I have modeled my table after the ones in Hayes [2018] text. You'll notice that information in the table and text are minimally overlapping. APA style cautions us against redundancy in text and table.

Results

A simple mediation model examined the degree to which cultural mistrust mediated the effect of racial microaggressions on psychological well-being. Using the *lavaan* package (v 0.6-16) in R, coefficients for the each path, the indirect effect, and total effects were calculated. Additionally, the effect of covariate, anxiety, was mapped onto both the mediator and dependent variable. The model accounted for 36% of the variance in cultural mistrust and 35% of the variance in well-being. Supporting the notion of a mediated model, there was a statistically significant indirect effect ($B = -0.220, SE = 0.076, p = 0.004, 95CI[-0.385, -0.085]$) in combination with a non-significant direct effect ($B = -0.219, p = 0.140$) and a statistically significant total effect($B = -0.440, p < 0.001$).

5.7 STAY TUNED

A section on power analysis is planned and coming soon! My apologies that it's not quite Ready.

5.8 Residual and Related Questions...

..that you might have; or at least I had, but if had answered them earlier it would have disrupt the flow.

1. Are you sure you can claim a significant indirect effect in the presence of a non-significant total effect? Hayes [2018] is.

- In the section subtitled, “What about Baron & Kenny” (chapter 4), Hayes argues from both logical/philosophical and statistical perspectives that the size of the total effect does not constrain or determine the size of the indirect effect. That is, an indirect effect can be different from zero even when the total effect is not (pp. 117-119).
2. The output we get is different from the output in the journal article being used as the research vignette. Why? And should we worry about it?
- We are simulating data. This gives us some advantages in that (unless we specify it), we never have missingness and our variables should be normally distributed. Because we are working from means, standard deviations, and correlations, our data will never be the same as the original researcher. That said, we can compare our results to the journal to *check out work*. In fact, in this very chapter, I got turned around (e.g., first accidentally swapping the mediator and IV; then using the wrong DV) and was able to compare my work against the journal article to correct my errors.
3. Some of the statistics you are reporting are different than the ones in Hayes and the ones that use the PROCESS macro (e.g., what happened to the *F* test)?
- The default estimator for *lavaan* is maximum likelihood (ML) and Hayes uses ordinary least squares (OLS). This affects both the values of coefficients, standard errors, AND the type of statistics that are reported.
 - You can ask for OLS regression by adding the statement “estimator =”GLS”. Even with this option, I have not discovered a way to obtain the *F* tests for the overall model. Researchers seem to be comfortable with this, even asking for less than we did (e.g., many do not request R square).
 - Best I can tell, researchers who do want this might use a combination of packages, using GLS estimators in *lavaan* (this easily gets them the bootstrapped CIs) and the move to a different regression package to get the intercepts and *F* tests. If I did this I would triple check to make sure that all the output really lined up.
4. Why did we ignore the traditional fit statistics associated with structural equation modeling (e.g., CFI, RMSEA).
- I hesitate to do this with models that do not include latent variables. Therefore, we asked for an “in-between” amount of info that should be sufficient for publication submission (any editor may have their own preferences and ask for more).
5. What if I have missing data?
- When we enter the *lavaan* world we do get options other than multiple imputation. In today’s example we used the “sem” fitting function. Unless otherwise specified, listwise deletion (deleting the entire case when one of its variables is used to estimate the model) is the default in *lavaan*. If data are MCAR or MAR, you can add the argument *missing* = “ml” (or its alias *missing* = “fiml”). More here <https://users.ugent.be/~yrosseel/lavaan/lavaan2.pdf> on the 1.7/Missing data in lavaan slide.
 - That said, the type of estimator matters. If you estimate your data with GLS (generalized least squares) or WLS (weighted least squares), you are required to have complete data (however you got it). We used maximum likelihood and, even though we had non-missing data, I used the *missing* = “fiml” code.

5.9 Practice Problems

The three problems described below are designed to grow with the subsequent chapters on complex mediation and conditional process analysis (i.e., moderated mediation). Therefore, I recommend that you select a dataset that includes at least four variables. If you are new to this topic, you may wish to select variables that are all continuously scaled. The IV and moderator (subsequent chapters) could be categorical (if they are dichotomous, please use 0/1 coding; if they have more than one category it is best if they are ordered). You will likely encounter challenges that were not covered in this chapter. Search for and try out solutions, knowing that there are multiple paths through the analysis.

The suggested practice problem for this chapter is to conduct a simple mediation.

5.9.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed

If this topic feels a bit overwhelming, simply change the random seed in the data simulation, then rework the problem. This should provide minor changes to the data (maybe in the second or third decimal point), but the results will likely be very similar.

5.9.2 Problem #2: Rework the research vignette, but swap one or more variables

Use the simulated data, but select one of the other models that was evaluated in the Kim et al. [2017] study. Compare your results to those reported in the manuscript.

5.9.3 Problem #3: Use other data that is available to you

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 a simple mediation.

5.9.4 Grading Rubric

Assignment Component	Points Possible	Points Earned
1. Assign each variable to the X, Y, or M roles (ok but not required to include a cov)	5	_____
2. Import the data and format the variables in the model	5	_____
3. Specify and run the lavaan model	5	_____
4. Use tidySEM to create a figure that represents your results	5	_____
5. Create a table that includes regression output for the M and Y variables	5	_____
6. Represent your work in an APA-style write-up	5	_____
7. Explanation to grader	5	_____

Assignment Component	Points Possible	Points Earned
8. Be able to hand-calculate the indirect, direct, and total effects from the a, b, & c' paths	5	_____
Totals	35	_____

5.10 Homeworked Example

[Screencast Link](#)

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to conduct a simple mediation.

5.10.1 Assign each variable to the X, Y, or M roles (ok but not required to include a covariate)

X = Centering: explicit recentering (0 = precentered; 1 = recentered) M = TradPed: traditional pedagogy (continuously scaled with higher scores being more favorable) Y = SRPed: socially responsive pedagogy (continuously scaled with higher scores being more favorable)

Specify a research model

I am hypothesizing that the evaluation of social responsive pedagogy is predicted by intentional recentering through traditional pedagogy.

Import the data and format the variables in the model

```
raw <- readRDS("ReC.rds")
```

I need to score the TradPed and SRPed variables

```
TradPed_vars <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback",
  "ClearOrganization", "ClearPresentation")
raw$TradPed <- sjstats::mean_n(raw[, ..TradPed_vars], 0.75)

SRPed_vars <- c("InclusvClassrm", "EquitableEval", "MultPerspectives",
  "DEIintegration")
raw$SRPed <- sjstats::mean_n(raw[, ..SRPed_vars], 0.75)
```

I will create a babydf.

```
babydf <- dplyr::select(raw, Centering, TradPed, SRPed)
```

Let's check the structure of the variables:

```
str(babydf)
```

Specify and run the lavaan model

```
ReCMed <- "
  SRPed ~ b*TradPed + c_p*Centering
  TradPed ~ a*Centering

  indirect := a*b
  direct   := c_p
  total_c  := c_p + (a*b)
  ""

set.seed(231002) #needed for reproducible results since lavaan introduced randomness into some random numbers
ReCfit <- lavaan::sem(ReCMed, data = babydf, se = "bootstrap", missing = "fiml")
ReCsummary <- lavaan::summary(ReCfit, standardized = T, rsq = T, fit = TRUE,
  ci = TRUE)
ReC_ParamEsts <- lavaan::parameterEstimates(ReCfit, boot.ci.type = "bca.simple",
  standardized = TRUE)
ReCsummary
```

```
## lavaan 0.6.16 ended normally after 14 iterations
##
##   Estimator                               ML
##   Optimization method                    NLMINB
##   Number of model parameters             7
##   Number of observations                310
##   Number of missing patterns              4
##
##   Model Test User Model:
##
##   Test statistic                         0.000
##   Degrees of freedom                      0
##
##   Model Test Baseline Model:
##
##   Test statistic                         216.492
##   Degrees of freedom                      3
##   P-value                                0.000
##
```

```

## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)           1.000
## Tucker-Lewis Index (TLI)             1.000
##
## Robust Comparative Fit Index (CFI)    1.000
## Robust Tucker-Lewis Index (TLI)      1.000
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)        -506.434
## Loglikelihood unrestricted model (H1) -506.434
##
## Akaike (AIC)                         1026.868
## Bayesian (BIC)                       1053.024
## Sample-size adjusted Bayesian (SABIC) 1030.823
##
## Root Mean Square Error of Approximation:
##
## RMSEA                               0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.000
## P-value H_0: RMSEA <= 0.050          NA
## P-value H_0: RMSEA >= 0.080          NA
##
## Robust RMSEA                          0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.000
## P-value H_0: Robust RMSEA <= 0.050   NA
## P-value H_0: Robust RMSEA >= 0.080   NA
##
## Standardized Root Mean Square Residual:
##
## SRMR                                0.000
##
## Parameter Estimates:
##
## Standard errors                      Bootstrap
## Number of requested bootstrap draws  1000
## Number of successful bootstrap draws 1000
##
## Regressions:
##                   Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## SRPed ~
##   TradPed (b)    0.549   0.046 12.067  0.000   0.458   0.645
##   Centerng (c_p) 0.127   0.047  2.684  0.007   0.036   0.219
## TradPed ~
##   Centerng (a)  -0.101   0.090 -1.121  0.262  -0.287   0.080

```

```

## Std.lv Std.all
##
##      0.549    0.716
##      0.127    0.107
##
##     -0.101   -0.066
##
## Intercepts:
##                   Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .SRPed          2.006    0.231   8.689   0.000   1.543    2.442
## .TradPed        4.394    0.139  31.707   0.000   4.109    4.675
## Std.lv Std.all
##      2.006    3.440
##      4.394    5.778
##
## Variances:
##                   Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .SRPed          0.165    0.017   9.467   0.000   0.130    0.203
## .TradPed        0.576    0.070   8.225   0.000   0.444    0.724
## Std.lv Std.all
##      0.165    0.486
##      0.576    0.996
##
## R-Square:
##                   Estimate
## SRPed           0.514
## TradPed         0.004
##
## Defined Parameters:
##                   Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## indirect        -0.056   0.052  -1.077   0.282  -0.168    0.042
## direct          0.127   0.047   2.682   0.007   0.036    0.219
## total_c         0.071   0.068   1.045   0.296  -0.055    0.204
## Std.lv Std.all
##     -0.056   -0.047
##      0.127    0.107
##      0.071    0.060

```

ReC_ParamEsts

	lhs	op	rhs	label	est	se	z	pvalue	ci.lower	ci.upper
## 1	SRPed	~	TradPed	b	0.549	0.046	12.067	0.000	0.459	0.645
## 2	SRPed	~	Centering	c_p	0.127	0.047	2.684	0.007	0.032	0.211
## 3	TradPed	~	Centering	a	-0.101	0.090	-1.121	0.262	-0.292	0.075
## 4	SRPed	~~	SRPed		0.165	0.017	9.467	0.000	0.135	0.209
## 5	TradPed	~~	TradPed		0.576	0.070	8.225	0.000	0.462	0.741
## 6	Centering	~~	Centering		0.241	0.000	NA	NA	0.241	0.241

```

## 7      SRPed ~1          2.006 0.231  8.689  0.000   1.534  2.433
## 8      TradPed ~1        4.394 0.139 31.707  0.000   4.114  4.683
## 9      Centering ~1      1.406 0.000     NA     NA   1.406  1.406
## 10     indirect :=      a*b indirect -0.056 0.052 -1.077  0.282  -0.171  0.038
## 11     direct :=         c_p   direct  0.127 0.047  2.682  0.007   0.032  0.211
## 12     total_c :=        c_p+(a*b) total_c  0.071 0.068  1.045  0.296  -0.053  0.205
##      std.lv std.all std.nox
## 1    0.549   0.716   0.716
## 2    0.127   0.107   0.217
## 3   -0.101  -0.066  -0.133
## 4    0.165   0.486   0.486
## 5    0.576   0.996   0.996
## 6    0.241   1.000   0.241
## 7    2.006   3.440   3.440
## 8    4.394   5.778   5.778
## 9    1.406   2.863   1.406
## 10   -0.056  -0.047  -0.096
## 11   0.127   0.107   0.217
## 12   0.071   0.060   0.122

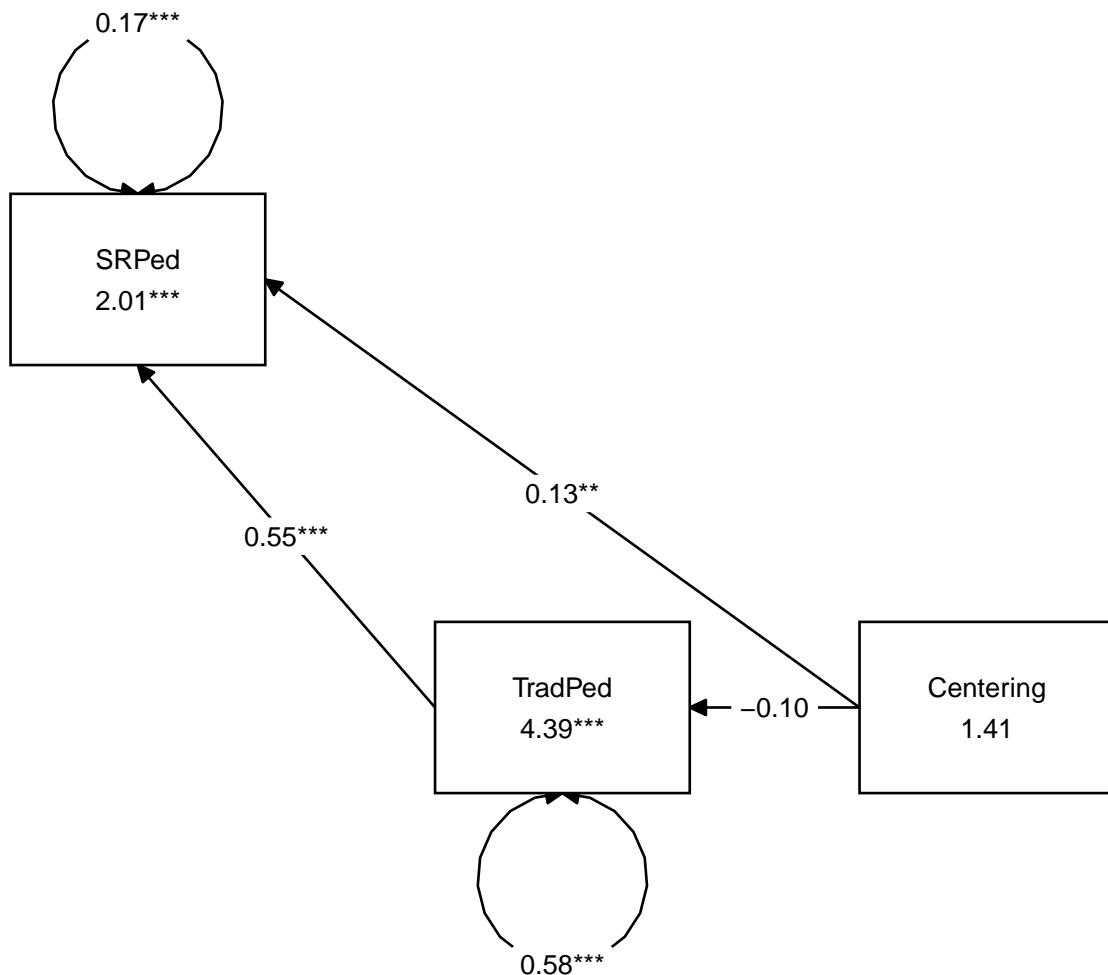
```

Use tidySEM to create a figure that represents your results

```

# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = ReCfit)

```



```
tidySEM::get_layout(ReCfit)
```

```
##      [,1]     [,2]     [,3]
## [1,] "SRPed" "TradPed" "Centering"
## attr(,"class")
## [1] "layout_matrix" "matrix"       "array"
```

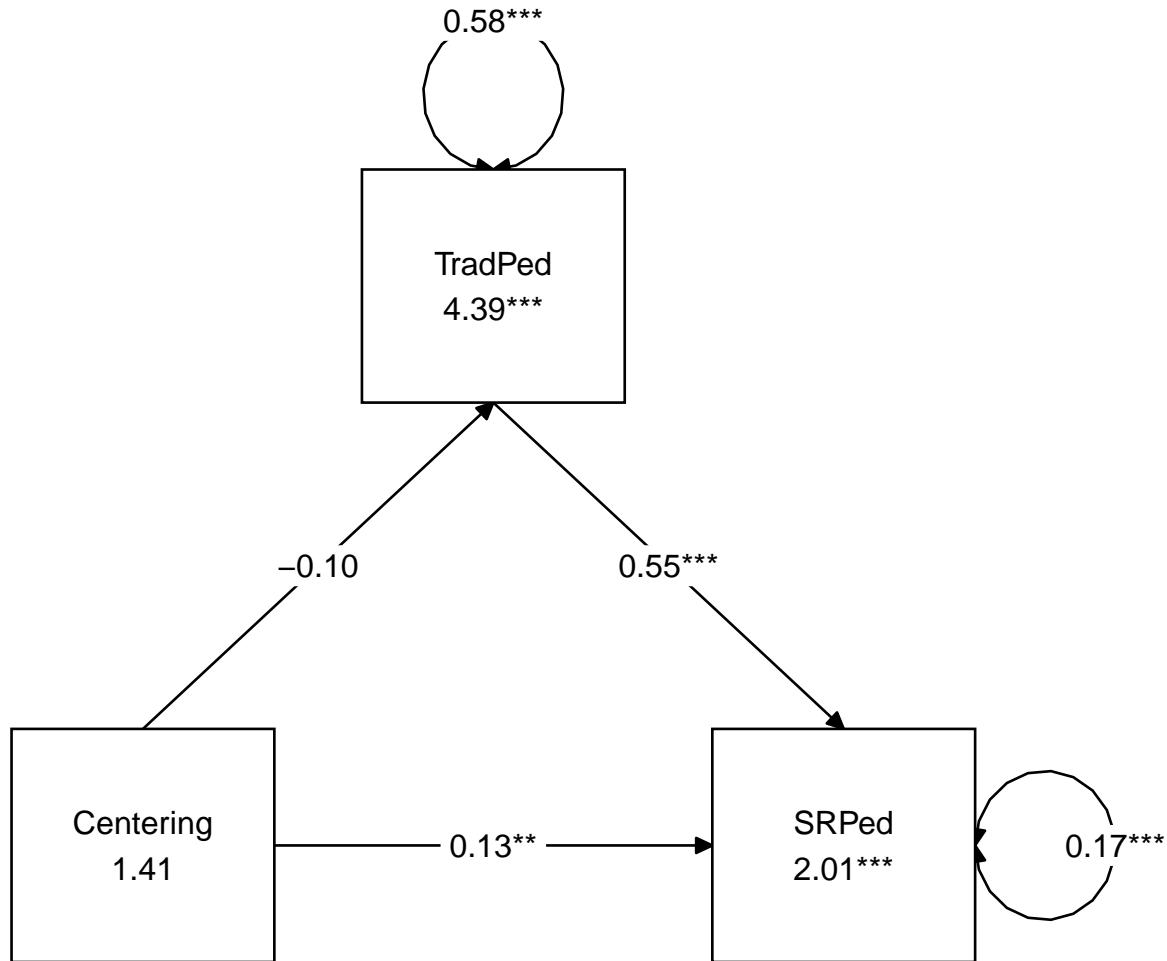
We can write code to remap them

```
med_map <- tidySEM::get_layout("", "TradPed", "", "Centering", "", "SRPed",
                                rows = 2)
med_map
```

```
##      [,1]     [,2]     [,3]
## [1,] ""        "TradPed" ""
## [2,] "Centering" ""        "SRPed"
```

```
## attr(,"class")
## [1] "layout_matrix" "matrix"      "array"
```

```
tidySEM::graph_sem(ReCfit, layout=med_map, rect_width = 1.5, rect_height = 1.25, spacing_x = 2)
```



Create a table that includes regression output for the M and Y variables

```
write.csv(ReC_ParamEsts, file = "ReCSimpMed.csv")
```

Table 1

Model Coefficients Assessing Traditional Pedagogy as a Mediator Between Centering and Socially Responsive Pedagogy

Traditional Pedagogy (M)					Socially Responsive Pedagogy (Y)			
Antecedent	path	B	SE	p	path	B	SE	p
constant	i_M	4.394	0.139	< 0.001	i_Y	2.006	0.231	< 0.001
Centering (X)	a	-0.101	0.090	0.262	c'	0.127	0.047	0.007
TradPed (M)					b	0.549	0.046	< 0.001

$R^2 = 0.4\%$	$R^2 = 51\%$
---------------	--------------

Note. Centering: 0 = pre-centered, 1 = recentered. TradPed is traditional pedagogy. The value of the indirect effect was $B = -0.056, SE = 0.051, p = 0.272, 95CI(-0.163, 0.035)$

Represent your work in an APA-style write-up

A simple mediation model examined the degree to which evaluations of traditional pedagogy mediated the relation of explicit recentering on socially responsive pedagogy. Using the *lavaan* package (v 0.6-16) in R, coefficients for each path, the indirect effect, and total effects were calculated. These values are presented in Table 1 and illustrated in Figure 1. Results suggested that negligible (.4%) of the variance was accounted for in traditional pedagogy. In contrast 51% of the variance was accounted for in socially responsive pedagogy. The indirect effect ($B = -0.056, SE = 0.051, p = 0.272, 95CI[-0.163, 0.035]$) was statistically significant. Comparing total and direct effects, the total effect of centering and traditional pedagogy on socially responsive pedagogy was not statistically significant ($B = 0.071, p = 0.302$). In contrast, the direct effect was ($B = 0.127, p = 0.008$ was not). This suggests that while centering and traditional pedagogy do influence socially responsive pedagogy, their influence is relatively independent.

```
apaTables::apa.cor.table(babydf, table.number = 1, show.sig.stars = TRUE,
  landscape = TRUE, filename = NA)
```

```
##
## 
## Table 1
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##      Variable   M     SD    1
##      1. TradPed 4.25  0.76
## 
##      2. SRPed   4.52  0.58 .71**
##                           [.65, .76]
##
```

```
##  
## 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.  
##
```

Explanation to grader

Be able to hand-calculate the indirect, direct, and total effects from the a, b, & c' paths

- Indirect = $a \cdot b$
- Direct = Total minus indirect
- Total = $(a \cdot b) + c'$

Chapter 6

Complex Mediation

[Screencasted Lecture Link](#)

The focus of this chapter is the extension of simple mediation to models with multiple mediators. In these models with greater complexity we look at both parallel and serial mediation. There is also more elaboration on some of the conceptual issues related to the estimation of indirect effects.

6.1 Navigating this Lesson

There is about 1 hour and 20 minutes 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, there are a few 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

Learning objectives from this lecture include the following:

- Define *epiphénomérité* and explain how it is related to (and supports the notion of) multiple mediation.
- Distinguish between parallel and serial mediation models.
- Locate and interpret *lavaan* output from multiply mediated models including
 - identifying coefficients,
 - percentage of variance accounted for,
- all the effects (total, direct, indirect, total indirect),
- contrasts (comparing the significance of the indirect effects).
- Explain the limitations of the classic approach [[Baron and Kenny, 1986](#)] to mediation.

6.1.2 Planning for Practice

The suggestions for practice in this chapter include conducting parallel, serial, and/or mediation models. Options of graded complexity could include:

- Rework the problem in the chapter by changing the random seed in the code that simulates the data. This should provide minor changes to the data, but the results will likely be very similar.
- There are a number of variables in the dataset that sourced the research vignettes for this and the prior chapter on [simple mediation](#). Swap out one or more variables in a parallel or serial (or both) model.
- Conduct a parallel or serial (or both) mediation with data to which you have access. This could include data you simulate on your own or from a published article.

6.1.3 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.

- Hayes, A. F. (2022). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. New York, NY: Guilford Press.
 - **Chapter 5: More than one mediator:** This chapter walks the reader through parallel and serial mediation models. We will do both!
 - **Appendix A: Using PROCESS:** An essential tool for PROCESS users because, even when we are in the R environment, this is the “idea book.” That is, the place where all the path models are presented in figures.

6.1.4 Packages

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

```

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

```

6.2 Complex Mediation

The simple mediation model is quite popular, but also limiting in that it:

- frequently oversimplifies the processes we want to study, and
- is likely mis-specified, in that there are unmodeled mechanisms.

Hayes [2022b] identified four reasons to consider multiply mediated models:

- We are generally interested in MULTIPLE mechanisms
- A mechanism (such as a mediator) in the model, might, itself be mediated (i.e., mediated mediation)
- *Epiphenomenality* (“unknown confounds”): a proposed mediator could be related to an outcome not because it causes the outcome, but because it is correlated with another variable that is causally influencing the outcome. This is a noncausal alternative explanation for an association.
- Including multiple mediators allows formal comparison of the strength of the mediating mechanisms.

There are two multiple mediator models that we will consider: parallel, serial.

6.3 Workflow for Complex Mediation

The following is a proposed workflow for conducting a complex mediation.

Conducting a parallel or serial (i.e., complex) mediation involves the following steps:

1. Conducting an a priori power analysis to determine the appropriate sample size.
 - This will require estimates of effect that are drawn from pilot data, the literature, or both.
2. **Scrubbing** and **scoring** the data.
 - Guidelines for such are presented in the respective lessons.
3. Conducting data diagnostics, this includes:
 - item and scale level missingness,

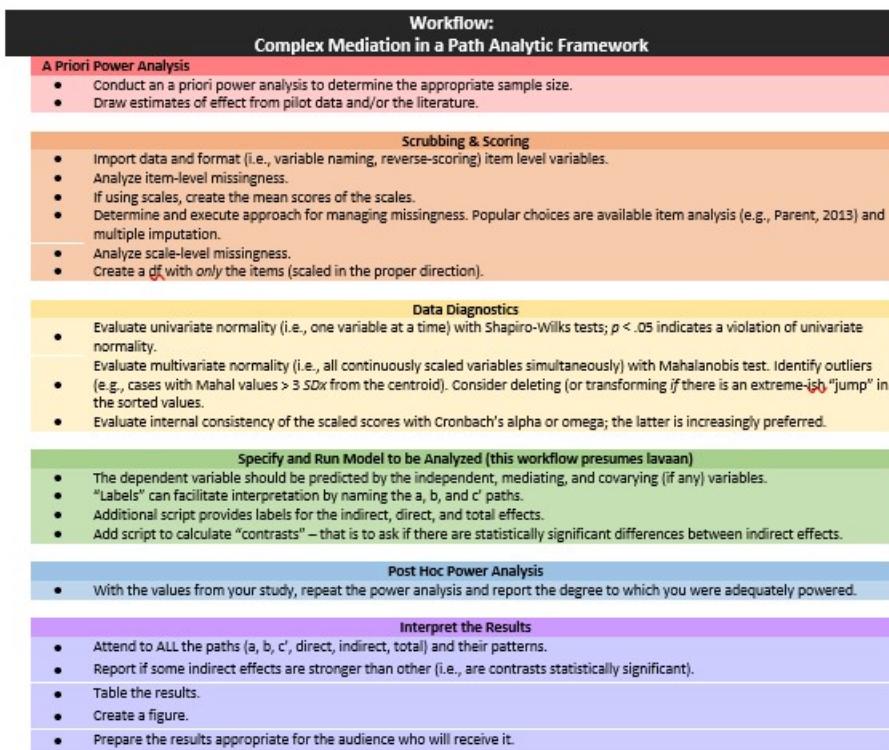


Figure 6.1: A colorful image of a workflow for complex mediation

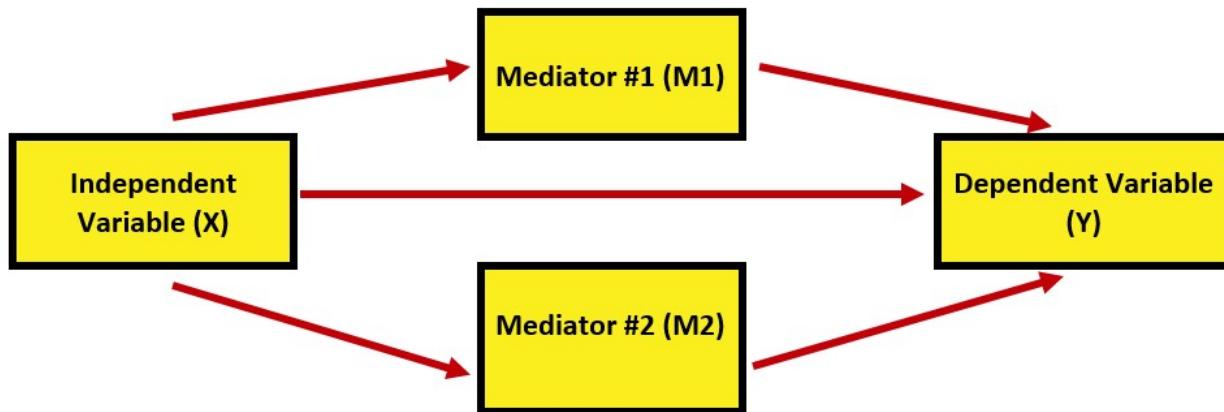
- internal consistency coefficients (e.g., alphas or omegas) for scale scores,
 - univariate and multivariate normality
- Specifying and running the model (this lesson presumes it will with the R package, *lavaan*).
 - The dependent variable should be predicted by the independent, mediating, and covarying (if any) variables.
 - "Labels" can facilitate interpretation by naming the a, b, and c' paths.
 - Additional script provides labels for the indirect, direct, and total effects.
 - With multiple indirect effects, specify contrasts to see if they are statistically significantly different form each other.
 - Conducting a post hoc power analysis.
 - Informed by your own results, you can see if you were adequately powered to detect a statistically significant effect, if, in fact, one exists.
 - Interpret and report the results.
 - Interpret ALL the paths and their patterns.
 - Report if some indirect effects are stronger than others (i.e., results of the contrasts).
 - Create a table and figure.
 - Prepare the results in a manner that is useful to your audience.

6.4 Parallel Mediation

Parallel multiple mediation: An antecedent variable X is modeled as influencing consequent Y directly as well as indirectly through two or more mediators, with the condition that no mediator causally influences another [Hayes, 2022b, p. 161]

With multiple mediation we introduce additional effects:

- *Direct effect, c'* (this is not new) quantifies how much two cases that differ by a unit on X are estimated to differ on Y – independent of all mediators.
- *Specific indirect effect, $a_i b_i$* , the individual mediated effects
- *Total indirect effects*, $\sum_{i=1}^k a_i b_i$ the sum of the values of the specific indirect effects. The total indirect effect can also be calculated by subtracting the direct effects from the total effects: $c - c'$
- *Total effect of X on Y, $c = c' + \sum_{i=1}^k a_i b_i$* (also not new) the sum of the direct and indirect effects. The total effect can also be estimated by regressing Y on X alone.
- *Contrasts* allow us to directly compare separate mediating effects to see if one indirect effect is stronger than the other.



In this parallel model, we can describe these effects this way:

- *Direct effect:* The effect of IV on the DV, accounting for two mediators (indirect effects) in the model.
- *Specific indirect effects:* There are indirect (or mediating) paths from the IV to the DV; through M1 and M2, respectively.
- *Total indirect effect of X on Y:* A sum of the value of indirect effects through the specific indirect effects (M1 and M2).
- *Total effect:* The sum of the direct and indirect effects. Also calculated by regressing Y (dependent variable) on X (independent variable) alone, without any other variables in the model.

Recall that for a complex mediation to be parallel, there can be no causal links between mediators. This is true in this example.

6.4.1 A Mechanical Example

Let's work a mechanical example with simulated data that assures a statistically significant outcome. Credit to this example is from the Paulo Toffanin website [Toffanin, 2017].

We can bake our own data by updating the script we used in simple mediation to add a second mediator.

6.4.1.1 Data Simulation

```
# Concerned that identical variable names across book chapters may be
# problematic, I'm adding 'p' in front the 'Data' variable.
set.seed(230925)
X <- rnorm(100)
M1 <- 0.5 * X + rnorm(100)
M2 <- -0.35 * X + rnorm(100)
Y <- 0.7 * M2 + 0.48 * M1 + rnorm(100)
pData <- data.frame(X = X, Y = Y, M1 = M1, M2 = M2)
```

Using what we learned in conducting a simple mediation in *lavaan*, we can look at the figure of our proposed model and *backwardstrace* the paths to write the code.

Remember...

- The model exists between 2 single quotation marks (the odd looking ' and ' at the beginning and end).
- You can write the Y as I have done in the R chunk below, or you can write the Y separately from each arrow, such as
 - $Y \sim b1*M1$
 - $Y \sim b2*M2$
 - $Y \sim c_p*X$
- Everything else transfers from our simple mediation, remember that
 - the asterisk (“*”) allows us to assign labels (a1, a2, b1, b2, etc.) to the paths; these are helpful for intuitive interpretation
 - that eyes/nose notation (:=) is used when creating a new variable that is a function of variables in the model, but not in the dataset (i.e., the a and b path).
 - in traditional mediation speak, the direct path from X to Y is c' (c prime) and the total effect of X to Y (with nothing else in the model) is just c . Hence the c_p label for c prime.
- Something new: the *contrast* statement (only one in this example, but you could have more) allows us to compare the indirect effects to each other. We specify it in the lavaan model, but then need to test it in a subsequent set of script.

- Note: In the online example, the writer adds code to correlate M1 and M2. This didn't/doesn't seem right to me and then, later, when we amend it to be a serial model, it made even less sense to have them be correlated.

6.4.1.2 Specifying *lavaan* code

```

parallel_med <- "
  Y ~ b1*M1 + b2*M2 + c_p*X
  M1 ~ a1*X
  M2 ~ a2*X

  indirect1 := a1 * b1
  indirect2 := a2 * b2
  contrast := indirect1 - indirect2
  total_indirects := indirect1 + indirect2
  total_c      := c_p + (indirect1) + (indirect2)
  direct := c_p
"

set.seed(230925) #needed for reproducibility especially when specifying bootstrapped confidence intervals
parallel_fit <- lavaan::sem(parallel_med, data = pData, se = "bootstrap",
  missing = "fiml", bootstrap = 1000)
pfit_sum <- lavaan::summary(parallel_fit, standardized = TRUE, rsq = T,
  fit = TRUE, ci = TRUE)
pfit_ParEsts <- lavaan::parameterEstimates(parallel_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
pfit_sum

## lavaan 0.6.16 ended normally after 1 iteration
##
##             ML
## Estimator
## Optimization method           NLMINB
## Number of model parameters    11
## 
##             100
## Number of observations
## Number of missing patterns     1
## 
## Model Test User Model:
## 
##             2.475
## Test statistic
## Degrees of freedom              1
## P-value (Chi-square)            0.116
## 
## Model Test Baseline Model:
## 
##             126.642
## Test statistic
## Degrees of freedom                6

```

```

## P-value          0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)          0.988
## Tucker-Lewis Index (TLI)            0.927
##
## Robust Comparative Fit Index (CFI)    0.988
## Robust Tucker-Lewis Index (TLI)      0.927
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)        -433.660
## Loglikelihood unrestricted model (H1) -432.423
##
## Akaike (AIC)                      889.321
## Bayesian (BIC)                    917.977
## Sample-size adjusted Bayesian (SABIC) 883.237
##
## Root Mean Square Error of Approximation:
##
## RMSEA                     0.121
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.322
## P-value H_0: RMSEA <= 0.050       0.161
## P-value H_0: RMSEA >= 0.080       0.772
##
## Robust RMSEA                  0.121
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.322
## P-value H_0: Robust RMSEA <= 0.050 0.161
## P-value H_0: Robust RMSEA >= 0.080 0.772
##
## Standardized Root Mean Square Residual:
##
## SRMR                   0.046
##
## Parameter Estimates:
##
## Standard errors           Bootstrap
## Number of requested bootstrap draws 1000
## Number of successful bootstrap draws 1000
##
## Regressions:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## Y ~
##   M1      (b1)    0.456    0.107    4.243    0.000    0.241    0.667
##   M2      (b2)    0.743    0.075    9.972    0.000    0.605    0.894

```

```

##      X          (c_p)    0.030    0.099    0.305    0.760   -0.161    0.221
##      M1 ~
##      X          (a1)    0.510    0.081    6.308    0.000    0.353    0.657
##      M2 ~
##      X          (a2)   -0.381    0.126   -3.014    0.003   -0.630   -0.129
##      Std.lv  Std.all
##
##      0.456    0.383
##      0.743    0.693
##      0.030    0.025
##
##      0.510    0.502
##
##      -0.381   -0.338
##
##  ## Intercepts:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##      .Y        0.113    0.098   1.155    0.248   -0.088    0.297
##      .M1      -0.089    0.099  -0.897    0.370   -0.279    0.098
##      .M2       0.017    0.121   0.139    0.889   -0.215    0.273
##      Std.lv  Std.all
##      0.113    0.083
##      -0.089   -0.078
##      0.017    0.013
##
##  ## Variances:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##      .Y        0.855    0.106   8.031    0.000    0.622    1.035
##      .M1      0.970    0.118   8.193    0.000    0.728    1.191
##      .M2       1.415    0.181   7.815    0.000    1.048    1.742
##      Std.lv  Std.all
##      0.855    0.465
##      0.970    0.748
##      1.415    0.886
##
##  ## R-Square:
##      Estimate
##      Y        0.535
##      M1      0.252
##      M2       0.114
##
##  ## Defined Parameters:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##      indirect1     0.233    0.068   3.435    0.001    0.110    0.376
##      indirect2    -0.283    0.094  -3.026    0.002   -0.473   -0.094
##      contrast      0.516    0.103   5.001    0.000    0.307    0.712
##      total_indircts -0.051    0.127  -0.400    0.689   -0.302    0.198
##      total_c       -0.021    0.131  -0.157    0.876   -0.277    0.238

```

```

##      direct          0.030    0.099    0.305    0.760   -0.161    0.221
## Std.lv  Std.all
##      0.233    0.192
##     -0.283   -0.234
##      0.516    0.426
##     -0.051   -0.042
##     -0.021   -0.017
##      0.030    0.025

```

pfit_ParEsts

	lhs	op	rhs	label	est	se	
## 1	Y	~	M1	b1	0.456	0.107	
## 2	Y	~	M2	b2	0.743	0.075	
## 3	Y	~	X	c_p	0.030	0.099	
## 4	M1	~	X	a1	0.510	0.081	
## 5	M2	~	X	a2	-0.381	0.126	
## 6	Y	~~	Y		0.855	0.106	
## 7	M1	~~	M1		0.970	0.118	
## 8	M2	~~	M2		1.415	0.181	
## 9	X	~~	X		1.253	0.000	
## 10	Y	~1			0.113	0.098	
## 11	M1	~1			-0.089	0.099	
## 12	M2	~1			0.017	0.121	
## 13	X	~1			0.009	0.000	
## 14	indirect1 :=		a1*b1	indirect1	0.233	0.068	
## 15	indirect2 :=		a2*b2	indirect2	-0.283	0.094	
## 16	contrast :=	indirect1-indirect2		contrast	0.516	0.103	
## 17	total_indirects :=	indirect1+indirect2	total_indirects		-0.051	0.127	
## 18	total_c := c_p+(indirect1)+(indirect2)		total_c		-0.021	0.131	
## 19	direct :=	c_p	direct		0.030	0.099	
	z	pvalue	ci.lower	ci.upper	std.lv	std.all	std.nox
## 1	4.243	0.000	0.227	0.658	0.456	0.383	0.383
## 2	9.972	0.000	0.597	0.890	0.743	0.693	0.693
## 3	0.305	0.760	-0.160	0.227	0.030	0.025	0.022
## 4	6.308	0.000	0.356	0.660	0.510	0.502	0.448
## 5	-3.014	0.003	-0.624	-0.125	-0.381	-0.338	-0.302
## 6	8.031	0.000	0.671	1.078	0.855	0.465	0.465
## 7	8.193	0.000	0.758	1.248	0.970	0.748	0.748
## 8	7.815	0.000	1.113	1.834	1.415	0.886	0.886
## 9	NA	NA	1.253	1.253	1.253	1.000	1.253
## 10	1.155	0.248	-0.078	0.301	0.113	0.083	0.083
## 11	-0.897	0.370	-0.286	0.094	-0.089	-0.078	-0.078
## 12	0.139	0.889	-0.234	0.240	0.017	0.013	0.013
## 13	NA	NA	0.009	0.009	0.009	0.008	0.009
## 14	3.435	0.001	0.124	0.395	0.233	0.192	0.172
## 15	-3.026	0.002	-0.483	-0.105	-0.283	-0.234	-0.209

```
## 16 5.001 0.000 0.299 0.708 0.516 0.426 0.380
## 17 -0.400 0.689 -0.304 0.193 -0.051 -0.042 -0.037
## 18 -0.157 0.876 -0.251 0.252 -0.021 -0.017 -0.015
## 19 0.305 0.760 -0.160 0.227 0.030 0.025 0.022
```

6.4.1.3 A note on indirect effects and confidence intervals

Before we move onto interpretation, I want to stop and look at both p values and confidence intervals. Especially with Hayes [2022b] PROCESS macro, there is a great deal of emphasis on the use of bootstrapped confidence intervals to determine the statistical significance of the indirect effects. In fact, PROCESS output has (at least historically) not provided p values with the indirect effects. This is because, especially in the ordinary least squares context, bias-corrected bootstrapped confidence intervals are more powerful (i.e., they are more likely to support a statistically significant result) than p values.

An excellent demonstration of this phenomena was provided by Mallinckrodt et al. [2006] where they compared confidence intervals produced by the normal theory method to those that are bias corrected. The bias corrected intervals were more powerful to determining if there were statistically significant indirect effects.

The method we have specified in *lavaan* produced bias-corrected confidence intervals. The p values and corresponding confidence intervals should be consistent with each other. That is, if $p < .05$, then the CI95s should not pass through zero. Of course we can always check to be certain this is true. For this reason, I will report p values in my results. There are reviewers, though, who may prefer that you report CI95s (or both).

6.4.1.4 Figures and Tables

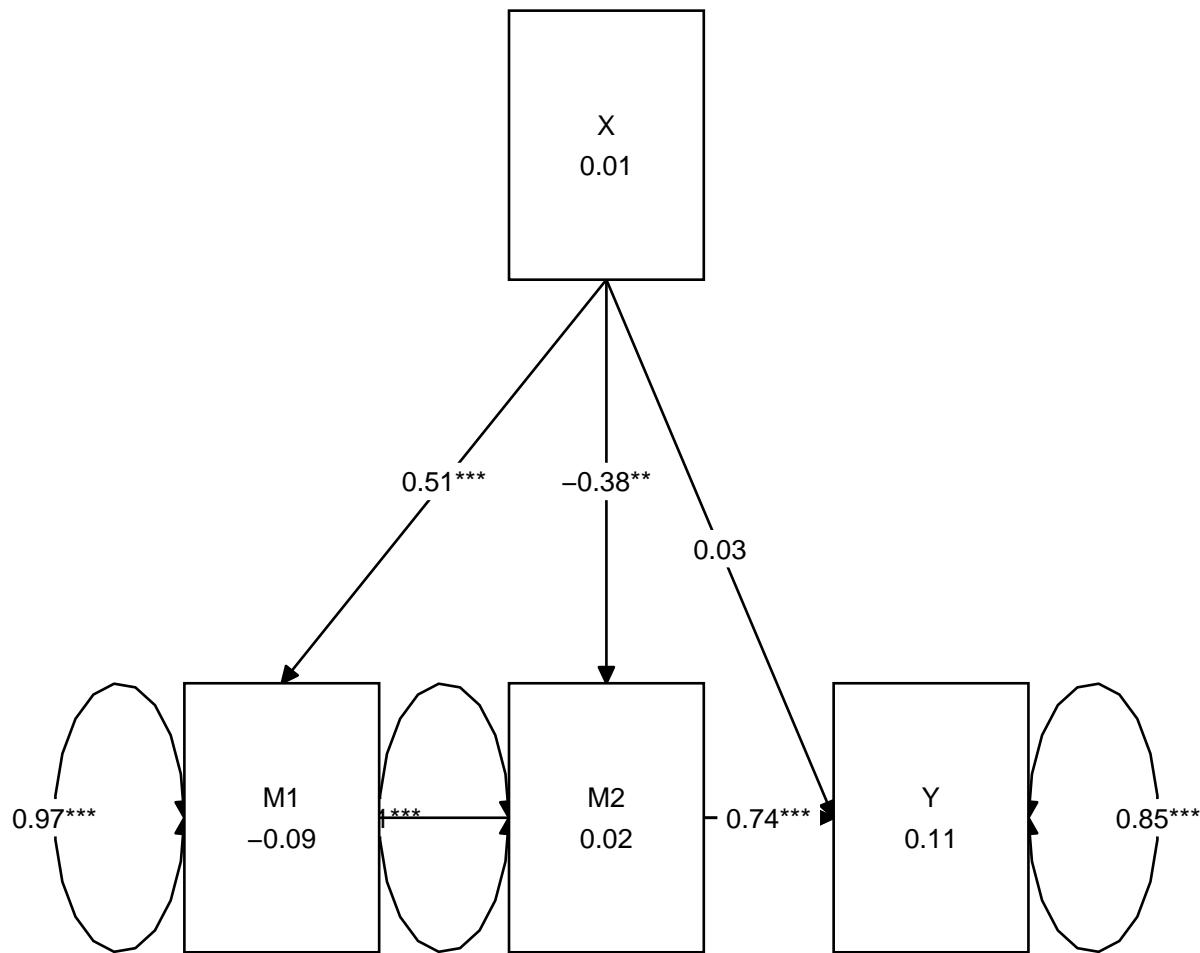
To assist in table preparation, it is possible to export the results to a .csv file that can be manipulated in Excel, Microsoft Word, or other program to prepare an APA style table.

```
write.csv(pfit_ParEsts, file = "pfit_ParEsts.csv")
```

We can use the package [tidySEM](#) to create a figure that includes the values on the path.

Here's what the base package gets us

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = parallel_fit)
```



We can create model that communicates more intuitively with a little tinkering. First, let's retrieve the current “map” of the layout.

```
tidySEM::get_layout(parallel_fit)
```

```
##      [,1] [,2] [,3]
## [1,] NA   "X"  NA
## [2,] "M1" "M2" "Y"
## attr(),"class")
## [1] "layout_matrix" "matrix"       "array"
```

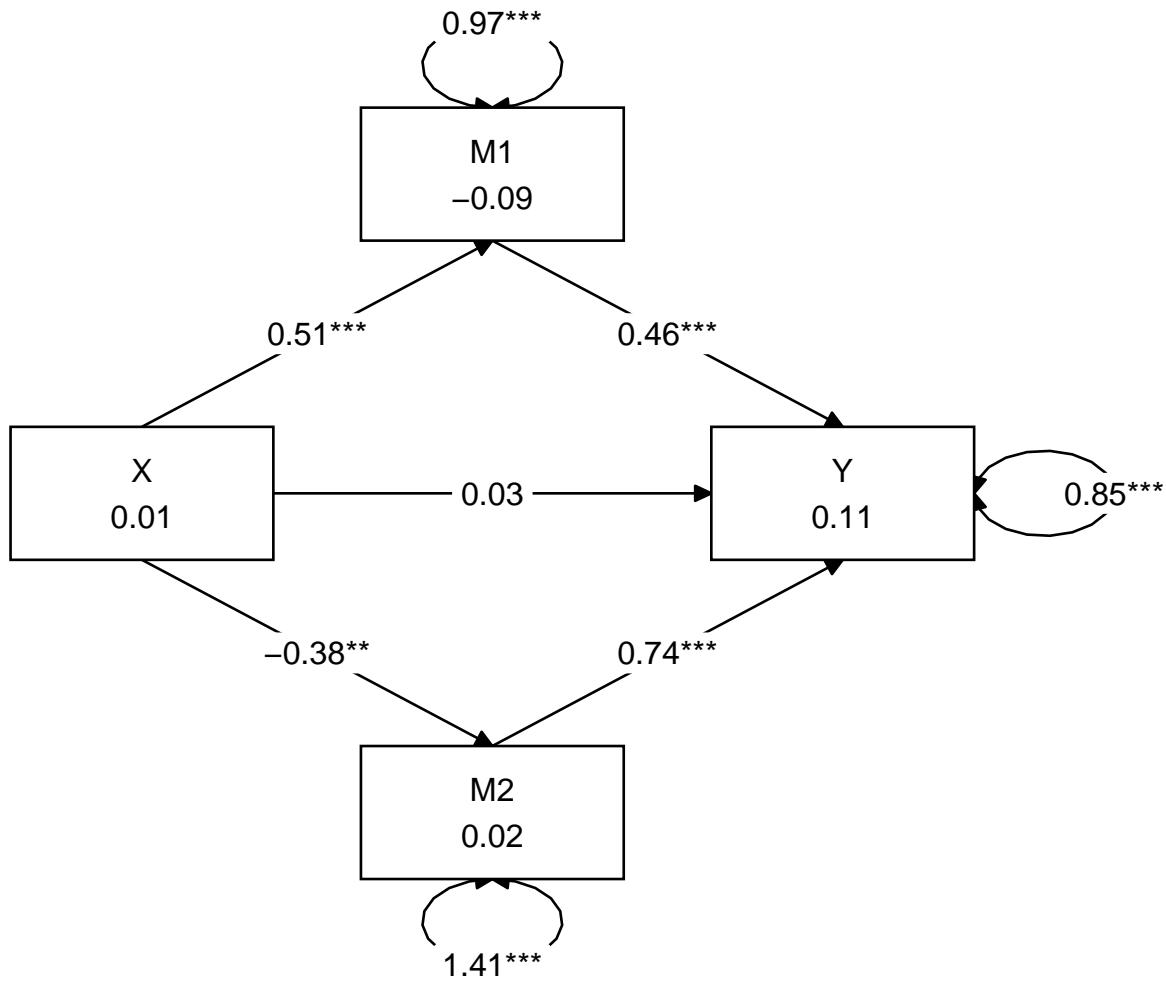
To create the figure I showed at the beginning of the chapter, we will want three rows and three columns.

```
parallel_map <- tidySEM::get_layout("", "M1", "", "X", "", "Y", "", "M2",
 "", rows = 3)
parallel_map
```

```
##      [,1] [,2] [,3]
## [1,] ""   "M1"  ""
## [2,] "X"  ""   "Y"
## [3,] ""   "M2"  ""
## attr(,"class")
## [1] "layout_matrix" "matrix"       "array"
```

We can update our figure by supplying this new map and adjusting the object and text sizes.

```
tidySEM::graph_sem(parallel_fit, layout = parallel_map, rect_width = 1.5,
rect_height = 1.25, spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



There are a number of ways to tabalize the data. You might be surprised to learn that a number of articles that analyze mediating effects focus their presentation on those values and not the traditional intercepts and B weights. This is the approach I have taken in this chapter.

Table 1

 Model Coefficients Assessing M1 and M2 as Parallel Mediators Between X and Y

Predictor	B	SE _B	p	R ²
M1				.25
Constant	-0.089	0.099	0.370	
X (a_1)	0.510	0.081	<0.001	
M2				.11
Constant	0.017	0.121	0.889	
X (a_2)	-0.381	0.126	0.003	
DV				.54
Constant	0.113	0.098	0.248	
M1 (b_1)	0.456	0.107	<0.001	
M2 (b_2)	0.743	0.075	<0.001	
X (c')	0.030	0.099	0.760	
Summary of Effects	B	SE _B	p	95% CI
Total	-0.021	0.131	0.876	-0.251, 0.252
Indirect 1 ($a_1 * a_2$)	0.233	0.068	0.001	0.124, 0.395
Indirect 2 ($b_1 * b_2$)	-0.283	0.094	0.002	-0.483, -0.105
Total indirects	-0.051	0.127	0.689	-0.304, 0.193
Contrast (Ind1 - Ind2)	0.516	0.103	<0.001	0.299, 0.708

Note. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

6.4.1.5 APA Style Writeup

You may notice that my write-up includes almost no statistical output. This is consistent with APA style that avoids redundancy in text and table. When I want to emphasize a specific result, I may duplicate some output in the text.

A model of parallel multiple mediation was analyzed examining the degree to which importance of M1 and M2 mediated the relation of X on Y. Hayes [2022b] recommended this strategy over simple mediation models because it allows for all mediators to be

examined, simultaneously. The resultant direct and indirect values for each path account for other mediation paths. Using the *lavaan* (*v. 0.6-16*) package in R, coefficients for specific indirect, total indirect, direct, and total were computed. Path coefficients refer to regression weights, or slopes, of the expected changes in the dependent variable given a unit change in the independent variables.

Results (depicted in Figure 1 and presented in Table 1) suggest that 54% of the variance in Y is accounted for by the model. Neither the total nor direct effect of X on Y were statistically significant. In contrast, both indirect effects were statistically significant. A pairwise comparison of the specific indirect effects indicated that the strength of the effects were statistically significantly different from each other. In summary, the effect of X on Y is mediated through M1 and M2, with a stronger influence through M2.

You may notice this write-up included only one statistic. I offered this as an example of avoiding redundancy in text and table. When tables and figures convey maximal information, the results section may be used to describe the patterns – including numbers when they reduce the cognitive load for the readers and reviewers.

Let's turn now to the research vignette and work an example with simulated data from that example. Because the research vignette use an entirely new set of output I will either restart R or clear my environment so that there are a few less objects “in the way.”

6.4.2 Research Vignette

The research vignette comes from the Lewis, Williams, Peppers, and Gadson’s [2017] study titled, “Applying Intersectionality to Explore the Relations Between Gendered Racism and Health Among Black Women.” The study was published in the Journal of Counseling Psychology. Participants were 231 Black women who completed an online survey.

Variables used in the study included:

- **GRMS:** Gendered Racial Microaggressions Scale [Lewis and Neville, 2015] is a 26-item scale that assesses the frequency of nonverbal, verbal, and behavioral negative racial and gender slights experienced by Black women. Scaling is along six points ranging from 0 (*never*) to 5 (*once a week or more*). Higher scores indicate a greater frequency of gendered racial microaggressions. An example item is, “Someone has tried to ‘put me in my place.’”
- **MntlHlth** and **PhysHlth:** Short Form Health Survey - Version 2 [Ware et al., 1995] is a 12-item scale used to report self-reported mental (six items) and physical health (six items). Although the article did not specify, when this scale is used in other contexts [e.g., Kim et al., 2017], a 6-point scale has been reported. Higher scores indicate higher mental health (e.g., little or no psychological distress) and physical health (e.g., little or no reported symptoms in physical functioning). An example of an item assessing mental health was, “How much of the time during the last 4 weeks have you felt calm and peaceful?”; an example of a physical health item was, “During the past 4 weeks, how much did pain interfere with your normal work?”

- **Sprtlty, SocSup, Engmgt, and DisEngmt** are four subscales from the Brief Coping with Problems Experienced Inventory [Carver, 1997]. The 28 items on this scale are presented on a 4-point scale ranging from 1 (*I usually do not do this at all*) to 4(*I usually do this a lot*). Higher scores indicate a respondents' tendency to engage in a particular strategy. Instructions were modified to ask how the female participants responded to recent experiences of racism and sexism as Black women. The four subscales included spirituality (religion, acceptance, planning), interconnectedness/social support (vent emotions, emotional support,instrumental social support), problem-oriented/engagement coping (active coping, humor, positive reinterpretation/positive reframing), and disengagement coping (behavioral disengagement, substance abuse, denial, self-blame, self-distraction).
- **GRICentlty:** The Multidimensional Inventory of Black Identity Centrality subscale [Sellers et al.] was modified to measure the intersection of racial and gender identity centrality. The scale included 10 items scaled from 1 (*strongly disagree*) to 7 (*strongly agree*). An example item was, “Being a *Black woman* is important to my self-image.” Higher scores indicated higher levels of gendered racial identity centrality.

6.4.2.1 Data Simulation

The *lavaan::simulateData* function was used. If you have taken psychometrics, you may recognize the code as one that creates latent variables from item-level data. In trying to be as authentic as possible, we retrieved factor loadings from psychometrically oriented articles that evaluated the measures [Nadal, 2011, Veit and Ware, 1983]. For all others we specified a factor loading of 0.80. We then approximated the *measurement model* by specifying the correlations between the latent variable. We sourced these from the correlation matrix from the research vignette [Lewis et al., 2017]. The process created data with multiple decimals and values that exceeded the boundaries of the variables. For example, in all scales there were negative values. Therefore, the final element of the simulation was a linear transformation that rescaled the variables back to the range described in the journal article and rounding the values to integer (i.e., with no decimal places).

#Entering the intercorrelations, means, and standard deviations from the journal article

```
Lewis_generating_model <- '
  ##measurement model
  GRMS  =~ .69*Ob1 + .69*Ob2 + .60*Ob3 + .59*Ob4 + .55*Ob5 + .55*Ob6 + .54*Ob7 + .50*Ob8
  MntlHlth  =~ .8*MH1 + .8*MH2 + .8*MH3 + .8*MH4 + .8*MH5 + .8*MH6
  PhysHlth  =~ .8*PhH1 + .8*PhH2 + .8*PhH3 + .8*PhH4 + .8*PhH5 + .8*PhH6
  Spirituality  =~ .8*Spirit1 + .8*Spirit2
  SocSupport  =~ .8*SocS1 + .8*SocS2
  Engagement  =~ .8*Eng1 + .8*Eng2
  Disengagement  =~ .8*dEng1 + .8*dEng2
  GRIC  =~ .8*Cntrlty1 + .8*Cntrlty2 + .8*Cntrlty3 + .8*Cntrlty4 + .8*Cntrlty5 + .8*Cntrlty6

  # Means
  GRMS ~ 1.99*1
  Spirituality ~2.82*1
  SocSupport ~ 2.48*1
```

```

Engagement ~ 2.32*1
Disengagement ~ 1.75*1
GRIC ~ 5.71*1
MntlHlth ~3.56*1 #Lewis et al used sums instead of means, I recast as means to facilitate comparison
PhysHlth ~ 3.51*1 #Lewis et al used sums instead of means, I recast as means to facilitate comparison

# Correlations
GRMS ~ 0.20*Spirituality
GRMS ~ 0.28*SocSupport
GRMS ~ 0.30*Engagement
GRMS ~ 0.41*Disengagement
GRMS ~ 0.19*GRIC
GRMS ~ -0.32*MntlHlth
GRMS ~ -0.18*PhysHlth

Spirituality ~ 0.49*SocSupport
Spirituality ~ 0.57*Engagement
Spirituality ~ 0.22*Disengagement
Spirituality ~ 0.12*GRIC
Spirituality ~ -0.06*MntlHlth
Spirituality ~ -0.13*PhysHlth

SocSupport ~ 0.46*Engagement
SocSupport ~ 0.26*Disengagement
SocSupport ~ 0.38*GRIC
SocSupport ~ -0.18*MntlHlth
SocSupport ~ -0.08*PhysHlth

Engagement ~ 0.37*Disengagement
Engagement ~ 0.08*GRIC
Engagement ~ -0.14*MntlHlth
Engagement ~ -0.06*PhysHlth

Disengagement ~ 0.05*GRIC
Disengagement ~ -0.54*MntlHlth
Disengagement ~ -0.28*PhysHlth

GRIC ~ -0.10*MntlHlth
GRIC ~ 0.14*PhysHlth

MntlHlth ~ 0.47*PhysHlth
'

set.seed(230925)
dfLewis <- lavaan::simulateData(model = Lewis_generating_model,
                                model.type = "sem",
                                meanstructure = T,

```

```

            sample.nobs=231,
            standardized=FALSE)

#used to retrieve column indices used in the rescaling script below
#col_index <- as.data.frame(colnames(dfLewis))

for(i in 1:ncol(dfLewis)){ # for loop to go through each column of the dataframe
  if(i >= 1 & i <= 25){ # apply only to GRMS variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 5))
  }
  if(i >= 26 & i <= 37){ # apply only to mental and physical health variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 6))
  }
  if(i >= 38 & i <= 45){ # apply only to coping variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 4))
  }
  if(i >= 46 & i <= 55){ # apply only to GRIC variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 7))
  }
}

#rounding to integers so that the data resembles that which was collected
library(tidyverse)
dfLewis <- dfLewis %>% round(0)

#quick check of my work
#psych::describe(dfLewis)

```

The script below allows you to store the simulated data as a file on your computer. This is optional – the entire lesson can be worked with the simulated data.

If you prefer the .rds format, use this script (remove the hashtags). The .rds format has the advantage of preserving any formatting of variables. A disadvantage is that you cannot open these files outside of the R environment.

Script to save the data to your computer as an .rds file.

```
#saveRDS(dfLewis, 'dfLewis.rds')
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
#dfLewis<- readRDS('dfLewis.rds')
```

If you prefer the .csv format (think “Excel lite”) use this script (remove the hashtags). An advantage of the .csv format is that you can open the data outside of the R environment. A disadvantage is that it may not retain any formatting of variables

Script to save the data to your computer as a .csv file.

```
# write.table(dfLewis, file = 'dfLewis.csv', sep = ',',
# col.names=TRUE, row.names=FALSE)
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
#dfLewis<- read.csv ('dfLewis.csv', header = TRUE)
```

6.4.3 Scrubbing, Scoring, and Data Diagnostics

Because the focus of this lesson is on complex mediation, we have used simulated data. If this were real, raw, data, it would be important to [scrub](#), [score](#), and conduct [data diagnostics](#) to evaluate the suitability of the data for the proposed analyses.

Because we are working with item level data we do need to score the scales used in the researcher's model. Because we are using simulated data and the authors already reverse coded any such items, we will omit that step.

As described in the [Scoring](#) chapter, we calculate mean scores of these variables by first creating concatenated lists of variable names. Next we apply the `sjstats::mean_n` function to obtain mean scores when a given percentage (we'll specify 80%) of variables are non-missing. Functionally, this would require the two-item variables (e.g., engagement coping and disengagement coping) to have non-missingness. We simulated a set of data that does not have missingness, none-the-less, this specification is useful in real-world settings.

Note that I am only scoring the variables used in the models demonstrated in this lesson. The remaining variables are available as practice options.

```
GRMS_vars <- c("Ob1", "Ob2", "Ob3", "Ob4", "Ob5", "Ob6", "Ob7", "Ob8",
               "Ob9", "Ob10", "Ma1", "Ma2", "Ma3", "Ma4", "Ma5", "Ma6", "Ma7", "St1",
               "St2", "St3", "St4", "St5", "An1", "An2", "An3")
Eng_vars <- c("Eng1", "Eng2")
dEng_vars <- c("dEng1", "dEng2")
MntlHlth_vars <- c("MH1", "MH2", "MH3", "MH4", "MH5", "MH6")

dfLewis$GRMS <- sjstats::mean_n(dfLewis[, GRMS_vars], 0.8)
dfLewis$Engmt <- sjstats::mean_n(dfLewis[, Eng_vars], 0.8)
dfLewis$DisEngmt <- sjstats::mean_n(dfLewis[, dEng_vars], 0.8)
dfLewis$MntlHlth <- sjstats::mean_n(dfLewis[, MntlHlth_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfLewis$GRMS <-
# sjstats::mean_n(dfLewis[, ..GRMS_vars], 0.80)
```

Now that we have scored our data, let's trim the variables to just those we need.

```
Lewis_df <- dplyr::select(dfLewis, GRMS, Engmt, DisEngmt, MntlHlth)
```

Let's check a table of means, standard deviations, and correlations to see if they align with the published article.

```
Lewis_table <- apaTables::apa.cor.table(Lewis_df, table.number = 1, show.sig.stars = TRUE,
  landscape = TRUE, filename = "Lewis_Corr.doc")
print(Lewis_table)
```

```
##
##
## Table 1
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##      Variable    M     SD   1          2          3
## 1. GRMS    2.56 0.72
##
## 2. Engmt    2.48 0.53 .52**       [.42, .61]
##
## 3. DisEngmt 2.48 0.52 .53**       .32**       [.43, .62]  [.20, .43]
##
## 4. MntlHlth 3.16 0.81 -.56**      -.23**      -.48**      [-.64, -.47] [-.35, -.11] [-.57, -.37]
##
##
## 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.
##
```

While they are not exact, they approximate the magnitude and patterns in the correlation matrix in the article [Lewis et al., 2017].

6.4.3.1 Specifying the *lavaan* model

The Lewis et al. article [2017] reports four mediation analyses, each repeated for mental and physical outcomes. Thus, their write-up reports eight simple mediation models. Graphically, their analyses were efficiently presented in a figure that looked (to me) like parallel mediation. Correspondingly, it made sense to me that we could try this in our research vignette. In the upcoming chapter on

conditional process analysis, we will work the moderated mediation that was a primary focus of their research.

Below is the model we will work. Specifically, we will evaluate whether gendered racial microaggressions impact mental health separately, thorough mediated paths of engagement and disengagement. We will also be able to see if the strength of those mediated paths are statistically, significantly, different from each other.

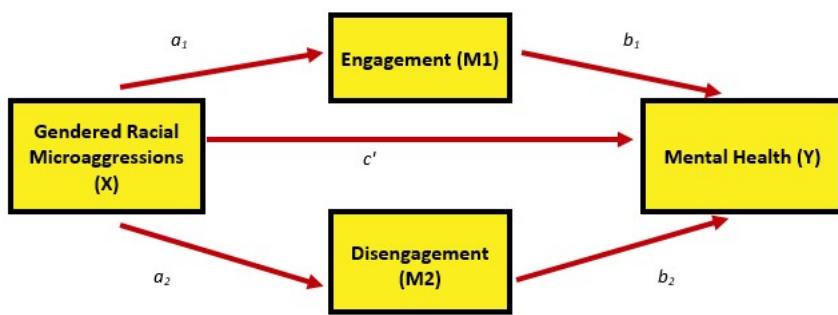


Figure 6.2: An image of the parallel mediation we will work

We can use the guidelines above to specify our model and then request summaries of the fit indices and parameter estimates.

```

parallel_Lewis <- "
  MntlHlth ~ b1*Engmt + b2*DisEngmt + c_p*GRMS
  Engmt ~ a1*GRMS
  DisEngmt ~ a2*GRMS

  indirect1 := a1 * b1
  indirect2 := a2 * b2
  contrast := indirect1 - indirect2
  total_indirects := indirect1 + indirect2
  total_c := c_p + (indirect1) + (indirect2)
  direct := c_p
"
set.seed(230925) #necessary for reproducible results because lavaan introduces randomness into the results
para_Lewis_fit <- lavaan::sem(parallel_Lewis, data = Lewis_df, se = "bootstrap",
  bootstrap = 1000, missing = "fiml") #holds the 'whole' result
pLewis_sum <- lavaan::summary(para_Lewis_fit, standardized = TRUE, rsq = T,
  fit = TRUE, ci = TRUE) #today, we really only need the R-squared from here
pLewis_ParEsts <- lavaan::parameterEstimates(para_Lewis_fit, boot.ci.type = "bca.simple",
  standardized = TRUE) #provides our estimates, se, p values for all the elements we specified
pLewis_sum
pLewis_ParEsts
  
```

6.4.3.2 Table and Figure

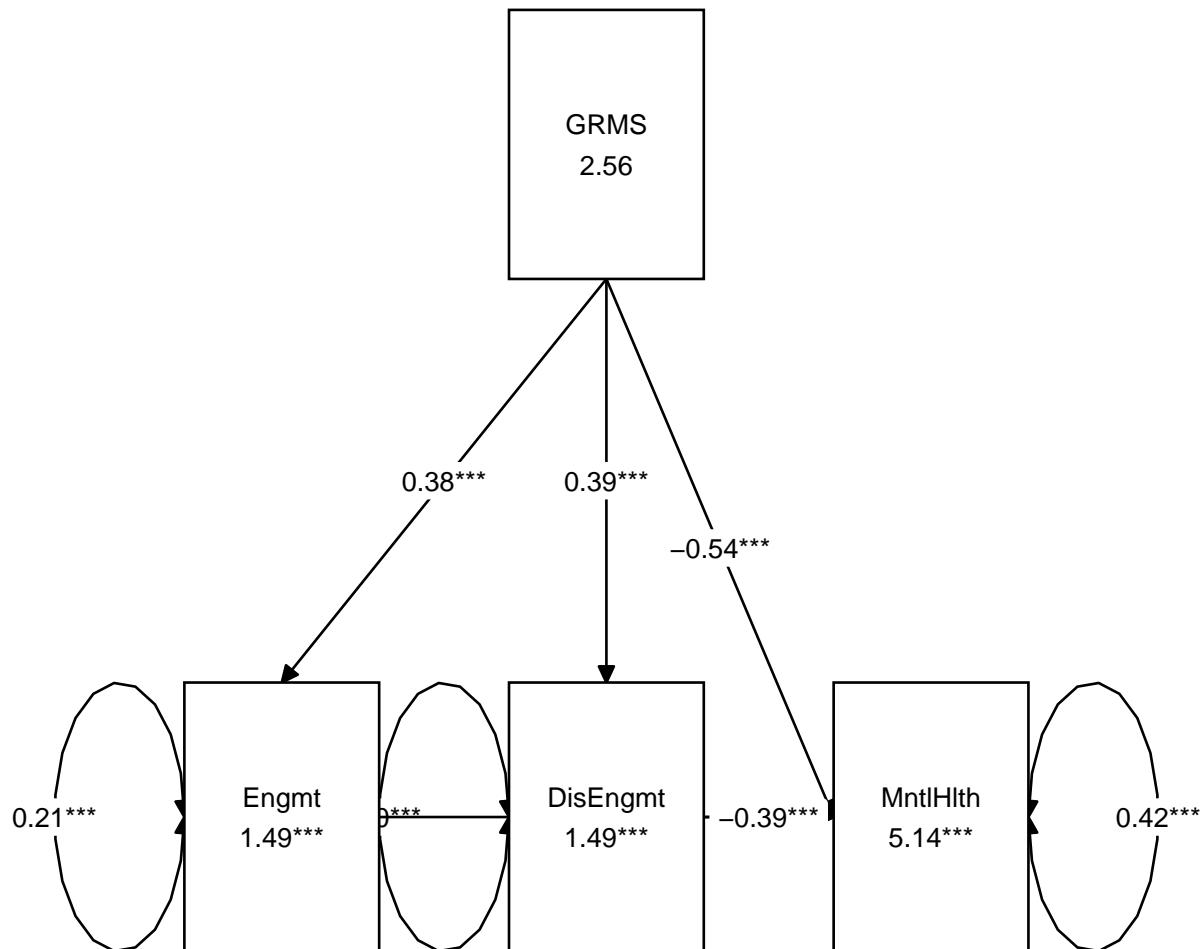
To assist in table preparation, it is possible to export the results to a .csv file that can be manipulated in Excel, Microsoft Word, or other program to prepare an APA style table.

```
write.csv(pLewis_ParEsts, file = "pLewis_ParEsts.csv")
```

We can use the package [tidySEM](#) to create a figure that includes the values on the path.

Here's what the base package gets us

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = para_Lewis_fit)
```



We can create model that communicates more intuitively with a little tinkering. First, let's retrieve the current "map" of the layout.

```
tidySEM::get_layout(para_Lewis_fit)
```

```
##      [,1]     [,2]     [,3]
## [1,] NA      "GRMS"    NA
## [2,] "Engmt"  "DisEngmt" "MntlHlth"
## attr(),"class")
## [1] "layout_matrix" "matrix"      "array"
```

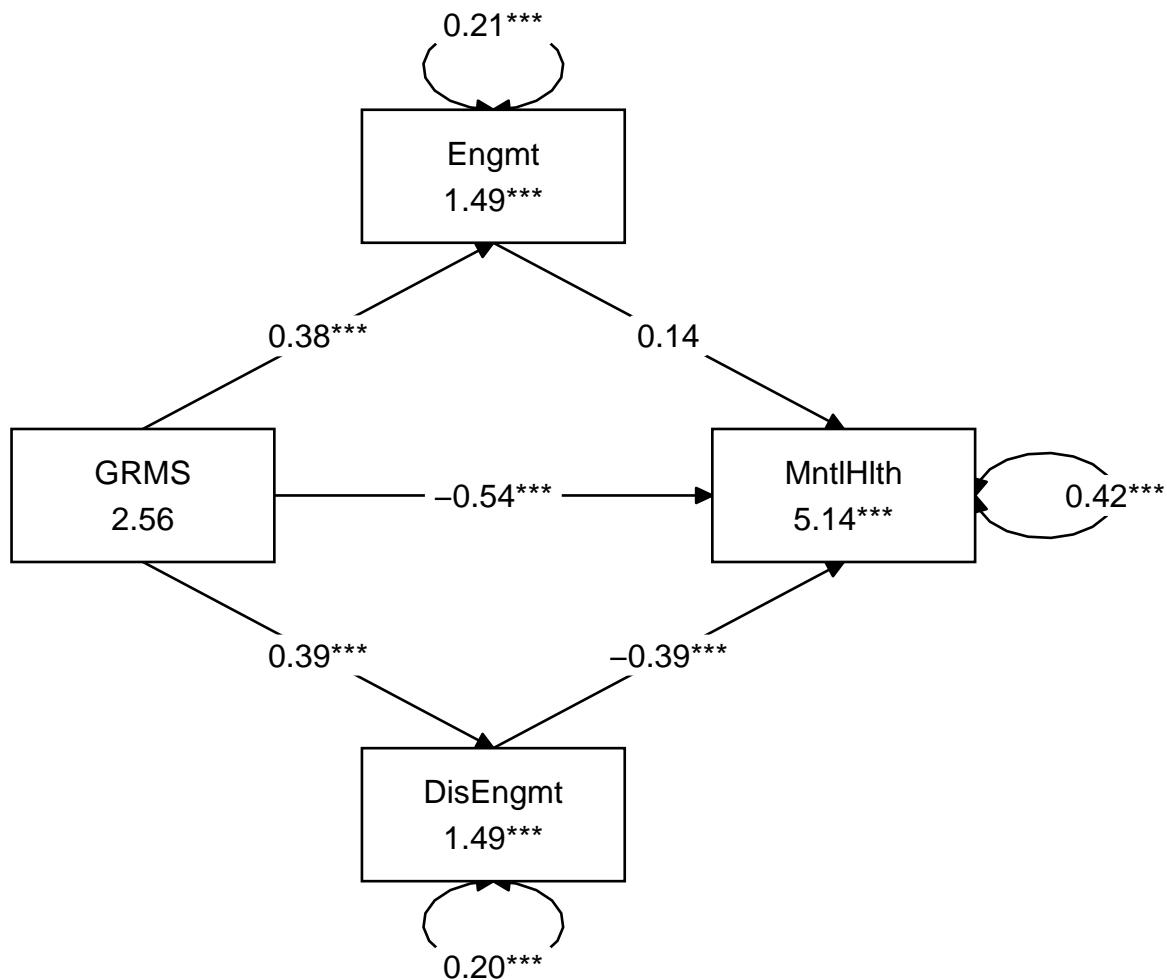
To create the figure I showed at the beginning of the chapter, we will want three rows and three columns.

```
pLewis_map <- tidySEM::get_layout("", "Engmt", "", "GRMS", "", "MntlHlth",
      "", "DisEngmt", "", rows = 3)
pLewis_map
```

```
##      [,1]     [,2]     [,3]
## [1,] ""      "Engmt"   ""
## [2,] "GRMS"  ""       "MntlHlth"
## [3,] ""      "DisEngmt" ""
## attr(),"class")
## [1] "layout_matrix" "matrix"      "array"
```

We can update our figure by supplying this new map and adjusting the object and text sizes.

```
tidySEM::graph_sem(para_Lewis_fit, layout = pLewis_map, rect_width = 1.5,
rect_height = 1.25, spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



Now let's make a table.

Table 2

Model Coefficients Assessing Engagement and Disengagement Coping as Parallel Mediators Between Predicting Mental Health from Gendered Racial Microaggressions

Predictor	B	SE _B	p	R ²
Engagement coping (M1)				.27
Constant	1.494	0.111	<0.001	
GRMS (a_1)	0.384	0.042	<0.001	

Disengagement coping (M2)				.28
Constant	1.490	0.100	<0.001	
GRMS (a_2)	0.386	0.038	<0.001	
Mental Health (DV)				.37
Constant	5.141	0.226	<0.001	
Engagement (b_1)	0.144	0.089	0.106	
Disengagement (b_2)	-0.391	0.089	<0.001	
GRMS (c')	-0.535	0.076	<0.001	
Summary of Effects	<i>B</i>	<i>SE_B</i>	<i>p</i>	95% CI
Total	-0.631	0.058	<0.001	-0.739, -0.505
Indirect 1 ($a_1 * a_2$)	0.055	0.036	0.121	-0.003, 0.140
Indirect 2 ($b_1 * b_2$)	-0.151	0.038	<0.001	-0.235, -0.085
Total indirects	-0.096	0.051	0.059	-0.193, 0.007
Contrast (Ind1 - Ind2)	0.206	0.054	<0.001	0.109, 0.338

Note. GRMS = gendered racial microaggressions. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

- The model accounts for 37% of the variance in predicting mental health outcomes.
- The total effect of GRMS on mental health is -0.631 ($p < 0.001$) is negative and statistically significant. That is, gendered racial microaggressions have a statistically significant negative effect on mental health.
- The direct effect of GRMS on mental health is -0.535 ($p < 0.001$); while this is lower than the total effect, it remains statistically significant.
 - Using Baron and Kenny's [1986] causal steps logic, the fact that the direct effect does not decrease in a statistically significant manner does not provide helpful, logical support for mediation. According to Hayes [2022b] this difference is not necessary. That is, a statistically significant indirect effect can stand on its own.
- Indirect effect #1 ($a_1 \times b_1$ or GRMS through engagement coping) is 0.055 ($p = 0.121, CI_{95}[-0.003, 0.140]$) and not statistically significant. Because they can be inconsistent with the p values, we should always check the confidence intervals to see if they pass through zero. In this case they do.
- Indirect effect #2 ($a_2 \times b_2$, or GRMS through disengagement to coping) is -0.151 ($p < 0.001, CI_{95}[-0.235, -0.085]$). The p value is significant and the 95% confidence interval does not pass through zero. Thus, gendered racial microaggressions lead to greater disengagement (a_2). In turn, disengagement has negative effects on mental health (b_2).
- The total indirect effect (i.e., sum of all specific indirect effects) (-0.096, $p = 0.059$) is not statistically significant.

- We examine the contrast to see if the indirect effects statistically significantly different from each other: $B = 0.206, p < 0.001$. They are. This is not surprising since the path mediated by engagement was not statistically significant but the path mediated by disengagement was statistically significant.

6.4.3.3 APA Style Writeup

Hayes [Hayes, 2022a] provides helpful guidelines for presenting statistical results. Here is a summary of his recommendations.

- Pack as much statistical info as possible into a table(s) or figure(s).
- Use statistics in the text as punctuation; avoid redundancy in text and table.
- Avoid using abbreviations for variables in the text itself; rather focus on the construct names rather than their shorthand
- Avoid focusing on what you hypothesized (e.g., avoid, “Results supported/did not support hypothesis A1”) and instead focus on what you found. The reader is more interested in the results, not your forecasts.
- Hayes prefers reporting unstandardized metrics because they map onto the measurement scales used in the study. He believes this is especially important when dichotomous variables are used.
- There is “no harm” in reporting hypothesis tests and CIs for the a and b paths, but whether/not these paths are statistically significant does not determine the significance of the indirect effect.
- Be precise with language:
- OK: X exerts an effect on Y directly and/or indirectly through M.
- Not OK: the indirect effect of M
- Report direct and indirect effects and their corresponding inferential tests
- Hayes argues that a statistically significant indirect effect is, in fact statistic. He dislikes narration of the Baron and Kenny [1986] process and steps.

Here's my attempt to write up the simulated data from the Lewis et al. [2017] article.

Method

Data Analysis

Parallel multiple mediation is appropriate when testing the influence of an independent variable (X) on the dependent variable (Y) directly, as well as indirectly through two or more mediators. A condition of parallel multiple mediation is that no mediator causally influences another [Hayes, 2022b]. Using data simulated from Lewis et al. [2017] we utilized parallel multiple mediation analysis to test the influence of gendered racial microaggressions (X, GRMS) on mental health outcomes (Y, MntlHlth) directly as well as indirectly through the mediators engagement coping (M1, Engmt) and disengaged coping (M2, DisEngmt). Using the *lavaan* (v. 0.6-16) package in R we followed the procedures outlined in Hayes [2022b] by analyzing the strength and significance of four sets of effects: specific indirect, the total indirect, the direct, and total.

Results

Preliminary Analyses Descriptive statistics were computed, and all variables were assessed for skewness and kurtosis. *More narration, here.* A summary of descriptive statistics and a correlation matrix for the study is provided in Table 2. These bivariate relations provide evidence to support the test of mediation analysis.

Parallel Multiple Mediation Analysis A model of parallel mediation examined the degree to which engagement and disengagement coping strategies mediated the relation of gendered racial microaggressions on mental health outcomes in Black women. Hayes [2022b] recommended this strategy over simple mediation models because it allows for all mediators to be examined, simultaneously. The resultant direct and indirect values for each path account for other mediation paths. Using the *lavaan* (v. 0.6-17) package in R, coefficients for specific indirect, total indirect, direct, and total were computed. Path coefficients refer to regression weights, or slopes, of the expected changes in the dependent variable given a unit change in the independent variables.

Results (depicted in Figure 2 and presented in Table 3) suggest that 37% of the variance in mental health outcomes is accounted for by the model. The indirect effect predicting mental health from gendered racial microaggressions via engagement coping was not statistically significant $*B = 0.055, SE = 0.036, p = 0.121, CI95[-0.003, 0.140]$). Looking at the individual paths we see that a_1 was positive and statistically significant (GRMS leads to increased engagement coping), but the subsequent link, b_1 (engagement to mental health) was not. The indirect effect predicting mental health from gendered racial microaggressions through disengagement to coping was statistically significant $B = -0.151, SE = 0.038, p < 0.001, CI95[-0.235, -0.085]$). In this case, gendered racial microaggressions led to greater disengagement coping (a_2). In turn, disengagement coping had negative effects on mental health (b_2). Curiously, the total indirect effect (i.e., the sum of the specific indirect effects was not statistically significant. It is possible that the positive and negative valences of the indirect effects “cancelled each other out.” A pairwise comparison of the specific indirect effects indicated that the strength of the effects were statistically significantly different from each other. Given that the path through engagement coping was not significant, but the path through disengagement coping was, this statistically significant difference is not surprising.

Hints for Writing Method/Results Sections

- When you find an article you like, make note of it and put it in a very special folder. In recent years, I have learned to rely on full-text versions stored in my Zotero app.
- Once you know your method (measure, statistic, etc.) begin collecting others articles that are similar to it. To write results sections I will often reference multiple articles.

- When it iss time to write have all these resources handy and use them as guides/models.
- Put as much info as possible in the table. Become a table-pro. That is, learn how to merge/split cells, use borders/shading, the decimal tab, and so forth. Don’t make the borders disappear until the last thing you do before submitting. This is because you ALWAYS have to update your tables and seeing the borders makes it easier.

6.5 Serial Multiple Mediator Model

Recall that one of the conditions of the *parallel mediator model* was that “no mediator causally influences another.”

Regarding these correlated mediators [Hayes, 2022b]:

- Typically, two or more mediators that are causally located between X and Y will be correlated
 - if for no other reason than that they share a common cause (X).
- Estimating the partial correlation between two mediators after controlling for X is one way to examine whether all of their association is accounted for by this common cause.
- *Partial correlation* is the Pearson correlation between the residuals from a model estimating Y from a set of covariates, and the residuals from a model estimating X from the same set of covariates.
- Partial correlations allow the assessment of their association, independent of what they have in common with the covariates that were regressed onto Y and X, separately.
- If two (or more) mediators remain correlated after adjusting for X, then
 - the correlation is *spurious*, they share another (unmodeled) common cause.
 - the remaining association is *epiphenomenal*. That is, a proposed mediator could be related to an outcome not because it causes the outcome, but because it is correlated with another variable that is causally influencing the outcome. This is a noncausal alternative explanation for an association. Also, many things correlated with the cause of Y will also tend to be correlated with X, but it doesn't make all those things cause Y
 - *or one mediator causally affects another*

The goal of a serial multiple mediator model is to investigate the direct and indirect effects of X on Y while modeling a process in which X causes M1, which in turn causes M2, and so forth, concluding with Y as the final consequent.

As before, we will calculate:

- *Direct effect, c'* : the estimated difference in Y between two cases that differ by one unit on X but who are equal on all mediators in the model.
- *Specific indirect effects, $a_1b_1, a_2b_2, a_3b_3, \text{etc.}$* : constructed by multiplying the regression weights corresponding to each step in an indirect pathway; interpreted as the estimated difference in Y between two cases that differ by one unit on X through the causal sequence from X to mediator(s) to Y.
- *Total indirect effect of X*: sum of all specific indirect effects
- *Total effect of X*: the total indirect effect of X plus the direct effect of X; can also be estimated by regressing Y from X only.
- *Pairwise comparisons (contrasts) between indirect effects* (i.e., is one indirect effect stronger than another)

6.5.1 We stick with the Lewis et al. [2017] example, but modify it.

Our parallel multiple mediator model of gendered racial microaggressions on mental health through engagement and disengagement coping strategies assumed no causal association between the mediators. Noting the statistically significant correlation between engagement and disengagement, what if engagement influenced disengagement, which, in turn influenced mental health.

If this is our goal (image), how many direct and indirect effects are contained in this model? Using the same processes as before, let's plan our model:

- We add a path predicting disengagement from engagement, and label it with a d_{21}

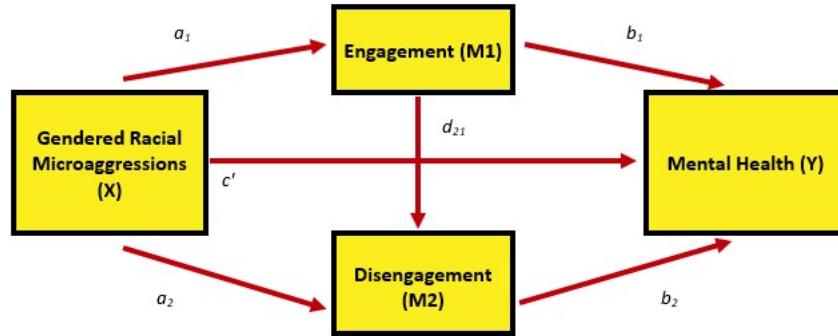


Figure 6.3: An image of the serial mediation we will work

- Regarding the notation, it makes sense that we use a d to designate a new type of path; I don't know why we use a subscript of 21
- We specify a third indirect path that multiplies those 3 paths (a_1, d_{21}, b_2) together
- We add a third contrast so that we get all the combinations of indirect comparisons: 1-2, 1-3 2-3
- We update our total_indirects calculation to include indirect#3
- We update our total_c calculation to include indirect#3

6.5.2 Specify the *lavaan* model

```

serial_Lewis <- "
  MntlHlth ~ b1*Engmt + b2*DisEngmt + c_p*GRMS
  Engmt ~ a1*GRMS
  DisEngmt ~ a2*GRMS
  DisEngmt ~ d21*Engmt

  indirect1 := a1 * b1
  indirect2 := a2 * b2
  indirect3 := a1 * d21 * b2
  contrast1 := indirect1 - indirect2
  contrast2 := indirect1 - indirect3
  contrast3 := indirect2 - indirect3
  total_indirects := indirect1 + indirect2 + indirect3
  total_c := c_p + indirect1 + indirect2 + indirect3
  direct := c_p
"
set.seed(230925) #necessary for reproducible results because lavaan introduces randomness into
serial_Lewis_fit <- lavaan::sem(serial_Lewis, data = Lewis_df, se = "bootstrap",
  missing = "fiml", bootstrap = 1000)
sLewis_sum <- lavaan::summary(serial_Lewis_fit, standardized = TRUE, rsq = T,
  fit = TRUE, ci = TRUE)
sLewis_ParEsts <- lavaan::parameterEstimates(serial_Lewis_fit, boot.ci.type = "bca.simple",
  
```

```
standardized = TRUE)

sLewis_sum
sLewis_ParEsts
```

6.5.2.1 Table and Figure

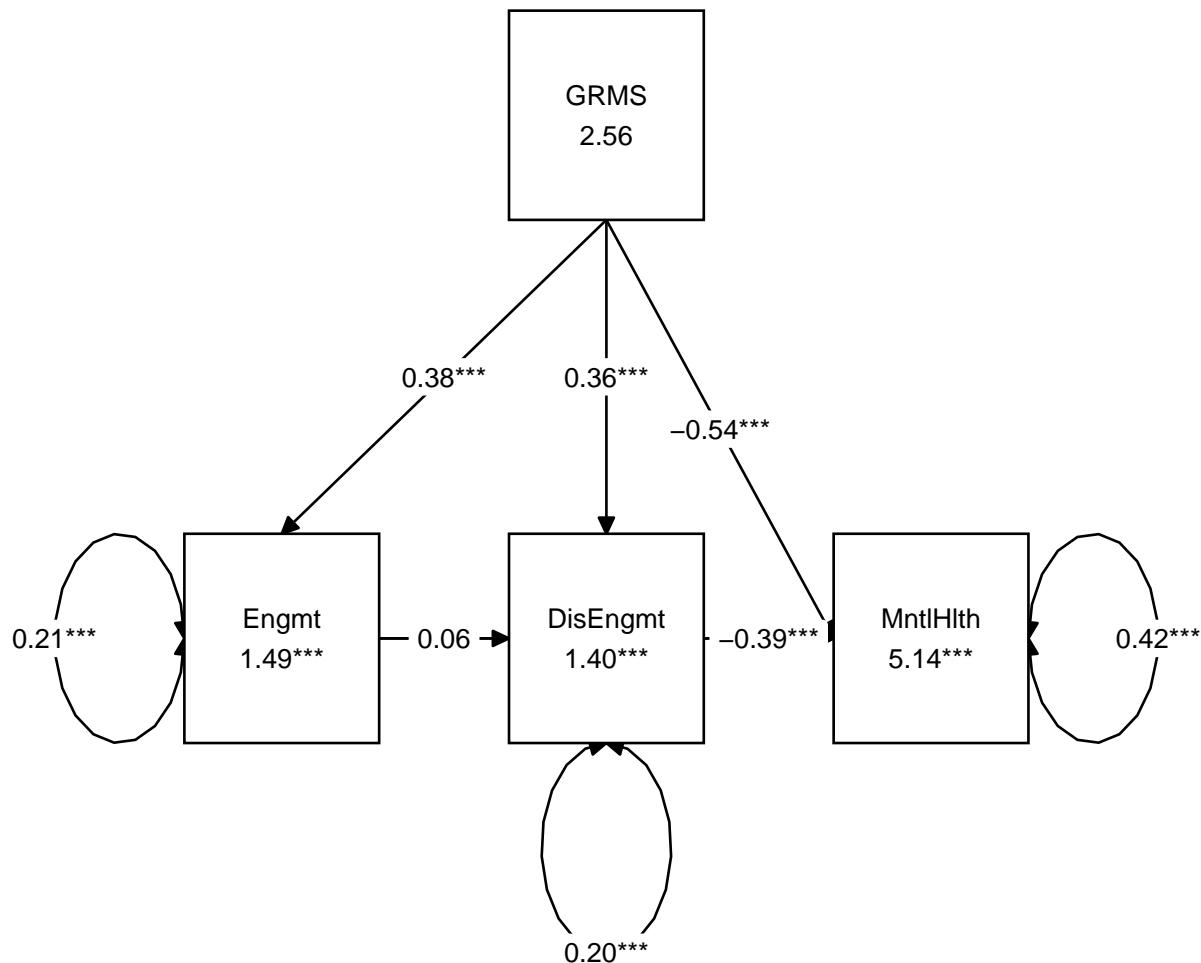
To assist in table preparation, it is possible to export the results to a .csv file that can be manipulated in Excel, Microsoft Word, or other program to prepare an APA style table.

```
write.csv(sLewis_ParEsts, file = "sLewis_ParEsts.csv")
```

We can use the package [tidySEM](#) to create a figure that includes the values on the path.

Here's what the base package gets us

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = serial_Lewis_fit)
```



We can create model that communicates more intuitively with a little tinkering. First, let's retrieve the current "map" of the layout.

```
tidySEM::get_layout(serial_Lewis_fit)
```

```
##      [,1]     [,2]     [,3]
## [1,] NA      "GRMS"    NA
## [2,] "Engmt"  "DisEngmt" "MntlHlth"
## attr(,"class")
## [1] "layout_matrix" "matrix"       "array"
```

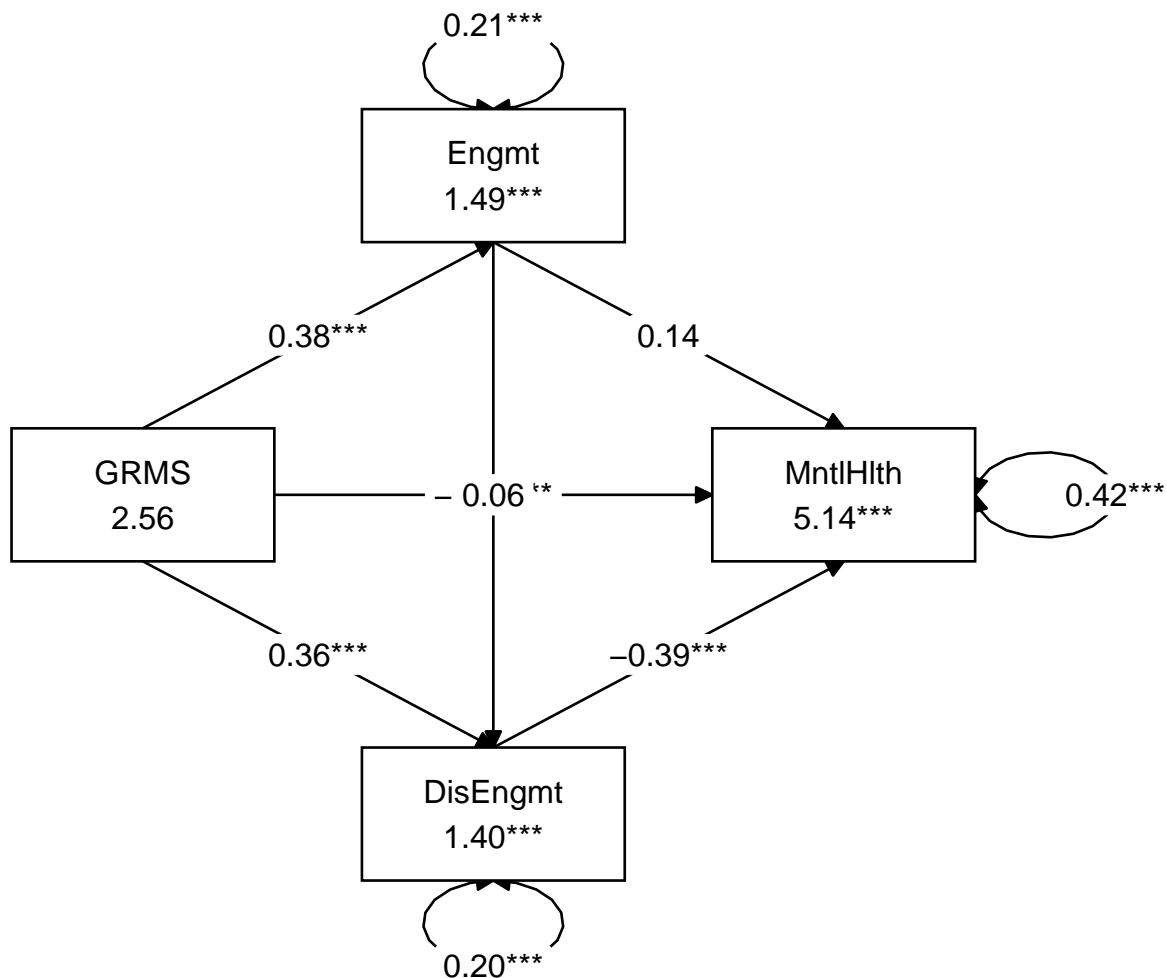
To create the figure I showed at the beginning of the chapter, we will want three rows and three columns.

```
sLewis_map <- tidySEM::get_layout("", "Engmt", "", "GRMS", "", "MntlHlth",
  "", "DisEngmt", "", rows = 3)
sLewis_map
```

```
##      [,1]   [,2]      [,3]
## [1,] ""    "Engmt"    ""
## [2,] "GRMS" ""    "MntlHlth"
## [3,] ""    "DisEngmt" ""
## attr(,"class")
## [1] "layout_matrix" "matrix"      "array"
```

We can update our figure by supplying this new map and adjusting the object and text sizes.

```
tidySEM::graph_sem(serial_Lewis_fit, layout = sLewis_map, rect_width = 1.5,
rect_height = 1.25, spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



Now let's make a table.

Table 4

 Model Coefficients Assessing Engagement and Disengagement Coping in a Model of Serial Mediation Predicting Mental Health from Gendered Racial Microaggressions

Predictor	<i>B</i>	<i>SE_B</i>	<i>p</i>	<i>R</i> ²
Engagement coping (M1)				.27
Constant	1.494	0.111	<0.001	
GRMS (a_1)	0.384	0.042	<0.001	
Disengagement coping (M2)				.29
Constant	1.400	0.128	<0.001	
GRMS (a_2)	0.363	0.046	<0.001	
Engagement (d_{21})	0.061	0.059	0.304	
Mental Health (DV)				.37
Constant	5.141	0.226	<0.001	
Engagement (b_1)	0.144	0.089	0.106	
Disengagement (b_2)	-0.391	0.089	<0.001	
GRMS (c')	-0.535	0.076	<0.001	
Effects	<i>B</i>	<i>SE_B</i>	<i>p</i>	95% CI
Total effect	-0.631	0.059	<0.001	-0.739, -0.505
Indirect 1 ($a_1 * a_2$)	0.055	0.036	0.121	-0.003, 0.140
Indirect 2 ($b_1 * b_2$)	-0.142	0.038	<0.001	-0.227, -0.078
Indirect 3 ($b_1 * d_{21} * b_2$)	-0.009	0.009	0.339	-0.032, 0.007
Total indirects	-0.096	0.051	0.059	-0.193, 0.007
Contrast1 (Ind1 - Ind2)	0.197	0.053	<0.001	0.106, 0.318
Contrast2 (Ind1 - Ind3)	0.064	0.039	0.095	0.002, 0.156
Contrast3 (Ind2 - Ind3)	-0.133	0.039	0.001	-0.230, -0.066

Note. GRMS = gendered racial microaggressions. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

Working through the data, we should be able to find these items:

- The model accounts for 37% of the variance in predicting mental health outcomes.

- The total effect of GRMS (X) on mental health (Y) is $-0.631, (p < .001)$; it is negative and statistically significant.
- The direct effect of GRMS (X) on mental health (Y) ($-0.535, p < 0.001$) is still negative. Although someone lower in magnitude, it is still statistically significant. While inconsistent with the Baron and Kenny [1986] logic of mediation, Hayes [Hayes, 2022b] argues that a statistically significant indirect effect can stand on its own.
- Indirect effect #1 ($a_1 \times b_1$ or GRMS through engagement coping to mental health) is $B = 0.055, p = 0.121$. As in the parallel mediation, p is $> .05$ and the 95% CIs pass through zero ($-0.003, 0.140$). Examining the individual paths, there is a statistically significant relationship from GRMS to engagement, but not from engagement to mental health.
- Indirect effect #2 ($a_2 \times b_2$, or GRMS through disengagement coping to mental health, is $B = -0.142, p < 0.001, 95CI(-0.227, -0.078)$). Each of the paths is statistically significant from zero and so is the indirect effect.
- Indirect effect #3 ($a_2 \times d_{21} \times b_2$; GRMS through engagement coping through disengagement coping to mental health) is $B = -0.009, p = 0.339, 95CI(-0.032, 0.007)$. This indirect effect involves a_1 (GRMS to engagement) and b_2 which are significant. However, the path from engagement coping to disengagement coping is not significant.
- Total indirect: $B = -0.096, p = 0.059$ is the sum of all specific indirect effects and is not statistically significant. The positive and negative indirects likely cancel each other out.
- With **contrasts** we ask: Are the indirect effects statistically significantly different from each other?
 - Contrast 1 (indirect 1 v 2): $B = 0.197, p < 0.001$, yes
 - Contrast 2 (indirect 1 v 3): $B = 0.064, p = 0.095$, no
 - Contrast 3 (indirect 2 v 3): $B = -0.133, p = 0.001p$, yes
 - This formal test of contrasts is an important one. It is not ok to infer that effects are statistically significantly different than each other on the basis of their estimates or p values. The formal test allows us to claim (with justification) that there are statistically significant differences between indirect effects 1 and 2; and 2 and 3.

6.5.3 APA Style Writeup

Method

Data Analysis Serial multiple mediation is appropriate when testing the influence of an independent variable (X) on the dependent variable (Y) directly, as well as indirectly through two or more mediators (M) and there is reason to hypothesize that variables that are causally prior in the model affect all variables later in the causal sequence [Hayes, 2022b]. We utilized serial multiple mediation analysis to test the influence of gendered racial microaggressions (X, GRMS) on mental health (Y, MntlHlth) directly as well as indirectly through the mediators engagement coping (M1, Engmt) and disengagement coping (M2, DisEngmt). Moreover, we hypothesized a causal linkage between the engagement coping mediator to the disengagement coping mediator such that a third specific indirect effect began with GRMS (X) through engagement coping (M1) through disengagement coping (M2) to mental health (Y). Using the *lavaan* (v. 0.6-16) package in R we followed the procedures outlined in Hayes [2022b] by analyzing the strength and significance of four sets of effects: specific indirect, the total indirect, the direct, and total. Bootstrap analysis, a nonparametric sampling procedure, was used to test the significance of the indirect effects.

Hayes would likely recommend that we say this with fewer acronyms and more words/story.

Results Preliminary Analyses Descriptive statistics were computed, and all variables were assessed univariate normality. *You would give your results regarding skew, kurtosis, Shapiro Wilks', here. If relevant, you could also describe multivariate normality.* A summary of descriptive statistics and a correlation matrix for the study is provided in Table 1. These bivariate relations provide evidence to support the test of mediation analysis.

Serial Multiple Mediation Analysis A model of serial multiple mediation was analyzed examining the degree to which engagement and disengagement coping mediated the relationship between gendered racial microaggressions and mental health outcomes. Hayes [2022b] recommended this strategy over simple mediation models because it allows for all mediators to be examined, simultaneously and allows the testing of the seriated effect of prior mediators onto subsequent ones. Using the *lavaan* (v. 0.6-16) package in R, coefficients for specific indirect, total indirect, direct, and total were computed. Path coefficients refer to regression weights, or slopes, of the expected changes in the dependent variable given a unit change in the independent variables.

Results (depicted in Figure # and presented in Table #) suggest that 37% of the variance in behavioral intentions is accounted for by the three variables in the model. Two of the specific indirect effects were significant and were statistically significantly different from each other. Specifically, the effect of gendered racial microaggressions through disengagement coping to mental health ($B = -0.142, SE = 0.038, p < .001, 95CI[-0.227, -0.078]$) was stronger than the indirect effect from gendered racial microaggressions through engagement coping through disengagement coping to mental health ($B = 0.055, SE = 0.036, p = 0.121, 95CI[-0.003, 0.140]$). Interpreting the results suggests that, mental health outcomes are negatively impacted by gendered racial microaggressions direct and indirectly through disengagement coping. It is this latter path that has the greatest impact.

Note: In a manner consistent with the Lewis et al. [2017] article, the APA Results section can be fairly short. This is especially true when a well-organized table presents the results. In fact, I could have left all the numbers out of this except for the R^2 (because it was not reported in the table).

6.6 STAY TUNED

A section on power analysis is planned and coming soon! My apologies that it's not quite Ready.

6.7 Troubleshooting and FAQs

An indirect effect that was (seemingly) significant in a simple (single) mediation disappears when additional mediators are added.

- Correlated mediators (e.g., multicollinearity) is a likely possibility.
- Which is correct? Maybe both...

A total effect was not significant, but there is one or more statistically significant specific indirect effect

- Recall that a total effect equals the sum of direct and indirect effects. If one specific indirect effect is positive and another is negative, this could account for the NS total effect.

- If the direct effect is NS, but the indirect effects are significant, this might render the total effect NS.
- The indirect effects might operate differently in subpopulations (males, females).

Your editor/peer reviewer/dissertation chair-or-committee member may insist that you do this the Baron & Kenny way (aka “the causal steps approach”).

- Hayes [Hayes, 2022a] provides compelling arguments for how to justify your (I believe correct) decision to just use the PROCESS (aka, bootstrapped, bias corrected, CIs)approach.
- My favorite line in his text reads, ” (the Baron and Kenny way)...is still being taught and recommended by researchers who don’t follow the methodology literature.”

How can I extend a mediation (only) model to include multiple Xs, Ys, or COVs?

- There is fabulous, fabulous narration and syntax for doing all of this in Hayes text. Of course his mechanics are in PROCESS, but *lavaan* is easy to use by just “drawing more paths” via the syntax. We’ll get more practice as we go along.

What about effect sizes? Shouldn’t we be including/reporting them?

- Yes! The closest thing we have reported to an effect size is R^2 , which assess proportion of variance accounted for in the M and Y variables.
- In PROCESS and path analysis this is still emerging. Hayes chapter 4 presents a handful of options for effect sizes beyond R^2 .

6.8 Practice Problems

The three problems described below are designed to be grow in this series of chapters that begins with simple mediation and progresses through complex mediation, moderated moderation, and conditional process analysis. The goal of this assignment is to conduct a complex (e.g., parallel or serial) mediation.

I recommend that you select a dataset that includes at least four variables. If you are new to this topic, you may wish to select variables that are all continuously scaled. The IV and moderator (in subsequent chapters) *could* be categorical (if they are dichotomous, please use 0/1 coding; if they have more than one category it is best if they are ordered). You will likely encounter challenges that were not covered in this chapter. Search for and try out solutions, knowing that there are multiple paths through the analysis.

The suggested practice problem for this chapter is to conduct a parallel or serial mediation (or both).

6.8.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed

If conducting a parallel or serial mediation feels a bit overwhelming, simply change the random seed in the data simulation, then rework one of the chapter problems (i.e., parallel or serial mediation). This should provide minor changes to the data (maybe in the second or third decimal point), but the results will likely be very similar.

6.8.2 Problem #2: Rework the research vignette, but swap one or more variables

Conduct the complex mediation (parallel or serial) using the simulated data provided in this chapter, but swap out one or more of the variables. This could mean changing roles for the variables that were the focus of the chapter, or substituting one or more variables for those in the simulated data but not modeled in the chapter.

6.8.3 Problem #3: Use other data that is available to you

To conduct the parallel or serial mediation, use data for which you have permission and access. This could be IRB approved data you have collected or from your lab; data you simulate from a published article; data from an open science repository; or data from other chapters (or the “homeworked example”) in this OER.

6.8.4 Grading Rubric

Assignment Component		
1. Assign each variable to the X, Y, M1, and M2 roles	5	_____
4. Use tidySEM to create a figure that represents your results	5	_____
5. Create a table that includes a summary of the effects (indirect, direct, total, total indirect) as well as contrasts	5	_____
6. Represent your work in an APA-style write-up	5	_____
7. Explanation to grader	5	_____
8. Be able to hand-calculate the indirect, direct, and total effects from the a, b, & c' paths	5	_____
Totals	40	_____

6.9 Homeworked Example

Screencast Link

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to conduct a complex (i.e., parallel or serial) mediation.

Assign each variable to the X, Y, M1, and M2 roles

X = Centering: explicit recentering (0 = precentered; 1 = recentered) M1 = TradPed: traditional pedagogy (continuously scaled with higher scores being more favorable) M2 = SRPed: socially responsive pedagogy (continuously scaled with higher scores being more favorable) Y = Valued: valued by me (continuously scaled with higher scores being more favorable)

In this *parallel mediation*, I am hypothesizing that the perceived course value to the students is predicted by intentional recentering through their assessments of traditional and socially responsive pedagogy.

It helps me to make a quick sketch:

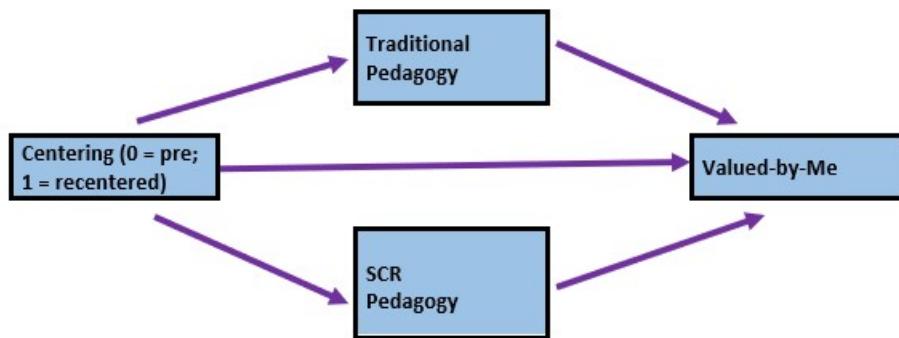


Figure 6.4: An image of the parallel mediation model for the homeworked example.

Import the data and format the variables in the model

```
raw <- readRDS("ReC.rds")
```

The approach we are taking to complex mediation does not allow dependency in the data. Therefore, we will include only those who took the multivariate class (i.e., excluding responses for the ANOVA and psychometrics courses).

```
raw <- (dplyr::filter(raw, Course == "Multivariate"))
```

I need to score the TradPed, SRPed, and Valued variables

```
TradPed_vars <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback",
                  "ClearOrganization", "ClearPresentation")
raw$TradPed <- sjstats::mean_n(raw[, .TradPed_vars], 0.75)
```

```
Valued_vars <- c("Val0bjectives", "IncrUnderstanding", "IncrInterest")
raw$Valued <- sjstats::mean_n(raw[, ..Valued_vars], 0.75)

SRPed_vars <- c("InclusvClassrm", "EquitableEval", "MultPerspectives",
  "DEIintegration")
raw$SRPed <- sjstats::mean_n(raw[, ..SRPed_vars], 0.75)
```

I will create a babydf.

```
babydf <- dplyr::select(raw, Centering, TradPed, Valued, SRPed)
```

Let's check the structure of the variables:

```
str(babydf)
```

```
## Classes 'data.table' and 'data.frame':  84 obs. of  4 variables:
## $ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ TradPed   : num  3.8 5 4.8 4 4.2 3 5 4.6 4 4.8 ...
## $ Valued    : num  4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
## $ SRPed     : num  4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

At this point, these my only inclusion/exclusion criteria. I can determine how many students (who consented) completed any portion of the survey.

Specify and run the lavaan model

```
ReCpMed <- "
  Valued ~ b1*TradPed + b2*SRPed + c_p*Centering
  TradPed ~ a1*Centering
  SRPed ~ a2*Centering

  indirect1 := a1 * b1
  indirect2 := a2 * b2
  contrast := indirect1 - indirect2
  total_indirects := indirect1 + indirect2
  total_c      := c_p + (indirect1) + (indirect2)
  direct      := c_p

  "

set.seed(230916) #needed for reproducible results since lavaan includes randomness in its est
ReCpMedfit <- lavaan::sem(ReCpMed, data = babydf, se = "bootstrap", missing = "fiml")
ReCpMedsummary <- lavaan::summary(ReCpMedfit, standardized = T, rsq = T,
```

```

    fit = TRUE, ci = TRUE)
ReC_pMedParamEsts <- lavaan::parameterEstimates(ReCpMedfit, boot.ci.type = "bca.simple",
                                              standardized = TRUE)
ReCpMedsummary

## lavaan 0.6.16 ended normally after 23 iterations
##
## Estimator                               ML
## Optimization method                    NLMINB
## Number of model parameters            11
##
## Number of observations                84
## Number of missing patterns            3
##
## Model Test User Model:
## 
## Test statistic                         54.059
## Degrees of freedom                     1
## P-value (Chi-square)                  0.000
##
## Model Test Baseline Model:
## 
## Test statistic                         145.642
## Degrees of freedom                     6
## P-value                                0.000
##
## User Model versus Baseline Model:
## 
## Comparative Fit Index (CFI)           0.620
## Tucker-Lewis Index (TLI)              -1.280
##
## Robust Comparative Fit Index (CFI)    0.613
## Robust Tucker-Lewis Index (TLI)       -1.323
##
## Loglikelihood and Information Criteria:
## 
## Loglikelihood user model (H0)        -202.536
## Loglikelihood unrestricted model (H1) -175.506
##
## Akaike (AIC)                          427.071
## Bayesian (BIC)                        453.810
## Sample-size adjusted Bayesian (SABIC)  419.110
##
## Root Mean Square Error of Approximation:
## 
## RMSEA                                0.795
## 90 Percent confidence interval - lower 0.623

```

```

## 90 Percent confidence interval - upper          0.982
## P-value H_0: RMSEA <= 0.050                  0.000
## P-value H_0: RMSEA >= 0.080                  1.000
##
## Robust RMSEA                                0.815
## 90 Percent confidence interval - lower       0.641
## 90 Percent confidence interval - upper       1.004
## P-value H_0: Robust RMSEA <= 0.050         0.000
## P-value H_0: Robust RMSEA >= 0.080         1.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                                         0.217
##
## Parameter Estimates:
##
## Standard errors                               Bootstrap
## Number of requested bootstrap draws          1000
## Number of successful bootstrap draws         1000
##
## Regressions:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## Valued ~
##   TradPed (b1)    0.686   0.131   5.217   0.000   0.451   0.955
##   SRPed   (b2)    0.119   0.146   0.816   0.414  -0.193   0.400
##   Centerng (c_p)  0.015   0.103   0.143   0.886  -0.182   0.230
## TradPed ~
##   Centerng (a1)  0.312   0.137   2.283   0.022   0.047   0.582
## SRPed ~
##   Centerng (a2)  0.353   0.113   3.124   0.002   0.130   0.569
## Std.lv Std.all
##
##      0.686   0.747
##      0.119   0.104
##      0.015   0.011
##
##      0.312   0.210
##
##      0.353   0.296
##
## Intercepts:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .Valued        0.710   0.469   1.514   0.130  -0.177   1.664
## .TradPed       3.870   0.231  16.773   0.000   3.419   4.291
## .SRPed        4.029   0.186  21.617   0.000   3.675   4.396
## Std.lv Std.all
##      0.710   1.077
##      3.870   5.396

```

```

##      4.029    7.013
##
## Variances:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .Valued                  0.181   0.027   6.658   0.000   0.118   0.224
## .TradPed                 0.492   0.128   3.837   0.000   0.259   0.733
## .SRPed                  0.301   0.060   5.007   0.000   0.193   0.425
## Std.lv     Std.all
## 0.181     0.418
## 0.492     0.956
## 0.301     0.912
##
## R-Square:
##                               Estimate
## Valued                   0.582
## TradPed                  0.044
## SRPed                    0.088
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## indirect1                0.214   0.105   2.045   0.041   0.032   0.442
## indirect2                 0.042   0.053   0.790   0.429   -0.080   0.148
## contrast                  0.172   0.125   1.373   0.170   -0.024   0.469
## total_indircts            0.256   0.109   2.346   0.019   0.051   0.469
## total_c                   0.271   0.142   1.914   0.056   0.003   0.576
## direct                     0.015   0.103   0.143   0.887   -0.182   0.230
## Std.lv     Std.all
## 0.214     0.157
## 0.042     0.031
## 0.172     0.126
## 0.256     0.188
## 0.271     0.199
## 0.015     0.011

```

ReC_pMedParamEsts

##	lhs	op	rhs	label	est	se
## 1	Valued	~	TradPed	b1	0.686	0.131
## 2	Valued	~	SRPed	b2	0.119	0.146
## 3	Valued	~	Centering	c_p	0.015	0.103
## 4	TradPed	~	Centering	a1	0.312	0.137
## 5	SRPed	~	Centering	a2	0.353	0.113
## 6	Valued	~~	Valued		0.181	0.027
## 7	TradPed	~~	TradPed		0.492	0.128
## 8	SRPed	~~	SRPed		0.301	0.060
## 9	Centering	~~	Centering		0.233	0.000
## 10	Valued	~1			0.710	0.469

```

## 11      TradPed ~1          3.870 0.231
## 12      SRPed ~1          4.029 0.186
## 13      Centering ~1      1.369 0.000
## 14      indirect1 :=      a1*b1      indirect1 0.214 0.105
## 15      indirect2 :=      a2*b2      indirect2 0.042 0.053
## 16      contrast :=      indirect1-indirect2   contrast 0.172 0.125
## 17 total_indirects :=    indirect1+indirect2 total_indirects 0.256 0.109
## 18      total_c := c_p+(indirect1)+(indirect2)  total_c 0.271 0.142
## 19      direct :=          c_p        direct 0.015 0.103
##           z pvalue ci.lower ci.upper std.lv std.all std.nox
## 1  5.217 0.000  0.415  0.918  0.686  0.747  0.747
## 2  0.816 0.414 -0.161  0.434  0.119  0.104  0.104
## 3  0.143 0.886 -0.194  0.207  0.015  0.011  0.022
## 4  2.283 0.022  0.047  0.582  0.312  0.210  0.435
## 5  3.124 0.002  0.134  0.571  0.353  0.296  0.614
## 6  6.658 0.000  0.143  0.268  0.181  0.418  0.418
## 7  3.837 0.000  0.279  0.813  0.492  0.956  0.956
## 8  5.007 0.000  0.204  0.454  0.301  0.912  0.912
## 9   NA    NA    0.233  0.233  0.233  1.000  0.233
## 10 1.514 0.130 -0.109  1.770  0.710  1.077  1.077
## 11 16.773 0.000  3.383  4.286  3.870  5.396  5.396
## 12 21.617 0.000  3.666  4.382  4.029  7.013  7.013
## 13  NA    NA    1.369  1.369  1.369  2.837  1.369
## 14 2.045 0.041  0.034  0.451  0.214  0.157  0.325
## 15 0.790 0.429 -0.044  0.174  0.042  0.031  0.064
## 16 1.373 0.170 -0.026  0.456  0.172  0.126  0.261
## 17 2.346 0.019  0.053  0.472  0.256  0.188  0.389
## 18 1.914 0.056  0.003  0.577  0.271  0.199  0.411
## 19 0.143 0.887 -0.194  0.207  0.015  0.011  0.022

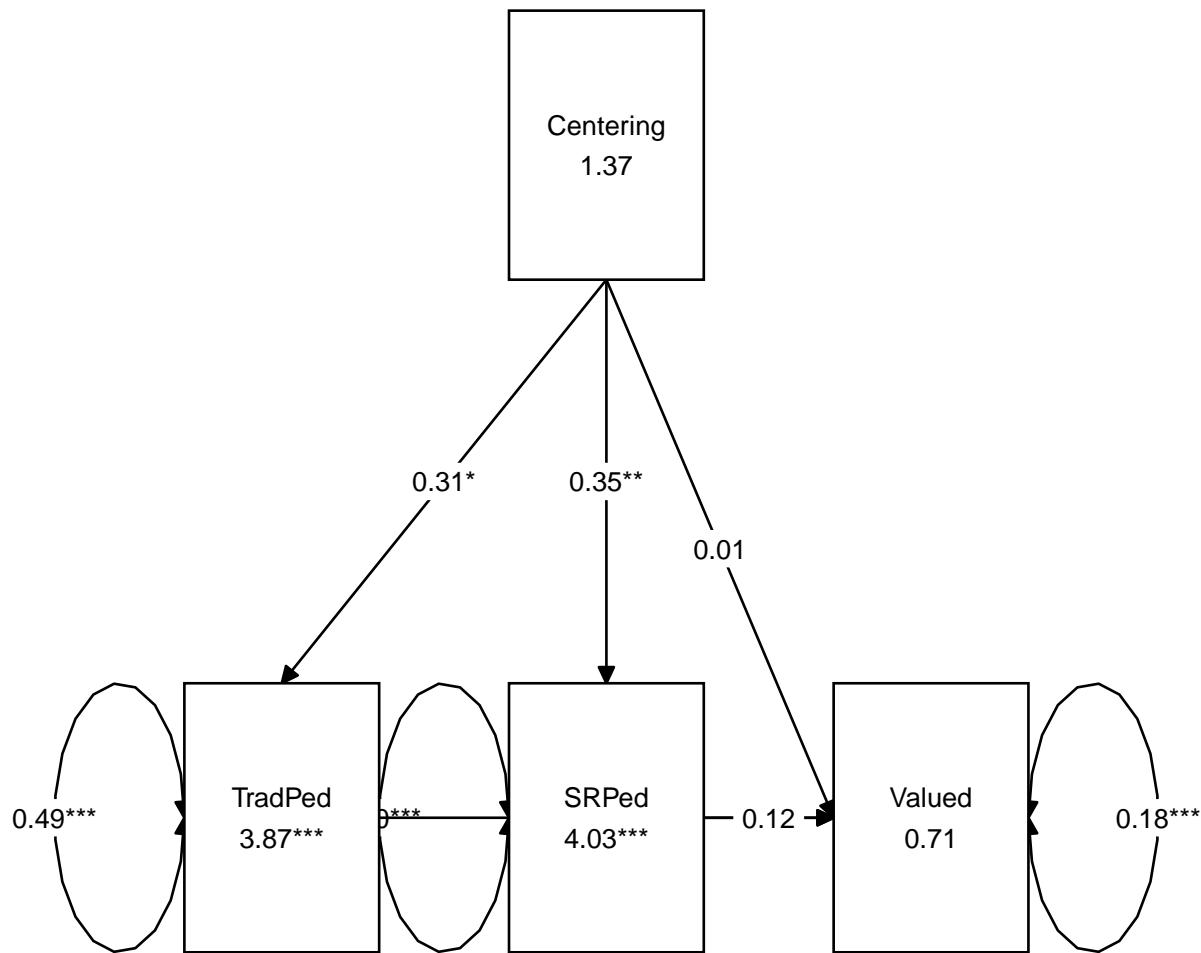
```

Use tidySEM to create a figure that represents your results

```

# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = ReCpMedfit)

```



```
tidySEM::get_layout(ReCpMedfit)
```

```
##      [,1]      [,2]      [,3]
## [1,] NA "Centering" NA
## [2,] "TradPed" "SRPed"   "Valued"
## attr(),"class")
## [1] "layout_matrix" "matrix"       "array"
```

To create the figure I showed at the beginning of the chapter, we will want three rows and three columns.

```
ReCpMed_map <- tidySEM::get_layout("", "TradPed", "", "Centering", "", 
"Valued", "", "SRPed", "", rows = 3)
ReCpMed_map
```

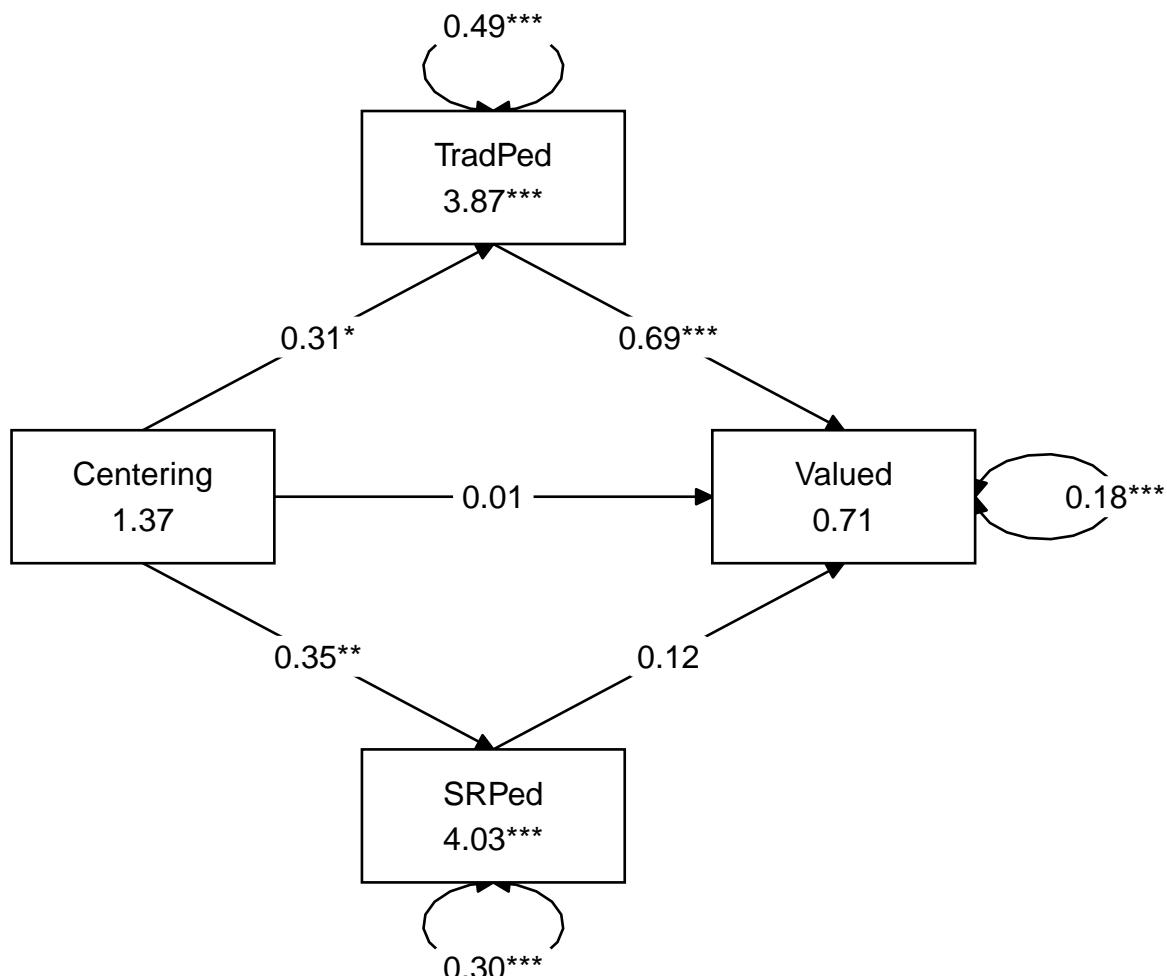
```
##      [,1]      [,2]      [,3]
```

```

## [1] ""           "TradPed"   ""
## [2] "Centering" ""          "Valued"
## [3] ""           "SRPed"    ""
## attr(),"class")
## [1] "layout_matrix" "matrix"      "array"

tidySEM::graph_sem(ReCpMedfit, layout = ReCpMed_map, rect_width = 1.5,
rect_height = 1.25, spacing_x = 2, spacing_y = 3, text_size = 4.5)

```



Create a table that includes a summary of the effects (indirect, direct, total, total indirect) as well as contrasts

I will write my results to a .csv file.

```
write.csv(ReC_pMedParamEsts, file = "ReC_pMedParamEsts.csv")
```

Table 1

Model Coefficients Assessing Students' Appraisal of Traditional and Socially Responsive Pedagogy in a Model of Parallel Mediation Predicting Perceived Course Value from Explicit Recentering

Predictor	<i>B</i>	<i>SE_B</i>	<i>p</i>	<i>R</i> ²
Traditional Pedagogy (M1)				
Constant	3.870	0.231	<0.001	
Centering (<i>a</i> ₁)	0.312	0.137	0.022	
Socially Responsive Pedagogy (M2)				
Constant	4.029	0.186	<0.001	.09
Centering (<i>a</i> ₂)	0.353	0.113	0.002	
Perceived Course Value (DV)				
Constant	0.710	0.469	0.130	
Traditional Pedagogy (<i>b</i> ₁)	0.686	0.131	<0.001	
Socially Rx Pedagogy (<i>b</i> ₂)	0.119	0.146	0.414	
Centering (<i>c'</i>)	0.015	0.103	0.886	
Effects				
Total effect	0.271	0.142	0.056	0.003, 0.577
Indirect 1 (<i>a</i> ₁ * <i>b</i> ₁)	0.214	0.105	0.041	0.034, 0.451
Indirect 2 (<i>a</i> ₂ * <i>b</i> ₂)	0.042	0.053	0.429	-0.044, 0.174
Total indirects	0.256	0.109	0.019	0.053, 0.472
Contrast1 (Ind1 - Ind2)	0.172	0.125	0.170	-0.026, 0.456

Note. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

Represent your work in an APA-style write-up

A model of parallel mediation analyzed the degree to which students' perceptions of traditional and socially responsive pedagogy mediated the relationship between explicit recentering of the course and course value. Hayes [2022b] recommended this strategy over simple mediation models because it allows for all mediators to be examined, simultaneously. The resultant direct and indirect values for each path account for other mediation paths. Using the *lavaan* (v. 0.6-16) package in R, coefficients for specific indirect, total indirect, direct, and total were computed. Path coefficients refer to regression weights, or slopes, of the expected changes in the dependent variable given a unit change in the independent variables.

Results (depicted in Figure 1 and presented in Table 1) suggest that 58% of the variance in perceptions of course value is accounted for by the model. The indirect effect predicting course value from explicit recentering through traditional pedagogy was statistically significant ($B = 0.214, SE = 0.105, p = 0.041, 95CI[0.034, 0.451]$). Examining the individual paths we see that a_1 was positive and statistically significant (recentering is associated with higher evaluations of traditional pedagogy). The b_1 path was similarly statistically significant (traditional pedagogy was associated with course valuation). The indirect effect predicting course value from recentering through socially responsive pedagogy was not statistically significant $B = 0.042, SE = 0.053, p = 0.429, 95CE[-0.044, 0.174]$). While explicit recentering had a statistically significant effect on ratings of socially responsive pedagogy (i.e., the a_2 path), socially responsive pedagogy did not have a statistically significant effect on perceptions of course value (i.e., the b_2 path). The drop in magnitude and near-significance from the total effect ($B = 0.271, p = 0.056$) to the direct effect ($B = 0.015, p = 0.886$) supports the presence of mediation. A pairwise comparison of the specific indirect effects indicated that the strength of the effects were not statistically significantly different from each other. In summary, the effects of explicit recentering on perceived value to the student appears to be mediated through students evaluation of traditional pedagogy.

Explanation to grader

Be able to hand-calculate the indirect, direct, and total effects from the a, b, & c' paths

- Indirect = a^*b
- Direct = Total minus indirect
- Total = $(a^*b) + c'$

A homework idea

Augment this model to a serial mediation – adding a path from traditional pedagogy to socially responsive pedagogy.

MODERATION

Chapter 7

Simple Moderation in OLS and MLE

[Screencasted Lecture Link](#)

The focus of this lecture is an overview of simple moderation. Sounds simple? Wait, there's more! The focus of this lecture is the transition:

- from null hypothesis significance testing (NHST) to modeling
- from *ordinary least squares* (OLS) to *maximum likelihood estimation* (MLE)

In making the transition we will work a moderation/interaction problem with both `lm()` and `lavvan/sems()` functions.

7.1 Navigating this Lesson

There is about 1 hour and 10 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

Learning objectives from this lecture include the following:

- Distinguish between NHST and model building approaches
- Name the primary characteristics that distinguish ordinary least squares from maximum likelihood approaches to regression.
- Interpret “the usual” things we find in regression: B/beta weights, R, R^2 .
- Define and interpret simple slopes and probing an interaction, this includes
 - pick-a-point and Johnson-Neyman approaches

- interpreting interaction plots/figures
- Recognize the path specification in *lavaan*. That is, you should be able to figure out a diagram from the *lavaan* code. In reverse, you should be able to write (or identify) the proper code in *lavaan*.

7.1.2 Planning for Practice

As is typical for this OER, the suggestions for homework are graded in complexity. I recommend you select an option that builds on your confidence but provides a bit of stretch. I also suggest you utilize a dataset that has at least four variables that are suitable for growing into a complex moderation (additive or moderated) or moderated mediation as well as a moderated mediation. This will be easiest if the variables are continuous in nature. In these chapters, I do not describe how to use categorical variables in dependent (e.g., consequent or endogenous) roles. However, dichotomous and ordered factors are suitable as independent variables and covariates.

- Rework the problem in the chapter by changing the random seed in the code that simulates the data. This should provide minor changes to the data, but the results will likely be very similar.
- There are a number of variables in the dataset. Swap out one or more variables in the simple moderation and compare your solution to the one in the chapter (and/or one you mimicked in the journal article).
- Conduct a simple moderation with data to which you have access. This could include data you simulate on your own or from a published article.

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.

Regarding ordinary least squares (OLS) versus maximum likelihood estimation (MLE), these articles are extremely helpful:

- Cohen, J. (2003). Maximum likelihood estimation. Section 13.2.9 (pp. 498-499). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Erlbaum Associates.
- Cumming, G. (2014). The New Statistics: Why and How. *Psychological Science*, 25(1), 7–29. <https://doi.org/10.1177/0956797613504966>
- Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1), 90–100. [https://doi.org/10.1016/S0022-2496\(02\)00028-7](https://doi.org/10.1016/S0022-2496(02)00028-7) (skim for big ideas)
- Rodgers, J. L. (2010). The epistemology of mathematical and statistical modeling: A quiet methodological revolution. *American Psychologist*, 65(1), 1–12. <https://doi.org/10.1037/a0018326>

Regarding the topic of moderation, I drew heavily from these resources.

- Hayes, A. F. (2022). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. New York, NY: Guilford Press.
 - **Chapter 7: Fundamentals of moderation analysis:** This chapter focuses on the basics of moderation analysis. Our goal is to transfer and apply the knowledge to models we run in lavaan. An excellent review of centering, visualizations, and probing moderation models.
 - **Chapter 8: Extending the fundamental principles of moderation analysis:** Hayes addresses common regression concerns such as (a) hierarchical vs. simultaneous entry and (b) comparison of moderated regression with 2x2 factorial ANOVA.
 - **Chapter 9: Some myths and additional extensions of moderation Aanalysis.** Hayes identifies “truths and myths” about mean centering and standardization. For sure these are important topics and his take on them is clear and compelling.
 - **Appendix A Using PROCESS:** An essential tool for PROCESS users because, even when we are in the R environment, this is the “idea book.” That is, the place where all the path models are presented in figures.

The research vignette for this chapter:

- Lewis, J. A., Williams, M. G., Peppers, E. J., & Gadson, C. A. (2017). Applying intersectionality to explore the relations between gendered racism and health among Black women. *Journal of Counseling Psychology*, 64(5), 475–486. <https://doi-org.ezproxy.spu.edu/10.1037/cou0000231>

7.1.4 Packages

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(apaTables)) {
  install.packages("apaTables")
}
if (!require(lavaan)) {
  install.packages("lavaan")
}
if (!require(tidyverse)) {
  install.packages("tidyverse")
}
if (!require(psych)) {
  install.packages("psych")
}
if (!require(jtools)) {
  install.packages("jtools")
}
if (!require(broom)) {
  install.packages("broom")
```

```

}
if (!require(interactions)) {
  install.packages("interactions")
}
if (!require(tidySEM)) {
  install.packages("tidySEM")
}

```

7.2 On *Modeling*: Introductory Comments on the simultaneously invisible and paradigm-shifting transition we are making

7.2.1 NHST versus modeling

At least a decade old now, Rogers' [2010] article in the *American Psychologist* is one of my favorites. In it, he explores the notion of *statistical modeling*. He begins with criticisms of null hypothesis statistical testing by describing how it has become a awkward and incongruent blend of Fisherian (i.e., R.A. Fisher) and Neyman-Pearson (i.e., Jerzy Neyman and E. S. Pearson) approaches.

Table 1

Contributions of the Fisherian and Neyman-Pearson Approaches to NHST [Rodgers, 2010]

Fisher	Neyman-Pearson
Developed NHST to answer scientific questions and evaluate theory.	Sought to draw conclusions in applied settings such as quality control.
Took an incremental approach to hypothesis testing that involved replication and (potentially) self-correcting; as such viewed <i>replication</i> as a critical element.	Placed emphasis on the importance of each individual decision.
Never used the terms, “alternative hypothesis” or “alpha level.” Rather, Fisher used the distribution of the null model to examine “whether the data look weird or not.”	Designed their approach to detect an “alternative hypothesis.”
Gave us the null hypothesis and <i>p</i> value.	Gave us the alternative hypothesis, alpha level, and power.

Over time, these overlapping, but inconsistent, approaches became intertwined. Many students of statistics do not recognize the incompatibilities. Undoubtedly, it makes statistics more difficult to learn (and teach). Below are some of the challenges that Rodgers [2010] outlined.

- Rejecting the null does not provide logical or strong support for the alternative

7.2. ON MODELING: INTRODUCTORY COMMENTS ON THE SIMULTANEOUSLY INVISIBLE AND PARA

- Failing to reject the null does not provide logical or strong support for the null.
- NHST is backwards because it evaluates the probability of the data given the hypothesis, rather than the probability of the hypothesis given the data.
- All point-estimate null hypotheses can be rejected if the sample size is large enough.
- Statistical significance does not necessitate practical significance.

Consequently, we have ongoing discussion/debates about power, effect sizes, sample size, Type I and II errors, confidence intervals, fit statistics, and the relations between them.

7.2.2 Introducing: *The Model*

Understanding modeling in our *scientist-practitioner* context probably needs to start with understanding the *mathematical model*. Niemark and Este [1967] defined a mathematical model as a set of assumptions together with implications drawn from them by mathematical reasoning. Luce [Luce, 1995] suggested that mathematical equations capture model-specific features by highlighting some aspects while ignoring others. The use of mathematics helps us uncover the “structure.” For example, the *mean* is a mathematical model. *I always like to stop and think about that notion...about what the mean represents and what it doesn’t.* Pearl [2000] defined the model as an idealized representation of reality that highlights some aspects and ignores others by suggesting that a model:

- matches the reality it describes in some important ways.
- is simpler than that reality.

As we transition from the NHST approach to statistical modeling there is [Rodgers, 2010]:

- decreased emphasis on
 - null hypothesis
 - *p* values
- increased emphasis on
 - model residuals
 - degrees of freedom
 - additional indices of *fit*

Further, statistical models [Rodgers, 2010]:

- are more readily falsifiable
- require greater theoretical precision
- include assumptions that are more readily evaluated
- offer more practical application

Circling back around to Fisher and Neyman-Pearson, Rogers [2010] contended that Fisher’s work provided a framework for modeling because of the model process of specification, estimation, and goodness of fit. As we move into more complex modeling, we will spend a great deal of time

understanding parameters and their relationship to degrees of freedom. Fisher viewed degrees of freedom as *statistical currency* that could be used in exchange for the estimation of parameters.

If this topic is exciting to you, let me refer you to Cumming's [[Cumming, 2014](#)] article, "The New Statistics: Why and How," in the Journal, *Psychological Science"

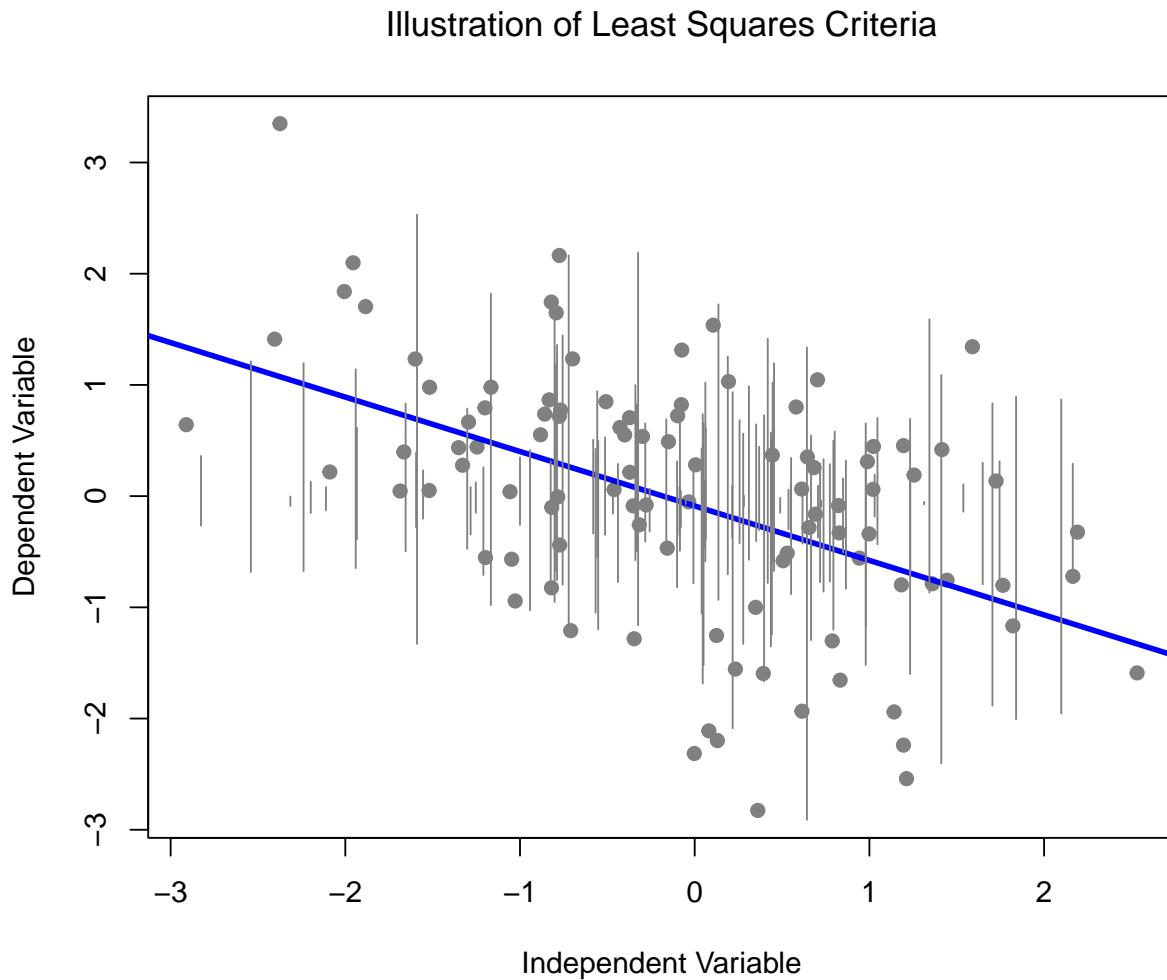
7.3 OLS to ML for Estimation

7.3.1 Ordinary least squares (OLS)

Known by a variety of names, the estimation algorithm typically used in regression models (linear, hierarchical, multiple, sequential) is *ordinary least squares* (OLS; also termed least squares criterion, general least squares, etc.). As we move into multivariate (and then psychometrics) we are going to transition our estimation method from OLS to MLE. Consequently, it is essential to understand some underlying differences [[Cohen et al., 2003](#), [Myung, 2003](#)]

In OLS regression:

- The estimated values of regression coefficients are chosen so that the sum of squared errors is minimized (aka, the *least squares criteria*). Consequently,
 - the mean of errors is zero, and
 - the errors correlate *zero* with each predictor
- The solution to OLS regression is *analytic*
 - the equations from which the coefficients are created are *known normal equations*. Among other places, you can look them up in CCW&A [[Cohen and Nagel, 1934](#)] Appendix 1)



7.3.2 Maximum likelihood estimation (MLE): A brief orientation

Although I started this chapter with a critique of NHST, Fisher is credited [Myung, 2003] with the original development of the central principle of *maximum likelihood estimation* which is that the desired probability distribution is the one that makes the observed data *most likely*. As such, the *MLE estimate* is a resulting parameter vector that maximizes the likelihood function. Myung's [2003] tutorial provides an excellent review. My summary is derived from Dr. Myung article. A *likelihood* is a measure of how *typical* a person (or sample) is of that population.

- When there is one IV the MLE distribution behaves like a chi-square distribution (which also tests observed versus expected data).
- There is a point in the MLE curve that represents where the maximum likelihood exists that the data is likely given the model.
- When there are multiple IVs, this simple curve takes the shape of a k dimensional geometrical surface.

Extended to regression, we are interested in the *likelihoods* of individuals having particular scores on Y, given values on predictors x_1 to x_k (and the specific values of regression coefficients chosen as the parameter estimates)

- MLE provides *maximum likelihood estimates* of the regression coefficients (and SEs) that is, estimates that make a sample as likely or typical as possible
- L is a symbol for *maximum likelihood of a sample*
- The solutions are *iterative* (i.e., identified by trial-and-error; with each trial informed by the prior)
 - a statistical criteria is specified for the coefficients to be chosen
 - different values of coefficients are tried
 - these *iterations* continue until the regression coefficients cease to change by more than a small amount (i.e., the *convergence criteria*)
 - hopefully, a set of coefficients is found that makes the solution as close to the statistical criteria (i.e., maximum likelihood) as possible
- The *optimization algorithm* does not guarantee that a set of parameters will be found; convergence failures may be caused by
 - multicollinearity among predictors
 - a large number of predictors
 - the *local maxima problem*; the optimization algorithm returns sub-optimal parameter values [Myung, 2003]
- MLE is a *full information model*
 - calculates the estimates of model parameters all at once
- MLE is for large samples
- MLE assumptions include
 - independence of observations
 - multivariate normality of endogenous variables
 - independence of exogeneous variables and disturbances
 - correct specification of the model (MLE is only appropriate for testing theoretically informed models)

7.3.3 OLS and MLE Comparison

In this table we can compare OLS and MLE in a side-by-side manner. **Table 2**

Comparing OLS and MLE [Cohen et al., 2003, Myung, 2003]

Criterion	Ordinary Least Squares (OLS)	Maximum Likelihood Estimation (MLE)
-----------	------------------------------	-------------------------------------

Parameter values chosen to...	minimize the distance between the predictions from regression line and the observations; considered to be those that are <i>most accurate</i>	be those that are <i>most likely</i> to have produced the data
Parameter values are obtained by	equations that are known and linear (you can find them in the “back of the book”)	a non-linear optimization algorithm
Preferred when...	sample size is small	sample size is large, for complex models, non-linear models, and when OLS and MLE results differ
In R...	the <i>lm()</i> function in base R	<i>lavaan</i> and other packages*; specifying the FIML option allows for missing data (without imputation)

7.3.4 Hayes and PROCESS (aka conditional process analysis)

In the early 2000s, the bias-corrected, bootstrapped, confidence interval (CI) was identified as a more powerful approach to assessing indirect effects than the classic Sobel test. Because programs did not produce them, no one was using them. Preacher, Edwards, Lambert, Hayes, and colleagues created Excel worksheets that would calculate these (they were so painful). Hayes turned this process into a *series* of macros to do a variety of things for SPSS and other programs. Because of his clear, instructional, text, PROCESS is popular. In 2021, Hayes released the PROCESS macro for R. It can be downloaded at the [ProcessMacro website](#). The 2022 of Hayes’ textbook now includes instruction for using the Process Macro for R. Although PROCESS produces bias-corrected, bootstrapped confidence intervals, for models with indirect effects, PROCESS utilizes OLS as the estimator. Additionally, the Process Macro for R does not work like a typical R package. Further, at my latest review, I could not determine how to create figures (in R) that would represent the results. Thus, I am continuing to teach this topic with *lavaan*.

Although most regression models can be completed with the *lm()* function in base R, it can be instructive to run a handful of these familiar models with *lavaan* (or even PROCESS) as a precursor to more complicated models.

7.4 Introducing the *lavaan* package

In the regression classes (as well as in research designs that are cross-sectional, non-linear, and can be parsimoniously and adequately measured with OLS regression) we typically use the base R function, *lm()* (“linear model”) which relies on an OLS algorithm. You can learn about it with this simple code:

```
#?lm
```

Rosseel’s [2020] *lavaan* package was developed for SEM, but is readily adaptable to most multiple regression models. Which do we use and when?

- For relatively simple models that involve only predictors, covariates, and moderators, `lm()` is adequate.
- Models that involve mediation need to use *lavaan*
- SEM/CFA needs *lavaan*
- If your sample size is small, *but* you are planning a mediation, it gets tricky (try to increase your sample size) because MLE estimators rely on large sample sizes (how big? hard to say).

7.4.1 The FIML magic for which we have been waiting

There are different types of maximum likelihood. In this chapter we'll utilize *full information maximum likelihood* (FIML). FIML is one of the most practical missing data estimation approaches around and is especially used in SEM and CFA. When data are thought to be MAR (missing at random) or MCAR (missing completely at random), it has been shown to produce unbiased parameter estimates and standard errors.

The FIML approach works by estimating a likelihood function for each individual based on the variables that are present so that all available data are used. Model fit is calculated from (or informed by) the fit functions for all individual cases. Hence, “FIML” is *full information* maximum likelihood.

When I am able to use *lavaan*, my approach is to use Parent’s AIA (available information analysis, -[Parent \[2013\]](#)) approach to scoring data, then specify a FIML approach (i.e., adding `missing = 'fiml'`) in my *lavaan* code. Even though the text-book examples we work have complete data, I will try to include this code so that it will be readily available for you, should you use the as templates for your own data.

In this portion of the ReCentering Psych Stats series we are headed toward more complex models that include both mediation and moderation. Hayes [[Hayes, 2018](#)] would call this “conditional process analysis.” Others would simply refer to it as “path analysis.” Although all these terms are sometimes overlapping, *path analysis* is a distinction from *structural equation modeling* (SEM) where latent variables are composed of the observed variables. Let’s take a look at some of the nuances of the whole SEM world and how it relates to PROCESS.

SEM is broad term (that could include CFA and path analysis) but is mostly reserved for models with some type of latent variable (i.e., some might exclude path analysis from its definitions). SEM typically uses some form of MLE (not ordinary least squares).

Latent variables (circles in the model, below) are those that are “created” in the analytic process but will never appear as a column in your dataset. It may be easiest to think of a latent variable as a scale score – where you sum (or average) the indicator item values to get the score (except we don’t do that). Rather, the LV is “indicated” by variance the indicator/observed/manifest variables share with each other.

The image below is of a simple mediation model but the variables in the model are latent, and indicated by each of the 3 observed/manifest variables. PROCESS (in SPSS) could not assess this model because PROCESS uses ordinary least squares regression and SEM will use a maximum likelihood estimator.

Confirmatory factor analysis (CFA) is what we will do (or have done) in psychometrics. CFA is used to evaluate the structural validity of a scale or measure. In CFA, first-order factors represent subscales and a second-order factor (not required) might provide support for a total scale score.

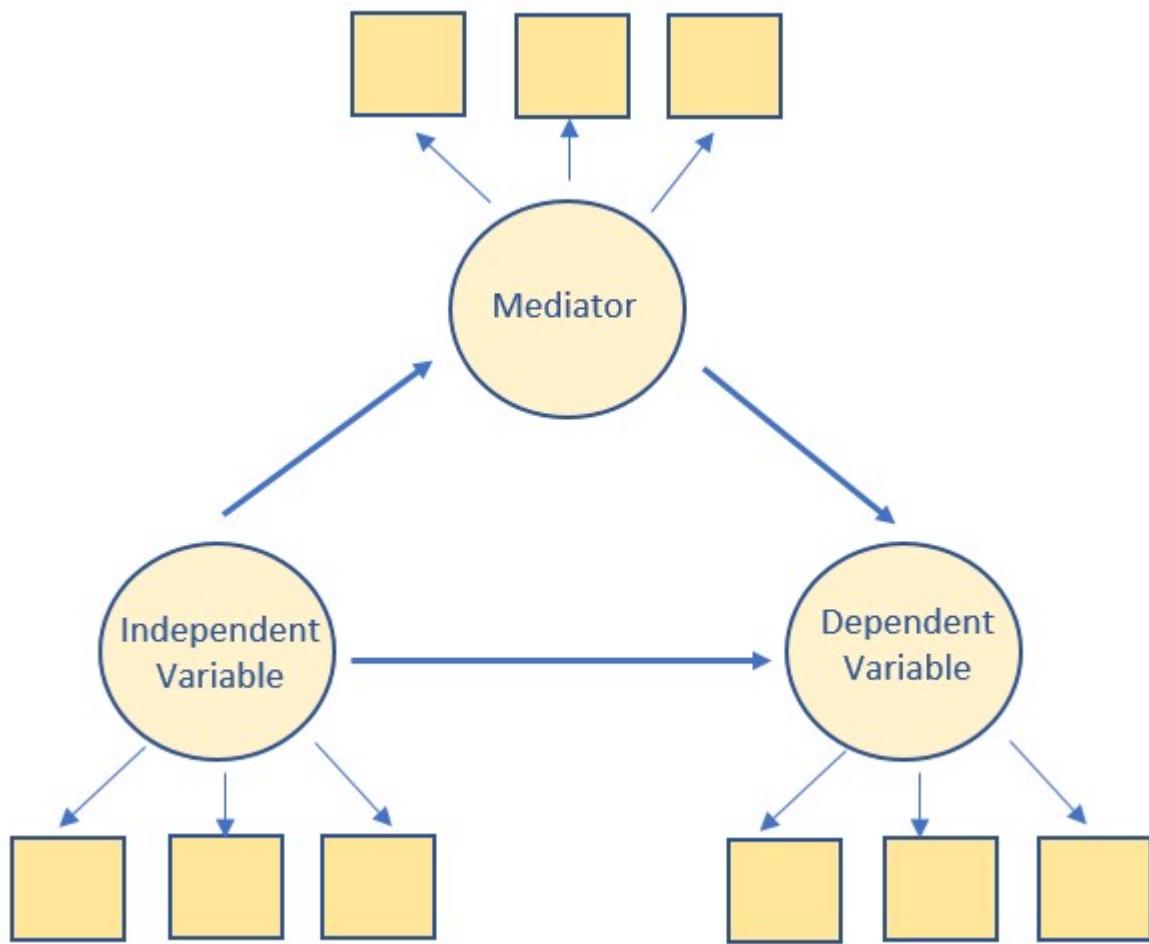


Figure 7.1: Image of a simple mediation model with latent variables

For example, in the above figure, the three squares represent the observed (or manifest) items to which a person respond. In CFA, we evaluate their adequacy to represent the latent variable (circle) construct. It's a little more complicated than this, but this will get you started. Mediation/indirect effects are not assessed in a pure CFA.

Path analysis is a form of SEM, but without latent variables. That is, all the variables in the model are directly observed. They are represented by squares/rectangles and each has a corresponding column in a dataset. PROCESS in SPSS is entirely path analysis.

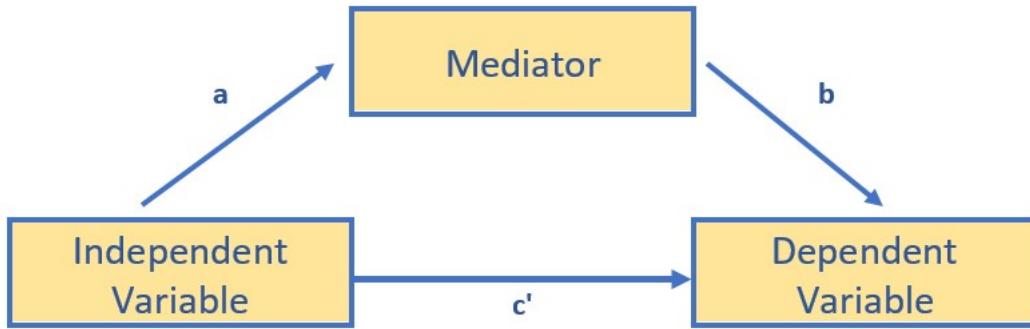


Figure 7.2: Image of a simple mediation in path analysis

Hybrid models are a form of SEM that include observed/manifest variables as predictors along with other latent variables. In the diagram below, you see tiny little measurement models (3 indicators that “create” or “inform” an LV, think baby CFA) and one predictor that is manifest. An example might be a categorical predictor (e.g., treatment, control).

7.5 Picking up with Moderation

Moderation: The effect of X (IV) on some variable Y (DV) is moderated if its size, sign, or strength depends on or can be predicted by W (moderator). In that case, W is said to be a *moderator* of X’s effect on Y. Or, that W and X *interact* in their influence on Y.

Identifying a moderator of an effect helps establish the *boundary conditions* of that effect or the circumstances, stimuli, or type of people for which the effect is large versus small, present versus absent, positive versus negative, and so forth.

Conditional vs Unconditional Effects: Consider the following two equations:

$$\hat{Y} = i_y + b_1X + b_2W + e_y$$

and

$$\hat{Y} = i_y + b_1X + b_2W + b_3XW + e_y$$

The first equation constrains X’s effect to be unconditional on W, meaning that it is invariant across all values of W. By introducing the interaction term (b_3XW), we can evaluate a model where X’s effect can be dependent on W. That is, for different values of W, X’s effect on Y is different. The resulting equation (#2) is the *simple linear moderation model*. In it, X’s effect on Y is *conditional*.

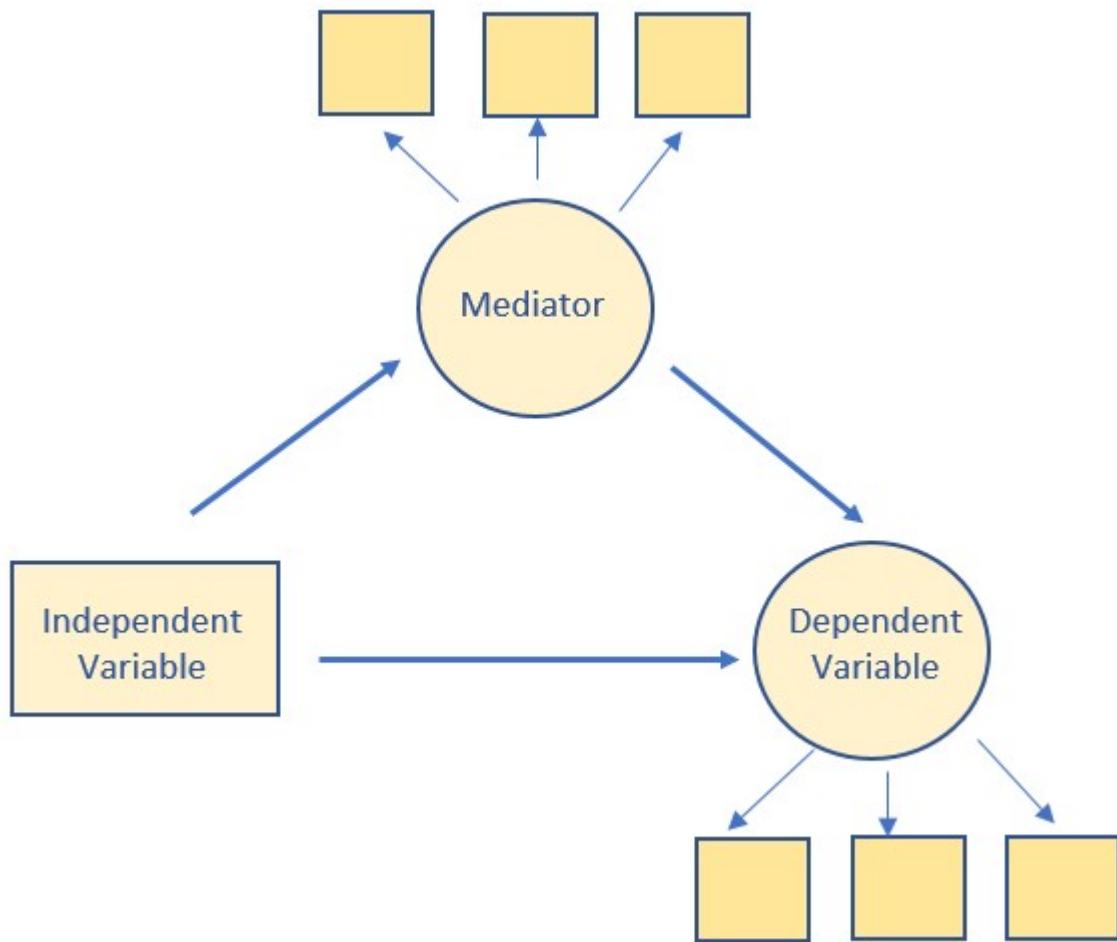


Figure 7.3: Image of a simple mediation in path analysis

7.6 Workflow for a Simple Moderation

Below is a workflow comparing the approaches to analyzing a regression model (moderators only) with OLS and MLE. Of course you would precede both options with a thorough scrubbing, scoring, and data diagnostics. Please refer to the earlier lessons for workflows for those processes.

Workflow (and comparison) of moderated models with OLS and MLE approaches

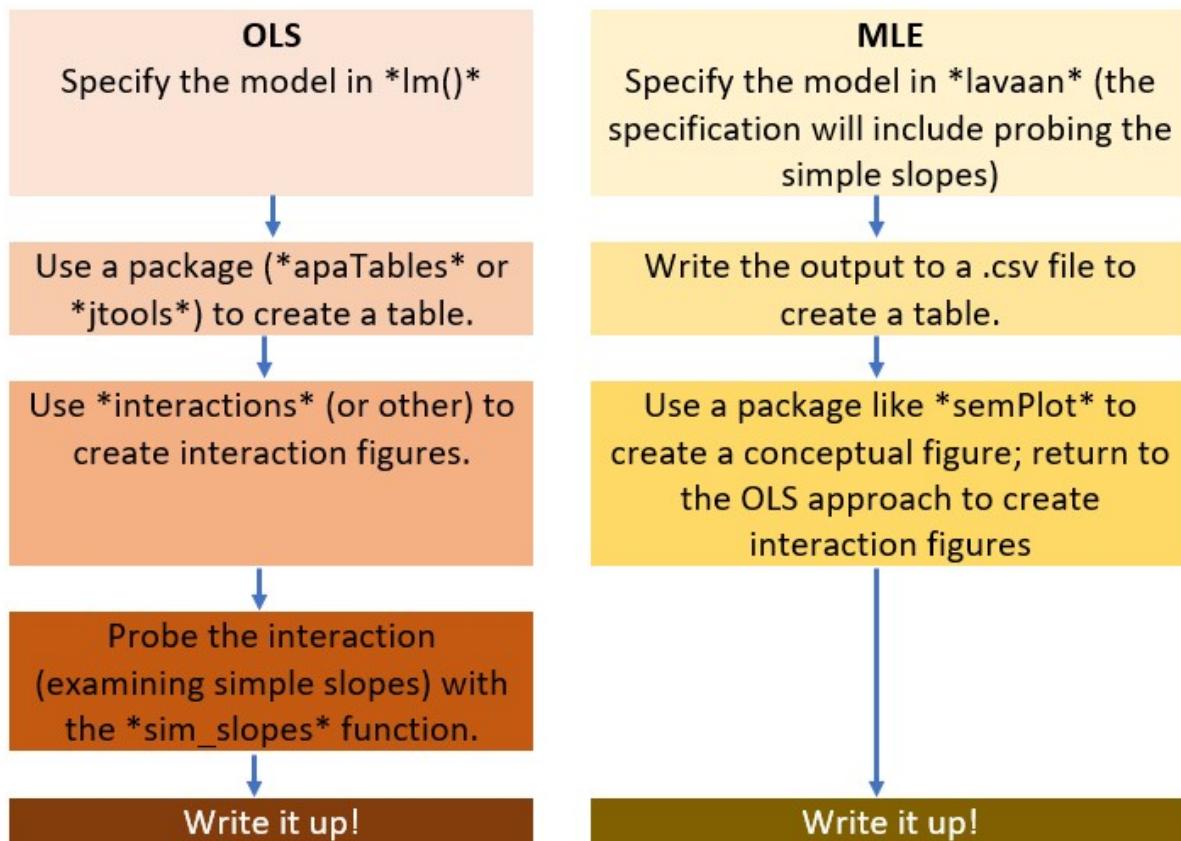


Figure 7.4: Image of a simple mediation in path analysis

The Bonus Track at the end of the chapter includes script templates with just X and Y variables.

7.6.1 Research Vignette

The research vignette comes from the Lewis, Williams, Peppers, and Gadson's [2017] study titled, "Applying Intersectionality to Explore the Relations Between Gendered Racism and Health Among Black Women." The study was published in the Journal of Counseling Psychology. Participants were 231 Black women who completed an online survey.

Variables used in the study included:

- **GRMS:** Gendered Racial Microaggressions Scale [Lewis and Neville, 2015] is a 26-item scale that assesses the frequency of nonverbal, verbal, and behavioral negative racial and gender slights experienced by Black women. Scaling is along six points ranging from 0 (*never*) to 5 (*once a week or more*). Higher scores indicate a greater frequency of gendered racial microaggressions. An example item is, “Someone has tried to ‘put me in my place.’”
- **MntlHlth** and **PhysHlth**: Short Form Health Survey - Version 2 [Ware et al., 1995] is a 12-item scale used to report self-reported mental (six items) and physical health (six items). Although the article did not specify, when this scale is used in other contexts [e.g., Kim et al., 2017], a 6-point scale has been reported. Higher scores indicate higher mental health (e.g., little or no psychological distress) and physical health (e.g., little or no reported symptoms in physical functioning). An example of an item assessing mental health was, “How much of the time during the last 4 weeks have you felt calm and peaceful?”; an example of a physical health item was, “During the past 4 weeks, how much did pain interfere with your normal work?”
- **Sprtly**, **SocSup**, **Engmgt**, and **DisEngmnt** are four subscales from the Brief Coping with Problems Experienced Inventory [Carver, 1997]. The 28 items on this scale are presented on a 4-point scale ranging from 1 (*I usually do not do this at all*) to 4(*I usually do this a lot*). Higher scores indicate a respondents’ tendency to engage in a particular strategy. Instructions were modified to ask how the female participants responded to recent experiences of racism and sexism as Black women. The four subscales included spirituality (religion, acceptance, planning), interconnectedness/social support (vent emotions, emotional support,instrumental social support), problem-oriented/engagement coping (active coping, humor, positive reinterpretation/positive reframing), and disengagement coping (behavioral disengagement, substance abuse, denial, self-blame, self-distraction).
- **GRIcntlty:** The Multidimensional Inventory of Black Identity Centrality subscale [Sellers et al.] was modified to measure the intersection of racial and gender identity centrality. The scale included 10 items scaled from 1 (*strongly disagree*) to 7 (*strongly agree*). An example item was, “Being a *Black woman* is important to my self-image.” Higher scores indicated higher levels of gendered racial identity centrality.

7.6.1.1 Data Simulation

The *lavaan::simulateData* function was used. If you have taken psychometrics, you may recognize the code as one that creates latent variables form item-level data. In trying to be as authentic as possible, we retrieved factor loadings from psychometrically oriented articles that evaluated the measures [Nadal, 2011, Veit and Ware, 1983]. For all others we specified a factor loading of 0.80. We then approximated the *measurement model* by specifying the correlations between the latent variable. We sourced these from the correlation matrix from the research vignette [Lewis et al., 2017]. The process created data with multiple decimals and values that exceeded the boundaries of the variables. For example, in all scales there were negative values. Therefore, the final element of the simulation was a linear transformation that rescaled the variables back to the range described in the journal article and rounding the values to integer (i.e., with no decimal places).

```
#Entering the intercorrelations, means, and standard deviations from the journal article
```

```

Lewis_generating_model <- '
  ##measurement model
  GRMS  =~ .69*Ob1 + .69*Ob2 + .60*Ob3 + .59*Ob4 + .55*Ob5 + .55*Ob6 + .54*Ob7 + .50*Ob8
  MntlHlth  =~ .8*MH1 + .8*MH2 + .8*MH3 + .8*MH4 + .8*MH5 + .8*MH6
  PhysHlth  =~ .8*PhH1 + .8*PhH2 + .8*PhH3 + .8*PhH4 + .8*PhH5 + .8*PhH6
  Spirituality  =~ .8*Spirit1 + .8*Spirit2
  SocSupport  =~ .8*SocS1 + .8*SocS2
  Engagement  =~ .8*Eng1 + .8*Eng2
  Disengagement  =~ .8*dEng1 + .8*dEng2
  GRIC  =~ .8*Cntrlty1 + .8*Cntrlty2 + .8*Cntrlty3 + .8*Cntrlty4 + .8*Cntrlty5 + .8*Cntrlty6

  # Means
  GRMS ~ 1.99*1
  Spirituality ~2.82*1
  SocSupport ~ 2.48*1
  Engagement ~ 2.32*1
  Disengagement ~ 1.75*1
  GRIC ~ 5.71*1
  MntlHlth ~3.56*1 #Lewis et al used sums instead of means, I recast as means to facilitate comparison
  PhysHlth ~ 3.51*1 #Lewis et al used sums instead of means, I recast as means to facilitate comparison

  # Correlations (ha!)
  GRMS ~ 0.20*Spirituality
  GRMS ~ 0.28*SocSupport
  GRMS ~ 0.30*Engagement
  GRMS ~ 0.41*Disengagement
  GRMS ~ 0.19*GRIC
  GRMS ~ -0.32*MntlHlth
  GRMS ~ -0.18*PhysHlth

  Spirituality ~ 0.49*SocSupport
  Spirituality ~ 0.57*Engagement
  Spirituality ~ 0.22*Disengagement
  Spirituality ~ 0.12*GRIC
  Spirituality ~ -0.06*MntlHlth
  Spirituality ~ -0.13*PhysHlth

  SocSupport ~ 0.46*Engagement
  SocSupport ~ 0.26*Disengagement
  SocSupport ~ 0.38*GRIC
  SocSupport ~ -0.18*MntlHlth
  SocSupport ~ -0.08*PhysHlth

  Engagement ~ 0.37*Disengagement
  Engagement ~ 0.08*GRIC
  Engagement ~ -0.14*MntlHlth
  Engagement ~ -0.06*PhysHlth

```

```

Disengagement ~ 0.05*GRIC
Disengagement ~ -0.54*MntlHlth
Disengagement ~ -0.28*PhysHlth

GRIC ~ -0.10*MntlHlth
GRIC ~ 0.14*PhysHlth

MntlHlth ~ 0.47*PhysHlth
'

set.seed(230925)
dfLewis <- lavaan::simulateData(model = Lewis_generating_model,
                                 model.type = "sem",
                                 meanstructure = T,
                                 sample.nobs=231,
                                 standardized=FALSE)

#used to retrieve column indices used in the rescaling script below
#col_index <- as.data.frame(colnames(dfLewis))

for(i in 1:ncol(dfLewis)){ # for loop to go through each column of the dataframe
  if(i >= 1 & i <= 25){ # apply only to GRMS variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 5))
  }
  if(i >= 26 & i <= 37){ # apply only to mental and physical health variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 6))
  }
  if(i >= 38 & i <= 45){ # apply only to coping variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 4))
  }
  if(i >= 46 & i <= 55){ # apply only to GRIC variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 7))
  }
}
#rounding to integers so that the data resembles that which was collected
library(tidyverse)
dfLewis <- dfLewis %>% round(0)

#quick check of my work
#psych::describe(dfLewis)

```

The script below allows you to store the simulated data as a file on your computer. This is optional – the entire lesson can be worked with the simulated data.

If you prefer the .rds format, use this script (remove the hashtags). The .rds format has the advantage of preserving any formatting of variables. A disadvantage is that you cannot open these

files outside of the R environment.

Script to save the data to your computer as an .rds file.

```
# saveRDS(dfLewis, 'dfLewis.rds')
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfLewis<- readRDS('dfLewis.rds')
```

If you prefer the .csv format (think “Excel lite”) use this script (remove the hashtags). An advantage of the .csv format is that you can open the data outside of the R environment. A disadvantage is that it may not retain any formatting of variables

Script to save the data to your computer as a .csv file.

```
# write.table(dfLewis, file = 'dfLewis.csv', sep = ',',
# col.names=TRUE, row.names=FALSE)
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfLewis<- read.csv ('dfLewis.csv', header = TRUE)
```

7.6.2 Scrubbing, Scoring, and Data Diagnostics

Because the focus of this lesson is on moderation, we have used simulated data (which serves to avoid problems like missingness and non-normal distributions). If this were real, raw, data, it would be important to [scrub](#), [score](#), and conduct [data diagnostics](#) to evaluate the suitability of the data for the proposed analyses.

Because we are working with item level data we do need to score the scales used in the researcher’s model. Because we are using simulated data and the authors already reverse coded any such items, we will omit that step.

As described in the [Scoring](#) chapter, we calculate mean scores of these variables by first creating concatenated lists of variable names. Next we apply the `sjstats::mean_n` function to obtain mean scores when a given percentage (we’ll specify 80%) of variables are non-missing. Functionally, this would require the two-item variables (e.g., engagement coping and disengagement coping) to have non-missingness. We simulated a set of data that does not have missingness, none-the-less, this specification is useful in real-world settings.

Note that I am only scoring the variables used in the models demonstrated in this lesson. The remaining variables are available as practice options.

```
GRMS_vars <- c("Ob1", "Ob2", "Ob3", "Ob4", "Ob5", "Ob6", "Ob7", "Ob8",
               "Ob9", "Ob10", "Ma1", "Ma2", "Ma3", "Ma4", "Ma5", "Ma6", "Ma7", "St1",
               "St2", "St3", "St4", "St5", "An1", "An2", "An3")
Eng_vars <- c("Eng1", "Eng2")
```

```

dEng_vars <- c("dEng1", "dEng2")
MntlHlth_vars <- c("MH1", "MH2", "MH3", "MH4", "MH5", "MH6")
Cntrlty_vars <- c("Cntrlty1", "Cntrlty2", "Cntrlty3", "Cntrlty4", "Cntrlty5",
  "Cntrlty6", "Cntrlty7", "Cntrlty8", "Cntrlty9", "Cntrlty10")

dfLewis$GRMS <- sjstats::mean_n(dfLewis[, GRMS_vars], 0.8)
dfLewis$Engmt <- sjstats::mean_n(dfLewis[, Eng_vars], 0.8)
dfLewis$DisEngmt <- sjstats::mean_n(dfLewis[, dEng_vars], 0.8)
dfLewis$MntlHlth <- sjstats::mean_n(dfLewis[, MntlHlth_vars], 0.8)
dfLewis$Centrality <- sjstats::mean_n(dfLewis[, Cntrlty_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfLewis$GRMS <-
# sjstats::mean_n(dfLewis[, ..GRMS_vars], 0.80)

```

Now that we have scored our data, let's trim the variables to just those we need.

```
Lewis_df <- dplyr::select(dfLewis, GRMS, Centrality, MntlHlth)
```

Let's check a table of means, standard deviations, and correlations to see if they align with the published article.

```

Lewis_table <- apaTables::apa.cor.table(Lewis_df, table.number = 1, show.sig.stars = TRUE,
  landscape = TRUE, filename = "Lewis_Corr.doc")
print(Lewis_table)

```

Table 1

Means, standard deviations, and correlations with confidence intervals

Variable	M	SD	1	2
1. GRMS	2.56	0.72		
2. Centrality	3.94	0.76	.24**	
			[.11, .36]	
3. MntlHlth	3.16	0.81	-.56**	-.09
			[-.64, -.47]	[-.21, .04]

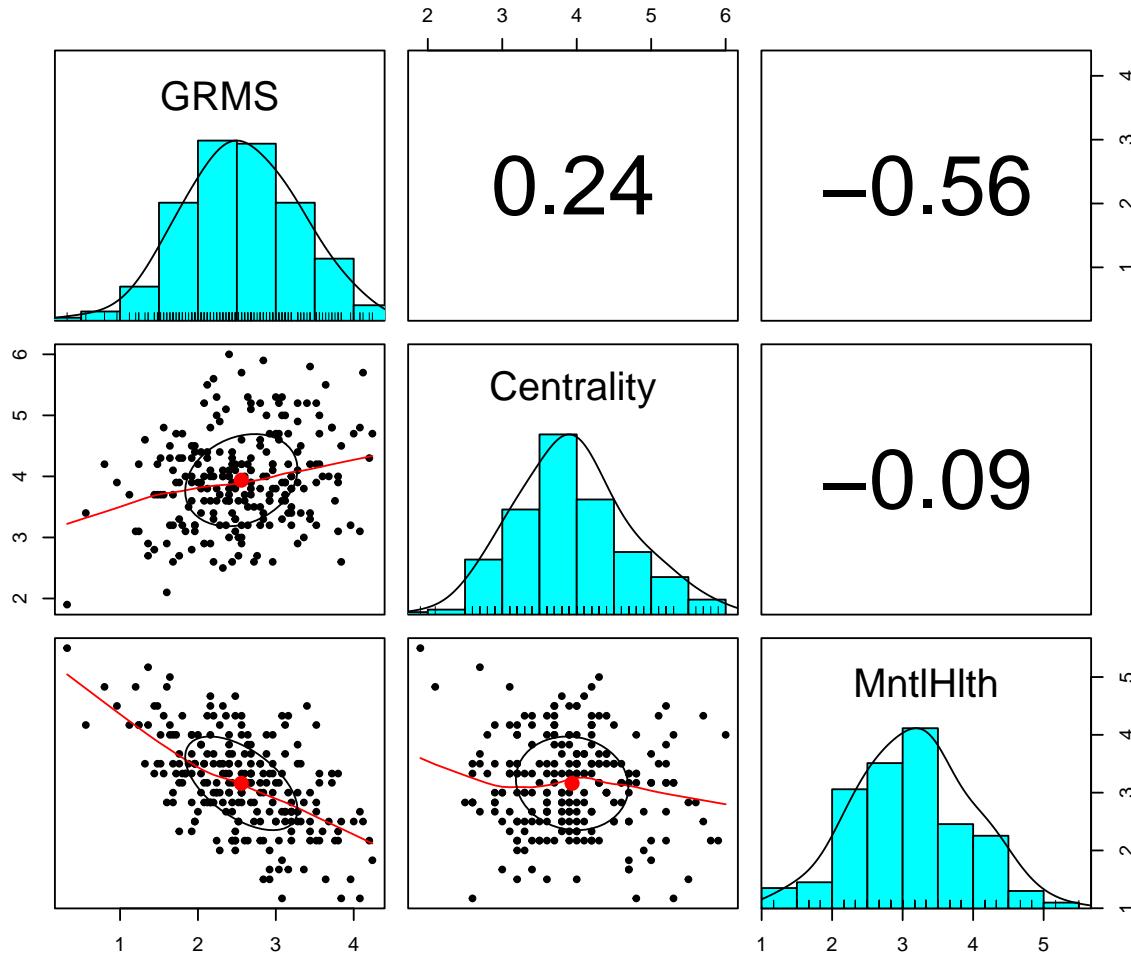
Note. M and SD are used to represent mean and standard deviation, respectively. Values in square brackets indicate the 95% confidence interval.

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

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

The `psych::pairs.panels` function provides another view.

```
library(psych)
psych::pairs.panels(Lewis_df)
```



While they are not exact, they approximate the magnitude and patterns in the correlation matrix in the article [Lewis et al., 2017].

The Lewis et al. [2017] article included a moderated mediation. Within this larger model were two moderated paths. As we work up to analyzing that moderated mediation, we will work a simple moderation predicting mental health from gendered racial microaggressions, moderated by gendered racial identity centrality.

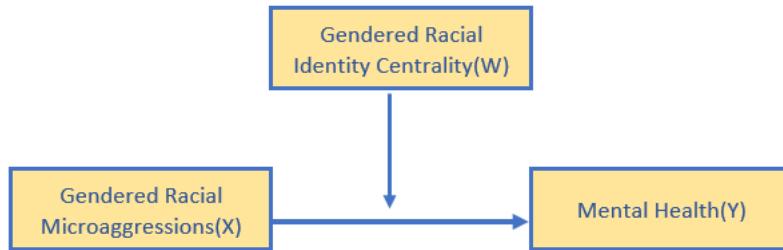


Figure 7.5: Conceptual diagram of a proposed simple moderation model using Kim et al. data

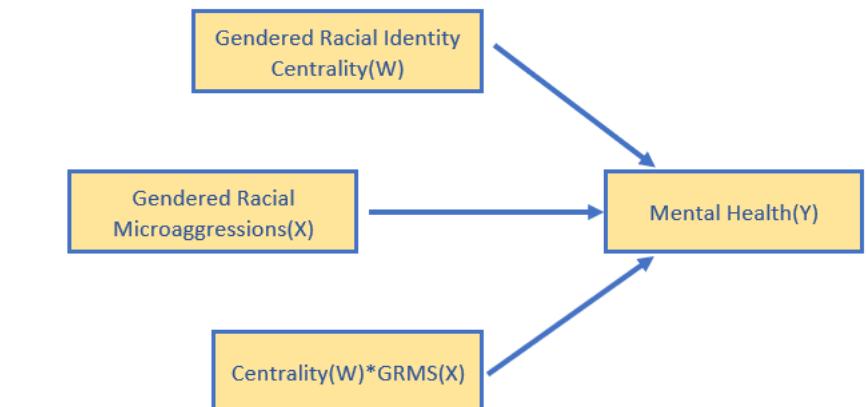


Figure 7.6: Statistical diagram of a proposed simple moderation model using Kim et al. data

Here is the formulaic rendering:

$$Y = i_Y + b_1X + b_2W + b_3XW + e_Y$$

7.7 Working the Simple Moderation with OLS and MLE

7.7.1 OLS with *lm()*

In this demonstration we will use the *lm()* function in base R to evaluate gendered racial identity centrality (Centrality) as a moderator to the relationship between gendered racial microaggressions (GRMS) on mental health (MntlHlth). Ordinary least squares is the estimator used in *lm()*. We will probe the moderating effect with both pick-a-point and Johnson-Neyman approaches.

Let's specify this simple moderation model with base R's *lm()* function. We'll use the *jtools::summ* function to produce a journal-friendly table and *interactions::interaction_plot* for information rich figures.

```
LewisSimpMod <- lm(MntlHlth ~ GRMS * Centrality, data = Lewis_df)
# the base R output if you prefer this view summary(LewisSimpMod)
```

Table 3

```
LewisSimpMod_summ <- jtools::summ(LewisSimpMod, digits = 3)
LewisSimpMod_summ
```

Observations	231
Dependent variable	MntlHlth
Type	OLS linear regression

F(3,227)	37.386
R ²	0.331
Adj. R ²	0.322

	Est.	S.E.	t val.	p
(Intercept)	6.138	0.767	8.007	0.000
GRMS	-1.248	0.290	-4.299	0.000
Centrality	-0.351	0.199	-1.764	0.079
GRMS:Centrality	0.157	0.073	2.132	0.034

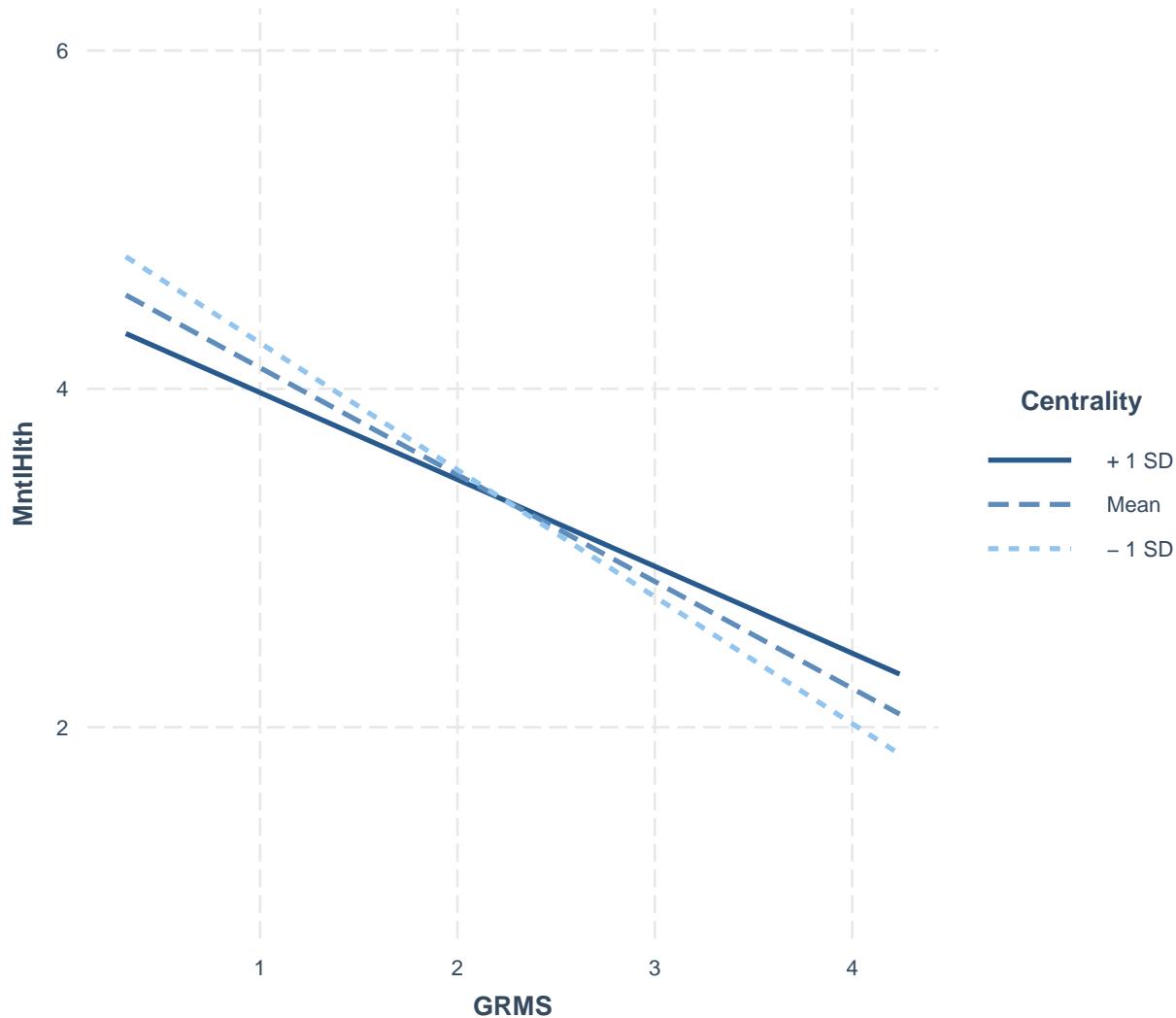
Standard errors: OLS

The following code can export the OLS regression results into a .csv. This can be opened with Excel for use in table-making. Note that this makes use of the *broom* package.

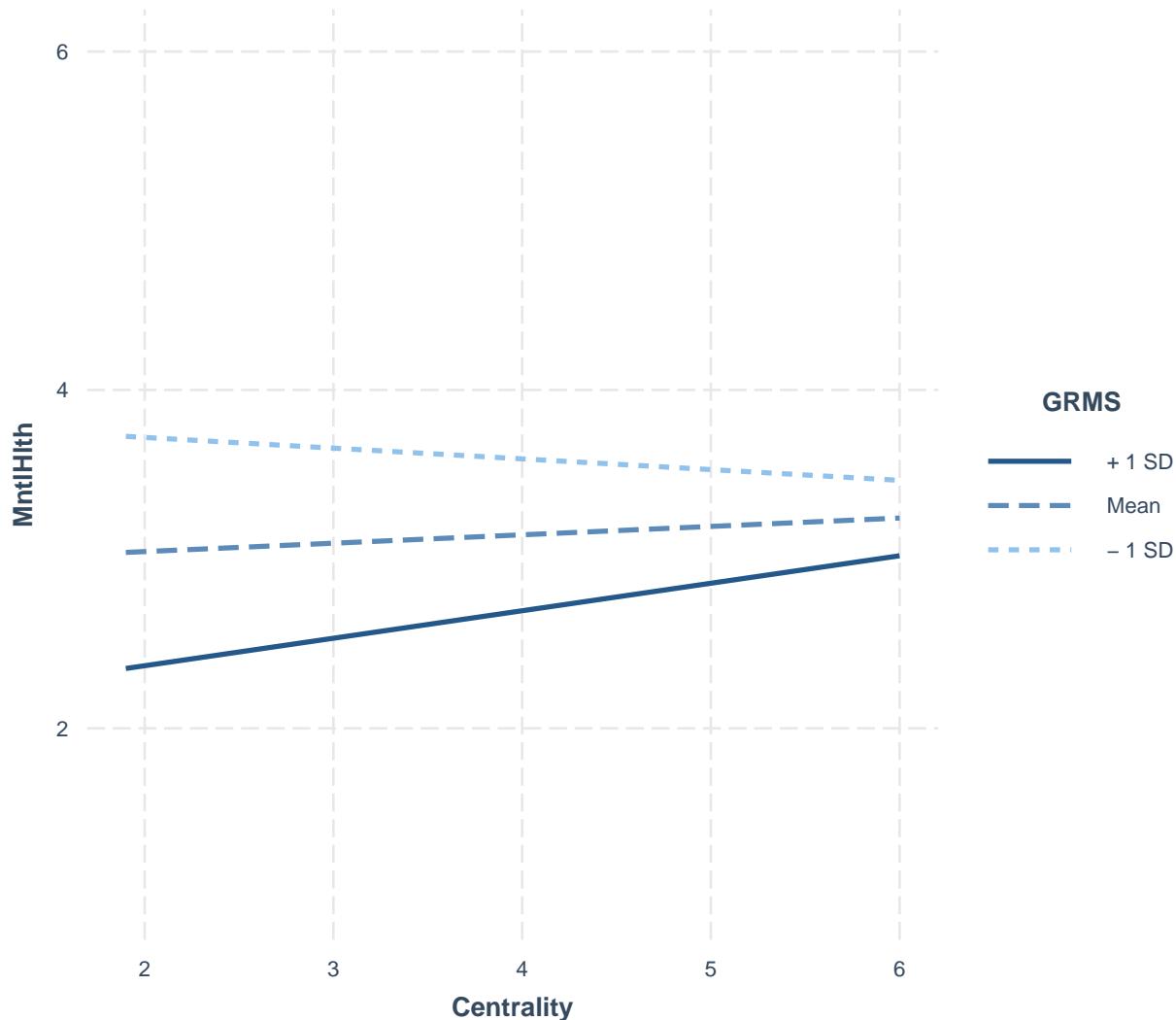
```
LewSimpModOLS <- as.data.frame(broom::tidy(LewisSimpMod))
write.csv(LewSimpModOLS, "LewSimpModOLS.csv")
```

Looking at these results we can see that the predictors account for about 33% of variance in anxiety. Further, there is a statistically significant interaction of GRMS and Centrality on MntlHlth. The *interaction_plot()* function from the package, *interactions* can illustrate these effects. In the case of interactions/moderations, I like to run them “both ways” to see which makes more sense.

```
interactions::interaction_plot(LewisSimpMod, pred = GRMS, modx = Centrality) +
  ylim(1, 6)
```



```
interactions::interaction_plot(LewisSimpMod, pred = Centrality, modx = GRMS) +
  ylim(1, 6)
```



The first figure (where Centrality is the moderator) illustrates that the slope representing the effect of gendered racial microaggressions on mental health is steepest for those with the lowest levels of gendered racial identity centrality. In fact, those with the lowest levels of gendered racial identity centrality have the highest mental health (when gendered racial microaggressions are low) and the lowest mental health (when gendered racial microaggressions are high). In contrast, the slope is less steep for those with the highest levels of gendered racial identity centrality.

The second figure represents the same data, but positions gendered racial microaggressions as the moderator. Here we really see the effect of gendered racial microaggressions as both a main effect (i.e., alone, it had a statistically significant effect on mental health) and as it interacts with gendered racial identity centrality. Those who experience the lowest levels of gendered racial microaggressions have the highest levels of mental health and there *appears to be* (we'll need to check the simple slopes to be certain) a negative slope such that mental health scores are lower as centrality scores increase. In contrast, those reporting the highest levels of microaggressions have the highest mental health scores. However, there is a positive slope such that mental health scores increase (again we'll need to check the simple slopes to see if this is a statistically significant increase) with centrality.

Next, let's probe the interaction with simple slopes. With these additional inferential tests we can see where in the distribution of the moderator, X has an effect on Y that is different from zero (and

where it does not). There are two common approaches.

The Johnson-Neyman is a *floodlight* approach and provides an indication of the places in the distribution of W (moderator) that X has an effect on Y that is different than zero. The *analysis of simple slopes* or a *spotlight* approach, probes the distribution at specific values (often the $M +/- 1SD$).

This first analysis corresponds with the first plot, where centrality is the moderator.

```
interactions::sim_slopes(LewisSimpMod, pred = GRMS, modx = Centrality)
```

JOHNSON-NEYMAN INTERVAL

When Centrality is OUTSIDE the interval [5.92, 58.37], the slope of GRMS is $p < .05$.

Note: The range of observed values of Centrality is [1.90, 6.00]

SIMPLE SLOPES ANALYSIS

Slope of GRMS when Centrality = 3.182522 (- 1 SD):

Est.	S.E.	t val.	p
<hr/>			
-0.75	0.08	-9.36	0.00

Slope of GRMS when Centrality = 3.938095 (Mean):

Est.	S.E.	t val.	p
<hr/>			
-0.63	0.06	-10.01	0.00

Slope of GRMS when Centrality = 4.693668 (+ 1 SD):

Est.	S.E.	t val.	p
<hr/>			
-0.51	0.09	-5.85	0.00

The Johnson-Neyman in this case is a bit tricky to interpret. It tells us that the slope representing the effect of GRMS on mental health is statistically significant when the value of centrality is *outside* the values of 5.92 to 58.37. Curiously, our centrality values ranged from 1 to 6. Thus, in our sample, there would be a statistically significant effect of GRMS on mental health at nearly all levels of centrality.

I find the simple slopes analysis to be easier to read. Here, we are presented the regression coefficient representing the effect of GRMS on mental health at three levels of centrality (i.e., mean and +/- 1SD). With all p values less than 0.05, GRMS has a statistically significant effect on mental health irrespective of the level of gendered racial identity centrality.

If we switch the roles of the independent and moderator values, we can see the same data, differently.

```
interactions::sim_slopes(LewisSimpMod, pred = Centrality, modx = GRMS)
```

JOHNSON-NEYMAN INTERVAL

When GRMS is OUTSIDE the interval [-2.70, 3.17], the slope of Centrality is $p < .05$.

Note: The range of observed values of GRMS is [0.32, 4.24]

SIMPLE SLOPES ANALYSIS

Slope of Centrality when GRMS = 1.835658 (- 1 SD):

Est.	S.E.	t val.	p
-0.06	0.08	-0.78	0.44

Slope of Centrality when GRMS = 2.557056 (Mean):

Est.	S.E.	t val.	p
0.05	0.06	0.83	0.41

Slope of Centrality when GRMS = 3.278455 (+ 1 SD):

Est.	S.E.	t val.	p
0.16	0.08	2.06	0.04

Again, the Johnson-Neyman can be a little tricky to interpret. Our GRMS scores could range from 0 to 5. Keeping this range in mind, we know that centrality has a statistically significant effect on mental health when centrality scores are 3.17 or greater.

This is consistent with the simple slopes results where the statistically significant effect of centrality on mental health is observed when GRMS levels are one standard deviation above the mean.

To write up these results you would report the follow-up analysis that is consistent with how you stated the hypothesis. In this case we evaluated “the moderating effect of gendered racial identity centrality on the relationship between gendered racial microaggressions on mental health.” Correspondingly, we would show the first figure and the first simple slopes analyses.

7.7.1.1 An APA Style Write-up of OLS results

Method/Aalytic Strategy

Data were analyzed with an ordinary least squares approach with the base R (v. 4.3.1) function, *lm()*. We specified a model predicting mental health (MntlHlth) from the interacting effects of gendered racial microaggressions (GRMS) and gendered racial identity centrality (Centrality).

Results

Preliminary Analyses

- Missing data analyses and managing missing data
- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

Primary Analyses A multiple regression analysis was conducted to predict mental health from gendered racial microaggressions, moderated by gendered racial identity centrality. Results supported a statistically significant interaction effect that accounted for 33% of the variance ($B = 0.157$, $SE = 0.073$, $t = 0.034$). Probing the interaction effect with Johnson-Neyman and analysis of simple slopes approaches indicated that the relationship between gendered racial microaggressions and mental health is statistically significant throughout the range of gendered racial identity centrality. Results are listed in Table 1 and illustrated in Figure 1.

7.7.2 MLE with *lavaan::sem()*

Let's specify this same problem with a path analysis (i.e., using manifest or observed variables) in *lavaan*. There are a few things to note:

- The code below “draws our model.” It opens and close with ‘ marks
- “Labels” (e.g., b1, b2) are useful for identifying the paths.
 - Later, in SEM/CFA (latent variable modeling) we can use them to “fix and free” constraints; the asterisk makes them look like interactions, but they are not
- Interactions are specified with a colon
- We can use hashtags internal to the code to make notes to ourselves (or, in the case where your script will be available in an open science repository, inform others of your thought process)
- Following the specification of the model, we use the lavaan function *sem()* to conduct the estimation
 - adding *missing = ‘fiml’* is the magic we have been waiting for with regard to missing data
 - bootstrapping is an MLE tool that gives us greater power (more later in mediation)
 - the *summary()* and *parameterEstimates()* functions get us the desired output

```
LewisSimpModMLE <- "
  MntlHlth ~ b1*GRMS + b2*Centrality + b3*GRMS:Centrality

  #intercept (constant) of MntlHlth
  MntlHlth ~ MntlHlth.mean*1
  #mean of W (Centrality, in this case) for use in simple slopes
  Centrality ~ Centrality.mean*1
  #variance of W (Centrality, in this case) for use in simple slopes
```

```

Centrality ~~Centrality.var*Centrality

#simple slopes
SD.below := b1 + b3*(Centrality.mean - sqrt(Centrality.var))
mean := b1 + b3*(Centrality.mean)
SD.above := b1 + b3*(Centrality.mean + sqrt(Centrality.var))
"
set.seed(230925) #needed for reproducibility especially when asking for bootstrapped confidence intervals
LewMLEfit <- lavaan::sem(LewisSimpModMLE, data = Lewis_df, missing = "fiml",
  se = "bootstrap", bootstrap = 1000)

```

Warning in lav_partable_vnames(FLAT, "ov.x", warn = TRUE): lavaan WARNING:
 model syntax contains variance/covariance/intercept formulas
 involving (an) exogenous variable(s): [Centrality]; These
 variables will now be treated as random introducing additional
 free parameters. If you wish to treat those variables as fixed,
 remove these formulas from the model syntax. Otherwise, consider
 adding the fixed.x = FALSE option.

```

LewisMLEsummary <- lavaan::summary(LewMLEfit, standardized = TRUE, fit = TRUE,
  ci = TRUE)
LewisMLEParamEsts <- lavaan::parameterEstimates(LewMLEfit, boot.ci.type = "bca.simple",
  standardized = TRUE)
LewisMLEsummary

```

lavaan 0.6.16 ended normally after 14 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	7
Number of observations	231
Number of missing patterns	1

Model Test User Model:

Test statistic	567.225
Degrees of freedom	2
P-value (Chi-square)	0.000

Model Test Baseline Model:

Test statistic	659.975
Degrees of freedom	5
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.137
Tucker-Lewis Index (TLI)	-1.157
Robust Comparative Fit Index (CFI)	0.137
Robust Tucker-Lewis Index (TLI)	-1.157

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-494.947
Loglikelihood unrestricted model (H1)	-211.334
Akaike (AIC)	1003.894
Bayesian (BIC)	1027.991
Sample-size adjusted Bayesian (SABIC)	1005.805

Root Mean Square Error of Approximation:

RMSEA	1.106
90 Percent confidence interval - lower	1.031
90 Percent confidence interval - upper	1.184
P-value H_0: RMSEA <= 0.050	0.000
P-value H_0: RMSEA >= 0.080	1.000
Robust RMSEA	1.106
90 Percent confidence interval - lower	1.031
90 Percent confidence interval - upper	1.184
P-value H_0: Robust RMSEA <= 0.050	0.000
P-value H_0: Robust RMSEA >= 0.080	1.000

Standardized Root Mean Square Residual:

SRMR	0.218
------	-------

Parameter Estimates:

Standard errors	Bootstrap
Number of requested bootstrap draws	1000
Number of successful bootstrap draws	1000

Regressions:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
MntlHlth ~						
GRMS (b1)	-1.248	0.308	-4.052	0.000	-1.820	-0.550
Centrality (b2)	-0.351	0.205	-1.712	0.087	-0.713	0.097
GRMS:Cntr (b3)	0.157	0.079	1.984	0.047	-0.016	0.304
Std.lv Std.all						

```
-1.248 -1.033
-0.351 -0.304
 0.157  0.684
```

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
.MntlHlt (MnH.)	6.138	0.783	7.840	0.000	4.285	7.481
Cntrlty (Cnt.)	3.938	0.049	79.569	0.000	3.836	4.035
Std.lv Std.all						
	6.138	7.054				
	3.938	5.223				

Variances:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
Cntrlty (Cnt.)	0.568	0.054	10.608	0.000	0.464	0.675
.MntlHlt	0.438	0.040	10.834	0.000	0.357	0.510
Std.lv Std.all						
	0.568	1.000				
	0.438	0.578				

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
SD.below	-0.750	0.078	-9.605	0.000	-0.884	-0.570
mean	-0.632	0.059	-10.760	0.000	-0.742	-0.511
SD.above	-0.514	0.088	-5.812	0.000	-0.688	-0.341
Std.lv Std.all						
	-0.750	1.855				
	-0.632	2.539				
	-0.514	3.223				

LewisMLEParamEsts

	lhs op	rhs
1	MntlHlth ~	GRMS
2	MntlHlth ~	Centrality
3	MntlHlth ~	GRMS:Centrality
4	MntlHlth ~1	
5	Centrality ~1	
6	Centrality ~~	Centrality
7	MntlHlth ~~	MntlHlth
8	GRMS ~~	GRMS
9	GRMS ~~	GRMS:Centrality
10	GRMS:Centrality ~~	GRMS:Centrality
11	GRMS ~1	
12	GRMS:Centrality ~1	
13	SD.below := b1+b3*(Centrality.mean-sqrt(Centrality.var))	

```

14      mean := b1+b3*(Centrality.mean)
15      SD.above := b1+b3*(Centrality.mean+sqrt(Centrality.var))
          label   est    se     z pvalue ci.lower ci.upper std.lv std.all
1        b1 -1.248 0.308 -4.052 0.000 -1.822 -0.550 -1.248 -1.033
2        b2 -0.351 0.205 -1.712 0.087 -0.706 0.118 -0.351 -0.304
3        b3  0.157 0.079  1.984 0.047 -0.013 0.305 0.157 0.684
4  MntlHlth.mean  6.138 0.783  7.840 0.000  4.191 7.438 6.138 7.054
5  Centrality.mean 3.938 0.049 79.569 0.000  3.834 4.035 3.938 5.223
6  Centrality.var  0.568 0.054 10.608 0.000  0.469 0.682 0.568 1.000
7                    0.438 0.040 10.834 0.000  0.372 0.536 0.438 0.578
8                    0.518 0.000     NA     NA  0.518 0.518 0.518 1.000
9                    2.334 0.000     NA     NA  2.334 2.334 2.334 0.853
10                   14.446 0.000     NA     NA 14.446 14.446 14.446 1.000
11                   2.557 0.000     NA     NA  2.557 2.557 2.557 3.552
12                   10.199 0.000     NA     NA 10.199 10.199 10.199 2.683
13      SD.below -0.750 0.078 -9.605 0.000 -0.887 -0.574 -0.750 1.855
14      mean -0.632 0.059 -10.760 0.000 -0.742 -0.513 -0.632 2.539
15      SD.above -0.514 0.088 -5.812 0.000 -0.688 -0.341 -0.514 3.223

std.nox
1  -1.435
2  -0.290
3  0.180
4  7.054
5  5.223
6  1.000
7  0.578
8  0.518
9  2.334
10 14.446
11 2.557
12 10.199
13 -0.675
14 -0.495
15 -0.315

# adding rsquare=TRUE or rsq=T to both summary and parameterEstimates
# resulted in an error related to missing values in row names; could
# not find a solution

```

For reasons unknown to me, I haven't been able to use the commands to produce r-square values without receiving errors. Fortunately, there is a workaround and we can call for the r-square results directly.

```
lavaan:::lavInspect(LewMLEfit, "rsquare")
```

```
MntlHlth
0.422
```

Our model accounts for 42% of the variance in mental health.

To create a table outside of R, you can export these results as a .csv file (which can be opened in Excel).

```
write.csv(LewisMLEParamEsts, file = "LewisMLEParamEsts.csv")
```

Recall, this was our formula:

Here is the formulaic rendering:

$$Y = i_Y + b_1X + b_2W + b_3XW + e_Y$$

Looking at our data here's what we've learned:

$$\hat{Y} = 6.138 + (-1.248)X + (-0.351)W + 1.57XW$$

While the p values will wiggle around, it is reassuring that the regression weights are consistent across the OLS and MLE results. It is typical for the MLE p values to be less significant. This is, in part, due to the large sample size nature of this approach to data analysis.

We can use our *lavaan* output to create a figure that is typical of path and structural equation modeling analyses. We start by feeding the *tidySEM::graph_model* function the fit object. The function will make it's best guess for a figure. Typically, we will update it.

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
```

```
This is lavaan 0.6-16
lavaan is FREE software! Please report any bugs.
```

```
Attaching package: 'lavaan'
```

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

```
cor2cov
```

```
library(dplyr)
library(ggplot2)
library(tidySEM)
```

```
Loading required package: OpenMx
```

```
Attaching package: 'OpenMx'
```

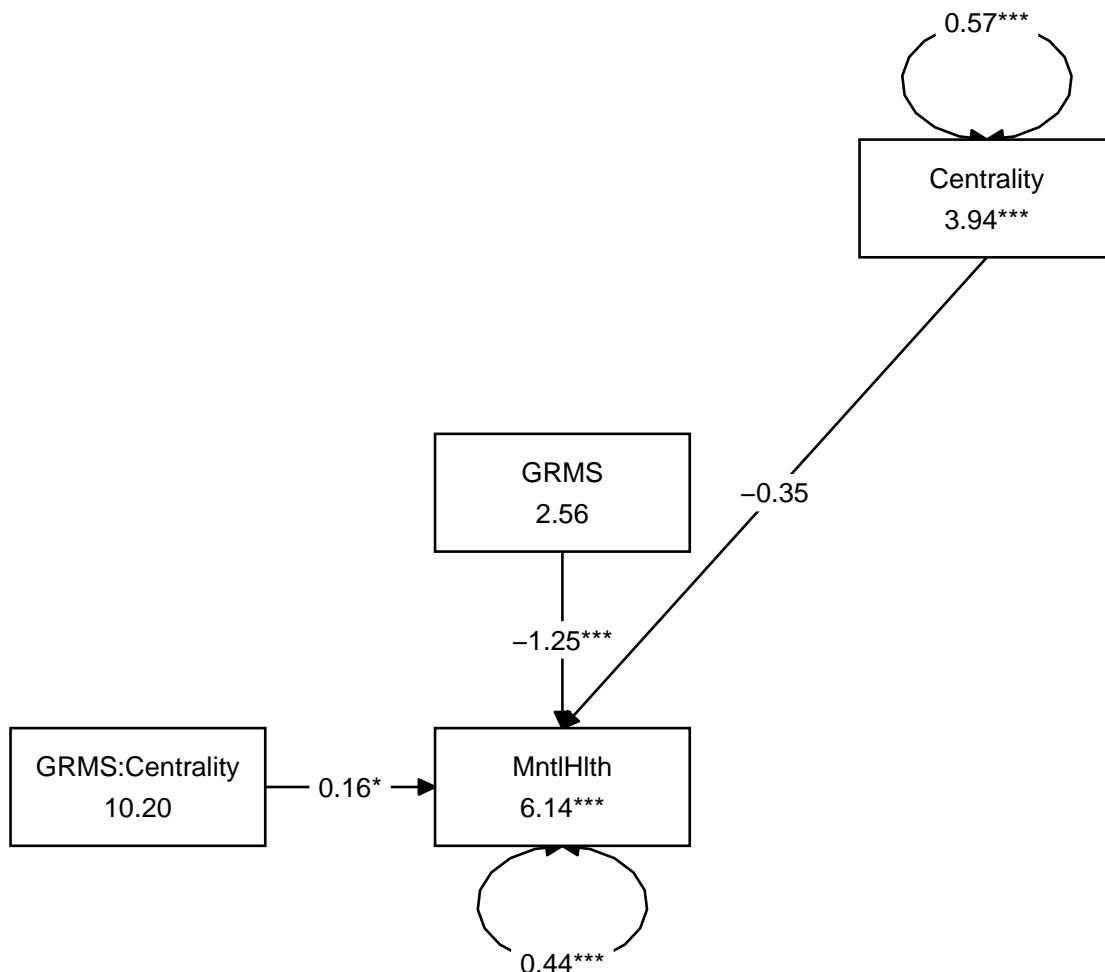
The following object is masked from 'package:psych':

```
tr
```

Registered S3 method overwritten by 'tidySEM':

```
method      from
predict.MxModel OpenMx
```

```
tidySEM::graph_sem(model = LewMLEfit)
```



We can use the `tidySEM::get_layout` function to understand how our model is being mapped.

```
tidySEM::get_layout(LewMLEfit)
```

	[,1]	[,2]	[,3]
[1,]	"GRMS:Centrality"	NA	"Centrality"
[2,]	"MntlHlth"	"GRMS"	NA

```
attr(,"class")
[1] "layout_matrix" "matrix"      "array"
```

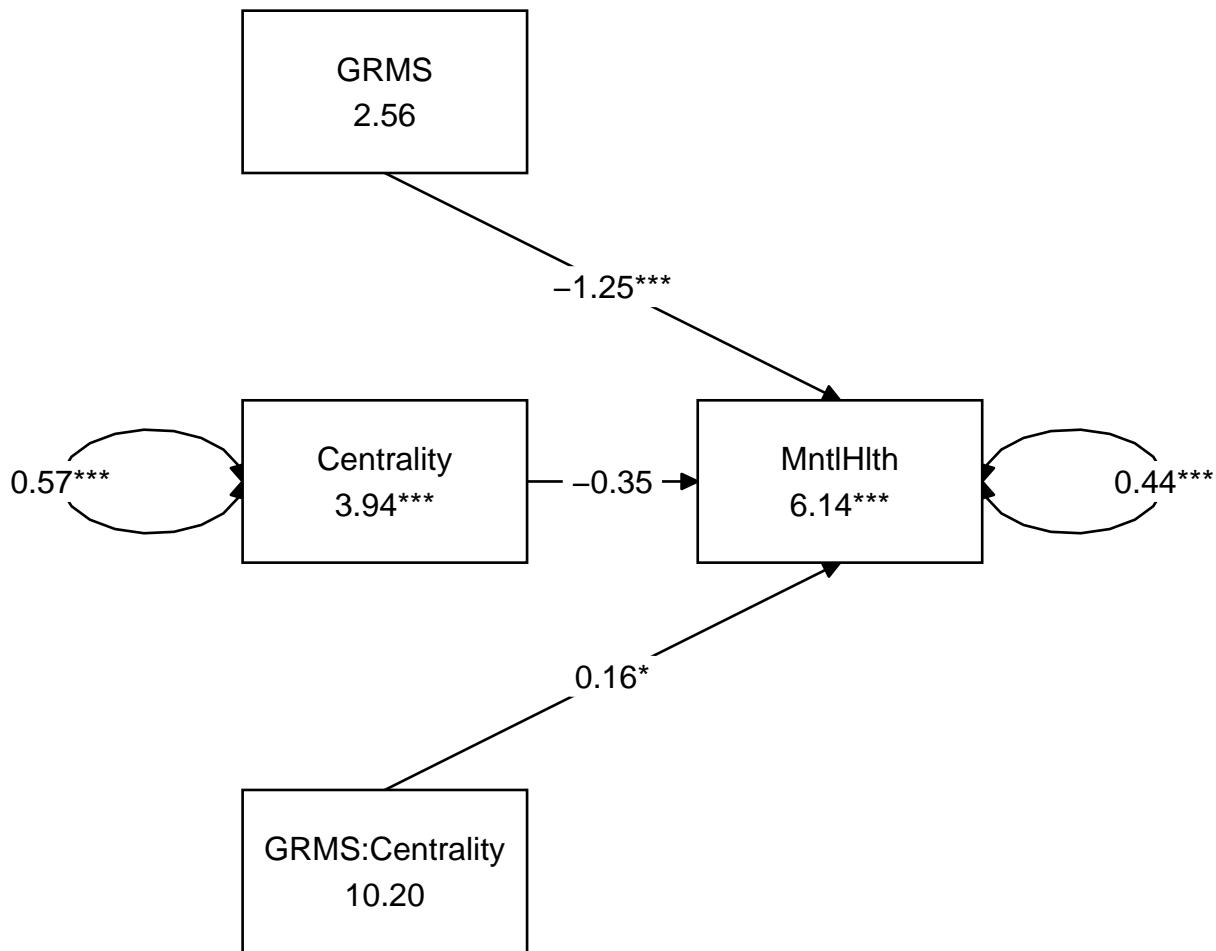
We can write code to remap them

```
mod_map <- tidySEM::get_layout("GRMS", "", "Centrality", "MntlHlth", "GRMS:Centrality",
                               "", rows = 3)
mod_map
```

```
[,1] [,2]
[1,] "GRMS" ""
[2,] "Centrality" "MntlHlth"
[3,] "GRMS:Centrality" ""
attr(,"class")
[1] "layout_matrix" "matrix"      "array"
```

We can update the *tidySEM::graph_sem* function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(LewMLEfit, layout = mod_map, rect_width = 1.25, rect_height = 1.25,
                    spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



If I had just run this with lavaan, I would want to plot the interaction and would do so with the OLS methods I demonstrated above.

7.7.3 Tabling the data

In this table, I gather the output from both the OLS and MLE approaches. Youll notice below that the B weights are identical to the third decimal place (shown). The standard errors and p values wiggle around a bit, but are consistent with each other (and lead to the same significant/non-significant conclusion). The R^2 values are different by nearly 10%.

Further comparison shows that the OLS output provides an F statistic that indicates whether or not the overall model is significant. These are commonly reported in Results. In contrast, the MLE output has a page or more of *fit statistics* (e.g., CFI, RMSEA, Chi-square goodness of fit) that are commonly reported in latent variable modeling such as SEM and CFA. Although some researchers will report them in path analysis, I tend to preer the focus on the strength and significance of the regression weights.

Table 4

A Comparison of OLS and MLE Regression Results

	OLS with the <i>lm()</i> in base R			MLE with <i>lavaan</i>		
	B	SE	p	B	SE	p
MntlHlth	6.138	0.767	0.000	6.138	0.783	<0.001
(Intercept)						
GRMS (X)	-1.248	0.290	0.000	-1.248	0.308	<0.001
Centrality (W)	-0.351	0.199	0.079	-0.351	0.205	0.087
GRMS:GRIC (XY)	0.157	0.073	0.034	0.157	0.079	0.047
<i>R</i> ²			<i>R</i> ²			
0.331			0.422			

7.7.4 APA Style Writeup

Method/Aalytic Strategy

Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16). We specified a model predicting mental health (MntlHlth) from the interacting effects of gendered racial microaggressions (GRMS) and gendered racial identity centrality (GRIC).

Results

Preliminary Analyses

- Missing data analyses and managing missing data
- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

Primary Analyses A multiple regression analysis was conducted to predict anxiety from racial and ethnic microaggressions and attitudes toward help-seeking. Results supported a model with a statistically significant interaction effect that accounted for 42% of the variance. Probing the interaction effect with a simple slopes analysis indicated that the relationship between gendered racial microaggressions and mental health was significant throughout the centrality distribution (i.e., $M \pm 1SD$). Results are listed in Table 2. The effect of the significant interaction can be seen in Figure 1 where the slope of the gendered racial microaggressions and mental health relationship is sharpest for those with the lowest levels of gendered racial identity centrality.

7.8 STAY TUNED

A section on power analysis is planned and coming soon! My apologies that it's not quite Ready.

7.9 Residual and Related Questions...

Wait. Why did we do this? And which would you use when?

- As we transition from NHST to statistical modeling we also (generally) transition between OLS and MLE.
 - I would use OLS with
 - smaller sample sizes
 - straightforward regression models (linear, multiple, simultaneous, hierarchical)
- I would use MLE with
 - nonlinear models
 - models involving latent variables
 - models with indirect effects
 - (larger sample sizes is prerequisite)

7.10 Practice Problems

The suggested practice problem for this chapter is to conduct a simple moderation with both the OLS(i.e., `lm()`) approach and the MLE(i.e., `lavaan`) approach and compare the results.

7.10.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed

If this topic feels a bit overwhelming, simply change the random seed in the data simulation, then rework the problem. This should provide minor changes to the data (maybe in the second or third decimal point), but the results will likely be very similar.

7.10.2 Problem #2: Rework the research vignette, but swap one or more variables

Use the simulated data, but select one of the other models that was evaluated in the Lewis et al. [2017] study.

7.10.3 Problem #3: Use other data that is available to you

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 the simple moderation with both approaches.

7.10.4 Grading Rubric

Regardless of your choic(es) complete all the elements listed in the grading rubric.

Assignment Component		
1. Assign each variable to the X, Y, and W roles	5	_____
2. Import the data and format the variables in the model	5	_____
3. Specify and run the OLS (i.e., <i>lm()</i>) model	5	_____
4. Probe the interaction with the simple slopes and Johnson-Neyman approaches	5	_____
5. Create an interaction figure	5	_____
6. Create a table (a package-produced table is fine)	5	_____
7. Create an APA style write-up of the results	5	_____
8. Repeat the analysis in <i>lavaan</i> (specify the model to include probing the interaction)	5	_____
9. Create a model figure	5	_____
10. Create a table	5	_____
11. Note similarities and differences in the OLS results	5	_____
12. Represent your work in an APA-style write-up	5	_____
13. Explanation to grader	5	_____
Totals	65	_____

7.11 Bonus Track:

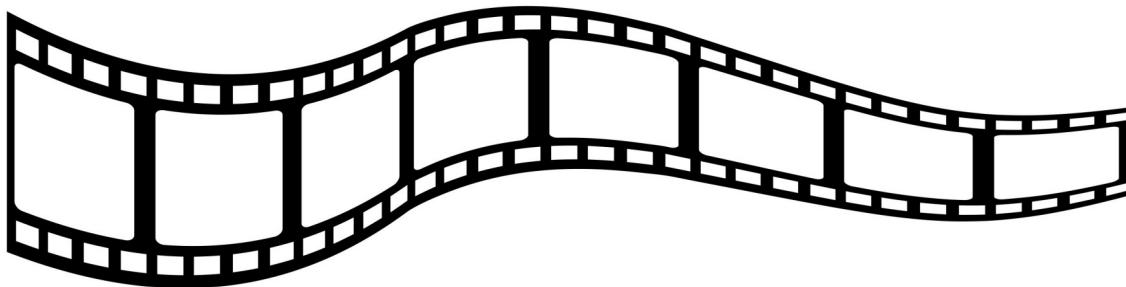


Figure 7.7: Image of a filmstrip

Below is template for a simple moderation conducted with the OLS approach using the base R function, *lm()*

```
library(jtools) #the summ function creates a terrific regression table
```

```
Attaching package: 'jtools'
```

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

```
get_data

library(interactions)
library(ggplot2)

# The regression OLSmodel <- lm(Y~X*W, data=my_df) summary(OLSmodel)

# Cool Table summ(OLSmodel, digits = 3)

# Probe Simple Slopes sim_slopes(OLSmodel, pred = X, modx = W)

# Figures interact_plot(OLSmodel, pred = W, modx = X)
# interact_plot(OLSmodel, pred = X, modx = W)
```

Below is a template for a simple moderation conducted with the MLE approach using the package, *lavaan*.

```
library(lavaan)
# set.seed(210501)#needed for reproducibility because lavaan
# introduces randomness in the calculations MLEmodel <- ' Y ~ b1*X +
# b2*W + b3*X:W intercept (constant) of Y Y ~ Y.mean*1 mean of W for
# use in simple slopes W ~ W.mean*1 variance of W for use in simple
# slopes W ~ W .var*W

# simple slopes SD.below := b1 + b3*(W.mean - sqrt(W.var)) mean := b1
# + b3*(W.mean) SD.above := b1 + b3*(W.mean + sqrt(W.var))
#'
# MLEmod_fit <- sem(MLEmodel, data = my_df, missing = 'fiml', se =
# 'bootstrap', bootstrap = 1000) MLEmod_fit_summary <-
# summary(MLEmod_fit, standardized = TRUE, rsq=T, ci=TRUE)
# MLEmodParamEsts <- parameterEstimates(MLEmod_fit, boot.ci.type =
# 'bca.simple', standardized=TRUE) MLEmod_fit_summary MLEmodParamEsts
```

7.12 Homeworked Example

Screencast Link

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson in ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to conduct a simple moderation (i.e., moderated regression) with both ordinary least squares (i.e., with the *lm()* function in base R) and maximum likelihood estimators (i.e., with the *lavaan::sem* function package) and compare the results.

Assign each variable to the X, Y, and W roles

Is the effect of centering on perceived value to the student moderated by socially responsive pedagogy?

- X = Centering, pre/re (0,1)
- W = Socially responsive pedagogy (1 to 4 scaling)
- Y = Value to the student (1 to 4 scaling)

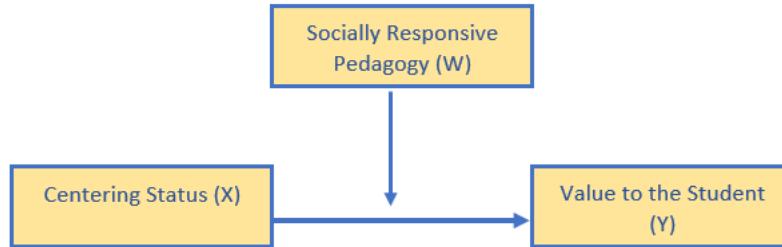


Figure 7.8: An image of the conceptual model of simple moderation for the homeworked example.

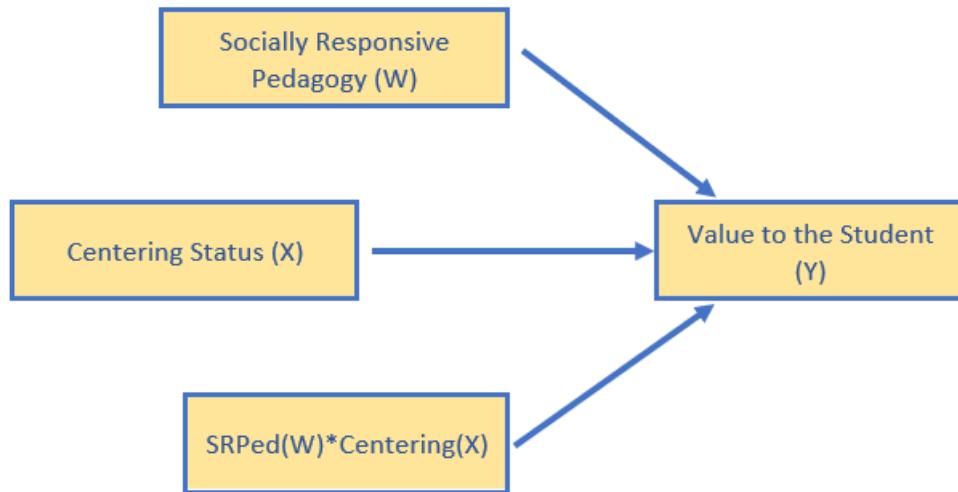


Figure 7.9: An image of the statistical model of simple moderation for the homeworked example.

Import the data and format the variables in the model

```
raw <- readRDS("ReC.rds")
```

The approach we are taking to complex mediation does not allow dependency in the data. Therefore, we will include only those who took the multivariate class (i.e., excluding responses for the ANOVA and psychometrics courses).

```
raw <- (dplyr::filter(raw, Course == "Multivariate"))
```

I need to score the SRPed and Valued variables

```
Valued_vars <- c("ValObjectives", "IncrUnderstanding", "IncrInterest")
raw$Valued <- sjstats::mean_n(raw[, ..Valued_vars], 0.75)

SRPed_vars <- c("InclusvClassrm", "EquitableEval", "MultPerspectives",
  "DEIntegration")
raw$SRPed <- sjstats::mean_n(raw[, ..SRPed_vars], 0.75)
```

I will create a babydf.

```
babydf <- dplyr::select(raw, Centering, Valued, SRPed)
```

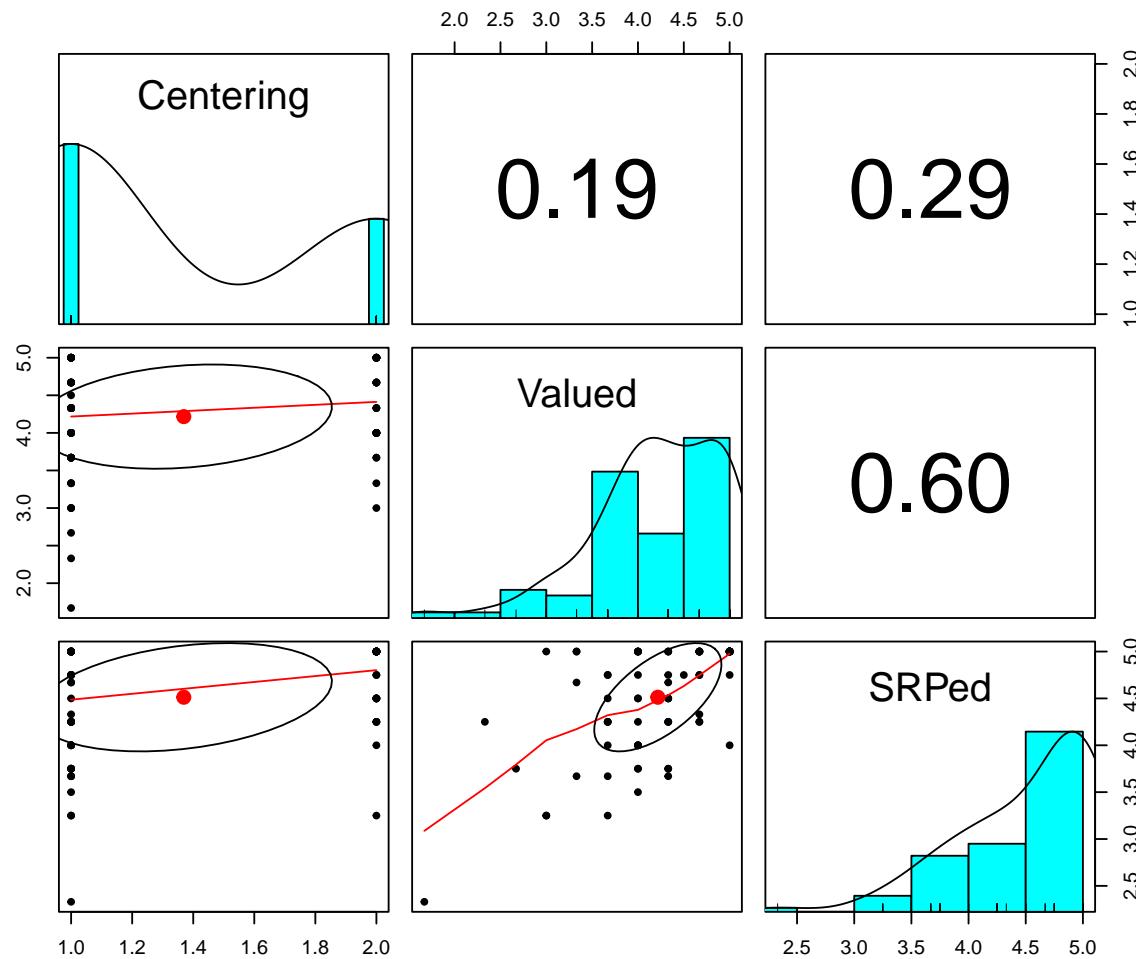
Let's check the structure of the variables:

```
str(babydf)
```

```
Classes 'data.table' and 'data.frame':  84 obs. of  3 variables:
 $ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
 $ Valued   : num  4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
 $ SRPed    : num  4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Quick peek at relations between variables:

```
psych::pairs.panels(babydf)
```



Specify and run the OLS/*lm()* model

```
ReC_SimpMod <- lm(Valued ~ Centering * SRPed, data = babydf)
# the base R output if you prefer this view
summary(ReC_SimpMod)
```

```
Call:
lm(formula = Valued ~ Centering * SRPed, data = babydf)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.7173 -0.3092  0.1286  0.4027  1.1286 

Coefficients:
```

Coefficients:

```

              Estimate Std. Error t value   Pr(>|t|)    
(Intercept)      1.0567    0.5631   1.876    0.0644 .  
CenteringRe     -0.5703    1.3183  -0.433    0.6665  
SRPed          0.7037    0.1272   5.530 0.000000422 *** 
CenteringRe:SRPed 0.1185    0.2813   0.421    0.6748  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.5606 on 77 degrees of freedom
(3 observations deleted due to missingness)
Multiple R-squared:  0.3674,    Adjusted R-squared:  0.3427 
F-statistic: 14.9 on 3 and 77 DF,  p-value: 0.0000009678

```

Although there is a statistically significant main effect for socially responsive pedagogy, all other effects (including the moderation effect) is non-significant. If this were “real research” we might stop, but let’s continue.

The following code can export the OLS regression results into a .csv. This can be opened with Excel for use in table-making. Note that this makes use of the *broom* package.

```
ReC_SimpModOLS <- as.data.frame(broom::tidy(ReC_SimpMod))
write.csv(ReC_SimpModOLS, "ReC_SimpModOLS.csv")
```

Probe the interaction with the simple slopes and Johnson-Neyman approaches

```
interactions::sim_slopes(ReC_SimpMod, pred = SRPed, modx = Centering)
```

Warning: Johnson-Neyman intervals are not available for factor moderators.

SIMPLE SLOPES ANALYSIS

Slope of SRPed when Centering = Re:

Est.	S.E.	t val.	p
0.82	0.25	3.28	0.00

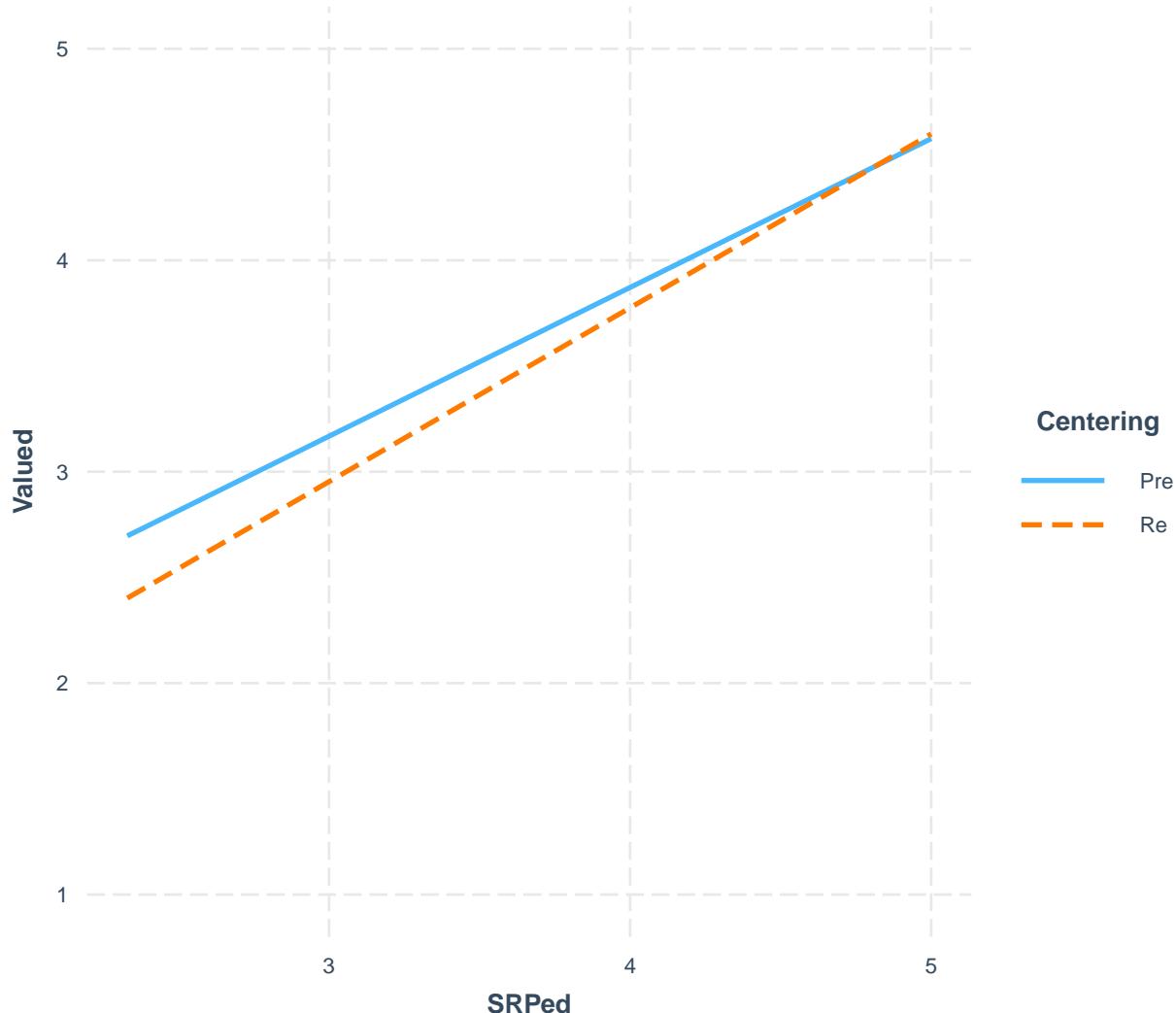
Slope of SRPed when Centering = Pre:

Est.	S.E.	t val.	p
0.70	0.13	5.53	0.00

Consistent with the main effect of socially responsive pedagogy, it has a positive effect on value at pre- and re-centered stages.

Create an interaction figure

```
library(ggplot2)
interactions:::interact_plot(ReC_SimpMod, pred = SRPed, modx = Centering) +
  ylim(1, 5)
```



```
# the following code will not run because for this function, the x
# variable cannot be a factor
# interactions:::interact_plot(ReC_SimpMod, pred = Centering, modx =
# SRPed)
```

Create a table (a package-produced table is fine)

```
ReC_SimpMod_summ <- jtools::summ(ReC_SimpMod, digits = 3)
ReC_SimpMod_summ
```

Observations	81 (3 missing obs. deleted)
Dependent variable	Valued
Type	OLS linear regression

F(3,77)	14.904
R ²	0.367
Adj. R ²	0.343

	Est.	S.E.	t val.	p
(Intercept)	1.057	0.563	1.876	0.064
CenteringRe	-0.570	1.318	-0.433	0.667
SRPed	0.704	0.127	5.530	0.000
CenteringRe:SRPed	0.118	0.281	0.421	0.675

Standard errors: OLS

Create an APA style write-up of the results

A multiple regression analysis was conducted to predict course value to the student from the centering (pre-, re-) stage, moderated by evaluation of socially responsive pedagogy. Although the model accounted for 37% of the variance, there was not a statistically significant interaction. Rather, socially responsive pedagogy had a strong main effect ($B = 0.704$, $SE = 0.127$, $p < 0.001$) that was true for both pre- and re-centered levels. Results are listed in Table 1 and illustrated in Figure 1.

Repeat the analysis in *lavaan* (specify the model to include probing the interaction)

```
str(babydf)
```

```
Classes 'data.table' and 'data.frame': 84 obs. of 3 variables:
$ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
$ Valued   : num 4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
$ SRPed    : num 4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
- attr(*, ".internal.selfref")=<externalptr>
```

```
babydf$CENTERING <- as.numeric(babydf$Centering)
babydf$CENTERING <- (babydf$CENTERING - 1)
str(babydf)
```

```
Classes 'data.table' and 'data.frame': 84 obs. of 4 variables:
 $ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
 $ Valued    : num 4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
 $ SRPed     : num 4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
 $ CENTERING: num 1 1 1 1 1 1 1 1 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

```
ReC_SimpMod_MLE <- "
  Valued ~ b1*CENTERING + b2*SRPed + b3*CENTERING:SRPed

  #intercept (constant) of Valued
  Valued ~ Valued.mean*1
  #mean of W (SRPed, in this case) for use in simple slopes
  SRPed ~ SRPed.mean*1
  #variance of W (SRPed, in this case) for use in simple slopes
  SRPed ~ SRPed.var*SRPed

  #simple slopes evaluating effect of SCRPed on Valued at each of the levels of centering
  Pre := b2 + b3*(0)
  Re := b2 + b3*(1)
"
set.seed(231002) #needed for reproducibility because lavaan introduces randomness in calculation
ReCMLEfit <- lavaan::sem(ReC_SimpMod_MLE, data = babydf, missing = "fiml",
  se = "bootstrap", bootstrap = 1000)
```

Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: 3 cases have missing data in object 'centering', which is treated as exogenous variable(s), while fixed.x = TRUE.

Warning in lav_partable_vnames(FLAT, "ov.x", warn = TRUE): lavaan WARNING:
 model syntax contains variance/covariance/intercept formulas
 involving (an) exogenous variable(s): [SRPed]; These variables
 will now be treated as random introducing additional free
 parameters. If you wish to treat those variables as fixed, remove
 these formulas from the model syntax. Otherwise, consider adding
 the fixed.x = FALSE option.

```
ReCMLEsummary <- lavaan::summary(ReCMLEfit, standardized = TRUE, fit = TRUE,
  ci = TRUE)
ReCMLEParamEsts <- lavaan::parameterEstimates(ReCMLEfit, boot.ci.type = "bca.simple",
  standardized = TRUE)
ReCMLEsummary
```

lavaan 0.6.16 ended normally after 12 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	7
	Used Total
Number of observations	81 84
Number of missing patterns	1

Model Test User Model:

Test statistic	25.909
Degrees of freedom	2
P-value (Chi-square)	0.000

Model Test Baseline Model:

Test statistic	62.994
Degrees of freedom	5
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.588
Tucker-Lewis Index (TLI)	-0.031
Robust Comparative Fit Index (CFI)	0.588
Robust Tucker-Lewis Index (TLI)	-0.031

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-136.024
Loglikelihood unrestricted model (H1)	-123.069
Akaike (AIC)	286.048
Bayesian (BIC)	302.809
Sample-size adjusted Bayesian (SABIC)	280.733

Root Mean Square Error of Approximation:

RMSEA	0.384
90 Percent confidence interval - lower	0.261
90 Percent confidence interval - upper	0.522
P-value H_0: RMSEA <= 0.050	0.000
P-value H_0: RMSEA >= 0.080	1.000
Robust RMSEA	0.384

90 Percent confidence interval - lower	0.261
90 Percent confidence interval - upper	0.522
P-value H_0: Robust RMSEA <= 0.050	0.000
P-value H_0: Robust RMSEA >= 0.080	1.000

Standardized Root Mean Square Residual:

SRMR	0.140
------	-------

Parameter Estimates:

Standard errors	Bootstrap
Number of requested bootstrap draws	1000
Number of successful bootstrap draws	1000

Regressions:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
Valued ~						
CENTERING (b1)	-0.570	1.138	-0.501	0.616	-2.882	1.692
SRPed (b2)	0.704	0.152	4.634	0.000	0.356	0.953
CENTERING (b3)	0.118	0.250	0.474	0.635	-0.373	0.643
Std.lv Std.all						
	-0.570	-0.405				
	0.704	0.594				
	0.118	0.400				

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
.Valued (Vld.)						
SRPed (SRP.)	4.512	0.061	74.224	0.000	4.389	4.635
Std.lv Std.all						
	1.057	1.553				
	4.512	7.856				

Variances:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
SRPed (SRP.)						
.Valued	0.330	0.060	5.500	0.000	0.214	0.453
Std.lv Std.all						
	0.330	1.000				
	0.299	0.645				

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper
Pre						
Re	0.704	0.152	4.632	0.000	0.356	0.953
Std.lv Std.all						
	0.822	0.199	4.124	0.000	0.368	1.182

```
0.704    0.594
0.822    0.994
```

ReCMLParamEsts

	lhs	op	rhs	label	est	se	z	pvalue
1	Valued	~	CENTERING	b1	-0.570	1.138	-0.501	0.616
2	Valued	~	SRPed	b2	0.704	0.152	4.634	0.000
3	Valued	~	CENTERING:SRPed	b3	0.118	0.250	0.474	0.635
4	Valued	~1		Valued.mean	1.057	0.675	1.564	0.118
5	SRPed	~1		SRPed.mean	4.512	0.061	74.224	0.000
6	SRPed	~~	SRPed	SRPed.var	0.330	0.060	5.500	0.000
7	Valued	~~	Valued		0.299	0.055	5.428	0.000
8	CENTERING	~~	CENTERING		0.233	0.000	NA	NA
9	CENTERING	~~	CENTERING:SRPed		1.104	0.000	NA	NA
10	CENTERING:SRPed	~~	CENTERING:SRPed		5.286	0.000	NA	NA
11	CENTERING	~1			0.370	0.000	NA	NA
12	CENTERING:SRPed	~1			1.753	0.000	NA	NA
13	Pre	:=	b2+b3*(0)	Pre	0.704	0.152	4.632	0.000
14	Re	:=	b2+b3*(1)	Re	0.822	0.199	4.124	0.000
	ci.lower	ci.upper	std.lv	std.all	std.nox			
1	-2.760	1.778	-0.570	-0.405	-0.838			
2	0.390	0.974	0.704	0.594	0.499			
3	-0.415	0.589	0.118	0.400	0.174			
4	-0.153	2.465	1.057	1.553	1.553			
5	4.373	4.620	4.512	7.856	7.856			
6	0.229	0.470	0.330	1.000	1.000			
7	0.205	0.423	0.299	0.645	0.645			
8	0.233	0.233	0.233	1.000	0.233			
9	1.104	1.104	1.104	0.994	1.104			
10	5.286	5.286	5.286	1.000	5.286			
11	0.370	0.370	0.370	0.767	0.370			
12	1.753	1.753	1.753	0.762	1.753			
13	0.390	0.974	0.704	0.594	0.499			
14	0.336	1.156	0.822	0.994	0.674			

```
# adding rsquare=TRUE or rsq=T to both summary and parameterEstimates
# resulted in an error related to missing values in row names; could
# not find a solution
```

For reasons unknown to me, I haven't been able to use the commands to produce r-square values without receiving errors. Fortunately, there is a workaround and we can call for the r-square results directly.

```
lavaan::lavInspect(ReCMLefit, "rsquare")
```

```
Valued
0.355
```

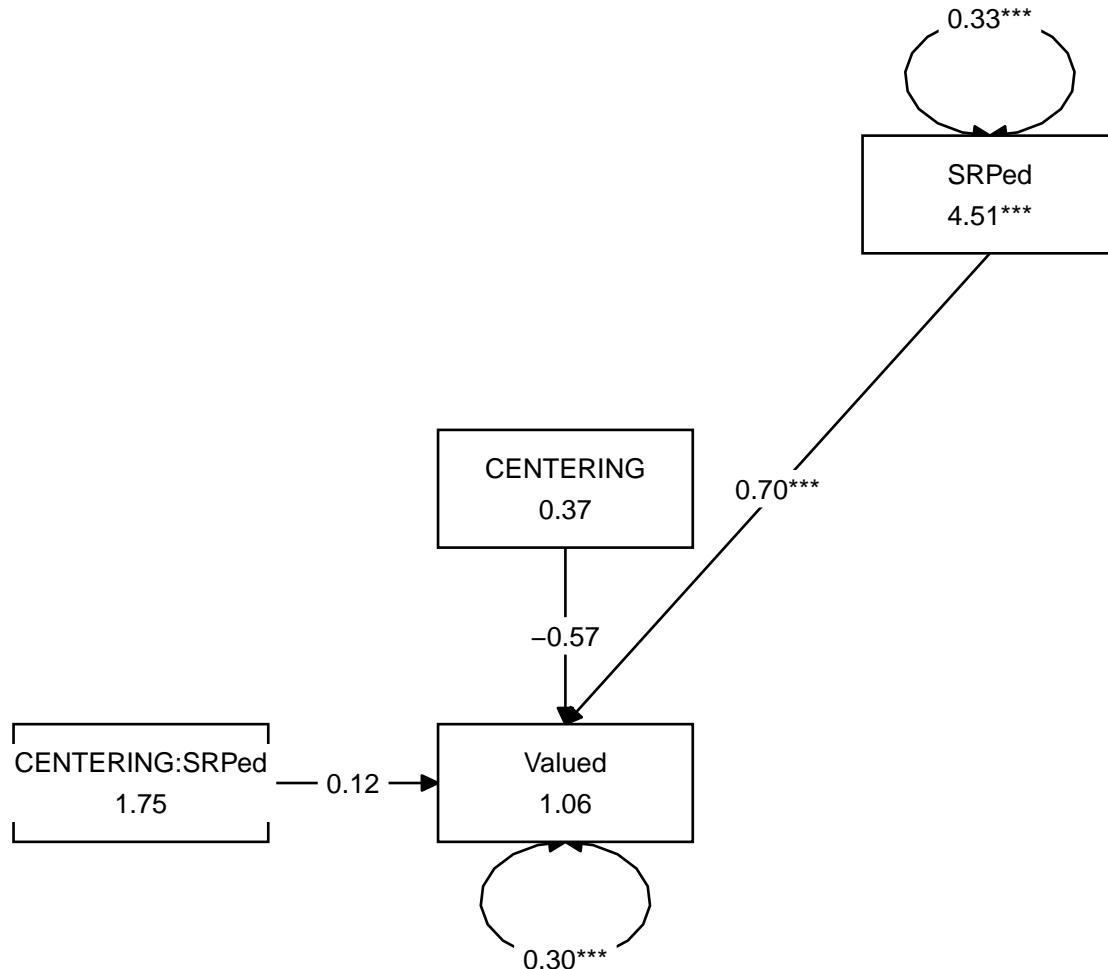
Our model accounts for 36% of the variance in value to the student.

To create a table outside of R, I can export these results as a .csv file (which can be opened in Excel).

```
write.csv(ReCMLEParamEsts, file = "ReCMLEParamEsts.csv")
```

Create a model figure

```
#only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = ReCMLEfit)
```



```
tidySEM::get_layout(ReCMLEfit)
```

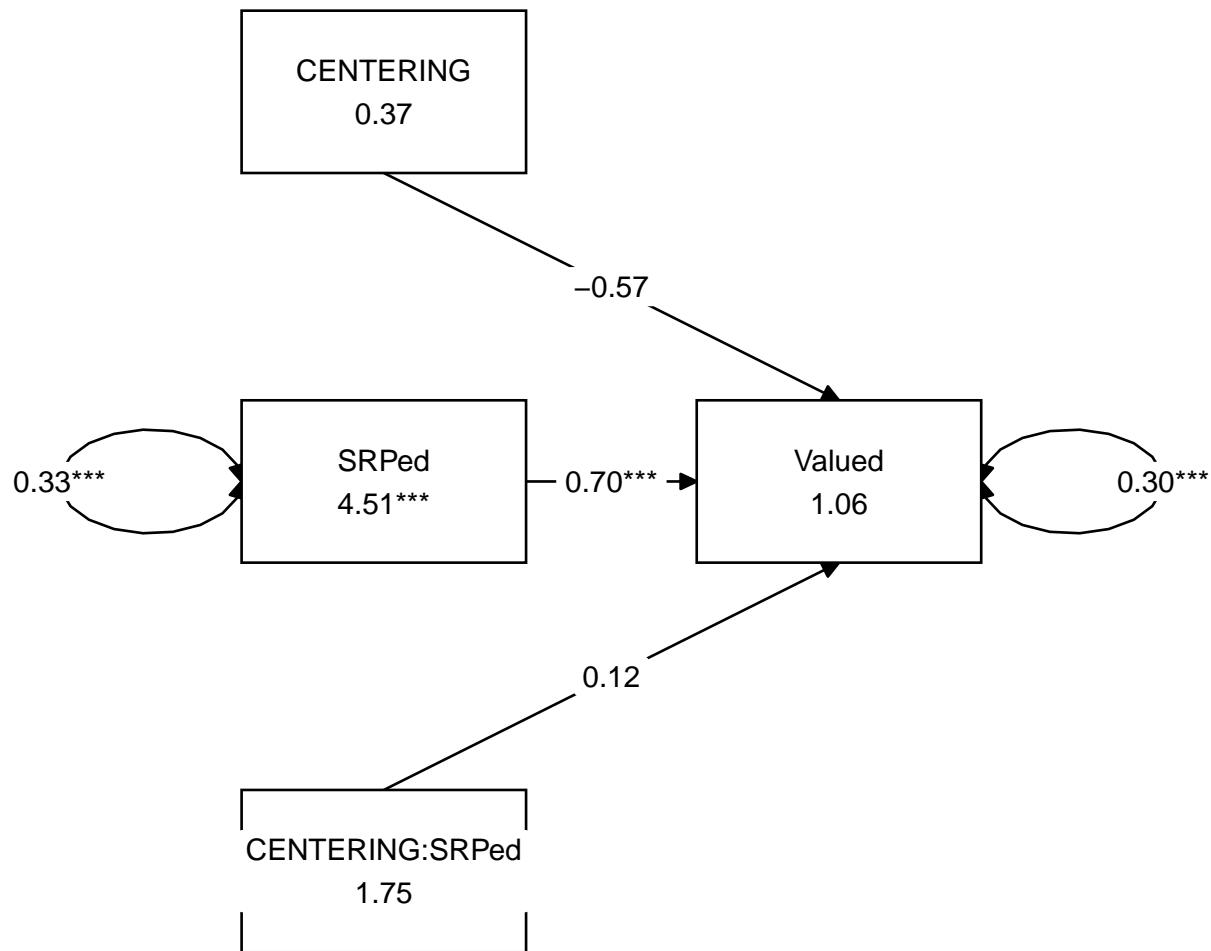
```
[,1] [,2] [,3]
[1,] NA "CENTERING" NA
[2,] "CENTERING:SRPed" "Valued" "SRPed"
attr(,"class")
[1] "layout_matrix" "matrix"      "array"
```

```
ReCmod_map <- tidySEM::get_layout("CENTERING", "", "SRPed", "Valued", "CENTERING:SRPed",
                                 "", rows = 3)
ReCmod_map
```

```
[,1] [,2]
[1,] "CENTERING" ""
[2,] "SRPed"      "Valued"
[3,] "CENTERING:SRPed" ""
attr(,"class")
[1] "layout_matrix" "matrix"      "array"
```

We can update the *tidySEM::graph_sem* function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(ReCMLEfit, layout = ReCmod_map, rect_width = 1.25, rect_height = 1.25,
                   spacing_x = 2, spacing_y = 3, text_size = 4.25)
```



Create a table

For a regular write-up, I would have only done the OLS or the MLE and had half of the table below. However, tabling it together will help me contrast the results.

Table 4

A Comparison of OLS and MLE Regression Results for the ReCentering Analysis

	OLS with the <i>lm()</i> in base R			MLE with <i>lavaan</i>		
	B	SE	p	B	SE	p
Valued (Intercept)	1.057	0.563	0.064	1.057	0.675	0.118
Centering (X)	-0.570	1.318	0.667	-0.570	1.138	0.616

SRPed (W)	0.704	0.127	<0.001	0.704	0.152	<0.001
Centering:SRPed(XY)	0.118	0.281	0.675	0.118	0.250	0.635
<hr/>				<hr/>		
R^2				R^2		
0.367				0.355		

7.12.1 Note similarities and differences in the OLS results

Regression weights are identical; p values of the lavaan/MLE results are more conservative and R^2 of lavaan results is a tad lower.

Represent your work in an APA-style write-up

A multiple regression analysis was conducted to predict course value to the student from the centering (pre-, re-) stage, moderated by evaluation of socially responsive pedagogy. Although the model accounted for 36% of the variance, there was not a statistically significant interaction. Rather, socially responsive pedagogy had a strong main effect ($B = 0.704, SE = 0.163, p < 0.001$) that was true for both pre- and re-centered levels. Results are listed in Table 1 and illustrated in Figure 1.

CONDITIONAL PROCESS ANALYSIS

Chapter 8

Moderated Mediation

[Screencasted Lecture Link](#)

The focus of this lecture is the moderated mediation. That is, are the effects of the indirect effect (sign, significance, strength, presence/absence) *conditional* on the effects of the moderator.

At the outset, please note that although I rely heavily on Hayes [2022a] text and materials, I am using the R package *lavaan* in these chapters. Recently, Hayes has introduced a [PROCESS macro for R](#). Because I am not yet up-to-speed on using this macro (it is not a typical R package) and because we will use *lavaan* for confirmatory factor analysis and structural equation modeling, I have chosen to utilize the *lavaan* package. A substantial difference is that the PROCESS macros use ordinary least squares and *lavaan* uses maximum likelihood estimators.

8.1 Navigating this Lesson

There is about 1 hour and 15 minutes of lecture. If you work through the materials with me it 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

Learning objectives from this lecture include the following:

- Outline a process of evaluating a moderated mediation in a piecewise [Hayes, 2022a] approach to model building
- Recognize conditional process modeling from R script.
- Using the R package *lavaan*,
 - specify a model with indirect effects,

- identify and interpret B weights, p values, and *CIs* for total, direct, and indirect effects,
- calculate the total effects of X and M on Y,
- identify the proportion of variance accounted for in predicting M and Y.
- Regarding conditional indirect effects
 - Interpret an index of moderated mediation
 - Know the essential components of calculating an index of moderated mediation
 - Probe a conditional indirect effect
- Interpret “the usual” things we find in regression: B/beta weights, R, R^2 , and figures

8.1.2 Planning for Practice

The suggestions for homework are graded in complexity and, if you like, can extend from the prior chapter on simple moderation. If you choose the first or second options, you can further amend the simulated data by making further variations such as sample size.

- Rework the problem in the chapter by changing the random seed in the code that simulates the data. This should provide minor changes to the data, but the results will likely be very similar.
- There are a number of variables in the dataset. Swap out one or more variables in the moderated mediation and compare your solution to the one in the chapter (and/or one you mimicked in the journal article).
- Conduct a moderated mediation with data to which you have access. This could include data you simulate on your own or from a published article.

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.

- Hayes, A. F. (2022). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. New York, NY: Guilford Press.
 - **Chapter 11, Fundamentals of conditional Process Analysis:** In this chapter Hayes disentangles conditional indirect effects.
 - **Chapter 12, Further examples of conditional process analysis:** Among the examples is one that includes covariates.
 - **Appendix A: Using PROCESS:** An essential tool for PROCESS users because, even when we are in the R environment, this is the “idea book.” That is, the place where all the path models are presented in figures.
- Lewis, J. A., Williams, M. G., Peppers, E. J., & Gadson, C. A. (2017). Applying intersectionality to explore the relations between gendered racism and health among Black women. *Journal of Counseling Psychology*, 64(5), 475–486. <https://doi-org.ezproxy.spu.edu/10.1037/cou0000231>

8.1.4 Packages

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")
}

## Loading required package: lavaan

## This is lavaan 0.6-16
## lavaan is FREE software! Please report any bugs.

if (!require(semPlot)) {
  install.packages("semPlot")
}

## Loading required package: semPlot

if (!require(tidyverse)) {
  install.packages("tidyverse")
}

## Loading required package: tidyverse

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.2    v readr     2.1.4
## vforcats   1.0.0    v stringr   1.5.0
## v ggplot2   3.4.3    v tibble    3.2.1
## v lubridate 1.9.2    v tidyverse 1.3.0
## v purrr     1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco

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

## Loading required package: psych
##
## Attaching package: 'psych'
```

```

##  

## The following objects are masked from 'package:ggplot2':  

##  

##     %+%, alpha  

##  

## The following object is masked from 'package:lavaan':  

##  

##     cor2cov  

if (!require(jtools)) {  

  install.packages("jtools")  

}  

## Loading required package: jtools

```

8.2 Conditional Process Analysis

8.2.1 The definitional and conceptual

Hayes [Hayes, 2022a] coined the term and suggests we also talk about “conditional process modeling.”

Conditional process analysis: used when the analytical goal is to describe and understand the conditional nature of the mechanism or mechanisms by which a variable transmits its effect on another.

We are integrating moderation and mediation mechanisms together into a single integrated analytical model.

- **Mediator:** Any causal system in which at least one causal antecedent X variable is proposed as influencing an outcome Y through a intervening variable M. In this model, there are two pathways by which X can influence Y: *direct* effect of X on Y, and *indirect* effect of X on Y through M.
 - Answers question, “How does X affect Y”
 - Partitions the X-to-Y relationship into two paths of influence: direct, indirect.
 - Indirect effect contains two components (a,b) that when multiplied ($a \cdot b$) yield an estimate of how much these two cases that differ by one unit on X are estimated to differ on Y through the effect of X on M, which in turn affects Y.
 - Keywords: how, through, via, indirect effect
- **Moderator:** The effect of X on some variable Y is moderated by W if its size, sign, or strength depends on or can be predicted by W.
 - Stated another way, W and X *interact* in their influence on Y.
 - Moderators help establish the boundary conditions of an effect or the circumstances, stimuli, or type of people for which the effect is large v. small, present v. absent, positive v. negative, and so forth.

- Keywords: “it depends,” interaction effect.

Why should we engage both mediators and moderators? Hayes [2022a] suggest that if we have only a mediator(s) in the model that we lose information if we “reduce complex responses that no doubt differ from person to person or situation to situation” (p. 394). He adds that “all effects are moderated by something” (p. 394). Correspondingly, he recommends we add them to a mediation analysis.

Hayes [2022a] suggests that “more complete” (p. 395) analyses model the mechanisms at work linking X to Y (mediator[s]) while simultaneously allowing those effects to be contingent on context, circumstance, or individual difference (moderator[s]).

What are conditional direct and indirect effects? Mediation analyses produce indirect (the product of a sequence of effects that are assumed to be causal) and direct (the unique contribution of X to Y, controlling for other variables in the model) effects. These effects (the X-to-Y/direct and X-to-M-to-Y/indirect), can also be moderated. This is our quest! Figure 11.2 in Hayes’ text [2022a] illustrates conceptually and statistically that we can specify moderation of any combination of direct and indirect paths/effects.

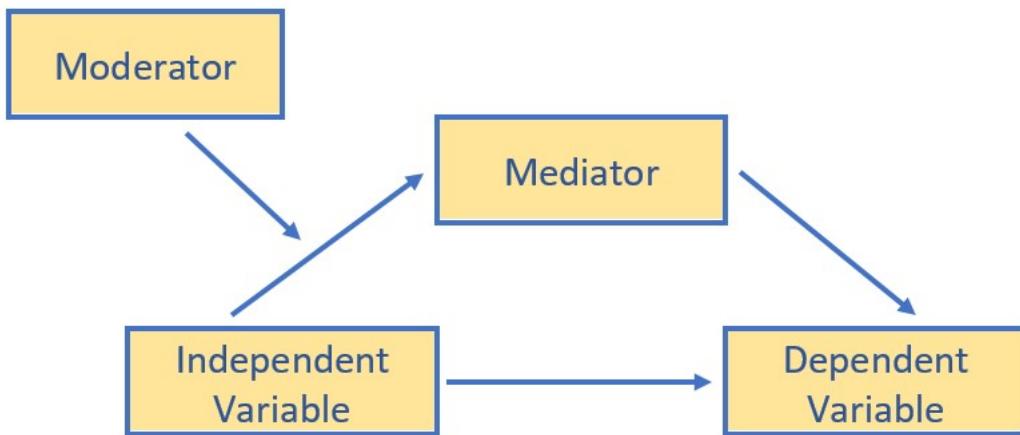


Figure 8.1: Image of conditional process analysis model where the moderator is hypothesized to change the path between the IV and mediator

Within the CPA framework we have lots of options that generally fall into two categories:

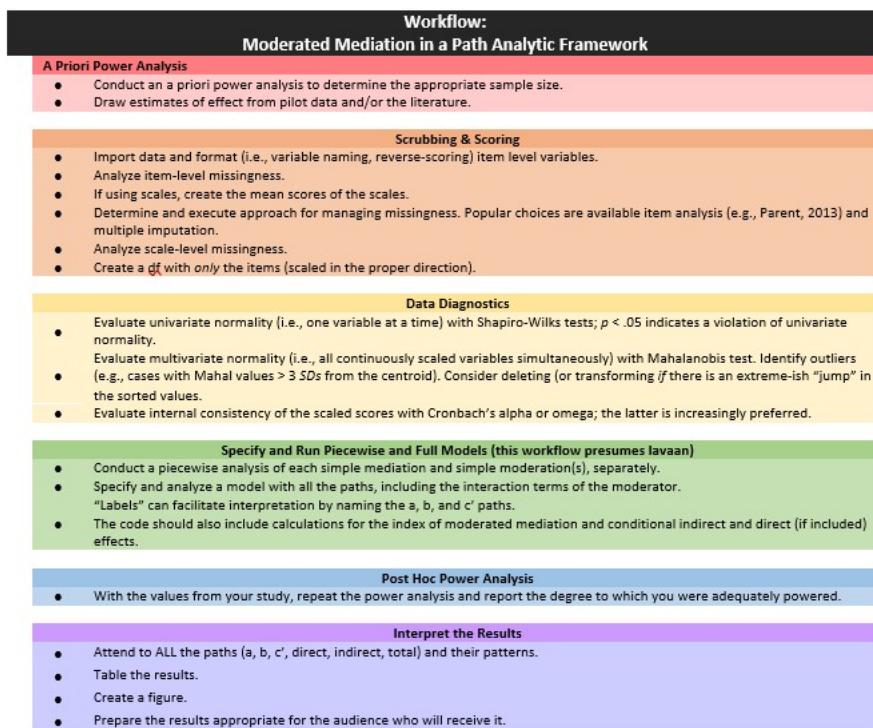
- *Moderated mediation:* when an indirect effect of X on Y through M is moderated; the mechanism represented by the *X-to-M-to-Y* chain of events operates to varying degrees (or not at all) for certain people or in certain contexts.
 - Any model in which the indirect effect ($a \cdot b$) changes as a function of one or more moderators. These moderators can be operating on the a , b , or c' paths or any possible combination of the three
 - X could moderate its own indirect effect on Y through M if the effect of M on Y depends on X, or
 - The indirect effect of X on Y through M could be contingent on a fourth variable if that fourth variable W moderates one or more of the relationships in a three-variable causal system, or

- An indirect effect could be contingent on a moderator variable
- *Mediated moderation:* an interaction between X and some moderator W on Y is carried through a mediator M;
 - mediated moderation analysis is simply a mediation analysis with the product of two variables serving as the causal agent of focus
 - An interaction between a moderator W and causal agent X on outcome Y could operate through a mediator M

Hayes argues that the mediated moderation hypotheses are “regularly articulated and tested by scientists” [2022a, p. 459]. He warns, though, that we should not confuse the “abundance of published examples of mediated moderation analyses...with the meaningfulness of the procedure itself” (p. 460). He later adds that mediation moderation is “neither interesting nor meaningful.” Why?

- Conceptualizing a process in terms of a mediated moderation misdirects attention toward a variable in the model that actually doesn’t measure anything.
- Most often there are moderated mediation models that are identical in equations and resulting coefficients - the difference is in the resulting attentional focus and interpretation.
- Hayes [2022a] recommends that models proposing mediated moderation be recast in terms of moderated mediation process.
- Consequently, we will not work a mediated moderation, but there is an example in his text.

8.3 Workflow for Moderated Mediation



Conducting a moderated mediation involves the following steps:

1. Conducting an a priori power analysis to determine the appropriate sample size.
 - This will require estimates of effect that are drawn from pilot data, the literature, or both.
2. **Scrubbing** and **scoring** the data.
 - Guidelines for such are presented in the respective lessons.
3. Conducting data diagnostics, this includes:
 - item and scale level missingness,
 - internal consistency coefficients (e.g., alphas or omegas) for scale scores,
 - univariate and multivariate normality
4. Beginning with a piecewise analysis of the simpler mediation and moderation(s) models in the larger model.
 - Make note of findings in each of the smaller models.
 - Changes in significance of results from smaller to larger models may indicate power problems and/or combinatorial effects of the variables.
5. Specifying and run the model overall model (this lesson presumes it will with the R package, *lavaan*).
 - The dependent variable should be predicted by the independent, mediating, and covarying (if any) variables and any of their proposed interactions.
 - “Labels” can facilitate interpretation by naming the a, b, and c’ paths.
 - Script should include calculations for the index of moderated mediation and conditional indirect and direct (if included in the model) effects.
6. Conducting a post hoc power analysis.
 - Informed by your own results, you can see if you were adequately powered to detect a statistically significant effect, if, in fact, one exists.
7. Interpret and report the results.
 - Interpret ALL the paths and their patterns.
 - Report if some indirect effects are stronger than others (i.e., results of the contrasts).
 - Create a table and figure.
 - Prepare the results in a manner that is useful to your audience.

In this workflow I call your attention to Hayes' [2022a] piecewise approach to building models. I embrace this approach for a number of reasons. One reason is that examining the smaller portions of the model allows us to really begin to understand the patterns in the data in a systematic way.

Another reason I appreciate the piecewise approach are our historically rigid traditions around hypothesis testing. In summarizing a strategic approach for testing structural equation models, Joreskog [Joreskog, 1993] identified three scenarios:

- *strictly confirmatory*: the traditional NHST approach of proposing a single, theoretically derived, model, and after analyzing the data either rejects or fails to reject the model. No further modifications are made/allowed.

- *alternative models*: the researcher proposes competing (also theoretically derived) models. Following analysis of a single set of empirical data, he or she selects one model as appropriate in representing the sample data.
- *model generating*: A priori, the researcher acknowledges that they may/may not find what they have theoretically proposed. So, a priori, they acknowledge that in the absence of ideal fit (which is the usual circumstance), they will proceed in an exploratory fashion to respecify/re-estimate the model. The goal is to find a model that is both substantively meaningful and statistically well-fitting.

A legacy of our field is the *strictly confirmatory* approach. I am thrilled when I see research experts (e.g., [Byrne, 2016b]) openly endorse a model building approach.

8.4 Research Vignette

Once again the research vignette comes from the Lewis, Williams, Peppers, and Gadson's [2017] study titled, "Applying Intersectionality to Explore the Relations Between Gendered Racism and Health Among Black Women." The study was published in the Journal of Counseling Psychology. Participants were 231 Black women who completed an online survey.

Variables used in the study included:

- **GRMS**: Gendered Racial Microaggressions Scale [Lewis and Neville, 2015] is a 26-item scale that assesses the frequency of nonverbal, verbal, and behavioral negative racial and gender slights experienced by Black women. Scaling is along six points ranging from 0 (never) to 5 (once a week or more). Higher scores indicate a greater frequency of gendered racial microaggressions. An example item is, "Someone has made a sexually inappropriate comment about my butt, hips, or thighs."
- **MntHlth** and **PhysHlth**: Short Form Health Survey - Version 2 [Ware et al., 1995] is a 12-item scale used to report self-reported mental (six items) and physical health (six items). Higher scores indicate higher mental health (e.g., little or no psychological distress) and physical health (e.g., little or no reported symptoms in physical functioning). An example of an item assessing mental health was, "How much of the time during the last 4 weeks have you felt calm and peaceful?"; an example of a physical health item was, "During the past 4 weeks, how much did pain interfere with your normal work?"
- **Sprtly**, **SocSup**, **Engmgt**, and **DisEngmgt** are four subscales from the Brief Coping with Problems Experienced Inventory [Carver, 1997]. The 28 items on this scale are presented on a 4-point scale ranging from 1 (*I usually do not do this at all*) to 4(*I usually do this a lot*). Higher scores indicate a respondents' tendency to engage in a particular strategy. Instructions were modified to ask how the female participants responded to recent experiences of racism and sexism as Black women. The four subscales included spirituality (religion, acceptance, planning), interconnectedness/social support (vent emotions, emotional support,instrumental social support), problem-oriented/engagement coping (active coping, humor, positive reinterpretation/positive reframing), and disengagement coping (behavioral disengagement, substance abuse, denial, self-blame, self-distraction).

- **GRICntly:** The Multidimensional Inventory of Black Identity Centrality subscale [Sellers et al.] was modified to measure the intersection of racial and gender identity centrality. The scale included 10 items scaled from 1 (*strongly disagree*) to 7 (*strongly agree*). An example item was, “Being a *Black woman* is important to my self-image.” Higher scores indicated higher levels of gendered racial identity centrality.

8.4.1 Simulating the data from the journal article

The *lavaan::simulateData* function was used. If you have taken psychometrics, you may recognize the code as one that creates latent variables from item-level data. In trying to be as authentic as possible, we retrieved factor loadings from psychometrically oriented articles that evaluated the measures [Nadal, 2011, Veit and Ware, 1983]. For all others we specified a factor loading of 0.80. We then approximated the *measurement model* by specifying the correlations between the latent variable. We sourced these from the correlation matrix from the research vignette [Lewis et al., 2017]. The process created data with multiple decimals and values that exceeded the boundaries of the variables. For example, in all scales there were negative values. Therefore, the final element of the simulation was a linear transformation that rescaled the variables back to the range described in the journal article and rounding the values to integer (i.e., with no decimal places).

#Entering the intercorrelations, means, and standard deviations from the journal article

```
Lewis_generating_model <- '
  ##measurement model
  GRMS  =~ .69*Ob1 + .69*Ob2 + .60*Ob3 + .59*Ob4 + .55*Ob5 + .55*Ob6 + .54*Ob7 + .50*Ob8
  MntlHlth  =~ .8*MH1 + .8*MH2 + .8*MH3 + .8*MH4 + .8*MH5 + .8*MH6
  PhysHlth  =~ .8*PhH1 + .8*PhH2 + .8*PhH3 + .8*PhH4 + .8*PhH5 + .8*PhH6
  Spirituality  =~ .8*Spirit1 + .8*Spirit2
  SocSupport  =~ .8*SocS1 + .8*SocS2
  Engagement  =~ .8*Eng1 + .8*Eng2
  Disengagement  =~ .8*dEng1 + .8*dEng2
  GRIC  =~ .8*Cntrlty1 + .8*Cntrlty2 + .8*Cntrlty3 + .8*Cntrlty4 + .8*Cntrlty5 + .8*Cntrlty6

  # Means
  GRMS ~ 1.99*1
  Spirituality ~2.82*1
  SocSupport ~ 2.48*1
  Engagement ~ 2.32*1
  Disengagement ~ 1.75*1
  GRIC ~ 5.71*1
  MntlHlth ~3.56*1 #Lewis et al used sums instead of means, I recast as means to facilitate
  PhysHlth ~ 3.51*1 #Lewis et al used sums instead of means, I recast as means to facilitate

  # Correlations
  GRMS ~ 0.20*Spirituality
  GRMS ~ 0.28*SocSupport
  GRMS ~ 0.30*Engagement
```

```

GRMS ~ 0.41*Disengagement
GRMS ~ 0.19*GRIC
GRMS ~ -0.32*MntlHlth
GRMS ~ -0.18*PhysHlth

Spirituality ~ 0.49*SocSupport
Spirituality ~ 0.57*Engagement
Spirituality ~ 0.22*Disengagement
Spirituality ~ 0.12*GRIC
Spirituality ~ -0.06*MntlHlth
Spirituality ~ -0.13*PhysHlth

SocSupport ~ 0.46*Engagement
SocSupport ~ 0.26*Disengagement
SocSupport ~ 0.38*GRIC
SocSupport ~ -0.18*MntlHlth
SocSupport ~ -0.08*PhysHlth

Engagement ~ 0.37*Disengagement
Engagement ~ 0.08*GRIC
Engagement ~ -0.14*MntlHlth
Engagement ~ -0.06*PhysHlth

Disengagement ~ 0.05*GRIC
Disengagement ~ -0.54*MntlHlth
Disengagement ~ -0.28*PhysHlth

GRIC ~ -0.10*MntlHlth
GRIC ~ 0.14*PhysHlth

MntlHlth ~ 0.47*PhysHlth
'

set.seed(230925)
dfLewis <- lavaan::simulateData(model = Lewis_generating_model,
                                 model.type = "sem",
                                 meanstructure = T,
                                 sample.nobs=231,
                                 standardized=FALSE)

#used to retrieve column indices used in the rescaling script below
#col_index <- as.data.frame(colnames(dfLewis))

for(i in 1:ncol(dfLewis)){ # for loop to go through each column of the dataframe
  if(i >= 1 & i <= 25){ # apply only to GRMS variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 5))
  }
}

```

```

if(i >= 26 & i <= 37){  # apply only to mental and physical health variables
  dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 6))
}
if(i >= 38 & i <= 45){  # apply only to coping variables
  dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 4))
}
if(i >= 46 & i <= 55){  # apply only to GRIC variables
  dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 7))
}
}

#rounding to integers so that the data resembles that which was collected
library(tidyverse)
dfLewis <- dfLewis %>% round(0)

#quick check of my work
#psych::describe(dfLewis)

```

The script below allows you to store the simulated data as a file on your computer. This is optional – the entire lesson can be worked with the simulated data.

If you prefer the .rds format, use this script (remove the hashtags). The .rds format has the advantage of preserving any formatting of variables. A disadvantage is that you cannot open these files outside of the R environment.

Script to save the data to your computer as an .rds file.

```
# saveRDS(dfLewis, 'dfLewis.rds')
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfLewis<- readRDS('dfLewis.rds')
```

If you prefer the .csv format (think “Excel lite”) use this script (remove the hashtags). An advantage of the .csv format is that you can open the data outside of the R environment. A disadvantage is that it may not retain any formatting of variables

Script to save the data to your computer as a .csv file.

```
# write.table(dfLewis, file = 'dfLewis.csv', sep = ',',
# col.names=TRUE, row.names=FALSE)
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfLewis<- read.csv ('dfLewis.csv', header = TRUE)
```

8.4.2 Scrubbing, Scoring, and Data Diagnostics

Because the focus of this lesson is on moderation, we have used simulated data (which serves to avoid problems like missingness and non-normal distributions). If this were real, raw, data, it would be important to [scrub](#), [score](#), and conduct [data diagnostics](#) to evaluate the suitability of the data for the proposed analyses.

Because we are working with item level data we do need to score the scales used in the researcher's model. Because we are using simulated data and the authors already reverse coded any such items, we will omit that step.

As described in the [Scoring](#) chapter, we calculate mean scores of these variables by first creating concatenated lists of variable names. Next we apply the `sjstats::mean_n` function to obtain mean scores when a given percentage (we'll specify 80%) of variables are non-missing. Functionally, this would require the two-item variables (e.g., engagement coping and disengagement coping) to have non-missingness. We simulated a set of data that does not have missingness, none-the-less, this specification is useful in real-world settings.

Note that I am only scoring the variables used in the models demonstrated in this lesson. The variables that are simulated but not scored could be used for the practice suggestions.

```
GRMS_vars <- c("Ob1", "Ob2", "Ob3", "Ob4", "Ob5", "Ob6", "Ob7", "Ob8",
               "Ob9", "Ob10", "Ma1", "Ma2", "Ma3", "Ma4", "Ma5", "Ma6", "Ma7", "St1",
               "St2", "St3", "St4", "St5", "An1", "An2", "An3")
Eng_vars <- c("Eng1", "Eng2")
dEng_vars <- c("dEng1", "dEng2")
MntlHlth_vars <- c("MH1", "MH2", "MH3", "MH4", "MH5", "MH6")
Cntrlty_vars <- c("Cntrlty1", "Cntrlty2", "Cntrlty3", "Cntrlty4", "Cntrlty5",
                  "Cntrlty6", "Cntrlty7", "Cntrlty8", "Cntrlty9", "Cntrlty10")

dfLewis$GRMS <- sjstats::mean_n(dfLewis[, GRMS_vars], 0.8)

## Registered S3 methods overwritten by 'broom':
##   method         from
##   tidy.glht     jtools
##   tidy.summary.glht jtools

dfLewis$Engmt <- sjstats::mean_n(dfLewis[, Eng_vars], 0.8)
dfLewis$DisEngmt <- sjstats::mean_n(dfLewis[, dEng_vars], 0.8)
dfLewis$MntlHlth <- sjstats::mean_n(dfLewis[, MntlHlth_vars], 0.8)
dfLewis$Centrality <- sjstats::mean_n(dfLewis[, Cntrlty_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting periods in front of the variable
# list. One example is provided. dfLewis$GRMS <-
# sjstats::mean_n(dfLewis[, ..GRMS_vars], 0.80)
```

Now that we have scored our data, let's trim the variables to just those we need.

```
Lewis_df <- dplyr::select(dfLewis, GRMS, Centrality, DisEngmt, MntlHlth)
```

Let's check a table of means, standard deviations, and correlations to see if they align with the published article.

```
Lewis_table <- apaTables::apa.cor.table(Lewis_df, table.number = 1, show.sig.stars = TRUE,
                                         landscape = TRUE, filename = "Lewis_Corr.doc")
print(Lewis_table)

##
##
## Table 1
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##      Variable      M      SD   1          2          3
## 1. GRMS     2.56  0.72
##
## 2. Centrality 3.94  0.76 .24**
##                      [.11, .36]
##
## 3. DisEngmt   2.48  0.52 .53**       .05
##                      [.43, .62]  [-.08, .18]
##
## 4. MntlHlth   3.16  0.81 -.56**      -.09       -.48**
##                      [-.64, -.47] [-.21, .04] [-.57, -.37]
##
## 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.
##
```

8.4.3 Quick peek at the data

```
library(psych)
psych::describe(Lewis_df)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
## GRMS	1	231	2.56	0.72	2.56	2.56	0.77	0.32	4.24	3.92	-0.08
## Centrality	2	231	3.94	0.76	3.90	3.92	0.74	1.90	6.00	4.10	0.25

```

## DisEngmt      3 231 2.48 0.52   2.50    2.47 0.74 1.00 4.00   3.00   0.11
## MntlHlth     4 231 3.16 0.81   3.17    3.16 0.74 1.17 5.50   4.33   0.05
##           kurtosis   se
## GRMS          -0.14 0.05
## Centrality    -0.08 0.05
## DisEngmt      -0.16 0.03
## MntlHlth      -0.20 0.05

```

8.5 Working the Moderated Mediation

The model we are testing is predicting a mental health (MntlHlth, Y) from gendered racial microaggressions (GRMS,X), mediated by disengagement coping (DisEngmt, M). The relationship between gendered racial microaggressions and disengagement coping (i.e., the a path) is expected to be moderated by gendered racial identity centrality (GRIcntlty, W). Gendered racial identity centrality is also expected to moderate the path between gendered racial microaggressions and mental health (i.e., the c' path). Thus, the specified model involves the evaluation of a conditional indirect effect.

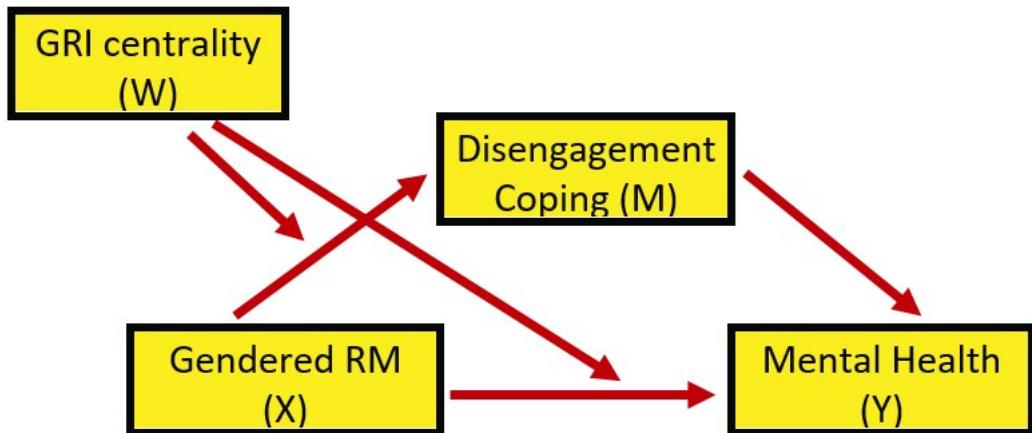


Figure 8.2: Image of conceptual representation of the conditional process analysis model where the moderator is hypothesized to change the a and c' paths

Hayes' [2022a] textbook and training materials frequently display the conceptual (above) and statistical models (below). These help facilitate understanding.

Looking at the diagram, with two consequent variables (i.e., those with arrows pointing to them) we can see two equations are needed to explain the model:

$$M = i_M + a_1 X + a_2 W + a_3 XW + e_M$$

$$Y = i_Y + c'_1 X + c'_2 W + c'_3 XW + bM + e_Y$$

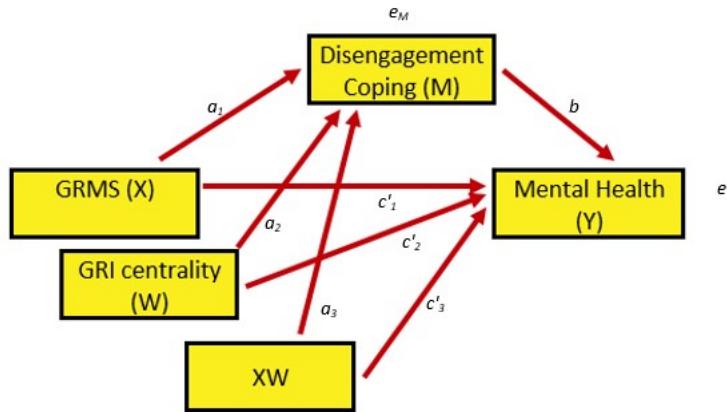


Figure 8.3: Image of statistical representation of the conditional process analysis model where the moderator is hypothesized to change the a and c' paths

When we have complicated models such as these, Hayes [2022a] suggests a piecewise approach to model building. Specifically, he decompose the model into its aggregate parts: a simple mediation and two simple moderation.

Let's start with the the simple moderations.

8.5.1 Piecewise Assembly of the Moderated Mediation

8.5.1.1 Analysis #1: A simple moderation

We are asking, “Does GRI centrality moderate the relationship between gendered racial microaggressions and disengagement coping?

Y = disengagement coping X = gendered racial microaggressions W = GRI centrality

The formula we are estimating:

$$Y = b_0 + b_1 X + b_2 W + b_3 XW + e_Y$$

Let's specify this simple moderation model with base R's `lm()` function. Let's use the `jtools` package so we get that great `summ` function and `interactions` for the awesome plot.

Since we are just working to understand our moderations, we can run them with “regular old” ordinary least squares.

```

# library(jtools) #the summ function creates a terrific regression
# table library(interactions) library(ggplot2)

Mod_a_path <- lm(DisEngmt ~ GRMS * Centrality, data = Lewis_df)
jtools::summ(Mod_a_path, digits = 3)
  
```

Looking at these results we can see that the predictors account for about 29% of variance in disengagement coping. Only the independent variable (X), GRMS is a significant predictor. Neither

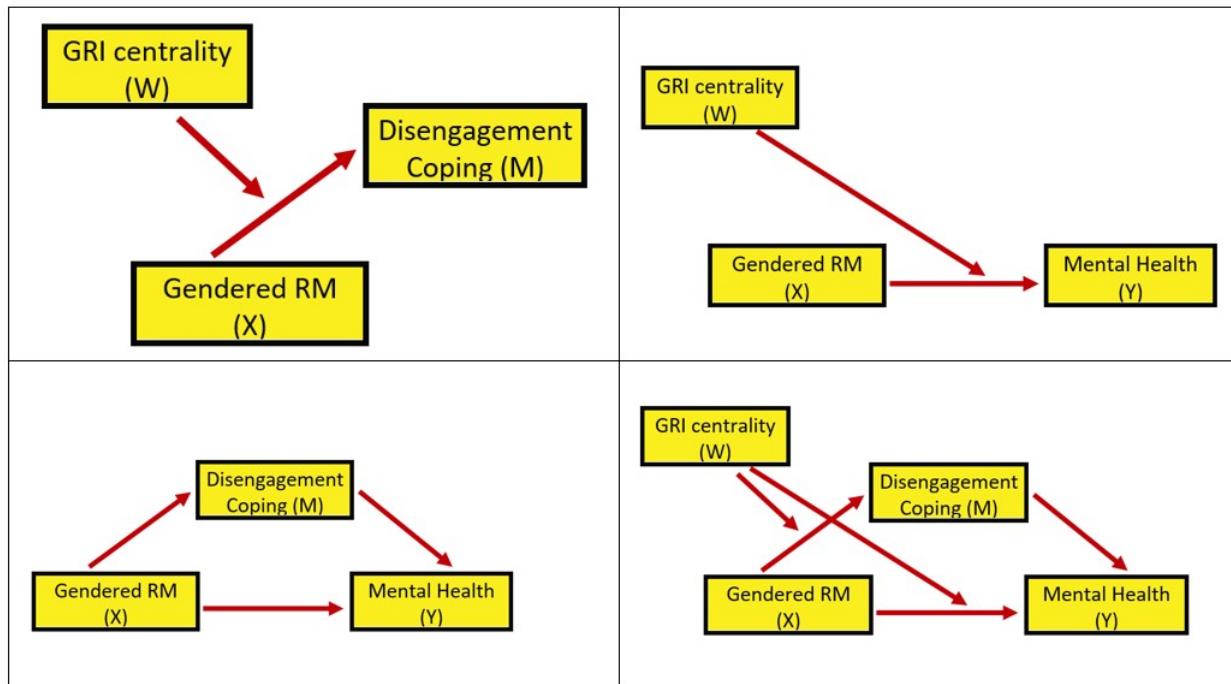


Figure 8.4: Image of statistical representation of the conditional process analysis model where the moderator is hypothesized to change the a and c' paths

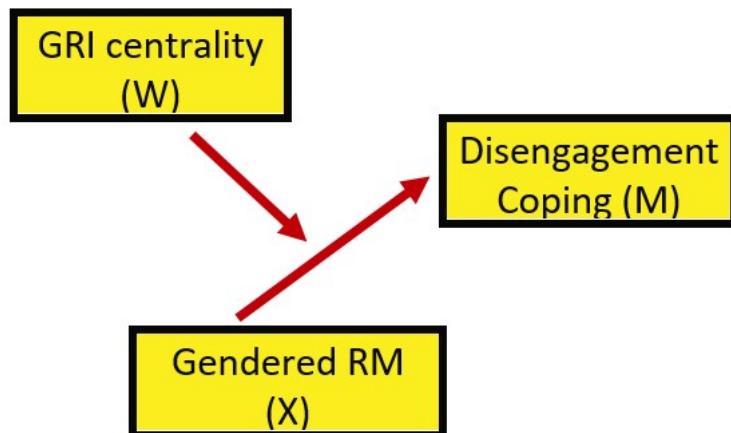


Figure 8.5: Image of statistical representation of the simple moderation estimating DisEngmt from GRMS, moderated by GRIcntly

Observations	231
Dependent variable	DisEngmt
Type	OLS linear regression

F(3,227)	31.245
R ²	0.292
Adj. R ²	0.283

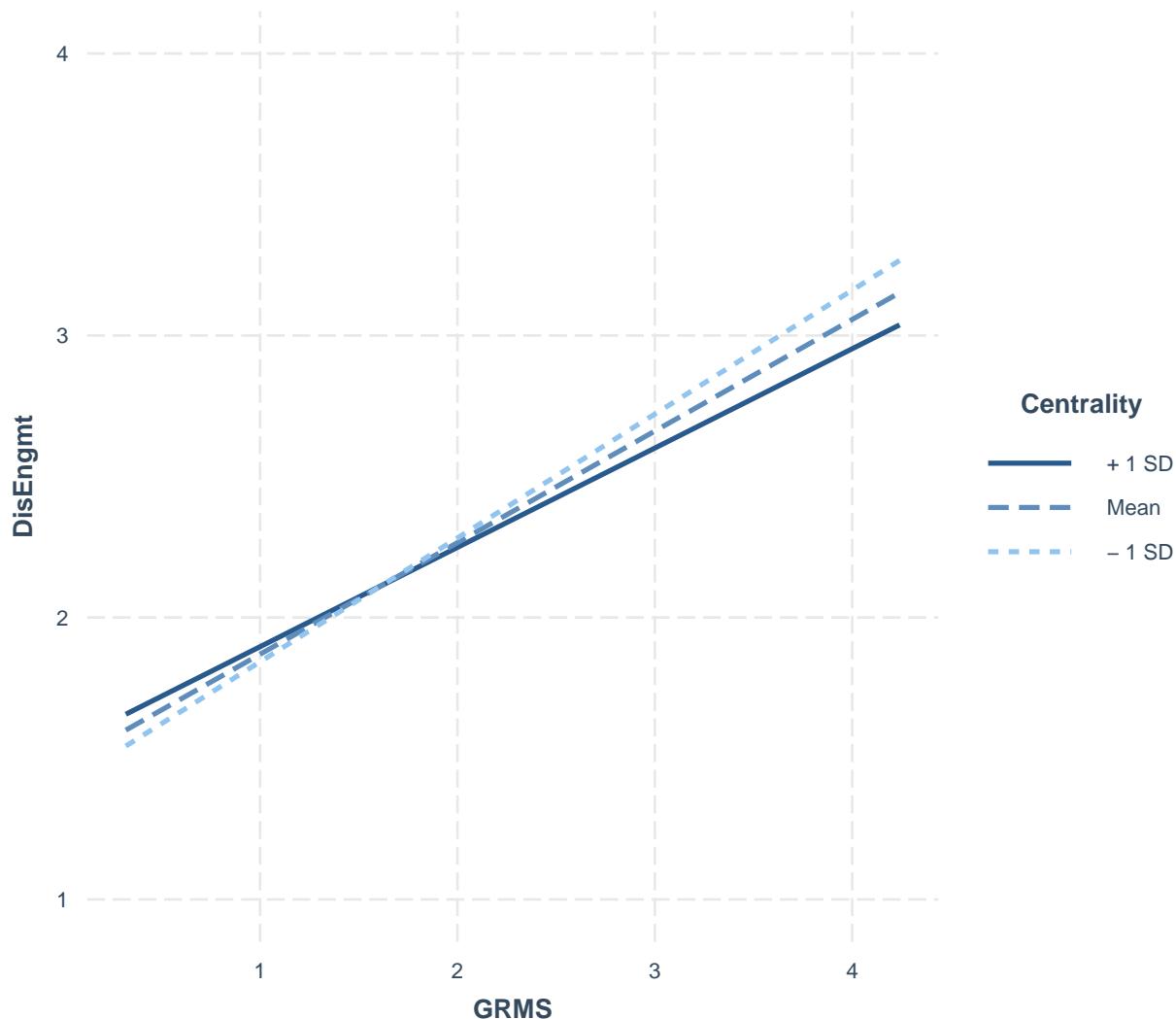
	Est.	S.E.	t val.	p
(Intercept)	1.107	0.510	2.169	0.031
GRMS	0.623	0.193	3.220	0.001
Centrality	0.093	0.132	0.703	0.483
GRMS:Centrality	-0.058	0.049	-1.179	0.240

Standard errors: OLS

the moderator (GRICntly, [Y])), nor its interaction with GRMS (GRMS:GRICntly, [XW]) are significant.

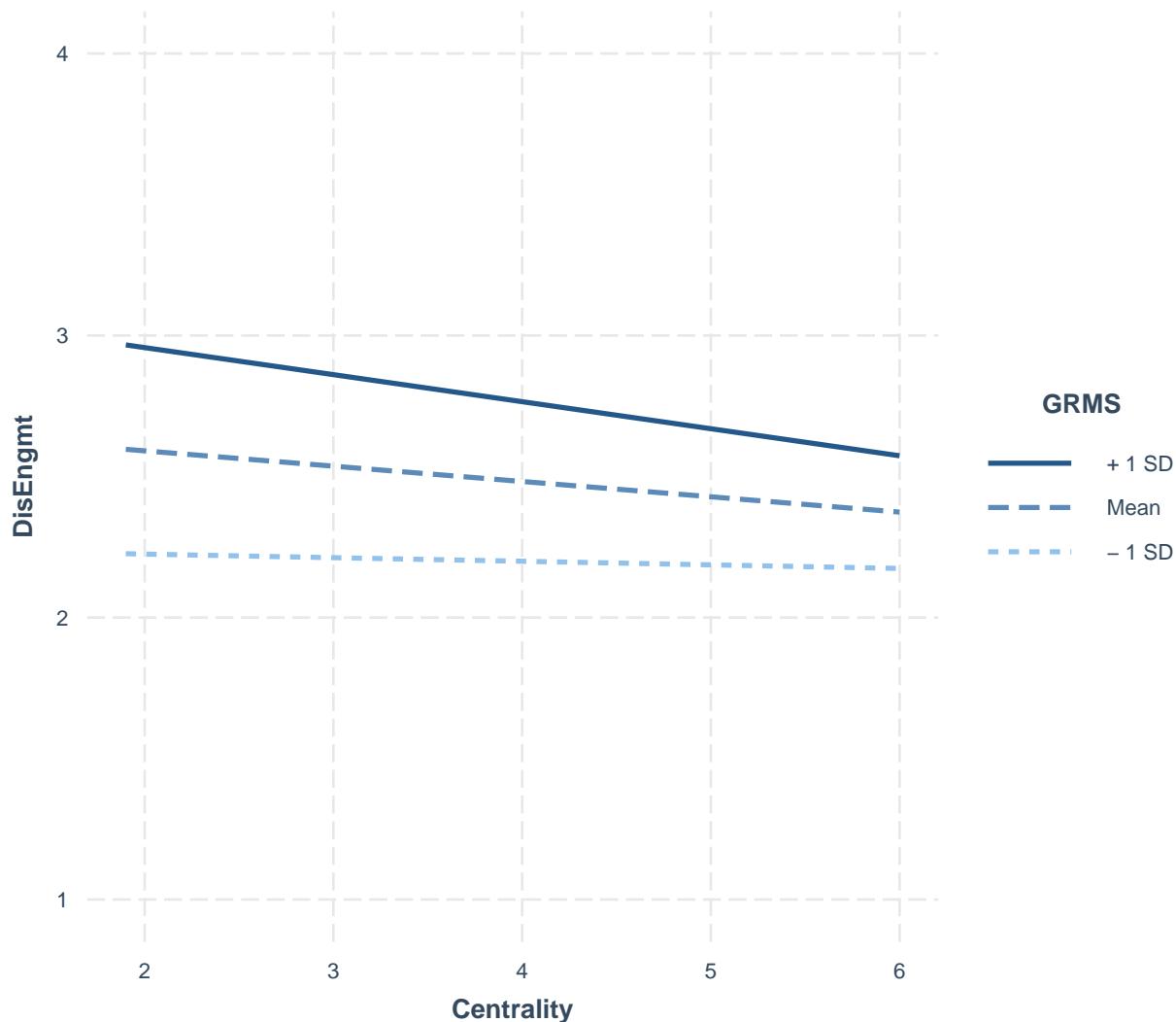
It's always helpful to graph the relationship. The *interaction_plot()* function from the package, *interactions* can make helpful illustrations. In the case of interactions/moderations, I like to run them "both ways" to see which makes more sense.

```
interactions::interaction_plot(Mod_a_path, pred = GRMS, modx = Centrality) +
  ylim(1, 4)
```



The figure with GRIcntrlty as the moderator, shows a very similar prediction of disengagement coping from gendered racial microaggressions across all three levels of gendered racial identity centrality.

```
interactions::interact_plot(Mod_a_path, pred = Centrality, modx = GRMS) +
  ylim(1, 4)
```



The figure that positions GRMS in the moderator role shows the significant main effect of GRMS on disengagement coping. It is clear that who experience the highest levels of gendered racial microaggressions are using a more disengaged coping style.

Next, let's probe the interaction with simple slopes. Probing the interaction is a common follow-up. With these additional inferential tests we can see where in the distribution of the moderator, X has an effect on Y that is different from zero (and where it does not). There are two common approaches.

The Johnson-Neyman is a *floodlight* approach and provides an indication of the places in the distribution of W (moderator) that X has an effect on Y that is different than zero. The *analysis of simple slopes* approach is thought of as a *spotlight* approach because probes the distribution at specific values (often the $M +/- 1SD$).

```
interactions::sim_slopes(Mod_a_path, pred = GRMS, modx = Centrality)
```

```
## JOHNSON-NEYMAN INTERVAL
##
```

```

## When Centrality is INSIDE the interval [-6.39, 6.37], the slope of GRMS is
## p < .05.
##
## Note: The range of observed values of Centrality is [1.90, 6.00]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of GRMS when Centrality = 3.182522 (- 1 SD):
##
##   Est.   S.E.   t val.      p
##   ----- -----
##   0.44   0.05     8.23   0.00
##
## Slope of GRMS when Centrality = 3.938095 (Mean):
##
##   Est.   S.E.   t val.      p
##   ----- -----
##   0.40   0.04     9.42   0.00
##
## Slope of GRMS when Centrality = 4.693668 (+ 1 SD):
##
##   Est.   S.E.   t val.      p
##   ----- -----
##   0.35   0.06     6.03   0.00

# sim_slopes(Mod_a_path, pred=GRIcntly, modx = GRMS) #sometimes I
# like to look at it in reverse -- like in the plots

```

The Johnson-Neyman suggests that between the GRIcntly values of -6.39 and 6.37, the relationship between GRMS is statistically significant. We see the same result in the pick-a-point approach where at the GRIcntly values of 3.18, 3.94, and 4.69, GRMS has a statistically significant effect on disengagement coping. Is this a contradiction to the non-significant interaction effect?

No. The test of interaction is an interaction about the relationship between W and X 's effect on Y . Just showing that X is significantly related to Y for a specific value does not address any dependence upon the moderator (W). Hayes [2022a] covers this well in his Chapter 14, in the section “Reporting a Moderation Analysis.”

What have we learned in this simple moderation?

- Only GRMS (X) has a statistically significant effect on disengagement coping.
- Neither the moderator (Centrality W) nor its interaction with GRMS (WX) are statistically significant. While there are no significant predictors (neither X, W, nor XW)
- The model accounts for about 29% of variance in the DV.

8.5.1.2 Analysis #2: Another simple moderation

We are asking, “Does gendered racial identity centrality moderate the relationship between gendered racial microaggressions and mental health?”

Y = mental health X = gendered racial microaggressions W = GRI centrality

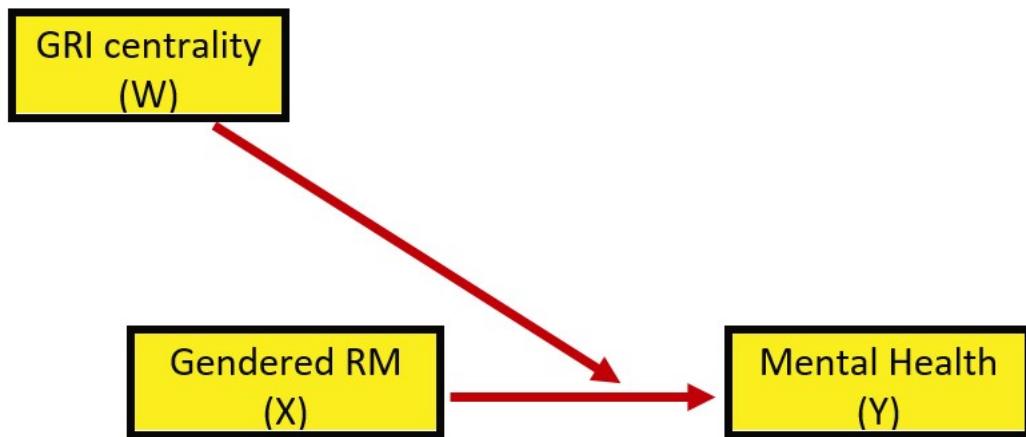


Figure 8.6: Image of statistical representation of the simple moderation estimating $MntlHlth$ from $GRMS$, moderated by $GRIcntly$

As before, this is our formulaic rendering:

$$Y = b_0 + b_1X + b_2W + b_3XW + e_Y$$

```
Mod_c_path <- lm(MntlHlth ~ GRMS * Centrality, data = Lewis_df)
jtools::summ(Mod_c_path, digits = 3)
```

Observations	231
Dependent variable	$MntlHlth$
Type	OLS linear regression

F(3,227)	37.386
R ²	0.331
Adj. R ²	0.322

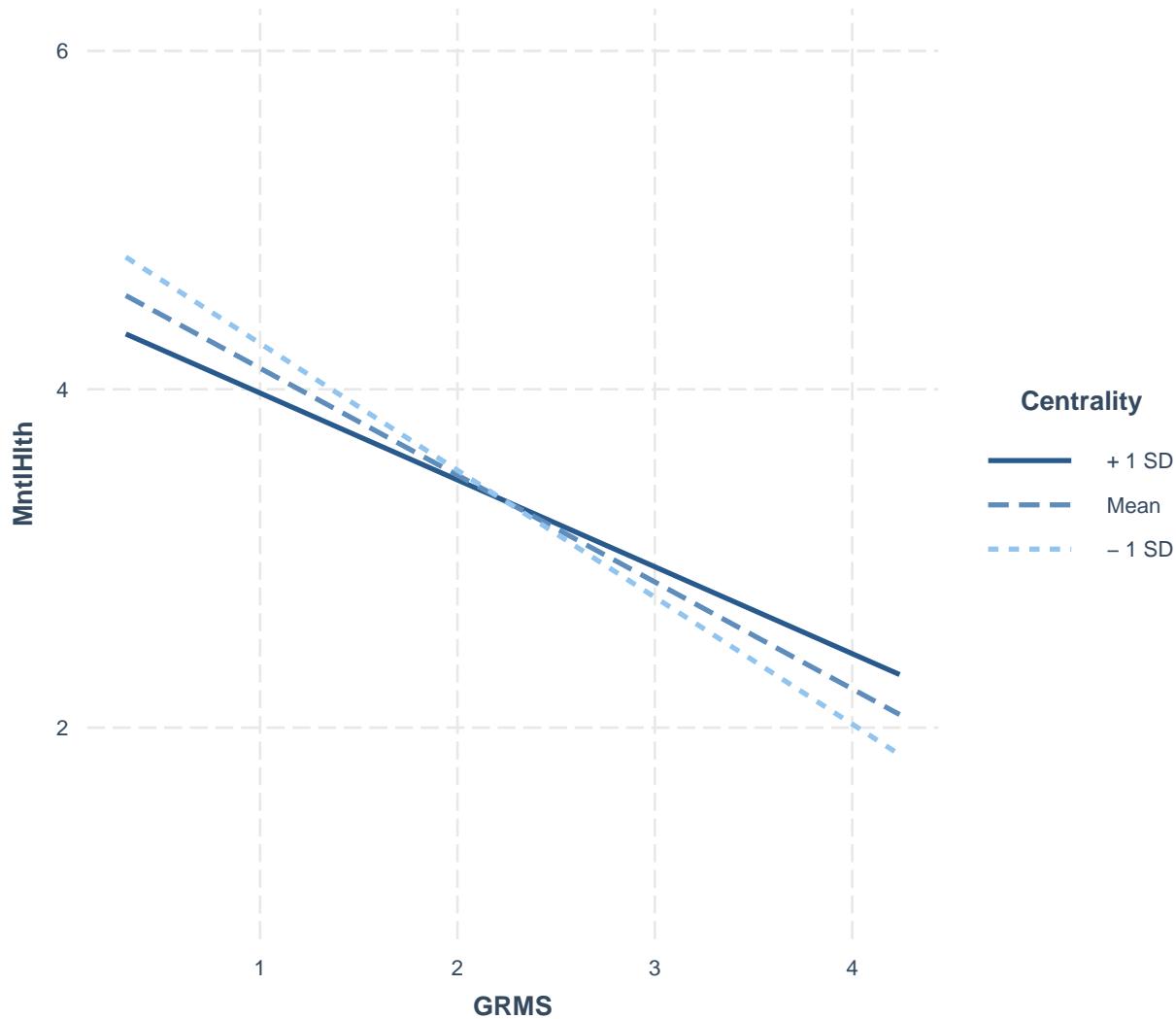
	Est.	S.E.	t val.	p
(Intercept)	6.138	0.767	8.007	0.000
GRMS	-1.248	0.290	-4.299	0.000
Centrality	-0.351	0.199	-1.764	0.079
GRMS:Centrality	0.157	0.073	2.132	0.034

Standard errors: OLS

In this model that is, overall, statistically significant, we account for about 33% of variance in the DV. Looking at these results we can see that there is a statistically significant main effect of GRMS on mental health as well as statistically significant GRMS:Centrality interaction effect.

Let's look at the plots.

```
interactions::interact_plot(Mod_c_path, pred = GRMS, modx = Centrality) +
  ylim(1, 6)
```



```
# interactions::interact_plot(Mod_c_path, pred = GRIcntrly, modx =
# GRMS + ylim(1,6))
```

The figure with GRIcntrly as the moderator, shows fanning out when mental health is high and GRMS is low.

Next, let's probe the interaction with simple slopes. Probing the interaction is a common follow-up. With these additional inferential tests we can see where in the distribution of the moderator, X has an effect on Y that is different from zero (and where it does not). There are two common approaches.

The Johnson-Neyman is a *floodlight* approach and provides an indication of the places in the distribution of W (moderator) that X has an effect on Y that is different than zero. The *analysis of simple slopes* or *spotlight* approach, probes the distribution at specific values (often the $M \pm 1SD$).

```

interactions::sim_slopes(Mod_c_path, pred = GRMS, modx = Centrality)

## JOHNSON-NEYMAN INTERVAL
##
## When Centrality is OUTSIDE the interval [5.92, 58.37], the slope of GRMS is
## p < .05.
##
## Note: The range of observed values of Centrality is [1.90, 6.00]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of GRMS when Centrality = 3.182522 (- 1 SD):
##
##      Est.    S.E.    t val.      p
## ----- -----
##    -0.75    0.08     -9.36    0.00
##
## Slope of GRMS when Centrality = 3.938095 (Mean):
##
##      Est.    S.E.    t val.      p
## ----- -----
##    -0.63    0.06    -10.01    0.00
##
## Slope of GRMS when Centrality = 4.693668 (+ 1 SD):
##
##      Est.    S.E.    t val.      p
## ----- -----
##    -0.51    0.09     -5.85    0.00

# sim_slopes(Mod_c_path, pred=GRICntlty, modx = GRMS) #sometimes I
# like to look at it in reverse -- like in the plots

```

The Johnson-Neyman suggests that between the Centrality values of 5.92 and 58.37], the relationship between GRMS is and mental health statistically significant. We see the same result in the pick-a-point approach where at the Centrality values of 3.19, 3.94, and 4.69, GRMS has a statistically significant effect on mental health. Is this a contradiction to the non-significant interaction effect?

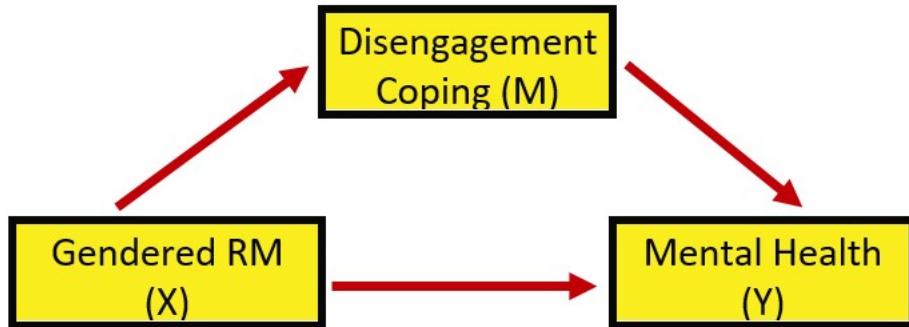
What have we learned in this simple moderation?

- There was a statistically significant main effect of GRMS on mental health as well as statistically significant GRMS:Centrality interaction effect.
- It is curious that in the presence of the statistically significant interaction effect, we did not see differences in significance in the analysis of simple slopes.
- The overall model was significant and accounted for 33% of variance in the DV.

8.5.1.3 Analysis #3: A simple mediation

We are asking, “Does disengagement coping mediate the relationship between gendered racial microaggressions and mental health?”

Y = mental health X = gendered racial microaggressions M = GRI centrality



Looking at the diagram, with two consequent variables (i.e., those with arrows pointing to them) we can see two equations are needed to explain the model:

$$M = i_M + aX + e_M$$

$$Y = i_Y + c'X + bM + e_Y$$

To conduct this analysis, I am using the guidelines in the chapter on simple mediation. We are switching to the *lavaan* package.

```

## lavaan 0.6.16 ended normally after 1 iteration
##
##   Estimator                      ML
## Optimization method            NLMINB
## Number of model parameters      7
##
##   Number of observations        231
## Number of missing patterns       1
##
## Model Test User Model:
##
##   Test statistic                 0.000
## Degrees of freedom                  0
##
## Parameter Estimates:
##
##   Standard errors                Bootstrap
## Number of requested bootstrap draws    1000
## Number of successful bootstrap draws   1000
##
## Regressions:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## MntlHlth ~
##   DisEngmt (b)   -0.381   0.089  -4.262   0.000  -0.547  -0.206
##   GRMS   (c_p)  -0.483   0.066  -7.271   0.000  -0.607  -0.354
## DisEngmt ~
##   GRMS     (a)   0.386   0.038  10.096   0.000   0.301   0.456
## Std.lv  Std.all
##
##   -0.381  -0.247
##   -0.483  -0.430
##
##   0.386   0.531
##
## Intercepts:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .DsEngmt (DsE.)  1.490   0.100 14.935   0.000   1.307   1.709
## .MntlHlt (MnH.)  5.342   0.195 27.390   0.000   4.953   5.721
## Std.lv  Std.all
##   1.490   2.845
##   5.342   6.604
##
## Variances:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .MntlHlth      0.420   0.040 10.544   0.000   0.343   0.496
## .DisEngmt       0.197   0.016 11.967   0.000   0.163   0.229
## Std.lv  Std.all
##   0.420   0.641

```

```

##      0.197    0.718
##
## R-Square:
##           Estimate
## MntlHlth      0.359
## DisEngmt      0.282
##
## Defined Parameters:
##           Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## indirect     -0.147   0.039  -3.824   0.000  -0.226  -0.075
## direct       -0.483   0.067  -7.268   0.000  -0.607  -0.354
## total_c      -0.631   0.058 -10.789   0.000  -0.742  -0.513
## Std.lv  Std.all
## -0.147  -0.131
## -0.483  -0.430
## -0.631  -0.561

```

LMed_ParEsts

	lhs	op	rhs	label	est	se	z	pvalue	ci.lower
## 1	MntlHlth	~	DisEngmt	b	-0.381	0.089	-4.262	0	-0.547
## 2	MntlHlth	~	GRMS	c_p	-0.483	0.066	-7.271	0	-0.606
## 3	DisEngmt	~	GRMS	a	0.386	0.038	10.096	0	0.311
## 4	DisEngmt	~1		DisEngmt.mean	1.490	0.100	14.935	0	1.297
## 5	MntlHlth	~1		MntlHlth.mean	5.342	0.195	27.390	0	4.900
## 6	MntlHlth	~~	MntlHlth		0.420	0.040	10.544	0	0.352
## 7	DisEngmt	~~	DisEngmt		0.197	0.016	11.967	0	0.166
## 8	GRMS	~~	GRMS		0.518	0.000	NA	NA	0.518
## 9	GRMS	~1			2.557	0.000	NA	NA	2.557
## 10	indirect	:=	a*b	indirect	-0.147	0.039	-3.824	0	-0.234
## 11	direct	:=	c_p	direct	-0.483	0.067	-7.268	0	-0.606
## 12	total_c	:=	c_p+(a*b)	total_c	-0.631	0.058	-10.789	0	-0.739
				ci.upper	std.lv	std.all	std.nox		
## 1					-0.206	-0.381	-0.247	-0.247	
## 2					-0.347	-0.483	-0.430	-0.597	
## 3					0.465	0.386	0.531	0.738	
## 4					1.698	1.490	2.845	2.845	
## 5					5.699	5.342	6.604	6.604	
## 6					0.513	0.420	0.641	0.641	
## 7					0.234	0.197	0.718	0.718	
## 8					0.518	0.518	1.000	0.518	
## 9					2.557	2.557	3.552	2.557	
## 10					-0.079	-0.147	-0.131	-0.182	
## 11					-0.347	-0.483	-0.430	-0.597	
## 12					-0.505	-0.631	-0.561	-0.780	

In this simple mediation we learn*:

- The a path (GRMS → DisEngmt) is statistically significant.
- The b path (DisEngmt → MntlHlth) is statistically significant.
- The total effect (GRMS → MntlHlth) is statistically significant.
- The direct effect (GRMS → MntlHlth when DisEngmt is in the model) is still significant.
- The indirect effect is statistically significant.
- The model accounts for 36% of the variance in mental health (DV) and 28% of the variance in disengagement coping (M).

Recall how the bootstrapped, bias-corrected confidence intervals can be different? It's always good to check. In this case, CI95s and the p values are congruent.

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)

## Loading required package: OpenMx

##
## Attaching package: 'OpenMx'

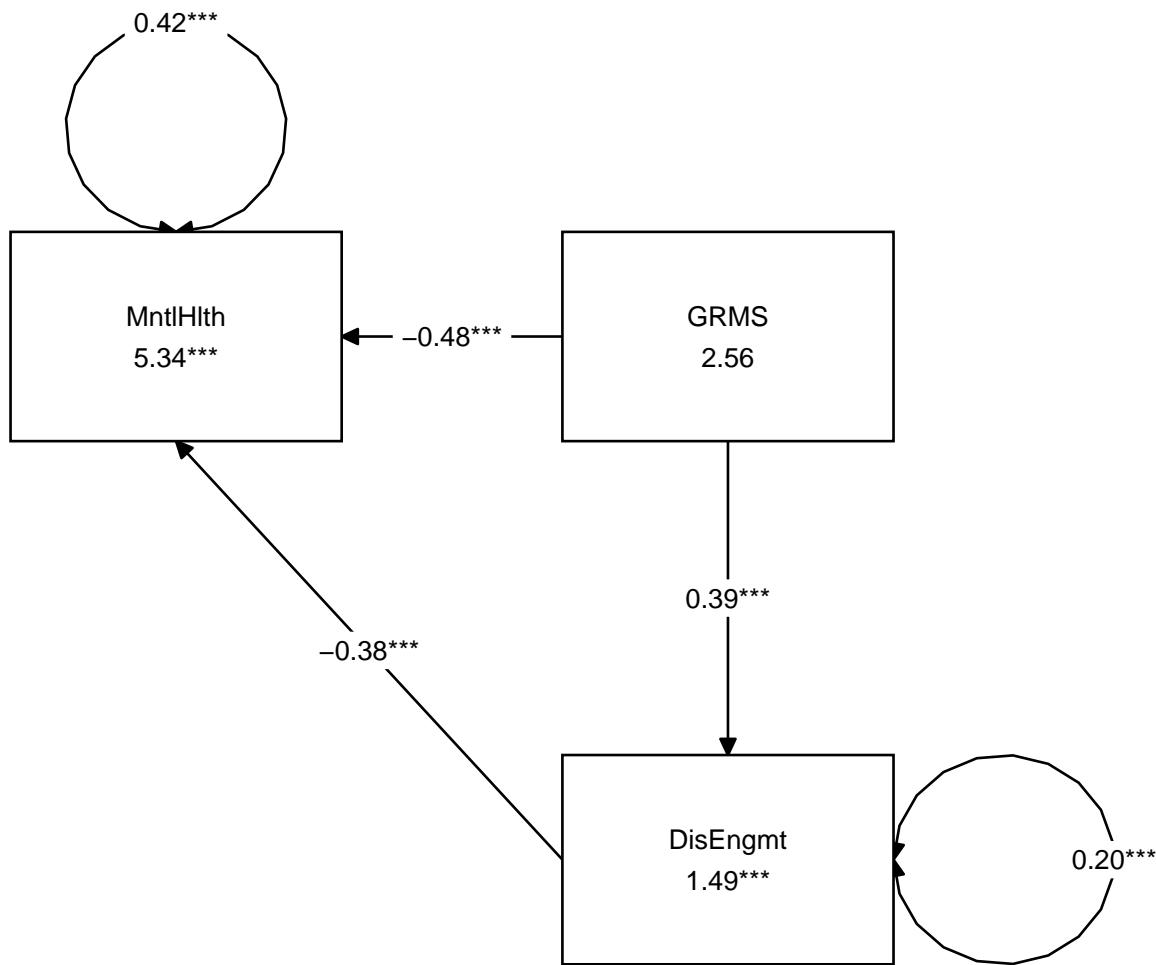
## The following object is masked from 'package:psych':
##      tr

## Registered S3 method overwritten by 'tidySEM':
##   method      from
##   predict.MxModel OpenMx

##
## Attaching package: 'tidySEM'

## The following object is masked from 'package:jtools':
##      get_data

tidySEM::graph_sem(model = LMed_fit)
```



We can use the `tidySEM::get_layout` function to understand how our model is being mapped.

```
tidySEM::get_layout(LMed_fit)
```

```
##      [,1]      [,2]
## [1,] NA        "GRMS"
## [2,] "MntlHlth" "DisEngmt"
## attr(),"class")
## [1] "layout_matrix" "matrix"       "array"
```

We can write code to remap them

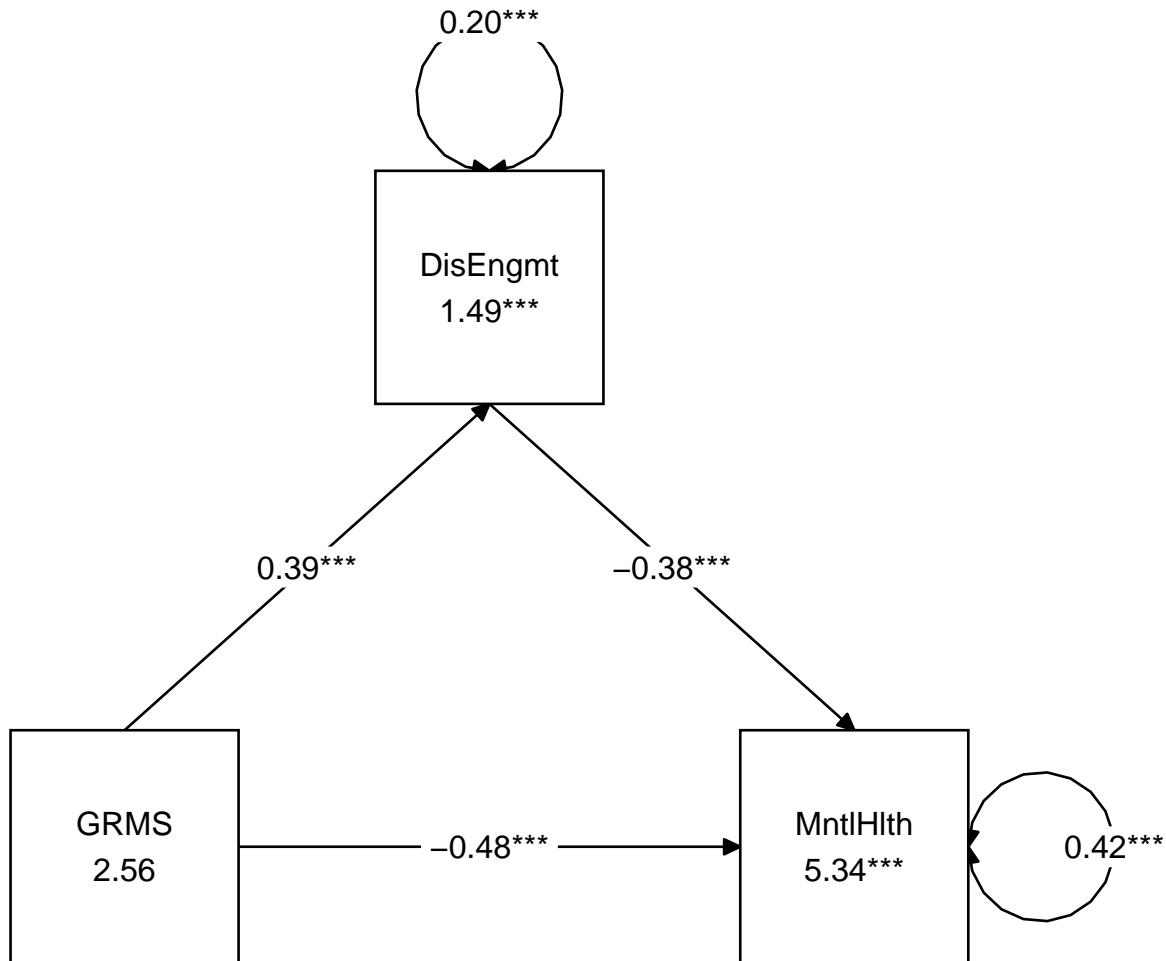
```
Lmedmap <- tidySEM::get_layout("", "DisEngmt", "", "GRMS", "", "MntlHlth",
  rows = 2)
Lmedmap
```

```
##      [,1]      [,2]      [,3]
```

```
## [1] ""      "DisEngmt" ""
## [2] "GRMS" ""           "MntlHlth"
## attr(,"class")
## [1] "layout_matrix" "matrix"      "array"
```

We can update the `tidySEM::graph_sem` function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(LMed_fit, layout = Lmedmap, rect_width = 1.25, rect_height = 1.25,
                    spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



8.6 The Moderated Mediation: A Combined analysis

For a quick reminder, the diagram with labeled paths will help specify this in *lavaan*.

Looking at the diagram, with two consequent variables (i.e., those with arrows pointing to them) we can see two equations are needed to explain the model:

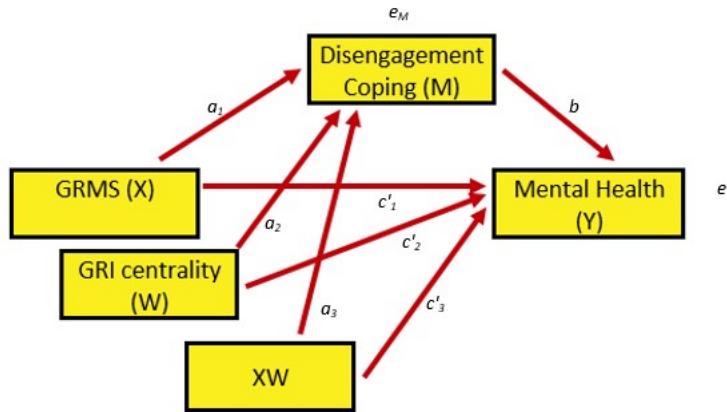


Figure 8.7: Image of statistical representation of the conditional process analysis model where the moderator is hypothesized to change the a and c' paths

$$M = i_M + a_1 X + a_2 W + a_3 XW + e_M$$

$$Y = i_Y + c'_1 X + c'_2 W + c'_3 XW + bM + e_Y$$

Y = MntlHlth X = GRMS W = DisEngmt M = GRIcntlty

8.6.1 Specification in *lavaan*

In the code below

- First specify the equations, hints
 - the a,b,c, labels are affixed with the *(asterisk)
 - interaction terms are identified with the colon
- Create code for the intercepts (Y and M) with the form: VarName ~ VarName.mean*1
- Create code for the mean and variance of all moderators (W, Z, etc.); these will be used in simple slopes.
 - Means use the form: VarName ~ VarName.mean*1
 - Variances use the form: VarName ~ VarName.var*VarName
- Calculate the *index of moderated mediation*: quantifies the relationship between the moderator and the indirect effect.
 - To the degree that the value of the IMM is different from zero and the associated inferential test is statistically significant (bootstrapped confidence intervals are preferred; more powerful), we can conclude that the indirect effect is moderated.
 - * The IMM is used in the formula to calculate the conditional indirect effects.
 - * Hayes argues that a statistically significant IMM suggest they are (boom, done, p. 430).

- Create code to calculate indirect effects conditional on ($M \pm 1SD$) moderator with the general form:
 - product of the indirect effect ($a*b$) PLUS
 - the product of the IMM and the moderated value
- Because our direct path is moderated, we will use a similar process to specify the direct effects conditional on ($M \pm 1SD$) moderator with the general form:
 - the direct effect (c_p1) PLUS
 - the moderated value (c_p3) at each of the three levels ($M \pm 1SD$)
- Although they don't tend to be reported, you can create total effects conditional on the ($M \pm 1SD$). These are simply the sum of the c_p and all indirect paths, specified individually, at their $M \pm 1SD$ conditional values.

```

Combined <- '
  #equations
  DisEngmt ~ a1*GRMS + a2*Centrality + a3*GRMS:Centrality
  MntlHlth ~ c_p1*GRMS + c_p2*Centrality + c_p3*GRMS:Centrality + b*DisEngmt

  #intercepts
  DisEngmt ~ DisEngmt.mean*1
  MntlHlth ~ MntlHlth.mean*1

  #means, variances of W for simple slopes
  Centrality ~ Centrality.mean*1
  Centrality ~~ Centrality.var*Centrality

  #index of moderated mediation, there will be an a and b path in the product
  #if the a and/or b path is moderated, select the label that represents the moderation

  imm := a3*b

  #Note that we first create the indirect product, then add to it the product of the imm and
  indirect.SDbelow := a1*b + imm*(Centrality.mean - sqrt(Centrality.var))
  indirect.mean := a1*b + imm*(Centrality.mean)
  indirect.SDabove := a1*b + imm*(Centrality.mean + sqrt(Centrality.var))

  #direct effect is also moderated so calculate with c_p1 + c_p3
  direct.SDbelow := c_p1 + c_p3*(Centrality.mean - sqrt(Centrality.var))
  direct.Smean := c_p1 + c_p3*(Centrality.mean)
  direct.SDabove := c_p1 + c_p3*(Centrality.mean + sqrt(Centrality.var))

  '

  set.seed(230925) #required for reproducible results because lavaan introduces randomness into
  Combined_fit <- lavaan::sem(Combined, data = Lewis_df, se = "bootstrap", missing = 'fiml', boot
  
```

Warning in lav_partable_vnames(FLAT, "ov.x", warn = TRUE): lavaan WARNING:

```

## model syntax contains variance/covariance/intercept formulas
## involving (an) exogenous variable(s): [Centrality]; These
## variables will now be treated as random introducing additional
## free parameters. If you wish to treat those variables as fixed,
## remove these formulas from the model syntax. Otherwise, consider
## adding the fixed.x = FALSE option.

cFITsum <- lavaan:::summary(Combined_fit, standardized = TRUE, rsq=T, ci=TRUE)
cParamEsts <- lavaan:::parameterEstimates(Combined_fit, boot.ci.type = "bca.simple", standardize=TRUE)
cFITsum

## lavaan 0.6.16 ended normally after 20 iterations
##
## Estimator                               ML
## Optimization method                    NLMINB
## Number of model parameters           13
##
## Number of observations                231
## Number of missing patterns            1
##
## Model Test User Model:
##
## Test statistic                         567.225
## Degrees of freedom                   2
## P-value (Chi-square)                 0.000
##
## Parameter Estimates:
##
## Standard errors                      Bootstrap
## Number of requested bootstrap draws  1000
## Number of successful bootstrap draws 1000
##
## Regressions:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## DisEngmt ~
##   GRMS    (a1)  0.623  0.194  3.212  0.001  0.166  0.942
##   Cntrlty (a2)  0.093  0.137  0.680  0.497 -0.215  0.335
##   GRMS:Cn (a3) -0.058  0.050 -1.142  0.253 -0.147  0.055
## MntlHlth ~
##   GRMS    (c_p1) -1.023  0.299 -3.417  0.001 -1.572 -0.389
##   Cntrlty (c_p2) -0.317  0.190 -1.671  0.095 -0.669  0.070
##   GRMS:Cn (c_p3) 0.136  0.074  1.841  0.066 -0.017  0.274
##   DsEngmt   (b)  -0.362  0.091 -3.965  0.000 -0.530 -0.176
## Std.lv  Std.all
##
##   0.623   0.846
##   0.093   0.133

```

```

## -0.058 -0.414
##
## -1.023 -0.846
## -0.317 -0.275
## 0.136 0.593
## -0.362 -0.220
##
## Intercepts:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .DsEngmt (DsE.) 1.107   0.521   2.126   0.034   0.276    2.325
## .MntlHlt (MnH.) 6.539   0.714   9.155   0.000   4.946    7.863
## Cntrlty (Cnt.)  3.938   0.049  79.569   0.000   3.836    4.035
## Std.lv Std.all
## 1.107   2.090
## 6.539   7.515
## 3.938   5.223
##
## Variances:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## Cntrlty (Cnt.) 0.568   0.054  10.608   0.000   0.464    0.675
## .DsEngmt        0.194   0.016  12.001   0.000   0.160    0.224
## .MntlHlt        0.412   0.040  10.379   0.000   0.333    0.482
## Std.lv Std.all
## 0.568   1.000
## 0.194   0.692
## 0.412   0.545
##
## R-Square:
##             Estimate
## DisEngmt      0.308
## MntlHlth     0.455
##
## Defined Parameters:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## imm           0.021   0.020   1.064   0.287   -0.022    0.060
## indirect.SDblw -0.159   0.045  -3.574   0.000   -0.247   -0.075
## indirect.mean -0.143   0.040  -3.600   0.000   -0.219   -0.064
## indirect.SDabv -0.128   0.040  -3.163   0.002   -0.209   -0.050
## direct.SDbelow -0.591   0.089  -6.661   0.000   -0.750   -0.404
## direct.Smean   -0.488   0.067  -7.251   0.000   -0.613   -0.356
## direct.SDabove -0.386   0.085  -4.521   0.000   -0.546   -0.216
## Std.lv Std.all
## 0.021   0.091
## -0.159   0.199
## -0.143   0.290
## -0.128   0.381
## -0.591   1.657
## -0.488   2.249

```

```
##      -0.386    2.842
```

```
cParamEsts
```

	lhs	op		rhs					
## 1	DisEngmt	~		GRMS					
## 2	DisEngmt	~		Centrality					
## 3	DisEngmt	~		GRMS:Centrality					
## 4	MntlHlth	~		GRMS					
## 5	MntlHlth	~		Centrality					
## 6	MntlHlth	~		GRMS:Centrality					
## 7	MntlHlth	~		DisEngmt					
## 8	DisEngmt	~1							
## 9	MntlHlth	~1							
## 10	Centrality	~1							
## 11	Centrality	~~		Centrality					
## 12	DisEngmt	~~		DisEngmt					
## 13	MntlHlth	~~		MntlHlth					
## 14	GRMS	~~		GRMS					
## 15	GRMS	~~		GRMS:Centrality					
## 16	GRMS:Centrality	~~		GRMS:Centrality					
## 17	GRMS	~1							
## 18	GRMS:Centrality	~1							
## 19	imm	:=		a3*b					
## 20	indirect.SDbelow	:=	a1*b+imm*(Centrality.mean-sqrt(Centrality.var))						
## 21	indirect.mean	:=	a1*b+imm*(Centrality.mean)						
## 22	indirect.SDabove	:=	a1*b+imm*(Centrality.mean+sqrt(Centrality.var))						
## 23	direct.SDbelow	:=	c_p1+c_p3*(Centrality.mean-sqrt(Centrality.var))						
## 24	direct.Smean	:=	c_p1+c_p3*(Centrality.mean)						
## 25	direct.SDabove	:=	c_p1+c_p3*(Centrality.mean+sqrt(Centrality.var))						
	label	est	se	z	pvalue	ci.lower	ci.upper	std.lv	std.all
## 1	a1	0.623	0.194	3.212	0.001	0.159	0.937	0.623	0.846
## 2	a2	0.093	0.137	0.680	0.497	-0.244	0.317	0.093	0.133
## 3	a3	-0.058	0.050	-1.142	0.253	-0.143	0.060	-0.058	-0.414
## 4	c_p1	-1.023	0.299	-3.417	0.001	-1.601	-0.420	-1.023	-0.846
## 5	c_p2	-0.317	0.190	-1.671	0.095	-0.669	0.070	-0.317	-0.275
## 6	c_p3	0.136	0.074	1.841	0.066	-0.009	0.281	0.136	0.593
## 7	b	-0.362	0.091	-3.965	0.000	-0.527	-0.171	-0.362	-0.220
## 8	DisEngmt.mean	1.107	0.521	2.126	0.034	0.328	2.410	1.107	2.090
## 9	MntlHlth.mean	6.539	0.714	9.155	0.000	4.983	7.866	6.539	7.515
## 10	Centrality.mean	3.938	0.049	79.569	0.000	3.832	4.035	3.938	5.223
## 11	Centrality.var	0.568	0.054	10.608	0.000	0.469	0.682	0.568	1.000
## 12		0.194	0.016	12.001	0.000	0.166	0.235	0.194	0.692
## 13		0.412	0.040	10.379	0.000	0.345	0.510	0.412	0.545
## 14		0.518	0.000	NA	NA	0.518	0.518	0.518	1.000
## 15		2.334	0.000	NA	NA	2.334	2.334	2.334	0.853
## 16		14.446	0.000	NA	NA	14.446	14.446	14.446	1.000

```

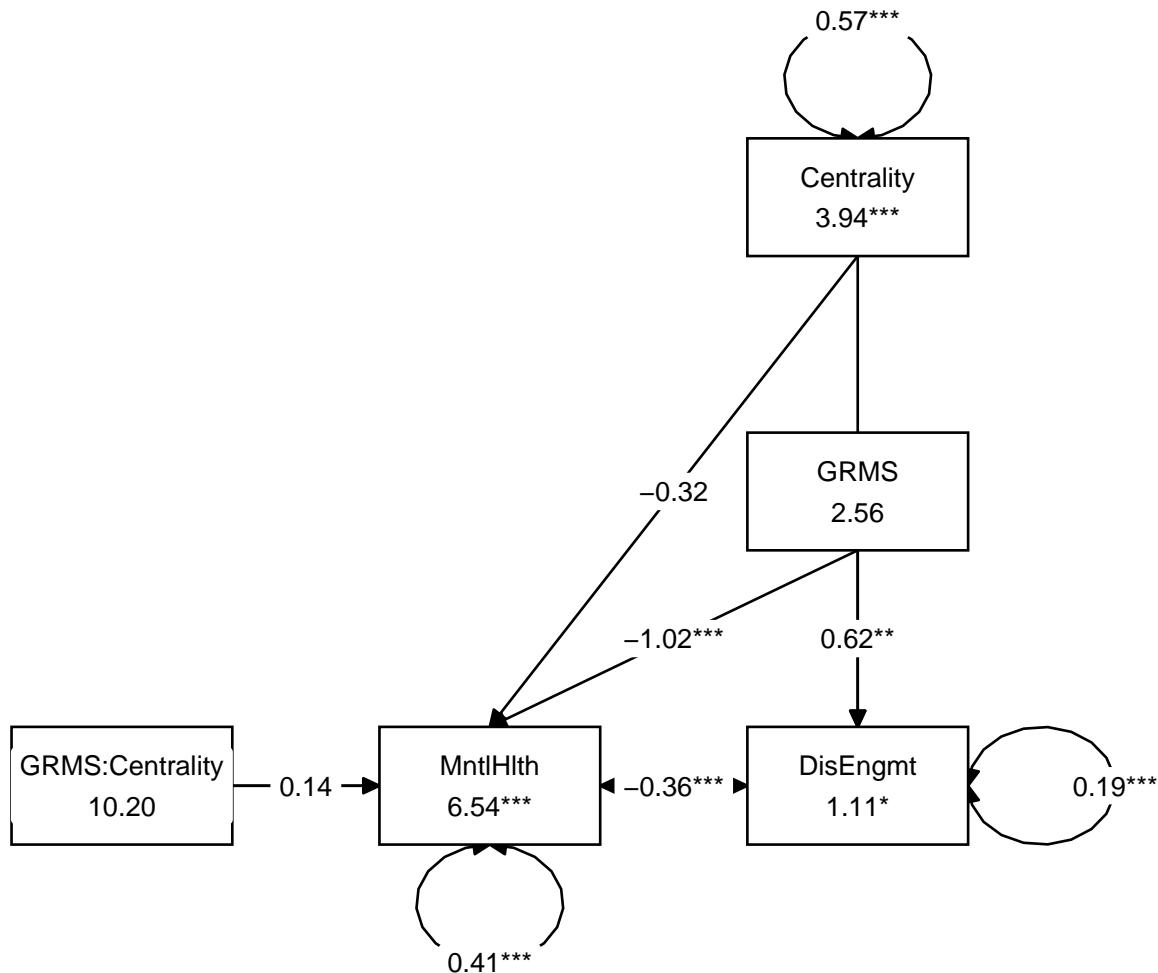
## 17          2.557 0.000    NA    NA   2.557   2.557  2.557  3.552
## 18          10.199 0.000   NA    NA  10.199  10.199 10.199  2.683
## 19          imm  0.021 0.020  1.064  0.287 -0.017  0.063  0.021  0.091
## 20 indirect.SDbelow -0.159 0.045 -3.574  0.000 -0.253 -0.083 -0.159  0.199
## 21 indirect.mean -0.143 0.040 -3.600  0.000 -0.226 -0.073 -0.143  0.290
## 22 indirect.SDabove -0.128 0.040 -3.163  0.002 -0.229 -0.061 -0.128  0.381
## 23 direct.SDbelow -0.591 0.089 -6.661  0.000 -0.754 -0.410 -0.591  1.657
## 24 direct.Smean -0.488 0.067 -7.251  0.000 -0.613 -0.355 -0.488  2.249
## 25 direct.SDabove -0.386 0.085 -4.521  0.000 -0.549 -0.218 -0.386  2.842
## std.nox
## 1   1.176
## 2   0.127
## 3  -0.109
## 4  -1.175
## 5  -0.262
## 6   0.156
## 7  -0.220
## 8   2.090
## 9   7.515
## 10  5.223
## 11  1.000
## 12  0.692
## 13  0.545
## 14  0.518
## 15  2.334
## 16 14.446
## 17  2.557
## 18 10.199
## 19  0.024
## 20 -0.158
## 21 -0.134
## 22 -0.110
## 23 -0.517
## 24 -0.361
## 25 -0.205

```

```

# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = Combined_fit)

```



We can use the `tidySEM::get_layout` function to understand how our model is being mapped.

```
tidySEM::get_layout(Combined_fit)
```

```
##      [,1]      [,2]      [,3]
## [1,] NA "Centrality" "GRMS"
## [2,] "GRMS:Centrality" "MntlHlth" "DisEngmt"
## attr(,"class")
## [1] "layout_matrix" "matrix"       "array"
```

We can write code to remap them

```
comb_map <- tidySEM::get_layout("", "DisEngmt", "", "GRMS", "", "MntlHlth",
  "Centrality", "", "", "", "GRMS:Centrality", "", rows = 4)
comb_map
```

```
##      [,1]      [,2]      [,3]
```

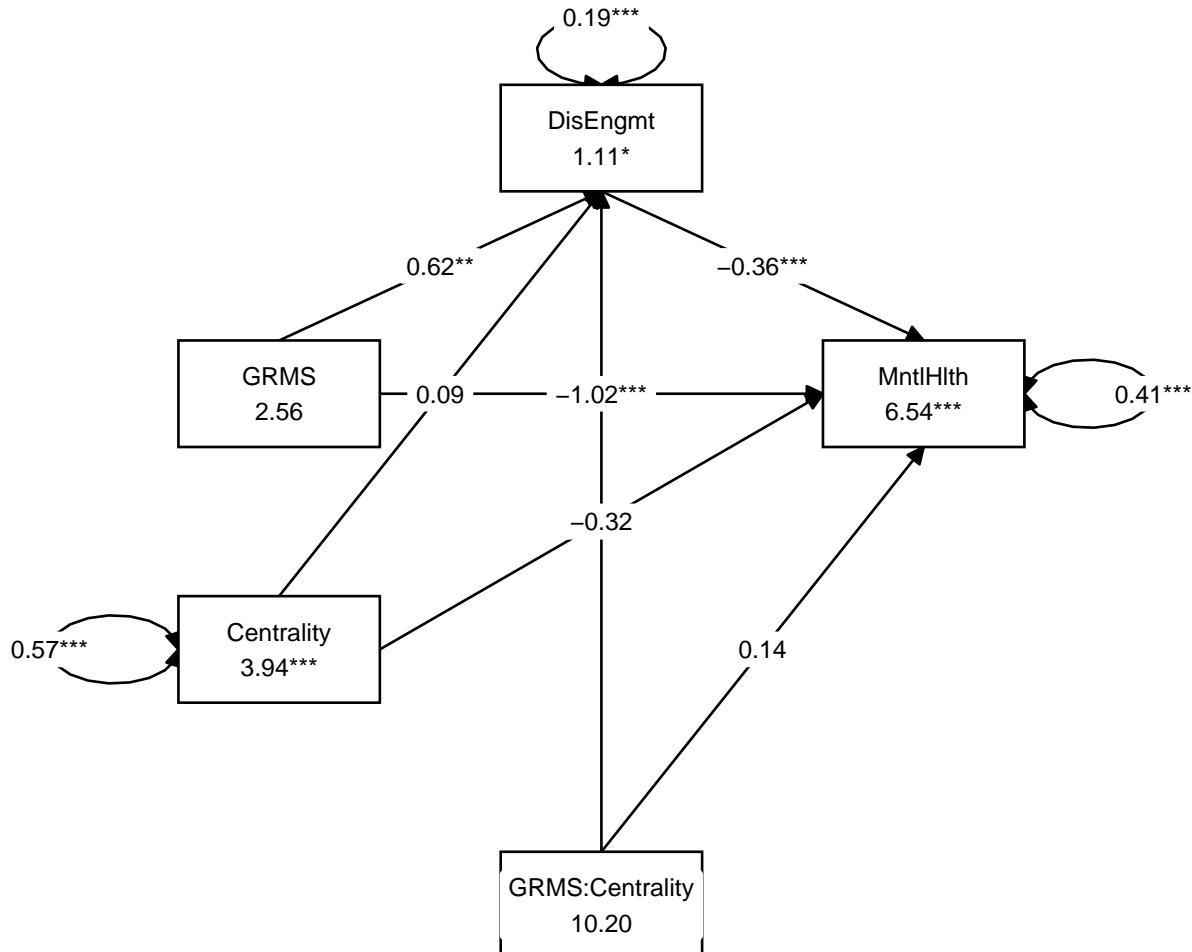
```

## [1] ""           "DisEngmt"      ""
## [2] "GRMS"        ""           "MntlHlth"
## [3] "Centrality"   ""           ""
## [4] ""           "GRMS:Centrality" ""
## attr(),"class")
## [1] "layout_matrix" "matrix"      "array"

```

We can update the `tidySEM::graph_sem` function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(Combined_fit, layout = comb_map, rect_width = 1.25,
rect_height = 1.25, spacing_x = 2, spacing_y = 3, text_size = 3.5)
```



```
write.csv(cParamEsts, file = "Combined_fit.csv") #optional to write it to a .csv file
```

8.6.2 Beginning the interpretation

We have already looked at some of the simple effects found within the more complex model. Let's grab the formulae.

$$\hat{M} = i_M + a_1 X + a_2 W + a_3 XW + e_M$$

$$\hat{Y} = i_Y + c'_1 X + c'_2 W + c'_3 XW + bM + e_Y$$

And substitute in our values

$$\hat{M} = 1.417 + 0.212X + (-0.027)W + 0.006XW$$

$$\hat{Y} = 31.703 + (-1.4115)X + (-0.556)W + 0.164XW + (-3.567)M$$

8.6.3 Tabling the data

Table 1

Analysis of Moderated Mediation for GRMS, Gendered Racial Identity Centrality, Coping, and Mental Health

Predictor	B	SE _B	p	R ²
Disengagement coping (M)				.31
Constant	1.107	0.521	0.034	
GRMS (a_1)	0.623	0.194	0.001	
Centrality (a_2)	0.093	0.137	0.497	
GRMS:Centrality (a_3)	-0.058	0.050	0.253	
Mental Health (DV)				.46
Constant	6.539	0.714	<0.001	
GRMS (c'_1)	-1.023	0.299	0.001	
Centrality (c'_2)	-0.317	0.190	0.095	
GRMS:Centrality (c'_3)	0.136	0.074	0.066	
Disengagement (b)	-0.362	0.091	<0.001	
Summary of Effects	B	SE _B	p	95% CI
IMM	0.021	0.020	0.287	-0.017, 0.063

Summary of Effects	<i>B</i>	<i>SE_B</i>	<i>p</i>	95% CI
Indirect ($-1SD$)	-0.159	0.045	<0.001	-0.253, -0.083
Indirect (M)	-0.143	0.040	<0.001	-0.226, -0.073
Indirect ($+1SD$)	-0.128	0.040	0.002	-0.229, -0.061
Direct ($-1SD$)	-0.591	0.089	<0.001	-0.754, -0.410
Direct (M)	-0.488	0.067	<0.001	-0.613, -0.355
Direct ($+1SD$)	-0.386	0.085	<0.001	-0.549, -0.218

Note. GRMS = gendered racial microaggressions. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

Thirty one percent of the variance in disengagement coping (mediator) and 46% of the variance in mental health (DV) are predicted by their respective models.

The model we tested suggested that both the indirect and direct effects should be moderated.

8.6.3.1 Conditional Indirect effects

An indirect effect can be moderated if either the *a* or *b* path (or both) is(are) moderated. If at least one of the indirect paths is part of a moderation, then the whole indirect (axb) path would be moderated. In this model, we specified a moderation of the *a* path. We know the *a* path is moderated if the moderation term is statistically significant.

In our case, a_3 GRMS:Centrality was not statistically significant ($B = -0.058, p = 0.253$). We also inspect the *Index of Moderated Mediation*. The IMM is the product of the moderated path (in this case, the value of a_3) and *b*. If this index is 0, then the slope of the line for the indirect effect is flat. The bootstrap confidence interval associated with this test is the way to determine whether/not this slope is statistically significant from zero. In our case, $IMM = 0.021$ ($p = 0.287\%$) with the 95% confidence interval ranging from $CI_{0.95} = -0.017$ to 0.063 . This suggests that we do not have a moderated mediation. Hayes claims the IMM saves us from formally comparing (think “contrasts” pairs of conditional indirect effects).

We can obtain more information about the potentially moderated indirect effect by *probing the conditional indirect effect*. Because an indirect effect is not normally distributed, Hayes discourages using a Johnson-Neyman approach and suggests that we use a pick-a-point. He usually selects the 16th, 50th, and 84th percentiles of the distribution. However, many researchers commonly report the mean \pm 1SD.

- at 1SD below the mean $B = -0.159, p < 0.001, 95CI(-0.253, -0.083)$;
- at the mean $B = -0.488, p < 0.001, 95CI(-0.613, -0.355)$;
- at 1SD above the mean, $B = -0.128, p = 0.002, 95CI(-0.229, -0.061)$.

Examining the relative consistency of the *B* weights and the consistently significant *p* values, we see that there was an indirect effect throughout the varying levels of the moderator, gendered racial identity centrality. Thus, it makes sense that this was not a *moderated mediation*.

8.6.3.2 Conditional Direct effect

The direct effect of X to Y estimates how differences in X relate to differences in Y holding constant the proposed mediator(s). We know the direct effect is moderated if the interaction term ($c'_p 3$) is statistically significant. In our case, it was not $B = 0.136, p = 0.066$. Probing a conditional direct effect is straightforward – we typically use the same points as we did in the probing of the conditional indirect effect.

- at 1SD below the mean $B = -0.591, p < 0.001, 95CI(-0.754, -0.410)$;
- at the mean $B = 0.143, p < 0.001, 95CI(-0.226, -0.073)$;
- at 1SD above the mean, $B = -0.386, p < 0.001, 95CI(-0.549, -0.218)$.

The statistically significant effect of GRMS on mental health at the three levels of gendered racial identity centrality is consistent with the non-significant interaction effect.

8.6.4 Model trimming

Hayes terms it *pruning* when he suggests that when there is no moderation of an effect, the researcher may want to delete that interaction term. In our case, neither the direct nor indirect effect was moderated (although the $+1SD$ was close ($B = 0.136, p = 0.066$)). Deleting these paths one at a time is typical practice because the small boost of power with each deleted path may “turn on” significance elsewhere. If I were to engage in model trimming, I would start with the indirect effect to see if the interaction term associated with the direct effect became statistically significant. This is consistent with the simple moderation we ran earlier where we saw a fanning out at one end of the distribution.

8.6.5 APA Style Write-up

As we look to write up our own results I encourage you to review the manuscript that sources our research vignette. The Lewis et al. [2017] write-up is an efficient one, simultaneously presenting the results of two outcome variables – mental and physical health. While our B weights from our simulated data map similarly onto those reported in the Lewis et al. manuscript, we do not get the statistically significant moderated mediation reported in the article.

Method/Aalytic Strategy

Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16). We specified a moderated mediation model predicting mental health from gendered racial microaggressions, mediated by disengagement coping. We further predicted that the relationships between gendered racial microaggressions to disengagement coping (i.e., the a path) and between gendered racial microaggressions to mental health (i.e., the c^1 path) would be moderated by gendered racial identity centrality.

Results

Preliminary Analyses

- Missing data analysis and managing missing data

- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

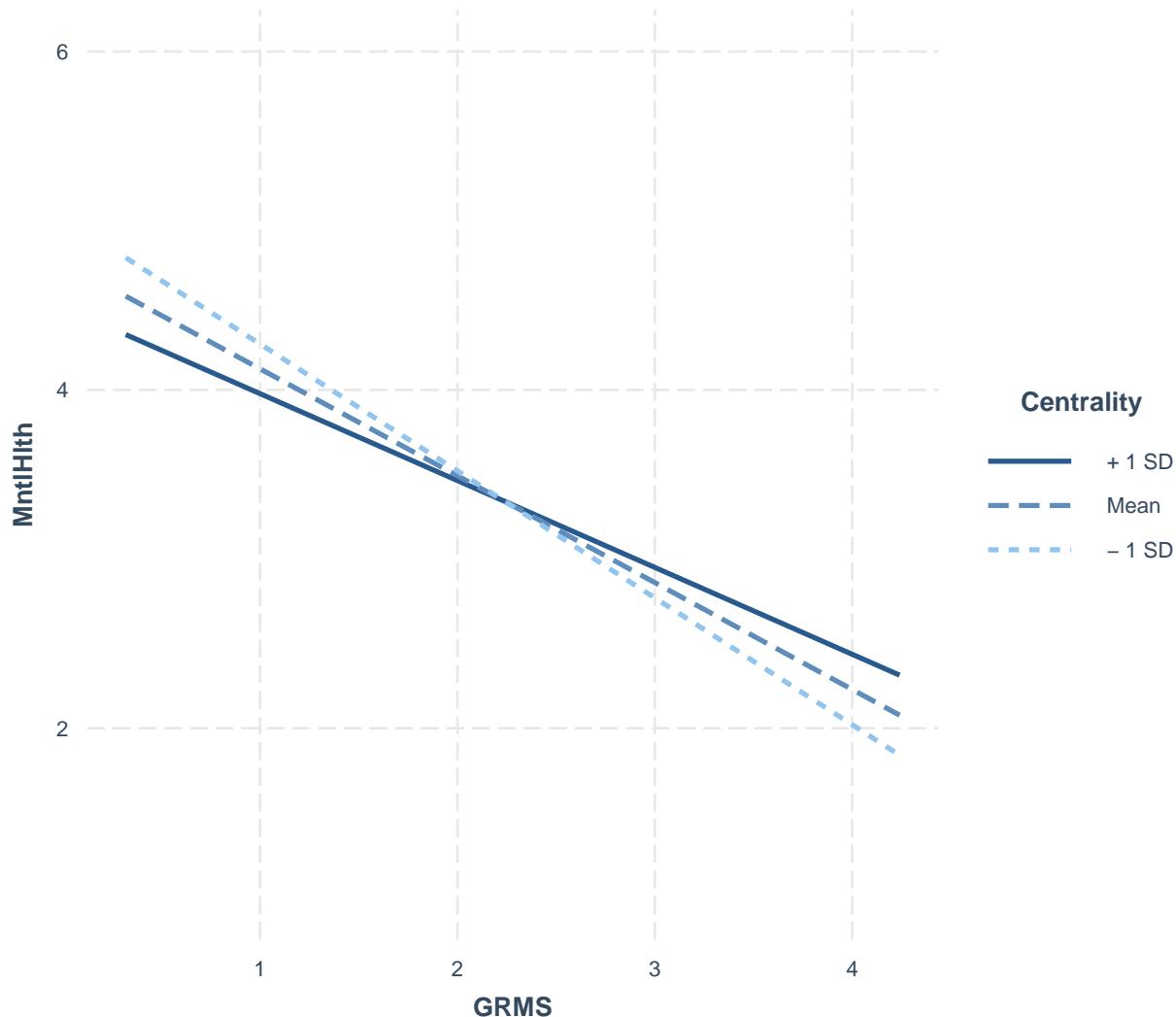
Primary Analyses Our analysis evaluated a moderation mediation model predicting mental health ($Y/\text{MntlHlth}$) from gendered racial microaggressions (X/GRMS) mediated by disengagement coping ($M/\text{DisEngmt}$). Gendered racial identity centrality ($W/\text{GRICntrlty}$) was our moderating variable. We specified a moderation of path a (X/GRMS to $M/\text{DisEngmt}$) and the direct path, c' (X/GRMS to $Y/\text{MntlHlth}$). Data were analyzed with maximum likelihood estimation in the R package *lavaan* (v. 0.6-7); the significance of effects were tested with 1000 bootstrap confidence intervals. Results of the full model are presented in Table 1 and illustrated in Figure 1 (*a variation of the semPlot or Hayes style representation*). The formula for the mediator and dependent variable are expressed below.

$$\hat{M} = 1.107 + 0.623X + (0.093)W + -0.058XW$$

$$\hat{Y} = 6.539 + (-1.023)X + (-0.317)W + 0.136XW + (-0.362)M$$

Results suggested a negative effect of gendered racial microaggressions on mental health that is mediated through disengagement coping. That is, in the presence of gendered racial microaggressions, participants increased disengagement coping which, in turn, had negative effects on mental health. The index of moderated mediation was not significant ($B = 0.021, p = 0.287, 95CI(-0.017, 0.063)$), suggesting that the indirect effects were not conditional on the values of the moderator. While there was no evidence of moderation on the indirect or direct paths, there was a statistically significant, and consistently strong, mediation throughout the range of the gendered racial identity centrality (moderator). *Because we did not have a moderated mediation, I would probably not include the rest of this paragraph (nor include the moderation figure). I just wanted to demonstrate how to talk about findings if they were significant (although I acknowledg throughat that these are non-significant).* Figure 2 illustrates the conditional effects (non-significant) of GRMS (X) on mental health (Y) among those at the traditional levels of $M \pm 1SD$. Our model accounted for 31% of the variance in the mediator (disengagement coping) and 46% of the variance in the dependent variable (mental health).

```
interactions:::interact_plot(Mod_c_path, pred = GRMS, modx = Centrality) +
  ylim(1, 6)
```



```
# interactions::interact_plot(Mod_c_path, pred = GRIcntly, modx =
# GRMS + ylim(1,6))
```

8.7 STAY TUNED

A section on power analysis is planned and coming soon! My apologies that it's not quite Ready.

8.8 Residual and Related Questions...

..that you might have; or at least I had, but if had answered them earlier it would have disrupt the flow.

1. Would you stop here? Or keep tinkering?

- I am tempted to delete moderation of the indirect effect to see if I can get a moderated direct effect. My choice would also depend on what I committed in any kind of pre-registration. My approach to science tends to be *model generating* [Joreskog, 1993] and in his text, Hayes [2022a] advised authors to write about what they found – not all the things they tried. This *tinkering* remains strongly in the vein of theoretically driven analyses.
2. The output we get is different from the output in the journal article being used as the research vignette. Why? And should we worry about it?
- We are simulating data. This gives us some advantages in that (unless we specify it), we never have missingness and our variables should be normally distributed. Because we are working from means and correlations, our data will never be the same as the original researcher.
3. Some of the statistics you are reporting are different than the ones in Hayes and the ones that use the PROCESS macro (e.g., what happened to the *F* test)?
- The default estimator for *lavaan* is maximum likelihood (ML) and Hayes uses ordinary least squares (OLS). This affects both the values of coefficients, standard errors, AND the type of statistics that are reported.
 - You can ask for OLS regression by adding the statement “estimator =”GLS”. Even with this option, I have not discovered a way to obtain the *F* tests for the overall model. Researchers seem to be comfortable with this, even asking for less than we did (e.g., many do not request R square).
 - Best I can tell, researchers who do want this might use a combination of packages, using GLS estimators in *lavaan* (this easily gets them the bootstrapped CIs) and the move to a different regression package to get the intercepts and *F* tests. If I did this I would triple check to make sure that all the output really lined up.
4. Why did you run the two simple moderations with the base R *lm()* function (in OLS) and the moderated mediation in *lavaan* (using MLE)?
- As demonstrated in the prior chapter on simple moderation, the regression weights associated with the OLS and MLE analyses should be the same. Given that I’m conducting the simple mediations and moderations with a primary purpose of understanding the simpler elements of the larger model, either approach will be useful in helping me understand the smaller sets of relations.
 - Very practically, using the base R *lm()* approach will help me create interaction plots that will be useful in presenting the results.
5. Why did we ignore the traditional fit statistics associated with structural equation modeling (e.g., CFI, RMSEA).
- I hesitate to do this with models that do not include latent variables. Therefore, we asked for an “in-between” amount of info that should be sufficient for publication submission (any editor may have their own preferences and ask for more).
6. What if I have missing data?
- When we enter the *lavaan* world we do get options other than multiple imputation. In today’s example we used the “sem” fitting function. Unless otherwise specified, listwise

deletion (deleting the entire case when one of its variables is used to estimate the model) is the default in *lavaan*. If data are MCAR or MAR, you can add the argument *missing* = “*ml*” (or its alias *missing* = “*fiml*”). More here <https://users.ugent.be/~yrosseel/lavaan/lavaan2.pdf> on the 1.7/Missing data in lavaan slide.

- That said, the type of estimator matters. If you estimate your data with GLS (generalized least squares) or WLS (weighted least squares), you are required to have complete data (however you got it). We used maximum likelihood and, even though we had non-missing data, I used the *missing* = “*fiml*” code.

8.9 Practice Problems

The three problems described below were designed to grow during the series of chapters on simple and complex mediation, complex moderation, and conditional process analysis (i.e., this chapter). I have recommended that you select a dataset that includes at least four variables. If you are new to this topic, you may wish to select variables that are all continuously scaled. The IV and moderator (next chapters) could be categorical (if they are dichotomous, please use 0/1 coding; if they have more than one category it is best if they are ordered). You will likely encounter challenges that were not covered in this chapter. Search for and try out solutions, knowing that there are multiple paths through the analysis.

The suggested practice problem for this chapter is to conduct a moderated mediation. At least one path (a or b) should be moderated.

8.9.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed

If this topic feels a bit overwhelming, simply change the random seed in the data simulation, then rework the problem. This should provide minor changes to the data (maybe in the second or third decimal point), but the results will likely be very similar.

8.9.2 Problem #2: Rework the research vignette, but swap one or more variables

Use the simulated data, but select one of the other models that was evaluated in the Lewis et al. [2017] study. For example, physical health was also used as a dependent variable in a separate but otherwise parallel analysis. Compare your results to those reported in the manuscript.

8.9.3 Problem #3: Use other data that is available to you

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 a simple mediation.

8.9.4 Grading Rubric

Assignment Component		
1. Describing the overall model hypothesis, assign each variable to the X, Y, M, or W roles	5	_____
2. Import and format the variables in the model	5	_____
3. Using a piecewise approach, run each of the simple models in the grander design	5	_____
4. Specify and run the entire lavaan model	5	_____
5. Use tidySEM to create a figure	5	_____
6. Create a table that includes regression output for the M and Y variables and the moderated effects	5	_____
7. Represent your work in an APA-style write-up	5	_____
8. Explanation to grader	5	_____
9. Be able to hand-calculate the indirect, direct, and total effects from the <i>a</i> , <i>b</i> , & <i>c'</i> paths	5	_____
Totals	45	_____

8.10 Homeworked Example 1: A moderation on the *a* path

Screencast Link

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to conduct a moderated mediation. At least one path (*a* or *b*) should be moderated.

Describing thy overall model hypothesis, assign each variable to the X, Y, M, and W roles

My analysis will evaluated a moderated mediation. Specifically, I predict that the effect of centering on perceived value to the student will be moderated by the students' evaluation of socially responsive pedagogy. I further hypothesize that this indirect effect will be moderated by traditional pedagogy and that the moderation will occur on the *a* path, that is, traditional pedagogy will moderate the effect of centering on socially responsive pedagogy.

- X = Centering, pre/re (0,1)
- W = Traditional Pedagogy (1 to 4 scaling)
- M = Socially Responsive Pedagogy (1 to 4 scaling)
- Y = Value to the student (1 to 4 scaling)

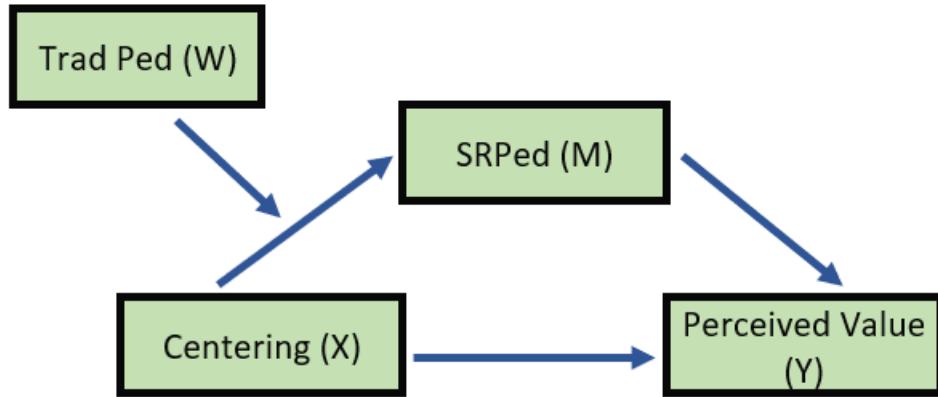


Figure 8.8: An image of the conceptual model of moderated mediation for the homeworked example.

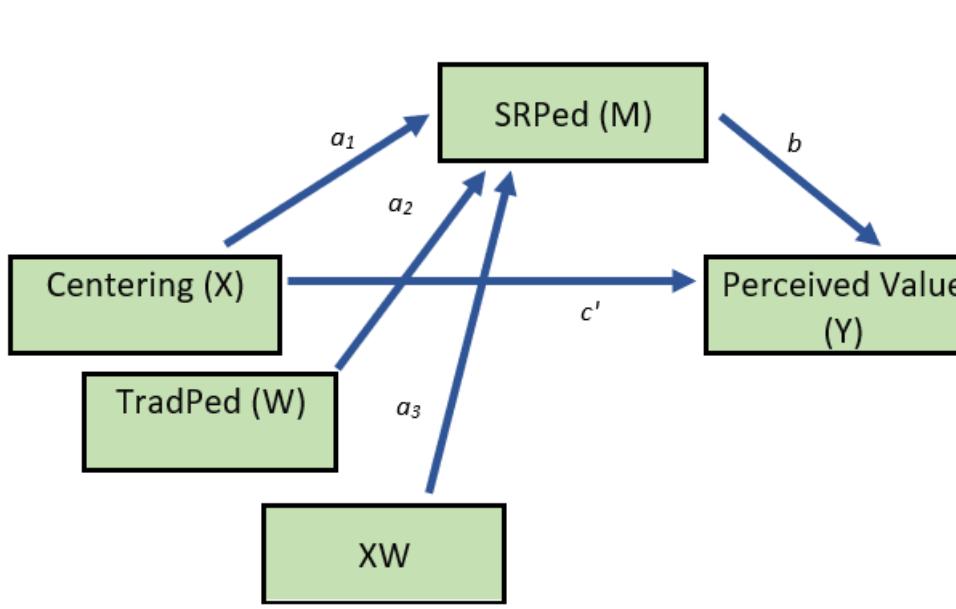


Figure 8.9: An image of the statistical model of simple moderation for the homeworked example.

Import the data and format the variables in the model

```
raw <- readRDS("ReC.rds")
```

The approach we are taking to moderated mediation does not allow dependency in the data. Therefore, we will include only those who took the multivariate class (i.e., excluding responses for the ANOVA and psychometrics courses).

```
raw <- (dplyr::filter(raw, Course == "Multivariate"))
```

I need to score the SRPed, TradPed, and Valued variables

```
TradPed_vars <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback",
  "ClearOrganization", "ClearPresentation")
raw$TradPed <- sjstats::mean_n(raw[, ..TradPed_vars], 0.75)

Valued_vars <- c("ValObjectives", "IncrUnderstanding", "IncrInterest")
raw$Valued <- sjstats::mean_n(raw[, ..Valued_vars], 0.75)

SRPed_vars <- c("InclusvClassrm", "EquitableEval", "MultPerspectives",
  "DEIntegration")
raw$SRPed <- sjstats::mean_n(raw[, ..SRPed_vars], 0.75)
```

I will create a babydf.

```
babydf <- dplyr::select(raw, Centering, TradPed, SRPed, Valued)
```

Let's check the structure of the variables:

```
str(babydf)
```

```
## Classes 'data.table' and 'data.frame': 84 obs. of 4 variables:
## $ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ TradPed : num 3.8 5 4.8 4 4.2 3 5 4.6 4 4.8 ...
## $ SRPed   : num 4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
## $ Valued  : num 4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

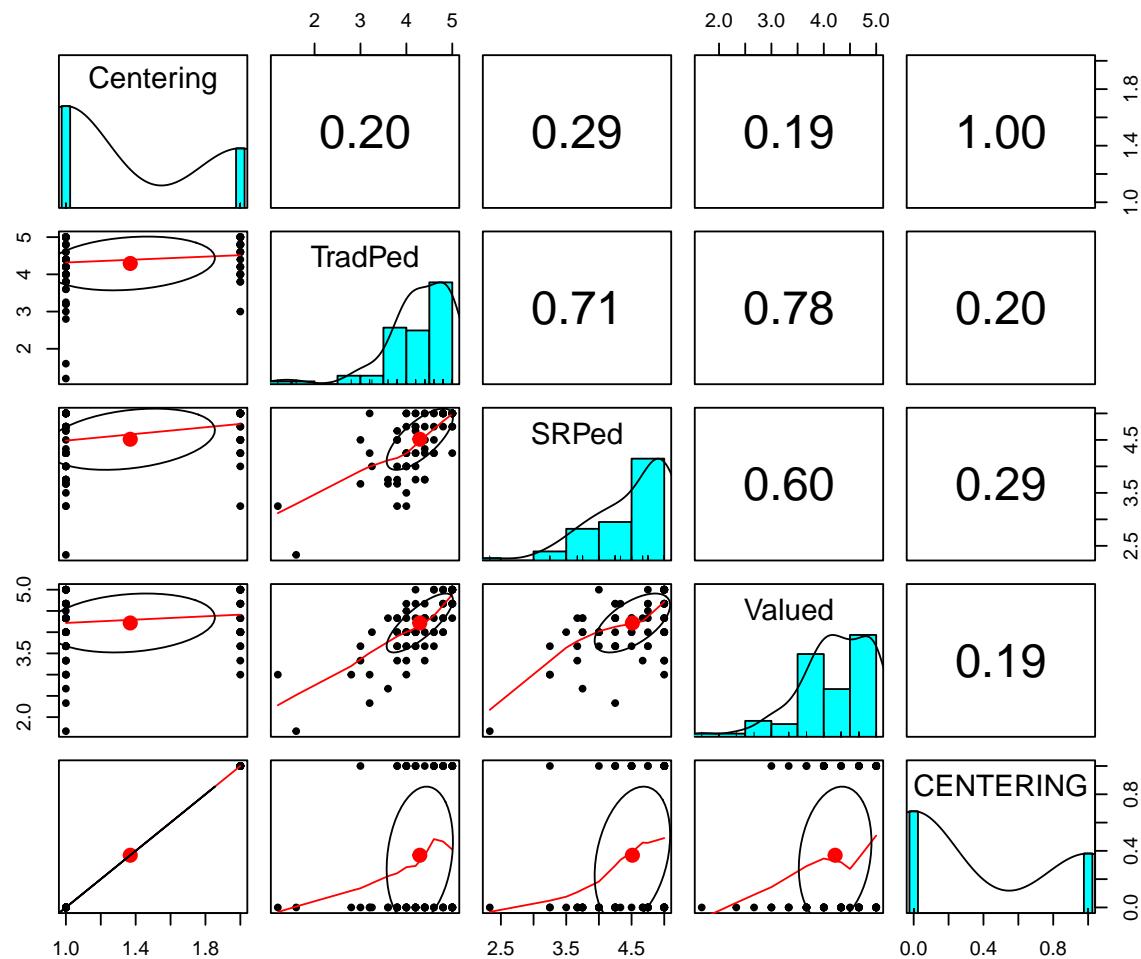
In later analyses, it will be important that Centering is a dummy-coded numerical variable:

```
babydf$CENTERING <- as.numeric(babydf$Centering)
babydf$CENTERING <- (babydf$CENTERING - 1)
str(babydf)
```

```
## Classes 'data.table' and 'data.frame': 84 obs. of 5 variables:
## $ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 2 ...
## $ TradPed   : num 3.8 5 4.8 4 4.2 3 5 4.6 4 4.8 ...
## $ SRPed     : num 4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
## $ Valued    : num 4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
## $ CENTERING: num 1 1 1 1 1 1 1 1 1 1 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

Quick peek at relations between variables:

```
psych::pairs.panels(babydf)
```



Using a piecewise approach, run each of the simple models in the grander design

Analysis #1: A simple moderation of the *a* path

We are asking, “Does traditional pedagogy moderate the relationship between centering and socially responsive pedagogy?

$Y = \text{socially responsive pedagogy}$ $X = \text{centering}$ $W = \text{traditional pedagogy}$

Let’s specify this simple moderation model with base R’s *lm()* function.

```
mod_a_path <- lm(SRPed ~ Centering * TradPed, data = babydf)
# the base R output if you prefer this view
summary(mod_a_path)
```

```
##
## Call:
## lm(formula = SRPed ~ Centering * TradPed, data = babydf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.30064 -0.24549  0.07396  0.15007  1.21341
##
## Coefficients:
##                               Estimate Std. Error t value     Pr(>|t|)
## (Intercept)             1.89621   0.30253   6.268 0.00000001948738 ***
## CenteringRe            1.15285   0.73720   1.564      0.122
## TradPed                0.59074   0.07064   8.362 0.00000000000204 ***
## CenteringRe:TradPed -0.21535   0.16486  -1.306      0.195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.399 on 77 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.5413, Adjusted R-squared:  0.5235
## F-statistic: 30.29 on 3 and 77 DF,  p-value: 0.000000000004875
```

We’ll use the *jtools* package so we get that great *summ* function and *interactions* for the awesome plot.

Since we are just working to understand our moderations, we can run them with “regular old” ordinary least squares.

```
# library(jtools) #the summ function creates a terrific regression
# table library(interactions)
library(ggplot2)

jtools::summ(mod_a_path, digits = 3)
```

Observations	81 (3 missing obs. deleted)
Dependent variable	SRPed
Type	OLS linear regression

F(3,77)	30.294
R ²	0.541
Adj. R ²	0.523

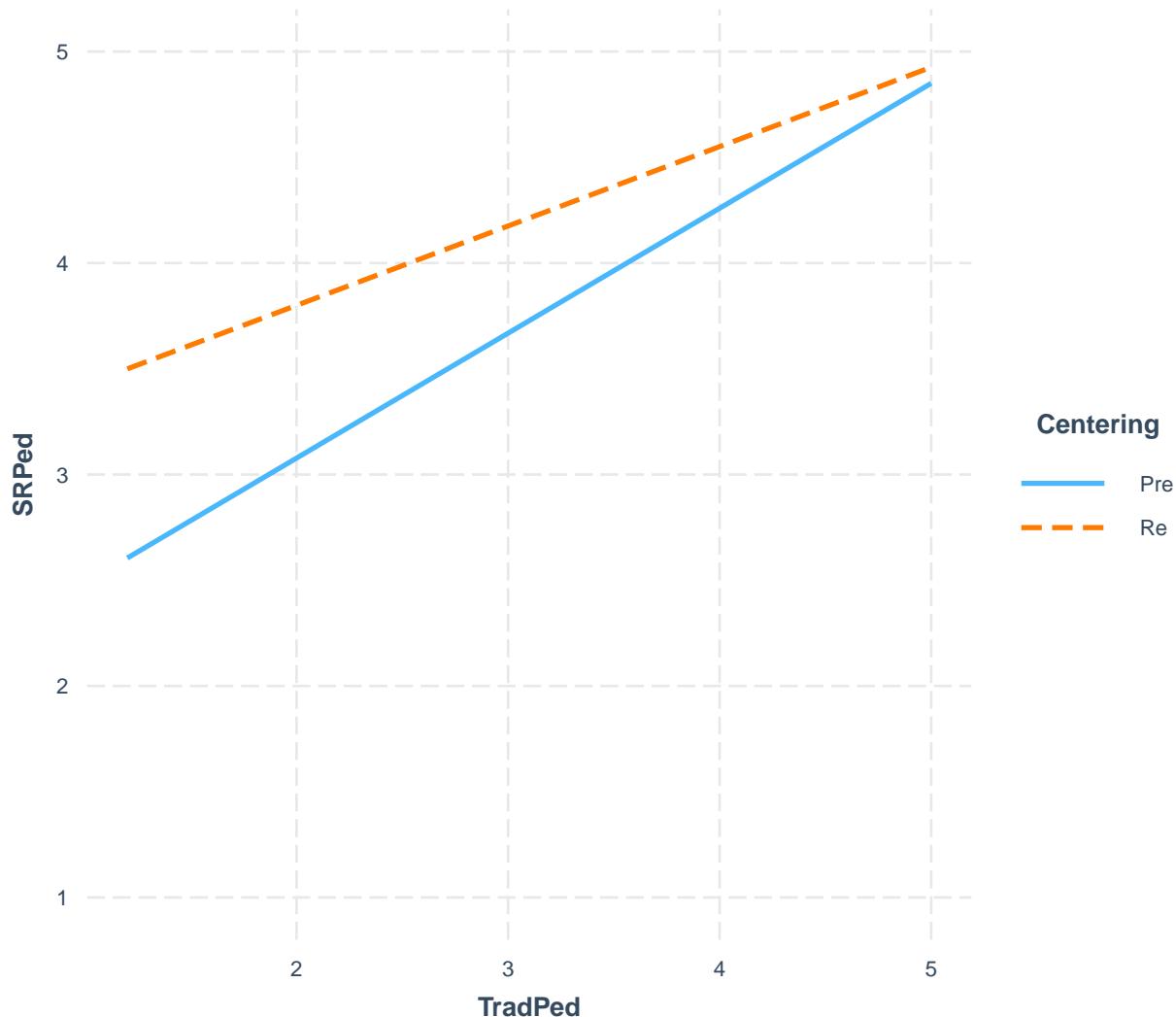
	Est.	S.E.	t val.	p
(Intercept)	1.896	0.303	6.268	0.000
CenteringRe	1.153	0.737	1.564	0.122
TradPed	0.591	0.071	8.362	0.000
CenteringRe:TradPed	-0.215	0.165	-1.306	0.195

Standard errors: OLS

Looking at these results we can see that the predictors account for about 54% of variance in perceived value to the student. Only the moderator (TradPed, W), traditional pedagogy is a significant predictor. Neither the independent variable (Centering, X), nor its interaction with Centering (Centering:TradPed, XW) are significant.

It's always helpful to graph the relationship. The *interaction_plot()* function from the package, *interactions* can make helpful illustrations. In the case of interactions/moderations, I like to run them "both ways" to see which makes more sense.

```
interactions::interaction_plot(mod_a_path, pred = TradPed, modx = Centering) +
  ylim(1, 5)
```



```
interactions::sim_slopes(mod_a_path, pred = TradPed, modx = Centering)
```

```
## Warning: Johnson-Neyman intervals are not available for factor moderators.
```

```
## SIMPLE SLOPES ANALYSIS
##
## Slope of TradPed when Centering = Re:
##
##   Est.    S.E.    t val.      p
##   ----- -----
##   0.38    0.15     2.52    0.01
##
## Slope of TradPed when Centering = Pre:
##
##   Est.    S.E.    t val.      p
##   ----- -----
##   0.59    0.07     8.36    0.00
```

```
# sim_slopes(Mod_a_path, pred=GRIcntlty, modx = GRMS) #sometimes I
# like to look at it in reverse -- like in the plots
```

Consistent with the non-significant interaction effect but the significant main effect, there was a statistically significant effect of traditional pedagogy on socially responsive pedagogy for both pre-centered and re-centered stages.

Traditional pedagogy is the only significant predictor in socially responsive pedagogy. Overall, the model accounts for 54% of the variance in socially responsive pedagogy.

Analysis #2: A simple mediation

We are asking, “Does socially responsive pedagogy mediate the relationship between centering and perceived value to the student?”

$Y = \text{perceived value}$ $X = \text{centering}$ $M = \text{socially responsive pedagogy}$

Note. I switched to using the CENTERING (all caps) variable because it is 0/1, numeric (better for lavaan).

```
library(lavaan)

medmodel <- "
    Valued ~ b*SRPed + c_p*CENTERING
    SRPed ~a*CENTERING

    #intercepts
    CENTERING ~ CENTERING.mean*1
    Valued ~ Valued.mean*1

    indirect := a*b
    direct := c_p
    total_c := c_p + (a*b)
    "

set.seed(230925) #required for reproducible results because lavaan introduces randomness into
medmodel_fit <- lavaan::sem(medmodel, data = babydf, se = "bootstrap",
    missing = "fiml")
medmodel_Sum <- lavaan::summary(medmodel_fit, standardized = T, rsq = T,
    ci = TRUE)
medmodel_ParEsts <- lavaan::parameterEstimates(medmodel_fit, boot.ci.type = "bca.simple",
    standardized = TRUE)
medmodel_Sum

## lavaan 0.6.16 ended normally after 25 iterations
##
##   Estimator                               ML
##   Optimization method                    NLMINB
##   Number of model parameters             9
```

```

## 
##   Number of observations                      84
##   Number of missing patterns                  2
##
## Model Test User Model:
##
##   Test statistic                           0.000
##   Degrees of freedom                       0
##
## Parameter Estimates:
##
##   Standard errors                         Bootstrap
##   Number of requested bootstrap draws      1000
##   Number of successful bootstrap draws     1000
##
## Regressions:
##             Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   Valued ~
##     SRPed      (b)  0.728   0.124   5.877   0.000   0.455   0.933
##     CENTERIN (c_p) 0.004   0.124   0.032   0.974  -0.225   0.257
##   SRPed ~
##     CENTERIN (a)  0.367   0.114   3.225   0.001   0.148   0.601
##   Std.lv  Std.all
##
##     0.728   0.608
##     0.004   0.003
##
##     0.367   0.307
##
## Intercepts:
##             Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   CENTERI (CENT)  0.369   0.054   6.872   0.000   0.274   0.476
##   .Valued (Vld.)  0.935   0.548   1.707   0.088   0.018   2.136
##   .SRPed          4.371   0.086  50.597   0.000   4.187   4.527
##   Std.lv  Std.all
##
##     0.369   0.765
##     0.935   1.355
##     4.371   7.580
##
## Variances:
##             Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   .Valued          0.299   0.055   5.453   0.000   0.186   0.408
##   .SRPed          0.301   0.058   5.157   0.000   0.196   0.413
##   CENTERING        0.233   0.014  16.390   0.000   0.199   0.249
##   Std.lv  Std.all
##
##     0.299   0.629
##     0.301   0.906
##     0.233   1.000

```

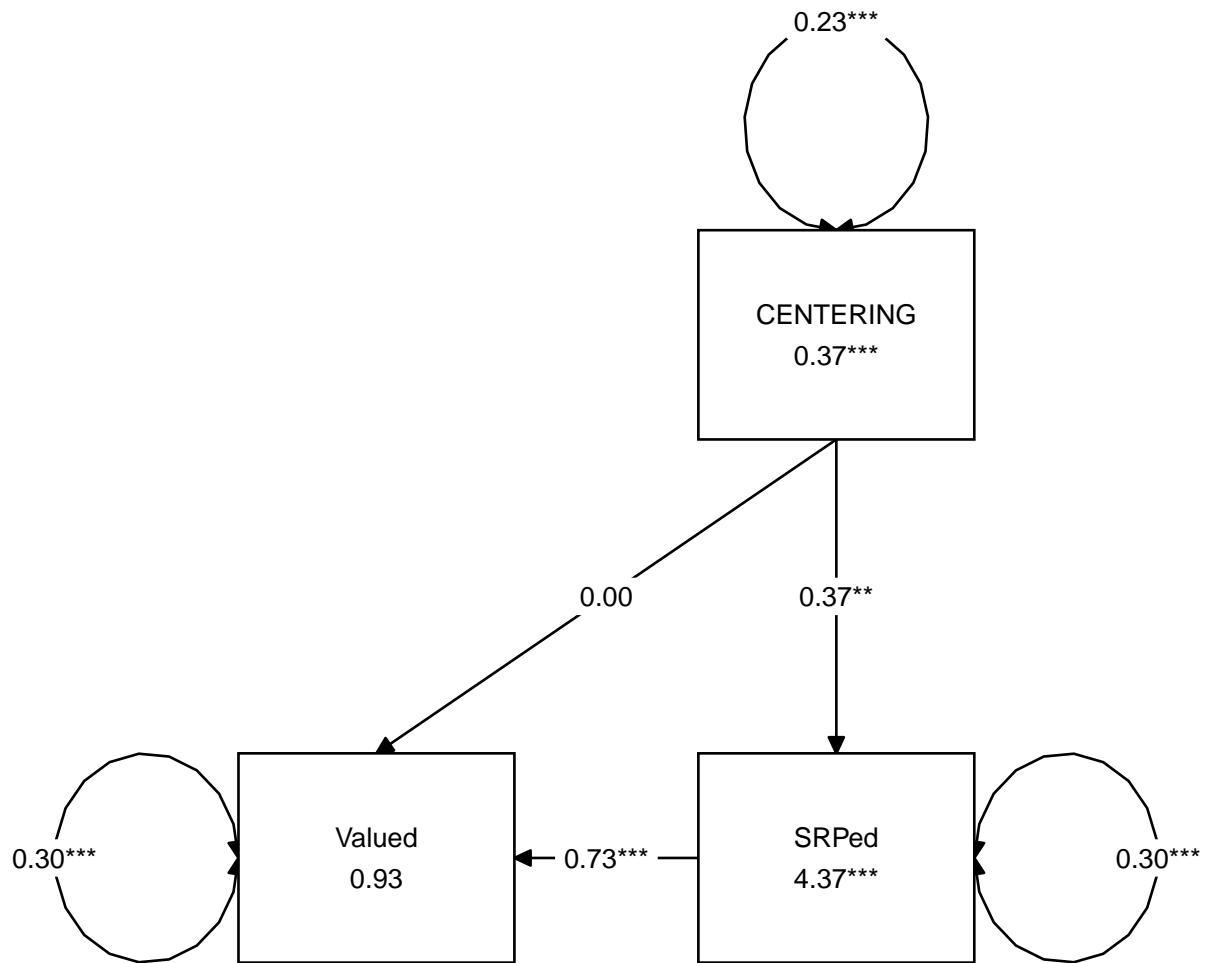
```
##
## R-Square:
##             Estimate
## Valued      0.371
## SRPed      0.094
##
## Defined Parameters:
##             Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## indirect    0.267   0.100  2.665   0.008   0.095   0.488
## direct     0.004   0.124  0.032   0.974  -0.225   0.257
## total_c    0.271   0.144  1.886   0.059  -0.014   0.551
## Std.lv Std.all
## 0.267   0.187
## 0.004   0.003
## 0.271   0.190
```

medmodel_ParEsts

	lhs	op	rhs	label	est	se	z	pvalue	ci.lower
## 1	Valued	~	SRPed	b	0.728	0.124	5.877	0.000	0.464
## 2	Valued	~	CENTERING	c_p	0.004	0.124	0.032	0.974	-0.225
## 3	SRPed	~	CENTERING	a	0.367	0.114	3.225	0.001	0.131
## 4	CENTERING	~1		CENTERING.mean	0.369	0.054	6.872	0.000	0.262
## 5	Valued	~1		Valued.mean	0.935	0.548	1.707	0.088	-0.040
## 6	Valued	~~	Valued		0.299	0.055	5.453	0.000	0.213
## 7	SRPed	~~	SRPed		0.301	0.058	5.157	0.000	0.205
## 8	CENTERING	~~	CENTERING		0.233	0.014	16.390	0.000	0.199
## 9	SRPed	~1			4.371	0.086	50.597	0.000	4.189
## 10	indirect	:=	a*b	indirect	0.267	0.100	2.665	0.008	0.109
## 11	direct	:=	c_p	direct	0.004	0.124	0.032	0.974	-0.225
## 12	total_c	:=	c_p+(a*b)	total_c	0.271	0.144	1.886	0.059	-0.025
				ci.upper	std.lv	std.all	std.nox		
## 1	0.939	0.728	0.608	0.608					
## 2	0.250	0.004	0.003	0.003					
## 3	0.587	0.367	0.307	0.307					
## 4	0.476	0.369	0.765	0.765					
## 5	2.084	0.935	1.355	1.355					
## 6	0.454	0.299	0.629	0.629					
## 7	0.424	0.301	0.906	0.906					
## 8	0.249	0.233	1.000	1.000					
## 9	4.528	4.371	7.580	7.580					
## 10	0.528	0.267	0.187	0.187					
## 11	0.250	0.004	0.003	0.003					
## 12	0.546	0.271	0.190	0.190					

Our model accounts for 9% of the variance in socially responsive pedagogy and 37% of the variance in perceived value to the student. The *a* path (Centering → SRPed), *b* path (SRPed → Valued), and indirect effect are all statistically significant.

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = medmodel_fit)
```



We can use the `tidySEM::get_layout` function to understand how our model is being mapped.

```
tidySEM::get_layout(medmodel_fit)
```

```
##      [,1]      [,2]
## [1,] "CENTERING" NA
## [2,] "SRPed"     "Valued"
## attr(,"class")
## [1] "layout_matrix" "matrix"       "array"
```

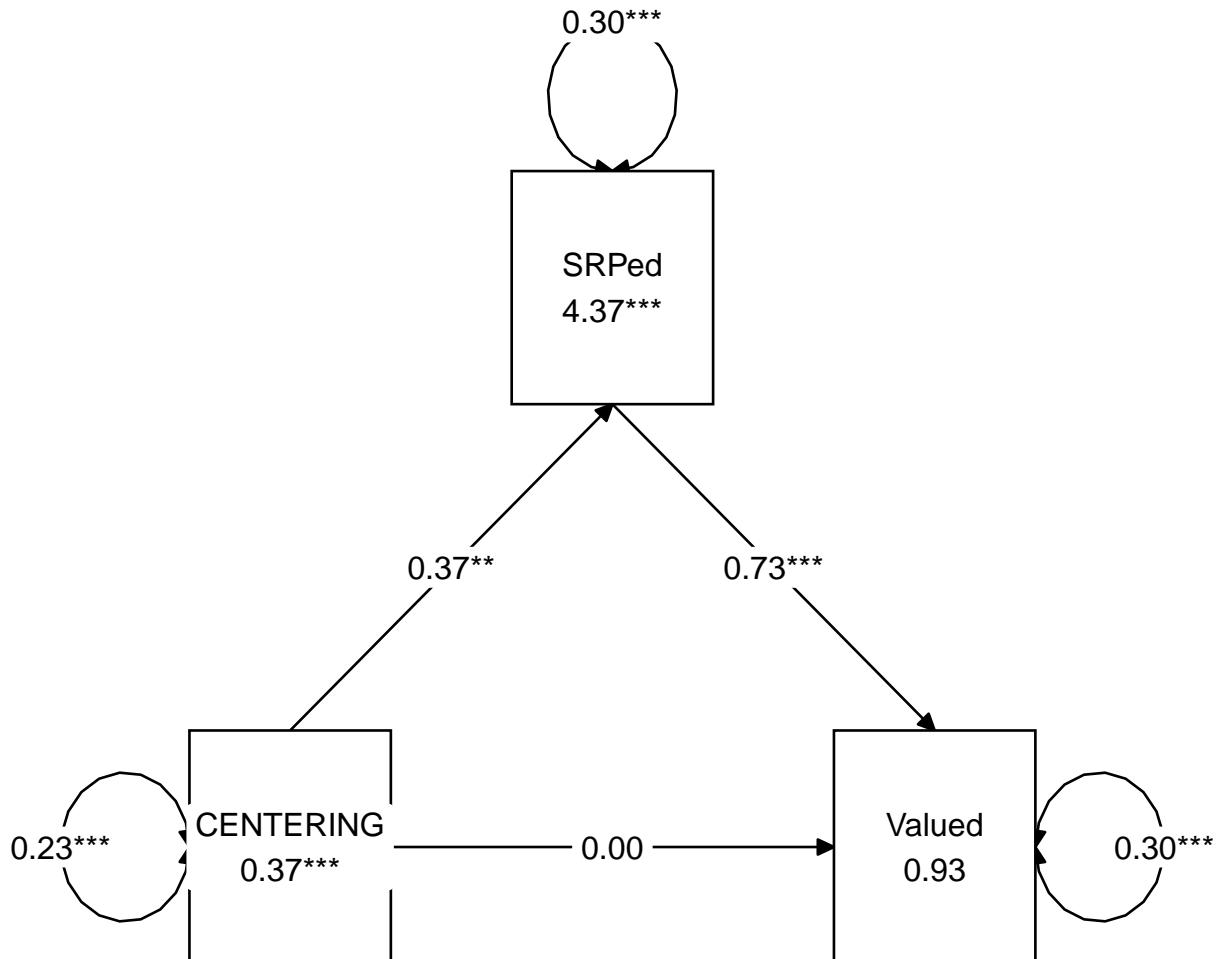
We can write code to remap them

```
medmap <- tidySEM::get_layout("", "SRPed", "", "CENTERING", "", "Valued",
  rows = 2)
medmap

##      [,1]      [,2]      [,3]
## [1,] ""        "SRPed"   ""
## [2,] "CENTERING"   ""        "Valued"
## attr(,"class")
## [1] "layout_matrix" "matrix"    "array"
```

We can update the *tidySEM::graph_sem* function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(medmodel_fit, layout = medmap, rect_width = 1.25, rect_height = 1.25,
  spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



```
### Specify and run the entire lavaan model {-}
```

```
set.seed(230925)
ModMedOnA <- "
  #equations
  SRPed ~ a1*CENTERING + a2*TradPed + a3*CENTERING:TradPed
  Valued ~ c_p*CENTERING + b*SRPed

  #intercepts
  SRPed ~ SRPed.mean*1
  Valued ~ Valued.mean*1

  #means, variances of W for simple slopes
  TradPed ~ TradPed.mean*1
  TradPed ~~ TradPed.var*TradPed

  #index of moderated mediation, there will be an a and b path in the product
  #if the a and/or b path is moderated, select the label that represents the moderation

  imm := a3*b

  #Note that we first create the indirect product, then add to it the product of the imm and
  indirect.SDbelow := a1*b + imm*(TradPed.mean - sqrt(TradPed.var))
  indirect.mean := a1*b + imm*(TradPed.mean)
  indirect.SDabove := a1*b + imm*(TradPed.mean + sqrt(TradPed.var))

  "
set.seed(230925)  #required for reproducible results because lavaan introduces randomness into
ModMedOnA_fit <- lavaan::sem(ModMedOnA, data = babydf, se = "bootstrap",
  missing = "fiml", bootstrap = 1000)

## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: 1
##           exogenous variable(s), while fixed.x = TRUE.

## Warning in lav_partable_vnames(FLAT, "ov.x", warn = TRUE): lavaan WARNING:
##           model syntax contains variance/covariance/intercept formulas
##           involving (an) exogenous variable(s): [TradPed]; These variables
##           will now be treated as random introducing additional free
##           parameters. If you wish to treat those variables as fixed, remove
##           these formulas from the model syntax. Otherwise, consider adding
##           the fixed.x = FALSE option.

ModMedOnAsum <- lavaan::summary(ModMedOnA_fit, standardized = TRUE, rsq = T,
  ci = TRUE)
ModMedOnAParamEsts <- lavaan::parameterEstimates(ModMedOnA_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
ModMedOnAsum
```

```

## lavaan 0.6.16 ended normally after 36 iterations
##
##   Estimator                               ML
## Optimization method                      NLMINB
## Number of model parameters                11
##
##                                         Used     Total
## Number of observations                   83      84
## Number of missing patterns               2
##
## Model Test User Model:
##
##   Test statistic                          60.195
##   Degrees of freedom                     4
##   P-value (Chi-square)                  0.000
##
## Parameter Estimates:
##
##   Standard errors                         Bootstrap
##   Number of requested bootstrap draws    1000
##   Number of successful bootstrap draws   1000
##
## Regressions:
##             Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   SRPed ~
##     CENTERIN (a1)  1.184   0.915   1.295   0.195  -1.022   2.719
##     TradPed   (a2)  0.597   0.095   6.304   0.000   0.451   0.819
##     CENTERIN (a3) -0.222   0.194  -1.143   0.253  -0.543   0.246
##   Valued ~
##     CENTERIN (c_p) -0.011   0.122  -0.094   0.925  -0.234   0.243
##     SRPed   (b)    0.737   0.119   6.189   0.000   0.481   0.939
##   Std.lv  Std.all
##
##     1.184   0.965
##     0.597   0.728
##    -0.222  -0.819
##     -
##     -0.011  -0.008
##     0.737   0.625
##
## Intercepts:
##             Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   .SRPed  (SRP.)  1.865   0.433   4.308   0.000   0.826   2.538
##   .Valued  (Vld.)  0.900   0.527   1.708   0.088  -0.011   2.024
##   TradPed (TrP.)  4.292   0.078  55.116   0.000   4.130   4.443
##   Std.lv  Std.all
##     1.865   3.164
##     0.900   1.294

```

```

##      4.292    5.977
##
## Variances:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## TradPed (TrP.)        0.516   0.142   3.627   0.000   0.263   0.839
## .SRPed                 0.151   0.029   5.119   0.000   0.089   0.206
## .Valued                0.296   0.054   5.489   0.000   0.182   0.399
## Std.lv    Std.all
## 0.516     1.000
## 0.151     0.434
## 0.296     0.611
##
## R-Square:
##                               Estimate
## SRPed                  0.566
## Valued                 0.389
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## imm                   -0.164   0.144  -1.136   0.256  -0.416   0.185
## indirect.SDblw       0.288   0.177   1.625   0.104  -0.107   0.602
## indirect.mean         0.171   0.085   2.002   0.045   0.000   0.337
## indirect.SDabv       0.053   0.073   0.730   0.465  -0.093   0.190
## Std.lv    Std.all
## -0.164    -0.512
## 0.288    -1.944
## 0.171    -2.456
## 0.053    -2.967

```

ModMedOnAParamEsts

```

##          lhs op                                rhs
## 1        SRPed ~                                CENTERING
## 2        SRPed ~                                TradPed
## 3        SRPed ~                                CENTERING:TradPed
## 4        Valued ~                                CENTERING
## 5        Valued ~                                SRPed
## 6        SRPed ~1
## 7        Valued ~1
## 8        TradPed ~1
## 9        TradPed ~~                                TradPed
## 10       SRPed ~~                                SRPed
## 11       Valued ~~                                Valued
## 12       CENTERING ~~                                CENTERING
## 13       CENTERING ~~                                CENTERING:TradPed
## 14 CENTERING:TradPed ~~                                CENTERING:TradPed
## 15       CENTERING ~1

```

```

## 16 CENTERING:TradPed ~1
## 17 imm := a3*b
## 18 indirect.SDbelow := a1*b+imm*(TradPed.mean-sqrt(TradPed.var))
## 19 indirect.mean := a1*b+imm*(TradPed.mean)
## 20 indirect.SDabove := a1*b+imm*(TradPed.mean+sqrt(TradPed.var))

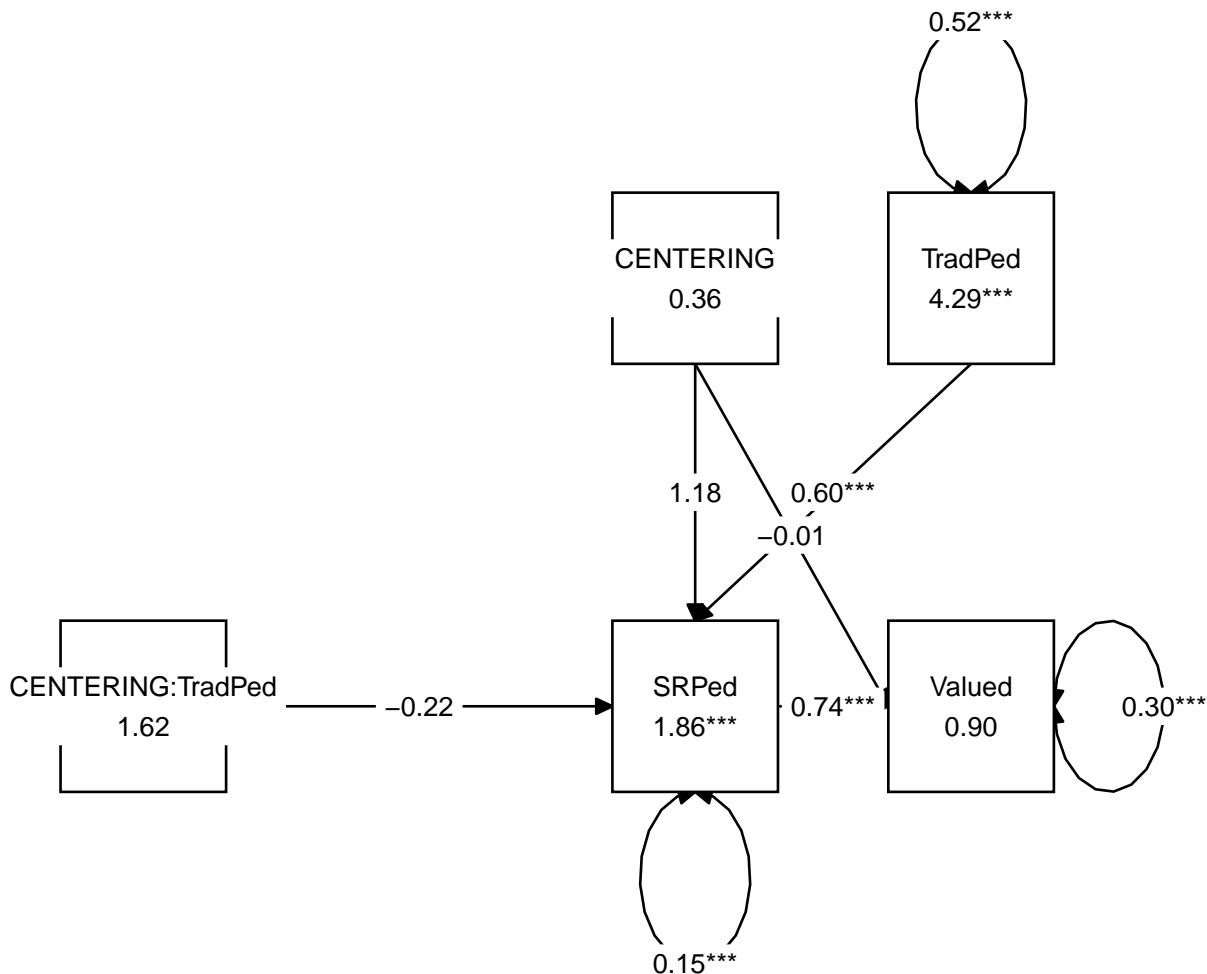
##          label   est    se     z pvalue ci.lower ci.upper std.lv std.all
## 1          a1  1.184 0.915  1.295  0.195 -1.175   2.630  1.184  0.965
## 2          a2  0.597 0.095  6.304  0.000  0.448   0.806  0.597  0.728
## 3          a3 -0.222 0.194 -1.143  0.253 -0.523   0.287 -0.222 -0.819
## 4          c_p -0.011 0.122 -0.094  0.925 -0.236   0.237 -0.011 -0.008
## 5          b   0.737 0.119  6.189  0.000  0.487   0.946  0.737  0.625
## 6          SRPed.mean 1.865 0.433  4.308  0.000  0.894   2.551  1.865  3.164
## 7          Valued.mean 0.900 0.527  1.708  0.088 -0.027   1.985  0.900  1.294
## 8          TradPed.mean 4.292 0.078 55.116  0.000  4.129   4.434  4.292  5.977
## 9          TradPed.var  0.516 0.142  3.627  0.000  0.272   0.850  0.516  1.000
## 10         0.151 0.029  5.119  0.000  0.103   0.219  0.151  0.434
## 11         0.296 0.054  5.489  0.000  0.203   0.425  0.296  0.611
## 12         0.231 0.000    NA    NA  0.231   0.231  0.231  1.000
## 13         1.036 0.000    NA    NA  1.036   1.036  1.036  0.991
## 14         4.733 0.000    NA    NA  4.733   4.733  4.733  1.000
## 15         0.361 0.000    NA    NA  0.361   0.361  0.361  0.752
## 16         1.622 0.000    NA    NA  1.622   1.622  1.622  0.745
## 17         imm -0.164 0.144 -1.136  0.256 -0.411   0.196 -0.164 -0.512
## 18 indirect.SDbelow 0.288 0.177  1.625  0.104 -0.096   0.631  0.288 -1.944
## 19 indirect.mean  0.171 0.085  2.002  0.045  0.013   0.361  0.171 -2.456
## 20 indirect.SDabove 0.053 0.073  0.730  0.465 -0.100   0.184  0.053 -2.967

##          std.nox
## 1          2.010
## 2          0.487
## 3         -0.377
## 4         -0.016
## 5          0.625
## 6          3.164
## 7          1.294
## 8          5.977
## 9          1.000
## 10         0.434
## 11         0.611
## 12         0.231
## 13         1.036
## 14         4.733
## 15         0.361
## 16         1.622
## 17        -0.235
## 18         0.084
## 19        -0.151
## 20        -0.386

```

Use tidySEM to create a figure

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = ModMedOnA_fit)
```



We can use the `tidySEM::get_layout` function to understand how our model is being mapped.

```
tidySEM::get_layout(ModMedOnA_fit)
```

```
##      [,1]      [,2]      [,3]      [,4]
## [1,] NA "CENTERING:TradPed" NA NA
## [2,] NA      NA          NA          NA
```

```
## [3,] "Valued" "SRPed"           "CENTERING" "TradPed"
## attr(,"class")
## [1] "layout_matrix" "matrix"      "array"
```

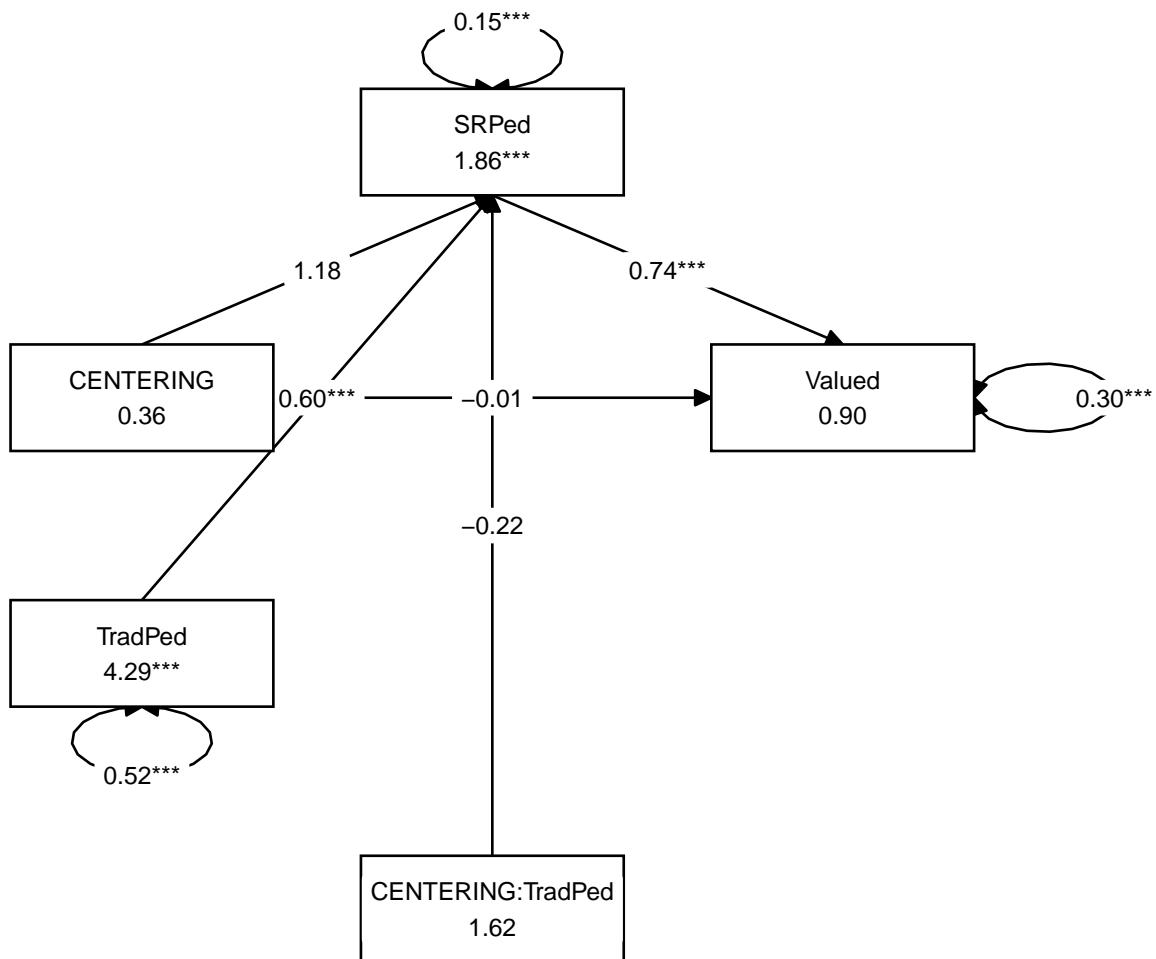
We can write code to remap them

```
comb_map <- tidySEM::get_layout("", "SRPed", "", "CENTERING", "", "Valued",
                                "TradPed", "", "", "", "CENTERING:TradPed", "", rows = 4)
comb_map
```

```
##      [,1]      [,2]      [,3]
## [1,] ""       "SRPed"    ""
## [2,] "CENTERING"  ""      "Valued"
## [3,] "TradPed"   ""      ""
## [4,] ""       "CENTERING:TradPed"  ""
## attr(,"class")
## [1] "layout_matrix" "matrix"      "array"
```

We can update the *tidySEM::graph_sem* function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(ModMedOnA_fit, layout = comb_map, rect_width = 1.5,
                    rect_height = 1.25, spacing_x = 2, spacing_y = 3, text_size = 3.5)
```



Create a table that includes regression output for the M and Y variables and the moderated effects

Exporting the results to a .csv file will help us create a journal-ready table, outside of the R environment.

```
write.csv(ModMedOnAParamEsts, file = "ModMedOnAParamEsts.csv") #optional to write it to a .csv
```

Table 1

Analysis of Moderated Mediation for Centering, Socially Responsive Pedagogy, Traditional Pedagogy, and Perceived Value to the Student

Predictor	B	SE _B	p	R ²
Socially Responsive Pedagogy(M)				.57
Constant	1.865	0.433	<0.001	
Centering (a_1)	1.184	0.915	0.195	
TradPed (a_2)	0.597	0.095	<0.001	
Centering:TradPed (a_3)	-0.222	0.194	0.253	
Perceived Value (DV)				.39
Constant	0.900	0.527	0.088	
Centering (c')	-0.011	0.122	0.925	
SRPed (b)	0.737	0.119	<0.001	
Summary of Effects	B	SE _B	p	95% CI
IMM	-0.164	0.144	0.256	-0.411, 0.196
Indirect (-1SD)	0.288	0.177	0.104	-0.096, 0.631
Indirect (M)	0.171	0.085	0.045	0.013, 0.361
Indirect (+1SD)	0.053	0.073	0.465	-0.100 0.184

Note. SRPed = socially responsive pedagogy; TradPed = traditional pedagogy. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

APA Style Write-up

Method/Aalytic Strategy

Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16). We specified a moderated mediation model predicting perceived value to the student from centering stage, mediated by socially responsive pedagogy. We further predicted that the relationship between centering to socially responsive pedagogy (i.e., the a path) would be moderated by traditional pedagogy.

Results

Preliminary Analyses

- Missing data analysis and managing missing data
- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

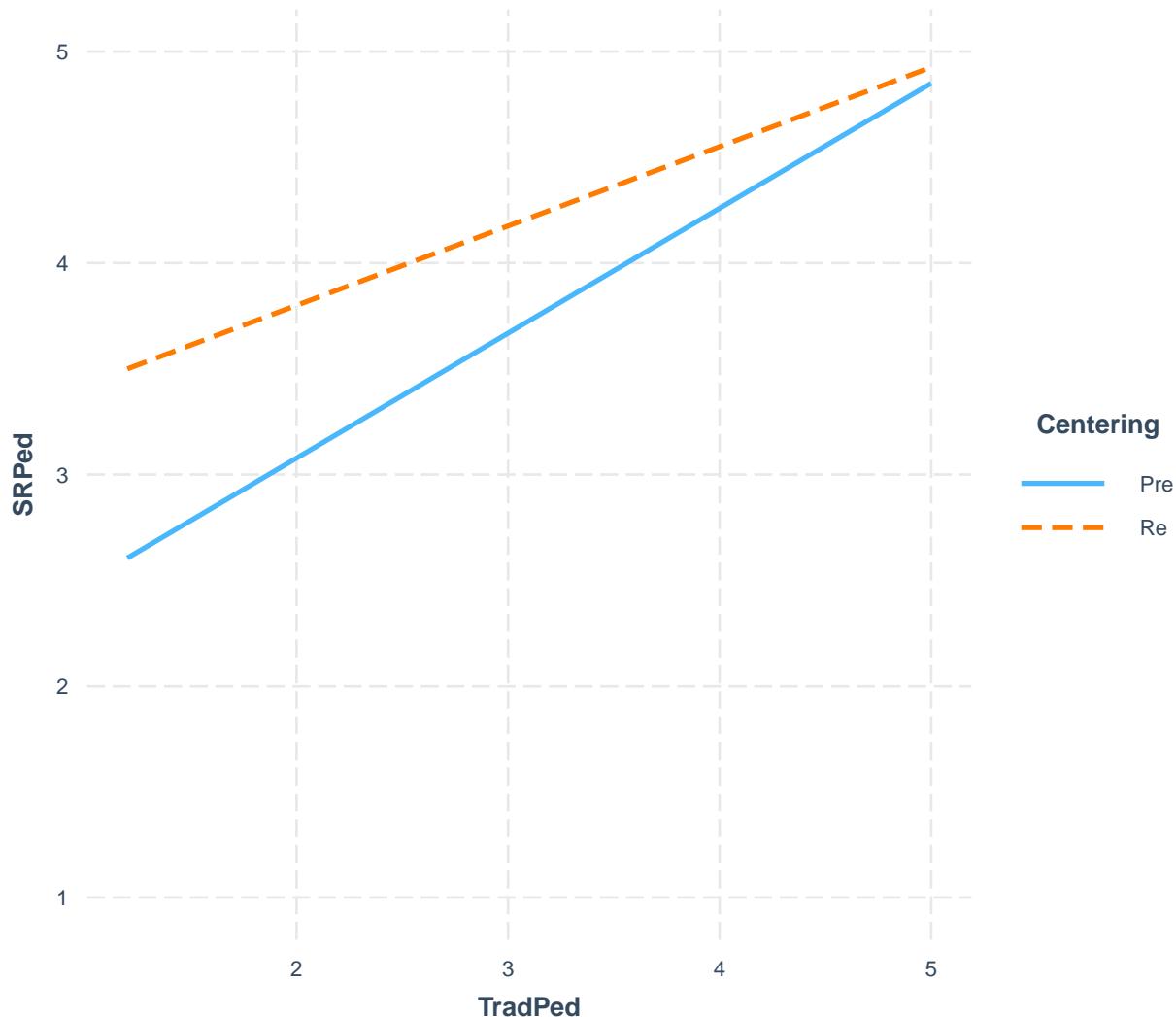
Primary Analyses Our analysis evaluated a moderation mediation model perceived value to the student (Y/Valued) from centering stage ($X/\text{Centering}$) mediated by socially responsive pedagogy (M/SRPed). Traditional pedagogy ($W/\text{TradPed}$) was our moderating variable. We specified a moderation of path a ($X/\text{Centering}$ to M/SRPed). Data were analyzed with maximum likelihood estimation in the R package *lavaan* (v. 0.6-16); the significance of effects were tested with 1000 bootstrap confidence intervals. Results of the full model are presented in Table 1 and illustrated in Figure 1. The formula for the mediator and dependent variable are expressed below.

$$\hat{M} = 1.865 + (1.184)X + (0.597)W + (-0.222)XW$$

$$\hat{Y} = 0.900 + (-0.011)X + (0.060)M + (0.737)X$$

Regarding the presence of a moderated mediation, results were mixed. On the one hand, the index of moderated mediation was not statistically significant ($IMM = -0.164, p = 0.256$). In contrast, the conditional indirect effect at the mean of the moderator was statistically significant ($B = 0.171, p = 0.045$), while the conditional indirect effects at $\pm 1SD$ were not. A possible explanation for the lack of statistical significance is the relative lower power ($N = 84$) for analysis of a relatively complicated model. The model accounted for 57% of the variance in socially responsive pedagogy and 37% of the variance in perceived value to the student. Figure 1 illustrates the conditional effects of traditional pedagogy on socially responsive pedagogy at pre- and re-centered stages.

```
interactions::interact_plot(mod_a_path, pred = TradPed, modx = Centering) +
  ylim(1, 5)
```



8.11 Homeworked Example 2: A moderation on the *b* path

[Screencast Link](#)

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to conduct a moderated mediation. At least one path (a or b) should be moderated.

Describing they overall model hypothesis, assign each variable to the X, Y, M, and W roles

My analysis will evaluated a moderated mediation. Specifically, I predict that the effect of centering on perceived value to the student will be moderated by the students' evaluation of socially responsive

pedagogy. I further hypothesize that this indirect effect will be moderated by traditional pedagogy and that the moderation will occur on the b path, that is, traditional pedagogy will moderate the relationship between socially responsive pedagogy and perceived value to the student.

- X = Centering, pre/re (0,1)
- W = Traditional Pedagogy (1 to 4 scaling)
- M = Socially Responsive Pedagogy (1 to 4 scaling)
- Y = Value to the student (1 to 4 scaling)

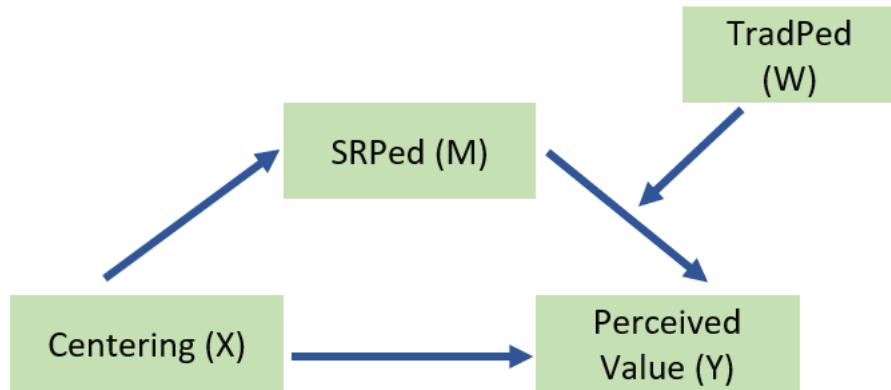


Figure 8.10: An image of the conceptual model of moderated mediation for the homeworked example.

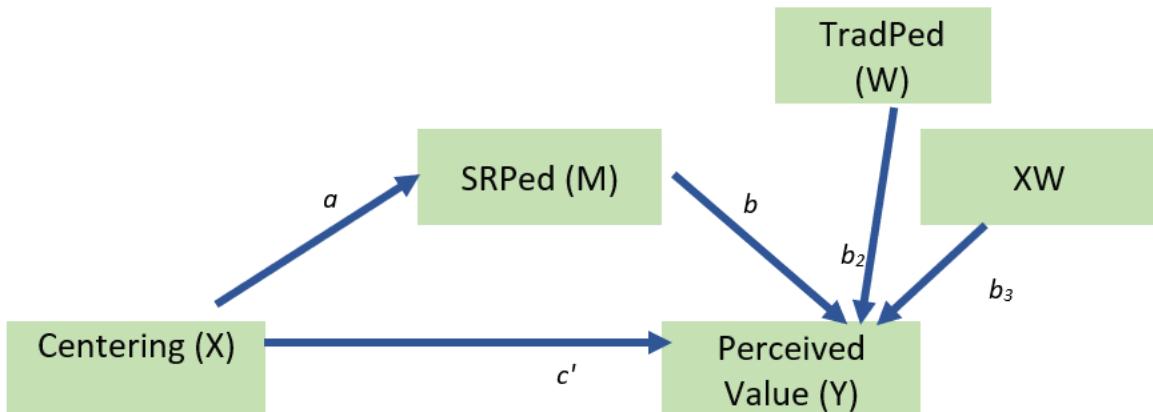


Figure 8.11: An image of the statistical model of simple moderation for the homeworked example.

Import the data and format the variables in the model

```
raw <- readRDS("ReC.rds")
```

The approach we are taking to moderated mediation does not allow dependency in the data. Therefore, we will include only those who took the multivariate class (i.e., excluding responses for the ANOVA and psychometrics courses).

```
raw <- (dplyr::filter(raw, Course == "Multivariate"))
```

I need to score the SRPed, TradPed, and Valued variables

```
TradPed_vars <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback",
  "ClearOrganization", "ClearPresentation")
raw$TradPed <- sjstats::mean_n(raw[, ..TradPed_vars], 0.75)

Valued_vars <- c("ValObjectives", "IncrUnderstanding", "IncrInterest")
raw$Valued <- sjstats::mean_n(raw[, ..Valued_vars], 0.75)

SRPed_vars <- c("InclusvClassrm", "EquitableEval", "MultPerspectives",
  "DEIntegration")
raw$SRPed <- sjstats::mean_n(raw[, ..SRPed_vars], 0.75)
```

I will create a babydf.

```
babydf <- dplyr::select(raw, Centering, TradPed, SRPed, Valued)
```

Let's check the structure of the variables:

```
str(babydf)
```

```
## Classes 'data.table' and 'data.frame':  84 obs. of  4 variables:
## $ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ TradPed  : num  3.8 5 4.8 4 4.2 3 5 4.6 4 4.8 ...
## $ SRPed    : num  4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
## $ Valued   : num  4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

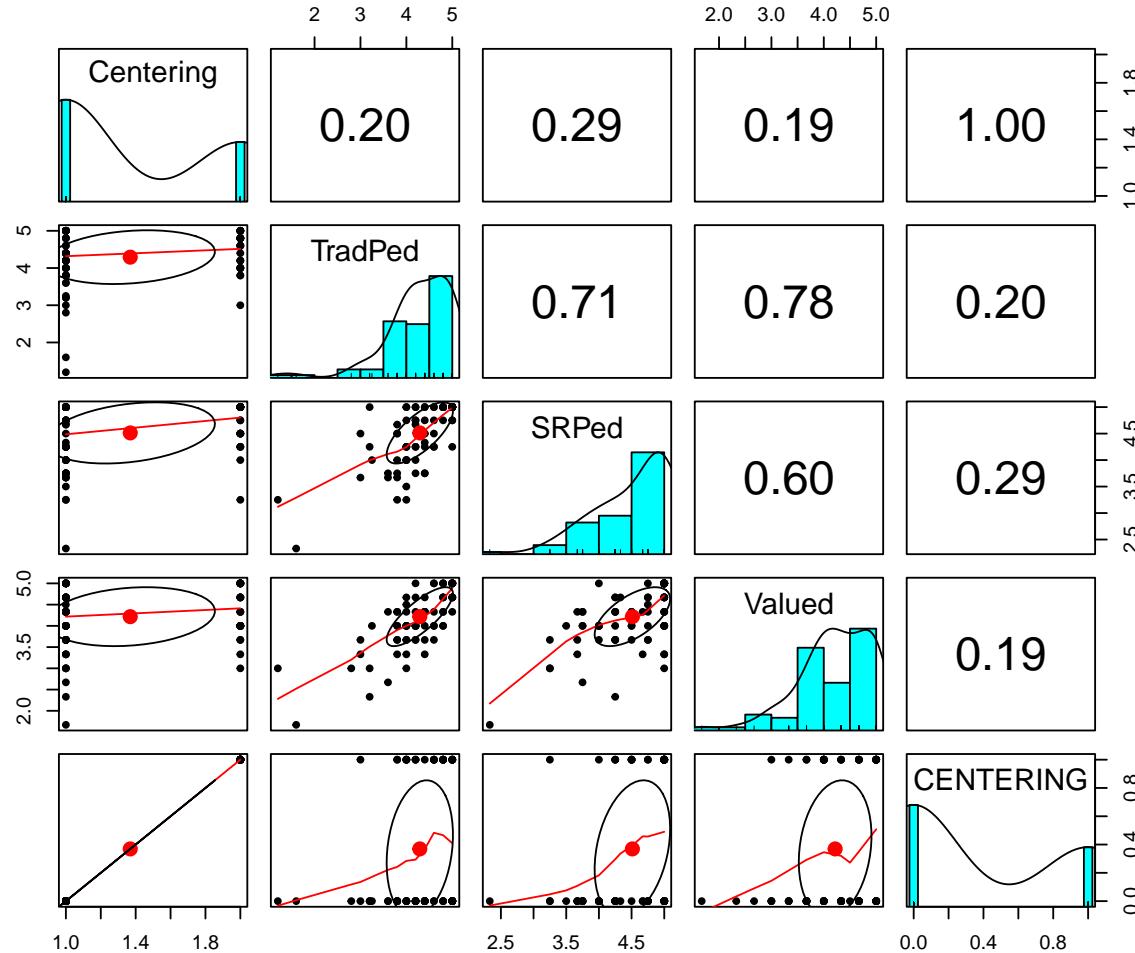
In later analyses, it will be important that Centering is a dummy-coded numerical variable:

```
babydf$CENTERING <- as.numeric(babydf$Centering)
babydf$CENTERING <- (babydf$CENTERING - 1)
str(babydf)
```

```
## Classes 'data.table' and 'data.frame':  84 obs. of  5 variables:
## $ Centering: Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ TradPed  : num  3.8 5 4.8 4 4.2 3 5 4.6 4 4.8 ...
## $ SRPed    : num  4.5 5 5 5 4.75 4.5 5 4.5 5 5 ...
## $ Valued   : num  4.33 5 4.67 3.33 4 3.67 5 4 4.67 4.67 ...
## $ CENTERING: num  1 1 1 1 1 1 1 1 1 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

Quick peek at relations between variables:

```
psych::pairs.panels(babydf)
```



Using a piecewise approach, run each of the simple models in the grander design

Analysis #1: A simple moderation

We are asking, “Does traditional pedagogy moderate the relationship between socially responsive pedagogy and perceived value to the student?

$Y = \text{perceived value to the student}$ $X = \text{socially responsive pedagogy}$ $W = \text{traditional pedagogy}$

Let’s specify this simple moderation model with base R’s `lm()` function.

```
Mod_b_path <- lm(Valued ~ SRPed * TradPed, data = babydf)
# the base R output if you prefer this view
summary(Mod_b_path)
```

```

## 
## Call:
## lm(formula = Valued ~ SRPed * TradPed, data = babydf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.06146 -0.26683  0.08376  0.19957  0.94852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.97179   1.29638   1.521   0.132
## SRPed        -0.19009   0.33922  -0.560   0.577
## TradPed      0.35774   0.32949   1.086   0.281
## SRPed:TradPed 0.07962   0.07850   1.014   0.314
##
## Residual standard error: 0.4393 on 77 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.6114, Adjusted R-squared:  0.5963
## F-statistic: 40.39 on 3 and 77 DF,  p-value: 0.0000000000000008713

```

We'll use the *jtools* package so we get that great *summ* function and *interactions* for the awesome plot.

Since we are just working to understand our moderation, we can run them with “regular old” ordinary least squares.

```

# library(jtools) #the summ function creates a terrific regression
# table library(interactions)
library(ggplot2)

jtools::summ(Mod_b_path, digits = 3)

```

Observations	81 (3 missing obs. deleted)
Dependent variable	Valued
Type	OLS linear regression
<hr/>	
F(3,77)	40.389
R ²	0.611
Adj. R ²	0.596

Looking at these results we can see that the predictors account for about 61% of variance in disengagement coping. Only the moderator (W), traditional pedagogy is a significant predictor. Neither the independent variable (Centering, [X]), nor its interaction with Centering (Centering:TradPed, [XW]) are significant.

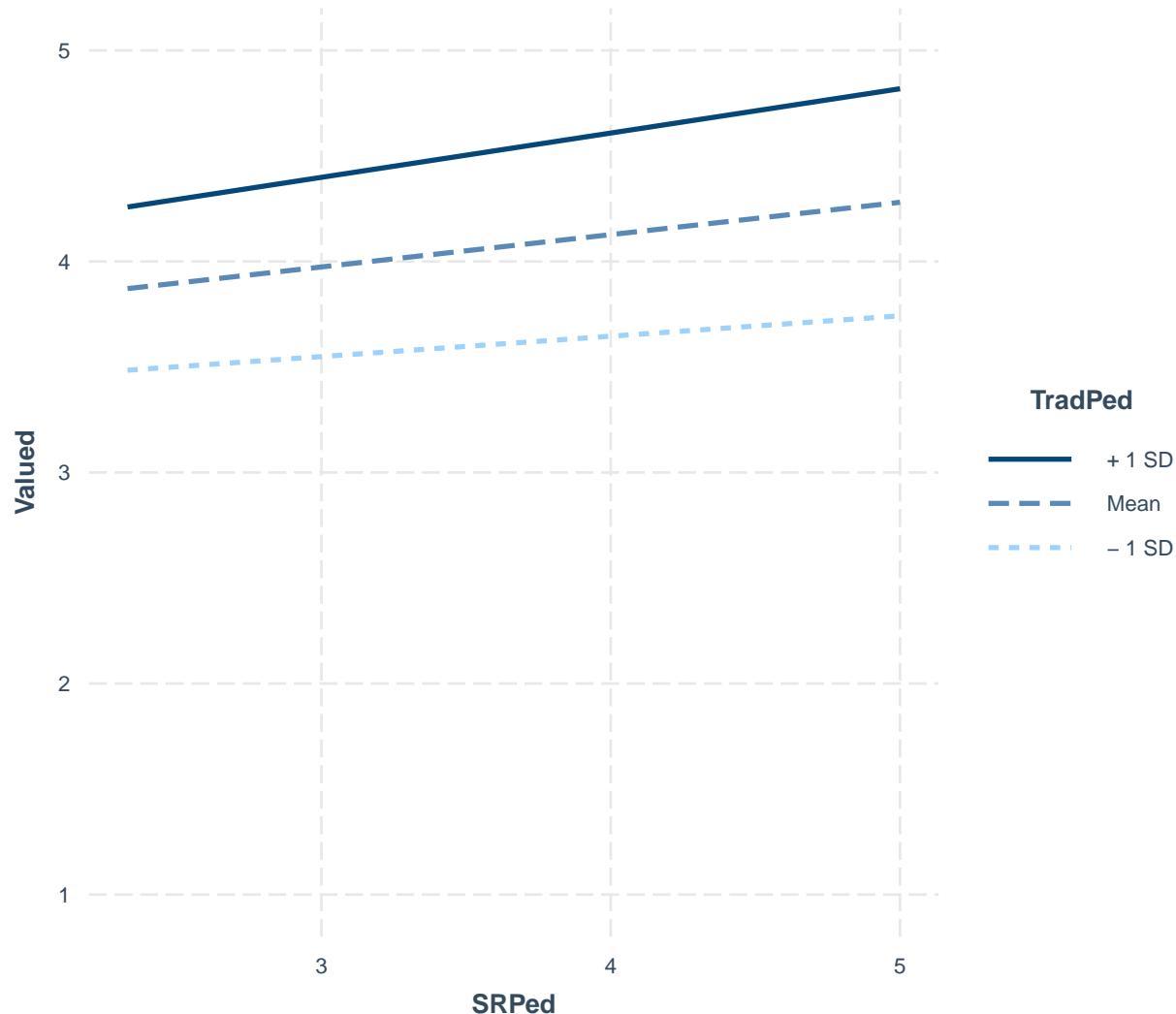
It's always helpful to graph the relationship. The *interaction_plot()* function from the package, *interactions* can make helpful illustrations. This plot represents the model we specified with each of the variables in their respective roles.

	Est.	S.E.	t val.	p
(Intercept)	1.972	1.296	1.521	0.132
SRPed	-0.190	0.339	-0.560	0.577
TradPed	0.358	0.329	1.086	0.281
SRPed:TradPed	0.080	0.078	1.014	0.314

Standard errors: OLS

```
interactions::interact_plot(Mod_b_path, pred = SRPed, modx = TradPed) +
  ylim(1, 5)
```

Warning: 5.02351626747081 is outside the observed range of TradPed



Even though this graph looks as though socially responsive pedagogy leads to higher perceived value (a gentle, positive slope) the non-significant main effect indicates this slope is not statistically significant.

Further, even though it *appears* that higher ratings of traditional pedagogy are associated with greater perceived value, this main effect was also non-significant.

Consistent with the non-significant interaction effect, there is no evidence of an interaction effect in the figure.

```
interactions::sim_slopes(Mod_b_path, pred = SRPed, modx = TradPed)

## Warning: 5.02351626747081 is outside the observed range of TradPed

## JOHNSON-NEYMAN INTERVAL
##
## The Johnson-Neyman interval could not be found. Is the p value for your
## interaction term below the specified alpha?
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of SRPed when TradPed = 3.599941 (- 1 SD):
##
##   Est.    S.E.    t val.      p
##   ---- - ----- -----
##   0.10    0.13     0.77    0.44
##
## Slope of SRPed when TradPed = 4.311728 (Mean):
##
##   Est.    S.E.    t val.      p
##   ---- - ----- -----
##   0.15    0.12     1.25    0.22
##
## Slope of SRPed when TradPed = 5.023516 (+ 1 SD):
##
##   Est.    S.E.    t val.      p
##   ---- - ----- -----
##   0.21    0.14     1.46    0.15

# interactions::sim_slopes(Mod_b_path, pred=TradPed, modx = SRPed)
# #sometimes I like to look at it in reverse -- like in the plots
```

Consistent with both the non-significant interaction term and the the non-significant main effect of socially responsive pedagogy, there was not statistically significant effect of socially responsive pedagogy on perceived value to the student at the traditional values of the mean (4.31) and $\pm 1SD$ (3.60, 5.02) of the moderator, traditional pedagogy.

Analysis #2: A simple mediation

We are asking, “Does socially responsive pedagogy mediate the relationship between centering and perceived value to the student?”

Y = perceived value X = centering M = socially responsive pedagogy

Note. I switched to using the CENTERING (all caps) variable because it is 0/1, numeric (better for lavaan).

```
library(lavaan)

medmodel <- "
  Valued ~ b*SRPed + c_p*CENTERING
  SRPed ~a*CENTERING

  #intercepts
  CENTERING ~ CENTERING.mean*1
  Valued ~ Valued.mean*1

  indirect := a*b
  direct   := c_p
  total_c  := c_p + (a*b)
  "

set.seed(230925) #required for reproducible results because lavaan introduces randomness into
medmodel_fit <- lavaan::sem(medmodel, data = babydf, se = "bootstrap",
  missing = "fiml")
medmodel_Sum <- lavaan::summary(medmodel_fit, standardized = T, rsq = T,
  ci = TRUE)
medmodel_ParEsts <- lavaan::parameterEstimates(medmodel_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
medmodel_Sum
```

```
## lavaan 0.6.16 ended normally after 25 iterations
##
##   Estimator                               ML
##   Optimization method                    NLMINB
##   Number of model parameters             9
## 
##   Number of observations                84
##   Number of missing patterns            2
## 
## Model Test User Model:
##
##   Test statistic                         0.000
##   Degrees of freedom                      0
## 
## Parameter Estimates:
##
##   Standard errors                        Bootstrap
##   Number of requested bootstrap draws    1000
##   Number of successful bootstrap draws   1000
## 
```

```

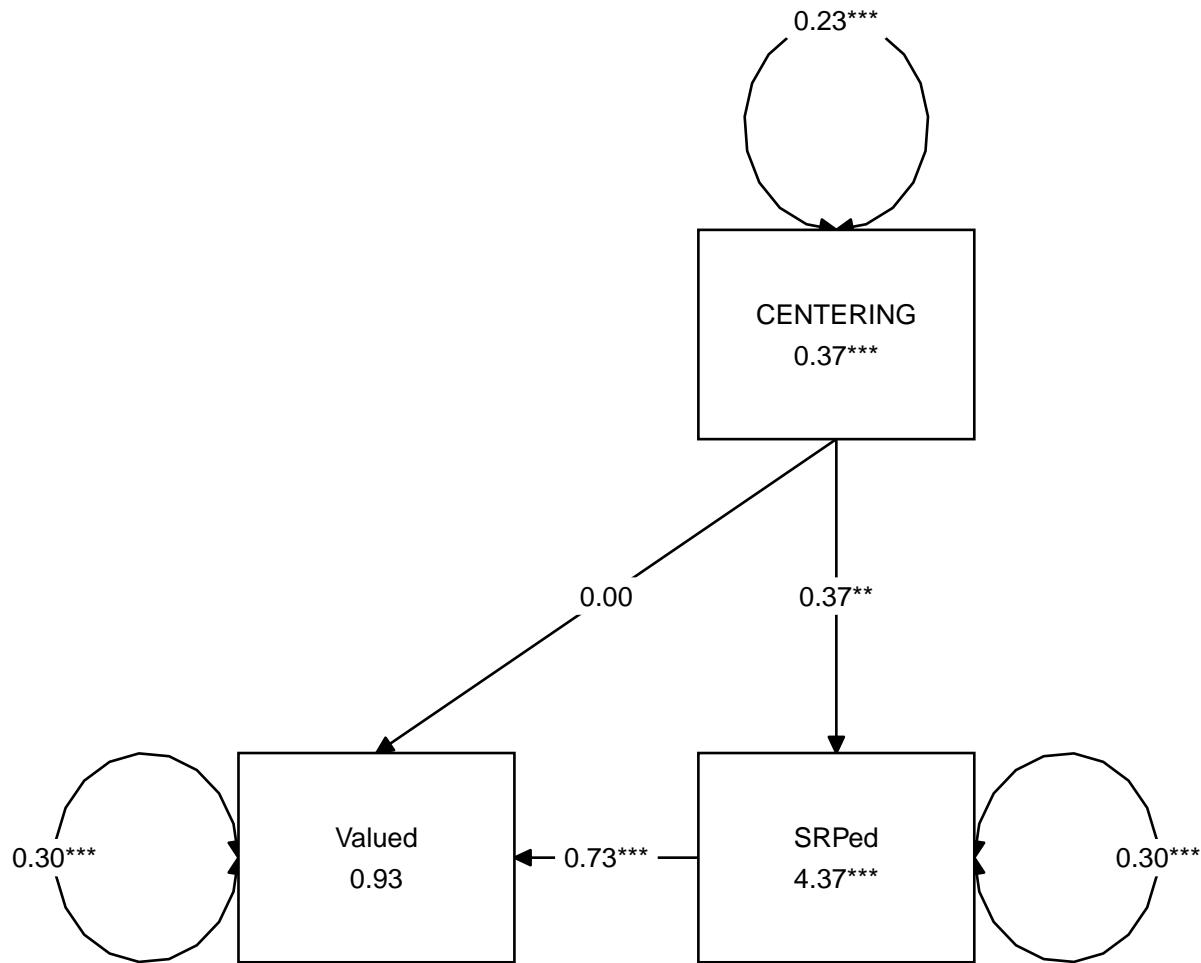
## Regressions:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## Valued ~
##   SRPed      (b)    0.728    0.124   5.877   0.000    0.455    0.933
##   CENTERIN (c_p)  0.004    0.124   0.032   0.974   -0.225    0.257
## SRPed ~
##   CENTERIN (a)    0.367    0.114   3.225   0.001    0.148    0.601
## Std.lv  Std.all
##
##   0.728    0.608
##   0.004    0.003
##
##   0.367    0.307
##
## Intercepts:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
##   CENTERI (CENT)  0.369    0.054   6.872   0.000    0.274    0.476
##   .Valued   (Vld.) 0.935    0.548   1.707   0.088    0.018    2.136
##   .SRPed        4.371    0.086  50.597   0.000    4.187    4.527
## Std.lv  Std.all
##   0.369    0.765
##   0.935    1.355
##   4.371    7.580
##
## Variances:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
##   .Valued       0.299    0.055   5.453   0.000    0.186    0.408
##   .SRPed        0.301    0.058   5.157   0.000    0.196    0.413
##   CENTERING    0.233    0.014  16.390   0.000    0.199    0.249
## Std.lv  Std.all
##   0.299    0.629
##   0.301    0.906
##   0.233    1.000
##
## R-Square:
##                               Estimate
##   Valued        0.371
##   SRPed        0.094
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
##   indirect      0.267    0.100   2.665   0.008    0.095    0.488
##   direct        0.004    0.124   0.032   0.974   -0.225    0.257
##   total_c      0.271    0.144   1.886   0.059   -0.014    0.551
## Std.lv  Std.all
##   0.267    0.187
##   0.004    0.003
##   0.271    0.190

```

```
medmodel_ParEsts
```

##	lhs	op	rhs	label	est	se	z	pvalue	ci.lower
## 1	Valued	~	SRPed	b	0.728	0.124	5.877	0.000	0.464
## 2	Valued	~	CENTERING	c_p	0.004	0.124	0.032	0.974	-0.225
## 3	SRPed	~	CENTERING	a	0.367	0.114	3.225	0.001	0.131
## 4	CENTERING	~1		CENTERING.mean	0.369	0.054	6.872	0.000	0.262
## 5	Valued	~1		Valued.mean	0.935	0.548	1.707	0.088	-0.040
## 6	Valued	~~	Valued		0.299	0.055	5.453	0.000	0.213
## 7	SRPed	~~	SRPed		0.301	0.058	5.157	0.000	0.205
## 8	CENTERING	~~	CENTERING		0.233	0.014	16.390	0.000	0.199
## 9	SRPed	~1			4.371	0.086	50.597	0.000	4.189
## 10	indirect	:=	a*b	indirect	0.267	0.100	2.665	0.008	0.109
## 11	direct	:=	c_p	direct	0.004	0.124	0.032	0.974	-0.225
## 12	total_c	:=	c_p+(a*b)	total_c	0.271	0.144	1.886	0.059	-0.025
##	ci.upper	std.lv	std.all	std.nox					
## 1	0.939	0.728	0.608	0.608					
## 2	0.250	0.004	0.003	0.003					
## 3	0.587	0.367	0.307	0.307					
## 4	0.476	0.369	0.765	0.765					
## 5	2.084	0.935	1.355	1.355					
## 6	0.454	0.299	0.629	0.629					
## 7	0.424	0.301	0.906	0.906					
## 8	0.249	0.233	1.000	1.000					
## 9	4.528	4.371	7.580	7.580					
## 10	0.528	0.267	0.187	0.187					
## 11	0.250	0.004	0.003	0.003					
## 12	0.546	0.271	0.190	0.190					

```
# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = medmodel_fit)
```



We can use the *tidySEM::get_layout* function to understand how our model is being mapped.

```
tidySEM::get_layout(medmodel_fit)
```

```
##      [,1]      [,2]
## [1,] "CENTERING" NA
## [2,] "SRPed"     "Valued"
## attr(,"class")
## [1] "layout_matrix" "matrix"       "array"
```

We can write code to remap them

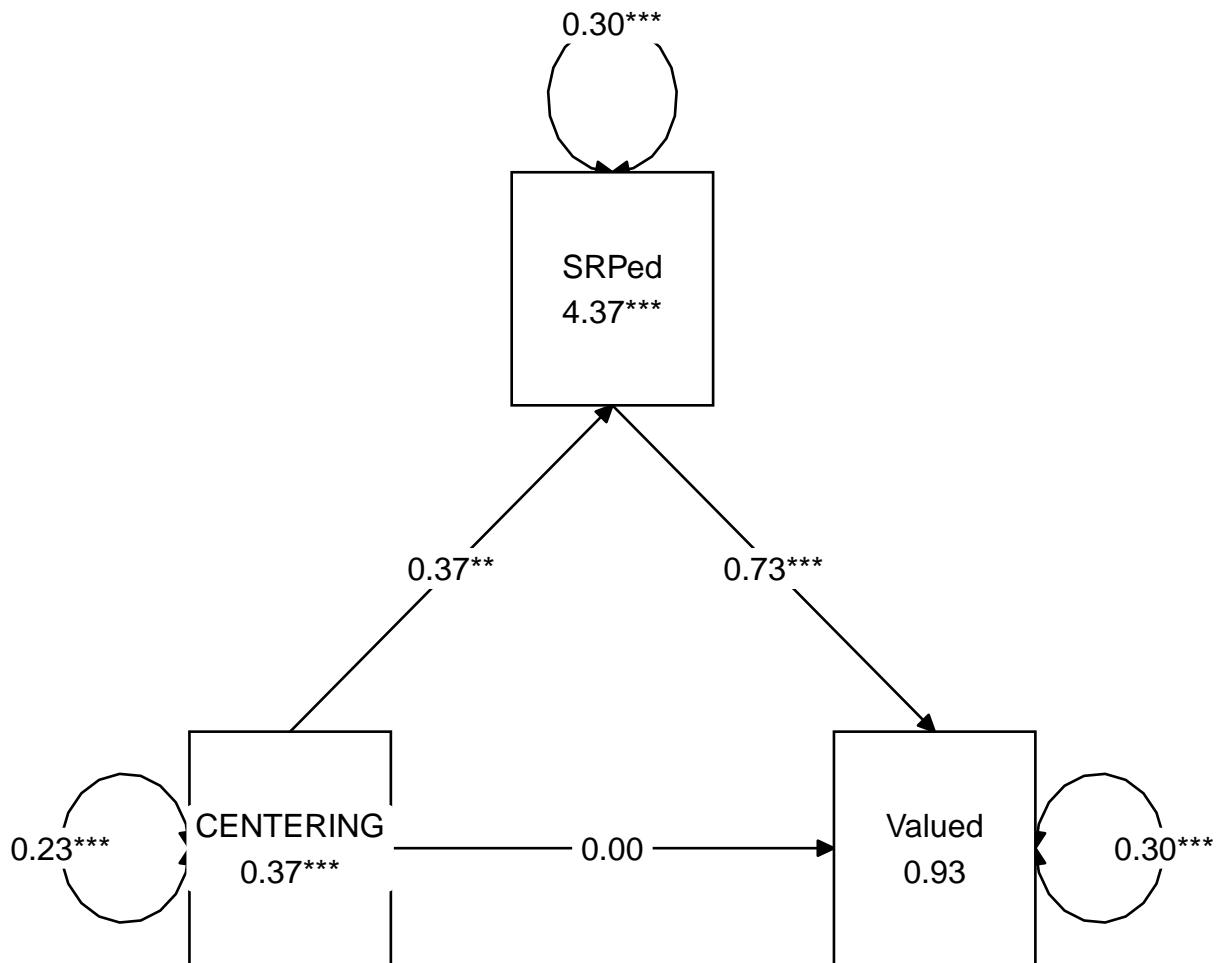
```
medmap <- tidySEM::get_layout("", "SRPed", "", "CENTERING", "", "Valued",
                                rows = 2)
medmap
```

```
##      [,1]      [,2]      [,3]
```

```
## [1] ""           "SRPed"   ""
## [2] "CENTERING" ""        "Valued"
## attr(,"class")
## [1] "layout_matrix" "matrix"    "array"
```

We can update the `tidySEM::graph_sem` function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(medmodel_fit, layout = medmap, rect_width = 1.25, rect_height = 1.25,
                    spacing_x = 2, spacing_y = 3, text_size = 4.5)
```



```
#### Specify and run the entire lavaan model {-}
```

```
set.seed(230925)
ModMedOnB <- "
  #equations
  SRPed ~ a*CENTERING
```

```

Valued ~ cp*CENTERING + b1*SRPed + b2*TradPed + b3*SRPed:TradPed

#intercepts
SRPed ~ SRPed.mean*1
Valued ~ Valued.mean*1

#means, variances of W for simple slopes
TradPed ~ TradPed.mean*1
TradPed ~~ TradPed.var*TradPed

#index of moderated mediation, there will be an a and b path in the product
#if the a and/or b path is moderated, select the label that represents the moderation

imm := a*b3

#Note that we first create the indirect product, then add to it the product of the imm and
indirect.SDbelow := a*b1 + imm*(TradPed.mean - sqrt(TradPed.var))
indirect.mean := a*b1 + imm*(TradPed.mean)
indirect.SDabove := a*b1 + imm*(TradPed.mean + sqrt(TradPed.var))

"
set.seed(230925) #required for reproducible results because lavaan introduces randomness into
ModMedOnB_fit <- lavaan::sem(ModMedOnB, data = babydf, se = "bootstrap",
  missing = "fiml", bootstrap = 1000)

## Warning in lav_partable_vnames(FLAT, "ov.x", warn = TRUE): lavaan WARNING:
##   model syntax contains variance/covariance/intercept formulas
##   involving (an) exogenous variable(s): [TradPed]; These variables
##   will now be treated as random introducing additional free
##   parameters. If you wish to treat those variables as fixed, remove
##   these formulas from the model syntax. Otherwise, consider adding
##   the fixed.x = FALSE option.

ModMedOnBsum <- lavaan::summary(ModMedOnB_fit, standardized = TRUE, rsq = T,
  ci = TRUE)
ModMedOnBParamEsts <- lavaan::parameterEstimates(ModMedOnB_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
ModMedOnBsum

## lavaan 0.6.16 ended normally after 29 iterations
## 
##   Estimator                      ML
##   Optimization method            NLMINB
##   Number of model parameters      13
## 
##   Number of observations        84

```

```

## Number of missing patterns                                3
##
## Model Test User Model:
##
## Test statistic                                         390.225
## Degrees of freedom                                     5
## P-value (Chi-square)                                    0.000
##
## Parameter Estimates:
##
## Standard errors                                     Bootstrap
## Number of requested bootstrap draws                  1000
## Number of successful bootstrap draws                1000
##
## Regressions:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## SRPed ~
##   CENTERING (a)      0.352    0.114   3.077   0.002    0.133    0.590
## Valued ~
##   CENTERING (cp)    0.019    0.108   0.174   0.862   -0.195    0.225
##   SRPed (b1)        0.060    0.528   0.114   0.909   -1.726    0.436
##   TradPed (b2)      0.633    0.497   1.274   0.203   -0.894    0.977
##   SRPd:TrdP (b3)   0.014    0.113   0.126   0.900   -0.032    0.396
## Std.lv Std.all
##
##   0.352    0.296
##
##   0.019    0.014
##   0.060    0.055
##   0.633    0.723
##   0.014    0.110
##
## Intercepts:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## .SRPed (SRP.)     4.382    0.087  50.609   0.000    4.196    4.537
## .Valued (Vld.)    0.939    2.160   0.435   0.664   -0.148    8.017
## TradPed (TrP.)    4.296    0.076  56.560   0.000    4.142    4.434
## SRPd:TP          19.741   0.524  37.679   0.000   18.697   20.707
## Std.lv Std.all
##   4.382    7.627
##   0.939    1.497
##   4.296    5.993
##   19.741   4.104
##
## Variances:
##                               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## TradPed (TrP.)    0.514    0.140   3.681   0.000    0.272    0.803
## .SRPed             0.301    0.059   5.110   0.000    0.195    0.416

```

```

## .Valued           0.181   0.029   6.327   0.000   0.111   0.223
## SRPd:TP          23.133   4.269   5.419   0.000  15.303  32.150
## Std.lv  Std.all
## 0.514   1.000
## 0.301   0.912
## 0.181   0.461
## 23.133  1.000
##
## R-Square:
##                   Estimate
## SRPed            0.088
## Valued           0.539
##
## Defined Parameters:
##                   Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
## imm              0.005   0.040   0.125   0.901  -0.012   0.145
## indirect.SDblw  0.039   0.066   0.593   0.553  -0.122   0.151
## indirect.mean    0.043   0.055   0.784   0.433  -0.074   0.156
## indirect.SDabv  0.046   0.057   0.809   0.418  -0.065   0.165
## Std.lv  Std.all
## 0.005   0.032
## 0.039   0.178
## 0.043   0.210
## 0.046   0.243

```

ModMedOnBParamEsts

```

##      lhs op          rhs
## 1     SRPed ~          CENTERING
## 2     Valued ~          CENTERING
## 3     Valued ~          SRPed
## 4     Valued ~          TradPed
## 5     Valued ~          SRPed:TradPed
## 6     SRPed ~1
## 7     Valued ~1
## 8     TradPed ~1
## 9     TradPed ~~        TradPed
## 10    SRPed ~~          SRPed
## 11    Valued ~~          Valued
## 12    SRPed:TradPed ~~  SRPed:TradPed
## 13    CENTERING ~~      CENTERING
## 14    SRPed:TradPed ~1
## 15    CENTERING ~1
## 16    imm :=             a*b3
## 17 indirect.SDbelow := a*b1+imm*(TradPed.mean-sqrt(TradPed.var))
## 18 indirect.mean :=      a*b1+imm*(TradPed.mean)
## 19 indirect.SDabove := a*b1+imm*(TradPed.mean+sqrt(TradPed.var))

```

```

##          label    est     se      z pvalue ci.lower ci.upper std.lv std.all
## 1           a  0.352  0.114  3.077  0.002   0.117   0.573   0.352   0.296
## 2           cp  0.019  0.108  0.174  0.862  -0.201   0.223   0.019   0.014
## 3           b1  0.060  0.528  0.114  0.909  -1.264   0.564   0.060   0.055
## 4           b2  0.633  0.497  1.274  0.203  -0.889   0.982   0.633   0.723
## 5           b3  0.014  0.113  0.126  0.900  -0.032   0.396   0.014   0.110
## 6      SRPed.mean 4.382  0.087 50.609  0.000   4.191   4.537   4.382   7.627
## 7      Valued.mean 0.939  2.160  0.435  0.664  -0.117   8.165   0.939   1.497
## 8      TradPed.mean 4.296  0.076 56.560  0.000   4.137   4.431   4.296   5.993
## 9      TradPed.var 0.514  0.140  3.681  0.000   0.275   0.811   0.514   1.000
## 10          0.301  0.059  5.110  0.000   0.205   0.430   0.301   0.912
## 11          0.181  0.029  6.327  0.000   0.141   0.281   0.181   0.461
## 12          23.133 4.269  5.419  0.000  15.792  33.016  23.133   1.000
## 13          0.233  0.000     NA     NA   0.233   0.233   0.233   1.000
## 14          19.741 0.524 37.679  0.000  18.724  20.761  19.741   4.104
## 15          0.369  0.000     NA     NA   0.369   0.369   0.369   0.765
## 16          imm  0.005  0.040  0.125  0.901  -0.011   0.147   0.005   0.032
## 17 indirect.SDbelow 0.039  0.066  0.593  0.553  -0.056   0.197   0.039   0.178
## 18 indirect.mean  0.043  0.055  0.784  0.433  -0.046   0.179   0.043   0.210
## 19 indirect.SDabove 0.046  0.057  0.809  0.418  -0.063   0.167   0.046   0.243
##      std.nox
## 1      0.613
## 2      0.030
## 3      0.055
## 4      0.487
## 5      0.110
## 6      7.627
## 7      1.497
## 8      5.993
## 9      1.000
## 10     0.912
## 11     0.461
## 12     1.000
## 13     0.233
## 14     4.104
## 15     0.369
## 16     0.067
## 17     0.369
## 18     0.436
## 19     0.503

```

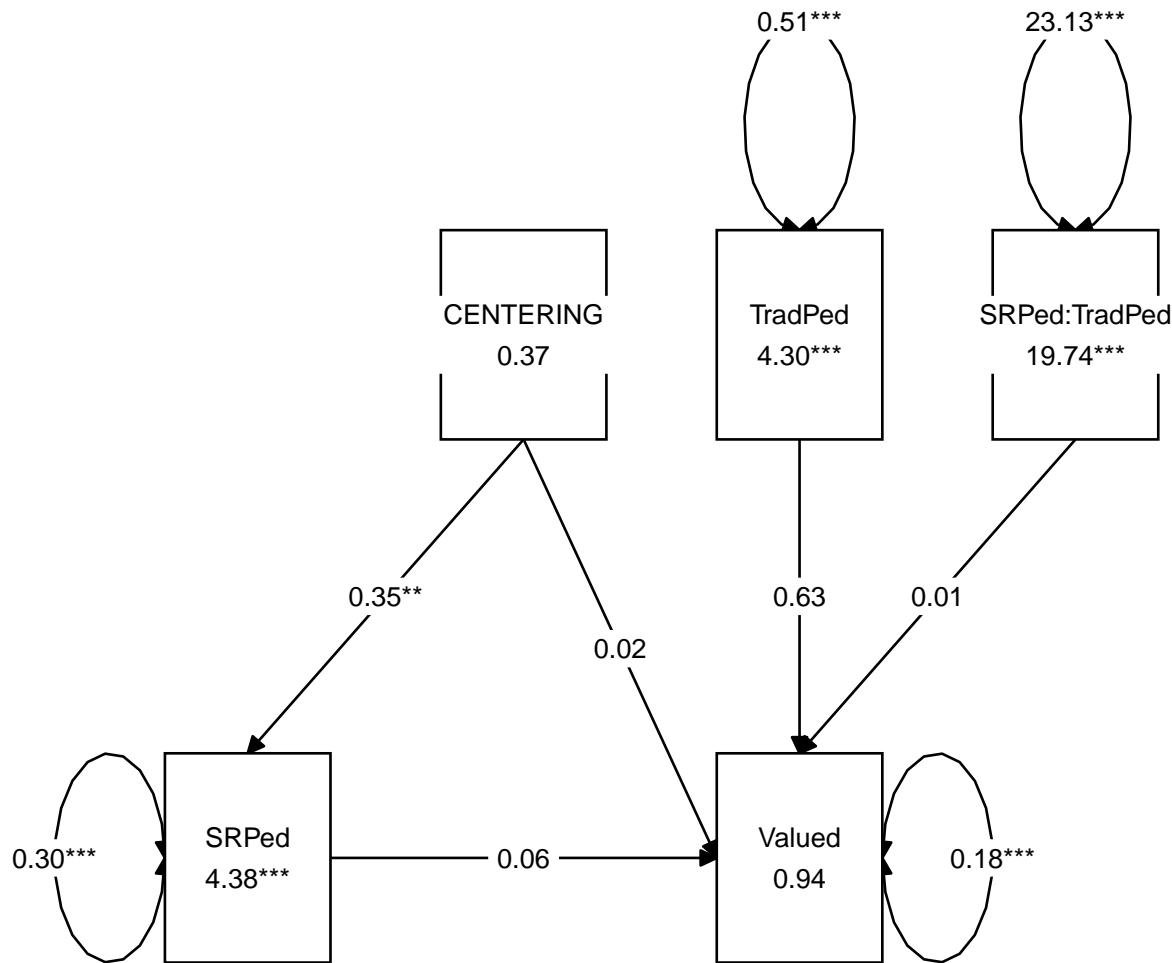
Use tidySEM to create a figure

```

# only worked when I used the library to turn on all these pkgs
library(lavaan)
library(dplyr)

```

```
library(ggplot2)
library(tidySEM)
tidySEM::graph_sem(model = ModMedOnB_fit)
```



We can use the `tidySEM::get_layout` function to understand how our model is being mapped.

```
tidySEM::get_layout(ModMedOnB_fit)
```

```
##      [,1]     [,2]     [,3]     [,4]
## [1,] NA     "CENTERING" "TradPed" "SRPed:TradPed"
## [2,] "SRPed" NA     "Valued"  NA
## attr("class")
## [1] "layout_matrix" "matrix"       "array"
```

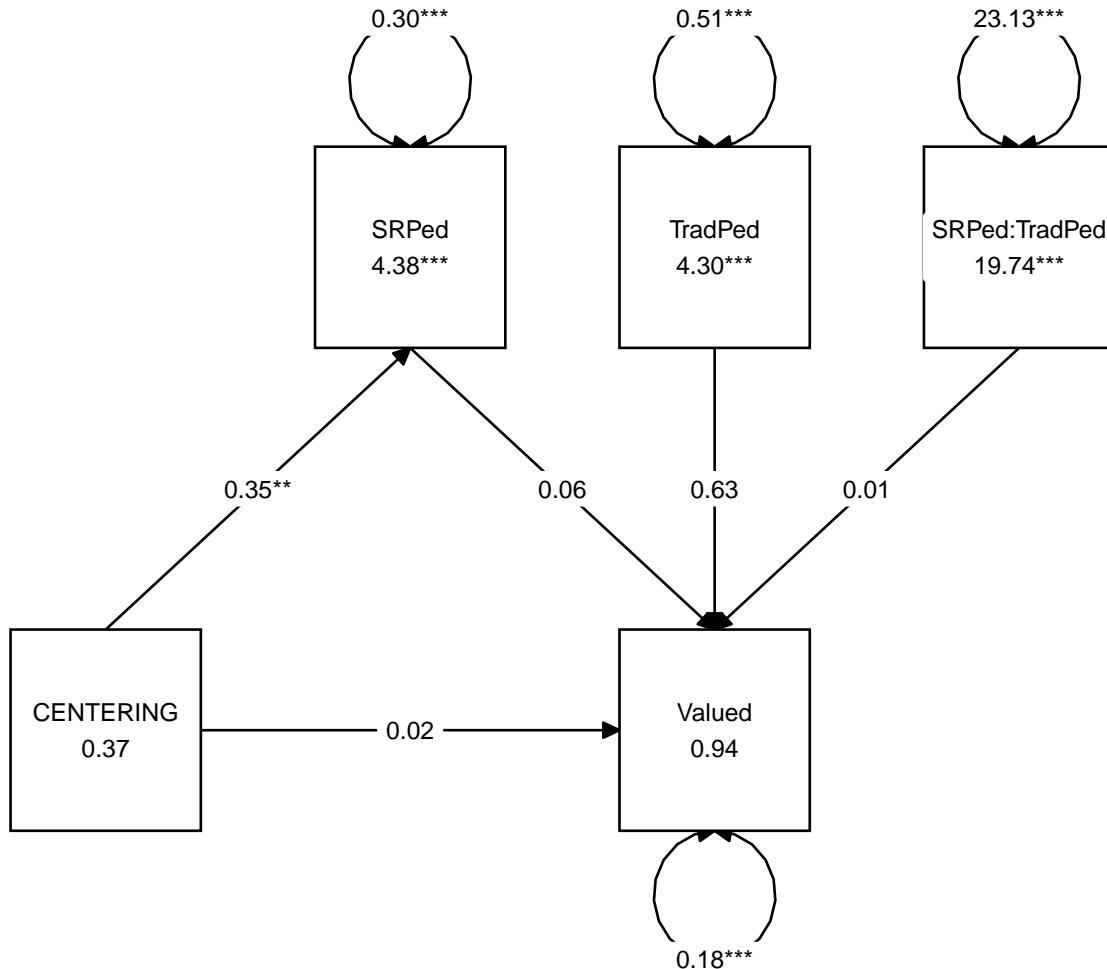
We can write code to remap them

```
comb_map <- tidySEM::get_layout("", "SRPed", "TradPed", "SRPed:TradPed",
  "CENTERING", "", "Valued", "", rows = 2)
comb_map
```

```
##      [,1]      [,2]      [,3]      [,4]
## [1,] ""       "SRPed"  "TradPed" "SRPed:TradPed"
## [2,] "CENTERING" ""       "Valued"  ""
## attr(,"class")
## [1] "layout_matrix" "matrix"     "array"
```

We can update the *tidySEM::graph_sem* function with our new model to produce something that will better convey our analyses and its results.

```
tidySEM::graph_sem(ModMedOnB_fit, layout = comb_map, rect_width = 1.25,
  rect_height = 1.25, spacing_x = 2, spacing_y = 3, text_size = 3.5)
```



Create a table that includes regression output for the M and Y variables and the moderated effects

Exporting the results to a .csv file will help us create a journal-ready table, outside of the R environment.

```
write.csv(ModMedOnBParamEsts, file = "ModMedOnBParamEsts.csv") #optional to write it to a .cs
```

Table 1

Analysis of Moderated Mediation for Centering, Socially Responsive Pedagogy, Traditional Pedagogy, and Perceived Value to the Student

Predictor	B	SE _B	p	R ²
Socially Responsive Pedagogy(M)				.09
Constant	4.382	0.087	<0.001	
Centering (a)	0.352	0.114	0.002	
Perceived Value (DV)				.54
Constant	0.939	2.160	0.664	
Centering (c')	0.019	0.108	0.862	
SRPed (b ₁)	0.060	0.528	0.909	
TradPed (b ₂)	0.633	0.497	0.203	
SRPed:TradPed (b ₃)	0.014	0.113	0.900	
Summary of Effects	B	SE _B	p	95% CI
IMM	0.005	0.040	0.901	-0.011, 0.147
Indirect (-1SD)	0.039	0.066	0.553	-0.056, 0.197
Indirect (M)	0.043	0.055	0.433	-0.046, 0.179
Indirect (+1SD)	0.046	0.057	0.418	-0.063 0.167

Note. SRPed = socially responsive pedagogy; TradPed = traditional pedagogy. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

APA Style Write-up

Method/Aalytic Strategy

Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16). We specified a moderated mediation model predicting perceived value to the student from centering stage, mediated by socially responsive pedagogy. We further predicted that the relationship between socially responsive pedagogy to perceived value (i.e., the *b* path) would be moderated by traditional pedagogy.

Results

Preliminary Analyses

- Missing data analysis and managing missing data
- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

Primary Analyses Our analysis evaluated a moderation mediation model perceived value to the student (Y/Valued) from centering stage (X/Centering) mediated by socially responsive pedagogy (M/SRPed). Traditional pedagogy (W/TradPed) was our moderating variable. We specified a moderation of path *b* (M/SRPed to Y/Valued). Data were analyzed with maximum likelihood estimation in the R package *lavaan* (v. 0.6-16); the significance of effects were tested with 1000 bootstrap confidence intervals. Results of the full model are presented in Table 1 and illustrated in Figure 1. The formula for the mediator and dependent variable are expressed below.

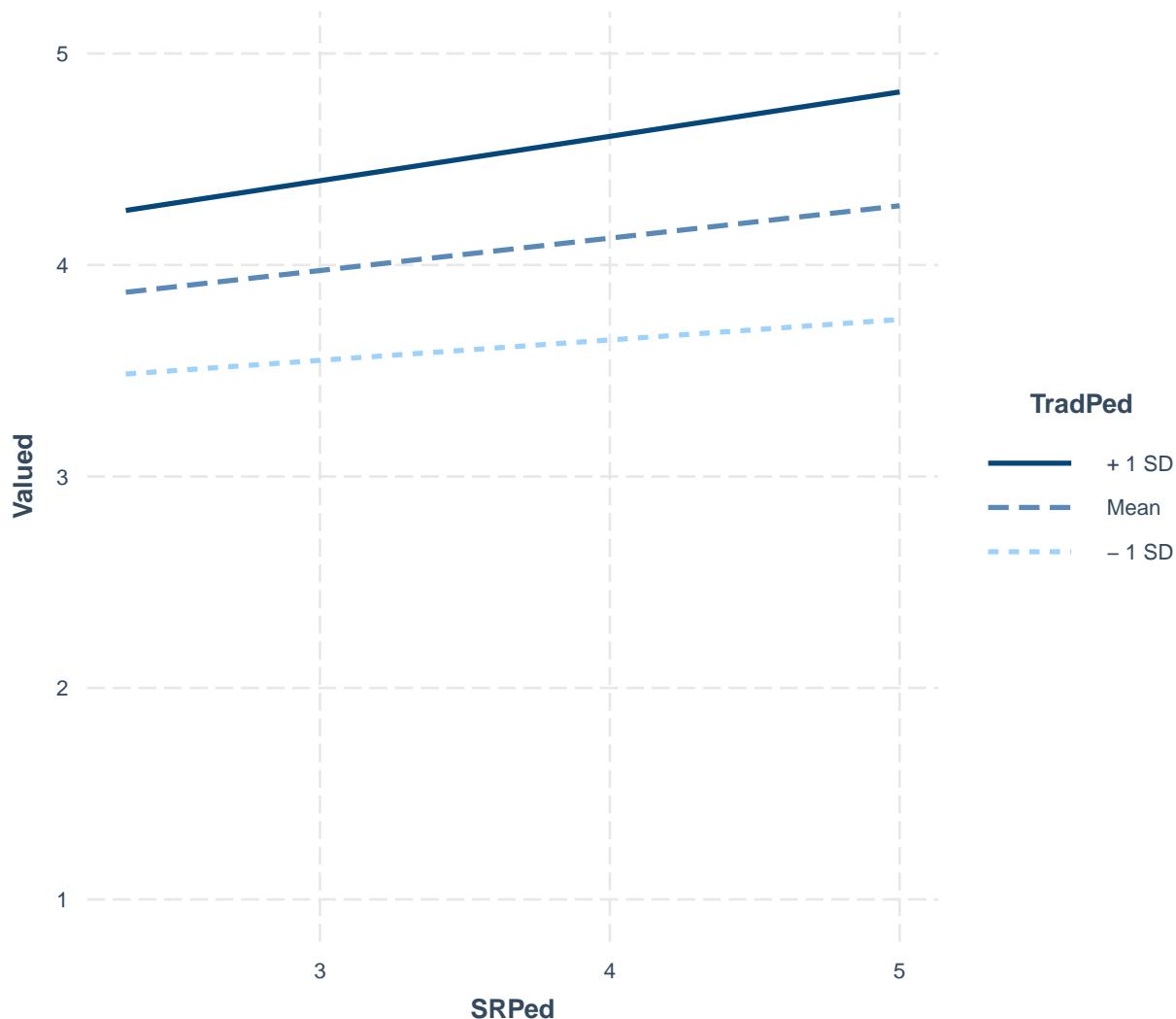
$$\hat{M} = 4.382 + 0.352X$$

$$\hat{Y} = 0.939 + (0.019)X + (0.060)M + (0.633)W + 0.014MW$$

Although the model accounted for 9% of variance in socially responsive pedagogy (mediator) and 54% of variance in the perceived value to the student (dependent variable), results suggested non significant main, interaction, and mediated effects. Correspondingly, the index of moderated mediation was not significant ($B = 0.005, p = 0.901, 95CI(-0.011, 0.147)$), suggesting that the indirect effects were not conditional on the values of the moderator. Figure 1 illustrates the conditional effects (all non-significant) of socially responsive pedagogy (M) on perceived value to the student (Y) among across the $M \pm 1SD$ of the moderator, traditional pedagogy. Complete results are found in Table 1.

```
interactions::interact_plot(Mod_b_path, pred = SRPed, modx = TradPed) +
  ylim(1, 5)
```

```
## Warning: 5.02351626747081 is outside the observed range of TradPed
```



```
# interactions::interact_plot(Mod_c_path, pred = TradPed, modx =
# SRPed + ylim(1,5))
```

STRUCTURAL EQUATION MODELING

Chapter 9

Establishing the Measurement Model

[Screencasted Lecture Link](#)

This lesson opens a series on structural equation modeling devoted to the full latent variable model. Full latent variable models test the directional linkages between variables in the model and they contain both (a) measurement and (b) structural components. Thus, evaluating a full latent variable model is completed in two larger steps which establish the measurement model first and then proceed to evaluating the structural model. The focus of this lesson is on the first step – establishing the measurement model.

9.1 Navigating this Lesson

There is about two hours 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

Learning objectives from this lecture include the following:

- Specify a measurement model with item-level indicators.
- Respecify a measurement model with parcelled indicators.
- Interpret goodness-of-fit indices (e.g., Chi-square, CFI, RMSEA) associated with the model.
- Interpret the regression weights associated with the model.
- List pros and cons of using parcels in measurement models.

9.1.2 Planning for Practice

This is the first of a two-part lesson on structural equation modeling. In this lesson we specify and evaluate the *measurement model* that precedes evaluating a full *structural model*. This means that you will want to have a structural model in mind. For the practice, this should involve a minimum of three variables.

The suggestions for homework are graded in complexity. If you have completed one or more of the prior lessons where path analysis (i.e., no latent variables) were used, you might consider recycling those for this set of practice problems.

- Rework the problem in the chapter by changing the random seed in the code that simulates the data. This should provide minor changes to the data, but the results will likely be very similar.
- Use the research data from the chapter, but evaluate a different set of variables.
- Use data from another lesson or data that is available to you.

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.

- 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 6, Application 4: Testing the Factorial Validity of a Causal Structure
- Kline, R. (2016). Principles and practice of structural equation modeling (Fourth ed., Methodology in the social sciences). New York: The Guilford Press.
 - Chapter 4, Data Preparation and Psychometrics Review
 - Chapter 10, Specification and Identification of Structural Regression Models
 - Chapter 11, Estimation and Local Fit Testing
- Little, T. D., Rhemtulla, M., Gibson, K., & Schoemann, A. M. (2013). Why the items versus parcels controversy needn't be one. *Psychological Methods*, 18(3), 285–300. <https://doi.org/10.1037/a0033266>
 - I conducted a brief literature search for updated information on parceling, this one continues to be at the top of articles considered to be authoritative.
- Kim, P. Y., Kendall, D. L., & Cheon, H.-S. (2017). Racial microaggressions, cultural mistrust, and mental health outcomes among Asian American college students. *American Journal of Orthopsychiatry*, 87(6), 663–670. <https://doi-org.ezproxy.spu.edu/10.1037/ort0000203>
 - This is the research vignette for this lesson.

9.1.4 Packages

The script below will (a) check to see if the following packages are installed on your computer and, if not (b) install them. You may wish to remove the hashtags and run this chunk if this is the first time you are conducting analyses such as these.

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

9.2 Introduction to Structural Equation Modeling (SEM)

In the lesson progression in Recentering Psych Stats, we used ordinary least squares (OLS) approaches as we learned analysis of variance (and hopefully coming soon multiple regression). As we entered more complex modeling, we began to use maximum likelihood estimators (MLE). A comparison of these two approaches was provided in the lesson on [Simple Moderation in OLS and MLE](#).

SEM is yet another progression in regression and it has several distinguishing aspects.

- SEM uses **latent variables**. Latent variables are not directly observed or measured (i.e., they do not exist as a column in your data). Rather, they are *inferred* from other observed variables. The latent variable (i.e., depression) is presumed to *cause* scores on the observed (sometimes termed *manifest*) variables. In these lessons, we can easily think of latent variables as the factor (or scale) and the observed/manifest variables as its items.
 - To clarify, SEM models can incorporate latent (unobserved) and manifest (observed) in the same model.
- SEM evaluates *causal processes* through a series of structural (i.e., regression) equations.
- SEM provides **global fit indices** that provide an overall evaluation of the *goodness of fit* of the model. State another way, they indicate how closely the model's predictions align with the actual data.
- SEM tests **multiple hypotheses, simultaneously**. That is, we can easily combine separate smaller models (e.g., simple mediation, simple moderation) into a grander model (e.g., moderated mediation).
- SEM permits **multiple dependent variables**. Actually, in SEM we typically refer to variables as *exogenous* (variables that only serve as predictors) and *endogenous* (variables that are predicted [even if they also predict]).
- In contrast to traditional multivariate procedures that can neither assess nor correct for measurement error, SEM provides **explicit estimates of error variance parameters**.
- SEM models have a long history of being **represented pictorially** and the conventions of these figures make it possible for them to efficiently convey the findings.

With all of these advantages, SEM is widely used for nonexperimental research.

In this lesson we start on the journey toward evaluating a full latent variable model; sometimes these are called hybrid models [noa, b] because they are a mix of path analysis and confirmatory factor analysis (CFA). Today we focus on the CFA portion because we will specify (and likely respecify) the *measurement model*. In evaluating the measurement model we will specify a model where each of the constructs (factors) is represented in its latent form. That is each construct is represented as a factor (a latent variable) by its manifest, item-level, variables. In our measurement model we will allow all of the factors to covary with each other. It is important to note that this model will have the best fit of all because all of the structural paths are saturated. Stated another way, the subsequent test of the structural model will have worse fit. This means that if the fit of the measurement model is below our thresholds, we will investigate options for improving it before moving to evaluation of the structural model.

9.3 Workflow for Evaluating a Structural Model

The following workflow is one that provides an overview of the entire process of evaluating a structural model.

Workflow: Evaluating a Structural Model	
A Priori Power Analysis <ul style="list-style-type: none"> Conduct an a priori power analysis to determine the appropriate sample size. Draw estimates of effect from pilot data and/or the literature. 	
Scrubbing & Scoring <ul style="list-style-type: none"> Import data and format (i.e., variable naming, reverse-scoring) item level variables. Analyze item-level missingness. If using scales, create the mean scores of the scales. Determine and execute approach for managing missingness. Popular choices are available item analysis (e.g., Parent, 2013) and multiple imputation. Analyze scale-level missingness. Create a df with <i>only</i> the items (scaled in the proper direction). 	
Data Diagnostics <ul style="list-style-type: none"> Evaluate univariate normality (i.e., one variable at a time) with Shapiro-Wilks tests; $p < .05$ indicates a violation of univariate normality. Evaluate multivariate normality (i.e., all continuously scaled variables simultaneously) with Mahalanobis test. Identify outliers (e.g., cases with Mahal values $> 3 SDs$ from the centroid). Consider deleting (or transforming if there is an extreme-ish “jump” in the sorted values). Evaluate internal consistency of the scaled scores with Cronbach’s alpha or omega; the latter is increasingly preferred. 	
Specify and evaluate a measurement model <ul style="list-style-type: none"> In this just-identified (saturated) model, all latent variables are specified as covarying <ul style="list-style-type: none"> For LVs with 3 items or more, remember to set a marker/reference variable. For LVs with 2 items, constrain the loadings to be equal For single-item indicators fix the error variance to zero (or a non-zero estimate of unreliability) Evaluate results with <i>global</i> (e.g., χ^2, CFI, RMSEA, SRMR) and <i>local</i> (i.e., factor loadings and covariances) fit indices In the event of poor fit, respecify LVs with multiple indicators with parcels. Nested alternative measurement models can be compared with χ^2 difference, ΔCFI tests; non-nested models with AIC, and BIC tests 	
Specify and evaluate a structural model <ul style="list-style-type: none"> Replace the covariances with paths that represent the a priori hypotheses <ul style="list-style-type: none"> These models could take a variety of forms. It is possible to respecify models through trimming or building approaches. Evaluate results using: <ul style="list-style-type: none"> global fit indices (e.g., χ^2, CFI, RMSEA, SRMS), local fit indices (i.e., strength and significance of factor loadings, covariances, and additional model parameters [e.g., indirect effects]) Consider respecifying and evaluating one or more <i>alternative</i> models: <ul style="list-style-type: none"> <i>Forward searching</i> frees parameters (i.e., adds paths or covariances) and can use modification indices as a guide. <i>Backward searching</i> restrains parameters (i.e., deletes paths or covariances) and can use low and non-significant paths as a guide. Compare the fit of the alternate models <ul style="list-style-type: none"> Nested models can be compared with χ^2 difference, ΔCFI tests Non-nested models can use AIC, and BIC comparisons (lower values suggest better fit) 	
Quick Guide for Global and Comparative Fit Statistics <ul style="list-style-type: none"> $\chi^2, p > .05$; this test is sensitive to sample size and this value can be difficult to attain CFI $> .95$ (or at least $.90$) RMSEA (and associated 90%CI) are $< .05$ ($< .08$, or at least $< .10$) SRMR $< .08$ (or at least $< .10$) Combination rule: CFI $< .95$ and SRMR $< .08$ AIC and BIC are compared; the lowest values suggest better models $\chi^2\Delta$ is statistically significant; the model with the superior global fit is the better model ΔCFI is greater than 0.01; the model with CFI values closest to 1.0 has better fit 	

Evaluating a structural model involves the following steps:

- A Priori Power Analysis
 - Conduct an a priori power analysis to determine the appropriate sample size. __ Draw estimates of effect from pilot data and/or the literature.

- Scrubbing & Scoring
 - Import data and format (i.e., variable naming, reverse-scoring) item level variables.
 - Analyze item-level missingness.
 - If using scales, create the mean scores of the scales.
 - Determine and execute approach for managing missingness. Popular choices are available item analysis (e.g., Parent, 2013) and multiple imputation.
 - Analyze scale-level missingness.
 - Create a df with only the items (scaled in the proper direction).
- Data Diagnostics
 - Evaluate univariate normality (i.e., one variable at a time) with Shapiro-Wilks tests; $p < .05$ indicates a violation of univariate normality.
 - Evaluate multivariate normality (i.e., all continuously scaled variables simultaneously) with Mahalanobis test. Identify outliers (e.g., cases with Mahal values > 3 SDs from the centroid). Consider deleting (or transforming if there is an extreme-ish “jump” in the sorted values).
 - Evaluate internal consistency of the scaled scores with Cronbach’s alpha or omega; the latter is increasingly preferred. Specify and evaluate a measurement model
 - In this just-identified (saturated) model, all latent variables are specified as covarying.
 - * For LVs with 3 items or more, remember to set a marker/reference variable,
 - * For LVs with 2 items, constrain the loadings to be equal,
 - * For single-item indicators fix the error variance to zero (or a non-zero estimate of unreliability).
 - Evaluate results with global (e.g., χ^2 , CFI, RMSEA, SRMR) and local (i.e., factor loadings and covariances) fit indices.
 - In the event of poor fit, respecify LVs with multiple indicators with parcels.
 - Nested alternative measurement models can be compared with χ^2 difference, ΔCFI tests; non-nested models with AIC, and BIC tests .
- Specify and evaluate a structural model.
 - Replace the covariances with paths that represent the a priori hypotheses.
 - * These models could take a variety of forms.
 - * It is possible to respecify models through trimming or building approaches.
 - Evaluate results using
 - * *global* fit indices (e.g., χ^2 , CFI, RMSEA, SRMS),
 - * *local* fit indices (i.e., strength and significance of factor loadings, covariances, and additional model parameters [e.g., indirect effects]).
 - Consider respecifying and evaluating one or more *alternative* models.
 - * *Forward searching* involves freeing parameters (adding paths or covariances) and can use modification indices as a guide.
 - * *Backward searching* involves restraining parameters (deleting paths or covariances) and can use low and non-significant paths as a guide.
 - Compare the fit of the alternate models.
 - * Nested models can be compared with χ^2 difference and ΔCFI tests.
 - * Non-nested models can be compared with AIC and BIC (lower values suggest better fit).

- Quick Guide for Global and Comparative Fit Statistics.
 - χ^2 , $p < .05$; this test is sensitive to sample size and this value can be difficult to attain
 - $CFI > .95$ (or at least $.90$)
 - RMSEA (and associated 90%CI) are $< .05$ ($< .08$, or at least $< .10$)
 - SRMR $< .08$ (or at least $< .10$)
 - Combination rule: $CFI < .95$ and SRMR $< .08$
 - AIC and BIC are compared; the lowest values suggest better models
 - $\chi^2\Delta$ is statistically significant; the model with the superior fit is the better model
 - δCFI is greater than 0.01; the model with CFI values closest to 1.0 has better fit

The focus of this lesson is on the specification, evaluation, and respecification of the measurement model.

9.4 The Measurement Model: Specification and Evaluation

Structural models include both *measurement* and *structural* portions. The **measurement model** has two primary purposes. First the measurement model **specifies the latent variables**. That is, CFA-like models (i.e., one per latent variable) define each latent variable (i.e., scale score – but not “scored”) by its observed indicators (i.e., survey items). Resulting factor loadings indicate the strength of the relationships between the observed items and their latent variable.

Second, the measurement model allows the researchers to **assess the goodness of model fit**. A well-fitting model is required for accurately interpreting the relationships between the latent variables in the structural model. Additionally, the fit of the structural model will never surpass that of the measurement model. Stated another way – if the fit of the measurement model is inferior, the structural model is likely to be worse. There is at least one exception – when both the structural and measurement models are just-identified (i.e., fully saturated with zero degrees of freedom) model fit will be identical.

The specification of the measurement model involves:

- **Identifying** each latent variable with its prescribed observed variables (i.e., scale items). Note that the latent variable will not exist in the dataset. When we engaged in OLS regression and path analysis we created scale and subscale scores. In SEM, we do not do this. Rather we allow the latent variable to be defined by items (but they are not averaged or summed in any way).
- Specifying a **saturated** model such that $df = 0$ and it is *just-identified*. You might think of the measurement model as a *correlated factors model* because covariances will be allowed between all latent variables.
 - The structural model is typically more parsimonious (i.e., not saturated) than the measurement model and will be characterized by directional paths or the explicit absence of paths between some of the variables.
- **Respecifying the measurement model** is optional (but frequent). This may involve addressing ill-fitting or poorly specified models by
 - correcting any mistakes in model specification,

- *parceling* multiple-item factors,
- attending to issues like *Heywood cases* (e.g., a negative error variance)

Compared to the measurement model, the *structural model* (i.e., the model that represents your hypotheses) will be parsimonious. Whereas the measurement model is *saturated* with 0 degrees of freedom, the structural model is often *overidentified* (i.e., with positive degrees of freedom; not saturated) and characterized by directional paths (not covariances) between some of the variables. This leads to a necessary discussion of degrees of freedom in the context of SEM.

9.4.1 Degrees of Freedom and Model Identification

When running statistics with ordinary least squares, degrees of freedom was associated with the number of data points (i.e., cases, sample size) and the number of predictors (i.e., regression coefficients) in the model. In OLS models, degrees of freedom as involved in the calculation of statistical tests such as the *t*-test and *F*-Test; that is, they help assess whether the model fits the data well and whether the estimated coefficients are statistically significant. Consistent with Fisher's notion that degrees of freedom are a form of statistical currency [Rodgers, 2010], a larger degree degrees of freedom allows for greater precision in parameter estimates.

In SEM, degrees of freedom in the numerator represents the number of *independent pieces of information* such as the number of observed *variables* (not cases) minus the number of estimated parameters. The degrees of freedom in the denominator represent the number of restrictions or constraints placed upon the model, taking into account its complexity, the number of latent variables, and the pattern of relationships. Whether degrees of freedom are positive, negative, or zero determines the identification status of the model.

Underidentified or undetermined models have fewer observations (knowns) than free model parameters (unknowns). This results in negative degrees of freedom ($df_M \leq 0$). This means that it is impossible to find a unique set of estimates. The classic example for this is: $a + b = 6$ where there are an infinite number of solutions.

Just-identified or just-determined models have an equal number of observations (knowns) as free parameters (unknowns). This results in zero degrees of freedom ($df_M = 0$). Just-identified scenarios will result in a unique solution. The classic example for this is

$$a + b = 6$$

$$2a + b = 10$$

The unique solution is $a = 4$, $b = 2$.

Over-identified or overdetermined models have more observations (knowns) than free parameters (unknowns). This results in positive degrees of freedom ($df_M > 0$). In this circumstance, there is no single solution, but one can be calculated when a statistical criterion is applied. For example, there is no single solution that satisfies all three of these formulas:

$$a + b = 6$$

$$2a + b = 10$$

$$3a + b = 12$$

When we add this instruction “Find value of a and b that yield total scores such that the sum of squared differences between the observations (6, 10, 12) and these total scores is as small as possible.” Curious about the answer? An excellent description is found in Kline [2016b]. Model identification is an incredibly complex topic. For example, it is possible to have theoretically identified models and yet they are statistically unidentified and then the researcher must hunt for the source of the problem. As we work through a handful of SEM lessons, we will return to degrees of freedom and model identification again (and again).

For this lesson on measurement models, we are primarily concerned about the identification of each of our measurement models. Little has argued that [2013] each latent variable in an SEM model should be defined by a just-identified solution; that is, three indicators per construct. Why? Just-identified latent variables provide precise definitions of the construct. When latent variables are defined with four or more indicators (i.e., they are locally over-identified), the degrees of freedom generated from the measurement model for each construct (as well as the between-construct relations) introduces two sources of model fit. Thus, it introduces a statistical confound. When there are only two indicators per construct (i.e., they are locally under-identified) models are more likely to fail to converge and they may result in improper solutions. There are circumstances where one- and two-item indicators are necessary and there are statistical work-arounds for these circumstances.

As we work through this lesson, I will demonstrate several scenarios of the measurement model. The purpose of this demonstration is to show how the different approaches result in different results, particularly around model fit. At the outset, let me underscore Little’s [2013] admonishment that the representation of the measurement model should be determined a priori.

There are many more nuances of SEM. Let’s get some of these practically in place by working the vignette. As I designed this series of lessons, my plan is to rework some of the examples we did with path analysis (with maximum likelihood). This will hopefully (a) reduce the cognitive load by having familiar examples and (b) a direct comparison of results from both approaches.

9.5 Research Vignette

The research vignette comes from the Kim, Kendall, and Cheon’s [2017], “Racial Microaggressions, Cultural Mistrust, and Mental Health Outcomes Among Asian American College Students.” Participants were 156 Asian American undergraduate students in the Pacific Northwest. The researchers posited the a priori hypothesis that cultural mistrust would mediate the relationship between racial microaggressions and two sets of outcomes: mental health (e.g., depression, anxiety, well-being) and help-seeking.

Variables used in the study included:

- **REMS:** Racial and Ethnic Microaggressions Scale (Nadal, 2011). The scale includes 45 items on a 2-point scale where 0 indicates no experience of a microaggressive event and 1 indicates it was experienced at least once within the past six months. Higher scores indicate more experience of microaggressions.
- **CMI:** Cultural Mistrust Inventory (Terrell & Terrell, 1981). This scale was adapted to assess cultural mistrust harbored among Asian Americans toward individuals from the mainstream U.S. culture (e.g., Whites). The CMI includes 47 items on a 7-point scale where higher scores indicate a higher degree of cultural mistrust.

- **ANX, DEP, PWB:** Subscales of the Mental Health Inventory (Veit & Ware, 1983) that assess the mental health outcomes of anxiety (9 items), depression (4 items), and psychological well-being (14 items). Higher scores (on a 6 point scale) indicate stronger endorsement of the mental health outcome being assessed.
 - **HlpSkg:** The Attitudes Toward Seeking Professional Psychological Help – Short Form (Fischer & Farina, 1995) includes 10 items on a 4-point scale (0 = disagree, 3 = agree) where higher scores indicate more favorable attitudes toward help seeking.

For the lessons on measurement and structural models, we will evaluate a simple mediation model, predicting psychological well-being from racial ethnic microaggressions through cultural mistrust.

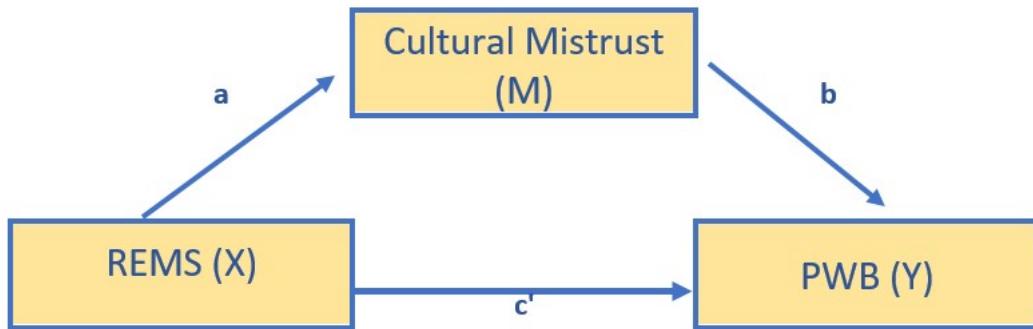


Figure 9.1: Image of the proposed statistical model

9.5.1 Simulating the data from the journal article

We used the `lavaan::simulateData` function for the simulation. If you have taken psychometrics, you may recognize the code as one that creates latent variables from item-level data. In trying to be as authentic as possible, we retrieved factor loadings from psychometrically oriented articles that evaluated the measures [Nadal, 2011, Veit and Ware, 1983]. For all others we specified a factor loading of 0.80. We then approximated the *measurement model* by specifying the correlations between all of the latent variable. We sourced these from the correlation matrix from the research vignette [Kim et al., 2017]. The process created data with multiple decimals and values that exceeded the boundaries of the variables. For example, in all scales there were negative values. Therefore, the final element of the simulation was a linear transformation that rescaled the variables back to the range described in the journal article and rounding the values to integer (i.e., with no decimal places).

```
# Entering the intercorrelations, means, and standard deviations from  
# the journal article  
Kim_generating_model <- "  
    ##measurement model  
    REMS =~ .82*Inf32 + .75*Inf38 + .74*Inf21 + .72*Inf17 + .69*Inf9 + .61*Inf36 + .51*In  
  
    CMI =~ .8*cmi1 + .8*cmi2 + .8*cmi3 + .8*cmi4 + .8*cmi5 + .8*cmi6 + .8*cmi7 + .8*cmi8 +
```

```

ANX =~ .80*Anx1 + .80*Anx2 + .77*Anx3 + .74*Anx4 + .74*Anx5 + .69*Anx6 + .69*Anx7 + .
DEP =~ .74*Dep1 + .83*Dep2 + .82*Dep3 + .74*Dep4
PWB =~ .83*pwb1 + .72*pwb2 + .67*pwb3 + .79*pwb4 + .77*pwb5 + .75*pwb6 + .74*pwb7 + .77*pwb8

HlpSkg =~ .8*hlpstk1 + .8*hlpstk2 + .8*hlpstk3 + .8*hlpstk4 + .8*hlpstk5 + .8*hlpstk6

# Means
REMS ~ 0.34*1
CMI ~ 3*1
ANX ~ 2.98*1
DEP ~ 2.36*1
PWB ~ 3.5*1
HlpSkg ~ 1.64*1
# Correlations (ha!)
REMS ~ 0.58*CMI
REMS ~ 0.26*ANX
REMS ~ 0.34*DEP
REMS ~ -0.25*PWB
REMS ~ -0.02*HlpSkg
CMI ~ 0.12*ANX
CMI ~ 0.19*DEP
CMI ~ -0.28*PWB
CMI ~ 0*HlpSkg
ANX ~ 0.66*DEP
ANX ~ -0.55*PWB
ANX ~ 0.07*HlpSkg
DEP ~ -0.66*PWB
DEP ~ 0.05*HlpSkg
PWB ~ 0.08*HlpSkg
""

set.seed(230916)
dfKim <- lavaan::simulateData(model = Kim_generating_model, model.type = "sem",
  meanstructure = T, sample.nobs = 156, standardized = FALSE)
library(tidyverse)

# used to retrieve column indices used in the rescaling script below
col_index <- as.data.frame(colnames(dfKim))

for (i in 1:ncol(dfKim)) {
  # for loop to go through each column of the dataframe apply only
  # to REMS variables
  if (i >= 1 & i <= 45) {
    dfKim[, i] <- scales::rescale(dfKim[, i], c(0, 1))
  }
  if (i >= 46 & i <= 116) {
    # apply only to CMI variables
  }
}

```

```

    dfKim[, i] <- scales::rescale(dfKim[, i], c(1, 7))
}
if (i >= 93 & i <= 116) {
  # apply only to mental health variables
  dfKim[, i] <- scales::rescale(dfKim[, i], c(1, 5))
}
if (i >= 117 & i <= 126) {
  # apply only to HlpSkng variables
  dfKim[, i] <- scales::rescale(dfKim[, i], c(0, 3))
}
}

# psych::describe(dfKim) +

library(tidyverse)
dfKim <- dfKim %>%
  round(0)

# I tested the rescaling the correlation between original and
# rescaled variables is 1.0 Kim_df_latent$INF32 <-
# scales::rescale(Kim_df_latent$Inf32, c(0, 1))
# cor.test(Kim_df_latent$Inf32, Kim_df_latent$INF32,
# method='pearson')

# Checking our work against the original correlation matrix
# round(cor(Kim_df),3)

```

The script below allows you to store the simulated data as a file on your computer. This is optional – the entire lesson can be worked with the simulated data.

If you prefer the .rds format, use this script (remove the hashtags). The .rds format has the advantage of preserving any formatting of variables. A disadvantage is that you cannot open these files outside of the R environment.

Script to save the data to your computer as an .rds file.

```
#saveRDS(dfKim, 'dfKim.rds')
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
#dfKim<- readRDS('dfKim.rds')
```

If you prefer the .csv format (think “Excel lite”) use this script (remove the hashtags). An advantage of the .csv format is that you can open the data outside of the R environment. A disadvantage is that it may not retain any formatting of variables

Script to save the data to your computer as a .csv file.

```
# write.table(dfKim, file = 'dfKim.csv', sep = ',', col.names=TRUE,
# row.names=FALSE)
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfKim<- read.csv ('dfKim.csv', header = TRUE)
```

9.6 Scrubbing, Scoring, and Data Diagnostics

Because the focus of this lesson is on the specific topic of establishing a measurement model for SEM and have used simulated data, we can skip many of the steps in scrubbing, scoring and data diagnostics. If this were real, raw, data, it would be important to [scrub](#), if needed [score](#), and conduct [data diagnostics](#) to evaluate the suitability of the data for the proposes anlayses.

9.7 Specifying the Measurement Model in *lavaan*

SEM in *lavaan* requires fluency with the R script:

- Latent variables (factors) must be *defined* by their manifest or latent indicators.
 - the special operator ($=\sim$, *is measured/defined by*) is used for this
 - Example: $f1 =\sim y1 + y2 + y3$
- Regression equations use the single tilda (\sim , *is regressed on*)
 - place DV (y) on left of operator
 - place IVs, separate by + on the right
 - Example: $y \sim f1 + f2 + x1 + x2$
 - * f is a latent variable in this example
 - * y , $x1$, and $x2$ are observed variables in this example
 - An asterisk can affix a label in subsequent calculations and in interpreting output
- 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: $f1 \sim\sim f2 *Intercepts (\sim 1)$ 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 models. Readibility of model syntax is improved by:

- splitting formulas over multiple lines

- using blank lines within single quote
- labeling with the hashtag

```

init_msmt_mod <- "
    ##measurement model
    REMS =~ Inf32 + Inf38 + Inf21 + Inf17 + Inf9 + Inf36 + Inf5 + Inf22 + SClass6 + SClass7
    CMI =~ cmi1 + cmi2 + cmi3 + cmi4 + cmi5 + cmi6 + cmi7 + cmi8 + cmi9 + cmi10 + cmi11 + cmi12
    PWB =~ pwb1 + pwb2 + pwb3 + pwb4 + pwb5 + pwb6 + pwb7 + pwb8 + pwb9 + pwb10 + pwb11

    # Covariances
    REMS ~~ CMI
    REMS ~~ PWB
    CMI ~~ PWB
    "


set.seed(230916)
init_msmt_fit <- lavaan::cfa(init_msmt_mod, data = dfKim)
# you can add missing = 'fiml' to the code; I deleted it because it
# was really slowing down the run
init_msmt_fit_sum <- lavaan::summary(init_msmt_fit, fit.measures = TRUE,
    standardized = TRUE)
init_msmt_fit_sum

## lavaan 0.6.16 ended normally after 118 iterations
##
##      Estimator                      ML
##      Optimization method            NLMINB
##      Number of model parameters     209
##      Number of observations        156
##
## Model Test User Model:
##
##      Test statistic                7271.391
##      Degrees of freedom              5147
##      P-value (Chi-square)           0.000
##
## Model Test Baseline Model:
##
##      Test statistic                13555.967
##      Degrees of freedom              5253
##      P-value                         0.000
##
## User Model versus Baseline Model:

```

```

##  

## Comparative Fit Index (CFI) 0.744  

## Tucker-Lewis Index (TLI) 0.739  

##  

## Loglikelihood and Information Criteria:  

##  

## Loglikelihood user model (H0) -15294.957  

## Loglikelihood unrestricted model (H1) -11659.262  

##  

## Akaike (AIC) 31007.915  

## Bayesian (BIC) 31645.335  

## Sample-size adjusted Bayesian (SABIC) 30983.784  

##  

## Root Mean Square Error of Approximation:  

##  

## RMSEA 0.051  

## 90 Percent confidence interval - lower 0.049  

## 90 Percent confidence interval - upper 0.054  

## P-value H_0: RMSEA <= 0.050 0.193  

## P-value H_0: RMSEA >= 0.080 0.000  

##  

## Standardized Root Mean Square Residual:  

##  

## SRMR 0.061  

##  

## 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  

## REMS =~  

## Inf32 1.000 0.282 0.572  

## Inf38 1.063 0.169 6.289 0.000 0.300 0.606  

## Inf21 0.894 0.151 5.935 0.000 0.252 0.560  

## Inf17 0.970 0.165 5.871 0.000 0.273 0.552  

## Inf9 1.010 0.163 6.213 0.000 0.285 0.596  

## Inf36 1.026 0.169 6.079 0.000 0.289 0.579  

## Inf5 0.907 0.156 5.811 0.000 0.256 0.545  

## Inf22 0.945 0.165 5.711 0.000 0.266 0.533  

## SClass6 1.146 0.173 6.631 0.000 0.323 0.654  

## SClass31 1.039 0.169 6.133 0.000 0.293 0.586  

## SClass8 0.893 0.150 5.959 0.000 0.252 0.563  

## SClass40 1.036 0.168 6.184 0.000 0.292 0.592  

## SClass2 0.963 0.164 5.855 0.000 0.271 0.550  

## SClass34 0.949 0.161 5.882 0.000 0.268 0.554

```

##	SClass11	0.867	0.162	5.345	0.000	0.244	0.489
##	mInv27	1.045	0.168	6.232	0.000	0.294	0.599
##	mInv30	0.970	0.158	6.133	0.000	0.273	0.586
##	mInv39	1.143	0.174	6.573	0.000	0.322	0.645
##	mInv7	0.946	0.161	5.868	0.000	0.267	0.552
##	mInv26	1.122	0.173	6.483	0.000	0.316	0.633
##	mInv33	1.080	0.166	6.516	0.000	0.305	0.637
##	mInv4	0.744	0.147	5.060	0.000	0.210	0.457
##	mInv14	0.910	0.150	6.075	0.000	0.256	0.578
##	mInv10	0.817	0.159	5.130	0.000	0.230	0.465
##	Exot3	1.164	0.174	6.673	0.000	0.328	0.660
##	Exot29	1.092	0.169	6.455	0.000	0.308	0.629
##	Exot45	1.099	0.172	6.389	0.000	0.310	0.620
##	Exot35	1.103	0.168	6.559	0.000	0.311	0.643
##	Exot42	1.000	0.167	5.998	0.000	0.282	0.568
##	Exot23	0.862	0.154	5.589	0.000	0.243	0.518
##	Exot13	0.761	0.150	5.079	0.000	0.214	0.459
##	Exot20	0.805	0.158	5.087	0.000	0.227	0.460
##	Exot43	0.671	0.139	4.840	0.000	0.189	0.433
##	mEnv37	1.052	0.164	6.412	0.000	0.296	0.623
##	mEnv24	1.248	0.178	7.002	0.000	0.352	0.709
##	mEnv19	1.186	0.176	6.757	0.000	0.334	0.672
##	mEnv28	0.931	0.164	5.671	0.000	0.262	0.528
##	mEnv18	1.068	0.171	6.258	0.000	0.301	0.602
##	mEnv41	0.972	0.166	5.861	0.000	0.274	0.551
##	mEnv12	0.870	0.162	5.373	0.000	0.245	0.493
##	mWork25	1.111	0.172	6.461	0.000	0.313	0.630
##	mWork15	1.146	0.170	6.747	0.000	0.323	0.671
##	mWork1	1.065	0.170	6.260	0.000	0.300	0.602
##	mWork16	0.932	0.165	5.652	0.000	0.263	0.525
##	mWork44	0.952	0.165	5.773	0.000	0.268	0.540
##	CMI =~						
##	cmi1	1.000				0.767	0.654
##	cmi2	0.945	0.128	7.373	0.000	0.725	0.639
##	cmi3	1.006	0.126	7.995	0.000	0.772	0.702
##	cmi4	0.979	0.129	7.618	0.000	0.751	0.664
##	cmi5	0.958	0.131	7.339	0.000	0.735	0.636
##	cmi6	0.914	0.123	7.459	0.000	0.701	0.648
##	cmi7	1.003	0.136	7.380	0.000	0.769	0.640
##	cmi8	1.083	0.140	7.739	0.000	0.831	0.676
##	cmi9	0.953	0.133	7.191	0.000	0.731	0.621
##	cmi10	0.993	0.129	7.711	0.000	0.762	0.673
##	cmi11	0.990	0.122	8.083	0.000	0.759	0.711
##	cmi12	1.089	0.142	7.646	0.000	0.836	0.666
##	cmi13	1.066	0.144	7.403	0.000	0.818	0.642
##	cmi14	1.018	0.137	7.418	0.000	0.781	0.644
##	cmi15	0.865	0.121	7.154	0.000	0.663	0.618
##	cmi16	0.971	0.138	7.032	0.000	0.745	0.606

```

##   cmi17      1.102    0.146    7.522    0.000    0.846    0.654
##   cmi18      1.042    0.138    7.557    0.000    0.800    0.658
##   cmi19      0.940    0.130    7.240    0.000    0.722    0.626
##   cmi20      0.835    0.118    7.085    0.000    0.641    0.611
##   cmi21      0.813    0.111    7.330    0.000    0.623    0.635
##   cmi22      0.991    0.128    7.723    0.000    0.760    0.674
##   cmi23      0.970    0.143    6.782    0.000    0.745    0.582
##   cmi24      0.952    0.129    7.376    0.000    0.730    0.640
##   cmi25      1.053    0.142    7.433    0.000    0.808    0.645
##   cmi26      0.811    0.126    6.455    0.000    0.622    0.551
##   cmi27      1.029    0.130    7.939    0.000    0.790    0.696
##   cmi28      0.995    0.127    7.810    0.000    0.764    0.683
##   cmi29      0.784    0.123    6.378    0.000    0.602    0.543
##   cmi30      0.993    0.131    7.608    0.000    0.762    0.663
##   cmi31      1.010    0.133    7.601    0.000    0.775    0.662
##   cmi32      1.051    0.131    8.036    0.000    0.806    0.706
##   cmi33      1.094    0.137    7.971    0.000    0.840    0.699
##   cmi34      1.035    0.138    7.492    0.000    0.794    0.651
##   cmi35      0.938    0.134    7.024    0.000    0.720    0.605
##   cmi36      0.842    0.125    6.730    0.000    0.646    0.577
##   cmi37      0.990    0.147    6.754    0.000    0.760    0.579
##   cmi38      1.129    0.147    7.663    0.000    0.866    0.668
##   cmi39      0.985    0.128    7.692    0.000    0.756    0.671
##   cmi40      1.181    0.145    8.124    0.000    0.906    0.715
##   cmi41      1.007    0.131    7.697    0.000    0.773    0.672
##   cmi42      1.082    0.139    7.775    0.000    0.830    0.679
##   cmi43      1.205    0.144    8.397    0.000    0.925    0.744
##   cmi44      0.880    0.118    7.435    0.000    0.675    0.645
##   cmi45      0.922    0.120    7.672    0.000    0.708    0.669
##   cmi46      0.926    0.137    6.778    0.000    0.711    0.581
##   cmi47      1.139    0.164    6.942    0.000    0.874    0.597
## PWB ==
##   pwb1      1.000          0.516    0.619
##   pwb2      1.070    0.158    6.752    0.000    0.551    0.697
##   pwb3      0.552    0.134    4.112    0.000    0.285    0.380
##   pwb4      0.607    0.127    4.766    0.000    0.313    0.449
##   pwb5      0.931    0.154    6.032    0.000    0.480    0.598
##   pwb6      0.722    0.132    5.476    0.000    0.372    0.529
##   pwb7      0.550    0.136    4.035    0.000    0.284    0.372
##   pwb8      0.708    0.137    5.161    0.000    0.365    0.493
##   pwb9      0.642    0.126    5.091    0.000    0.331    0.485
##   pwb10     1.006    0.167    6.017    0.000    0.519    0.596
##   pwb11     0.573    0.147    3.890    0.000    0.296    0.357
##
## Covariances:
##   REMS ~~~
##   CMI       Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   REMS      0.130    0.028    4.584    0.000    0.601    0.601

```

##	PWB	-0.081	0.019	-4.177	0.000	-0.559	-0.559
##	CMI ~~						
##	PWB	-0.225	0.051	-4.415	0.000	-0.568	-0.568
##							
##	Variances:						
##		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
##	.Inf32	0.164	0.019	8.653	0.000	0.164	0.673
##	.Inf38	0.154	0.018	8.618	0.000	0.154	0.632
##	.Inf21	0.139	0.016	8.663	0.000	0.139	0.686
##	.Inf17	0.170	0.020	8.670	0.000	0.170	0.695
##	.Inf9	0.147	0.017	8.629	0.000	0.147	0.645
##	.Inf36	0.166	0.019	8.646	0.000	0.166	0.665
##	.Inf5	0.155	0.018	8.676	0.000	0.155	0.703
##	.Inf22	0.179	0.021	8.686	0.000	0.179	0.716
##	.SClass6	0.140	0.016	8.557	0.000	0.140	0.573
##	.SClass31	0.164	0.019	8.640	0.000	0.164	0.657
##	.SClass8	0.136	0.016	8.661	0.000	0.136	0.683
##	.SClass40	0.158	0.018	8.633	0.000	0.158	0.649
##	.SClass2	0.169	0.020	8.672	0.000	0.169	0.697
##	.SClass34	0.162	0.019	8.669	0.000	0.162	0.693
##	.SClass11	0.190	0.022	8.716	0.000	0.190	0.761
##	.mInv27	0.155	0.018	8.626	0.000	0.155	0.642
##	.mInv30	0.143	0.017	8.640	0.000	0.143	0.657
##	.mInv39	0.145	0.017	8.569	0.000	0.145	0.583
##	.mInv7	0.162	0.019	8.671	0.000	0.162	0.695
##	.mInv26	0.150	0.017	8.586	0.000	0.150	0.600
##	.mInv33	0.136	0.016	8.580	0.000	0.136	0.594
##	.mInv4	0.166	0.019	8.734	0.000	0.166	0.791
##	.mInv14	0.131	0.015	8.647	0.000	0.131	0.666
##	.mInv10	0.192	0.022	8.730	0.000	0.192	0.784
##	.Exot3	0.140	0.016	8.548	0.000	0.140	0.565
##	.Exot29	0.145	0.017	8.591	0.000	0.145	0.604
##	.Exot45	0.154	0.018	8.602	0.000	0.154	0.616
##	.Exot35	0.137	0.016	8.572	0.000	0.137	0.586
##	.Exot42	0.166	0.019	8.656	0.000	0.166	0.677
##	.Exot23	0.161	0.019	8.697	0.000	0.161	0.732
##	.Exot13	0.172	0.020	8.733	0.000	0.172	0.789
##	.Exot20	0.192	0.022	8.733	0.000	0.192	0.788
##	.Exot43	0.155	0.018	8.747	0.000	0.155	0.812
##	.mEnv37	0.138	0.016	8.598	0.000	0.138	0.612
##	.mEnv24	0.122	0.014	8.459	0.000	0.122	0.497
##	.mEnv19	0.136	0.016	8.529	0.000	0.136	0.548
##	.mEnv28	0.178	0.021	8.690	0.000	0.178	0.721
##	.mEnv18	0.159	0.018	8.622	0.000	0.159	0.637
##	.mEnv41	0.172	0.020	8.671	0.000	0.172	0.696
##	.mEnv12	0.188	0.022	8.714	0.000	0.188	0.757
##	.mWork25	0.149	0.017	8.590	0.000	0.149	0.603
##	.mWork15	0.128	0.015	8.531	0.000	0.128	0.550

##	.mWork1	0.158	0.018	8.622	0.000	0.158	0.637
##	.mWork16	0.181	0.021	8.691	0.000	0.181	0.724
##	.mWork44	0.175	0.020	8.680	0.000	0.175	0.708
##	.cmi1	0.789	0.091	8.647	0.000	0.789	0.573
##	.cmi2	0.760	0.088	8.660	0.000	0.760	0.591
##	.cmi3	0.614	0.071	8.591	0.000	0.614	0.507
##	.cmi4	0.717	0.083	8.636	0.000	0.717	0.560
##	.cmi5	0.797	0.092	8.663	0.000	0.797	0.596
##	.cmi6	0.681	0.079	8.652	0.000	0.681	0.580
##	.cmi7	0.853	0.099	8.660	0.000	0.853	0.591
##	.cmi8	0.821	0.095	8.623	0.000	0.821	0.543
##	.cmi9	0.851	0.098	8.676	0.000	0.851	0.614
##	.cmi10	0.700	0.081	8.626	0.000	0.700	0.547
##	.cmi11	0.564	0.066	8.578	0.000	0.564	0.495
##	.cmi12	0.874	0.101	8.633	0.000	0.874	0.556
##	.cmi13	0.953	0.110	8.657	0.000	0.953	0.588
##	.cmi14	0.862	0.100	8.656	0.000	0.862	0.586
##	.cmi15	0.713	0.082	8.679	0.000	0.713	0.618
##	.cmi16	0.958	0.110	8.688	0.000	0.958	0.633
##	.cmi17	0.956	0.111	8.646	0.000	0.956	0.572
##	.cmi18	0.839	0.097	8.643	0.000	0.839	0.568
##	.cmi19	0.807	0.093	8.672	0.000	0.807	0.608
##	.cmi20	0.689	0.079	8.684	0.000	0.689	0.627
##	.cmi21	0.575	0.066	8.664	0.000	0.575	0.597
##	.cmi22	0.693	0.080	8.625	0.000	0.693	0.545
##	.cmi23	1.084	0.125	8.705	0.000	1.084	0.662
##	.cmi24	0.771	0.089	8.660	0.000	0.771	0.591
##	.cmi25	0.915	0.106	8.655	0.000	0.915	0.584
##	.cmi26	0.889	0.102	8.724	0.000	0.889	0.697
##	.cmi27	0.663	0.077	8.598	0.000	0.663	0.515
##	.cmi28	0.667	0.077	8.615	0.000	0.667	0.534
##	.cmi29	0.864	0.099	8.728	0.000	0.864	0.705
##	.cmi30	0.741	0.086	8.637	0.000	0.741	0.561
##	.cmi31	0.770	0.089	8.638	0.000	0.770	0.562
##	.cmi32	0.654	0.076	8.585	0.000	0.654	0.501
##	.cmi33	0.736	0.086	8.594	0.000	0.736	0.511
##	.cmi34	0.858	0.099	8.649	0.000	0.858	0.576
##	.cmi35	0.897	0.103	8.688	0.000	0.897	0.634
##	.cmi36	0.838	0.096	8.708	0.000	0.838	0.667
##	.cmi37	1.144	0.131	8.706	0.000	1.144	0.665
##	.cmi38	0.931	0.108	8.631	0.000	0.931	0.554
##	.cmi39	0.697	0.081	8.628	0.000	0.697	0.550
##	.cmi40	0.783	0.091	8.572	0.000	0.783	0.488
##	.cmi41	0.727	0.084	8.628	0.000	0.727	0.549
##	.cmi42	0.803	0.093	8.619	0.000	0.803	0.538
##	.cmi43	0.690	0.081	8.524	0.000	0.690	0.447
##	.cmi44	0.638	0.074	8.654	0.000	0.638	0.583
##	.cmi45	0.618	0.072	8.630	0.000	0.618	0.552

##	.cmi46	0.989	0.114	8.705	0.000	0.989	0.662
##	.cmi47	1.378	0.159	8.694	0.000	1.378	0.643
##	.pwb1	0.428	0.055	7.736	0.000	0.428	0.617
##	.pwb2	0.322	0.045	7.163	0.000	0.322	0.514
##	.pwb3	0.481	0.056	8.537	0.000	0.481	0.856
##	.pwb4	0.389	0.046	8.389	0.000	0.389	0.799
##	.pwb5	0.415	0.053	7.853	0.000	0.415	0.643
##	.pwb6	0.356	0.044	8.148	0.000	0.356	0.720
##	.pwb7	0.502	0.059	8.551	0.000	0.502	0.862
##	.pwb8	0.415	0.050	8.269	0.000	0.415	0.757
##	.pwb9	0.356	0.043	8.292	0.000	0.356	0.765
##	.pwb10	0.489	0.062	7.863	0.000	0.489	0.645
##	.pwb11	0.598	0.070	8.576	0.000	0.598	0.873
##	REMS	0.079	0.021	3.838	0.000	1.000	1.000
##	CMI	0.589	0.129	4.555	0.000	1.000	1.000
##	PWB	0.266	0.067	3.949	0.000	1.000	1.000

Evaluating our measurement model involves inspection of (a) the strength, significance, and direction of each of the indicators on their respective factors, (b) the global fit indices, and (c) the direction and degree to which the factors are correlated. While these three are the big buckets of evaluation, the *lavaan::cfalpha* output is rich with information.

If you wish to export the results for creation of tables, *tidySEM* has a number of functions that make this helpful. When you feed them to an object, the object can be downloaded as a .csv file

The `tidySEM::table_fit` function will display all of the global fit indices.

```
init_msmt_fitstats <- tidySEM:::table_fit(init_msmt_fit)
```

```
## Registered S3 method overwritten by 'tidySEM':  
##   method           from  
##   predict.MxModel OpenMx
```

init_msmt_fitstats

```
##   rmsea.notclose.h0      rmr rmr_nomean      srmr srmr_bentler
## 1          0.08 0.04292907 0.04292907 0.06061422 0.06061422
##   srmr_bentler_nomean    crmr crmr_nomean srmr_mplus srmr_mplus_nomean
## 1          0.06061422 0.0612056 0.0612056 0.06061422 0.06061422
##   cn_05   cn_01      gfi      agfi      pgfi      mfi      ecvi
## 1 115.028 116.5502 0.598818 0.5825275 0.5754511 0.001103858 49.29096
```

The `tidySEM::table_results` function produces all of the factor loadings, covariances, and variances,

```
init_msmt_pEsts <- tidySEM::table_results(init_msmt_fit, digits = 3, columns = NULL)
init_msmt_pEsts
```

	lhs	op	rhs	est	se	pval	confint	est_sig	est_std
## 1	REMS	=~	Inf32	1.000	0.000	<NA>	[1.000, 1.000]	1.000	0.572
## 2	REMS	=~	Inf38	1.063	0.169	0.000	[0.731, 1.394]	1.063***	0.606
## 3	REMS	=~	Inf21	0.894	0.151	0.000	[0.599, 1.190]	0.894***	0.560
## 4	REMS	=~	Inf17	0.970	0.165	0.000	[0.646, 1.294]	0.970***	0.552
## 5	REMS	=~	Inf9	1.010	0.163	0.000	[0.692, 1.329]	1.010***	0.596
## 6	REMS	=~	Inf36	1.026	0.169	0.000	[0.695, 1.357]	1.026***	0.579
## 7	REMS	=~	Inf5	0.907	0.156	0.000	[0.601, 1.213]	0.907***	0.545
## 8	REMS	=~	Inf22	0.945	0.165	0.000	[0.620, 1.269]	0.945***	0.533
## 9	REMS	=~	SClass6	1.146	0.173	0.000	[0.807, 1.485]	1.146***	0.654
## 10	REMS	=~	SClass31	1.039	0.169	0.000	[0.707, 1.371]	1.039***	0.586
## 11	REMS	=~	SClass8	0.893	0.150	0.000	[0.599, 1.187]	0.893***	0.563
## 12	REMS	=~	SClass40	1.036	0.168	0.000	[0.708, 1.364]	1.036***	0.592
## 13	REMS	=~	SClass2	0.963	0.164	0.000	[0.640, 1.285]	0.963***	0.550
## 14	REMS	=~	SClass34	0.949	0.161	0.000	[0.633, 1.266]	0.949***	0.554
## 15	REMS	=~	SClass11	0.867	0.162	0.000	[0.549, 1.185]	0.867***	0.489
## 16	REMS	=~	mInv27	1.045	0.168	0.000	[0.716, 1.373]	1.045***	0.599
## 17	REMS	=~	mInv30	0.970	0.158	0.000	[0.660, 1.280]	0.970***	0.586
## 18	REMS	=~	mInv39	1.143	0.174	0.000	[0.802, 1.483]	1.143***	0.645
## 19	REMS	=~	mInv7	0.946	0.161	0.000	[0.630, 1.262]	0.946***	0.552
## 20	REMS	=~	mInv26	1.122	0.173	0.000	[0.782, 1.461]	1.122***	0.633
## 21	REMS	=~	mInv33	1.080	0.166	0.000	[0.755, 1.405]	1.080***	0.637
## 22	REMS	=~	mInv4	0.744	0.147	0.000	[0.456, 1.033]	0.744***	0.457
## 23	REMS	=~	mInv14	0.910	0.150	0.000	[0.616, 1.203]	0.910***	0.578
## 24	REMS	=~	mInv10	0.817	0.159	0.000	[0.505, 1.129]	0.817***	0.465
## 25	REMS	=~	Exot3	1.164	0.174	0.000	[0.822, 1.506]	1.164***	0.660
## 26	REMS	=~	Exot29	1.092	0.169	0.000	[0.760, 1.423]	1.092***	0.629
## 27	REMS	=~	Exot45	1.099	0.172	0.000	[0.762, 1.437]	1.099***	0.620
## 28	REMS	=~	Exot35	1.103	0.168	0.000	[0.774, 1.433]	1.103***	0.643
## 29	REMS	=~	Exot42	1.000	0.167	0.000	[0.673, 1.326]	1.000***	0.568
## 30	REMS	=~	Exot23	0.862	0.154	0.000	[0.560, 1.164]	0.862***	0.518
## 31	REMS	=~	Exot13	0.761	0.150	0.000	[0.467, 1.054]	0.761***	0.459
## 32	REMS	=~	Exot20	0.805	0.158	0.000	[0.495, 1.115]	0.805***	0.460
## 33	REMS	=~	Exot43	0.671	0.139	0.000	[0.399, 0.943]	0.671***	0.433
## 34	REMS	=~	mEnv37	1.052	0.164	0.000	[0.730, 1.373]	1.052***	0.623

## 35	REMS ==	mEnv24	1.248	0.178	0.000	[0.898, 1.597]	1.248***	0.709
## 36	REMS ==	mEnv19	1.186	0.176	0.000	[0.842, 1.530]	1.186***	0.672
## 37	REMS ==	mEnv28	0.931	0.164	0.000	[0.609, 1.253]	0.931***	0.528
## 38	REMS ==	mEnv18	1.068	0.171	0.000	[0.734, 1.403]	1.068***	0.602
## 39	REMS ==	mEnv41	0.972	0.166	0.000	[0.647, 1.297]	0.972***	0.551
## 40	REMS ==	mEnv12	0.870	0.162	0.000	[0.553, 1.188]	0.870***	0.493
## 41	REMS ==	mWork25	1.111	0.172	0.000	[0.774, 1.448]	1.111***	0.630
## 42	REMS ==	mWork15	1.146	0.170	0.000	[0.813, 1.478]	1.146***	0.671
## 43	REMS ==	mWork1	1.065	0.170	0.000	[0.732, 1.399]	1.065***	0.602
## 44	REMS ==	mWork16	0.932	0.165	0.000	[0.609, 1.255]	0.932***	0.525
## 45	REMS ==	mWork44	0.952	0.165	0.000	[0.629, 1.275]	0.952***	0.540
## 46	CMI ==	cmi1	1.000	0.000	<NA>	[1.000, 1.000]	1.000	0.654
## 47	CMI ==	cmi2	0.945	0.128	0.000	[0.694, 1.196]	0.945***	0.639
## 48	CMI ==	cmi3	1.006	0.126	0.000	[0.760, 1.253]	1.006***	0.702
## 49	CMI ==	cmi4	0.979	0.129	0.000	[0.727, 1.231]	0.979***	0.664
## 50	CMI ==	cmi5	0.958	0.131	0.000	[0.703, 1.214]	0.958***	0.636
## 51	CMI ==	cmi6	0.914	0.123	0.000	[0.674, 1.154]	0.914***	0.648
## 52	CMI ==	cmi7	1.003	0.136	0.000	[0.736, 1.269]	1.003***	0.640
## 53	CMI ==	cmi8	1.083	0.140	0.000	[0.809, 1.358]	1.083***	0.676
## 54	CMI ==	cmi9	0.953	0.133	0.000	[0.693, 1.213]	0.953***	0.621
## 55	CMI ==	cmi10	0.993	0.129	0.000	[0.740, 1.245]	0.993***	0.673
## 56	CMI ==	cmi11	0.990	0.122	0.000	[0.750, 1.230]	0.990***	0.711
## 57	CMI ==	cmi12	1.089	0.142	0.000	[0.810, 1.369]	1.089***	0.666
## 58	CMI ==	cmi13	1.066	0.144	0.000	[0.784, 1.348]	1.066***	0.642
## 59	CMI ==	cmi14	1.018	0.137	0.000	[0.749, 1.287]	1.018***	0.644
## 60	CMI ==	cmi15	0.865	0.121	0.000	[0.628, 1.101]	0.865***	0.618
## 61	CMI ==	cmi16	0.971	0.138	0.000	[0.700, 1.242]	0.971***	0.606
## 62	CMI ==	cmi17	1.102	0.146	0.000	[0.815, 1.389]	1.102***	0.654
## 63	CMI ==	cmi18	1.042	0.138	0.000	[0.772, 1.312]	1.042***	0.658
## 64	CMI ==	cmi19	0.940	0.130	0.000	[0.686, 1.195]	0.940***	0.626
## 65	CMI ==	cmi20	0.835	0.118	0.000	[0.604, 1.066]	0.835***	0.611
## 66	CMI ==	cmi21	0.813	0.111	0.000	[0.595, 1.030]	0.813***	0.635
## 67	CMI ==	cmi22	0.991	0.128	0.000	[0.739, 1.242]	0.991***	0.674
## 68	CMI ==	cmi23	0.970	0.143	0.000	[0.690, 1.251]	0.970***	0.582
## 69	CMI ==	cmi24	0.952	0.129	0.000	[0.699, 1.205]	0.952***	0.640
## 70	CMI ==	cmi25	1.053	0.142	0.000	[0.775, 1.330]	1.053***	0.645
## 71	CMI ==	cmi26	0.811	0.126	0.000	[0.565, 1.057]	0.811***	0.551
## 72	CMI ==	cmi27	1.029	0.130	0.000	[0.775, 1.283]	1.029***	0.696
## 73	CMI ==	cmi28	0.995	0.127	0.000	[0.746, 1.245]	0.995***	0.683
## 74	CMI ==	cmi29	0.784	0.123	0.000	[0.543, 1.026]	0.784***	0.543
## 75	CMI ==	cmi30	0.993	0.131	0.000	[0.737, 1.249]	0.993***	0.663
## 76	CMI ==	cmi31	1.010	0.133	0.000	[0.749, 1.270]	1.010***	0.662
## 77	CMI ==	cmi32	1.051	0.131	0.000	[0.794, 1.307]	1.051***	0.706
## 78	CMI ==	cmi33	1.094	0.137	0.000	[0.825, 1.363]	1.094***	0.699
## 79	CMI ==	cmi34	1.035	0.138	0.000	[0.764, 1.306]	1.035***	0.651
## 80	CMI ==	cmi35	0.938	0.134	0.000	[0.676, 1.200]	0.938***	0.605
## 81	CMI ==	cmi36	0.842	0.125	0.000	[0.597, 1.088]	0.842***	0.577
## 82	CMI ==	cmi37	0.990	0.147	0.000	[0.703, 1.277]	0.990***	0.579

## 83	CMI =~	cmi38	1.129	0.147	0.000	[0.840, 1.418]	1.129***	0.668
## 84	CMI =~	cmi39	0.985	0.128	0.000	[0.734, 1.236]	0.985***	0.671
## 85	CMI =~	cmi40	1.181	0.145	0.000	[0.896, 1.465]	1.181***	0.715
## 86	CMI =~	cmi41	1.007	0.131	0.000	[0.751, 1.264]	1.007***	0.672
## 87	CMI =~	cmi42	1.082	0.139	0.000	[0.809, 1.354]	1.082***	0.679
## 88	CMI =~	cmi43	1.205	0.144	0.000	[0.924, 1.486]	1.205***	0.744
## 89	CMI =~	cmi44	0.880	0.118	0.000	[0.648, 1.111]	0.880***	0.645
## 90	CMI =~	cmi45	0.922	0.120	0.000	[0.687, 1.158]	0.922***	0.669
## 91	CMI =~	cmi46	0.926	0.137	0.000	[0.658, 1.194]	0.926***	0.581
## 92	CMI =~	cmi47	1.139	0.164	0.000	[0.817, 1.461]	1.139***	0.597
## 93	PWB =~	pwb1	1.000	0.000	<NA>	[1.000, 1.000]	1.000	0.619
## 94	PWB =~	pwb2	1.070	0.158	0.000	[0.759, 1.380]	1.070***	0.697
## 95	PWB =~	pwb3	0.552	0.134	0.000	[0.289, 0.815]	0.552***	0.380
## 96	PWB =~	pwb4	0.607	0.127	0.000	[0.358, 0.857]	0.607***	0.449
## 97	PWB =~	pwb5	0.931	0.154	0.000	[0.629, 1.234]	0.931***	0.598
## 98	PWB =~	pwb6	0.722	0.132	0.000	[0.464, 0.980]	0.722***	0.529
## 99	PWB =~	pwb7	0.550	0.136	0.000	[0.283, 0.818]	0.550***	0.372
## 100	PWB =~	pwb8	0.708	0.137	0.000	[0.439, 0.977]	0.708***	0.493
## 101	PWB =~	pwb9	0.642	0.126	0.000	[0.395, 0.889]	0.642***	0.485
## 102	PWB =~	pwb10	1.006	0.167	0.000	[0.678, 1.334]	1.006***	0.596
## 103	PWB =~	pwb11	0.573	0.147	0.000	[0.285, 0.862]	0.573***	0.357
## 104	REMS ~~	CMI	0.130	0.028	0.000	[0.074, 0.186]	0.130***	0.601
## 105	REMS ~~	PWB	-0.081	0.019	0.000	[-0.119, -0.043]	-0.081***	-0.559
## 106	CMI ~~	PWB	-0.225	0.051	0.000	[-0.324, -0.125]	-0.225***	-0.568
## 107	Inf32 ~~	Inf32	0.164	0.019	0.000	[0.127, 0.201]	0.164***	0.673
## 108	Inf38 ~~	Inf38	0.154	0.018	0.000	[0.119, 0.189]	0.154***	0.632
## 109	Inf21 ~~	Inf21	0.139	0.016	0.000	[0.107, 0.170]	0.139***	0.686
## 110	Inf17 ~~	Inf17	0.170	0.020	0.000	[0.132, 0.209]	0.170***	0.695
## 111	Inf9 ~~	Inf9	0.147	0.017	0.000	[0.114, 0.181]	0.147***	0.645
## 112	Inf36 ~~	Inf36	0.166	0.019	0.000	[0.129, 0.204]	0.166***	0.665
## 113	Inf5 ~~	Inf5	0.155	0.018	0.000	[0.120, 0.190]	0.155***	0.703
## 114	Inf22 ~~	Inf22	0.179	0.021	0.000	[0.139, 0.219]	0.179***	0.716
## 115	SClass6 ~~	SClass6	0.140	0.016	0.000	[0.108, 0.172]	0.140***	0.573
## 116	SClass31 ~~	SClass31	0.164	0.019	0.000	[0.127, 0.202]	0.164***	0.657
## 117	SClass8 ~~	SClass8	0.136	0.016	0.000	[0.105, 0.167]	0.136***	0.683
## 118	SClass40 ~~	SClass40	0.158	0.018	0.000	[0.122, 0.194]	0.158***	0.649
## 119	SClass2 ~~	SClass2	0.169	0.020	0.000	[0.131, 0.208]	0.169***	0.697
## 120	SClass34 ~~	SClass34	0.162	0.019	0.000	[0.125, 0.199]	0.162***	0.693
## 121	SClass11 ~~	SClass11	0.190	0.022	0.000	[0.147, 0.233]	0.190***	0.761
## 122	mInv27 ~~	mInv27	0.155	0.018	0.000	[0.120, 0.191]	0.155***	0.642
## 123	mInv30 ~~	mInv30	0.143	0.017	0.000	[0.111, 0.176]	0.143***	0.657
## 124	mInv39 ~~	mInv39	0.145	0.017	0.000	[0.112, 0.178]	0.145***	0.583
## 125	mInv7 ~~	mInv7	0.162	0.019	0.000	[0.126, 0.199]	0.162***	0.695
## 126	mInv26 ~~	mInv26	0.150	0.017	0.000	[0.116, 0.184]	0.150***	0.600
## 127	mInv33 ~~	mInv33	0.136	0.016	0.000	[0.105, 0.166]	0.136***	0.594
## 128	mInv4 ~~	mInv4	0.166	0.019	0.000	[0.129, 0.204]	0.166***	0.791
## 129	mInv14 ~~	mInv14	0.131	0.015	0.000	[0.101, 0.161]	0.131***	0.666
## 130	mInv10 ~~	mInv10	0.192	0.022	0.000	[0.149, 0.235]	0.192***	0.784

## 131	Exot3	~~	Exot3	0.140	0.016	0.000	[0.108, 0.172]	0.140***	0.565
## 132	Exot29	~~	Exot29	0.145	0.017	0.000	[0.112, 0.178]	0.145***	0.604
## 133	Exot45	~~	Exot45	0.154	0.018	0.000	[0.119, 0.189]	0.154***	0.616
## 134	Exot35	~~	Exot35	0.137	0.016	0.000	[0.106, 0.168]	0.137***	0.586
## 135	Exot42	~~	Exot42	0.166	0.019	0.000	[0.129, 0.204]	0.166***	0.677
## 136	Exot23	~~	Exot23	0.161	0.019	0.000	[0.125, 0.197]	0.161***	0.732
## 137	Exot13	~~	Exot13	0.172	0.020	0.000	[0.133, 0.210]	0.172***	0.789
## 138	Exot20	~~	Exot20	0.192	0.022	0.000	[0.149, 0.235]	0.192***	0.788
## 139	Exot43	~~	Exot43	0.155	0.018	0.000	[0.120, 0.190]	0.155***	0.812
## 140	mEnv37	~~	mEnv37	0.138	0.016	0.000	[0.107, 0.170]	0.138***	0.612
## 141	mEnv24	~~	mEnv24	0.122	0.014	0.000	[0.094, 0.151]	0.122***	0.497
## 142	mEnv19	~~	mEnv19	0.136	0.016	0.000	[0.104, 0.167]	0.136***	0.548
## 143	mEnv28	~~	mEnv28	0.178	0.021	0.000	[0.138, 0.219]	0.178***	0.721
## 144	mEnv18	~~	mEnv18	0.159	0.018	0.000	[0.123, 0.196]	0.159***	0.637
## 145	mEnv41	~~	mEnv41	0.172	0.020	0.000	[0.133, 0.211]	0.172***	0.696
## 146	mEnv12	~~	mEnv12	0.188	0.022	0.000	[0.146, 0.230]	0.188***	0.757
## 147	mWork25	~~	mWork25	0.149	0.017	0.000	[0.115, 0.183]	0.149***	0.603
## 148	mWork15	~~	mWork15	0.128	0.015	0.000	[0.098, 0.157]	0.128***	0.550
## 149	mWork1	~~	mWork1	0.158	0.018	0.000	[0.122, 0.194]	0.158***	0.637
## 150	mWork16	~~	mWork16	0.181	0.021	0.000	[0.140, 0.222]	0.181***	0.724
## 151	mWork44	~~	mWork44	0.175	0.020	0.000	[0.135, 0.214]	0.175***	0.708
## 152	cmi1	~~	cmi1	0.789	0.091	0.000	[0.610, 0.968]	0.789***	0.573
## 153	cmi2	~~	cmi2	0.760	0.088	0.000	[0.588, 0.932]	0.760***	0.591
## 154	cmi3	~~	cmi3	0.614	0.071	0.000	[0.474, 0.754]	0.614***	0.507
## 155	cmi4	~~	cmi4	0.717	0.083	0.000	[0.555, 0.880]	0.717***	0.560
## 156	cmi5	~~	cmi5	0.797	0.092	0.000	[0.617, 0.977]	0.797***	0.596
## 157	cmi6	~~	cmi6	0.681	0.079	0.000	[0.526, 0.835]	0.681***	0.580
## 158	cmi7	~~	cmi7	0.853	0.099	0.000	[0.660, 1.047]	0.853***	0.591
## 159	cmi8	~~	cmi8	0.821	0.095	0.000	[0.635, 1.008]	0.821***	0.543
## 160	cmi9	~~	cmi9	0.851	0.098	0.000	[0.659, 1.043]	0.851***	0.614
## 161	cmi10	~~	cmi10	0.700	0.081	0.000	[0.541, 0.860]	0.700***	0.547
## 162	cmi11	~~	cmi11	0.564	0.066	0.000	[0.435, 0.693]	0.564***	0.495
## 163	cmi12	~~	cmi12	0.874	0.101	0.000	[0.676, 1.073]	0.874***	0.556
## 164	cmi13	~~	cmi13	0.953	0.110	0.000	[0.737, 1.169]	0.953***	0.588
## 165	cmi14	~~	cmi14	0.862	0.100	0.000	[0.667, 1.057]	0.862***	0.586
## 166	cmi15	~~	cmi15	0.713	0.082	0.000	[0.552, 0.874]	0.713***	0.618
## 167	cmi16	~~	cmi16	0.958	0.110	0.000	[0.742, 1.174]	0.958***	0.633
## 168	cmi17	~~	cmi17	0.956	0.111	0.000	[0.739, 1.173]	0.956***	0.572
## 169	cmi18	~~	cmi18	0.839	0.097	0.000	[0.649, 1.030]	0.839***	0.568
## 170	cmi19	~~	cmi19	0.807	0.093	0.000	[0.625, 0.990]	0.807***	0.608
## 171	cmi20	~~	cmi20	0.689	0.079	0.000	[0.534, 0.845]	0.689***	0.627
## 172	cmi21	~~	cmi21	0.575	0.066	0.000	[0.445, 0.705]	0.575***	0.597
## 173	cmi22	~~	cmi22	0.693	0.080	0.000	[0.536, 0.851]	0.693***	0.545
## 174	cmi23	~~	cmi23	1.084	0.125	0.000	[0.840, 1.328]	1.084***	0.662
## 175	cmi24	~~	cmi24	0.771	0.089	0.000	[0.596, 0.945]	0.771***	0.591
## 176	cmi25	~~	cmi25	0.915	0.106	0.000	[0.708, 1.122]	0.915***	0.584
## 177	cmi26	~~	cmi26	0.889	0.102	0.000	[0.690, 1.089]	0.889***	0.697
## 178	cmi27	~~	cmi27	0.663	0.077	0.000	[0.512, 0.814]	0.663***	0.515

```

## 179   cmi28 ~~   cmi28  0.667 0.077 0.000  [0.515, 0.819] 0.667*** 0.534
## 180   cmi29 ~~   cmi29  0.864 0.099 0.000  [0.670, 1.058] 0.864*** 0.705
## 181   cmi30 ~~   cmi30  0.741 0.086 0.000  [0.573, 0.910] 0.741*** 0.561
## 182   cmi31 ~~   cmi31  0.770 0.089 0.000  [0.595, 0.945] 0.770*** 0.562
## 183   cmi32 ~~   cmi32  0.654 0.076 0.000  [0.504, 0.803] 0.654*** 0.501
## 184   cmi33 ~~   cmi33  0.736 0.086 0.000  [0.568, 0.904] 0.736*** 0.511
## 185   cmi34 ~~   cmi34  0.858 0.099 0.000  [0.663, 1.052] 0.858*** 0.576
## 186   cmi35 ~~   cmi35  0.897 0.103 0.000  [0.695, 1.100] 0.897*** 0.634
## 187   cmi36 ~~   cmi36  0.838 0.096 0.000  [0.649, 1.027] 0.838*** 0.667
## 188   cmi37 ~~   cmi37  1.144 0.131 0.000  [0.886, 1.401] 1.144*** 0.665
## 189   cmi38 ~~   cmi38  0.931 0.108 0.000  [0.719, 1.142] 0.931*** 0.554
## 190   cmi39 ~~   cmi39  0.697 0.081 0.000  [0.539, 0.856] 0.697*** 0.550
## 191   cmi40 ~~   cmi40  0.783 0.091 0.000  [0.604, 0.963] 0.783*** 0.488
## 192   cmi41 ~~   cmi41  0.727 0.084 0.000  [0.562, 0.893] 0.727*** 0.549
## 193   cmi42 ~~   cmi42  0.803 0.093 0.000  [0.621, 0.986] 0.803*** 0.538
## 194   cmi43 ~~   cmi43  0.690 0.081 0.000  [0.532, 0.849] 0.690*** 0.447
## 195   cmi44 ~~   cmi44  0.638 0.074 0.000  [0.494, 0.783] 0.638*** 0.583
## 196   cmi45 ~~   cmi45  0.618 0.072 0.000  [0.478, 0.758] 0.618*** 0.552
## 197   cmi46 ~~   cmi46  0.989 0.114 0.000  [0.766, 1.211] 0.989*** 0.662
## 198   cmi47 ~~   cmi47  1.378 0.159 0.000  [1.068, 1.689] 1.378*** 0.643
## 199   pwb1 ~~   pwb1   0.428 0.055 0.000  [0.319, 0.536] 0.428*** 0.617
## 200   pwb2 ~~   pwb2   0.322 0.045 0.000  [0.234, 0.410] 0.322*** 0.514
## 201   pwb3 ~~   pwb3   0.481 0.056 0.000  [0.371, 0.591] 0.481*** 0.856
## 202   pwb4 ~~   pwb4   0.389 0.046 0.000  [0.298, 0.479] 0.389*** 0.799
## 203   pwb5 ~~   pwb5   0.415 0.053 0.000  [0.311, 0.518] 0.415*** 0.643
## 204   pwb6 ~~   pwb6   0.356 0.044 0.000  [0.270, 0.442] 0.356*** 0.720
## 205   pwb7 ~~   pwb7   0.502 0.059 0.000  [0.387, 0.617] 0.502*** 0.862
## 206   pwb8 ~~   pwb8   0.415 0.050 0.000  [0.317, 0.514] 0.415*** 0.757
## 207   pwb9 ~~   pwb9   0.356 0.043 0.000  [0.272, 0.441] 0.356*** 0.765
## 208   pwb10 ~~  pwb10  0.489 0.062 0.000  [0.367, 0.611] 0.489*** 0.645
## 209   pwb11 ~~  pwb11  0.598 0.070 0.000  [0.462, 0.735] 0.598*** 0.873
## 210   REMS ~~   REMS   0.079 0.021 0.000  [0.039, 0.120] 0.079*** 1.000
## 211   CMI ~~    CMI   0.589 0.129 0.000  [0.335, 0.842] 0.589*** 1.000
## 212   PWB ~~    PWB   0.266 0.067 0.000  [0.134, 0.398] 0.266*** 1.000

##      se_std pval_std      confint_std est_sig_std          label
## 1     0.056  0.000  [0.463, 0.681] 0.572***  REMS.BY.Inf32
## 2     0.052  0.000  [0.504, 0.709] 0.606***  REMS.BY.Inf38
## 3     0.057  0.000  [0.449, 0.671] 0.560***  REMS.BY.Inf21
## 4     0.057  0.000  [0.440, 0.665] 0.552***  REMS.BY.Inf17
## 5     0.053  0.000  [0.492, 0.701] 0.596***  REMS.BY.Inf9
## 6     0.055  0.000  [0.471, 0.686] 0.579***  REMS.BY.Inf36
## 7     0.058  0.000  [0.431, 0.659] 0.545***  REMS.BY.Inf5
## 8     0.059  0.000  [0.417, 0.648] 0.533***  REMS.BY.Inf22
## 9     0.048  0.000  [0.561, 0.747] 0.654***  REMS.BY.SClass6
## 10    0.054  0.000  [0.479, 0.692] 0.586***  REMS.BY.SClass31
## 11    0.056  0.000  [0.453, 0.674] 0.563***  REMS.BY.SClass8
## 12    0.054  0.000  [0.487, 0.698] 0.592***  REMS.BY.SClass40
## 13    0.058  0.000  [0.438, 0.663] 0.550***  REMS.BY.SClass2

```

## 14	0.057	0.000	[0.442, 0.666]	0.554***	REMS.BY.SClass34
## 15	0.063	0.000	[0.367, 0.612]	0.489***	REMS.BY.SClass11
## 16	0.053	0.000	[0.495, 0.703]	0.599***	REMS.BY.mInv27
## 17	0.054	0.000	[0.479, 0.692]	0.586***	REMS.BY.mInv30
## 18	0.048	0.000	[0.551, 0.740]	0.645***	REMS.BY.mInv39
## 19	0.057	0.000	[0.440, 0.664]	0.552***	REMS.BY.mInv7
## 20	0.050	0.000	[0.535, 0.730]	0.633***	REMS.BY.mInv26
## 21	0.049	0.000	[0.541, 0.734]	0.637***	REMS.BY.mInv33
## 22	0.065	0.000	[0.330, 0.585]	0.457***	REMS.BY.mInv4
## 23	0.055	0.000	[0.470, 0.686]	0.578***	REMS.BY.mInv14
## 24	0.064	0.000	[0.339, 0.591]	0.465***	REMS.BY.mInv10
## 25	0.047	0.000	[0.568, 0.752]	0.660***	REMS.BY.Exot3
## 26	0.050	0.000	[0.531, 0.727]	0.629***	REMS.BY.Exot29
## 27	0.051	0.000	[0.520, 0.720]	0.620***	REMS.BY.Exot45
## 28	0.049	0.000	[0.548, 0.739]	0.643***	REMS.BY.Exot35
## 29	0.056	0.000	[0.459, 0.678]	0.568***	REMS.BY.Exot42
## 30	0.060	0.000	[0.400, 0.636]	0.518***	REMS.BY.Exot23
## 31	0.065	0.000	[0.332, 0.586]	0.459***	REMS.BY.Exot13
## 32	0.065	0.000	[0.333, 0.587]	0.460***	REMS.BY.Exot20
## 33	0.067	0.000	[0.303, 0.564]	0.433***	REMS.BY.Exot43
## 34	0.051	0.000	[0.524, 0.722]	0.623***	REMS.BY.mEnv37
## 35	0.042	0.000	[0.628, 0.791]	0.709***	REMS.BY.mEnv24
## 36	0.046	0.000	[0.583, 0.762]	0.672***	REMS.BY.mEnv19
## 37	0.059	0.000	[0.411, 0.644]	0.528***	REMS.BY.mEnv28
## 38	0.053	0.000	[0.499, 0.705]	0.602***	REMS.BY.mEnv18
## 39	0.057	0.000	[0.438, 0.664]	0.551***	REMS.BY.mEnv41
## 40	0.062	0.000	[0.370, 0.615]	0.493***	REMS.BY.mEnv12
## 41	0.050	0.000	[0.532, 0.728]	0.630***	REMS.BY.mWork25
## 42	0.046	0.000	[0.581, 0.760]	0.671***	REMS.BY.mWork15
## 43	0.053	0.000	[0.499, 0.706]	0.602***	REMS.BY.mWork1
## 44	0.060	0.000	[0.408, 0.642]	0.525***	REMS.BY.mWork16
## 45	0.058	0.000	[0.426, 0.655]	0.540***	REMS.BY.mWork44
## 46	0.047	0.000	[0.561, 0.746]	0.654***	CMI.BY.cmi1
## 47	0.048	0.000	[0.544, 0.734]	0.639***	CMI.BY.cmi2
## 48	0.042	0.000	[0.620, 0.784]	0.702***	CMI.BY.cmi3
## 49	0.046	0.000	[0.574, 0.754]	0.664***	CMI.BY.cmi4
## 50	0.049	0.000	[0.540, 0.732]	0.636***	CMI.BY.cmi5
## 51	0.048	0.000	[0.554, 0.741]	0.648***	CMI.BY.cmi6
## 52	0.048	0.000	[0.545, 0.735]	0.640***	CMI.BY.cmi7
## 53	0.045	0.000	[0.588, 0.763]	0.676***	CMI.BY.cmi8
## 54	0.050	0.000	[0.523, 0.720]	0.621***	CMI.BY.cmi9
## 55	0.045	0.000	[0.585, 0.761]	0.673***	CMI.BY.cmi10
## 56	0.041	0.000	[0.631, 0.791]	0.711***	CMI.BY.cmi11
## 57	0.046	0.000	[0.577, 0.756]	0.666***	CMI.BY.cmi12
## 58	0.048	0.000	[0.548, 0.737]	0.642***	CMI.BY.cmi13
## 59	0.048	0.000	[0.550, 0.738]	0.644***	CMI.BY.cmi14
## 60	0.051	0.000	[0.518, 0.717]	0.618***	CMI.BY.cmi15
## 61	0.052	0.000	[0.504, 0.707]	0.606***	CMI.BY.cmi16

## 62	0.047	0.000	[0.562, 0.746]	0.654***	CMI.BY.cmi17
## 63	0.047	0.000	[0.566, 0.749]	0.658***	CMI.BY.cmi18
## 64	0.050	0.000	[0.529, 0.724]	0.626***	CMI.BY.cmi19
## 65	0.051	0.000	[0.510, 0.712]	0.611***	CMI.BY.cmi20
## 66	0.049	0.000	[0.539, 0.731]	0.635***	CMI.BY.cmi21
## 67	0.045	0.000	[0.586, 0.762]	0.674***	CMI.BY.cmi22
## 68	0.054	0.000	[0.476, 0.688]	0.582***	CMI.BY.cmi23
## 69	0.048	0.000	[0.545, 0.735]	0.640***	CMI.BY.cmi24
## 70	0.048	0.000	[0.551, 0.739]	0.645***	CMI.BY.cmi25
## 71	0.057	0.000	[0.439, 0.662]	0.551***	CMI.BY.cmi26
## 72	0.042	0.000	[0.613, 0.779]	0.696***	CMI.BY.cmi27
## 73	0.044	0.000	[0.597, 0.769]	0.683***	CMI.BY.cmi28
## 74	0.058	0.000	[0.431, 0.656]	0.543***	CMI.BY.cmi29
## 75	0.046	0.000	[0.572, 0.753]	0.663***	CMI.BY.cmi30
## 76	0.046	0.000	[0.571, 0.752]	0.662***	CMI.BY.cmi31
## 77	0.041	0.000	[0.625, 0.787]	0.706***	CMI.BY.cmi32
## 78	0.042	0.000	[0.617, 0.782]	0.699***	CMI.BY.cmi33
## 79	0.047	0.000	[0.558, 0.744]	0.651***	CMI.BY.cmi34
## 80	0.052	0.000	[0.503, 0.707]	0.605***	CMI.BY.cmi35
## 81	0.055	0.000	[0.470, 0.684]	0.577***	CMI.BY.cmi36
## 82	0.054	0.000	[0.473, 0.686]	0.579***	CMI.BY.cmi37
## 83	0.045	0.000	[0.579, 0.757]	0.668***	CMI.BY.cmi38
## 84	0.045	0.000	[0.583, 0.760]	0.671***	CMI.BY.cmi39
## 85	0.040	0.000	[0.636, 0.794]	0.715***	CMI.BY.cmi40
## 86	0.045	0.000	[0.583, 0.760]	0.672***	CMI.BY.cmi41
## 87	0.044	0.000	[0.593, 0.766]	0.679***	CMI.BY.cmi42
## 88	0.037	0.000	[0.671, 0.816]	0.744***	CMI.BY.cmi43
## 89	0.048	0.000	[0.552, 0.739]	0.645***	CMI.BY.cmi44
## 90	0.045	0.000	[0.580, 0.758]	0.669***	CMI.BY.cmi45
## 91	0.054	0.000	[0.475, 0.688]	0.581***	CMI.BY.cmi46
## 92	0.053	0.000	[0.494, 0.700]	0.597***	CMI.BY.cmi47
## 93	0.058	0.000	[0.505, 0.733]	0.619***	PWB.BY.pwb1
## 94	0.051	0.000	[0.597, 0.797]	0.697***	PWB.BY.pwb2
## 95	0.076	0.000	[0.230, 0.529]	0.380***	PWB.BY.pwb3
## 96	0.072	0.000	[0.308, 0.590]	0.449***	PWB.BY.pwb4
## 97	0.060	0.000	[0.479, 0.716]	0.598***	PWB.BY.pwb5
## 98	0.066	0.000	[0.400, 0.659]	0.529***	PWB.BY.pwb6
## 99	0.077	0.000	[0.221, 0.523]	0.372***	PWB.BY.pwb7
## 100	0.069	0.000	[0.358, 0.628]	0.493***	PWB.BY.pwb8
## 101	0.069	0.000	[0.349, 0.621]	0.485***	PWB.BY.pwb9
## 102	0.060	0.000	[0.477, 0.714]	0.596***	PWB.BY.pwb10
## 103	0.078	0.000	[0.205, 0.509]	0.357***	PWB.BY.pwb11
## 104	0.054	0.000	[0.495, 0.707]	0.601***	REMS.WITH.CMI
## 105	0.066	0.000	[-0.688, -0.430]	-0.559***	REMS.WITH.PWB
## 106	0.064	0.000	[-0.694, -0.442]	-0.568***	CMI.WITH.PWB
## 107	0.064	0.000	[0.549, 0.798]	0.673***	Variances.Inf32
## 108	0.063	0.000	[0.508, 0.757]	0.632***	Variances.Inf38
## 109	0.063	0.000	[0.562, 0.810]	0.686***	Variances.Inf21

```

## 110 0.063 0.000 [0.571, 0.819] 0.695*** Variances.Inf17
## 111 0.064 0.000 [0.520, 0.769] 0.645*** Variances.Inf9
## 112 0.064 0.000 [0.540, 0.790] 0.665*** Variances.Inf36
## 113 0.063 0.000 [0.579, 0.827] 0.703*** Variances.Inf5
## 114 0.063 0.000 [0.593, 0.840] 0.716*** Variances.Inf22
## 115 0.062 0.000 [0.451, 0.695] 0.573*** Variances.SClass6
## 116 0.064 0.000 [0.532, 0.782] 0.657*** Variances.SClass31
## 117 0.063 0.000 [0.558, 0.807] 0.683*** Variances.SClass8
## 118 0.064 0.000 [0.524, 0.774] 0.649*** Variances.SClass40
## 119 0.063 0.000 [0.573, 0.821] 0.697*** Variances.SClass2
## 120 0.063 0.000 [0.569, 0.818] 0.693*** Variances.SClass34
## 121 0.061 0.000 [0.641, 0.881] 0.761*** Variances.SClass11
## 122 0.064 0.000 [0.517, 0.766] 0.642*** Variances.mInv27
## 123 0.064 0.000 [0.532, 0.782] 0.657*** Variances.mInv30
## 124 0.063 0.000 [0.461, 0.706] 0.583*** Variances.mInv39
## 125 0.063 0.000 [0.571, 0.819] 0.695*** Variances.mInv7
## 126 0.063 0.000 [0.476, 0.723] 0.600*** Variances.mInv26
## 127 0.063 0.000 [0.471, 0.717] 0.594*** Variances.mInv33
## 128 0.059 0.000 [0.674, 0.907] 0.791*** Variances.mInv4
## 129 0.064 0.000 [0.541, 0.790] 0.666*** Variances.mInv14
## 130 0.060 0.000 [0.666, 0.901] 0.784*** Variances.mInv10
## 131 0.062 0.000 [0.443, 0.686] 0.565*** Variances.Exot3
## 132 0.063 0.000 [0.481, 0.728] 0.604*** Variances.Exot29
## 133 0.063 0.000 [0.492, 0.740] 0.616*** Variances.Exot45
## 134 0.063 0.000 [0.463, 0.709] 0.586*** Variances.Exot35
## 135 0.064 0.000 [0.553, 0.802] 0.677*** Variances.Exot42
## 136 0.062 0.000 [0.609, 0.854] 0.732*** Variances.Exot23
## 137 0.060 0.000 [0.672, 0.906] 0.789*** Variances.Exot13
## 138 0.060 0.000 [0.671, 0.905] 0.788*** Variances.Exot20
## 139 0.058 0.000 [0.699, 0.926] 0.812*** Variances.Exot43
## 140 0.063 0.000 [0.488, 0.736] 0.612*** Variances.mEnv37
## 141 0.059 0.000 [0.382, 0.613] 0.497*** Variances.mEnv24
## 142 0.061 0.000 [0.428, 0.668] 0.548*** Variances.mEnv19
## 143 0.063 0.000 [0.598, 0.844] 0.721*** Variances.mEnv28
## 144 0.064 0.000 [0.513, 0.762] 0.637*** Variances.mEnv18
## 145 0.063 0.000 [0.572, 0.820] 0.696*** Variances.mEnv41
## 146 0.061 0.000 [0.637, 0.878] 0.757*** Variances.mEnv12
## 147 0.063 0.000 [0.480, 0.727] 0.603*** Variances.mWork25
## 148 0.061 0.000 [0.430, 0.671] 0.550*** Variances.mWork15
## 149 0.064 0.000 [0.513, 0.762] 0.637*** Variances.mWork1
## 150 0.063 0.000 [0.601, 0.847] 0.724*** Variances.mWork16
## 151 0.063 0.000 [0.585, 0.832] 0.708*** Variances.mWork44
## 152 0.061 0.000 [0.452, 0.693] 0.573*** Variances.cmi1
## 153 0.062 0.000 [0.470, 0.713] 0.591*** Variances.cmi2
## 154 0.059 0.000 [0.392, 0.622] 0.507*** Variances.cmi3
## 155 0.061 0.000 [0.440, 0.679] 0.560*** Variances.cmi4
## 156 0.062 0.000 [0.474, 0.717] 0.596*** Variances.cmi5
## 157 0.062 0.000 [0.459, 0.701] 0.580*** Variances.cmi6

```

```

## 158 0.062 0.000 [0.469, 0.712] 0.591*** Variances.cmi7
## 159 0.060 0.000 [0.425, 0.661] 0.543*** Variances.cmi8
## 160 0.063 0.000 [0.491, 0.736] 0.614*** Variances.cmi9
## 161 0.061 0.000 [0.428, 0.666] 0.547*** Variances.cmi10
## 162 0.058 0.000 [0.381, 0.608] 0.495*** Variances.cmi11
## 163 0.061 0.000 [0.437, 0.675] 0.556*** Variances.cmi12
## 164 0.062 0.000 [0.466, 0.709] 0.588*** Variances.cmi13
## 165 0.062 0.000 [0.464, 0.707] 0.586*** Variances.cmi14
## 166 0.063 0.000 [0.496, 0.741] 0.618*** Variances.cmi15
## 167 0.063 0.000 [0.510, 0.756] 0.633*** Variances.cmi16
## 168 0.061 0.000 [0.452, 0.693] 0.572*** Variances.cmi17
## 169 0.061 0.000 [0.448, 0.688] 0.568*** Variances.cmi18
## 170 0.062 0.000 [0.486, 0.730] 0.608*** Variances.cmi19
## 171 0.063 0.000 [0.504, 0.750] 0.627*** Variances.cmi20
## 172 0.062 0.000 [0.475, 0.719] 0.597*** Variances.cmi21
## 173 0.060 0.000 [0.427, 0.664] 0.545*** Variances.cmi22
## 174 0.063 0.000 [0.538, 0.785] 0.662*** Variances.cmi23
## 175 0.062 0.000 [0.469, 0.712] 0.591*** Variances.cmi24
## 176 0.062 0.000 [0.463, 0.705] 0.584*** Variances.cmi25
## 177 0.063 0.000 [0.574, 0.820] 0.697*** Variances.cmi26
## 178 0.059 0.000 [0.400, 0.631] 0.515*** Variances.cmi27
## 179 0.060 0.000 [0.416, 0.651] 0.534*** Variances.cmi28
## 180 0.063 0.000 [0.582, 0.827] 0.705*** Variances.cmi29
## 181 0.061 0.000 [0.441, 0.681] 0.561*** Variances.cmi30
## 182 0.061 0.000 [0.442, 0.682] 0.562*** Variances.cmi31
## 183 0.058 0.000 [0.387, 0.616] 0.501*** Variances.cmi32
## 184 0.059 0.000 [0.396, 0.626] 0.511*** Variances.cmi33
## 185 0.062 0.000 [0.455, 0.697] 0.576*** Variances.cmi34
## 186 0.063 0.000 [0.511, 0.757] 0.634*** Variances.cmi35
## 187 0.063 0.000 [0.544, 0.791] 0.667*** Variances.cmi36
## 188 0.063 0.000 [0.541, 0.788] 0.665*** Variances.cmi37
## 189 0.061 0.000 [0.434, 0.673] 0.554*** Variances.cmi38
## 190 0.061 0.000 [0.431, 0.668] 0.550*** Variances.cmi39
## 191 0.058 0.000 [0.376, 0.601] 0.488*** Variances.cmi40
## 192 0.061 0.000 [0.430, 0.668] 0.549*** Variances.cmi41
## 193 0.060 0.000 [0.420, 0.656] 0.538*** Variances.cmi42
## 194 0.055 0.000 [0.339, 0.555] 0.447*** Variances.cmi43
## 195 0.062 0.000 [0.462, 0.705] 0.583*** Variances.cmi44
## 196 0.061 0.000 [0.433, 0.671] 0.552*** Variances.cmi45
## 197 0.063 0.000 [0.539, 0.785] 0.662*** Variances.cmi46
## 198 0.063 0.000 [0.520, 0.767] 0.643*** Variances.cmi47
## 199 0.072 0.000 [0.475, 0.758] 0.617*** Variances.pwb1
## 200 0.071 0.000 [0.375, 0.653] 0.514*** Variances.pwb2
## 201 0.058 0.000 [0.742, 0.970] 0.856*** Variances.pwb3
## 202 0.065 0.000 [0.672, 0.925] 0.799*** Variances.pwb4
## 203 0.072 0.000 [0.502, 0.784] 0.643*** Variances.pwb5
## 204 0.070 0.000 [0.583, 0.857] 0.720*** Variances.pwb6
## 205 0.057 0.000 [0.750, 0.974] 0.862*** Variances.pwb7

```

```
## 206 0.068 0.000 [0.624, 0.890] 0.757*** Variances.pwb8
## 207 0.067 0.000 [0.633, 0.897] 0.765*** Variances.pwb9
## 208 0.072 0.000 [0.504, 0.786] 0.645*** Variances.pwb10
## 209 0.055 0.000 [0.764, 0.981] 0.873*** Variances.pwb11
## 210 0.000 <NA> [1.000, 1.000] 1.000 Variances.REMS
## 211 0.000 <NA> [1.000, 1.000] 1.000 Variances.CMI
## 212 0.000 <NA> [1.000, 1.000] 1.000 Variances.PWB
```

The *tidySEM::table_cors* function will return a correlation matrix of the latent variables.

```
init_msmt_LVcorr <- tidySEM::table_cors(init_msmt_fit, digits = 3)
init_msmt_LVcorr
```

```
##          CMI          PWB          REMS
## CMI "1.000"      "-0.568***" "0.601***"
## PWB "-0.568***" "1.000"      "-0.559***"
## REMS "0.601***" "-0.559***" "1.000"
```

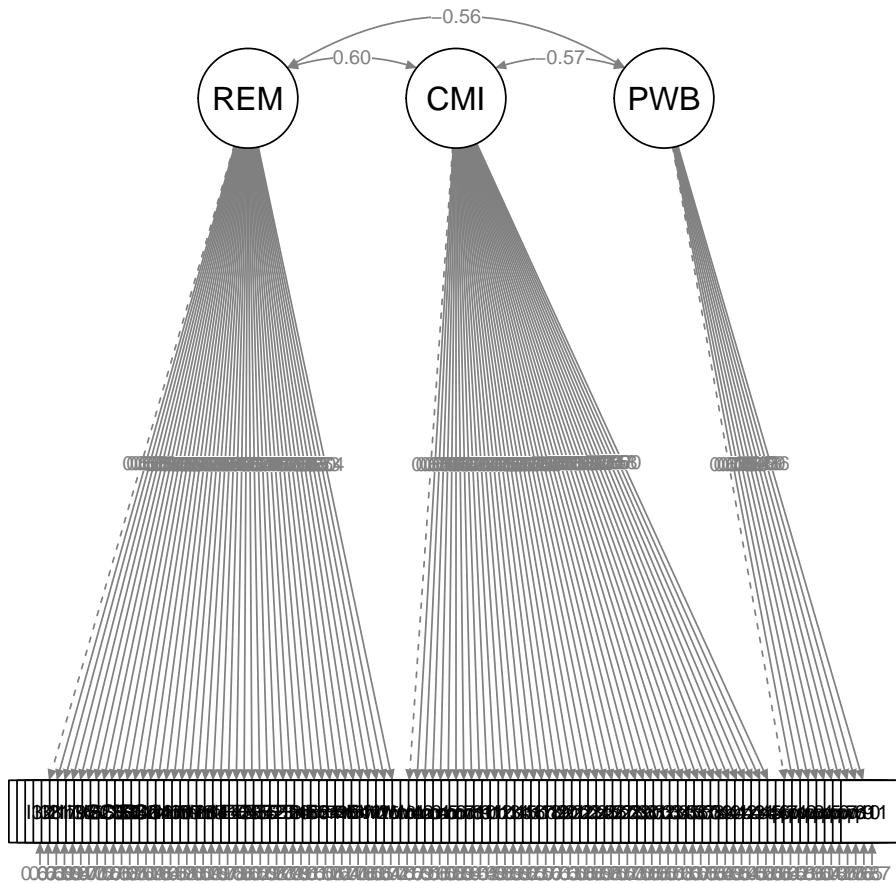
The *write.csv* function can export each of these objects to .csv files.

```
write.csv(init_msmt_fitstats, file = "init_msmt_fitstats.csv")
write.csv(init_msmt_pEsts, file = "init_msmt_pEsts.csv")
write.csv(init_msmt_LVcorr, file = "init_msmt_LVcorr.csv")
```

Before we interpret the output, let's also create a figure. This will help us conceptualize what we have just modeled and check our work. At this stage our model has a bazillion variables. Having tried both *tidySEM* and *semPlot*, I've gone with a quick *semPlot::semPaths* for this illustration. It at least allows us to see that we have allowed the latent variables to co-vary, that the first of each indicator variables was set to 1.0, and there were no unintentional cross-loadings.

This is not our structural prediction. Rather this is the pre-prediction. The fit of our structural model will, very likely be worse than this fit.

```
semPlot::semPaths(init_msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
  node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(5,
  5, 5, 5))
```



9.8 Interpreting the Output

Now that we've had 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 (0.6.16 that I'm using on 10/15/2023),
- maximum likelihood (ML) was used as the estimator,
- confirmation that the specification converged normally after 118 iterations,
- 156 cases were used in this analysis,

9.8.1 Global Fit Indices

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 can be more ambiguity and challenge to

interpreting these results. Consequently, researchers will often report a handful of measures that draw from *goodness* and *badness* of fit options.

- *goodness* of fit indices are those where values closer to 1.00 are better
- *badness* of fit indices are those where values closer to 0.00 are better

9.8.1.1 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 ($p > .05$) supports the model. This is backwards from our usual *reject-support test* approach. Kline [2016b] 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. The χ^2 is frequently criticized because:

- *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;
- low power (i.e., small sample sizes) makes it more likely that the model will be retained;
- CFA and SEM models require large samples and so the χ^2 is frequently statistically significant – which rejects the researchers' model;

For our initial measurement model CFA $\chi^2(5147) = 7271.391, p < .001$, this significant value is not what we want because it says that our specified model is different than the covariances in the model. At this stage of evaluating the *measurement model*, this is really critical information. Even though we have freed our latent variables to all covary with each other (which is like the natural state of the covariance matrix to which the model is being compared), the two are statistically significantly different.

9.8.1.2 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: 13555.967). On its own, this model is not useful to us. It is used, though, in comparisons of *incremental fit*.

9.8.1.3 Incremental Fit Indices (Located in the *User versus Baseline Models* section)

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. Because the two measures are so related, only one should be reported (I typically see the CFI).

CFI: compares the amount of departure from close fit for the researcher's model against that of the independence/baseline (null) model. When the User and Baseline fits are identical the 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 74% 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.

For our initial measurement model CFA, CFI = 0.744 and TLI = 0.739. While these predict around 74% better than the baseline/independence model, it does not come close to the standard of $\geq .95$.

9.8.1.4 Loglikelihood and Information Criteria

The **Akaike 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 31007.915 (AIC) and 31645.335 (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.

9.8.1.5 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 ideal. Watching the upper bound of the confidence interval is important to see that it isn't sneaking into the danger zone.

For our initial measurement model RMSEA = 0.051, 90% CI(0.049, 0.054). This value is within the accepted thresholds.

9.8.1.6 Standardized Root Mean Square Residual

The SRMR is an absolute fit index that is another *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 $SRMR \geq .10$. For our initial measurement model, $SRMR = 0.061$. This is within the thresholds of acceptability.

Hu and Bentler [1999] have suggested **combination rule** (which is somewhat contested) suggested that the SRMR be interpreted along with the CFI such that: $CFI \geq .95$ and $SRMR \leq .05$. Our initial measurement model does not pass this test: $CFI = 0.744$, $SRMR = 0.061$.

9.8.1.7 Factor Loadings

Let's inspect 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 Inf32 might seem incomplete – there is only a 1.000 and a value for the Std.lv. Recall that specifying items on a latent variable requires one item to be fixed to 1.000. This “sets the scale” of each latent variable. The default in *lavaan::cfa* and *lavaan::sem* is to assign the first of the items used to define the latent variable to this role. Coefficients that are fixed to 1.0 to scale a factor have no standard errors and therefore no significance test. If we looked at the `*semPlot::semPaths**` we can see that arrow line to each of the first indicators per latent variable is different than the others. This is a pictoral representation of setting the scaling on one of the indicator variables.

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 each of our factors (i.e., REMS, CMI, PWB) are all strong, positive, and statistically significant ($p < 0.001$) and positive.

9.8.1.8 Adequacy of the Initial Measurement Model

I've created a table that allows me to compare our results to the thresholds. I will report the chi-square, CFI, RMSEA, and SRMR. Researchers have different preferences (and different data may be better analyzed by certain indices), so you might find that an editor, professor, or reviewer will ask for something else.

Criteria	Our Results	Criteria met?
Factor loadings significant, strong, proper valence	all $p < 0.001$, lowest = .357(pwb)	Yes
Non-significant chi-square $CFI \geq .95$ (or at least .90)	$\chi^2(5147) = 7271.391, p < .001$ CFI = 0.744	No No
$RMSEA \leq .05$ (or < .08, at least < .10, also 90CI)	RMSEA = 0.051, 90CI[0.049, 0.054]	Yes
$SRMR \leq .08$ (at least < .10)	SRMR = 0.061	No
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	CFI = 0.744, SRMR = 0.061	No

Our initial measurement model that was defined by all items used as individual indicators for the latent variables and that freed the latent variables to covary had mixed results. While the factor loadings were significant, strong, and properly balanced and the RMSEA and SRMR were within acceptable limits, the chi-square was statistically significant and the CFI was well below 0.95: $\chi^2(5147) = 7271.391, p < .001$, $CFI = 0.744$, $RMSEA = 0.051$, $90CI[0.049, 0.054]$, $SRMR = 0.061$. The statistically significant chi-square is not what we want, but also not surprising. While the RMSEA and SRMR are within reasonable limits, the CFI is really substandard.

Concerned about poor model fit associated with the CFI, some researchers might choose to abandon a latent variable approach. After all, the [simple mediation](#) we conducted with observed/manifest variables earlier in this text was fine, right? Given the benefits that SEM offers, particularly around the ability to assess measurement error, I suggest that it is worth our time to consider the contribution of *parcelling* to see if we can improve the fit of our measurement model.

9.9 Parcelling

In the context of SEM, parcelling can reduce the complexity in latent variables by creating composites or *parcels* of observed variables. For latent variables with numerous indicators, *parcelling* can be an option that simplifies the model and can improve fit. A *parcel* is an aggregate-level indicator comprised of the sum (or average) of two or more items, responses, or behaviors. Parcels represent the *manifest (observed)* variables.

Little et al. [2002] outlined the pros and cons of parcelling and provided a practical guidelines for doing so. Although parcelling has become more common place, it remains controversial. Kline [2016b] is not a huge fan (see pp. 331-332) and Byrne [Byrne, 2016a] merely provided a general description of the process without demonstrating it in the examples we've provided. Little, remains a proponent and has updated the rationale and circumstances when parcelling is appropriate [Little et al., 2013].

9.9.1 The Pros and Cons of Parcelling

The Little et al. [2002] article reviewed the rather heated arguments for and against parcels on from three perspectives: philosophical, psychometric, modelers. Below I highlight each.

9.9.1.1 Philosophical Arguments

Empiricist-conservativists claim that modeled data should be (or be as close to possible) as the original individual responses of the research participants. Re-modeling data risks mis-representation and could be considered to be “cheating.” Further, they claim that parceling items “fundamentally undermines the objective empirical purpose of the techniques that have been developed to model multivariate data” (Little et al., 2002, p. 152).

Pragmatic-liberals suggest that representing each and every source of variance for each item is impossible. At best, we hope that our models represent the important common sources of variance across samples of items. Consequently, our goal is to build replicable models based on stable, meaningful indicators of core constructs. Correspondingly, pragmatic-liberals suggest that using parcels as the lowest level of data to be modeled is acceptable if the research project utilized a strict, rule-bound system of data collection and reporting. Further, it is essential for the researcher to describe what they have done and provide a rationale for doing so. As always, editors/reviewers have the right to reject the work and subsequent researchers can refute it.

9.9.1.2 Psychometric Arguments

Psychometricians point out that, compared to parcels, items have fewer, larger, and less equal intervals between scale points. Consequently, models built on item-level data tend to have lower reliability, lower communality, smaller ratio of common-to-unique factor variance, and increased likelihood of distributional violations. Additionally, because parcels permit factors to be defined with fewer constructs, they are preferred – especially when sample size is an issue. In short, models built on parceled data:

- Are more parsimonious (i.e., have fewer estimated parameters)
- Have fewer chances for residuals to be correlated or cross-loadings (b/c fewer indicators are needed and unique variances are smaller)
- Lead to reductions in various sources of sampling error
- Provide greater representation of the construct because aggregate scores are used
- Provide a more consistent and reliable representation of the construct because aggregate scores are used
- Reduce the problems associated when latent variables have a large number of indicators

Psychomtricians do offer one significant caution – parceling should only be used with unidimensional constructs and not with multidimensional ones. Practically, this could mean that the latent variable has a confirmatory factor analysis that supports a unidimensional, second order, or *g* scale in a bifactor structure. If unidimensionality is uncertain, preliminary psychometric evaluation could be in order. There seems to be some controversy about “how unidimensional” it should be. Little [2013] seems to acknowledge that multidimensional instruments are sometimes used.

Further, parceling is completely inappropriate when establishing the psychometric properties of an instrument. The use of parceling should be reserved for utilizing that instrument to test a theoretical model.

9.9.1.3 Modelers' Arguments

Modeling at the item level increases the likelihood that subsets of items will share specific sources of variance (which, themselves, represent latent variables). Because they are unlikely to be hypothesized by the researcher, they will lead to systematic sources of common variance that were not specified a priori. In contrast, parceling eliminates or at least reduces unwanted sources of variance and leads to better initial model fit and reduces the likelihood of misspecification. Further, because parceling improves the psychometric characteristics of items, solutions are more stable (i.e., requiring more iterations to converge, yielding relatively large standard errors of the measurement model, poorer fit).

Modelers have pointed out that item-level modeling inflates Type I error. Here's how:

- If we assume that 5% of all correlations are error ($p < .05$), a model with 3 constructs – each measured with 10 variables – would result in 22 spurious correlations.
- In contrast, a structural model with 3 constructs, each measured with 3 parcels each, would yield ~ 2 spurious correlations. The nature of which would be evidenced with a failure to replicate.

Modelers do point out issues related to model identification. Representing a latent variable with one or two items is possible, but is suboptimal because the latent variable is underidentified. A just-identified latent contains 3 indicators; 4 or more leads to an overidentified latent variable. In arguments made more than a decade after the 2002 article, Little and colleagues [2013] recommend using 3 indicators (which could be parcels) per construct.

Modelers who caution against modeling suggest that parcel-based models attempt to cancel out random and systematic error by aggregating across these errors. While the typical improvement in model fit is desirable, some argue is that it changes the reality of the data, and therefore misrepresents it. In-so-doing, it can hide mis-specification of the model. Some have argued that parceling should be reserved for theoretical work and is probably not appropriate for applied work when norms based on established measures are used.

9.9.2 Practical Procedures for Parcelling

There are several approaches to creating parcels. Byrne [2016a] distinguished between *purposive* and *random* approaches to parceling.

9.9.2.1 Purposive Approaches to Parcelling

Utilizing *subscale scores* as parcels is, perhaps, the most common (and intuitively appealing) example of a purposive approach to parceling. This approach can be appropriate in certain circumstances if (a) it is theoretically justified and (b) if the psychometric properties of the scales are sound. This approach can be problematic if (a) the factor loadings of the subscales are unequal and/or (b) if there aren't at least three subscales. While it is possible to have a factor with just two loadings, these are more likely to have improper solutions such as Heywood cases.

The *item-to-construct balance* is another purposive approach. The goal is to derive parcels that are equally balanced in terms of difficulty and discrimination (i.e., intercept and slope). If the researcher were creating a latent variable with three parcels (i.e., three indicators), the researcher would obtain and rank order the factor loadings. The three items with the highest loadings would be assigned to the three parcels first (let's call them A, B, C), the next three highest loadings would be added to the parcels in a reversed order (C, B, A), the next three reversed again (A, B, C), continuing back and forth until all items are assigned. In some conditions, parcels may have differential numbers of items in order to achieve a reasonable balance.

9.9.2.2 Random Approaches to Parceling

Random assignment is, perhaps, the most common way that parcels are created. Little et al. [2013] has recommended that each construct in an SEM model should be *just-identified* with three indicators per construct. Because this provides a precise definition of the construct, it is seen to be a super test. Thus, who engage in this practice will assign indicators, randomly (without replacement) to the three parcels. The result should be parcels with roughly equal common factor variance.

To illustrate the similarities and differences between approaches, I will evaluate two measurement models: one by using subscale scores and another with random assignment to three parcels per construct.

9.10 Parceling with Subscale Scores

Parceling with subscale scores means that you simply include the items that belong to each subscale in a parcel. Thus, it is often necessary to consult the journal articles and/or test manuals that provide information about the instrument. When simulating the data from the journal article, I was able to retrieve information about the psychometric development and evaluation of the REMS [Nadal, 2011], but not the CMI [Terrell and Terrell, 1981] or the MHI/PwB [Veit and Ware, 1983]. Thus, in working the subscale example, I will make some incorrect assumptions about the relations between the items and measures.

The 45 items of the REMS are divided between six subscales; thus, parcelling by subscale will result in six indicators for the REMS factor. In simulating the data, I was able to use the factor loadings from each item as it relates to each scale. To facilitate subsequent analyses, I provided abbreviations of the scale names. Practically speaking, creating parcels is to score the subscales. In the script below, I create first create concatenated lists of the variables. Second, I calculate mean scores if 80% of the items for each respondent are non-missing. You may recognize this as being consistent with Parent's [2013] available information analysis (AIA) approach to managing missingness. I have written more on the [AIA approach](#) in a lesson on [Scoring](#) data.

```
Inf_vars <- c("Inf32", "Inf38", "Inf21", "Inf17", "Inf9", "Inf36", "Inf5",
  "Inf22")
SClass_vars <- c("SClass8", "SClass40", "SClass2", "SClass34", "SClass11")
mInv_vars <- c("mInv27", "mInv39", "mInv7", "mInv26", "mInv33", "mInv4",
  "mInv14", "mInv10")
Exot_vars <- c("Exot3", "Exot29", "Exot45", "Exot35", "Exot42", "Exot23",
```

```

"Exot13", "Exot20", "Exot43")
mEnv_vars <- c("mEnv37", "mEnv24", "mEnv19", "mEnv28", "mEnv18", "mEnv41",
  "mEnv12")
mWork_vars <- c("mWork25", "mWork15", "mWork1", "mWork16", "mWork44")

dfKim$Inf_P <- sjstats::mean_n(dfKim[, Inf_vars], 0.8)
dfKim$SClass_P <- sjstats::mean_n(dfKim[, SClass_vars], 0.8)
dfKim$mInv_P <- sjstats::mean_n(dfKim[, mInv_vars], 0.8)
dfKim$Exot_P <- sjstats::mean_n(dfKim[, Exot_vars], 0.8)
dfKim$mEnv_P <- sjstats::mean_n(dfKim[, mEnv_vars], 0.8)
dfKim$mWork_P <- sjstats::mean_n(dfKim[, mWork_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting to periods in front of the variable
# list. One example is provided. dfKim$mWork_P <-
# sjstats::mean_n(dfKim[,, mWork_vars], .80)

```

We learn from Kim et al's description [Kim et al., 2017] that the CMI has four factors. Because I could not retrieve an article with original psychometrics, I was not able to provide factor correlations and my variable names do not reflect scale membership. Let's pretend that the items are in order of the scales. I will assign 12 items each to the first three scales and 11 items to the fourth scale.

```

IntRel_vars <- c("cmi1", "cmi2", "cmi3", "cmi4", "cmi5", "cmi6", "cmi7",
  "cmi8", "cmi9", "cmi10", "cmi11", "cmi12")
EdTrain_vars <- c("cmi13", "cmi14", "cmi15", "cmi16", "cmi17", "cmi18",
  "cmi19", "cmi20", "cmi21", "cmi22", "cmi23", "cmi24")
BusWrk_vars <- c("cmi25", "cmi26", "cmi27", "cmi28", "cmi29", "cmi30",
  "cmi31", "cmi32", "cmi33", "cmi34", "cmi35", "cmi36")
PolLaw_vars <- c("cmi37", "cmi38", "cmi39", "cmi40", "cmi41", "cmi42",
  "cmi43", "cmi44", "cmi45", "cmi46", "cmi47")

dfKim$IntRel <- sjstats::mean_n(dfKim[, IntRel_vars], 0.8)
dfKim$EdTrain <- sjstats::mean_n(dfKim[, EdTrain_vars], 0.8)
dfKim$BusWrk <- sjstats::mean_n(dfKim[, BusWrk_vars], 0.8)
dfKim$PolLaw <- sjstats::mean_n(dfKim[, PolLaw_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting to periods in front of the variable
# list. One example is provided. dfKim$PolLaw <-
# sjstats::mean_n(dfKim[,, PolLaw_vars], .80)

```

The description of the MHI only indicates that there are items on the PWB scale (and not that there are further subscales). This gives us some choice about how to divide the items. I would likely be inclined to randomly divide them across three scales.

The following code will provide random assignments.

```

set.seed(230916)
items <- c("pwb1", "pwb2", "pwb3", "pwb4", "pwb5", "pwb6", "pwb7", "pwb8",
  "pwb9", "pwb10", "pwb11")
parcels <- c("PWB_p1", "PWB_p_2", "PWB_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))

```

```

##   items  parcel
## 1 pwb7  PWB_p1
## 2 pwb9  PWB_p_2
## 3 pwb1  PWB_p3
## 4 pwb11 PWB_p1
## 5 pwb2  PWB_p_2
## 6 pwb3  PWB_p3
## 7 pwb4  PWB_p1
## 8 pwb10 PWB_p_2
## 9 pwb5  PWB_p3
## 10 pwb8 PWB_p1
## 11 pwb6 PWB_p_2

```

We can now create the parcels using the same scoring procedure as we did for the REMS and CMI instruments.

```

PWB_p1_vars <- c("pwb7", "pwb11", "pwb4", "pwb8")
PWB_p2_vars <- c("pwb9", "pwb2", "pwb10", "pwb2")
PWB_p3_vars <- c("pwb1", "pwb3", "pwb5")

dfKim$p1PWB <- sjstats::mean_n(dfKim[, PWB_p1_vars], 0.75)
dfKim$p2PWB <- sjstats::mean_n(dfKim[, PWB_p2_vars], 0.75)
dfKim$p3PWB <- sjstats::mean_n(dfKim[, PWB_p3_vars], 0.75)

# If the scoring code above does not work for you, try the format
# below which involves inserting periods in front of the variable
# list. One example is provided. dfKim$p3PWB <-
# sjstats::mean_n(dfKim[, ..PWB_p3_vars], .75)

```

Before we continue to respecifying the measurement model, let me point out that a downside of using subscales as parcels is that it Little et al's [2013] recommendation is that each latent variable be represented with a just-identified (i.e., 3-parcel) solution. Like our circumstance where the REMS has six subscales and the CMI has four, it is frequently the case where measures have differing numbers of solutions. When there are more than three parcels, the fit of the measurement model is likely to be worse than if there were three parcels per latent variable.

9.10.1 Measurement Model with Subscale Parcels

Let's respecify our measurement model with parcels created from subscale means.

```

subsc_msmt_mod <- "
  ##measurement model
  REMS =~ Inf_P + SClass_P + mInv_P + Exot_P + mEnv_P + mWork_P

  CMI =~ IntRel + EdTrain + BusWrk + PolLaw

  PWB =~ p1PWB + p2PWB + p3PWB

  # Covariances
  REMS ~~ CMI
  REMS ~~ PWB
  CMI ~~ PWB
  "

set.seed(230916)
subsc_msmt_fit <- lavaan:::cfa(subsc_msmt_mod, data = dfKim, missing = "fiml")
# , missing = 'fiml' #deleted this from the above code because it
# seemed to be slowing it down
subsc_msmt_fit_sum <- lavaan:::summary(subsc_msmt_fit, fit.measures = TRUE,
                                         standardized = TRUE)
subsc_msmt_fit_sum

## lavaan 0.6.16 ended normally after 127 iterations
##
##    Estimator                      ML
##    Optimization method            NLMINB
##    Number of model parameters      42
##    Number of observations          156
##    Number of missing patterns        1
##
## Model Test User Model:
##
##    Test statistic                 71.488
##    Degrees of freedom                  62
##    P-value (Chi-square)                0.192
##
## Model Test Baseline Model:
##
##    Test statistic                 2095.199
##    Degrees of freedom                  78
##    P-value                           0.000
##
## User Model versus Baseline Model:
##
##    Comparative Fit Index (CFI)       0.995

```

```

## Tucker-Lewis Index (TLI)          0.994
##
## Robust Comparative Fit Index (CFI) 0.995
## Robust Tucker-Lewis Index (TLI)    0.994
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)      -379.365
## Loglikelihood unrestricted model (H1) -343.621
##
## Akaike (AIC)                      842.731
## Bayesian (BIC)                    970.825
## Sample-size adjusted Bayesian (SABIC) 837.881
##
## Root Mean Square Error of Approximation:
##
## RMSEA                           0.031
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.060
## P-value H_0: RMSEA <= 0.050       0.839
## P-value H_0: RMSEA >= 0.080       0.001
##
## Robust RMSEA                     0.031
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.060
## P-value H_0: Robust RMSEA <= 0.050 0.839
## P-value H_0: Robust RMSEA >= 0.080 0.001
##
## Standardized Root Mean Square Residual:
##
## SRMR                            0.024
##
## Parameter Estimates:
##
## Standard errors                  Standard
## Information                       Observed
## Observed information based on   Hessian
##
## Latent Variables:
##                   Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## REMS =~
##   Inf_P        1.000
##   SClass_P     0.980    0.062 15.722  0.000  0.275  0.884
##   mInv_P      0.993    0.063 15.852  0.000  0.273  0.878
##   Exot_P       0.986    0.058 17.018  0.000  0.271  0.907
##   mEnv_P       1.073    0.063 17.131  0.000  0.295  0.903
##   mWork_P      1.066    0.069 15.337  0.000  0.293  0.864
## CMI =~

```

```

##      IntRel          1.000
##      EdTrain         0.972  0.040  24.289  0.000  0.739  0.939
##      BusWrk          0.974  0.040  24.475  0.000  0.740  0.939
##      PolLaw          1.056  0.040  26.144  0.000  0.802  0.953
## PWB =~
##      p1PWB          1.000
##      p2PWB          1.610  0.229  7.044  0.000  0.492  0.802
##      p3PWB          1.373  0.206  6.654  0.000  0.419  0.708
##
## Covariances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## REMS ~~
##      CMI            0.126  0.021  6.024  0.000  0.604  0.604
##      PWB           -0.047  0.010 -4.615  0.000 -0.554 -0.554
## CMI ~~
##      PWB           -0.130  0.027 -4.753  0.000 -0.560 -0.560
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .Inf_P          0.502  0.025 20.145  0.000  0.502  1.613
## .SClass_P        0.485  0.025 19.596  0.000  0.485  1.569
## .mInv_P          0.523  0.025 21.003  0.000  0.523  1.682
## .Exot_P          0.508  0.024 21.218  0.000  0.508  1.699
## .mEnv_P          0.504  0.026 19.255  0.000  0.504  1.542
## .mWork_P          0.462  0.027 16.985  0.000  0.462  1.360
## .IntRel          4.030  0.064 62.745  0.000  4.030  5.024
## .EdTrain         3.824  0.063 60.645  0.000  3.824  4.855
## .BusWrk          3.907  0.063 61.872  0.000  3.907  4.954
## .PolLaw          4.003  0.067 59.382  0.000  4.003  4.754
## .p1PWB          3.014  0.039 78.207  0.000  3.014  6.262
## .p2PWB          3.207  0.049 65.324  0.000  3.207  5.230
## .p3PWB          3.141  0.047 66.276  0.000  3.141  5.306
## REMS            0.000
## CMI             0.000
## PWB             0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .Inf_P          0.021  0.003  7.367  0.000  0.021  0.219
## .SClass_P        0.023  0.003  7.567  0.000  0.023  0.238
## .mInv_P          0.022  0.003  7.505  0.000  0.022  0.230
## .Exot_P          0.016  0.002  6.921  0.000  0.016  0.177
## .mEnv_P          0.020  0.003  7.002  0.000  0.020  0.184
## .mWork_P          0.029  0.004  7.651  0.000  0.029  0.254
## .IntRel          0.066  0.010  6.438  0.000  0.066  0.102
## .EdTrain         0.074  0.011  6.821  0.000  0.074  0.119
## .BusWrk          0.074  0.011  6.790  0.000  0.074  0.118
## .PolLaw          0.065  0.011  6.126  0.000  0.065  0.092

```

##	.p1PWB	0.139	0.019	7.313	0.000	0.139	0.598
##	.p2PWB	0.134	0.029	4.615	0.000	0.134	0.357
##	.p3PWB	0.175	0.028	6.316	0.000	0.175	0.498
##	REMS	0.076	0.011	6.991	0.000	1.000	1.000
##	CMI	0.578	0.073	7.935	0.000	1.000	1.000
##	PWB	0.093	0.024	3.957	0.000	1.000	1.000

As we look at the results we can easily see the benefits to the model. The factor loadings are strong, significant, and all scoring in the correct (positive) direction. The fit indices are much improved. Let's take a look according to the criteria we are using:

Criteria	Our Results	Criteria met?
Factor loadings significant, strong, proper valence	all $p < 0.001$, lowest = .634(pwb)	Yes
Non-significant chi-square	$\chi^2(62) = 71.488, p = 0.192$	Yes
$CFI \geq .95$ (or at least .90)	$CFI = 0.995$	Yes
$RMSEA \leq .05$ (or $< .08$, at least $< .10$, also 90CI)	$RMSEA = 0.031, 90CI[0.000, 0.060]$	Yes
$SRMR \leq .08$ (at least $< .10$)	$SRMR = 0.024$	Yes
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	$CFI = 0.995, SRMR = 0.024$	Yes

The measurement model created by parcels that represent subscales of the instruments (i.e., defined by prior psychometric evaluation) has dramatically improved. The chi-square is no longer statistically significant and the CFI is > 0.95 : $\chi^2(62) = 71.488, p = 0.192, CFI = 0.995, RMSEA = 0.031, 90CI[0.000, 0.060], SRMR = 0.024$. This might be a sufficient solution. An option that is more consistent with Little et al's [Little et al., 2013] recommendation for factors that are just-identified, is one that is constructed randomly.

9.10.2 Measurement Model with Just-Identified Random Parcels

In the prior example, we created three parcels through random assignment for the PWB scale. We can use those same parcels. We repeat those steps for the REMS and CMI scales. This code randomly assigns the 45 REMS items across the three parcels.

```
set.seed(230916)
items <- c("Inf32", "Inf38", "Inf21", "Inf17", "Inf9", "Inf36", "Inf5",
          "Inf22", "SClass6", "SClass31", "SClass8", "SClass40", "SClass2", "SClass34",
          "SClass11", "mInv27", "mInv30", "mInv39", "mInv7", "mInv26", "mInv33",
          "mInv4", "mInv14", "mInv10", "Exot3", "Exot29", "Exot45", "Exot35",
          "Exot42", "Exot23", "Exot13", "Exot20", "Exot43", "mEnv37", "mEnv24",
          "mEnv19", "mEnv28", "mEnv18", "mEnv41", "mEnv12", "mWork25", "mWork15",
          "mWork1", "mWork16", "mWork44")
parcels <- c("REMS_p1", "REMS_p2", "REMS_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))
```

```

##      items  parcel
## 1    mWork44 REMS_p1
## 2    mEnv41 REMS_p2
## 3    mWork25 REMS_p3
## 4    Exot20 REMS_p1
## 5   SClass11 REMS_p2
## 6     Inf38 REMS_p3
## 7   SClass8 REMS_p1
## 8     Inf5  REMS_p2
## 9     Inf17 REMS_p3
## 10   Exot43 REMS_p1
## 11   mEnv19 REMS_p2
## 12   Exot23 REMS_p3
## 13   mInv4  REMS_p1
## 14   mInv39 REMS_p2
## 15   mInv33 REMS_p3
## 16  SClass31 REMS_p1
## 17     Inf22 REMS_p2
## 18   mEnv28 REMS_p3
## 19  SClass40 REMS_p1
## 20   mEnv18 REMS_p2
## 21     Inf36 REMS_p3
## 22   mWork16 REMS_p1
## 23   mInv30 REMS_p2
## 24   mInv14 REMS_p3
## 25   Exot45 REMS_p1
## 26     Inf32 REMS_p2
## 27   mEnv37 REMS_p3
## 28   SClass2 REMS_p1
## 29  SClass34 REMS_p2
## 30     Inf21 REMS_p3
## 31   Exot3  REMS_p1
## 32   Exot29 REMS_p2
## 33   mEnv12 REMS_p3
## 34   mEnv24 REMS_p1
## 35   mInv27 REMS_p2
## 36   Exot42 REMS_p3
## 37   Exot35 REMS_p1
## 38   mInv10 REMS_p2
## 39   SClass6 REMS_p3
## 40   mInv7  REMS_p1
## 41   mWork1 REMS_p2
## 42     Inf9  REMS_p3
## 43   Exot13 REMS_p1
## 44   mWork15 REMS_p2
## 45   mInv26 REMS_p3

```

This code provides means for each of the three REMS parcels.

```

REMS_p1_vars <- c("mWork44", "Exot20", "SClass8", "Exot43", "mInv4", "SClass31",
  "SClass40", "mWork16", "Exot45", "SClass2", "Exot3", "mEnv24", "Exot35",
  "mInv7", "Exot13")
REMS_p2_vars <- c("mEnv41", "SClass11", "Inf5", "mEnv19", "mInv39", "Inf22",
  "mEnv18", "mInv30", "Inf32", "SClass34", "Exot29", "mInv27", "mInv10",
  "mWork1", "mWork15")
REMS_p3_vars <- c("mWork25", "Inf38", "Inf17", "Exot23", "mInv33", "mEnv28",
  "Inf36", "mInv14", "mEnv37", "Inf21", "mEnv12", "Exot42", "SClass6",
  "Inf9", "mInv26")

dfKim$p1REMS <- sjstats::mean_n(dfKim[, REMS_p1_vars], 0.8)
dfKim$p2REMS <- sjstats::mean_n(dfKim[, REMS_p2_vars], 0.8)
dfKim$p3REMS <- sjstats::mean_n(dfKim[, REMS_p3_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting periods in front of the variable
# list. One example is provided. dfKim$p3REMS <-
# sjstats::mean_n(dfKim[, ..REMS_p3_vars], .80)

```

We can repeat the process for the CMI. First, we assign the 47 CMI items to the three parcels.

```

set.seed(230916)
items <- c(IntRel_vars <- c("cmi1", "cmi2", "cmi3", "cmi4", "cmi5", "cmi6",
  "cmi7", "cmi8", "cmi9", "cmi10", "cmi11", "cmi12", "cmi13", "cmi14",
  "cmi15", "cmi16", "cmi17", "cmi18", "cmi19", "cmi20", "cmi21", "cmi22",
  "cmi23", "cmi24", "cmi25", "cmi26", "cmi27", "cmi28", "cmi29", "cmi30",
  "cmi31", "cmi32", "cmi33", "cmi34", "cmi35", "cmi36", "cmi37", "cmi38",
  "cmi39", "cmi40", "cmi41", "cmi42", "cmi43", "cmi44", "cmi45", "cmi46",
  "cmi47"))
parcels <- c("CMI_p1", "CMI_p2", "CMI_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))

##    items parcel
## 1  cmi45 CMI_p1
## 2  cmi39 CMI_p2
## 3  cmi41 CMI_p3
## 4  cmi32 CMI_p1
## 5  cmi15 CMI_p2
## 6  cmi2  CMI_p3
## 7  cmi11 CMI_p1
## 8  cmi7  CMI_p2
## 9  cmi4  CMI_p3
## 10 cmi33 CMI_p1
## 11 cmi38 CMI_p2
## 12 cmi30 CMI_p3
## 13 cmi22 CMI_p1
## 14 cmi18 CMI_p2

```

```

## 15 cmi46 CMI_p3
## 16 cmi12 CMI_p1
## 17 cmi40 CMI_p2
## 18 cmi6 CMI_p3
## 19 cmi26 CMI_p1
## 20 cmi47 CMI_p2
## 21 cmi17 CMI_p3
## 22 cmi25 CMI_p1
## 23 cmi27 CMI_p2
## 24 cmi23 CMI_p3
## 25 cmi44 CMI_p1
## 26 cmi1 CMI_p2
## 27 cmi10 CMI_p3
## 28 cmi13 CMI_p1
## 29 cmi14 CMI_p2
## 30 cmi3 CMI_p3
## 31 cmi31 CMI_p1
## 32 cmi42 CMI_p2
## 33 cmi29 CMI_p3
## 34 cmi34 CMI_p1
## 35 cmi8 CMI_p2
## 36 cmi28 CMI_p3
## 37 cmi37 CMI_p1
## 38 cmi19 CMI_p2
## 39 cmi21 CMI_p3
## 40 cmi43 CMI_p1
## 41 cmi20 CMI_p2
## 42 cmi35 CMI_p3
## 43 cmi24 CMI_p1
## 44 cmi5 CMI_p2
## 45 cmi36 CMI_p3
## 46 cmi9 CMI_p1
## 47 cmi16 CMI_p2

```

This code provides means for each of the three CMI parcels.

```

CMI_p1_vars <- c("cmi45", "cmi32", "cmi11", "cmi33", "cmi22", "cmi12",
                 "cmi26", "cmi25", "cmi44", "cmi13", "cmi31", "cmi34", "cmi37", "cmi43",
                 "cmi24", "cmi9")
CMI_p2_vars <- c("cmi39", "cmi15", "cmi7", "cmi38", "cmi18", "cmi40", "cmi47",
                 "cmi27", "cmi1", "cmi14", "cmi42", "cmi8", "cmi19", "cmi20", "cmi5",
                 "cmi16")
CMI_p3_vars <- c("cmi41", "cmi2", "cmi4", "cmi30", "cmi46", "cmi6", "cmi17",
                 "cmi23", "cmi10", "cmi3", "cmi29", "cmi28", "cmi21", "cmi35", "cmi36")

dfKim$p1CMI <- sjstats::mean_n(dfKim[, CMI_p1_vars], 0.8)
dfKim$p2CMI <- sjstats::mean_n(dfKim[, CMI_p2_vars], 0.8)

```

```
dfKim$p3CMI <- sjstats::mean_n(dfKim[, CMI_p3_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfKim$p3CMI <-
# sjstats::mean_n(dfKim[, ..CMI_p3_vars], .80)
```

9.10.2.1 Evaluating the Randomly Identified Measurement Model

We can now evaluate the measurement model that is defined by each scale's items that were randomly assigned to three parcels, each.

```
rp3_msmt_mod <- "
##measurement model
REMS =~ p1REMS + p2REMS + p3REMS

CMI =~ p1CMI + p2CMI + p3CMI

PWB =~ p1PWB + p2PWB + p3PWB

# Covariances
REMS ~~ CMI
REMS ~~ PWB
CMI ~~ PWB
"

set.seed(230916)
rp3_msmt_fit <- lavaan::cfa(rp3_msmt_mod, data = dfKim, missing = "fiml")
rp3_msmt_fit_sum <- lavaan::summary(rp3_msmt_fit, fit.measures = TRUE,
    standardized = TRUE)
rp3_msmt_fit_sum

## lavaan 0.6.16 ended normally after 106 iterations
##
##   Estimator                               ML
##   Optimization method                     NLMINB
##   Number of model parameters             30
##   Number of observations                 156
##   Number of missing patterns              1
##
## Model Test User Model:
##
##   Test statistic                          15.965
##   Degrees of freedom                      24
```

```

## P-value (Chi-square) 0.889
##
## Model Test Baseline Model:
##
## Test statistic 1366.804
## Degrees of freedom 36
## P-value 0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI) 1.000
## Tucker-Lewis Index (TLI) 1.009
##
## Robust Comparative Fit Index (CFI) 1.000
## Robust Tucker-Lewis Index (TLI) 1.009
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0) -368.149
## Loglikelihood unrestricted model (H1) -360.166
##
## Akaike (AIC) 796.297
## Bayesian (BIC) 887.793
## Sample-size adjusted Bayesian (SABIC) 792.834
##
## Root Mean Square Error of Approximation:
##
## RMSEA 0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.031
## P-value H_0: RMSEA <= 0.050 0.988
## P-value H_0: RMSEA >= 0.080 0.000
##
## Robust RMSEA 0.000
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.031
## P-value H_0: Robust RMSEA <= 0.050 0.988
## P-value H_0: Robust RMSEA >= 0.080 0.000
##
## Standardized Root Mean Square Residual:
##
## SRMR 0.017
##
## Parameter Estimates:
##
## Standard errors Standard
## Information Observed
## Observed information based on Hessian

```

```

## 
## Latent Variables:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## REMS    =~
##   p1REMS      1.000
##   p2REMS      1.055  0.044  23.770 0.000  0.286  0.946
##   p3REMS      1.031  0.046  22.399 0.000  0.280  0.929
## CMI     =~
##   p1CMI       1.000
##   p2CMI       1.006  0.035  28.780 0.000  0.778  0.957
##   p3CMI       0.937  0.034  27.373 0.000  0.724  0.948
## PWB    =~
##   p1PWB       1.000
##   p2PWB       1.606  0.228  7.055  0.000  0.491  0.801
##   p3PWB       1.372  0.206  6.662  0.000  0.420  0.709
## 
## Covariances:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## REMS ~~
##   CMI        0.128  0.021  6.199 0.000  0.608  0.608
##   PWB       -0.047  0.010 -4.712 0.000 -0.562 -0.562
## CMI ~~
##   PWB       -0.132  0.028 -4.763 0.000 -0.559 -0.559
## 
## Intercepts:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS    0.494  0.023 21.448 0.000  0.494  1.717
## .p2REMS    0.504  0.024 20.790 0.000  0.504  1.665
## .p3REMS    0.522  0.024 21.642 0.000  0.522  1.733
## .p1CMI     3.888  0.064 60.479 0.000  3.888  4.842
## .p2CMI     3.886  0.065 59.667 0.000  3.886  4.777
## .p3CMI     4.051  0.061 66.190 0.000  4.051  5.299
## .p1PWB     3.014  0.039 78.207 0.000  3.014  6.262
## .p2PWB     3.207  0.049 65.324 0.000  3.207  5.230
## .p3PWB     3.141  0.047 66.276 0.000  3.141  5.306
## REMS       0.000
## CMI        0.000
## PWB        0.000
## 
## Variances:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS    0.009  0.002  5.466 0.000  0.009  0.109
## .p2REMS    0.010  0.002  5.330 0.000  0.010  0.106
## .p3REMS    0.012  0.002  6.278 0.000  0.012  0.137
## .p1CMI     0.047  0.009  5.046 0.000  0.047  0.072
## .p2CMI     0.056  0.010  5.597 0.000  0.056  0.084
## .p3CMI     0.060  0.009  6.292 0.000  0.060  0.102
## .p1PWB     0.138  0.019  7.309 0.000  0.138  0.597

```

##	.p2PWB	0.135	0.029	4.646	0.000	0.135	0.358
##	.p3PWB	0.174	0.028	6.320	0.000	0.174	0.498
##	REMS	0.074	0.009	7.838	0.000	1.000	1.000
##	CMI	0.598	0.073	8.171	0.000	1.000	1.000
##	PWB	0.093	0.024	3.963	0.000	1.000	1.000

How do the results of the measurement model created by three parcels, created by random assignment of indicators to each, fare?

Criteria	Our Results	Criteria met?
Factor loadings significant, strong, proper valence	all $p < 0.001$, lowest = .635(pwb)	Yes
Non-significant chi-square	$\chi^2(24) = 15.965, p = 0.889$	Yes
$CFI \geq .95$ (or at least .90)	$CFI = 1.000$	Yes
$RMSEA \leq .05$ (or < .08, at least < .10, also 90CI)	$RMSEA = 0.000, 90CI[0.000, 0.031]$	Yes
$SRMR \leq .08$ (at least < .10)	$SRMR = 0.017$	Yes
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	$CFI = 1.000, SRMR = 0.017$	Yes

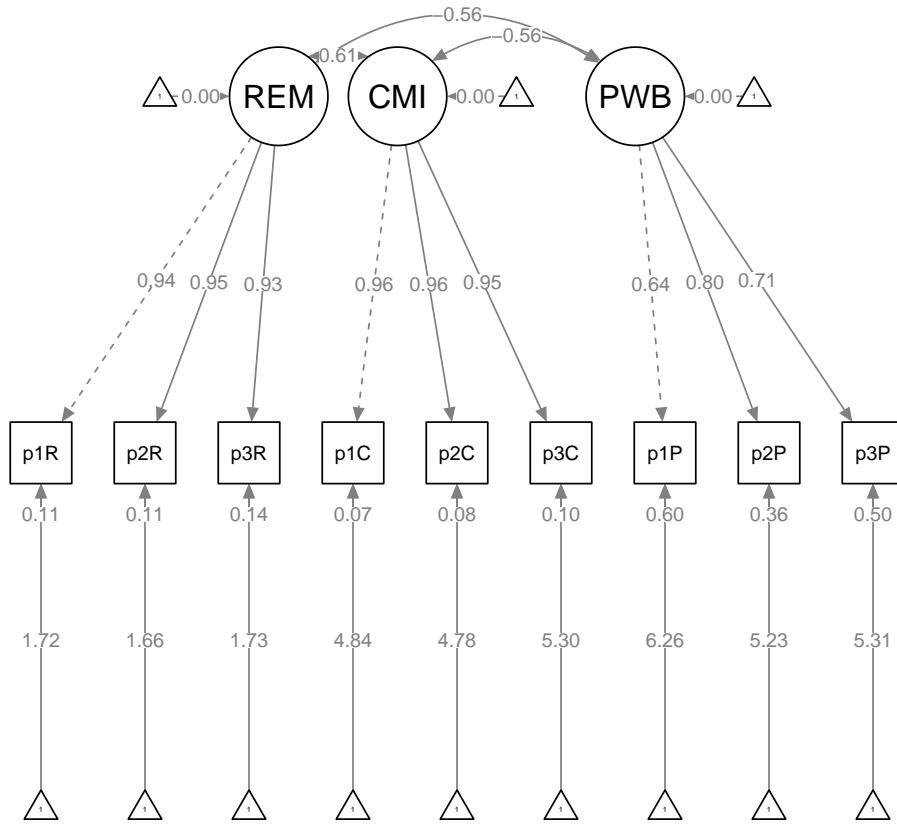
The measurement model created by three parcels, created by random assignment of indicators to each has improved, even further: $\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$

Results of the evaluation of the measurement model can be exported as .csv files with the following code. These produce output that include global fit indices, parameter estimates, and correlations between the latent variables, respectively. The *tidySEM::table_fit* function will display all of the global fit indices.

```
# global fit indices
rp3_msmt_fitstats <- tidySEM::table_fit(rp3_msmt_fit)
write.csv(rp3_msmt_fitstats, file = "rp3_msmt_fitstats.csv")
# parameter estimates
rp3_msmt_pEsts <- tidySEM::table_results(rp3_msmt_fit, digits = 3, columns = NULL)
write.csv(rp3_msmt_pEsts, file = "rp3_msmt_pEsts.csv")
# correlations between latent variables
rp3_msmt_LVcorr <- tidySEM::table_cors(rp3_msmt_fit, digits = 3)
write.csv(rp3_msmt_LVcorr, file = "rp3_msmt_LVcorr.csv")
```

The diagramming function *semPlot::semPaths* can make a pretty good “guess” at simple models such as these. Given that it is unlikely that there would be room for displaying the measurement model in a journal article, there is generally no need to tinker with it too much.

```
semPlot::semPaths(rp3_msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(5,
5, 5, 5))
```



9.10.3 APA Style Write-up of the Results

Earlier in its history, researchers would spend considerable time describing their measurement model and, if they had used it, defending their use of parceling. One of my favorite write-ups occurs in Mallinckrodt et al.'s 2005 article[2005]. Because it is one of the most complete descriptions I have seen, I have consulted it again and again as I have made decisions about my own models. More recent articles describe [Autin et al., 2022, Tokar et al., 2020] generally offer less narration. This is likely because (a) the use of parceling in measurement models is more routine and familiar to the reader/reviewer and (b) in journal articles with limited space, more is allocated for the results and their meaning.

Method/Aalytic Strategy

We specified a model predicting psychological wellbeing from racial ethnic microaggressions, mediated by cultural mistrust. Our primary analysis occurred in two stages. In the first stage we specified and evaluated a measurement model. Our three latent

variables (REM, CMI, PWB) were each indicated by three parcels where items were randomly assigned to each parcel. The latent variables were all allowed to covary with each other. In the second stage we specified and evaluated the structural model. Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16).

Results

Preliminary Analyses

- Missing data analyses and managing missing data
- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

Primary Analyses

Analyzing our proposed multiple mediator model followed the two-step procedure of first evaluating a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index ($CFI \geq 0.95$), root-mean-square error of approximation ($RMSEA \leq 0.06$), and the standard root-mean-square residual ($SRMR \leq 0.08$). To establish acceptable fit, we used $CFI \geq 0.90$, $RMSEA \leq 0.10$, and $SRMR \leq 0.10$ [Weston and Gore, 2006].

We evaluated the measurement model by following recommendations by Little et al. [Little et al., 2002, 2013]. Specifically, each latent variable was represented by three parcels. Parcels were created by randomly assigning scale items to the parcels and then calculating the mean, if at least 65% of the items were non-missing. Factor loadings were all strong, statistically significant, and properly valenced. Global fit statistics were within acceptable thresholds ($\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$). Thus, we proceeded to testing the structural model.

Table 1

Factor Loadings for the Measurement Model					
Latent variable and indicator	est	SE	p	est_std	
Racial/Ethnic Microaggressions					
Parcel 1	1.000	0.000		0.944	

Parcel 2	1.055	0.044	<0.001	0.946
Parcel 3	1.031	0.046	<0.001	0.929
Cultura Mistrust				
Parcel 1	1.000	0.000		0.963
Parcel 2	1.006	0.035	<0.001	0.957
Parcel 3	0.937	0.034	<0.001	0.948
Psychological Well-Being				
Parcel 1	1.000	0.000		0.635
Parcel 2	1.606	0.228	<0.001	0.801
Parcel 3	1.372	0.206	<0.001	0.709

9.11 Residual and Related Questions...

9.11.1 Wait! Why did we do this?

Evaluating a structural model involves two steps. The first is the evaluation of the measurement model. This model allows all of the latent variables to freely covary (i.e., to correlate with each other). If this model has poor fit, the structural model will likely be worse. Therefore, evaluating the measurement model, and potentially respecifying it with parcelized indicators can be a helpful step prior to evaluating the structural model. In fact, it is a step I highly recommend, because, as we saw, it generally improves model fit. And the just-identified solution better accounts for the measurement error.

9.11.2 What if one of my variables only has one or two indicators?

There will be times when we have fewer than three indicators per latent variable (i.e., construct). These can still be represented as latent variables

For two-indicator latent variables, Little et al. [2002] recommended placing an equality constraint on the two loadings associated with the construct because this would locate the construct at the true intersection of the two selected indicators. Procedurally this is fairly straightforward. We simply affix the same label to both items. Recall that in factor definitions, labels are assigned with the asterisk.

```
#TwoItemFactor =~ v1*Item1 + v1*Item2
```

In the case of the one-indicator construct, Little et al. [2002] wrote, “a single-indicator latent variable is essentially equivalent to a manifest variable. In this case, the error of measurement is either fixed at zero or fixed at a non-zero estimate of unreliability; additionally a second corresponding parameter would also need to be fixed because of issue of identification.”

This would be accomplished by two lines of code. The first occurs in the latent variable definitions. The second specifies the error variance of the single observed variable to be 0.00.

```
# OneItemFactor =~ OneItem  
# OneItem ~~ 0*OneItem
```

9.11.3 What if I have missing data?

If the data contain missing values, the default behavior in *lavaan* is listwise deletion. If we can presume that the missing mechanism is MCAR or MAR (e.g., there is no systematic missingness), we can specify a *full information maximum likelihood* (FIML) estimation procedure with the argument *missing = "ml"* (or its alias *missing = "fiml"*). Recall that we retained cases if they had 20% or less missing. Usin the “fiml” option is part of the AIA approach [Parent, 2013].

In the first set of code (i.e., the measurement model with all items as indicators) you may have noticed that I left a hashtagged comment in the code about the *missing = "fiml"* statement. Specifically, the script was taking forever to run. If this were real research, I would have willingly waited. In the parceled, more parsimonious, measurement models, including the code worked fine. Adding this statement also nets an indication of how many missing patterns are found in the data used for the analysis This is found in the introductory matter of the output.

9.12 Practice Problems

The suggested practice for this lesson is to evaluate the measurement model (i.e., all latent variables freed to covary) that precedes an evaluation of the structural (i.e., hypothesized relations only and therefore more restrictive) model.

If your data allows it, perhaps reanalyze one of your previously worked practice problems in its latent variable form. This would involved both this lesson’s measurement model and the next lesson’s structural model.

9.12.1 Problem #1: Rework the research vignette as demonstrated, but change the random seed

If this topic feels a bit overwhelming, simply change the random seed in the data simulation, then rework the problem. This should provide minor changes to the data (maybe in the second or third decimal point), but the results will likely be very similar.

9.12.2 Problem #2: Rework the research vignette, but swap one or more variables

Use the simulated data, but swap out one or more variables.

9.12.3 Problem #3: Try something entirely new.

Evaluate a measurement model 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).

9.12.4 Grading Rubric

Regardless of your choice(s) complete all the elements listed in the grading rubric.

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. Identify the structural model you will evaluate. It should have a minimum of three variables and could be one of the prior path-level models you already examined.	5	_____
2. Import the data and format the variables in the model.	5	_____
3. Specify and evaluate a measurement model with all items as indicators.	10	_____
4. Interpret the results.	5	_____
5. Specify and evaluate a measurement model with either the subscale or randomly assigned to 3 parcels approaches.	10	_____
6. Interpret the results.	5	_____
7. Make notes about similarities and differences in the all-items and parcelled approaches.	5	_____
8. APA style results with table and figure.	5	_____
9. Explanation to grader.	5	_____
Totals	55	_____

9.13 Homeworked Example

Screencast Link

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to evaluate the measurement model that would precede the evaluation of a structural model. And actually, we will need to evaluate two measurement models – an “all items” on indicators model and a parcelled model.

Identify the structural model you will evaluate

It should have a minimum of three variables and could be one of the prior path-level models you already examined

I will repeat the simple mediation that I suggested in path analysis. Specifically, I hypothesize that the evaluation of socially responsive pedagogy will be predicted by intentional recentering through traditional pedagogy.

X = Centering: explicit recentering (0 = precentered; 1 = recentered) M = TradPed: traditional pedagogy (continuously scaled with higher scores being more favorable) Y = SRPed: socially responsive pedagogy (continuously scaled with higher scores being more favorable)

Specify a research model

I am hypothesizing that the evaluation of social responsive pedagogy is predicted by intentional recentering through traditional pedagogy.

Import the data and format the variables in the model

```
raw <- readRDS("ReC.rds")
```

I don't need to score my scales, but it is important to know what they are:

TradPed has 5 items: ClearResponsibilities, EffectiveAnswers, Feedback, ClearOrganization, ClearPresentation SRPed has 4 items: InclusvClassrm, EquitableEval, MultPerspectives, DEIntegration Centering is 1 item – it's a factor with two levels pre, re.

I can create a babydf with just those items.

```
babydf <- dplyr::select(raw, Centering, ClearResponsibilities, EffectiveAnswers,
  Feedback, ClearOrganization, ClearPresentation, InclusvClassrm, EquitableEval,
  MultPerspectives, DEIntegration)
```

Let's check the structure of the variables:

```
str(babydf)
```

```
## Classes 'data.table' and 'data.frame': 310 obs. of 10 variables:
## $ Centering          : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ ClearResponsibilities: int 5 5 4 4 5 4 5 4 4 5 ...
## $ EffectiveAnswers    : int 5 3 5 3 5 3 4 3 2 3 ...
## $ Feedback             : int 5 3 4 2 5 NA 5 4 4 5 ...
## $ ClearOrganization    : int 3 4 3 4 4 4 5 4 4 5 ...
## $ ClearPresentation    : int 4 4 4 2 5 3 4 4 4 5 ...
## $ InclusvClassrm       : int 5 5 5 5 5 4 5 5 4 5 ...
## $ EquitableEval         : int 5 5 3 5 5 3 5 5 3 5 ...
## $ MultPerspectives      : int 5 5 4 5 5 4 5 5 5 5 ...
## $ DEIntegration         : int 5 5 5 5 5 4 5 5 5 5 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

The centering variable will need to be dummy coded as 0/1:

```
babydf$CENTERING <- as.numeric(babydf$Centering)
babydf$CENTERING <- (babydf$CENTERING - 1)
str(babydf)
```

```
## Classes 'data.table' and 'data.frame': 310 obs. of 11 variables:
## $ Centering : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ ClearResponsibilities: int 5 5 4 4 5 4 5 4 5 ...
## $ EffectiveAnswers : int 5 3 5 3 5 3 4 3 2 3 ...
## $ Feedback : int 5 3 4 2 5 NA 5 4 4 5 ...
## $ ClearOrganization : int 3 4 3 4 4 4 5 4 4 5 ...
## $ ClearPresentation : int 4 4 4 2 5 3 4 4 4 5 ...
## $ InclusvClassrm : int 5 5 5 5 5 4 5 5 4 5 ...
## $ EquitableEval : int 5 5 3 5 5 3 5 5 3 5 ...
## $ MultPerspectives : int 5 5 4 5 5 4 5 5 5 5 ...
## $ DEIintegration : int 5 5 5 5 5 4 5 5 5 5 ...
## $ CENTERING : num 1 1 1 1 1 1 1 1 1 1 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

Specify and evaluate a measurement model with all items as indicators

```
init_msmt_mod <- "
  ##measurement model
  CTR =~ CENTERING #this is a single item indicator, I had to add code below to set the
  TradPed =~ ClearResponsibilities + EffectiveAnswers + Feedback + ClearOrganization +
  SRPed =~ InclusvClassrm + EquitableEval + MultPerspectives + DEIintegration

  # Variance of the single item indicator
  CENTERING ~~ 0*CENTERING

  # Covariances
  CTR ~~ TradPed
  CTR ~~ SRPed
  TradPed ~~ SRPed
  "

set.seed(230916)
init_msmt_fit <- lavaan::cfa(init_msmt_mod, data = babydf, missing = "fiml")
init_msmt_fit_sum <- lavaan::summary(init_msmt_fit, fit.measures = TRUE,
  standardized = TRUE)
init_msmt_fit_sum

## lavaan 0.6.16 ended normally after 63 iterations
##
```

```

## Estimator                                ML
## Optimization method                      NLMINB
## Number of model parameters                32
##
## Number of observations                   310
## Number of missing patterns               13
##
## Model Test User Model:
##
## Test statistic                          178.307
## Degrees of freedom                     33
## P-value (Chi-square)                  0.000
##
## Model Test Baseline Model:
##
## Test statistic                         1680.505
## Degrees of freedom                     45
## P-value                               0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)           0.911
## Tucker-Lewis Index (TLI)              0.879
##
## Robust Comparative Fit Index (CFI)    0.906
## Robust Tucker-Lewis Index (TLI)       0.872
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)        -2773.332
## Loglikelihood unrestricted model (H1) -2684.179
##
## Akaike (AIC)                           5610.665
## Bayesian (BIC)                          5730.235
## Sample-size adjusted Bayesian (SABIC)  5628.743
##
## Root Mean Square Error of Approximation:
##
## RMSEA                                 0.119
## 90 Percent confidence interval - lower 0.102
## 90 Percent confidence interval - upper 0.137
## P-value H_0: RMSEA <= 0.050          0.000
## P-value H_0: RMSEA >= 0.080          1.000
##
## Robust RMSEA                            0.125
## 90 Percent confidence interval - lower 0.108
## 90 Percent confidence interval - upper 0.143
## P-value H_0: Robust RMSEA <= 0.050   0.000

```

```

## P-value H_0: Robust RMSEA >= 0.080          1.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                               0.060
##
## Parameter Estimates:
##
## Standard errors                      Standard
## Information                          Observed
## Observed information based on        Hessian
##
## Latent Variables:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR =~
##   CENTERING           1.000
##   TradPed =~
##     ClearRspnsblts  1.000
##     EffectivAnswnrs 0.967  0.056 17.123  0.000  0.694  0.845
##     Feedback          0.915  0.063 14.455  0.000  0.635  0.725
##     ClearOrganiztn   1.193  0.075 15.871  0.000  0.828  0.771
##     ClearPresenttn   1.111  0.063 17.767  0.000  0.771  0.841
##   SRPed =~
##     InclusvClassrm  1.000
##     EquitableEval   0.953  0.087 10.948  0.000  0.455  0.717
##     MultPerspectvs  1.466  0.116 12.623  0.000  0.699  0.839
##     DEIintegration  0.901  0.099  9.106  0.000  0.430  0.582
##
## Covariances:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR ~~
##   TradPed      -0.025  0.020 -1.235  0.217 -0.074 -0.074
##   SRPed        0.008  0.015  0.531  0.595  0.034  0.034
##   TradPed ~~
##   SRPed        0.284  0.033  8.598  0.000  0.858  0.858
##
## Intercepts:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING       0.406  0.028 14.570  0.000  0.406  0.828
## .ClearRspnsblts 4.439  0.047 94.893  0.000  4.439  5.405
## .EffectivAnswnrs 4.365  0.047 93.060  0.000  4.365  5.297
## .Feedback         4.236  0.050 84.626  0.000  4.236  4.835
## .ClearOrganiztn  4.006  0.061 65.611  0.000  4.006  3.732
## .ClearPresenttn  4.236  0.052 81.215  0.000  4.236  4.620
## .InclusvClassrm 4.613  0.039 118.389 0.000  4.613  6.788
## .EquitableEval   4.572  0.036 126.725 0.000  4.572  7.216
## .MultPerspectvs  4.391  0.048 92.287  0.000  4.391  5.265
## .DEIintegration  4.512  0.044 102.435 0.000  4.512  6.105

```

```

##      CTR          0.000          0.000          0.000
##    TradPed        0.000          0.000          0.000
##    SRPed         0.000          0.000          0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING     0.000          0.000          0.000          0.000
## .ClearRspnsblts 0.193       0.020       9.430       0.000       0.193       0.286
## .EffectivAns wrs 0.228       0.023      10.118       0.000       0.228       0.336
## .Feedback       0.364       0.033      10.987       0.000       0.364       0.475
## .ClearOrganiztn 0.466       0.044      10.692       0.000       0.466       0.405
## .ClearPresenttn 0.246       0.026      9.479       0.000       0.246       0.293
## .InclusvClassrm 0.234       0.023      10.216       0.000       0.234       0.507
## .EquitableEval   0.195       0.019      10.227       0.000       0.195       0.485
## .MultPerspectvs  0.206       0.026      7.877       0.000       0.206       0.297
## .DEIIntegration  0.362       0.034      10.528       0.000       0.362       0.662
##    CTR          0.241       0.019     12.450       0.000       1.000       1.000
##    TradPed       0.482       0.054      8.968       0.000       1.000       1.000
##    SRPed         0.228       0.035      6.557       0.000       1.000       1.000

```

Interpret the results

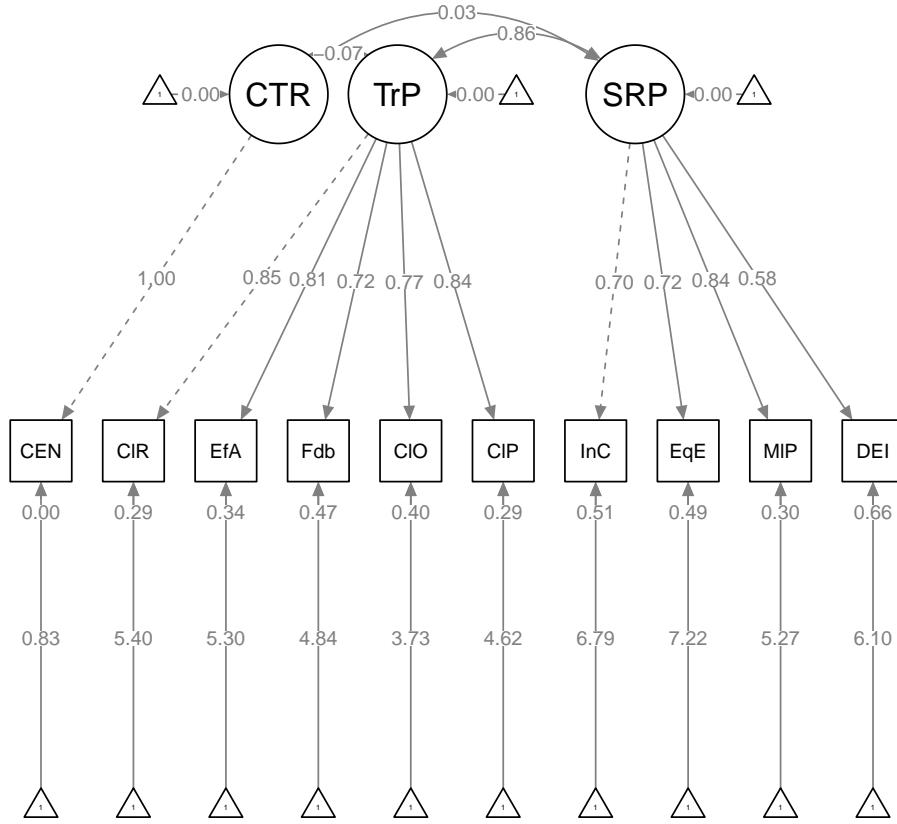
Criteria	Our Results	Criteria met?
Factor loadings significant, strong, proper valence	all $p < 0.001$, lowest = .582(SRPed)	Yes
Non-significant chi-square	$\chi^2(33) = 178.307, p < 0.001$	No
$CFI \geq .95$ (or at least .90)	$CFI = 0.911$	Almost!
$RMSEA \leq .05$ (or $< .08$, at least $< .10$, also 90CI)	$RMSEA = 0.119, 90CI[0.102, 0.137]$	No
$SRMR \leq .08$ (at least $< .10$)	$SRMR = 0.060$	Yes
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	$CFI = 0.911, SRMR = 0.060$	No

Results were mixed. Here is the fit string: $\chi^2(33) = 178.307, p < 0.001, CFI = 0.911, RMSEA = 0.119, 90CI[0.102, 0.137], SRMR = 0.060$

```

semPlot::semPaths(init_msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
  node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(5,
  5, 5, 5))

```



Results of the evaluation of the measurement model can be exported as .csv files with the following code. These produce output that include global fit indices, parameter estimates, and correlations between the latent variables, respectively. The *tidySEM::table_fit* function will display all of the global fit indices.

```
# global fit indices
init_msmt_fitstats <- tidySEM::table_fit(init_msmt_fit)
write.csv(init_msmt_fitstats, file = "init_msmt_fitstats.csv")
# parameter estimates
init_msmt_pEsts <- tidySEM::table_results(init_msmt_fit, digits = 3, columns = NULL)
write.csv(init_msmt_pEsts, file = "init_msmt_pEsts.csv")
# correlations between latent variables
init_msmt_LVcorr <- tidySEM::table_cors(init_msmt_fit, digits = 3)
write.csv(init_msmt_LVcorr, file = "init_msmt_LVcorr.csv")
```

Specify and evaluate a measurement model with either the subscale or randomly assigned to 3 parcels approaches

Each dataset has its unique challenges. This one has relatively few items per scale. I will therefore:

- Retain the single item indicator for CENTERING.
- Randomly assign the 5 items of the TradPed scale to 3 parcels (for parcels with 2, 2, and 1 member each)
 - I don't actually know that this is the best solution, but I will do it for demonstration purposes.
- Retain the four items as indicators for SRPed.

Here I assign the TradPed items to the 3 parcels.

```
set.seed(230916)
items <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback", "ClearOrganization",
          "ClearPresentation")
parcels <- c("p1_TR", "p2_TR", "p3_TR")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))

##           items parcel
## 1   ClearPresentation p1_TR
## 2       Feedback     p2_TR
## 3 ClearResponsibilities p3_TR
## 4   EffectiveAnswers p1_TR
## 5   ClearOrganization p2_TR
```

I can now create the parcels using the traditional scoring procedure. I want both items to be present to score, so I will leave the .75 requirement (as a placeholder).

As a variable, *ClearResponsibilities* will stand alone (i.e., the scoring mechanism won't work on a single variable).

```
TRp1_vars <- c("ClearPresentation", "EffectiveAnswers")
TRp2_vars <- c("Feedback", "ClearOrganization")

babydf$p1_TR <- sjstats::mean_n(babydf[, ..TRp1_vars], 0.75)
babydf$p2_TR <- sjstats::mean_n(babydf[, ..TRp2_vars], 0.75)

# If the scoring code above does not work for you, try the format
# below which involves removing the periods in front of the variable
# list. One example is provided. babydf$p3_TR <-
# sjstats::mean_n(babydf[, TRp3_vars], .75)
```

```

parc_msmt_mod <- "
  ##measurement model
  CTR =~ CENTERING #this is a single item indicator, I had to add code below to set the
  TradPed =~ p1_TR + p2_TR + ClearResponsibilities

  SRPed =~ InclusvClassrm + EquitableEval + MultPerspectives + DEIintegration

  # Variance of the single item indicator
  CENTERING ~~ 0*CENTERING

  # Covariances
  CTR ~~ TradPed
  CTR ~~ SRPed
  TradPed ~~ SRPed
  "

set.seed(230916)
parc_msmt_fit <- lavaan::cfa(parc_msmt_mod, data = babydf, missing = "fiml")
parc_msmt_fit_sum <- lavaan::summary(parc_msmt_fit, fit.measures = TRUE,
  standardized = TRUE)
parc_msmt_fit_sum

## lavaan 0.6.16 ended normally after 59 iterations
##
##    Estimator                               ML
##    Optimization method                    NLMINB
##    Number of model parameters             26
##    Number of observations                  310
##    Number of missing patterns              13
##
## Model Test User Model:
##
##    Test statistic                         112.981
##    Degrees of freedom                      18
##    P-value (Chi-square)                   0.000
##
## Model Test Baseline Model:
##
##    Test statistic                         1292.176
##    Degrees of freedom                      28
##    P-value                                0.000
##
## User Model versus Baseline Model:
##
##    Comparative Fit Index (CFI)           0.925

```

```

## Tucker-Lewis Index (TLI)          0.883
##
## Robust Comparative Fit Index (CFI) 0.920
## Robust Tucker-Lewis Index (TLI)    0.875
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)      -2051.356
## Loglikelihood unrestricted model (H1) -1994.866
##
## Akaike (AIC)                      4154.713
## Bayesian (BIC)                     4251.863
## Sample-size adjusted Bayesian (SABIC) 4169.401
##
## Root Mean Square Error of Approximation:
##
## RMSEA                           0.130
## 90 Percent confidence interval - lower 0.108
## 90 Percent confidence interval - upper 0.154
## P-value H_0: RMSEA <= 0.050       0.000
## P-value H_0: RMSEA >= 0.080       1.000
##
## Robust RMSEA                     0.138
## 90 Percent confidence interval - lower 0.115
## 90 Percent confidence interval - upper 0.162
## P-value H_0: Robust RMSEA <= 0.050 0.000
## P-value H_0: Robust RMSEA >= 0.080 1.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                            0.062
##
## Parameter Estimates:
##
## Standard errors                  Standard
## Information                       Observed
## Observed information based on   Hessian
##
## Latent Variables:
##                   Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR =~
##   CENTERING           1.000
## TradPed =~
##   p1_TR              1.000
##   p2_TR              1.069  0.055 19.427  0.000  0.702  0.870
##   ClearRspnsblts    1.000  0.052 19.076  0.000  0.702  0.869
## SRPed =~
##   InclusvClassrm   1.000

```

```

##   EquitableEval    0.959    0.088   10.922   0.000    0.455    0.719
##   MultPerspectvs  1.474    0.117   12.561   0.000    0.700    0.839
##   DEIintegration  0.905    0.100   9.089    0.000    0.430    0.582
##
## Covariances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR ~~
##   TradPed      -0.024  0.021  -1.152   0.249   -0.069  -0.069
##   SRPed        0.008  0.015   0.523   0.601    0.033   0.033
## TradPed ~~
##   SRPed       0.287  0.033   8.646   0.000    0.861   0.861
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING    0.406  0.028  14.570   0.000    0.406   0.828
## .p1_TR         4.300  0.046  93.680   0.000    4.300   5.332
## .p2_TR         4.116  0.049  83.545   0.000    4.116   4.767
## .ClearRspnsblts 4.439  0.047  94.888   0.000    4.439   5.405
## .InclusvClassrm 4.613  0.039  118.347   0.000    4.613   6.786
## .EquitableEval  4.572  0.036  126.727   0.000    4.572   7.216
## .MultPerspectvs 4.391  0.048  92.290   0.000    4.391   5.265
## .DEIintegration 4.512  0.044  102.459   0.000    4.512   6.106
##   CTR          0.000                               0.000   0.000
##   TradPed      0.000                               0.000   0.000
##   SRPed        0.000                               0.000   0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING    0.000                               0.000   0.000
## .p1_TR         0.158  0.019   8.404   0.000    0.158   0.243
## .p2_TR         0.183  0.022   8.492   0.000    0.183   0.245
## .ClearRspnsblts 0.182  0.020   8.974   0.000    0.182   0.270
## .InclusvClassrm 0.237  0.023  10.252   0.000    0.237   0.512
## .EquitableEval  0.194  0.019  10.203   0.000    0.194   0.483
## .MultPerspectvs 0.205  0.026   7.845   0.000    0.205   0.295
## .DEIintegration 0.361  0.034  10.531   0.000    0.361   0.662
##   CTR          0.241  0.019  12.450   0.000    1.000   1.000
##   TradPed      0.493  0.053   9.361   0.000    1.000   1.000
##   SRPed        0.226  0.035   6.513   0.000    1.000   1.000

```

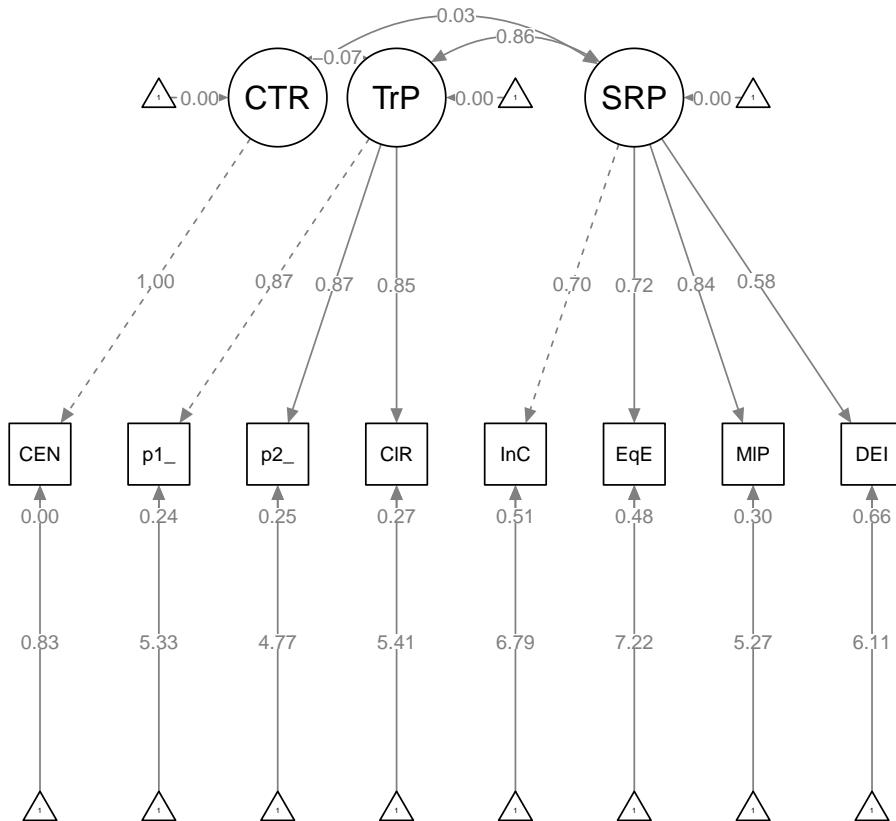
Interpret the results

Criteria	Our Results	Criteria met?
Factor loadings significant, strong, proper valence	all $p < 0.001$, lowest = .430(SRPed)	Yes

Criteria	Our Results	Criteria met?
Non-significant chi-square	$\chi^2(18) = 112.982, p < 0.001$	No
$CFI \geq .95$ (or at least .90)	$CFI = 0.925$	Almost!
$RMSEA \leq .05$ (or $< .08$, at least $< .10$, also 90CI)	$RMSEA = 0.130, 90CI[0.108, 0.154]$	Worsened!
$SRMR \leq .08$ (at least $< .10$)	$SRMR = 0.062$	Worsened!
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	$CFI = 0.925, SRMR = 0.062$	No

Results were mixed and somewhat worse than the initial model that included all item-level indicators. Here is the fit string: $\chi^2(18) = 112.982, p < 0.001, CFI = 0.925, RMSEA = 0.130, 90CI[0.108, 0.154], SRMR = 0.062$

```
semPlot::semPaths(parc_msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(5,
5, 5, 5))
```



```
# global fit indices
parc_msmt_fitstats <- tidySEM:::table_fit(parc_msmt_fit)
write.csv(parc_msmt_fitstats, file = "parc_msmt_fitstats.csv")
# parameter estimates
parc_msmt_pEsts <- tidySEM:::table_results(parc_msmt_fit, digits = 3, columns = NULL)
write.csv(parc_msmt_pEsts, file = "parc_msmt_pEsts.csv")
# correlations between latent variables
parc_msmt_LVcorr <- tidySEM:::table_cors(parc_msmt_fit, digits = 3)
write.csv(parc_msmt_LVcorr, file = "parc_msmt_LVcorr.csv")
```

Make notes about similarities and differences in the all-items and parcelled approaches

- Chi square was significant for both. A statistically significant chi-square is bad – it means that our measurement model covariance matrix is statistically significantly different than the sample covariance matrix.
- The CFI improved a tiny bit for the parcelled model.
- The RMSEA and SRMR worsened a tiny bit for the parcelled model.

Given that I didn't really have enough items to parcel and the fit worsened for some of the indicators for the parcelled model, I think I will stay with the measurement model with item-level indicators. When I created the scales for the course evaluations, I was mindful of degrees of freedom and identification status of the model and intentionally chose to have a few items.

APA style results with table and figure

Analyzing our proposed multiple mediator model followed the two-step procedure of first evaluating a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index (CFI) ≥ 0.95 , root-mean-square error of approximation (RMSEA) ≤ 0.06 , and the standard root-mean-square residual (SRMR) ≤ 0.08 . To establish acceptable fit, we used CFI ≥ 0.90 , RMSEA ≤ 0.10 , and SRMR ≤ 0.10 [Weston and Gore, 2006].

We evaluated the measurement model by following recommendations by Little et al. [Little et al., 2002, 2013]. Specifically, each latent variable was represented by each of the items on its subscale. Given that TradPed and SRPed had 5 and 4 items, respectively, we did not parcel items. The Centering variable with two levels (pre-centered, re-centered) was recoded as a dummy variable with 0, 1 coding. In the specification, its measurement error was fixed at zero. While all factor loadings were strong, statistically significant, and properly balanced, global fit statistics were mixed: $\chi^2(33) = 178.307, p < 0.001, CFI = 0.911, RMSEA = 0.119, 90CI[0.102, 0.137], SRMR = 0.060$. We proceeded to testing the strutural model with caution.

Table 1

Factor Loadings for the Measurement Model				
Latent variable and indicator	est	SE	p	est_std
Traditional Pedagogy				
ClearResponsibilities	1.000	0.000		0.845
EffectiveAnswers	0.967	0.056	<0.001	0.815
Feedback	0.915	0.063	<0.001	0.725
ClearOrganization	1.193	0.075	<0.001	0.771
ClearPresentation	1.111	0.063	<0.001	0.841
Socially Responsive Pedagogy				
InclusvClassrm	1.000	0.000		0.702
EquitableEval	0.953	0.087	<0.001	0.717
MultPerspectives	1.466	0.116	<0.001	0.839
DEIntegration	0.901	0.099	<0.001	0.582
CENTERING	0.000	0.000		0.000

Explanation to grader

Chapter 10

Specifying and Evaluating the Structural Model

[Screencasted Lecture Link](#)

In the prior lesson we engaged in the first of two stages in SEM by establishing the *measurement model*. In this stage we specify and evaluate the *structural model*. Additionally, using the same data we specify, evaluate, and compare *alternative models* and learn about the interpretive challenge of *equivalent* models.

10.1 Navigating this Lesson

There is about 1 hour and 30 minutes 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](#)

10.1.1 Learning Objectives

Learning objectives from this lecture include the following:

- Specify an SEM structural model.
- Interpret *global* fit indices (e.g., Chi-square, CFI, RMSEA).
- Interpret *local* fit indices (e.g., regression weights/factor loadings, parameter estimates that we specify).
- With the same data and variables, specify and interpret results from an alternative model.
- Compare the fit of nested models.
- Describe the interpretive challenge of *equivalent models*.

10.1.2 Planning for Practice

This is the second of a two-part lesson on structural equation modeling. In the prior lesson we established the *measurement model*. In this lesson we specify and evaluate a full *structural model*, respecify and evaluate an *alternative* model, and compare the two on the basis of their global fit indices. Your models should involve a minimum of three latent variables and should extend from the prior lesson on measurement models. As always, the suggestions for homework are graded in complexity.

- Rework the problem in the chapter by changing the random seed in the code that simulates the data. This should provide minor changes to the data, but the results will likely be very similar.
- Use the research data from the chapter, but evaluate a different set of variables.
- Use data from another lesson or data that is available to you.

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.

- Kline, R. (2016). Principles and practice of structural equation modeling (Fourth ed., Methodology in the social sciences). New York: The Guilford Press.
 - Chapter 10, Specification and Identification of Structural Regression Models
 - Chapter 14, Analysis of Structural Regression Models
- 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 6, Application 4: Testing the Factorial Validity of a Causal Structure
- Kim, P. Y., Kendall, D. L., & Cheon, H.-S. (2017). Racial microaggressions, cultural mistrust, and mental health outcomes among Asian American college students. *American Journal of Orthopsychiatry*, 87(6), 663–670. <https://doi-org.ezproxy.spu.edu/10.1037/ort0000203>
 - This is the research vignette for this lesson.

10.1.4 Packages

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(tidyverse)){install.packages('tidyverse')}
# if(!require(psych)){install.packages('psych')}
# if(!require(jtools)){install.packages('jtools')}
```

10.2 Evaluating Structural Models

The model we are testing in this lesson is *hybrid* that is, it contains both CFA and the structural paths. Although there are several detours along the way, the analytic approach has two large stages:

In the prior lesson we tested the *measurement model*. This included each of the factors and its indicators with covariances between each of the latent variables. We learned that the measurement model will generally have the best fit because all of the structural paths are saturated (i.e., there is a covariance between them). Consequently, it is critical that good fit be established in the measurement model because fit will not improve in subsequent models.

In this lesson we will test the *structural model*. This means we delete the covariances and respecify the model to include the directional paths and covariances we have hypothesized. Once we have tested our model, we will follow the lead of our research vignette [Kim et al., 2017] by rearranging the variables to test *alternative* or competing models. This introduces the challenge that is unique to SEM – one of *equivalent* models.

10.3 Workflow for Evaluating a Structural Model

Workflow: Evaluating a Structural Model	
A Priori Power Analysis	
<ul style="list-style-type: none"> Conduct an a priori power analysis to determine the appropriate sample size. Draw estimates of effect from pilot data and/or the literature. 	
Scrubbing & Scoring	
<ul style="list-style-type: none"> Import data and format (i.e., variable naming, reverse-scoring) item level variables. Analyze item-level missingness. If using scales, create the mean scores of the scales. Determine and execute approach for managing missingness. Popular choices are available item analysis (e.g., Parent, 2013) and multiple imputation. Analyze scale-level missingness. Create a df with <i>only</i> the items (scaled in the proper direction). 	
Data Diagnostics	
<ul style="list-style-type: none"> Evaluate univariate normality (i.e., one variable at a time) with Shapiro-Wilks tests; $p < .05$ indicates a violation of univariate normality. Evaluate multivariate normality (i.e., all continuously scaled variables simultaneously) with Mahalanobis test. Identify outliers (e.g., cases with Mahal values $> 3 SDs$ from the centroid). Consider deleting (or transforming <i>if</i> there is an extreme-ish “jump” in the sorted values). Evaluate internal consistency of the scaled scores with Cronbach’s alpha or omega; the latter is increasingly preferred. 	
Specify and evaluate a measurement model	
<ul style="list-style-type: none"> In this just-identified (saturated) model, all latent variables are specified as covarying <ul style="list-style-type: none"> For LVs with 3 items or more, remember to set a marker/reference variable. For LVs with 2 items, constrain the loadings to be equal For single-item indicators fix the error variance to zero (or a non-zero estimate of unreliability) Evaluate results with <i>global</i> (e.g., χ^2, CFI, RMSEA, SRMR) and <i>local</i> (i.e., factor loadings and covariances) fit indices In the event of poor fit, respecify LVs with multiple indicators with parcels. Nested alternative measurement models can be compared with χ^2 difference, ΔCFI tests; non-nested models with AIC, and BIC tests 	
Specify and evaluate a structural model	
<ul style="list-style-type: none"> Replace the covariances with paths that represent the a priori hypotheses <ul style="list-style-type: none"> These models could take a variety of forms. It is possible to respecify models through trimming or building approaches. Evaluate results using: <ul style="list-style-type: none"> global fit indices (e.g., χ^2, CFI, RMSEA, SRMS), local fit indices (i.e., strength and significance of factor loadings, covariances, and additional model parameters [e.g., indirect effects]) Consider respecifying and evaluating one or more <i>alternative</i> models: <ul style="list-style-type: none"> <i>Forward searching</i> frees parameters (i.e., adds paths or covariances) and can use modification indices as a guide. <i>Backward searching</i> restrains parameters (i.e., deletes paths or covariances) and can use low and non-significant paths as a guide. Compare the fit of the alternate models <ul style="list-style-type: none"> Nested models can be compared with χ^2 difference, ΔCFI tests Non-nested models can use AIC, and BIC comparisons (lower values suggest better fit) 	
Quick Guide for Global and Comparative Fit Statistics	
<ul style="list-style-type: none"> $\chi^2, p > .05$; this test is sensitive to sample size and this value can be difficult to attain CFI $> .95$ (or at least $.90$) RMSEA (and associated 90%CI) are $< .05$ ($< .08$, or at least $< .10$) SRMR $< .08$ (or at least $< .10$) Combination rule: CFI $< .95$ and SRMR $< .08$ AIC and BIC are compared; the lowest values suggest better models $\chi^2\Delta$ is statistically significant; the model with the superior global fit is the better model ΔCFI is greater than 0.01; the model with CFI values closest to 1.0 has better fit 	

Evaluating a structural model involves the following steps:

- A Priori Power Analysis

- Conduct an a priori power analysis to determine the appropriate sample size. __ Draw estimates of effect from pilot data and/or the literature.
- Scrubbing & Scoring
 - Import data and format (i.e., variable naming, reverse-scoring) item level variables.
 - Analyze item-level missingness.
 - If using scales, create the mean scores of the scales.
 - Determine and execute approach for managing missingness. Popular choices are available item analysis (e.g., Parent, 2013) and multiple imputation.
 - Analyze scale-level missingness.
 - Create a df with only the items (scaled in the proper direction).
- Data Diagnostics
 - Evaluate univariate normality (i.e., one variable at a time) with Shapiro-Wilks tests; $p < .05$ indicates a violation of univariate normality.
 - Evaluate multivariate normality (i.e., all continuously scaled variables simultaneously) with Mahalanobis test. Identify outliers (e.g., cases with Mahal values > 3 SDs from the centroid). Consider deleting (or transforming if there is an extreme-ish “jump” in the sorted values).
 - Evaluate internal consistency of the scaled scores with Cronbach’s alpha or omega; the latter is increasingly preferred. Specify and evaluate a measurement model
 - In this just-identified (saturated) model, all latent variables are specified as covarying.
 - * For LVs with 3 items or more, remember to set a marker/reference variable,
 - * For LVs with 2 items, constrain the loadings to be equal,
 - * For single-item indicators fix the error variance to zero (or a non-zero estimate of unreliability).
 - Evaluate results with global (e.g., X₂, CFI, RMSEA, SRMR) and local (i.e., factor loadings and covariances) fit indices.
 - In the event of poor fit, respecify LVs with multiple indicators with parcels.
 - Nested alternative measurement models can be compared with X₂ difference, ΔCFI tests; non-nested models with AIC, and BIC tests .
- Specify and evaluate a structural model.
 - Replace the covariances with paths that represent the a priori hypotheses.
 - * These models could take a variety of forms.
 - * It is possible to respecify models through trimming or building approaches.
 - Evaluate results using
 - * *global* fit indices (e.g., X₂, CFI, RMSEA, SRMS),
 - * *local* fit indices (i.e., strength and significance of factor loadings, covariances, and additional model parameters [e.g., indirect effects]).
 - Consider respecifying and evaluating one or more *alternative* models.
 - * *Forward searching* involves freeing parameters (adding paths or covariances) and can use modification indices as a guide.
 - * *Backward searching* involves restraining parameters (deleting paths or covariances) and can use low and non-significant paths as a guide.
 - Compare the fit of the alternate models.

- * Nested models can be compared with χ^2 difference and ΔCFI tests.
- * Non-nested models can be compared with AIC and BIC (lower values suggest better fit).
- Quick Guide for Global and Comparative Fit Statistics.
 - χ^2 , $p < .05$; this test is sensitive to sample size and this value can be difficult to attain
 - CFI $> .95$ (or at least $.90$)
 - RMSEA (and associated 90%CI) are $< .05$ ($< .08$, or at least $< .10$)
 - SRMR $< .08$ (or at least $< .10$)
 - Combination rule: CFI $< .95$ and SRMR $< .08$
 - AIC and BIC are compared; the lowest values suggest better models
 - $\chi^2\Delta$ is statistically significant; the model with the superior fit is the better model
 - δCFI is greater than 0.01; the model with CFI values closest to 1.0 has better fit

10.4 Research Vignette

The research vignette comes from the Kim, Kendall, and Cheon's [2017], "Racial Microaggressions, Cultural Mistrust, and Mental Health Outcomes Among Asian American College Students." Participants were 156 Asian American undergraduate students in the Pacific Northwest. The researchers posited the a priori hypothesis that cultural mistrust would mediate the relationship between racial microaggressions and two sets of outcomes: mental health (e.g., depression, anxiety, well-being) and help-seeking.

Variables used in the study included:

- **REMS:** Racial and Ethnic Microaggressions Scale (Nadal, 2011). The scale includes 45 items on a 2-point scale where 0 indicates no experience of a microaggressive event and 1 indicates it was experienced at least once within the past six months. Higher scores indicate more experience of microaggressions.
- **CMI:** Cultural Mistrust Inventory (Terrell & Terrell, 1981). This scale was adapted to assess cultural mistrust harbored among Asian Americans toward individuals from the mainstream U.S. culture (e.g., Whites). The CMI includes 47 items on a 7-point scale where higher scores indicate a higher degree of cultural mistrust.
- **ANX, DEP, PWB:** Subscales of the Mental Health Inventory (Veit & Ware, 1983) that assess the mental health outcomes of anxiety (9 items), depression (4 items), and psychological well-being (14 items). Higher scores (on a 6 point scale) indicate stronger endorsement of the mental health outcome being assessed.
- **HlpSkg:** The Attitudes Toward Seeking Professional Psychological Help – Short Form (Fischer & Farina, 1995) includes 10 items on a 4-point scale (0 = disagree, 3 = agree) where higher scores indicate more favorable attitudes toward help seeking.

Below is a figure of the model that we will be using structural equation modeling to test

10.4.1 Simulating the data from the journal article

We used the *lavaan::simulateData* function for the simulation. If you have taken psychometrics, you may recognize the code as one that creates latent variables from item-level data. In trying

to be as authentic as possible, we retrieved factor loadings from psychometrically oriented articles that evaluated the measures [Nadal, 2011, Veit and Ware, 1983]. For all others we specified a factor loading of 0.80. We then approximated the *measurement model* by specifying the correlations between the latent variable. We sourced these from the correlation matrix from the research vignette [Kim et al., 2017]. The process created data with multiple decimals and values that exceeded the boundaries of the variables. For example, in all scales there were negative values. Therefore, the final element of the simulation was a linear transformation that rescaled the variables back to the range described in the journal article and rounding the values to integer (i.e., with no decimal places).

```
# Entering the intercorrelations, means, and standard deviations from
# the journal article

Kim_generating_model <- "
  ##measurement model
  REMS =~ .82*Inf32 + .75*Inf38 + .74*Inf21 + .72*Inf17 + .69*Inf9 + .61*Inf36 + .51*In
  CMI =~ .8*cmi1 + .8*cmi2 + .8*cmi3 + .8*cmi4 + .8*cmi5 + .8*cmi6 + .8*cmi7 + .8*cmi8 +
  ANX =~ .80*Anx1 + .80*Anx2 + .77*Anx3 + .74*Anx4 + .74*Anx5 + .69*Anx6 + .69*Anx7 + .
  DEP =~ .74*Dep1 + .83*Dep2 + .82*Dep3 + .74*Dep4
  PWB =~ .83*pwb1 + .72*pwb2 + .67*pwb3 + .79*pwb4 + .77*pwb5 + .75*pwb6 + .74*pwb7 + .7
  HlpSkg =~ .8*hlpsskg1 + .8*hlpsskg2 + .8*hlpsskg3 + .8*hlpsskg4 + .8*hlpsskg5 + .8*hlpsskg6

  # Means
  REMS ~ 0.34*1
  CMI ~ 3*1
  ANX ~ 2.98*1
  DEP ~ 2.36*1
  PWB ~ 3.5*1
  HlpSkg ~ 1.64*1
  # Correlations
  REMS ~ 0.58*CMI
  REMS ~ 0.26*ANX
  REMS ~ 0.34*DEP
  REMS ~ -0.25*PWB
  REMS ~ -0.02*HlpSkg
  CMI ~ 0.12*ANX
  CMI ~ 0.19*DEP
  CMI ~ -0.28*PWB
  CMI ~ 0*HlpSkg
  ANX ~ 0.66*DEP
  ANX ~ -0.55*PWB
  ANX ~ 0.07*HlpSkg
  DEP ~ -0.66*PWB
  DEP ~ 0.05*HlpSkg
  PWB ~ 0.08*HlpSkg
```

```

"
set.seed(230916)
dfKim <- lavaan::simulateData(model = Kim_generating_model, model.type = "sem",
  meanstructure = T, sample.nobs = 156, standardized = FALSE)
library(tidyverse)

# used to retrieve column indices used in the rescaling script below
# col_index <- as.data.frame(colnames(dfKim))

for (i in 1:ncol(dfKim)) {
  # for loop to go through each column of the dataframe apply only
  # to REMS variables
  if (i >= 1 & i <= 45) {
    dfKim[, i] <- scales::rescale(dfKim[, i], c(0, 1))
  }
  if (i >= 46 & i <= 116) {
    # apply only to CMI variables
    dfKim[, i] <- scales::rescale(dfKim[, i], c(1, 7))
  }
  if (i >= 93 & i <= 116) {
    # apply only to mental health variables
    dfKim[, i] <- scales::rescale(dfKim[, i], c(1, 5))
  }
  if (i >= 117 & i <= 126) {
    # apply only to HlpSkng variables
    dfKim[, i] <- scales::rescale(dfKim[, i], c(0, 3))
  }
}

library(tidyverse)
dfKim <- dfKim %>%
  round(0)

```

The script below allows you to store the simulated data as a file on your computer. This is optional – the entire lesson can be worked with the simulated data.

If you prefer the .rds format, use this script (remove the hashtags). The .rds format has the advantage of preserving any formatting of variables. A disadvantage is that you cannot open these files outside of the R environment.

Script to save the data to your computer as an .rds file.

```
#saveRDS(dfKim, 'dfKim.rds')
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
#dfKim<- readRDS('dfKim.rds')
```

If you prefer the .csv format (think “Excel lite”) use this script (remove the hashtags). An advantage of the .csv format is that you can open the data outside of the R environment. A disadvantage is that it may not retain any formatting of variables

Script to save the data to your computer as a .csv file.

```
# write.table(dfKim, file = 'dfKim.csv', sep = ',', col.names=TRUE,
# row.names=FALSE)
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfKim<- read.csv ('dfKim.csv', header = TRUE)
```

10.5 Scrubbing, Scoring, and Data Diagnostics

Because the focus of this lesson is on the specific topic of specifying and evaluating a structural model for SEM and have used simulated data, we are skipping many of the steps in scrubbing, scoring and data diagnostics. If this were real, raw, data, it would be important to [scrub](#), if needed [score](#), and conduct [data diagnostics](#) to evaluate the suitability of the data for the proposed analyses.

10.6 Script for Specifying Models in *lavaan*

SEM in *lavaan* requires fluency with the R script. Below is a brief overview of the operators we use most frequently:

- Latent variables (factors) must be *defined* by their manifest or latent indicators.
 - the special operator ($=\sim$, *is measured/defined by*) is used for this
 - Example: $f1 \sim y1 + y2 + y3$
- Regression equations use the single tilda (\sim , *is regressed on*)
 - place DV (y) on left of operator
 - place IVs, separate by + on the right
 - Example: $y \sim f1 + f2 + x1 + x2$
 - * f is a latent variable in this example
 - * y , $x1$, and $x2$ are observed variables in this example
 - An asterisk can affix a label in subsequent calculations and in interpreting output
- 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: $f1 \sim f2$ *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 models. Readibility of model syntax is improved by:

- splitting formulas over multiple lines
- using blank lines within single quote
- labeling with the hashtag

10.7 Quick Specification of the Measurement Model

Recall that the first step in establishing a structural model is to specify, evaluate, and if necessary re-specify the measurement model. In the prior lesson I decided to randomly assigning items to three indicators per parcel. Because our data simulation produced item-level data, I will briefly repeat the code for the solution we chose. For more information on this process please revisit the lesson on [establishing the measurement model](#). Here is a representation of the measurement model we are specifying.

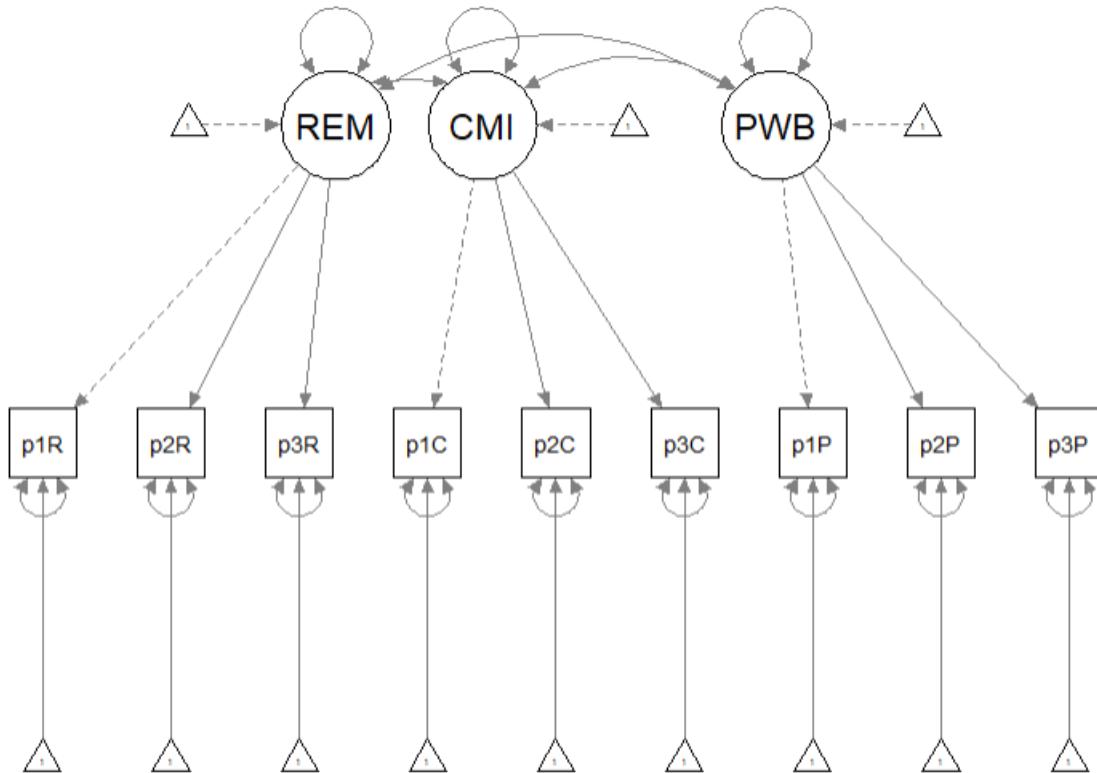


Figure 10.1: An image of the measurement model that we are specifying

This code randomly assigns the PWB items to three parcels.

```
set.seed(230916)
items <- c("pwb1", "pwb2", "pwb3", "pwb4", "pwb5", "pwb6", "pwb7", "pwb8",
  "pwb9", "pwb10", "pwb11")
parcels <- c("PWB_p1", "PWB_p_2", "PWB_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))

##      items  parcel
## 1    pwb7  PWB_p1
## 2    pwb9  PWB_p_2
## 3    pwb1  PWB_p3
## 4   pwb11  PWB_p1
## 5    pwb2  PWB_p_2
## 6    pwb3  PWB_p3
## 7    pwb4  PWB_p1
## 8   pwb10  PWB_p_2
## 9    pwb5  PWB_p3
## 10   pwb8  PWB_p1
## 11   pwb6  PWB_p_2
```

We can now create the parcels using the same scoring procedure as we did for the REMS and CMI instruments.

```
PWB_p1_vars <- c("pwb7", "pwb11", "pwb4", "pwb8")
PWB_p2_vars <- c("pwb9", "pwb2", "pwb10", "pwb2")
PWB_p3_vars <- c("pwb1", "pwb3", "pwb5")

dfKim$p1PWB <- sjstats::mean_n(dfKim[, PWB_p1_vars], 0.75)
dfKim$p2PWB <- sjstats::mean_n(dfKim[, PWB_p2_vars], 0.75)
dfKim$p3PWB <- sjstats::mean_n(dfKim[, PWB_p3_vars], 0.75)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfKim$p3PWB <-
# sjstats::mean_n(dfKim[, ..PWB_p3_vars], .75)
```

This code randomly assigns the REMS items to three parcels.

```
set.seed(230916)
items <- c("Inf32", "Inf38", "Inf21", "Inf17", "Inf9", "Inf36", "Inf5",
  "Inf22", "SClass6", "SClass31", "SClass8", "SClass40", "SClass2", "SClass34",
  "SClass11", "mInv27", "mInv30", "mInv39", "mInv7", "mInv26", "mInv33",
  "mInv4", "mInv14", "mInv10", "Exot3", "Exot29", "Exot45", "Exot35",
  "Exot42", "Exot23", "Exot13", "Exot20", "Exot43", "mEnv37", "mEnv24",
  "mEnv19", "mEnv28", "mEnv18", "mEnv41", "mEnv12", "mWork25", "mWork15",
```

```

  "mWork1", "mWork16", "mWork44")
parcels <- c("REMS_p1", "REMS_p2", "REMS_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))

##      items  parcel
## 1   mWork44 REMS_p1
## 2    mEnv41 REMS_p2
## 3   mWork25 REMS_p3
## 4    Exot20 REMS_p1
## 5   SClass11 REMS_p2
## 6     Inf38 REMS_p3
## 7   SClass8 REMS_p1
## 8     Inf5  REMS_p2
## 9     Inf17 REMS_p3
## 10   Exot43 REMS_p1
## 11   mEnv19 REMS_p2
## 12   Exot23 REMS_p3
## 13   mInv4  REMS_p1
## 14   mInv39 REMS_p2
## 15   mInv33 REMS_p3
## 16  SClass31 REMS_p1
## 17     Inf22 REMS_p2
## 18   mEnv28 REMS_p3
## 19  SClass40 REMS_p1
## 20   mEnv18 REMS_p2
## 21     Inf36 REMS_p3
## 22   mWork16 REMS_p1
## 23   mInv30 REMS_p2
## 24   mInv14 REMS_p3
## 25   Exot45 REMS_p1
## 26     Inf32 REMS_p2
## 27   mEnv37 REMS_p3
## 28  SClass2  REMS_p1
## 29  SClass34 REMS_p2
## 30     Inf21 REMS_p3
## 31   Exot3  REMS_p1
## 32   Exot29 REMS_p2
## 33   mEnv12 REMS_p3
## 34   mEnv24 REMS_p1
## 35   mInv27 REMS_p2
## 36   Exot42 REMS_p3
## 37   Exot35 REMS_p1
## 38   mInv10 REMS_p2
## 39  SClass6  REMS_p3
## 40     mInv7 REMS_p1
## 41   mWork1  REMS_p2
## 42     Inf9  REMS_p3

```

```
## 43  Exot13 REMS_p1
## 44  mWork15 REMS_p2
## 45  mInv26 REMS_p3
```

This code provides means for each of the three REMS parcels.

```
REMS_p1_vars <- c("mWork44", "Exot20", "SClass8", "Exot43", "mInv4", "SClass31",
  "SClass40", "mWork16", "Exot45", "SClass2", "Exot3", "mEnv24", "Exot35",
  "mInv7", "Exot13")
REMS_p2_vars <- c("mEnv41", "SClass11", "Inf5", "mEnv19", "mInv39", "Inf22",
  "mEnv18", "mInv30", "Inf32", "SClass34", "Exot29", "mInv27", "mInv10",
  "mWork1", "mWork15")
REMS_p3_vars <- c("mWork25", "Inf38", "Inf17", "Exot23", "mInv33", "mEnv28",
  "Inf36", "mInv14", "mEnv37", "Inf21", "mEnv12", "Exot42", "SClass6",
  "Inf9", "mInv26")

dfKim$p1REMS <- sjstats::mean_n(dfKim[, REMS_p1_vars], 0.8)
dfKim$p2REMS <- sjstats::mean_n(dfKim[, REMS_p2_vars], 0.8)
dfKim$p3REMS <- sjstats::mean_n(dfKim[, REMS_p3_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfKim$p3REMS <-
# sjstats::mean_n(dfKim[, ..REMS_p3_vars], .80)
```

We can repeat the process for the CMI. First, we assign the 47 CMI items to the three parcels.

```
set.seed(230916)
items <- c(IntRel_vars <- c("cmi1", "cmi2", "cmi3", "cmi4", "cmi5", "cmi6",
  "cmi7", "cmi8", "cmi9", "cmi10", "cmi11", "cmi12", "cmi13", "cmi14",
  "cmi15", "cmi16", "cmi17", "cmi18", "cmi19", "cmi20", "cmi21", "cmi22",
  "cmi23", "cmi24", "cmi25", "cmi26", "cmi27", "cmi28", "cmi29", "cmi30",
  "cmi31", "cmi32", "cmi33", "cmi34", "cmi35", "cmi36", "cmi37", "cmi38",
  "cmi39", "cmi40", "cmi41", "cmi42", "cmi43", "cmi44", "cmi45", "cmi46",
  "cmi47"))
parcels <- c("CMI_p1", "CMI_p2", "CMI_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))
```

```
##   items parcel
## 1  cmi45 CMI_p1
## 2  cmi39 CMI_p2
## 3  cmi41 CMI_p3
## 4  cmi32 CMI_p1
## 5  cmi15 CMI_p2
## 6  cmi2  CMI_p3
## 7  cmi11 CMI_p1
## 8  cmi7  CMI_p2
```

```
## 9 cmi4 CMI_p3
## 10 cmi33 CMI_p1
## 11 cmi38 CMI_p2
## 12 cmi30 CMI_p3
## 13 cmi22 CMI_p1
## 14 cmi18 CMI_p2
## 15 cmi46 CMI_p3
## 16 cmi12 CMI_p1
## 17 cmi40 CMI_p2
## 18 cmi6 CMI_p3
## 19 cmi26 CMI_p1
## 20 cmi47 CMI_p2
## 21 cmi17 CMI_p3
## 22 cmi25 CMI_p1
## 23 cmi27 CMI_p2
## 24 cmi23 CMI_p3
## 25 cmi44 CMI_p1
## 26 cmi1 CMI_p2
## 27 cmi10 CMI_p3
## 28 cmi13 CMI_p1
## 29 cmi14 CMI_p2
## 30 cmi3 CMI_p3
## 31 cmi31 CMI_p1
## 32 cmi42 CMI_p2
## 33 cmi29 CMI_p3
## 34 cmi34 CMI_p1
## 35 cmi8 CMI_p2
## 36 cmi28 CMI_p3
## 37 cmi37 CMI_p1
## 38 cmi19 CMI_p2
## 39 cmi21 CMI_p3
## 40 cmi43 CMI_p1
## 41 cmi20 CMI_p2
## 42 cmi35 CMI_p3
## 43 cmi24 CMI_p1
## 44 cmi5 CMI_p2
## 45 cmi36 CMI_p3
## 46 cmi9 CMI_p1
## 47 cmi16 CMI_p2
```

This code provides means for each of the three CMI parcels.

```
CMI_p1_vars <- c("cmi45", "cmi32", "cmi11", "cmi33", "cmi22", "cmi12",
  "cmi26", "cmi25", "cmi44", "cmi13", "cmi31", "cmi34", "cmi37", "cmi43",
  "cmi24", "cmi9")
CMI_p2_vars <- c("cmi39", "cmi15", "cmi7", "cmi38", "cmi18", "cmi40", "cmi47",
  "cmi27", "cmi1", "cmi14", "cmi42", "cmi18", "cmi19", "cmi20", "cmi5",
```

```

"cmi16")
CMI_p3_vars <- c("cmi41", "cmi2", "cmi4", "cmi30", "cmi46", "cmi6", "cmi17",
  "cmi23", "cmi10", "cmi3", "cmi29", "cmi28", "cmi21", "cmi35", "cmi36")

dfKim$p1CMI <- sjstats::mean_n(dfKim[, CMI_p1_vars], 0.8)
dfKim$p2CMI <- sjstats::mean_n(dfKim[, CMI_p2_vars], 0.8)
dfKim$p3CMI <- sjstats::mean_n(dfKim[, CMI_p3_vars], 0.8)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfKim$p3CMI <-
# sjstats::mean_n(dfKim[, ..CMI_p3_vars], .80)

```

Below is code for specifying the measurement model. Each of the latent variables/factors (REMS, CMI, PWB) is identified by three parcels. Each of the latent variables is allowed to covary with the others.

```

rp3_msmt_mod <- "
  ##measurement model
  REMS =~ p1REMS + p2REMS + p3REMS

  CMI =~ p1CMI + p2CMI + p3CMI

  PWB =~ p1PWB + p2PWB + p3PWB

  # Covariances
  REMS ~~ CMI
  REMS ~~ PWB
  CMI ~~ PWB
  ""

set.seed(230916)
rp3_msmt_fit <- lavaan::cfa(rp3_msmt_mod, data = dfKim, missing = "fiml")
rp3_msmt_fit_sum <- lavaan::summary(rp3_msmt_fit, fit.measures = TRUE,
  standardized = TRUE)
rp3_msmt_fit_sum

## lavaan 0.6.16 ended normally after 106 iterations
##
##   Estimator                               ML
##   Optimization method                     NLMINB
##   Number of model parameters              30
##   Number of observations                  156
##   Number of missing patterns               1
##
```



```

## Standard errors                               Standard
## Information                                  Observed
## Observed information based on               Hessian
##
## Latent Variables:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## REMS =~
##   p1REMS      1.000
##   p2REMS      1.055    0.044 23.770  0.000  0.271  0.944
##   p3REMS      1.031    0.046 22.399  0.000  0.280  0.929
## CMI =~
##   p1CMI      1.000
##   p2CMI      1.006    0.035 28.780  0.000  0.773  0.963
##   p3CMI      0.937    0.034 27.373  0.000  0.778  0.957
## PWB =~
##   p1PWB      1.000
##   p2PWB      1.606    0.228  7.055  0.000  0.491  0.801
##   p3PWB      1.372    0.206  6.662  0.000  0.420  0.709
##
## Covariances:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## REMS ~~
##   CMI         0.128    0.021  6.199  0.000  0.608  0.608
##   PWB        -0.047    0.010 -4.712  0.000 -0.562 -0.562
## CMI ~~
##   PWB        -0.132    0.028 -4.763  0.000 -0.559 -0.559
##
## Intercepts:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS      0.494    0.023 21.448  0.000  0.494  1.717
## .p2REMS      0.504    0.024 20.790  0.000  0.504  1.665
## .p3REMS      0.522    0.024 21.642  0.000  0.522  1.733
## .p1CMI       3.888    0.064 60.479  0.000  3.888  4.842
## .p2CMI       3.886    0.065 59.667  0.000  3.886  4.777
## .p3CMI       4.051    0.061 66.190  0.000  4.051  5.299
## .p1PWB       3.014    0.039 78.207  0.000  3.014  6.262
## .p2PWB       3.207    0.049 65.324  0.000  3.207  5.230
## .p3PWB       3.141    0.047 66.276  0.000  3.141  5.306
## REMS          0.000
## CMI           0.000
## PWB           0.000
##
## Variances:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS      0.009    0.002  5.466  0.000  0.009  0.109
## .p2REMS      0.010    0.002  5.330  0.000  0.010  0.106
## .p3REMS      0.012    0.002  6.278  0.000  0.012  0.137

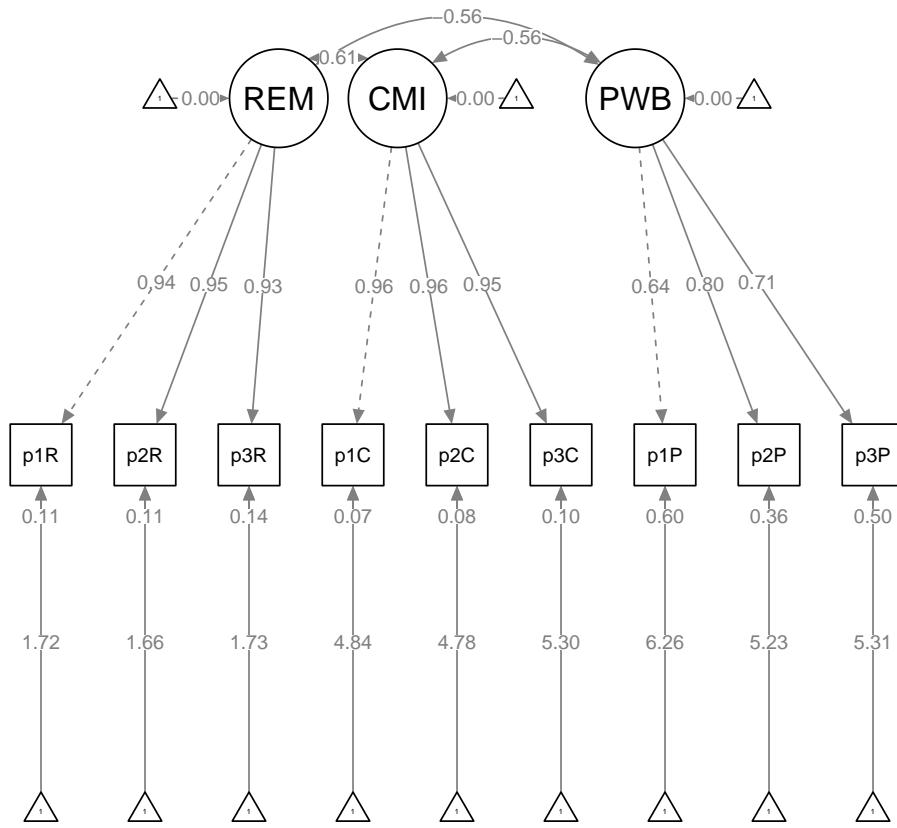
```

##	.p1CMI	0.047	0.009	5.046	0.000	0.047	0.072
##	.p2CMI	0.056	0.010	5.597	0.000	0.056	0.084
##	.p3CMI	0.060	0.009	6.292	0.000	0.060	0.102
##	.p1PWB	0.138	0.019	7.309	0.000	0.138	0.597
##	.p2PWB	0.135	0.029	4.646	0.000	0.135	0.358
##	.p3PWB	0.174	0.028	6.320	0.000	0.174	0.498
##	REMS	0.074	0.009	7.838	0.000	1.000	1.000
##	CMI	0.598	0.073	8.171	0.000	1.000	1.000
##	PWB	0.093	0.024	3.963	0.000	1.000	1.000

As we determined in the prior lesson, the factor loadings were all strong, statistically significant, and properly valenced. Further, global fit statistics were within acceptable thresholds ($\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$).

The figure below is an illustration of our measurement model with its results. It also conveys that each latent variable is indicated by three parcels and all of the latent variables are allowed to covary.

```
semPlot::semPaths(rp3_msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
  node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(5,
  5, 5, 5))
```



```
# semPlot::semPaths(rp3_msmt_fit) #ignore -- used to create a
# no-results figure earlier in the chapter
```

10.8 The Structural Model: Specification and Evaluation

The **structural model** evaluates the hypothesized relations between the latent variables. The structural model is typically more parsimonious (i.e., not saturated) than the measurement model and is characterized by directional paths (not covariances) between some (not all) of the variables.

When there are positive degrees of freedom in the structural mode (technically this becomes an *over-identified* model), it will have lower fit than the measurement model. This is why it was critical to establish a well-fitting measurement model, first.

Here's a quick reminder of the hypothesized model we are testing from Kim et al. [Kim et al., 2017]. We are evaluating a simple mediation model, predicting psychological well-being from racial microaggressions mediated by cultural mistrust. The model is *hybrid* because it include measurement models (i.e., latent variables indicated by their parcels), plus the hypothesized paths.

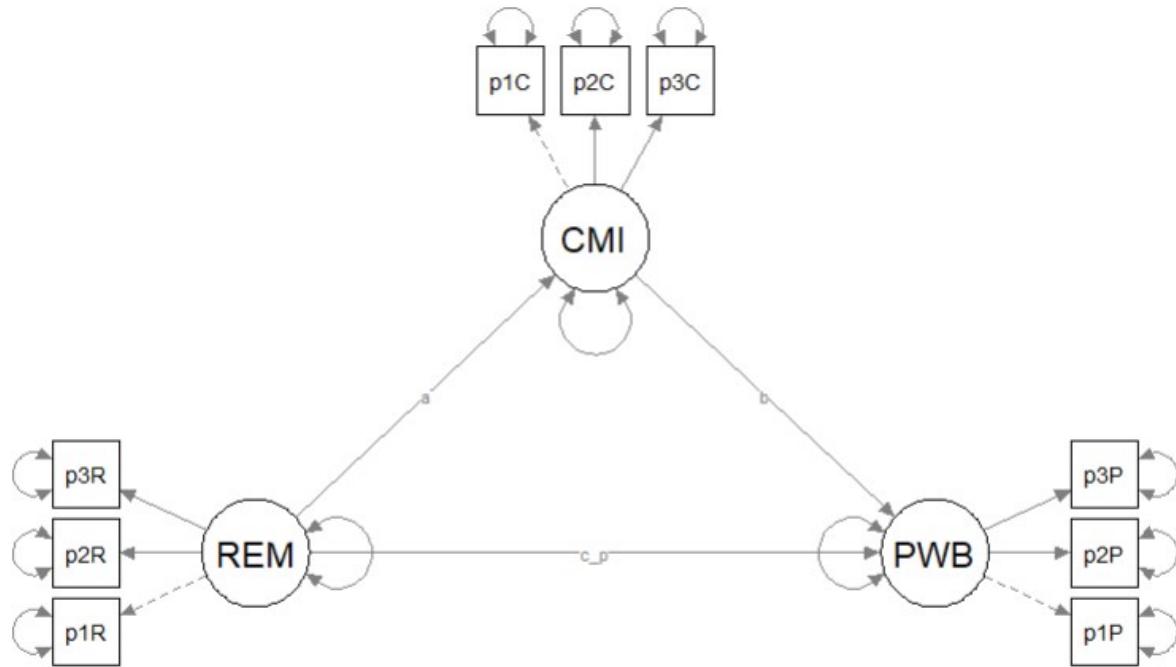


Figure 10.2: Image of the proposed statistical model – a simple mediation that will be tested with structural equation modeling

Having just confirmed that our measurement model is adequate, we now replace the covariances between latent variables with the paths (directional) and covariances (bidirectional) we hypothesize. These paths and covariances are *soft* hypotheses. That is, we are “freeing” them to relate. In SEM, *hard* hypotheses are where no path/covariance exists and the relationship between these variables is “fixed” to zero. This is directly related to degrees of freedom and the identification status (just-identified, over-identified, underidentified) of the model.

10.8.1 Model Identification

There are two necessary elements for identifying any type of SEM [Kline, 2016b], these include

- having degrees of freedom greater-than-or-equal to zero ($df_M \geq 0$), and
- assigning a scale to every latent variable (including disturbances or error terms)
 - *lavaan::sem* and *lavaan::cfa* automatically assign “1” to the first indicator in each latent variable

In the case of the specification of standard CFA models (i.e., the models we use in the psychometric evaluation of measures and surveys), the extent of our “your model must be identified” conversation stopped at:

- unidimensional models need to have a minimum of 3 items/indicators (manifest variables) per factor/scale (latent variable)

- multidimensional models need to have a minimum of 2 items/indicators (manifest variables) per factor/scale (latent variable)
- second order factors need three first-order factors in order to be identified
- nonstandard models include error variances that are free to correlate – they need closer scrutiny with regard to identification status

Model identification, though, is more complicated than that. At the risk of repetition, let's look again at model identification in hybrid models as it relates to the $df_M \geq 0$ criteria.

Underidentified or undetermined models have fewer observations (knowns) than free model parameters (unknowns). This results in negative degrees of freedom ($df_M \leq 0$). This means that it is impossible to find a unique set of estimates. The classic example for this is: $a + b = 6$ where there are an infinite number of solutions.

Just-identified or just-determined models have an equal number of observations (knowns) as free parameters (unknowns). This results in zero degrees of freedom ($df_M = 0$). Just-identified scenarios will result in a unique solution. The classic example for this is

$$\begin{aligned} a + b &= 6 \\ 2a + b &= 10 \end{aligned}$$

The unique solution is $a = 4$, $b = 2$.

Over-identified or overdetermined models have more observations (knowns) than free parameters (unknowns). This results in positive degrees of freedom ($df_M > 0$). In this circumstance, there is no single solution, but one can be calculated when a statistical criterion is applied. For example, there is no single solution that satisfies all three of these formulas:

$$\begin{aligned} a + b &= 6 \\ 2a + b &= 10 \\ 3a + b &= 12 \end{aligned}$$

When we add this instruction “Find value of a and b that yield total scores such that the sum of squared differences between the observations (6, 10, 12) and these total scores is as small as possible.” Curious about the answer? An excellent description is found in Kline [2016b].

Model identification is an incredibly complex topic. It is possible to have theoretically identified models and yet they are statistically unidentified and then the researcher must hunt for the source of the problem. For this lesson, I will simply walk through the steps that are commonly used in determining the identification status of a structural model.

10.8.1.1 Model identification for the overall SEM

In order to be evaluated, structural models need to be *just identified* ($df_M = 0$) or *overidentified* ($df_M > 0$). Computer programs are not (yet) good at estimating identification status because it is based on symbolism and not numbers. Therefore, we researchers must do the mental math to

ensure that our *knowns* (measured/observed variables) are equal (just-identified) or greater than (overidentified) our *unknowns* (parameters that will be estimated).

We calculate the *knowns* by identifying the number of measured variables (n) and popping that number into this equation: $\frac{n(n+1)}{2}$. *Unknowns* are counted and include: measurement regression paths, structural regression paths, error covariances, residual error variances, and covariances.

Lets calculate this for our model.

- **Knowns:** There are 9 observed variables, so we have 45 $(9(9+1)/2)$ pieces of information from which to drive the parameters of the model.
- **Unknowns:** We must estimate the following parameters
 - 6 measurement regression paths (we don't count marker variables or the single-indicator items)
 - 3 structural regression paths
 - 9 error covariances (1 for each indicator variable)
 - 2 residual error variances (any endogenous [predicted] latent variable has one of these)
 - 0 covariances
 - We have a total of: 20 unknowns

```
(9*(9+1))/2
```

```
## [1] 45
```

Our overall model is overidentified with $df_M = 20$. We know this because subtracted the unknowns (20) from the knowns (45). If we calculated this correctly, 25 will be the degrees of freedom associated with the chi-square test.

```
45-20
```

```
## [1] 25
```

10.8.1.2 Model identification for the structural portion of the model

It is possible to have an overidentified model but still be underidentified in the structural portion. In order to be evaluated, structural models need to be *just identified* ($df_M = 0$) or *overidentified* ($df_M > 0$). Before continuing, it is essential to understand that the structural part is (generally) the relations between the latent variables (although in some models there could be observed variables). In our case, our structural model consists only of four latent variables.

Especially for the structural portion of the model, statistical packages are not (yet) good at estimating identification status because it is based on symbolism and not numbers. Therefore, we researchers must make the calculations to ensure that our *knowns* are equal (just-identified) or greater than (overidentified) our *unknowns*.

Knowns: $\frac{k(k+1)}{2}$ where k is the number of *constructs* (humoR: konstrukts?)in the model. In our case, we have three constructs: $3(3+1)/2 = 6$

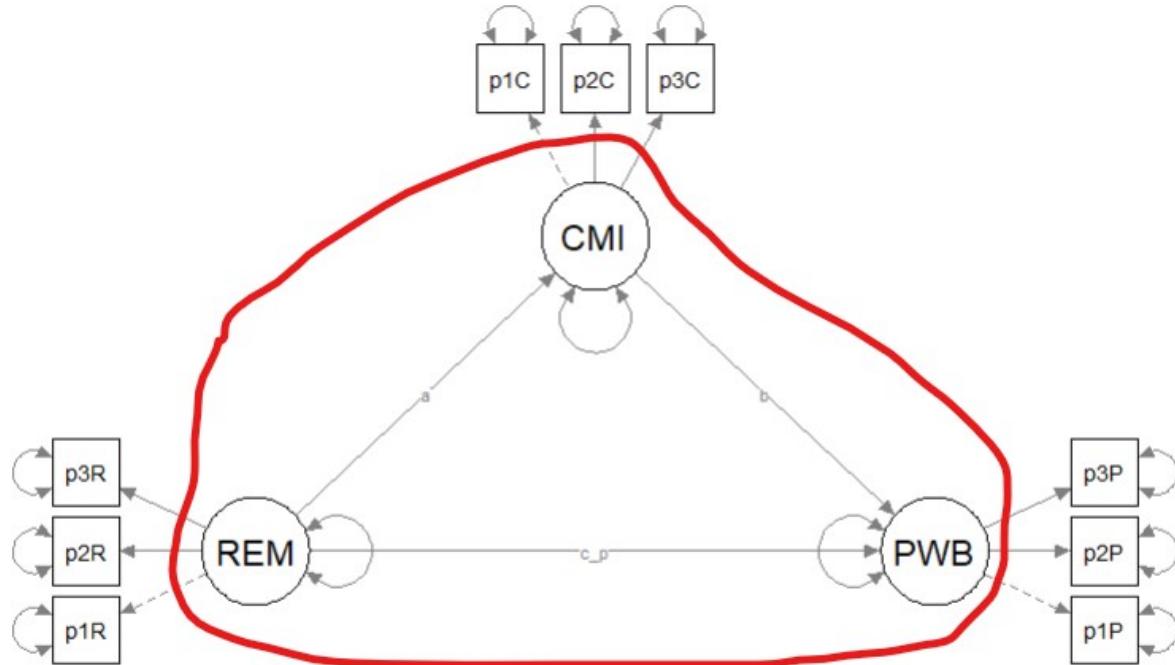


Figure 10.3: The structural portion of the model is circled in red.

```
(3*(3+1))/2
```

```
## [1] 6
```

Unknowns: are calculated with the following

- Exogenous (predictor) variables (1 variance estimated for each): we have 1 (REM)
- Endogenous (predicted) variables (1 disturbance variance for each): we have 2 (CMI, PWB)
- Correlations between variables (1 covariance for each pairing): we have 0
- Regression paths (arrows linking exogenous variables to endogenous variables): we have 3

With 6 knowns and 6 unknowns, we have 0 degrees of freedom in the structural portion of the model. This is a *just-identified*, fully saturated, model. This model is the nested model and will always have the best fit. When models are especially complex, some researchers will start with a just-identified model and trim paths to get to their hypothesized model and compare the fit to see if there are statistically significant differences as the trimming progresses. The researcher hopes that the fit of the more parsimonious model will not be statistically significantly different from the just-identified model.

Today, we already know the results of our *global fit indices* because when models are just-identified/fully saturated, the fit of the test of the structural model will be identical to the fit of the measurement model. As you might imagine, this introduces some challenge to our interpretation. Stated another way, the *measurement* model will always have the best fit because it is fully saturated (i.e., covariances between all latent variables) and its structure will best replicate the sample covariance matrix. When we have over-identified models (i.e., models with positive degrees

of freedom), our hope is that replacing covariances (double-headed arrows) with unidirectional paths and constraining some relations to be 0.0 will not result in a substantial deterioration of fit.

10.8.1.3 Specifying and Evaluating the Structural Model

Specifying our structural model in *lavaan* includes script for the measurement model, the structural model, and any additional model parameters (e.g., indirect and total effects) that we might add. In the script below you will see each of these elements. In fact, in today's example the script in the structural model is identical to that in the lesson on [simple mediation](#). Specifically,

- the mediator (CMI) is predicted by the independent variable (REMS),
- the dependent variable (PWB) is predicted by the independent variable (REMS) and the mediator (CMI),
- labels are assigned to represent the a , b , and c' paths
- calculations that use the labels will estimate the indirect, direct, and total paths

In the model specification below, there are more elements to note. Because *lavaan* has elements of randomness in its algorithms (particularly around its version of bias-corrected, bootstrapped confidence intervals), including a *set.seed* function will facilitate the reproducibility of results.

If the data contain missing values, the default behavior in *lavaan::sem* is listwise deletion. If we can presume that the missing mechanism is MCAR or MAR (e.g., there is no systematic missingness), we can specify a *full information maximum likelihood* (FIML) estimation procedure with the *missing = "fiml"* argument. Recall that we retained cases if they had 20% or less missing. Using the "fiml" option is part of the AIA approach [Parent, 2013].

In the *lavaan::summary* function, we will want to retrieve the global fit indices with the *fit.measures=TRUE*. Because SEM figures are often represented with standardized values, we will want *standardized = TRUE*. And if we wish to know the proportion of variance predicted in our endogenous variables, we will include *rsq = TRUE*.

In the *lavaan::parameterEstimates* we can obtain *lavaan*'s version of bias-corrected bootstrapped confidence intervals (they aren't quite the same) by including *boot.ci.type = "bca.simple"*.

```
Kim_struct_mod <- "
  ##measurement model
  REMS =~ p1REMS + p2REMS + p3REMS
  CMI =~ p1CMI + p2CMI + p3CMI
  PWB =~ p1PWB + p2PWB + p3PWB

  #structural model with labels for calculation of the indirect effect
  PWB ~ b*CMI + c_p*REMS
  CMI ~ a*REMS

  #calculations
  indirect := a*b
  direct := c_p
  total_c := c_p + (a*b)
```

```

  "
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence
Kim_struct_fit <- lavaan::sem(Kim_struct_mod, data = dfKim, missing = "fiml")
Kim_struct_summary <- lavaan::summary(Kim_struct_fit, fit.measures = TRUE,
  standardized = TRUE, rsq = TRUE)
Kim_struct_pEsts <- lavaan::parameterEstimates(Kim_struct_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
Kim_struct_summary

## lavaan 0.6.16 ended normally after 95 iterations
##
##             ML
## Estimator
## Optimization method           NLMINB
## Number of model parameters      30
##
##             156
## Number of observations
## Number of missing patterns       1
##
## Model Test User Model:
##
##             15.965
## Test statistic
## Degrees of freedom              24
## P-value (Chi-square)            0.889
##
## Model Test Baseline Model:
##
##             1366.804
## Test statistic
## Degrees of freedom                36
## P-value                           0.000
##
## User Model versus Baseline Model:
##
##             1.000
## Comparative Fit Index (CFI)
## Tucker-Lewis Index (TLI)          1.009
##
##             1.000
## Robust Comparative Fit Index (CFI)
## Robust Tucker-Lewis Index (TLI)    1.009
##
## Loglikelihood and Information Criteria:
##
##             -368.149
## Loglikelihood user model (H0)
## Loglikelihood unrestricted model (H1) -360.166
##
##             796.297
## Akaike (AIC)
## Bayesian (BIC)                   887.793
## Sample-size adjusted Bayesian (SABIC) 792.834
##

```

```

## Root Mean Square Error of Approximation:
##
##    RMSEA                               0.000
##    90 Percent confidence interval - lower 0.000
##    90 Percent confidence interval - upper 0.031
##    P-value H_0: RMSEA <= 0.050          0.988
##    P-value H_0: RMSEA >= 0.080          0.000
##
##    Robust RMSEA                         0.000
##    90 Percent confidence interval - lower 0.000
##    90 Percent confidence interval - upper 0.031
##    P-value H_0: Robust RMSEA <= 0.050   0.988
##    P-value H_0: Robust RMSEA >= 0.080   0.000
##
## Standardized Root Mean Square Residual:
##
##    SRMR                                0.017
##
## Parameter Estimates:
##
##    Standard errors                      Standard
##    Information                           Observed
##    Observed information based on        Hessian
##
## Latent Variables:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    REMS =~
##      p1REMS      1.000
##      p2REMS      1.055  0.044 23.770 0.000  0.271  0.944
##      p3REMS      1.031  0.046 22.399 0.000  0.280  0.929
##    CMI =~
##      p1CMI       1.000
##      p2CMI       1.006  0.035 28.780 0.000  0.773  0.963
##      p3CMI       0.937  0.034 27.373 0.000  0.778  0.957
##    PWB =~
##      p1PWB       1.000
##      p2PWB       1.606  0.228  7.055 0.000  0.491  0.801
##      p3PWB       1.372  0.206  6.662 0.000  0.420  0.709
##
## Regressions:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    PWB ~
##      CMI       (b) -0.136  0.042 -3.267 0.001 -0.345 -0.345
##      REMS     (c_p) -0.396  0.122 -3.250 0.001 -0.352 -0.352
##    CMI ~
##      REMS     (a)  1.732  0.199  8.698 0.000  0.608  0.608
##
## Intercepts:

```

```

##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS                  0.494   0.023  21.448  0.000  0.494  1.717
## .p2REMS                  0.504   0.024  20.790  0.000  0.504  1.665
## .p3REMS                  0.522   0.024  21.642  0.000  0.522  1.733
## .p1CMI                   3.888   0.064  60.479  0.000  3.888  4.842
## .p2CMI                   3.886   0.065  59.667  0.000  3.886  4.777
## .p3CMI                   4.051   0.061  66.190  0.000  4.051  5.299
## .p1PWB                   3.014   0.039  78.207  0.000  3.014  6.262
## .p2PWB                   3.207   0.049  65.324  0.000  3.207  5.230
## .p3PWB                   3.141   0.047  66.276  0.000  3.141  5.306
## REMS                      0.000
## CMI                       0.000
## PWB                       0.000
##
## Variances:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS                  0.009   0.002  5.466  0.000  0.009  0.109
## .p2REMS                  0.010   0.002  5.330  0.000  0.010  0.106
## .p3REMS                  0.012   0.002  6.278  0.000  0.012  0.137
## .p1CMI                   0.047   0.009  5.046  0.000  0.047  0.072
## .p2CMI                   0.056   0.010  5.597  0.000  0.056  0.084
## .p3CMI                   0.060   0.009  6.292  0.000  0.060  0.102
## .p1PWB                   0.138   0.019  7.309  0.000  0.138  0.597
## .p2PWB                   0.135   0.029  4.646  0.000  0.135  0.358
## .p3PWB                   0.174   0.028  6.320  0.000  0.174  0.498
## REMS                      0.074   0.009  7.838  0.000  1.000  1.000
## CMI                       0.377   0.048  7.871  0.000  0.631  0.631
## PWB                       0.057   0.015  3.716  0.000  0.610  0.610
##
## R-Square:
##                               Estimate
## p1REMS                  0.891
## p2REMS                  0.894
## p3REMS                  0.863
## p1CMI                   0.928
## p2CMI                   0.916
## p3CMI                   0.898
## p1PWB                   0.403
## p2PWB                   0.642
## p3PWB                   0.502
## CMI                      0.369
## PWB                      0.390
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## indirect                 -0.236   0.077  -3.085  0.002  -0.210  -0.210
## direct                   -0.396   0.122  -3.250  0.001  -0.352  -0.352
## total_c                  -0.633   0.114  -5.557  0.000  -0.562  -0.562

```

```
# Kim_struct_pEsts #although creating the object is useful to export
# as a .csv I didn't ask it to print into the book
```

Below is script that will export the global fit indices (via `tidySEM::table_fit`) and the parameter estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```
# global fit indices
Kim_struct_fitstats <- tidySEM::table_fit(Kim_struct_fit)
```

```
## Registered S3 method overwritten by 'tidySEM':
##   method      from
##   predict.MxModel  OpenMx

write.csv(Kim_struct_fitstats, file = "Kim_struct_fitstats.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(Kim_struct_pEsts, file = "Kim_struct_pEsts.csv")
```

10.8.1.4 Interpreting the Output

We can begin to interpret the output by tabling the regression weights. As we can see, our model accounts for 37% of the variance in our mediator, cultural mistrust and 39% of the variance in the dependent variable, psychological well-being. Our regression weights are all statistically significant. Specifically we note that:

- REMS has a negative effect on CMI
- CMI has a negative effect on PWB
- REMS has a negative effect on PWB
- There is a statistically significant indirect effect that is evidence in both the *p* value and 95% confidence interval

To be consistent with my figure, in this table I have included the standardized results (i.e., β).

Table 2

Model Coefficients Assessing the Effect of Racial Microaggressions on Psychological Well-Being Through Cultural Mistrust

Predictor	B	SE _B	p	β	R ²
Cultural Mistrust (M)					.37
Racial microaggressions (a)	1.732	0.199	< 0.001	0.608	
Psychological Well-Being (DV)					.39
Cultural mistrust (b)	-0.136	0.042	0.001	-0.345	

Predictor	B	SE_B	p	β	R^2
Racial microaggressions (c')	-0.396	0.122	0.001	-0.352	

Effects	B	SE_B	p	95% CI	
Indirect($a * b$)	-0.236	0.077	0.002	-0.210	-0.386, -0.086
Total effect	-0.633	0.114	< 0.001	-0.633	-0.856, -0.410

Note. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

In structural equation modeling we also interpret the global fit indices. Let's start by comparing them to our a priorily identified thresholds.

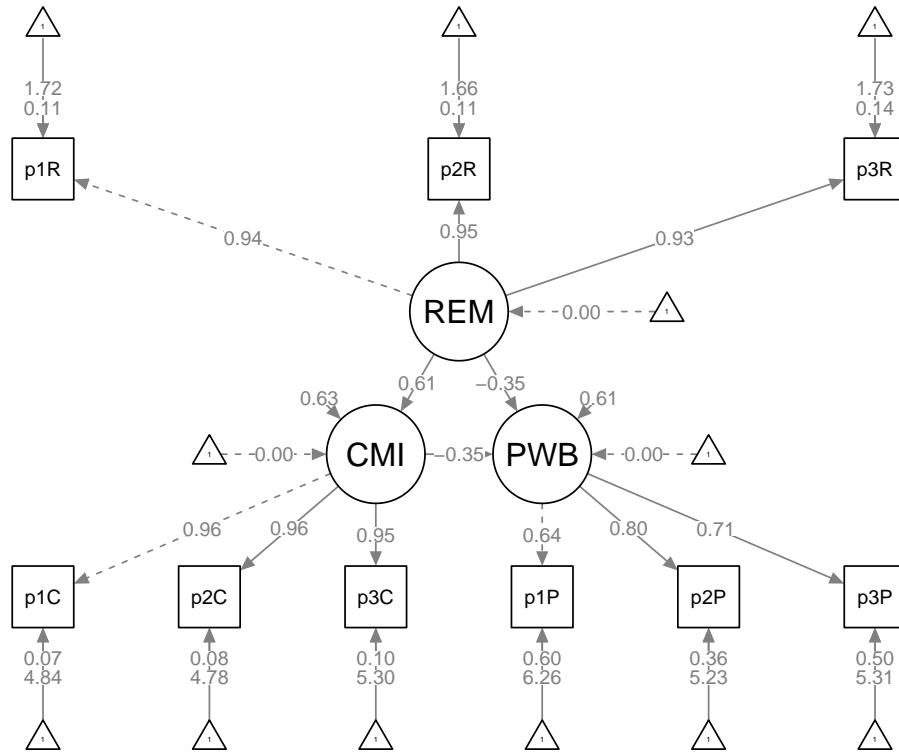
Criteria	Our Results	Criteria met?
Non-significant chi-square	$\chi^2(24) = 15.965, p = 0.889$	Yes
$CFI \geq .95$ (or at least .90)	$CFI = 1.000$	Yes
$RMSEA \leq .05$ (or < .08, at least < .10, also 90CI)	$RMSEA = 0.000, 90CI[0.000, 0.031]$	Yes
$SRMR \leq .08$ (at least < .10)	$SRMR = 0.017$	Yes
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	$CFI = 1.000, SRMR = 0.017$	Yes

The statistical string for the global fit indices can be represented this way: $\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$.

As I predicted, the global fit indices of the measurement model are identical to the structural model. Recall that in terms of model both models are *just-identified*. This means there are zero degrees of freedom in the structural model. That is, there are no additional paths (directional or bidirectional) that could be drawn between the latent variables.

Plotting the results can be useful in checking our work and, if correct, understanding the relations between the variables. The *semPlot::semPaths* function will produce an initial guess at what we might like that can be further tweaked.

```
plot_Kim_struct <- semPlot::semPaths(Kim_struct_fit, what = "col", whatLabels = "stand",
  sizeMan = 5, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(5, 5, 5, 5))
```



Although the code below may look daunting, I find it to be a fairly straightforward way to obtain figures that convey the model we are testing. We first start by identifying the desired location of our latent variables, using numbers to represent their position by “(column, row)”. In the table below, I have mapped my variables.

Grid for Plotting `semplot::sempath`

(1,1) empty	(1,2) CMI	(1,3) empty
(2,1) REM	(2,2) empty	(2,3) PWB

We place these values along with the names of our latent variables in to the `sempath::semplot` function.

```
# IMPORTANT: Must use the node names (take directly from the SemPlot)
# assigned by SemPlot You can change them as the last thing
m1_msmt <- semptools::layout_matrix(REM = c(2, 1), CMI = c(1, 2), PWB = c(2,
 3))
```

Next we provide instruction on the direction (up, down, left, right) we want the indicator/observed variables to face. We identify the direction by the location of each of our latent variables. For example, in the code below we want the indicators for the REM variable (2,1) to face left.

```
# tell where you want the indicators to face
m1_point_to <- semptools::layout_matrix(left = c(2, 1), up = c(1, 2), right = c(2,
3))
```

The next two sets of code work together to specify the order of the observed variables for each factor. in the top set of code the variable names indicate the order in which they will appear (i.e., p1R, p2R, p3R). In the second set of code, the listing the variable name three times (i.e., REM, REM, REM) serves as a placeholder for each of the indicators.

It is critical to note that we need to use the abbreviated variable names assigned by *semTools::semPaths* and not necessarily the names that are in the dataframe.

```
# the next two codes -- indicator_order and indicator_factor are
# paired together, they specify the order of observed variables for
# each factor
m1_indicator_order <- c("p1R", "p2R", "p3R", "p1C", "p2C", "p3C", "p1P",
"p2P", "p3P")

m1_indicator_factor <- c("REM", "REM", "REM", "CMI", "CMI", "CMI", "PWB",
"PWB", "PWB")
```

The next two sets of codes provide some guidance about how far away the indicator (square/rectangular) variables should be away from the latent (oval/circular) variables. Subsequently, the next set of values indicate how far away each of the indicator (square/rectangular) variables should be spread apart.

```
#next set of code pushes the indicator variables away from the factor
m1_indicator_push <- c(REM = .5,
                      CMI = 1,
                      PWB = 1)

m1_indicator_spread <- c(REM = 1, #spreading the boxes away from each other
                           CMI = 1,
                           PWB = 1)
```

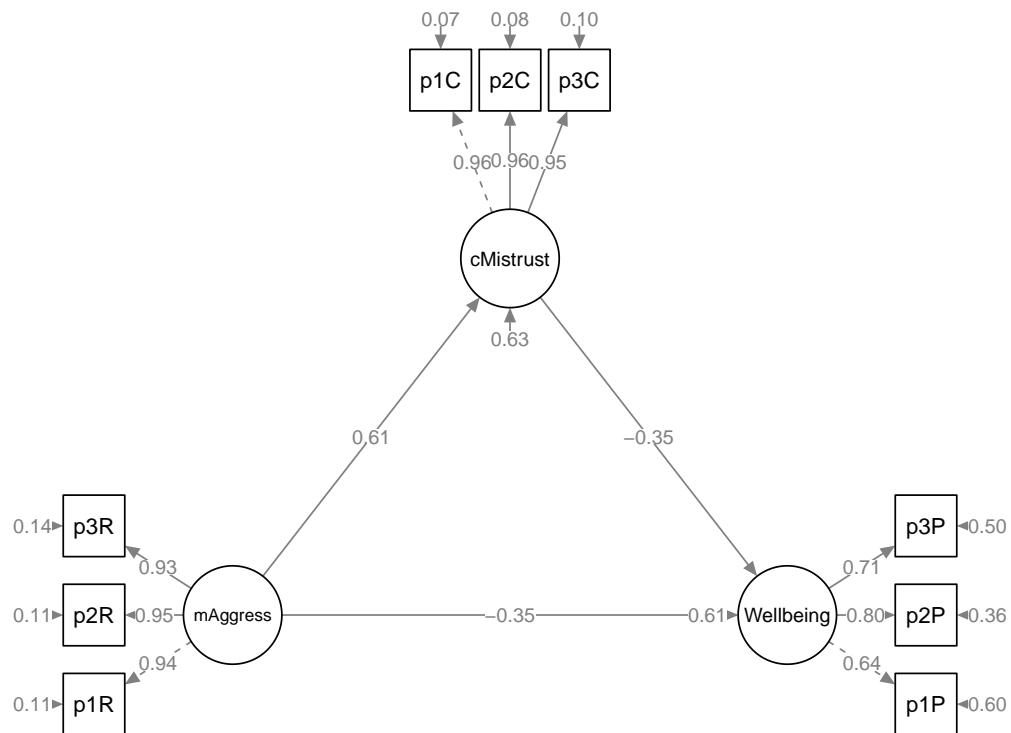
Finally, we can feed all of the objects that whole these instructions into the *sempTools::sem_set_layout* function. If desired, we can use the *sempTools::change_node_label* function to rename the latent variables. Again, make sure to use the variable names that *semPlot::semPaths* has assigned.

```
plot1 <- sempTools::set_sem_layout(plot_Kim_struct, indicator_order = m1_indicator_order,
                                    indicator_factor = m1_indicator_factor, factor_layout = m1_msmt, factor_point_to = m1_point_to,
                                    indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)

# changing node labels
```

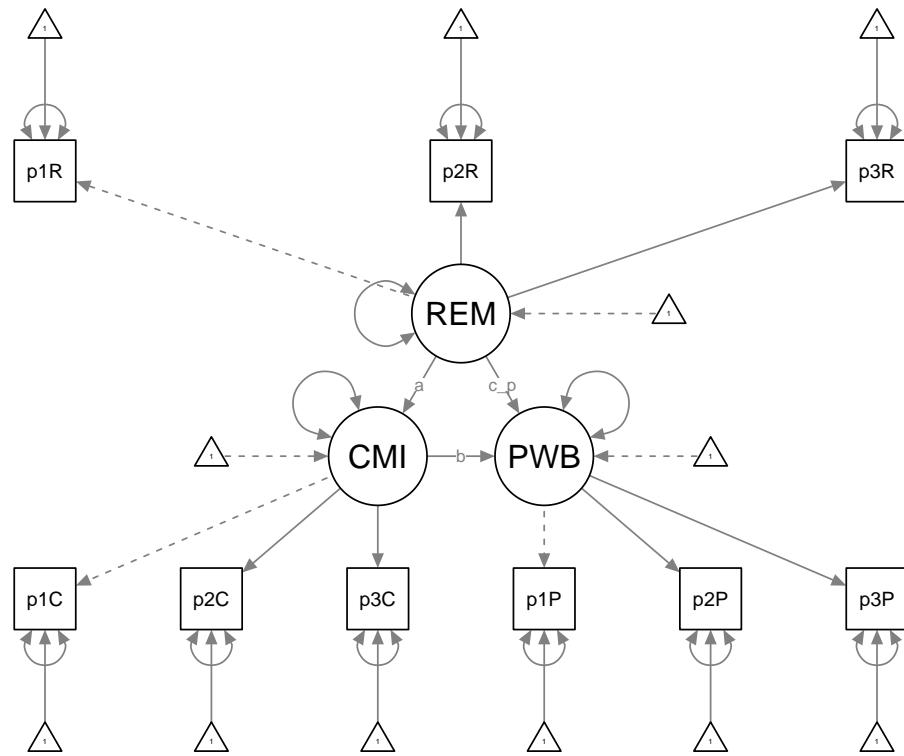
```
plot1 <- semtools::change_node_label(plot1, c(REM = "mAggress", CMI = "cMistrust",
    PWB = "Wellbeing"), label.cex = 1.1)

plot(plot1)
```

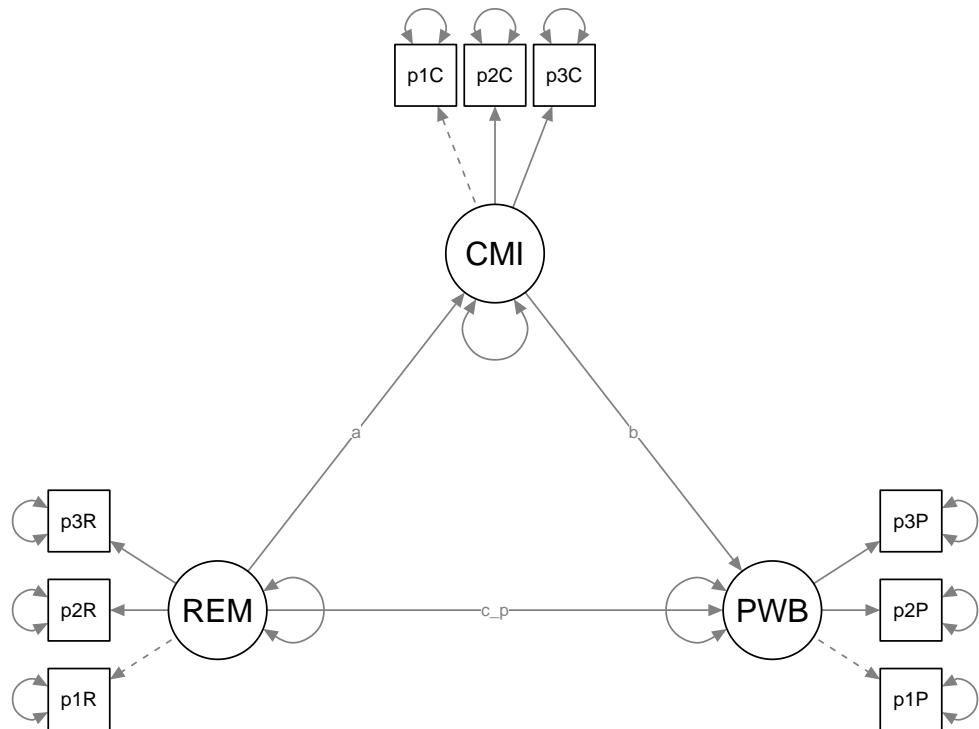


It can be useful to have a representation of the model without the results. This set of code produces those results. It does so by including only the name of the fitted object into the `semPlot::semPaths` function. Then it uses all the objects we just created as instructions for the figure's appearance.

```
# Code to plot the theoretical model (in case you don't want to print
# the results on the graph):
p_theoretical <- semPlot::semPaths(Kim_struct_fit)
```



```
p_theoretical <- semtools::set_sem_layout(p_theoretical, indicator_order = m1_indicator_order
  indicator_factor = m1_indicator_factor, factor_layout = m1_msmt, factor_point_to = m1_point
  indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)
plot(p_theoretical)
```



With tables and figures prepared, we can write up the results.

10.8.2 APA Style Write-up of the Results

Method/Analytic Strategy

We specified a structural equation model predicting psychological well-being (PWB) from racial microaggressions (REM), mediated by cultural mistrust (CMI). The primary analysis occurred in two stages. First, we specified and evaluated a measurement model. Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16).

Results

Preliminary Analyses

- Missing data analyses and managing missing data>

- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

Primary Analyses Analyzing our proposed multiple mediator model followed the two-step procedure of first evaluating a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index (CFI) ≥ 0.95 , root-mean-square error of approximation (RMSEA) ≤ 0.06 , and the standard root-mean-square residual (SRMR) ≤ 0.08 . To establish acceptable fit, we used CFI ≥ 0.90 , RMSEA ≤ 0.10 , and SRMR ≤ 0.10 [Weston and Gore, 2006].

We evaluated the measurement model by following recommendations by Little et al. [Little et al., 2002, 2013]. Specifically, each latent variable was represented by three parcels. Parcels were created by randomly assigning scale items to the parcels and then calculating the mean, if at least 65% of the items were non-missing. Factor loadings were all strong, statistically significant, and properly valenced. Global fit statistics were within acceptable thresholds ($\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$). Factor loadings associated with the parcelled measurement model are presented in Table 1 (i.e., [in the previous lesson](#)). Thus, we proceeded to testing the structural model.

Our structural model was a simple mediation, predicting psychological well-being (PWB) from racial microaggressions (REM), mediated by cultural mistrust (CMI). Given the just-identified nature (i.e., zero degrees of freedom in the structural portion), the global fit indices were identical to those of the measurement model ($\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$). As shown in Table 2, all regression weights were statistically significant. Further, there was a statistically significant indirect effect ($B = -0.236, p = 0.002$), supporting the hypothesis that cultural mistrust is a mediating mechanism for racial microaggressions' negative impact on psychological well-being. That is, racial microaggressions increase cultural mistrust ($B = 1.732, p < 0.001$) which in turn has a negative impact on psychological well-being ($B = -0.136, p = 0.001$). Results are illustrated in Figure 1.

10.9 What About Alternative Models?

The Kim et al. [2017] article is extra-excellent because it addressed concerns about alternative models. The authors tested two alternative models. In the first they switched the role of mediating and dependent variables. In the second, they predicted cultural mistrust and psychological well-being from racial microaggressions. The additional testing they did is rather serendipitous for our lesson because it allows us to address issues of model equivalency.

Because we will use the same three variables across these three analyses, there is no need to respecify the measurement model. Further, when we simply add, delete, or swap paths we have a circumstance of *nested models*. This allows us to formally compare the global fit indices of our models ($\Delta\chi^2, \Delta CFI, AIC, BIC$).

10.9.1 Swapping the Mediator and the Outcome

The first alternative model predicted cultural mistrust from racial microaggressions, mediated by psychological well-being.

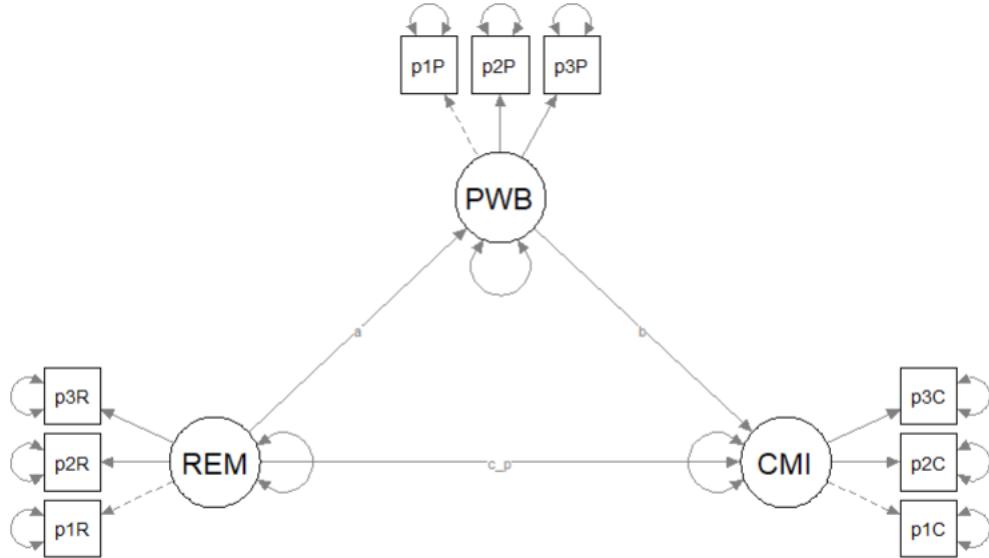


Figure 10.4: Image of the proposed alternative – a simple mediation that swaps PWB and CMI

```

Kim_swap_mod <- "
  ##measurement model
  REMS =~ p1REMS + p2REMS + p3REMS
  CMI =~ p1CMI + p2CMI + p3CMI
  PWB =~ p1PWB + p2PWB + p3PWB

  #structural model with labels for calculation of the indirect effect
  CMI ~ b*PWB + c_p*REMS
  PWB ~a*REMS

  #calculations
  indirect := a*b
  direct := c_p
  total_c := c_p + (a*b)
  "

set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
Kim_swap_fit <- lavaan::sem(Kim_swap_mod, data = dfKim, missing = "fiml")
Kim_swap_summary <- lavaan::summary(Kim_swap_fit, fit.measures = TRUE,
  standardized = TRUE, rsq = TRUE)
Kim_swap_pEsts <- lavaan::parameterEstimates(Kim_swap_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
Kim_swap_summary
  
```

```

## lavaan 0.6.16 ended normally after 100 iterations
##
##   Estimator                      ML
## Optimization method            NLMINB
## Number of model parameters    30
##
##   Number of observations        156
##   Number of missing patterns    1
##
## Model Test User Model:
##
##   Test statistic                 15.965
##   Degrees of freedom              24
##   P-value (Chi-square)           0.889
##
## Model Test Baseline Model:
##
##   Test statistic                 1366.804
##   Degrees of freedom                36
##   P-value                           0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)      1.000
##   Tucker-Lewis Index (TLI)          1.009
##
##   Robust Comparative Fit Index (CFI) 1.000
##   Robust Tucker-Lewis Index (TLI)    1.009
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)     -368.149
##   Loglikelihood unrestricted model (H1) -360.166
##
##   Akaike (AIC)                     796.297
##   Bayesian (BIC)                   887.793
##   Sample-size adjusted Bayesian (SABIC) 792.834
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                          0.000
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.031
##   P-value H_0: RMSEA <= 0.050       0.988
##   P-value H_0: RMSEA >= 0.080       0.000
##
##   Robust RMSEA                     0.000
##   90 Percent confidence interval - lower 0.000

```

```

## 90 Percent confidence interval - upper          0.031
## P-value H_0: Robust RMSEA <= 0.050          0.988
## P-value H_0: Robust RMSEA >= 0.080          0.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                               0.017
##
## Parameter Estimates:
##
## Standard errors                      Standard
## Information                           Observed
## Observed information based on        Hessian
##
## Latent Variables:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## REMS =~
##   p1REMS      1.000
##   p2REMS      1.055    0.044 23.770  0.000  0.286  0.946
##   p3REMS      1.031    0.046 22.399  0.000  0.280  0.929
## CMI =~
##   p1CMI      1.000
##   p2CMI      1.006    0.035 28.780  0.000  0.778  0.957
##   p3CMI      0.937    0.034 27.373  0.000  0.724  0.948
## PWB =~
##   p1PWB      1.000
##   p2PWB      1.606    0.228  7.055  0.000  0.491  0.801
##   p3PWB      1.372    0.206  6.662  0.000  0.420  0.709
##
## Regressions:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CMI ~
##   PWB       (b) -0.804    0.252 -3.197  0.001 -0.318 -0.318
##   REMS      (c_p) 1.223    0.243  5.027  0.000  0.429  0.429
## PWB ~
##   REMS      (a) -0.633    0.114 -5.557  0.000 -0.562 -0.562
##
## Intercepts:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS      0.494    0.023 21.448  0.000  0.494  1.717
## .p2REMS      0.504    0.024 20.790  0.000  0.504  1.665
## .p3REMS      0.522    0.024 21.642  0.000  0.522  1.733
## .p1CMI       3.888    0.064 60.479  0.000  3.888  4.842
## .p2CMI       3.886    0.065 59.667  0.000  3.886  4.777
## .p3CMI       4.051    0.061 66.190  0.000  4.051  5.299
## .p1PWB       3.014    0.039 78.207  0.000  3.014  6.262
## .p2PWB       3.207    0.049 65.324  0.000  3.207  5.230
## .p3PWB       3.141    0.047 66.276  0.000  3.141  5.306

```

```

##      REMS          0.000          0.000          0.000
##      .CMI          0.000          0.000          0.000
##      .PWB          0.000          0.000          0.000
##
## Variances:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##      .p1REMS     0.009  0.002   5.466  0.000  0.009  0.109
##      .p2REMS     0.010  0.002   5.330  0.000  0.010  0.106
##      .p3REMS     0.012  0.002   6.278  0.000  0.012  0.137
##      .p1CMI      0.047  0.009   5.046  0.000  0.047  0.072
##      .p2CMI      0.056  0.010   5.597  0.000  0.056  0.084
##      .p3CMI      0.060  0.009   6.292  0.000  0.060  0.102
##      .p1PWB      0.138  0.019   7.309  0.000  0.138  0.597
##      .p2PWB      0.135  0.029   4.646  0.000  0.135  0.358
##      .p3PWB      0.174  0.028   6.320  0.000  0.174  0.498
##      REMS        0.074  0.009   7.838  0.000  1.000  1.000
##      .CMI        0.336  0.044   7.557  0.000  0.562  0.562
##      .PWB        0.064  0.017   3.778  0.000  0.685  0.685
##
## R-Square:
##           Estimate
##      p1REMS     0.891
##      p2REMS     0.894
##      p3REMS     0.863
##      p1CMI      0.928
##      p2CMI      0.916
##      p3CMI      0.898
##      p1PWB      0.403
##      p2PWB      0.642
##      p3PWB      0.502
##      CMI        0.438
##      PWB        0.315
##
## Defined Parameters:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##      indirect    0.509  0.166   3.066  0.002  0.179  0.179
##      direct      1.223  0.243   5.027  0.000  0.429  0.429
##      total_c     1.732  0.199   8.698  0.000  0.608  0.608

# Kim_swap_pEsts #although creating the object is useful to export as
# a .csv I didn't ask it to print into the book

```

Here is the script to export results to .csv files.

```

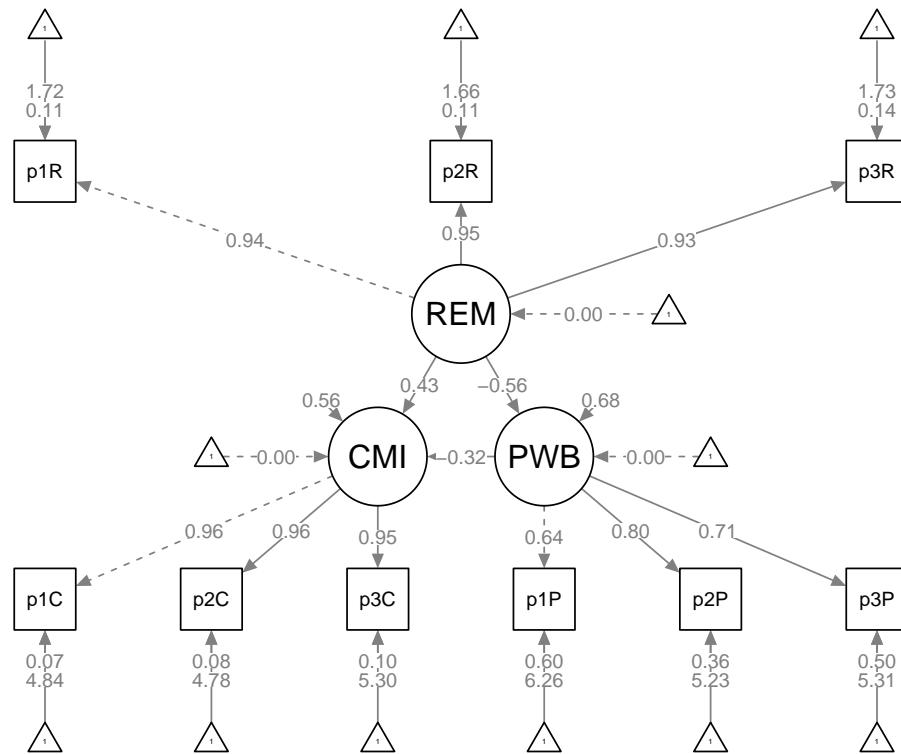
# global fit indices
Kim_swap_fitstats <- tidySEM:::table_fit(Kim_swap_fit)
write.csv(Kim_swap_fitstats, file = "Kim_swap_fitstats.csv")

```

```
# the code below writes the parameter estimates into a .csv file
write.csv(Kim_swap_pEsts, file = "Kim_swap_pEsts.csv")
```

Let's immediately plot this so we have a visual to assist our conceptualization.

```
plot_Kim_swap <- semPlot::semPaths(Kim_swap_fit, what = "col", whatLabels = "stand", sizeMan =
```

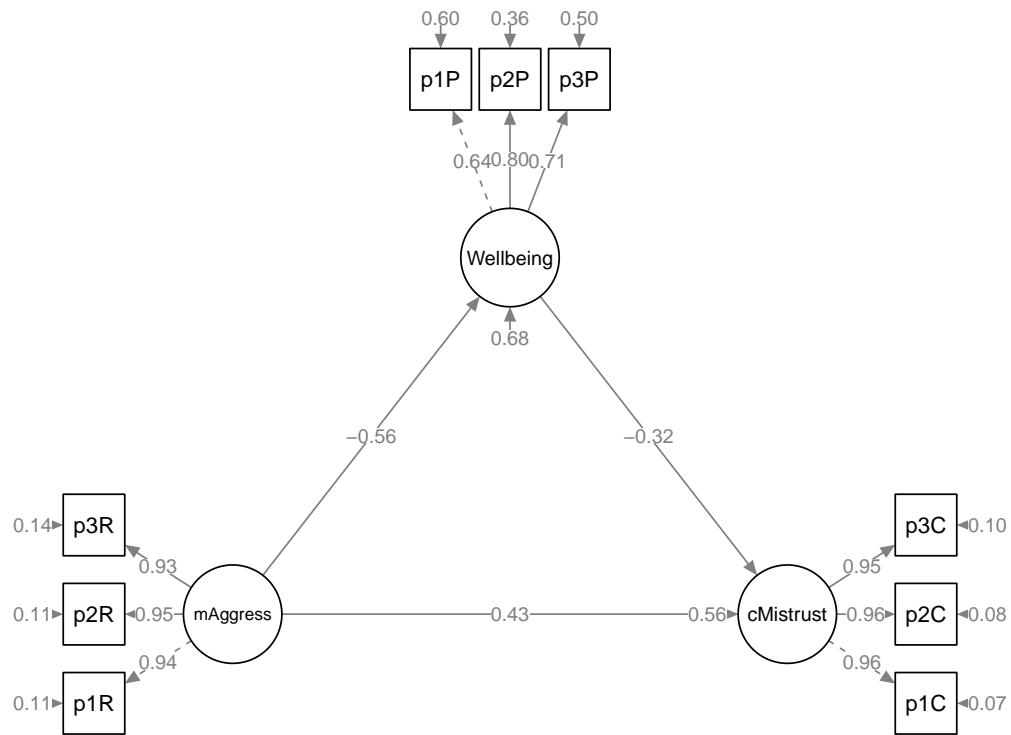


Below is the column/row mapping of my variables in their desired position.

Grid for Plotting `semplot::semPath`

(1,1) empty	(1,2) PWB	(1,3) empty
(2,1) REM	(2,2) empty	(2,3) CMI

We place these values along with the names of our latent variables in to the `semptools::layout_matrix` function.

**Table 3**

The Swap Model Coefficients Assessing the Effect of REMS on CMI through PWB

Predictor	<i>B</i>	<i>SE_B</i>	<i>p</i>	β	<i>R</i> ²
Psychological well-being (M)					.34
Racial microaggressions (<i>a</i>)	-0.633	0.114	< 0.001	-0.562	
Cultural mistrust (DV)					.44
Racial microaggressions (<i>b</i>)	-0.804	0.252	0.001	-0.318	
Psychological well-being (<i>c'</i>)	1.223	0.243	< 0.001	0.429	

Effects	<i>B</i>	<i>SE_B</i>	<i>p</i>	95% CI	
Indirect(<i>a</i> * <i>b</i>)	0.509	0.166	0.002	0.179	0.184, 0.834
Total effect	1.732	0.199	0.000	0.608	1.342, 2.122

Effects	B	SE _B	p	95% CI
---------	---	-----------------	---	--------

Note. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

Our string of global fit indices is: $\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$.

While the values of the individual paths as well as the indirect and total effects differ, the just-identified nature of this and the original model mean that global fit indices will be the identical. This problem of *model identification* is a known and troubling one that contributes to *equivalent models*. Researchers are challenged to have strong justification via research design and theory to support the models they have chosen and the results they are reporting.

If you read the Kim et al. [2017] closely, you will notice that they did find differences in their estimates of global fit. Their modeling was all at the path analytic (i.e., no latent variable) level. The differences were such that they preferred their original model.

10.9.2 REMS as a Predictor of Both CMI and PWB

Kim et al. [2017] also tested a model where racial microaggressions predicted both cultural mistrust and psychological well-being. Having two, simultaneous, dependent variables is a benefit of structural equation modeling.

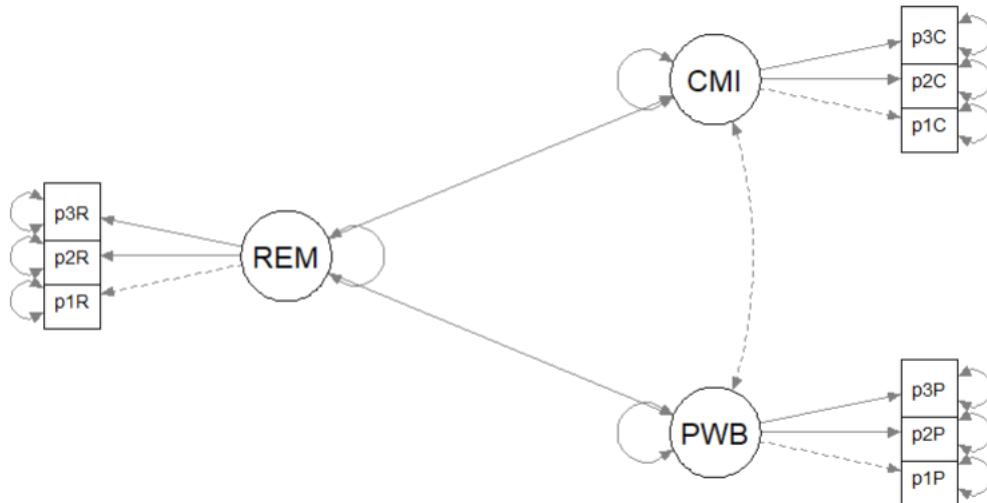


Figure 10.5: Image of another proposed alternative – a simple mediation that swaps PWB and CMI

The default in *lavaan::sem* is to allow all variables to covary if they aren't already involved in a directional relationship. To fix their relationship to zero we need to add “fixed.x = FALSE” to the *lavaan::summary* function and add “CMI~~0*PWB” to the model syntax.

In this model, fixing the relationship between CMI and REMS to zero will allow us to have an unsaturated, over-identified, model with +1 degrees of freedom. A priorily we know that the fit will be worse in this model than in the saturated measurement models and simple mediations.

```
Kim_2dvs_mod <- "
  ##measurement model
  REMS =~ p1REMS + p2REMS + p3REMS
  CMI =~ p1CMI + p2CMI + p3CMI
  PWB =~ p1PWB + p2PWB + p3PWB

  #structural model with labels for calculation of the indirect effect
  CMI ~ REMS
  PWB ~ REMS

  CMI ~~ 0*PWB
  ""

set.seed(230916)  #needed for reproducibility especially when specifying bootstrapped confidence intervals
Kim_2dvs_fit <- lavaan::sem(Kim_2dvs_mod, data = dfKim, missing = "fiml",
  fixed.x = FALSE)
Kim_2dvs_summary <- lavaan::summary(Kim_2dvs_fit, fit.measures = TRUE,
  standardized = TRUE, rsq = TRUE)
Kim_2dvs_pEsts <- lavaan::parameterEstimates(Kim_2dvs_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
Kim_2dvs_summary

## lavaan 0.6.16 ended normally after 101 iterations
##
##             ML
## Estimator
## Optimization method      NLMINB
## Number of model parameters    29
## 
##             156
## Number of observations
## Number of missing patterns      1
## 
## Model Test User Model:
## 
##             27.398
## Test statistic
## Degrees of freedom          25
## P-value (Chi-square)        0.336
## 
## Model Test Baseline Model:
## 
##             1366.804
## Test statistic
## Degrees of freedom          36
## P-value                      0.000
## 
## User Model versus Baseline Model:
```

```

##  

## Comparative Fit Index (CFI) 0.998  

## Tucker-Lewis Index (TLI) 0.997  

##  

## Robust Comparative Fit Index (CFI) 0.998  

## Robust Tucker-Lewis Index (TLI) 0.997  

##  

## Loglikelihood and Information Criteria:  

##  

## Loglikelihood user model (H0) -373.865  

## Loglikelihood unrestricted model (H1) -360.166  

##  

## Akaike (AIC) 805.731  

## Bayesian (BIC) 894.177  

## Sample-size adjusted Bayesian (SABIC) 802.382  

##  

## Root Mean Square Error of Approximation:  

##  

## RMSEA 0.025  

## 90 Percent confidence interval - lower 0.000  

## 90 Percent confidence interval - upper 0.070  

## P-value H_0: RMSEA <= 0.050 0.773  

## P-value H_0: RMSEA >= 0.080 0.018  

##  

## Robust RMSEA 0.025  

## 90 Percent confidence interval - lower 0.000  

## 90 Percent confidence interval - upper 0.070  

## P-value H_0: Robust RMSEA <= 0.050 0.773  

## P-value H_0: Robust RMSEA >= 0.080 0.018  

##  

## Standardized Root Mean Square Residual:  

##  

## SRMR 0.059  

##  

## Parameter Estimates:  

##  

## Standard errors Standard  

## Information Observed  

## Observed information based on Hessian  

##  

## Latent Variables:  

## Estimate Std.Err z-value P(>|z|) Std.lv Std.all  

## REMS =~  

## p1REMS 1.000  

## p2REMS 1.056 0.044 23.765 0.000 0.271 0.943  

## p3REMS 1.031 0.046 22.339 0.000 0.280 0.946  

## CMI =~  

## p1CMI 1.000  


```

```

##      p2CMI          1.006    0.035   28.748   0.000    0.778    0.957
##      p3CMI          0.936    0.034   27.395   0.000    0.724    0.948
## PWB =~
##      p1PWB          1.000
##      p2PWB          1.561    0.223   6.988   0.000    0.481    0.785
##      p3PWB          1.387    0.211   6.583   0.000    0.427    0.721
##
## Regressions:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CMI ~
##      REMS           1.750    0.199   8.806   0.000    0.613    0.613
## PWB ~
##      REMS          -0.652    0.114  -5.696   0.000   -0.574   -0.574
##
## Covariances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CMI ~~
##      .PWB           0.000
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS        0.494    0.023  21.448   0.000    0.494    1.717
## .p2REMS        0.504    0.024  20.790   0.000    0.504    1.665
## .p3REMS        0.522    0.024  21.642   0.000    0.522    1.733
## .p1CMI         3.888    0.064  60.479   0.000    3.888    4.842
## .p2CMI         3.886    0.065  59.667   0.000    3.886    4.777
## .p3CMI         4.051    0.061  66.190   0.000    4.051    5.299
## .p1PWB         3.014    0.039  78.207   0.000    3.014    6.262
## .p2PWB         3.207    0.049  65.324   0.000    3.207    5.230
## .p3PWB         3.141    0.047  66.276   0.000    3.141    5.306
##      REMS          0.000
##      .CMI           0.000
##      .PWB           0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1REMS        0.009    0.002   5.557   0.000    0.009    0.111
## .p2REMS        0.010    0.002   5.356   0.000    0.010    0.105
## .p3REMS        0.012    0.002   6.319   0.000    0.012    0.137
## .p1CMI         0.046    0.009   5.001   0.000    0.046    0.072
## .p2CMI         0.056    0.010   5.616   0.000    0.056    0.085
## .p3CMI         0.060    0.010   6.267   0.000    0.060    0.102
## .p1PWB         0.137    0.019   7.169   0.000    0.137    0.591
## .p2PWB         0.145    0.030   4.827   0.000    0.145    0.385
## .p3PWB         0.168    0.028   5.972   0.000    0.168    0.479
##      REMS          0.074    0.009   7.829   0.000    1.000    1.000
##      .CMI          0.373    0.047   7.864   0.000    0.624    0.624
##      .PWB          0.064    0.017   3.748   0.000    0.671    0.671

```

```

## R-Square:
##                               Estimate
## p1REMS                 0.889
## p2REMS                 0.895
## p3REMS                 0.863
## p1CMI                  0.928
## p2CMI                  0.915
## p3CMI                  0.898
## p1PWB                  0.409
## p2PWB                  0.615
## p3PWB                  0.521
## CMI                     0.376
## PWB                     0.329

# Kim_2dvs_pEsts#although creating the object is useful to export as
# a .csv I didn't ask it to print into the book

```

Here is the script to export results to .csv files.

```

# global fit indices
Kim_2dvs_fitstats <- tidySEM:::table_fit(Kim_2dvs_fit)
write.csv(Kim_2dvs_fitstats, file = "Kim_2dvs_fitstats.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(Kim_2dvs_pEsts, file = "Kim_2dvs_pEsts.csv")

```

Table 4

The Two DVs Model: Predicting PWB from CMI and REMS

Predictor	B	SE _B	p	β	R ²
Cultural mistrust (DV1)					.38
Racial microaggressions	1.750	0.199	< 0.001	0.613	
Psychological Well-Being (DV2)					.33
Racial microaggressions	-0.652	0.114	< 0.001	-0.573	

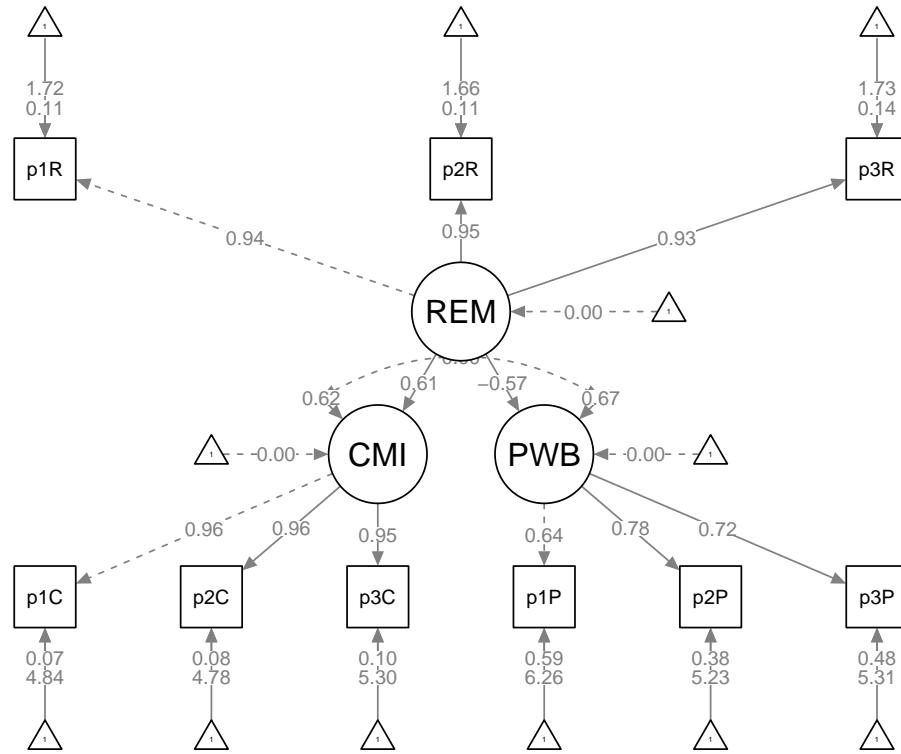
Our string of global fit indices is: $\chi^2(25) = 27.398, p = 0.336, CFI = 0.998, RMSEA = 0.025, 90CI[0.000, 0.070], SRMR = 0.059$. While the fit statistics are acceptable, they are slightly less strong than the fully saturated models with indirect effects.

I will use the techniques explained above to create a figure.

```

plot_Kim_2dvs <- semPlot::semPaths(Kim_2dvs_fit, what = "col", whatLabels = "stand",
  sizeMan = 5, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(5, 5, 5, 5))

```

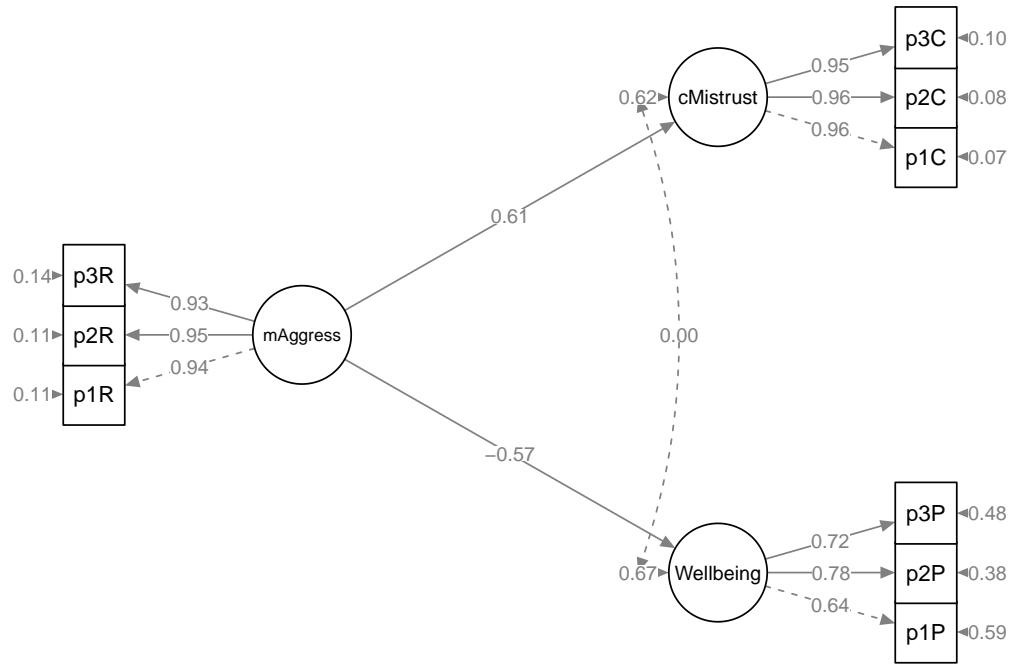


Below is the column/row mapping of my variables in their desired position.

Grid for Plotting semplot::sempath	
(1,1) empty	(1,2) CMI
(2,1) REM	(2,2) empty
(3,1) empty	(3,2) PWB

We place these values along with the names of our latent variables in to the *semp-tools::layout_matrix* function.

```
#IMPORTANT: Must use the node names (take directly from the SemPlot) assigned by SemPlot
#You can change them as the last thing
m3_msmt <- semptools::layout_matrix(REM = c(2,1),
                                      CMI = c(1,2),
                                      PWB = c(3,2))
#tell where you want the indicators to face
m3_point_to <- semptools::layout_matrix(left = c(2,1),
```

10.9.3 What if we allowed PWB and CMI to co-vary?

Let's try one last model. In this last model predicting two dependent variables, I fixed the relationship between CMI and PWB to be 0.00. That is, I did not allow them to covary/correlate. What if I freed them to relate? This will return us to a saturated, just-identified, model with a single degree of freedom. Want to guess what our fit will be?

To uncorrelate the DVs. I am removing the code that fixed the two variables to be uncorrelated. I am also changing “fixed.x = FALSE” to “fixed.x = TRUE” (which is the *lavaan* default).

```
Kim_2CorrDVs_mod <- "
  ##measurement model
  REMS =~ p1REMS + p2REMS + p3REMS
  CMI =~ p1CMI + p2CMI + p3CMI
  PWB =~ p1PWB + p2PWB + p3PWB

  #structural model with labels for calculation of the indirect effect
```

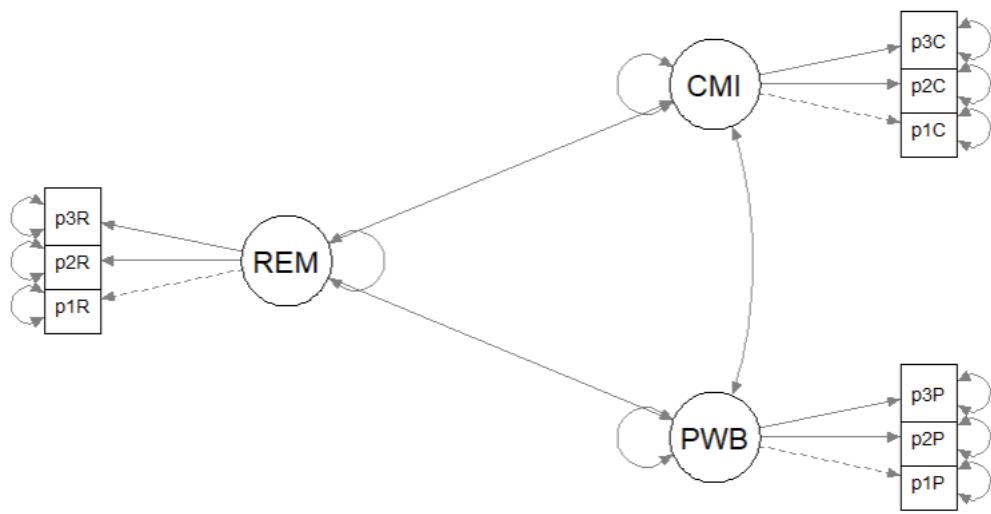


Figure 10.6: Image of another proposed alternative – a simple mediation that swaps PWB and CMI

```

CMI ~ REMS
PWB ~ REMS

##
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
Kim_2CorrDVs_fit <- lavaan::sem(Kim_2CorrDVs_mod, data = dfKim, missing = "fiml",
fixed.x = TRUE)
Kim_2CorrDVs_summary <- lavaan::summary(Kim_2CorrDVs_fit, fit.measures = TRUE,
standardized = TRUE, rsq = TRUE)
Kim_2CorrDVs_pEsts <- lavaan::parameterEstimates(Kim_2CorrDVs_fit, boot.ci.type = "bca.simple",
standardized = TRUE)
Kim_2CorrDVs_summary

## lavaan 0.6.16 ended normally after 99 iterations
## 
##   Estimator                               ML
##   Optimization method                    NLMINB
##   Number of model parameters             30
## 
##   Number of observations                156
##   Number of missing patterns              1
## 
## Model Test User Model:
## 
##   Test statistic                         15.965
##   Degrees of freedom                      24
##   P-value (Chi-square)                   0.889

```

```

## 
## Model Test Baseline Model:
## 
##   Test statistic           1366.804
##   Degrees of freedom       36
##   P-value                  0.000
## 
## User Model versus Baseline Model:
## 
##   Comparative Fit Index (CFI)      1.000
##   Tucker-Lewis Index (TLI)        1.009
## 
##   Robust Comparative Fit Index (CFI)      1.000
##   Robust Tucker-Lewis Index (TLI)        1.009
## 
## Loglikelihood and Information Criteria:
## 
##   Loglikelihood user model (H0)      -368.149
##   Loglikelihood unrestricted model (H1) -360.166
## 
##   Akaike (AIC)                      796.297
##   Bayesian (BIC)                     887.793
##   Sample-size adjusted Bayesian (SABIC) 792.834
## 
## Root Mean Square Error of Approximation:
## 
##   RMSEA                         0.000
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.031
##   P-value H_0: RMSEA <= 0.050      0.988
##   P-value H_0: RMSEA >= 0.080      0.000
## 
##   Robust RMSEA                    0.000
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.031
##   P-value H_0: Robust RMSEA <= 0.050 0.988
##   P-value H_0: Robust RMSEA >= 0.080 0.000
## 
## Standardized Root Mean Square Residual:
## 
##   SRMR                          0.017
## 
## Parameter Estimates:
## 
##   Standard errors                Standard
##   Information                     Observed
##   Observed information based on  Hessian
## 
```

```

## Latent Variables:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   REMS =~
##     p1REMS      1.000
##     p2REMS      1.055    0.044 23.770  0.000  0.286  0.946
##     p3REMS      1.031    0.046 22.399  0.000  0.280  0.929
##   CMI =~
##     p1CMI      1.000
##     p2CMI      1.006    0.035 28.780  0.000  0.778  0.957
##     p3CMI      0.937    0.034 27.373  0.000  0.724  0.948
##   PWB =~
##     p1PWB      1.000
##     p2PWB      1.606    0.228  7.055  0.000  0.491  0.801
##     p3PWB      1.372    0.206  6.662  0.000  0.420  0.709
##
## Regressions:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   CMI ~
##     REMS       1.732    0.199  8.698  0.000  0.608  0.608
##   PWB ~
##     REMS      -0.633    0.114 -5.557  0.000 -0.562 -0.562
##
## Covariances:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   .CMI ~~
##     .PWB      -0.051    0.017 -3.039  0.002 -0.331 -0.331
##
## Intercepts:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   .p1REMS      0.494    0.023 21.448  0.000  0.494  1.717
##   .p2REMS      0.504    0.024 20.790  0.000  0.504  1.665
##   .p3REMS      0.522    0.024 21.642  0.000  0.522  1.733
##   .p1CMI       3.888    0.064 60.479  0.000  3.888  4.842
##   .p2CMI       3.886    0.065 59.667  0.000  3.886  4.777
##   .p3CMI       4.051    0.061 66.190  0.000  4.051  5.299
##   .p1PWB       3.014    0.039 78.207  0.000  3.014  6.262
##   .p2PWB       3.207    0.049 65.324  0.000  3.207  5.230
##   .p3PWB       3.141    0.047 66.276  0.000  3.141  5.306
##   REMS        0.000
##   .CMI         0.000
##   .PWB         0.000
##
## Variances:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   .p1REMS      0.009    0.002  5.466  0.000  0.009  0.109
##   .p2REMS      0.010    0.002  5.330  0.000  0.010  0.106
##   .p3REMS      0.012    0.002  6.278  0.000  0.012  0.137
##   .p1CMI       0.047    0.009  5.046  0.000  0.047  0.072

```

```

## .p2CMI          0.056   0.010   5.597   0.000   0.056   0.084
## .p3CMI          0.060   0.009   6.292   0.000   0.060   0.102
## .p1PWB          0.138   0.019   7.309   0.000   0.138   0.597
## .p2PWB          0.135   0.029   4.646   0.000   0.135   0.358
## .p3PWB          0.174   0.028   6.320   0.000   0.174   0.498
## REMS            0.074   0.009   7.838   0.000   1.000   1.000
## .CMI             0.377   0.048   7.871   0.000   0.631   0.631
## .PWB             0.064   0.017   3.778   0.000   0.685   0.685
##
## R-Square:
##                               Estimate
## p1REMS           0.891
## p2REMS           0.894
## p3REMS           0.863
## p1CMI            0.928
## p2CMI            0.916
## p3CMI            0.898
## p1PWB            0.403
## p2PWB            0.642
## p3PWB            0.502
## CMI              0.369
## PWB              0.315

```

```

# Kim_2CorrDVs_pEsts#although creating the object is useful to export
# as a .csv I didn't ask it to print into the book

```

Here is the script to export results to .csv files.

```

# global fit indices
Kim_2CorrDVs_fitstats <- tidySEM::table_fit(Kim_2CorrDVs_fit)
write.csv(Kim_2CorrDVs_fitstats, file = "Kim_2CorrDVs_fitstats.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(Kim_2CorrDVs_pEsts, file = "Kim_2CorrDVs_pEsts.csv")

```

Table 4

The Two DVs Model: Predicting PWB from CMI and REMS

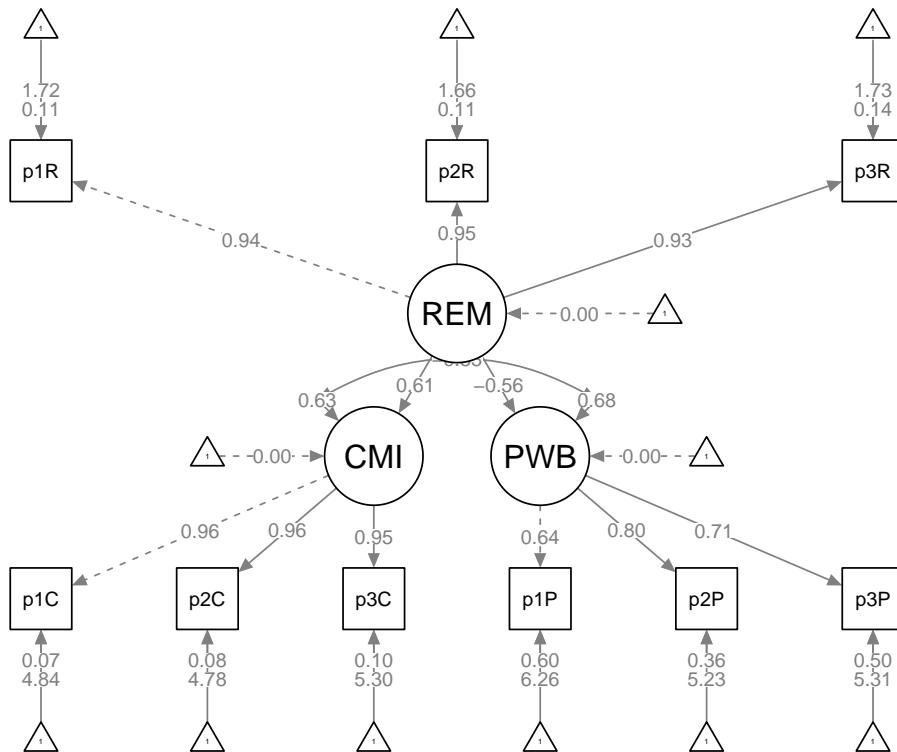
Predictor	B	SE _B	p	β	R ²
Cultural mistrust (DV1)					.38
Racial microaggressions	1.732	0.199	< 0.001	0608	
Psychological Well-Being (DV2)					.33
Racial microaggressions	-0.633	0.114	< 0.001	-0.562	

Note. In this model, CMI and PWB were free to covary. The correlation between these two variables was -0.56.

Our string of global fit indices has returned to the same values as the previously mediated models. The commonality between the three models is that they are all just-identified. That is, the structural paths are fully saturated with a single degree of freedom: $\chi^2(24) = 15.965, p = 0.889, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.031], SRMR = 0.017$.

I will use the techniques explained above to create a figure.

```
plot_Kim_2CorrDVs <- semPlot::semPaths(Kim_2CorrDVs_fit, what = "col",
                                         whatLabels = "stand", sizeMan = 5, node.width = 1, edge.label.cex = 0.75,
                                         style = "lisrel", mar = c(5, 5, 5, 5))
```



Below is the column/row mapping of my variables in their desired position.

Grid for Plotting semplot::sempath

(1,1) empty	(1,2) CMI
(2,1) REM	(2,2) empty
(3,1) empty	(3,2) PWB

We place these values along with the names of our latent variables in to the *semptools::layout_matrix* function.

```
#IMPORTANT: Must use the node names (take directly from the SemPlot)
#You can change them as the last thing
m4_msmt <- semptools::layout_matrix(REM = c(2,1),
                                    CMI = c(1,2),
                                    PWB = c(3,2))
#tell where you want the indicators to face
m4_point_to <- semptools::layout_matrix (left = c(2,1),
                                         right = c(1,2),
                                         right = c(3,2))

#the next two codes -- indicator_order and indicator_factor are paired together, they specify
m4_indicator_order <- c("p1R", "p2R", "p3R",
                        "p1C", "p2C", "p3C",
                        "p1P", "p2P", "p3P")

m4_indicator_factor <- c("REM", "REM", "REM",
                          "CMI", "CMI", "CMI",
                          "PWB", "PWB", "PWB")

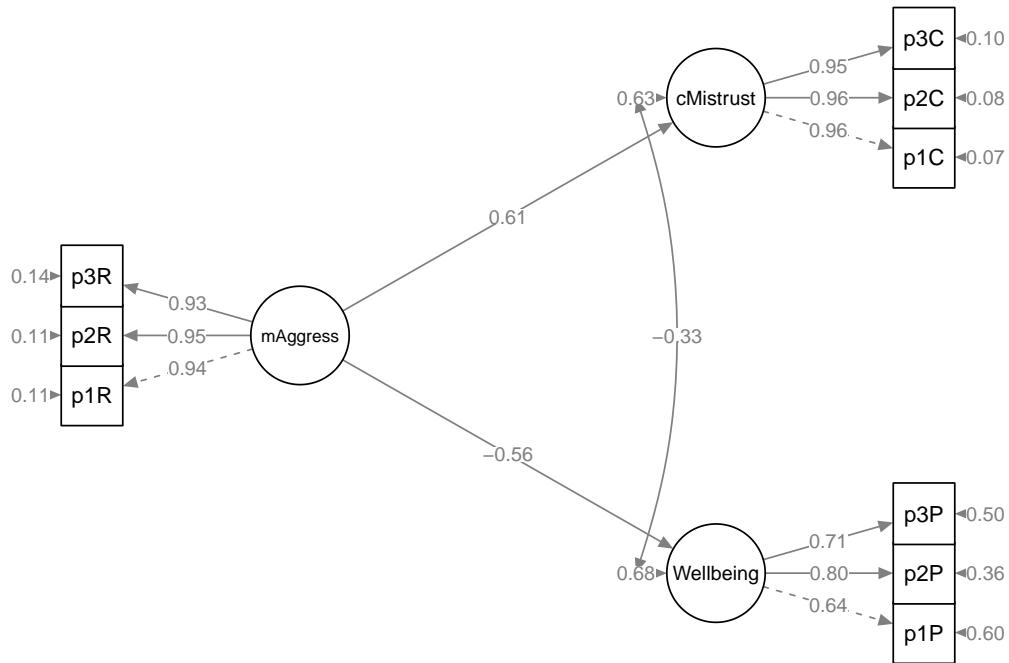
#next set of code pushes the indicator variables away from the factor
m4_indicator_push <- c(REM = .5,
                       CMI = 1,
                       PWB = 1)
m4_indicator_spread <- c(REM = 1, #spreading the boxes away from each other
                           CMI = 1,
                           PWB = 1)

#assemble all the instructions
plot4 <- semptools::set_sem_layout(plot_Kim_2CorrDVs,
                                    indicator_order = m4_indicator_order,
                                    indicator_factor = m4_indicator_factor,
                                    factor_layout = m4_msmt,
                                    factor_point_to = m4_point_to,
                                    indicator_push = m4_indicator_push,
                                    indicator_spread = m4_indicator_spread)

#changing node labels
plot4 <- semptools::change_node_label(plot4,
```

```
c(REM = "mAggress",
  CMI = "cMistrust",
  PWB = "Wellbeing"),
label.cex = 1.1)

plot(plot4)
```



10.9.4 Model comparisons

Because these models all use the same data and same variables, they are nested. That is, they add and delete paths (or change directional paths to bidirectional covariances). As a result we can compare their fit. I have used the *lavaan::lavTestLRT* function to demonstrate that with the exception of the “Kim_2dvs_fit” (the model that predicted CMI and PWB from REMS and did not allow PWB and REMS to covary), all model fit was the same. This is because of the just-identified, fully saturated nature of these models.

```

lavaan::lavTestLRT(rp3_msmt_fit, Kim_struct_fit, Kim_swap_fit, Kim_2dvs_fit,
  Kim_2CorrDVs_fit)

## Warning in lavaan::lavTestLRT(rp3_msmt_fit, Kim_struct_fit, Kim_swap_fit, :
## lavaan WARNING: some models have the same degrees of freedom

## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)
## rp3_msmt_fit  24 796.30 887.79 15.965
## Kim_struct_fit 24 796.30 887.79 15.965      0.000 0.00000      0
## Kim_swap_fit   24 796.30 887.79 15.965      0.000 0.00000      0
## Kim_2CorrDVs_fit 24 796.30 887.79 15.965      0.000 0.00000      0
## Kim_2dvs_fit    25 805.73 894.18 27.398     11.433 0.25861      1 0.0007214
##
## rp3_msmt_fit
## Kim_struct_fit
## Kim_swap_fit
## Kim_2CorrDVs_fit
## Kim_2dvs_fit    ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Note the line of output associated with *Kim_2dvs_fit*. This is the model (predicting two CMI and PWB from REM). As shown in this code, this model had 1 degree of freedom. In this test $p < 0.001$. This means that the fit of this model is statistically significantly worse than all of the just-identified models (where $df = 0$). While we might claim victory on all these other models, we must be careful. More saturated models (the nested models with fewer degrees of freedom that has more “sticks” [directional paths and bidirectional covariances]) will always have better fit. Allowing everything to relate, simply replicates the sample covariance matrix and doesn’t say much more than, “Yep. Everything is related.”

As we move into more complex models, we will work with over-identified models with positive degrees of freedom. We will then have the opportunity to engage in model trimming and model building.

10.10 STAY TUNED

A section on power analysis is planned and coming soon! My apologies that it’s not quite Ready.

10.11 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 test a

structural model that includes a simple mediation. Overall you will (a) start with an established measurement model, (b) test a structural model, (c) test a competing alternate model, (d) compare the fit of the initial and alternate models.

10.11.1 Problem #1: Change the random seed

Simply change the random seed in the data simulation, then rework one of the simple mediations evaluated in this chapter (i.e., REMS to PWB through CMI; or REMS to CMI through PWB).

10.11.2 Problem #2: Swap one or more of the variables

The Kim et al. [2017] study included additional variables: anxiety, depression, and helping-seeking. Consider substituting one or more variables with one of these.

10.11.3 Problem #3: Try something entirely new.

Conduct a hybrid analysis 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).

Regardless of your choice(s) complete all the elements listed in the grading rubric.

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. Identify the structural model you will evaluate. It should have a minimum of three variables and could be one of the prior path-level models you already examined.	5	_____
2. Import the data and format the variables in the model.	5	_____
3. Specify and evaluate a <i>measurement</i> model that you have established.	10	_____
4. Specify and evaluate a <i>structural</i> model.	10	_____
5. Respecify and evaluate an <i>alternative</i> structural model.	10	_____
6. Conduct a formal comparison of <i>global fit</i> .	5	_____
7. APA style results with table(s) and figure(s).	5	_____
8. Explanation to grader.	5	_____
Totals	55	_____

10.12 Homeworked Example

[Screencast Link](#)

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to evaluate the measurement model that would precede the evaluation of a structural model. And actually, we will need to evaluate two measurement models – an “all items” on indicators model and a parceled model.

Identify the structural model you will evaluate

It should have a minimum of three variables and could be one of the prior path-level models you already examined

I will repeat the simple mediation that I suggested in path analysis. Specifically, I hypothesize that the evaluation of socially responsive pedagogy will be predicted by intentional recentering through traditional pedagogy.

X = Centering: explicit recentering (0 = precentered; 1 = recentered) M = TradPed: traditional pedagogy (continuously scaled with higher scores being more favorable) Y = SRPed: socially responsive pedagogy (continuously scaled with higher scores being more favorable)

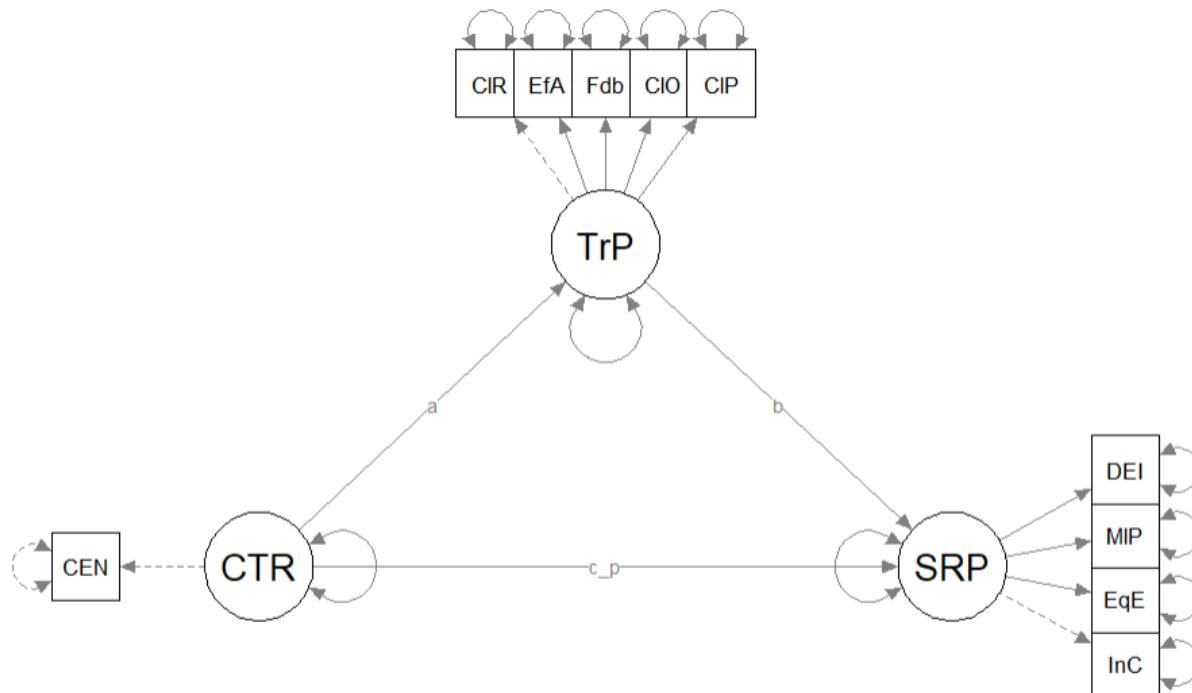


Figure 10.7: An image of the our hypothesized structural model

Specify a research model

I am hypothesizing that the evaluation of social responsive pedagogy is predicted by intentional recentering through traditional pedagogy.

Import the data and format the variables in the model

```
raw <- readRDS("ReC.rds")
```

I don't need to score my scales, but it is important to know what they are:

TradPed has 5 items: ClearResponsibilities, EffectiveAnswers, Feedback, ClearOrganization, ClearPresentation
SRPed has 4 items: InclusvClassrm, EquitableEval, MultPerspectives, DEIIntegration
Centering is 1 item – it's a factor with two levels pre, re.

I can create a babydf with just those items.

```
babydf <- dplyr::select(raw, Centering, ClearResponsibilities, EffectiveAnswers,
  Feedback, ClearOrganization, ClearPresentation, InclusvClassrm, EquitableEval,
  MultPerspectives, DEIIntegration)
```

Let's check the structure of the variables:

```
str(babydf)
```

```
## Classes 'data.table' and 'data.frame': 310 obs. of 10 variables:
## $ Centering : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ ClearResponsibilities: int 5 5 4 4 5 4 5 4 4 5 ...
## $ EffectiveAnswers : int 5 3 5 3 5 3 4 3 2 3 ...
## $ Feedback : int 5 3 4 2 5 NA 5 4 4 5 ...
## $ ClearOrganization : int 3 4 3 4 4 4 5 4 4 5 ...
## $ ClearPresentation : int 4 4 4 2 5 3 4 4 4 5 ...
## $ InclusvClassrm : int 5 5 5 5 5 4 5 5 4 5 ...
## $ EquitableEval : int 5 5 3 5 5 3 5 5 3 5 ...
## $ MultPerspectives : int 5 5 4 5 5 4 5 5 5 5 ...
## $ DEIIntegration : int 5 5 5 5 5 4 5 5 5 5 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

The centering variable will need to be dummy coded as 0/1:

```
babydf$CENTERING <- as.numeric(babydf$Centering)
babydf$CENTERING <- (babydf$CENTERING - 1)
str(babydf)
```

```

## Classes 'data.table' and 'data.frame': 310 obs. of 11 variables:
## $ Centering : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ ClearResponsibilities: int 5 5 4 4 5 4 5 4 5 ...
## $ EffectiveAnswers : int 5 3 5 3 5 3 4 3 2 3 ...
## $ Feedback : int 5 3 4 2 5 NA 5 4 4 5 ...
## $ ClearOrganization : int 3 4 3 4 4 4 5 4 4 5 ...
## $ ClearPresentation : int 4 4 4 2 5 3 4 4 4 5 ...
## $ InclusvClassrm : int 5 5 5 5 5 4 5 5 4 5 ...
## $ EquitableEval : int 5 5 3 5 5 3 5 5 3 5 ...
## $ MultPerspectives : int 5 5 4 5 5 4 5 5 5 5 ...
## $ DEIintegration : int 5 5 5 5 5 4 5 5 5 5 ...
## $ CENTERING : num 1 1 1 1 1 1 1 1 1 1 ...
## - attr(*, ".internal.selfref")=<externalptr>

```

Specify and evaluate a measurement model that you have established

That is, it is not necessary to repeat all the steps of the prior lesson's instructions to first specify a model with all indicators and then compare it to one that is more parsimonious. In the prior lesson, I decided that the unique characteristics of the items (5 indicators for TradPED, 4 indicators for SRPed) I would use all indicators for each variable. Here is a diagram of what we are specifying:

As shown below, the centering variable is a single item indicator.

```

msmt_mod <- "
  ##measurement model
  CTR =~ CENTERING #this is a single item indicator, I had to add code below to set the
  TradPed =~ ClearResponsibilities + EffectiveAnswers + Feedback + ClearOrganization +
  SRPed =~ InclusvClassrm + EquitableEval + MultPerspectives + DEIintegration

  # Variance of the single item indicator
  CENTERING ~~ 0*CENTERING

  # Covariances
  CTR ~~ TradPed
  CTR ~~ SRPed
  TradPed ~~ SRPed
  ""

set.seed(230916)
msmt_fit <- lavaan::cfa(msmt_mod, data = babydf, missing = "fiml")
msmt_fit_sum <- lavaan::summary(msmt_fit, fit.measures = TRUE, standardized = TRUE)
msmt_fit_sum

## lavaan 0.6.16 ended normally after 63 iterations
##
```

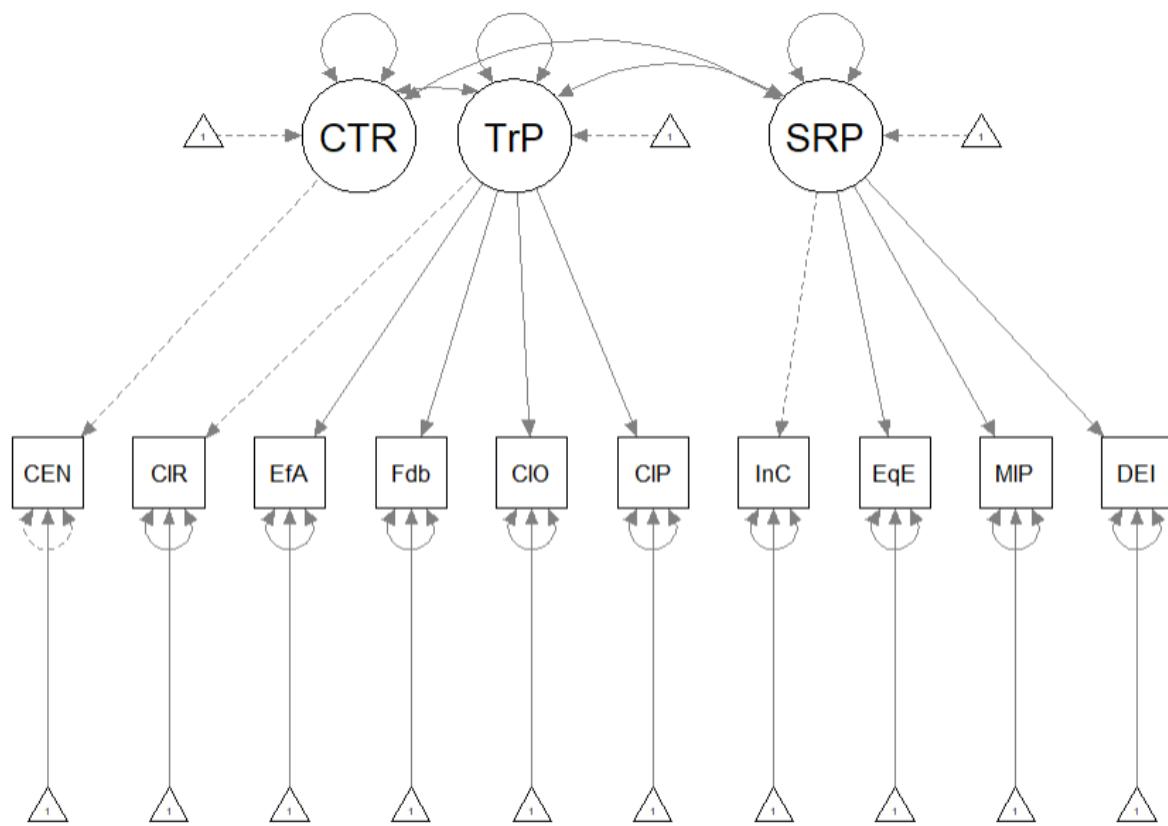


Figure 10.8: An image of the measurement model we will specify

```

## Estimator                                ML
## Optimization method                      NLMINB
## Number of model parameters                32
##
## Number of observations                   310
## Number of missing patterns               13
##
## Model Test User Model:
##
## Test statistic                           178.307
## Degrees of freedom                      33
## P-value (Chi-square)                    0.000
##
## Model Test Baseline Model:
##
## Test statistic                           1680.505
## Degrees of freedom                      45
## P-value                                 0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)             0.911
## Tucker-Lewis Index (TLI)                0.879
##
## Robust Comparative Fit Index (CFI)      0.906
## Robust Tucker-Lewis Index (TLI)         0.872
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)           -2773.332
## Loglikelihood unrestricted model (H1)   -2684.179
##
## Akaike (AIC)                            5610.665
## Bayesian (BIC)                          5730.235
## Sample-size adjusted Bayesian (SABIC)    5628.743
##
## Root Mean Square Error of Approximation:
##
## RMSEA                                  0.119
## 90 Percent confidence interval - lower  0.102
## 90 Percent confidence interval - upper  0.137
## P-value H_0: RMSEA <= 0.050            0.000
## P-value H_0: RMSEA >= 0.080            1.000
##
## Robust RMSEA                            0.125
## 90 Percent confidence interval - lower  0.108
## 90 Percent confidence interval - upper  0.143
## P-value H_0: Robust RMSEA <= 0.050    0.000

```

```

## P-value H_0: Robust RMSEA >= 0.080          1.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                               0.060
##
## Parameter Estimates:
##
## Standard errors                      Standard
## Information                          Observed
## Observed information based on        Hessian
##
## Latent Variables:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR =~
##   CENTERING           1.000
##   TradPed =~
##     ClearRspnsblts  1.000
##     EffectivAnswnrs 0.967  0.056 17.123  0.000  0.694  0.845
##     Feedback          0.915  0.063 14.455  0.000  0.635  0.725
##     ClearOrganiztn   1.193  0.075 15.871  0.000  0.828  0.771
##     ClearPresenttn   1.111  0.063 17.767  0.000  0.771  0.841
##   SRPed =~
##     InclusvClassrm  1.000
##     EquitableEval   0.953  0.087 10.948  0.000  0.455  0.717
##     MultPerspectvs  1.466  0.116 12.623  0.000  0.699  0.839
##     DEIintegration  0.901  0.099  9.106  0.000  0.430  0.582
##
## Covariances:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR ~~
##   TradPed      -0.025  0.020 -1.235  0.217 -0.074 -0.074
##   SRPed        0.008  0.015  0.531  0.595  0.034  0.034
##   TradPed ~~
##   SRPed        0.284  0.033  8.598  0.000  0.858  0.858
##
## Intercepts:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING       0.406  0.028 14.570  0.000  0.406  0.828
## .ClearRspnsblts 4.439  0.047 94.893  0.000  4.439  5.405
## .EffectivAnswnrs 4.365  0.047 93.060  0.000  4.365  5.297
## .Feedback         4.236  0.050 84.626  0.000  4.236  4.835
## .ClearOrganiztn  4.006  0.061 65.611  0.000  4.006  3.732
## .ClearPresenttn  4.236  0.052 81.215  0.000  4.236  4.620
## .InclusvClassrm 4.613  0.039 118.389 0.000  4.613  6.788
## .EquitableEval   4.572  0.036 126.725 0.000  4.572  7.216
## .MultPerspectvs  4.391  0.048 92.287  0.000  4.391  5.265
## .DEIintegration  4.512  0.044 102.435 0.000  4.512  6.105

```

```

##      CTR          0.000          0.000          0.000
##    TradPed        0.000          0.000          0.000
##    SRPed         0.000          0.000          0.000
##
## Variances:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING     0.000          0.000          0.000          0.000
## .ClearRspnsblts 0.193       0.020       9.430       0.000       0.193       0.286
## .EffectivAnswnrs 0.228       0.023      10.118       0.000       0.228       0.336
## .Feedback       0.364       0.033      10.987       0.000       0.364       0.475
## .ClearOrganiztn 0.466       0.044      10.692       0.000       0.466       0.405
## .ClearPresenttn 0.246       0.026      9.479       0.000       0.246       0.293
## .InclusvClassrm 0.234       0.023      10.216       0.000       0.234       0.507
## .EquitableEval   0.195       0.019      10.227       0.000       0.195       0.485
## .MultPerspectvs  0.206       0.026      7.877       0.000       0.206       0.297
## .DEIIntegration  0.362       0.034      10.528       0.000       0.362       0.662
##    CTR          0.241       0.019     12.450       0.000       1.000       1.000
##    TradPed       0.482       0.054      8.968       0.000       1.000       1.000
##    SRPed         0.228       0.035      6.557       0.000       1.000       1.000

msmt_fit_pEsts <- lavaan:::parameterEstimates(msmt_fit, boot.ci.type = "bca.simple",
                                              standardized = TRUE)
# msmt_fit_pEsts #To reduce redundancy in the book, I did not print
# the parameter estimates. Their object is used in exporting a .csv
# file.

```

Below is script that will export the global fit indices (via *tidySEM::table_fit*) and the parameter estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```

# global fit indices
msmt_globalfit <- tidySEM::table_fit(msmt_fit)
write.csv(msmt_globalfit, file = "msmt_globalfit.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(msmt_fit_pEsts, file = "msmt_fit_pEsts.csv")

```

Here is how I wrote up the results:

Analyzing our proposed multiple mediator model followed the two-step procedure of first evaluating a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index (CFI) ≥ 0.95 , root-mean-square error of approximation (RMSEA) ≤ 0.06 , and the standard root-mean-square residual (SRMR) ≤ 0.08 . To establish acceptable fit, we used CFI ≥ 0.90 , RMSEA ≤ 0.10 , and SRMR ≤ 0.10 [Weston and Gore, 2006].

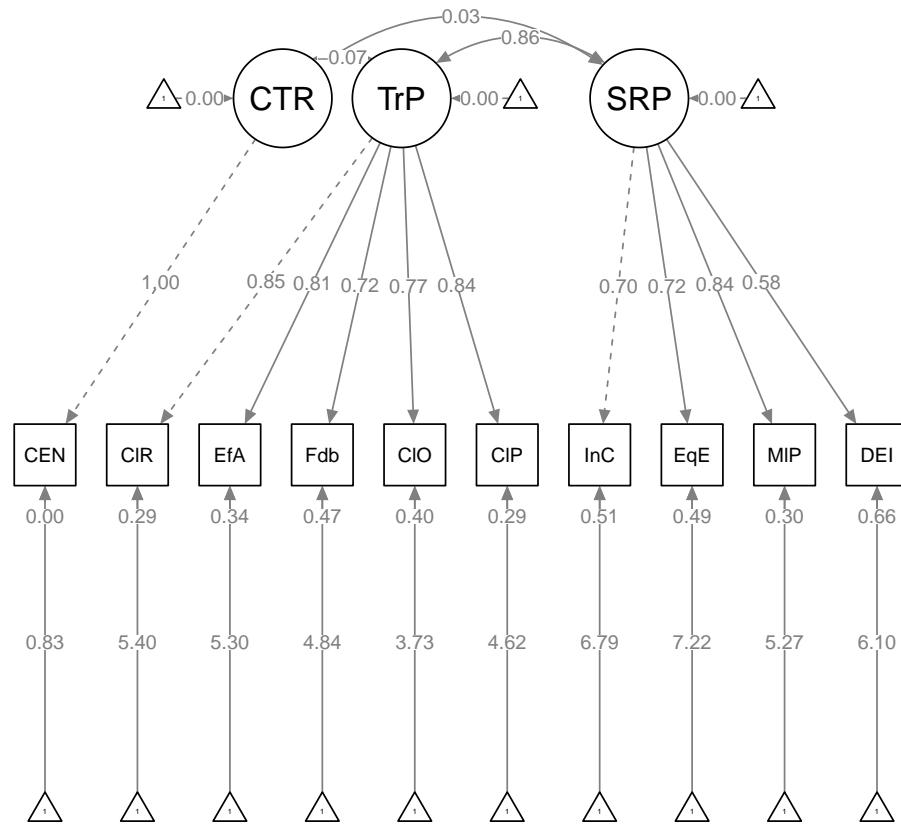
We evaluated the measurement model by following recommendations by Little et al. [Little et al., 2002, 2013]. Specifically, each latent variable was represented by each of the items on its subscale. Given that TradPed and SRPed had 5 and 4 items, respectively, we did not parcel items. The Centering variable with two levels (pre-centered, re-centered) was recoded as a dummy variable with 0, 1 coding. In the specification, its measurement error was fixed at zero. While all factor loadings were strong, statistically significant, and properly balanced, global fit statistics were mixed: $\chi^2(33) = 178.307, p < 0.001, CFI = 0.911, RMSEA = 0.119, 90CI[0.102, 0.137], SRMR = 0.060$. Factor loadings of each of the parcels are presented in Table 1. We proceeded to testing the structural model with caution.

Table 1

Factor Loadings for the Measurement Model					
Latent variable and indicator	est	SE	p	est_std	
Traditional Pedagogy					
ClearResponsibilities	1.000	0.000		0.845	
EffectiveAnswers	0.967	0.056	<0.001	0.815	
Feedback	0.915	0.063	<0.001	0.725	
ClearOrganization	1.193	0.075	<0.001	0.771	
ClearPresentation	1.111	0.063	<0.001	0.841	
Socially Responsive Pedagogy					
InclusvClassrm	1.000	0.000		0.702	
EquitableEval	0.953	0.087	<0.001	0.717	
MultPerspectives	1.466	0.116	<0.001	0.839	
DEIntegration	0.901	0.099	<0.001	0.582	
CENTERING	0.000	0.000		0.000	

Here is a figure of my measurement model:

```
semPlot::semPaths(msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(5,
5, 5, 5))
```



Specify and evaluate a *structural* model

As a reminder, I hypothesize that the evaluation of socially responsive pedagogy will be predicted by intentional recentering through traditional pedagogy.

X = Centering: explicit recentering (0 = precentered; 1 = recentered) M = TradPed: traditional pedagogy (continuously scaled with higher scores being more favorable) Y = SRPed: socially responsive pedagogy (continuously scaled with higher scores being more favorable)

```
ReC_struct_mod <- "
  #measurement model
  CTR =~ CENTERING #this is a single item indicator, I had to add code below to set the variance to zero
  TradPed =~ ClearResponsibilities + EffectiveAnswers + Feedback + ClearOrganization + C
  SRPed =~ InclusvClassrm + EquitableEval + MultPerspectives + DEIintegration

  # Variance of the single item indicator
  CENTERING ~~ 0*CENTERING"
```

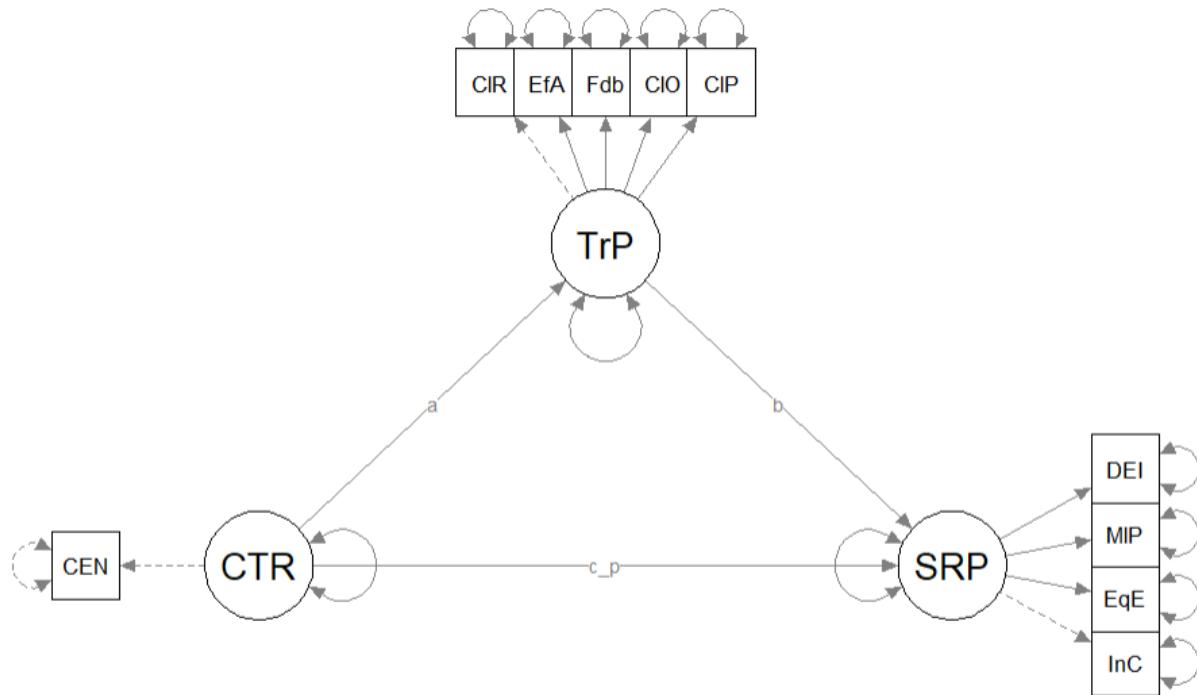


Figure 10.9: An image of the our hypothesized structural model

```

#structural model with labels for calculation of the indirect effect
SRPed ~ b*TradPed + c_p*CTR
TradPed ~ a*CTR

#calculations
indirect := a*b
direct := c_p
total_c := c_p + (a*b)
""

set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
ReC_struct_fit <- lavaan::sem(ReC_struct_mod, data = babydf, missing = "fiml")
ReC_struct_summary <- lavaan::summary(ReC_struct_fit, fit.measures = TRUE,
  standardized = TRUE, rsq = TRUE)
ReC_struct_pEsts <- lavaan::parameterEstimates(ReC_struct_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
ReC_struct_summary

## lavaan 0.6.16 ended normally after 61 iterations
## 
##   Estimator                               ML
##   Optimization method                     NLMINB
##   Number of model parameters               32

```

```

## 
##   Number of observations           310
##   Number of missing patterns       13
##
## Model Test User Model:
##
##   Test statistic                  178.307
##   Degrees of freedom              33
##   P-value (Chi-square)            0.000
##
## Model Test Baseline Model:
##
##   Test statistic                 1680.505
##   Degrees of freedom              45
##   P-value                         0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)      0.911
##   Tucker-Lewis Index (TLI)          0.879
##
##   Robust Comparative Fit Index (CFI) 0.906
##   Robust Tucker-Lewis Index (TLI)    0.872
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)     -2773.332
##   Loglikelihood unrestricted model (H1) -2684.179
##
##   Akaike (AIC)                     5610.665
##   Bayesian (BIC)                   5730.235
##   Sample-size adjusted Bayesian (SABIC) 5628.743
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                           0.119
##   90 Percent confidence interval - lower 0.102
##   90 Percent confidence interval - upper 0.137
##   P-value H_0: RMSEA <= 0.050      0.000
##   P-value H_0: RMSEA >= 0.080      1.000
##
##   Robust RMSEA                     0.125
##   90 Percent confidence interval - lower 0.108
##   90 Percent confidence interval - upper 0.143
##   P-value H_0: Robust RMSEA <= 0.050 0.000
##   P-value H_0: Robust RMSEA >= 0.080 1.000
##
## Standardized Root Mean Square Residual:

```

```

##  

##      SRMR          0.060  

##  

## Parameter Estimates:  

##  

##      Standard errors           Standard  

##      Information             Observed  

##      Observed information based on   Hessian  

##  

## Latent Variables:  

##  

##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all  

##  

##  CTR =~  

##    CENTERING     1.000  

##  

##  TradPed =~  

##    ClearRspnsblts 1.000  

##    EffectivAnswnrs 0.967 0.056 17.123 0.000 0.694 0.845  

##    Feedback       0.915 0.063 14.455 0.000 0.635 0.725  

##    ClearOrganiztn 1.193 0.075 15.871 0.000 0.828 0.771  

##    ClearPresenttn 1.111 0.063 17.767 0.000 0.771 0.841  

##  

##  SRPed =~  

##    InclusvClassrm 1.000  

##    EquitableEval  0.953 0.087 10.948 0.000 0.455 0.717  

##    MultPerspectvs 1.466 0.116 12.623 0.000 0.699 0.839  

##    DEIintegration 0.901 0.099 9.106 0.000 0.430 0.582  

##  

## Regressions:  

##  

##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all  

##  

##  SRPed ~  

##    TradPed     (b) 0.595 0.050 11.870 0.000 0.865 0.865  

##    CTR        (c_p) 0.095 0.042 2.247 0.025 0.098 0.098  

##  TradPed ~  

##    CTR        (a) -0.105 0.084 -1.241 0.215 -0.074 -0.074  

##  

## Intercepts:  

##  

##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all  

##  

##  .CENTERING      0.406 0.028 14.570 0.000 0.406 0.828  

##  .ClearRspnsblts 4.439 0.047 94.893 0.000 4.439 5.405  

##  .EffectivAnswnrs 4.365 0.047 93.060 0.000 4.365 5.297  

##  .Feedback        4.236 0.050 84.626 0.000 4.236 4.835  

##  .ClearOrganiztn  4.006 0.061 65.611 0.000 4.006 3.732  

##  .ClearPresenttn  4.236 0.052 81.215 0.000 4.236 4.620  

##  .InclusvClassrm 4.613 0.039 118.389 0.000 4.613 6.788  

##  .EquitableEval   4.572 0.036 126.725 0.000 4.572 7.216  

##  .MultPerspectvs  4.391 0.048 92.287 0.000 4.391 5.265  

##  .DEIintegration  4.512 0.044 102.435 0.000 4.512 6.105  

##  CTR            0.000  

##  .TradPed        0.000  

##  .SRPed          0.000

```

```

## 
## Variances:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING    0.000      0.000 0.000 0.000 0.000 0.000
## .ClearRspnsblts 0.193 0.020 9.430 0.000 0.193 0.286
## .EffectivAnswrs 0.228 0.023 10.118 0.000 0.228 0.336
## .Feedback     0.364 0.033 10.987 0.000 0.364 0.475
## .ClearOrganiztn 0.466 0.044 10.692 0.000 0.466 0.405
## .ClearPresenttn 0.246 0.026 9.479 0.000 0.246 0.293
## .InclusvClassrm 0.234 0.023 10.216 0.000 0.234 0.507
## .EquitableEval 0.195 0.019 10.227 0.000 0.195 0.485
## .MultPerspectvs 0.206 0.026 7.877 0.000 0.206 0.297
## .DEIintegration 0.362 0.034 10.528 0.000 0.362 0.662
## CTR          0.241 0.019 12.450 0.000 1.000 1.000
## .TradPed      0.479 0.053 8.963 0.000 0.994 0.994
## .SRPed        0.058 0.013 4.516 0.000 0.254 0.254
## 
## R-Square:
##           Estimate
## CENTERING      1.000
## ClearRspnsblts 0.714
## EffectivAnswrs 0.664
## Feedback       0.525
## ClearOrganiztn 0.595
## ClearPresenttn 0.707
## InclusvClassrm 0.493
## EquitableEval   0.515
## MultPerspectvs 0.703
## DEIintegration 0.338
## TradPed        0.006
## SRPed          0.746
## 
## Defined Parameters:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## indirect      -0.062 0.050 -1.236 0.216 -0.064 -0.064
## direct        0.095 0.042 2.247 0.025 0.098 0.098
## total_c       0.033 0.061 0.531 0.595 0.034 0.034

# ReC_struct_pEsts #although creating the object is useful to export
# as a .csv I didn't ask it to print into the book

```

Below is script that will export the global fit indices (via *tidySEM::table_fit*) and the parameter estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```

# global fit indices
ReC_globalfit <- tidySEM::table_fit(ReC_struct_fit)

```

```
write.csv(ReC_globalfit, file = "ReC_globalfit.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(ReC_struct_pEsts, file = "ReC_struct_pEsts.csv")
```

I will compare my results to the global fit criteria/thresholds:

Criteria	Our Results	Criteria met?
Non-significant chi-square	$\chi^2(33) = 178.307, p < 0.001$	No
$CFI \geq .95$ (or at least .90)	$CFI = 0.911$	Just barely
$RMSEA \leq .05$ (or < .08, at least < .10, also 90CI)	$RMSEA = 0.119, 90CI[0.102, 0.137]$	No
$SRMR \leq .08$ (at least < .10)	$SRMR = 0.060$	Yes
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	$CFI = 0.911, SRMR = 0.060$	No

Tabling the factor loadings and parameter estimates provide an organized way to examine local fit.

Table 2

Model Coefficients Assessing the Effect of Recentering on Socially Responsive Pedagogy Through Traditional Pedagogy

Predictor	B	SE _B	p	β	R ²
Traditional Pedagogy (M)					.01
Centering (a)	-0.105	0.084	0.215	-0.074	
Socially Responsive Pedagogy (DV)					.75
Traditional Pedagogy (b)	0.595	0.050	0.000	0.865	
Centering (c')	0.095	0.042	0.025	0.098	

Effects	B	SE _B	p	95% CI	
Indirect(a * b)	-0.062	0.050	0.216	-0.064	-0.161, 0.037
Total effect	0.033	0.061	0.595	0.034	-0.088, 0.153

Note. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

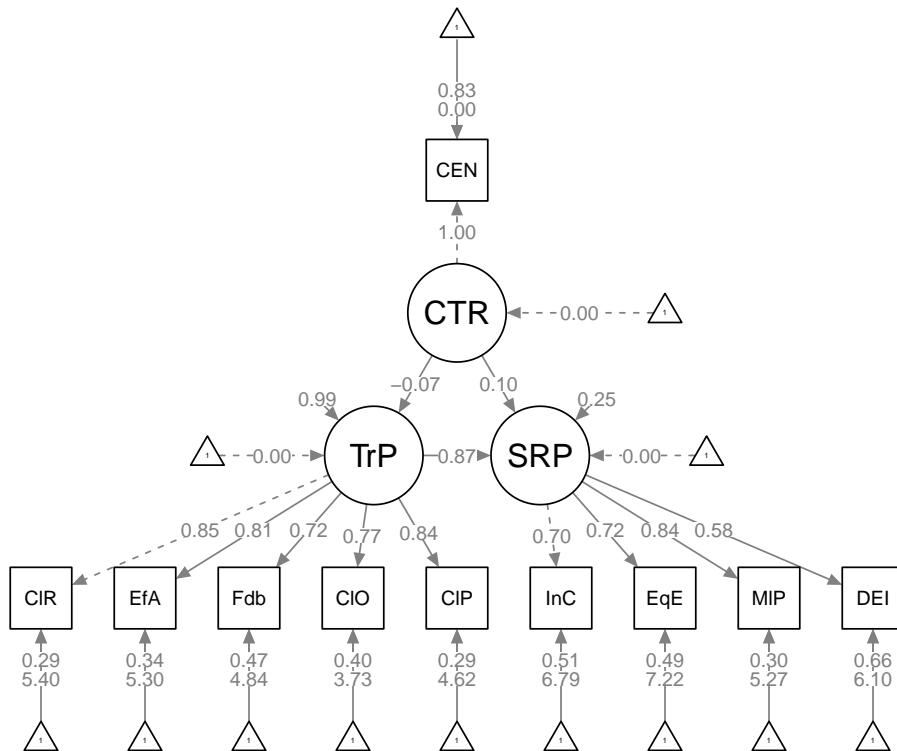
For a little extra learning, I compared these SEM results to the path analytic ones from the homeworked lesson on simple mediation. While the values are not identical, they are parallel. No indirect effect was found there, either.

Here's how I might write up this section of the results:

Like our measurement model, our structural model is just-identified with zero degrees of freedom. Consequently, the global fit indices are identical and provided an inconsistent evaluation of fit: $\chi^2(33) = 178.307, p < 0.001, CFI = 0.911, RMSEA = 0.119, 90CI[0.102, 0.137], SRMR = 0.060$. Indices of local fit (e.g., regression weights, parameter estimates) are presented in Table 2 and Figure 1 provides a graphical representation of our results. While results suggest that Centering ($B = 0.095, p = 0.025$) traditional pedagogy ($B = 0.595, p < 0.001$) had statistically significant effects on socially responsive pedagogy, there was no evidence of an indirect effect ($B = -0.062, SE = 0.050, p = 0.216$). The model accounted for only 1% of the variance in traditional pedagogy and 75% of the variance in socially responsive pedagogy.

Let's work up a figure

```
plot_ReC_struct <- semPlot::semPaths(ReC_struct_fit, what = "col", whatLabels = "stand",
  sizeMan = 5, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(5, 5, 5, 5))
```



Grid for Plotting <code>semplot::sempath</code>		
(1,1) empty	(1,2) TrP	(1,3) empty
(2,1) CTR	(2,2) empty	(2,3) SRP

We place these values along with the names of our latent variables in to the `sempath::semtools::layout_matrix` function.

Lots of things can go wrong in the code below. In preparing this example I lost time

- because I could not distinguish between the capital “I” (i) and the lowercase “l” (L) in the `SemPlot`, and
- because in the `m1_indicator_factor` code I did not list the latent variables enough time to match the number of indicators in the `m1_indicator_order`

After traversing several rabbit trails, taking a break, and returning, I could see my errors. That’s just R.

```
#IMPORTANT: Must use the node names (take directly from the SemPlot) assigned by SemPlot
#You can change them as the last thing
m1_msmt <- semtools::layout_matrix(CTR = c(2,1),
                                    TrP = c(1,2),
                                    SRP = c(2,3))

#tell where you want the indicators to face
m1_point_to <- semtools::layout_matrix(left = c(2,1),
                                         up = c(1,2),
                                         right = c(2,3))

#the next two codes -- indicator_order and indicator_factor are paired together, they specify
m1_indicator_order <- c("CLR", "EFA", "Fdb", "C10", "Clp",
                        "InC", "EqE", "M1P", "DEI",
                        "CEN")

m1_indicator_factor <- c("TrP", "TrP", "TrP", "TrP", "TrP",
                         "SRP", "SRP", "SRP", "SRP",
                         "CTR")

#next set of code pushes the indicator variables away from the factor
m1_indicator_push <- c(CTR = 1,
                      TrP = 2,
                      SRP = 5)
m1_indicator_spread <- c(CTR = 1, #spreading the boxes away from each other
                          TrP = 3,
                          SRP = 5)
```

Finally, we can feed all of the objects that whole these instructions into the `sempath::sem_set_layout` function. If desired, we can use the `sempath::change_node_label` function to rename the latent variables. Again, make sure to use the variable names that `semPlot::semPaths` has assigned.

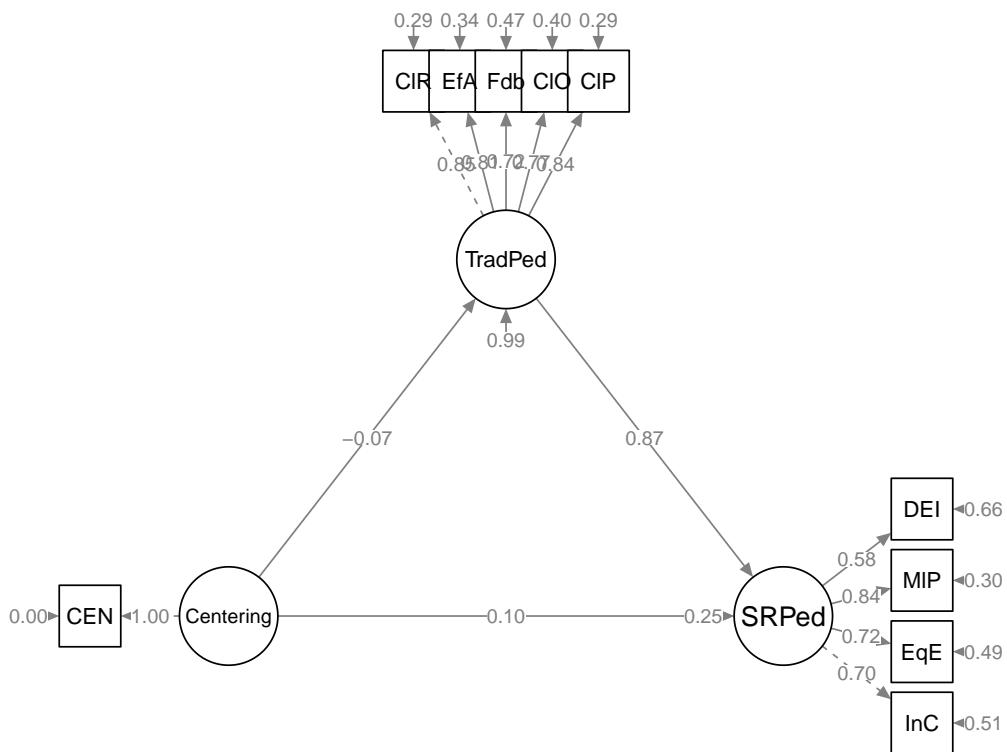
```

plot1 <- semtools::set_sem_layout(plot_ReC_struct, indicator_order = m1_indicator_order,
  indicator_factor = m1_indicator_factor, factor_layout = m1_msmt, factor_point_to = m1_point_to,
  indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)

# changing node labels
plot1 <- semtools::change_node_label(plot1, c(CTR = "Centering", TrP = "TradPed",
  SRP = "SRPed"), label.cex = 1.1)

plot(plot1)

```



Respecify and evaluate an *alternative* structural model

Having worked with this data quite a bit, traditional and socially responsive pedagogy are highly correlated. Thus, I think I would like to see how centering predicts both traditional pedagogy and socially responsive pedagogy. In this analysis, I want to fix the covariance between TradPed and SRPed to be 0.0. Why? I'm *just curious* to see what will happen.

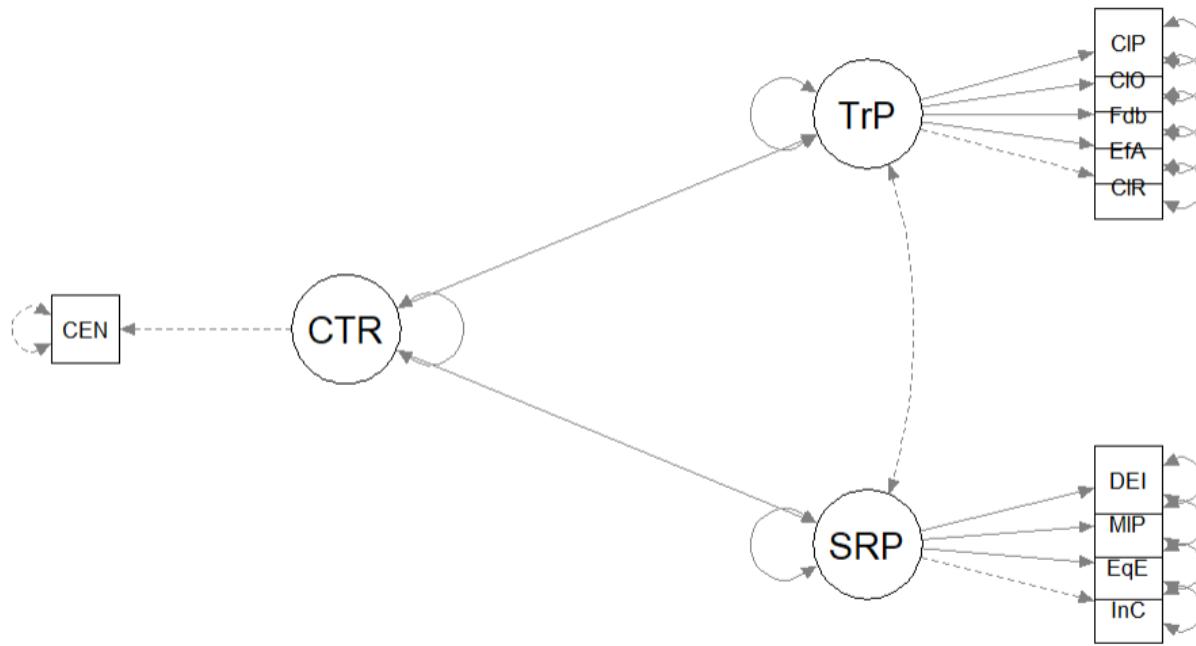


Figure 10.10: An image of the our alternative structural model

```

ReC_ALT_mod <- "
  #measurement model
  CTR =~ CENTERING #this is a single item indicator, I had to add code below to set the
  TradPed =~ ClearResponsibilities + EffectiveAnswers + Feedback + ClearOrganization + C
  SRPed =~ InclusvClassrm + EquitableEval + MultPerspectives + DEIintegration

  # Variance of the single item indicator
  CENTERING ~~ 0*CENTERING
  TradPed~~0*SRPed

  #structural model with labels for calculation of the indirect effect
  SRPed ~ CTR
  TradPed ~ CTR

  "
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence
ReC_ALT_fit <- lavaan::sem(ReC_ALT_mod, data = babydf, missing = "fiml",
  fixed.x = FALSE)
ReC_ALT_summary <- lavaan::summary(ReC_ALT_fit, fit.measures = TRUE, standardized = TRUE,
  rsq = TRUE)
ReC_ALT_pEsts <- lavaan::parameterEstimates(ReC_ALT_fit, boot.ci.type = "bca.simple",
  standardized = TRUE)
ReC_ALT_summary
  
```

```

## lavaan 0.6.16 ended normally after 61 iterations
##
##   Estimator                      ML
## Optimization method            NLMINB
## Number of model parameters    31
##
##   Number of observations        310
##   Number of missing patterns   13
##
## Model Test User Model:
##
##   Test statistic                414.407
##   Degrees of freedom             34
##   P-value (Chi-square)          0.000
##
## Model Test Baseline Model:
##
##   Test statistic                1680.505
##   Degrees of freedom              45
##   P-value                         0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)    0.767
##   Tucker-Lewis Index (TLI)        0.692
##
##   Robust Comparative Fit Index (CFI) 0.767
##   Robust Tucker-Lewis Index (TLI)  0.692
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0) -2891.382
##   Loglikelihood unrestricted model (H1) -2684.179
##
##   Akaike (AIC)                  5844.765
##   Bayesian (BIC)                 5960.598
##   Sample-size adjusted Bayesian (SABIC) 5862.278
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                        0.190
##   90 Percent confidence interval - lower 0.174
##   90 Percent confidence interval - upper 0.207
##   P-value H_0: RMSEA <= 0.050      0.000
##   P-value H_0: RMSEA >= 0.080      1.000
##
##   Robust RMSEA                   0.193
##   90 Percent confidence interval - lower 0.177

```

```

## 90 Percent confidence interval - upper          0.210
## P-value H_0: Robust RMSEA <= 0.050          0.000
## P-value H_0: Robust RMSEA >= 0.080          1.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                               0.278
##
## Parameter Estimates:
##
## Standard errors                      Standard
## Information                           Observed
## Observed information based on        Hessian
##
## Latent Variables:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR =~
##   CENTERING           1.000
## TradPed =~
##   ClearRspnsblts    1.000
##   EffectivAnswnrs   0.972  0.058 16.775  0.000  0.693  0.843
##   Feedback            0.886  0.065 13.687  0.000  0.614  0.700
##   ClearOrganiztn     1.220  0.076 15.960  0.000  0.845  0.787
##   ClearPresenttn    1.118  0.065 17.254  0.000  0.774  0.844
## SRPed =~
##   InclusvClassrm    1.000
##   EquitableEval      0.765  0.085  9.057  0.000  0.406  0.642
##   MultPerspectvs     1.202  0.117 10.282  0.000  0.638  0.765
##   DEIIntegration     0.995  0.087 11.443  0.000  0.528  0.712
##
## Regressions:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## SRPed ~
##   CTR                 0.077  0.070  1.100  0.271  0.071  0.071
## TradPed ~
##   CTR                -0.106  0.084 -1.254  0.210 -0.075 -0.075
##
## Covariances:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .TradPed ~~
##   .SRPed              0.000
##
## Intercepts:
##                         Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING           0.406  0.028 14.570  0.000  0.406  0.828
## .ClearRspnsblts     4.440  0.047 94.867  0.000  4.440  5.404
## .EffectivAnswnrs    4.365  0.047 93.061  0.000  4.365  5.298
## .Feedback             4.237  0.050 84.611  0.000  4.237  4.836

```

```

## .ClearOrganiztn    4.006   0.061   65.611   0.000   4.006   3.732
## .ClearPresenttn   4.236   0.052   81.215   0.000   4.236   4.620
## .InclusvClassrm   4.615   0.039  118.499   0.000   4.615   6.795
## .EquitableEval    4.573   0.036  126.851   0.000   4.573   7.224
## .MultPerspectvs   4.393   0.048   92.171   0.000   4.393   5.265
## .DEIintegration   4.508   0.044  102.473   0.000   4.508   6.075
## CTR                0.000
## .TradPed            0.000
## .SRPed              0.000
##
## Variances:
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CENTERING      0.000
## .ClearRspnsblts 0.195   0.022   8.903   0.000   0.195   0.289
## .EffectivAnswrs 0.226   0.023   9.663   0.000   0.226   0.332
## .Feedback        0.391   0.036  11.006   0.000   0.391   0.509
## .ClearOrganiztn  0.438   0.043  10.185   0.000   0.438   0.380
## .ClearPresenttn  0.241   0.027   8.888   0.000   0.241   0.287
## .InclusvClassrm 0.179   0.026   6.935   0.000   0.179   0.389
## .EquitableEval   0.236   0.024   9.772   0.000   0.236   0.588
## .MultPerspectvs  0.289   0.040   7.252   0.000   0.289   0.415
## .DEIintegration  0.272   0.032   8.387   0.000   0.272   0.493
## CTR                0.241   0.019  12.450   0.000   1.000   1.000
## .TradPed            0.477   0.054   8.838   0.000   0.994   0.994
## .SRPed              0.280   0.040   6.977   0.000   0.995   0.995
##
## R-Square:
##             Estimate
## CENTERING      1.000
## ClearRspnsblts 0.711
## EffectivAnswrs 0.668
## Feedback        0.491
## ClearOrganiztn  0.620
## ClearPresenttn  0.713
## InclusvClassrm 0.611
## EquitableEval   0.412
## MultPerspectvs  0.585
## DEIintegration  0.507
## TradPed           0.006
## SRPed              0.005

```

```

# ReC_ALT_pEsts #although creating the object is useful to export as
# a .csv I didn't ask it to print into the book

```

Below is script that will export the global fit indices (via `tidySEM::table_fit`) and the parameter estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```
# global fit indices
ReC_ALT_globalfit <- tidySEM::table_fit(ReC_ALT_fit)
write.csv(ReC_ALT_globalfit, file = "ReC_ALT_globalfit.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(ReC_ALT_pEsts, file = "ReC_ALT_pEsts.csv")
```

I will compare my results to the global fit criteria/thresholds:

Criteria	Our Results	Criteria met?
Non-significant chi-square	$\chi^2(34) = 414.408, p = < 0.001$	No
$CFI \geq .95$ (or at least .90)	$CFI = 0.767$	No
$RMSEA \leq .05$ (or $< .08$, at least $< .10$, also 90CI)	$RMSEA = 0.190, 90CI[0.174, 0.207]$	No
$SRMR \leq .08$ (at least $< .10$)	$SRMR = 0.278$	No
Combination rule: $CFI \geq .95$ & $SRMR \leq .08$	$CFI = 0.767, SRMR = 0.278$	No

Tabling the factor loadings and parameter estimates provide an organized way to examine local fit.

Table 2

Model Coefficients Assessing the Effect of Recentering on Socially Responsive Pedagogy Through Traditional Pedagogy

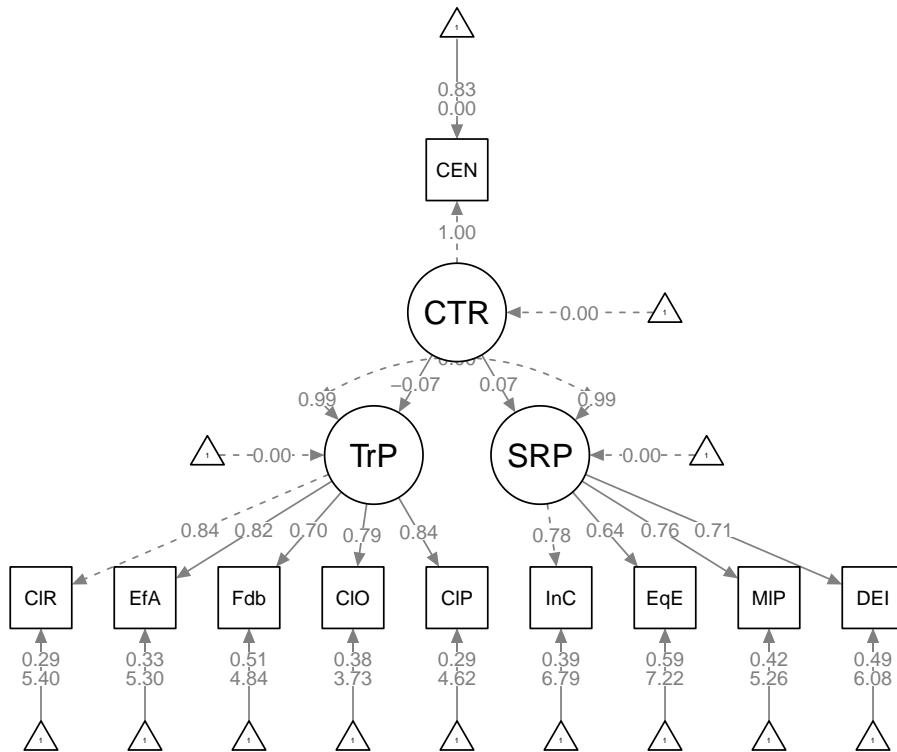
Predictor	B	SE _B	p	β	R ²
Traditional Pedagogy (M)					.01
Centering	-0.106	0.084	0.210	-0.075	
Socially Responsive Pedagogy (DV)					.01
Centering	0.077	0.070	0.271	0.071	

Here's how I might write up this section of the results:

In structural models where there is a cross-sectional analysis, the flow of causation is ambiguous. Therefore, we tested an alternative model, predicting both traditional and socially responsive pedagogy from centering status. The global fit indices of this alternative model suggested that fit was poor, $\chi^2(34) = 414.408, p = < 0.00, CFI = 0.767, RMSEA = 0.190, 90CI[0.174, 0.207], SRMR = 0.278$. Indices of local fit (e.g., regression weights, parameter estimates) are presented in Table 3 and Figure 2 provides a graphical representation of our results. Centering had non-significant effects on traditional pedagogy ($B = -0.106, p = 0.210$) and socially responsive pedagogy ($B = 0.077, p < 0.271$). Further, the model accounted for only 1% of the variance each, in traditional and socially responsive pedagogy.

Let's work up a figure

```
plot_ReC_ALT <- semPlot::semPaths(ReC_ALT_fit, what = "col", whatLabels = "stand",
  sizeMan = 5, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(5, 5, 5, 5))
```



Grid for Plotting `semplot::semPath`

(1,1) empty	(1,2) TrP
(2,1) CTR	(2,2) empty
(3,1) empty	(3,2) SRP

We place these values along with the names of our latent variables in to the `semptools::layout_matrix` function.

Lots of things can go wrong in the code below. In preparing this example I lost time

- because I could not distinguish between the capital “I” (i) and the lowercase “l” (L) in the SemPlot, and
 - because in the *m1_indicator_factor* code I did not list the latent variables enough time to match the number of indicators in the *m1_indicator_order*

After traversing several rabbit trails, taking a break, and returning, I could see my errors. That's just R.

```

#IMPORTANT: Must use the node names (take directly from the SemPlot) assigned by SemPlot
#You can change them as the last thing

m2_msmt <- semptools::layout_matrix(CTR = c(2,1),
                                      TrP = c(1,2),
                                      SRP = c(3,2))

#tell where you want the indicators to face
m2_point_to <- semptools::layout_matrix (left = c(2,1),
                                           right = c(1,2),
                                           right = c(3,2))

#the next two codes -- indicator_order and indicator_factor are paired together, they specify
m2_indicator_order <- c("ClR", "EfA", "Fdb", "ClO", "ClP",
                        "InC", "EqE", "MlP", "DEI",
                        "CEN")

m2_indicator_factor <- c("TrP", "TrP", "TrP", "TrP", "TrP",
                         "SRP", "SRP", "SRP", "SRP",
                         "CTR")

#next set of code pushes the indicator variables away from the factor
m2_indicator_push <- c(CTR = 1,
                       TrP = 2,
                       SRP = 5)

m2_indicator_spread <- c(CTR = 1, #spreading the boxes away from each other
                         TrP = 3,
                         SRP = 5)

```

Finally, we can feed all of the objects that whole these instructions into the `semtools::sem_set_layout` function. If desired, we can use the `semtools::change_node_label` function to rename the latent variables. Again, make sure to use the variable names that `semPlot::semPaths` has assigned.

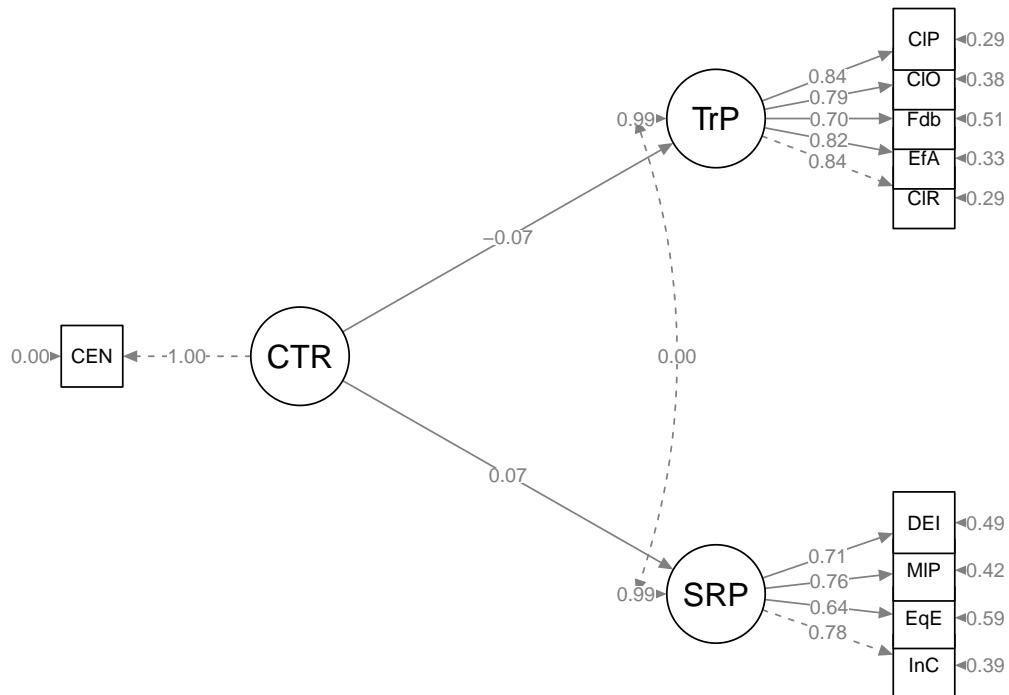
```

plot2 <- semptools::set_sem_layout(plot_ReC_ALT, indicator_order = m2_indicator_order,
  indicator_factor = m2_indicator_factor, factor_layout = m2_msmt, factor_point_to = m2_point_to,
  indicator_push = m2_indicator_push, indicator_spread = m2_indicator_spread)

# changing node labels
plot21 <- semptools::change_node_label(plot2, c(CTR = "Centering", TrP = "TradPed",
  SRP = "SRPed"), label.cex = 1.1)

```

```
plot(plot2)
```



10.12.1 Conduct a formal comparison of *global* fit.

We can use the *lavaan::lavTestLRT* to compare the global fit of the models.

```
lavaan::lavTestLRT(msmt_fit, ReC_struct_fit, ReC_ALT_fit)
```

```
## Warning in lavaan::lavTestLRT(msmt_fit, ReC_struct_fit, ReC_ALT_fit): lavaan
## WARNING: some models have the same degrees of freedom
```

```
##
## Chi-Squared Difference Test
##
##          Df      AIC      BIC  Chisq Chisq diff   RMSEA Df diff
##
```

```

## msmt_fit      33 5610.7 5730.2 178.31
## ReC_struct_fit 33 5610.7 5730.2 178.31          0.0 0.00000          0
## ReC_ALT_fit    34 5844.8 5960.6 414.41        236.1 0.87085          1
##                                         Pr(>Chisq)
## msmt_fit
## ReC_struct_fit
## ReC_ALT_fit    < 0.0000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

As predicted, the fit of the measurement and first structural models are identical. This is because they were both just-identified with zero degrees of freedom. The fit of the alternative model is statistically significantly different from the structural model. Results of the $\Delta\chi^2$ test were statistically significant: $\Delta\chi^2(1) = 236.1, p < 0.001$. We know from evaluating the global fit indices that the initial structural model (i.e., the simple mediation) is preferred.

APA style results with table and figure

Method/Aalytic Strategy

We specified a structural equation model predicting socially responsive pedagogy (SRPed) from centering status (Centering), mediated by traditional pedagogy (TradPed). The primary analysis occurred in two stages. First, we specified and evaluated a measurement model Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16).

Results

Preliminary Analyses

- Missing data analyses and managing missing data>
- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

Primary Analyses Analyzing our proposed multiple mediator model followed the two-step procedure of first evaluating a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index (CFI) ≥ 0.95 , root-mean-square error of approximation (RMSEA) ≤ 0.06 , and the standard root-mean-square residual (SRMR) ≤ 0.08 . To establish acceptable fit, we used CFI ≥ 0.90 , RMSEA ≤ 0.10 , and SRMR ≤ 0.10 [Weston and Gore, 2006].

We evaluated the measurement model by following recommendations by Little et al. [Little et al., 2002, 2013]. Specifically, each latent variable was represented by each of the items on its subscale. Given that TradPed and SRPed had 5 and 4 items, respectively, we did not parcel items. The Centering variable with two levels (pre-centered, re-centered) was recoded as a dummy variable with 0, 1 coding. In the specification, its measurement error was fixed at zero. While all factor loadings were strong, statistically significant, and properly valanced, global fit statistics were mixed: $\chi^2(33) = 178.307, p < 0.001, CFI = 0.911, RMSEA = 0.119, 90CI[0.102, 0.137], SRMR = 0.060$. Factor loadings of each of the parcels are presented in Table 1. We proceeded to testing the strutural model with caution.

Like our measurement model, our structural model is just-identified with zero degrees of freedom. Consequently, the global fit indices are identical and provided an inconsistent evaluation of fit: $\chi^2(33) = 178.307, p < 0.001, CFI = 0.911, RMSEA = 0.119, 90CI[0.102, 0.137], SRMR = 0.060$. Indices of local fit (e.g., regression weights, parameter estimates) are presented in Table 2 and Figure 1 provides a graphical representation of our results. While results suggest that Centering ($B = 0.095, p = 0.025$) traditional pedagogy ($B = 0.595, p < 0.001$) had statistically significant effects on socially responsive pedagogy, there was no evidence of an indirect effect ($B = -0.062, SE = 0.050, p = 0.216$). The model accounted for only 1% of the variance in traditional pedagogy and 75% of the variance in socially responsive pedagogy.

In structural models where there is a cross-sectional analysis, the flow of causation is ambiguous. Therefore, we tested an alternative model, predicting both traditional and socially responsive pedagogy from centering status. The global fit indices of this alternative model suggested that fit was poor, $\chi^2(34) = 414.408, p = < 0.00, CFI = 0.767, RMSEA = 0.190, 90CI[0.174, 0.207], SRMR = 0.278$. Indices of local fit (e.g., regression weights, parameter estimates) are presented in Table 3 and Figure 2 provides a graphical representation of our results. Centering had non-significant effects on traditional pedagogy ($B = -0.106, p = 0.210$) and socially responsive pedagogy ($B = 0.077, p < 0.271$). Further, the model accounted for only 1% of the variance each, in traditional and socially responsive pedagogy.

To formally compare the fit of the initial structural and alternative models we conducted a $\Delta\chi^2$ difference test. Resultstest were statistically significant: $\Delta\chi^2(1) = 236.1, p < 0.001$. We know from evaluating the global fit indices that the initial structural model (i.e., the simple mediation) is preferred.

Explanation to grader

A homework suggestion

There is one thing you could do to further improve this model's fit. Perhaps it could be one of your homework options!

Chapter 11

SEM: Model Respecifications

[Screencasted Lecture Link](#)

In the prior two lessons we engaged in the first of two steps structural equation modeling. We first established the *measurement model*. Next we specified and evaluated a *structural model*, which included testing alternative models. In this lesson we consider how to use model statistics to consider respecifying the model through *building* and *trimming*. Further, we learn how to compare these models to determine if the fit is improved (i.e., a goal when we free parameters to be in relation with each other), stayed the same (i.e., a goal when we trim paths, thereby fixing their relation to 0.00), or deteriorated.

11.1 Navigating this Lesson

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

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

Learning objectives from this lecture include the following:

- Explain the terms associated with model building (e.g. releasing/freeing constraints, forward search, adding paths) and model trimming (e.g., imposing constraints, deleting free parameters, backward search, deleting paths).
- Define and interpret *modification indices*.
- Use modification indices to determine whether or not to “add a path.”
- Describe how to use regression weights and significance in the determination of trimming a path.

- Compare the fit to determine if there are statistically significant differences between the baseline and modified models.
- Memorize which model (nested or nesting) will have better fit (without looking at the results).

11.1.2 Planning for Practice

In this lesson we focus on model modifications. In-so-doing, we use statistical criteria to determine whether or not to build or trim the model, we compare the fit of the models statistically, and we interpret whether or not we prefer the baseline or modified model. The homework assignment will require you to begin with an established structural model (for which the measurement model is strong), use statistical criterion to choose to add or delete at least one path or covariance, compare the fit of the two models, and determine which to retain. As always, the suggestions for practice are graded in complexity. You might:

- Rework the problem in the chapter by changing the random seed in the code that simulates the data. This should provide minor changes to the data, but the results will likely be very similar.
- Use the research data from the chapter, but evaluate a different set of variables.
- Use data from another lesson or data that is available to you.

11.1.3 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.

- Kline, R. (2016). *Principles and practice of structural equation modeling* (Fourth ed., Methodology in the social sciences). New York: The Guilford Press.
 - Chapter 10, Specification and Identification of Structural Regression Models
 - Chapter 14, Analysis of Structural Regression Models
- 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 6, Application 4: Testing the Factorial Validity of a Causal Structure
- Lewis, J. A., Williams, M. G., Peppers, E. J., & Gadson, C. A. (2017). Applying intersectionality to explore the relations between gendered racism and health among Black women. *Journal of Counseling Psychology*, 64(5), 475–486. <https://doi-org.ezproxy.spu.edu/10.1037/cou0000231>
 - This is the research vignette for this lesson.

11.1.4 Packages

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(tidyverse)){install.packages('tidyverse')}
# if(!require(psych)){install.packages('psych')}
# if(!require(jtools)){install.packages('jtools')}
```

11.2 Respecifying Structural Models

There is general consensus ([[Byrne, 2016c](#), [Chou and Bentler, 2002](#)]) that of the three types of approaches to model building (i.e., strictly confirmatory, alternative models, model generating [[Joreskog, 1993](#)]), that model generating is the most commonly used. This means that researchers will respecify the model by either adding paths or covariances, trimming paths or covariances, or both. At the risk of oversimplification, modelers generally take a model building or model trimming approaches. But of course – there is complexity and nuance that we should unravel.

11.2.1 Model Building

In model building, the researcher starts with a more parsimonious model and proceeds to a more general model. Chou and Bentler [[2002](#)] termed this *forward searching*. In general, this requires specifying a model that is *overidentified* (i.e., it has positive degrees of freedom). If the model does not fit the data well, we will request *modification indices*. These values are presented in the metric of the chi-square test and will tell us “by how much the chi-square value will decrease” if the two elements listed in each (of a long list) of the results is freed to relate (i.e., by a path or covariance). In a one-degree chi-square test, statistically significant change occurs when the modification index is greater than 3.841.

Freeing parameters is an activity that should be done with caution. Freeing a single parameter by adding a directional path or bidirectional covariance will reduce the degrees of freedom in the structural model by 1. Correspondingly, this respecified model will be the *nesting* model and will (in all likelihood) have better fit. The chi-square value will decrease by the amount indicated in the modification index and, presuming the parameter was freed because it would result in a statistically significant decrease in the unit of the chi-square, the respecified model with the additional path(s) may look as if it is a superior model. When fit is increasing incrementally, it can be tempting to keep freeing parameters until degrees of freedom are zero and the model is *just identified*. Doing so, risks *overparameterization*. This means that the model is *overfit*. Although it appears to fit the sample data well, it may fail to generalize to a different set of data that represents the population of interest. That is, the model is sample-specific which makes it less useful in making predictions or generalizing more broadly.

Thus, there are are cautions about model the forward searching approach:

- Any paths that are added (i.e., any parameters that are freed) should have strong theoretical justification.
- Any paths that are added should be statistically sensible. The procedure we use to produce modification indices, will produce them for every unsaturated paths. Sensible relations would generally include
 - directional paths between variables in the structural model, and
 - error covariances in the measurement model (for more on this see the lesson, [CFA: Hierarchical and Nested Models](#))
- Avoid adding paths that are not sensible. An example would be a path between an indicator in the measurement model and a structural variable.
 - but do think about why the modification index might be suggesting such; perhaps it is a clue to an error.

11.2.2 Model Trimming

In model trimming, the researcher starts with a more general model or saturated (i.e., zero degrees of freedom in the structural model) and, on the basis of statistical criteria, trims non-significant (i.e., and low regression weight) paths from the model. Chou and Bentler [2002] termed this *backward searching*. The general process is to identify and delete non-essential paths one at a time and evaluate the “hit to the fit.” Given that the more saturated model is the *nesting model*, one can expect that trimmed, *nested* model will have fit that is incrementally lower. The hope is that the difference does not bring the global fit indices into thresholds that indicate poor fit and that, when compared statistically, the nested model is not statistically significantly different than the nesting model.

Several prominent researchers appear to favor the model trimming/backward searching approach. Chou and Bentler [2002] acknowledged that while the forward searching approach is commonly utilized its ability to find the *true* model has been questioned. Their Monte Carlo based test study demonstrated that a backward search approach that imposed constraints with the z and Wald (W) tests identified the *true* model with greater than 60% accuracy. Kenny’s [Kenny, 2012] SEM website and workshops have also advocated for a model trimming approach that generally starts with a *just-identified* (i.e., saturated, $df = 0$) model and trims paths, one-at-a-time, to see if the adequately fitting result resembled the theoretical model they identified, apriorily.

11.2.3 Restating Approaches to Respecification

The terms, concepts, and logic related to nesting, saturation, and respecification can be confusing. Hopefully, this table can help clarify their meaning and role.

Approach to Respecification	Hypothesized (Original model) is	Comparison model(s) is	Typical Hope of $\Delta\chi^2$ test
Model building, forward search	overidentified, $df > 0$	more paths, fewer df	significant indicating additional path improved model fit

Approach to Respecification	Hypothesized (Original model) is	Comparison model(s) is	Typical Hope of $\Delta\chi^2$ test
Model trimming, backward search	just-identified, $df = 0$	fewer paths, more df	non-significant indicating deleted path did not lead to poorer model fit

In today's lesson, we will start with a rather complex, over-identified model. Following the example Byrne's [Byrne, 2016a] chapter, we will first use modification indices to see about freeing parameters (i.e., adding paths) and then inspect nonsignificant paths for possible constraint (e.g., deletion).

11.3 Workflow for Evaluating a Structural Model

Below is the overall workflow for evaluating a structural model. Today our focus is the specification and evaluation of the structural model.

Workflow: Evaluating a Structural Model	
A Priori Power Analysis	
<ul style="list-style-type: none"> Conduct an a priori power analysis to determine the appropriate sample size. Draw estimates of effect from pilot data and/or the literature. 	
Scrubbing & Scoring	
<ul style="list-style-type: none"> Import data and format (i.e., variable naming, reverse-scoring) item level variables. Analyze item-level missingness. If using scales, create the mean scores of the scales. Determine and execute approach for managing missingness. Popular choices are available item analysis (e.g., Parent, 2013) and multiple imputation. Analyze scale-level missingness. Create a df with <i>only</i> the items (scaled in the proper direction). 	
Data Diagnostics	
<ul style="list-style-type: none"> Evaluate univariate normality (i.e., one variable at a time) with Shapiro-Wilks tests; $p < .05$ indicates a violation of univariate normality. Evaluate multivariate normality (i.e., all continuously scaled variables simultaneously) with Mahalanobis test. Identify outliers (e.g., cases with Mahal values $> 3 SDs$ from the centroid). Consider deleting (or transforming if there is an extreme-ish "jump" in the sorted values). Evaluate internal consistency of the scaled scores with Cronbach's alpha or omega; the latter is increasingly preferred. 	
Specify and evaluate a measurement model	
<ul style="list-style-type: none"> In this just-identified (saturated) model, all latent variables are specified as covarying <ul style="list-style-type: none"> For LVs with 3 items or more, remember to set a marker/reference variable. For LVs with 2 items, constrain the loadings to be equal For single-item indicators fix the error variance to zero (or a non-zero estimate of unreliability) Evaluate results with <i>global</i> (e.g., χ^2, CFI, RMSEA, SRMR) and <i>local</i> (i.e., factor loadings and covariances) fit indices In the event of poor fit, respecify LVs with multiple indicators with parcels. Nested alternative measurement models can be compared with χ^2 difference, ΔCFI tests; non-nested models with AIC, and BIC tests 	
Specify and evaluate a structural model	
<ul style="list-style-type: none"> Replace the covariances with paths that represent the a priori hypotheses <ul style="list-style-type: none"> These models could take a variety of forms. It is possible to respecify models through trimming or building approaches. Evaluate results using: <ul style="list-style-type: none"> global fit indices (e.g., χ^2, CFI, RMSEA, SRMS), local fit indices (i.e., strength and significance of factor loadings, covariances, and additional model parameters [e.g., indirect effects]) Consider respecifying and evaluating one or more <i>alternative</i> models: <ul style="list-style-type: none"> <i>Forward searching</i> frees parameters (i.e., adds paths or covariances) and can use modification indices as a guide. <i>Backward searching</i> restrains parameters (i.e., deletes paths or covariances) and can use low and non-significant paths as a guide. Compare the fit of the alternate models <ul style="list-style-type: none"> Nested models can be compared with χ^2 difference, ΔCFI tests Non-nested models can use AIC, and BIC comparisons (lower values suggest better fit) 	
Quick Guide for Global and Comparative Fit Statistics	
<ul style="list-style-type: none"> $\chi^2, p > .05$; this test is sensitive to sample size and this value can be difficult to attain CFI $> .95$ (or at least $.90$) RMSEA (and associated 90%CI) are $< .05$ ($< .08$, or at least $< .10$) SRMR $< .08$ (or at least $< .10$) Combination rule: CFI $< .95$ and SRMR $< .08$ AIC and BIC are compared; the lowest values suggest better models $\chi^2\Delta$ is statistically significant; the model with the superior global fit is the better model ΔCFI is greater than 0.01; the model with CFI values closest to 1.0 has better fit 	

Evaluating a structural model involves the following steps:

- A Priori Power Analysis
 - Conduct an a priori power analysis to determine the appropriate sample size. __ Draw estimates of effect from pilot data and/or the literature.

- Scrubbing & Scoring
 - Import data and format (i.e., variable naming, reverse-scoring) item level variables.
 - Analyze item-level missingness.
 - If using scales, create the mean scores of the scales.
 - Determine and execute approach for managing missingness. Popular choices are available item analysis (e.g., Parent, 2013) and multiple imputation.
 - Analyze scale-level missingness.
 - Create a df with only the items (scaled in the proper direction).
- Data Diagnostics
 - Evaluate univariate normality (i.e., one variable at a time) with Shapiro-Wilks tests; $p < .05$ indicates a violation of univariate normality.
 - Evaluate multivariate normality (i.e., all continuously scaled variables simultaneously) with Mahalanobis test. Identify outliers (e.g., cases with Mahal values > 3 SDs from the centroid). Consider deleting (or transforming if there is an extreme-ish “jump” in the sorted values).
 - Evaluate internal consistency of the scaled scores with Cronbach’s alpha or omega; the latter is increasingly preferred. Specify and evaluate a measurement model
 - In this just-identified (saturated) model, all latent variables are specified as covarying.
 - * For LVs with 3 items or more, remember to set a marker/reference variable,
 - * For LVs with 2 items, constrain the loadings to be equal,
 - * For single-item indicators fix the error variance to zero (or a non-zero estimate of unreliability).
 - Evaluate results with global (e.g., χ^2 , CFI, RMSEA, SRMR) and local (i.e., factor loadings and covariances) fit indices.
 - In the event of poor fit, respecify LVs with multiple indicators with parcels.
 - Nested alternative measurement models can be compared with χ^2 difference, ΔCFI tests; non-nested models with AIC, and BIC tests .
- Specify and evaluate a structural model.
 - Replace the covariances with paths that represent the a priori hypotheses.
 - * These models could take a variety of forms.
 - * It is possible to respecify models through trimming or building approaches.
 - Evaluate results using
 - * *global* fit indices (e.g., χ^2 , CFI, RMSEA, SRMS),
 - * *local* fit indices (i.e., strength and significance of factor loadings, covariances, and additional model parameters [e.g., indirect effects]).
 - Consider respecifying and evaluating one or more *alternative* models.
 - * *Forward searching* involves freeing parameters (adding paths or covariances) and can use modification indices as a guide.
 - * *Backward searching* involves restraining parameters (deleting paths or covariances) and can use low and non-significant paths as a guide.
 - Compare the fit of the alternate models.
 - * Nested models can be compared with χ^2 difference and ΔCFI tests.
 - * Non-nested models can be compared with AIC and BIC (lower values suggest better fit).

- Quick Guide for Global and Comparative Fit Statistics.
 - χ^2 , $p < .05$; this test is sensitive to sample size and this value can be difficult to attain
 - $CFI > .95$ (or at least $.90$)
 - RMSEA (and associated 90%CI) are $< .05$ ($< .08$, or at least $< .10$)
 - SRMR $< .08$ (or at least $< .10$)
 - Combination rule: $CFI < .95$ and $SRMR < .08$
 - AIC and BIC are compared; the lowest values suggest better models
 - $\chi^2\Delta$ is statistically significant; the model with the superior fit is the better model
 - δCFI is greater than 0.01; the model with CFI values closest to 1.0 has better fit

11.4 Research Vignette

Once again the research vignette comes from the Lewis, Williams, Peppers, and Gadson's [2017] study titled, "Applying Intersectionality to Explore the Relations Between Gendered Racism and Health Among Black Women." The study was published in the Journal of Counseling Psychology. Participants were 231 Black women who completed an online survey.

Variables used in the study included:

- **GRMS:** Gendered Racial Microaggressions Scale [Lewis and Neville, 2015] is a 26-item scale that assesses the frequency of nonverbal, verbal, and behavioral negative racial and gender slights experienced by Black women. Scaling is along six points ranging from 0 (never) to 5 (once a week or more). Higher scores indicate a greater frequency of gendered racial microaggressions. An example item is, "Someone has made a sexually inappropriate comment about my butt, hips, or thighs."
- **MntlHlth** and **PhysHlth:** Short Form Health Survey - Version 2 [Ware et al., 1995] is a 12-item scale used to report self-reported mental (six items) and physical health (six items). Higher scores indicate higher mental health (e.g., little or no psychological distress) and physical health (e.g., little or no reported symptoms in physical functioning). An example of an item assessing mental health was, "How much of the time during the last 4 weeks have you felt calm and peaceful?"; an example of a physical health item was, "During the past 4 weeks, how much did pain interfere with your normal work?"
- **Sprtlty, SocSup, Engmgt**, and **DisEngmnt** are four subscales from the Brief Coping with Problems Experienced Inventory [Carver, 1997]. The 28 items on this scale are presented on a 4-point scale ranging from 1 (*I usually do not do this at all*) to 4(*I usually do this a lot*). Higher scores indicate a respondents' tendency to engage in a particular strategy. Instructions were modified to ask how the female participants responded to recent experiences of racism and sexism as Black women. The four subscales included spirituality (religion, acceptance, planning), interconnectedness/social support (vent emotions, emotional support,instrumental social support), problem-oriented/engagement coping (active coping, humor, positive reinterpretation/positive reframing), and disengagement coping (behavioral disengagement, substance abuse, denial, self-blame, self-distraction).
- **GRIcntlty:** The Multidimensional Inventory of Black Identity Centrality subscale [Sellers et al.] was modified to measure the intersection of racial and gender identity centrality. The

scale included 10 items scaled from 1 (*strongly disagree*) to 7 (*strongly agree*). An example item was, “Being a *Black woman* is important to my self-image.” Higher scores indicated higher levels of gendered racial identity centrality.

Today we will use the simulated data to evaluate a model that was suggested by a figure in the journal (one IV, four mediators, two dependent variables) but was not tested as a single structural model. Rather, the authors ran a series of four simple mediations per dependent variable for a total of eight separate analyses. The authors do not elaborate on their rationale for this approach. My guess is that they were limited in design by their use of ordinary least squares with the PROCESS macro in SPSS. Additionally, they may have been concerned about power when they considered a more complicated, latent variable, design.

Specifically, we will specify a parallel mediation model where two dependent variables (i.e., mental health, physical health) are predicted directly from gendered racial microaggressions and indirectly through four coping strategies (i.e., spirituality, social support, engagement, disengagement).

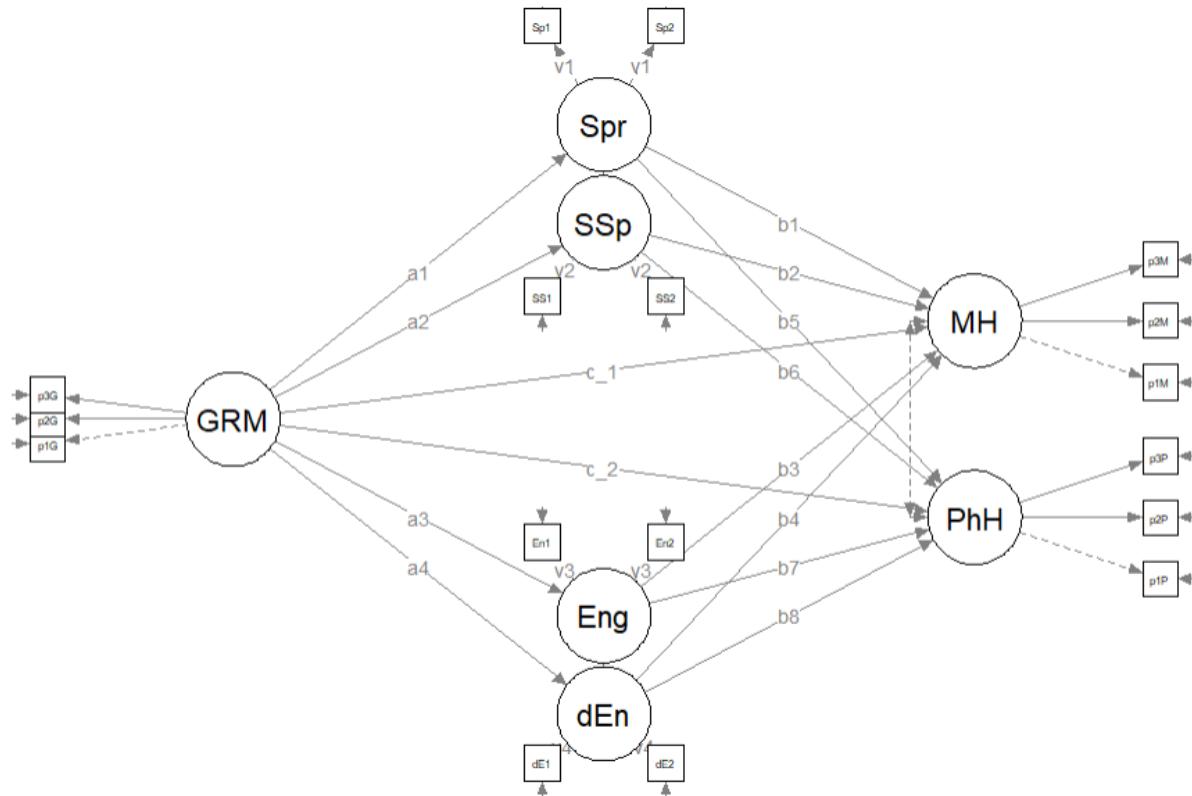


Figure 11.1: An image of the hypothesized (original) model being evaluated in this lesson.

11.4.1 Simulating the data from the journal article

The `lavaan::simulateData` function was used. If you have taken psychometrics, you may recognize the code as one that creates latent variables from item-level data. In trying to be as authentic as possible, we retrieved factor loadings from psychometrically oriented articles that evaluated the measures [Nadal, 2011, Veit and Ware, 1983]. For all others we specified a factor loading of 0.80.

We then approximated the *measurement model* by specifying the correlations between the latent variable. We sourced these from the correlation matrix from the research vignette [Lewis et al., 2017]. The process created data with multiple decimals and values that exceeded the boundaries of the variables. For example, in all scales there were negative values. Therefore, the final element of the simulation was a linear transformation that rescaled the variables back to the range described in the journal article and rounding the values to integer (i.e., with no decimal places).

#Entering the intercorrelations, means, and standard deviations from the journal article

```
Lewis_generating_model <- '
  ##measurement model
  GRMS  =~ .69*Ob1 + .69*Ob2 + .60*Ob3 + .59*Ob4 + .55*Ob5 + .55*Ob6 + .54*Ob7 + .50*Ob8
  MntlHlth =~ .8*MH1 + .8*MH2 + .8*MH3 + .8*MH4 + .8*MH5 + .8*MH6
  PhysHlth  =~ .8*PhH1 + .8*PhH2 + .8*PhH3 + .8*PhH4 + .8*PhH5 + .8*PhH6
  Spirituality  =~ .8*Spirit1 + .8*Spirit2
  SocSupport  =~ .8*SocS1 + .8*SocS2
  Engagement  =~ .8*Eng1 + .8*Eng2
  Disengagement  =~ .8*dEng1 + .8*dEng2
  GRIC  =~ .8*Cntrlty1 + .8*Cntrlty2 + .8*Cntrlty3 + .8*Cntrlty4 + .8*Cntrlty5 + .8*Cntrlty6

  # Means
  GRMS ~ 1.99*1
  Spirituality ~ 2.82*1
  SocSupport ~ 2.48*1
  Engagement ~ 2.32*1
  Disengagement ~ 1.75*1
  GRIC ~ 5.71*1
  MntlHlth ~ 3.56*1 #Lewis et al used sums instead of means, I recast as means to facilitate comparison
  PhysHlth ~ 3.51*1 #Lewis et al used sums instead of means, I recast as means to facilitate comparison

  # Correlations
  GRMS ~ 0.20*Spirituality
  GRMS ~ 0.28*SocSupport
  GRMS ~ 0.30*Engagement
  GRMS ~ 0.41*Disengagement
  GRMS ~ 0.19*GRIC
  GRMS ~ -0.32*MntlHlth
  GRMS ~ -0.18*PhysHlth

  Spirituality ~ 0.49*SocSupport
  Spirituality ~ 0.57*Engagement
  Spirituality ~ 0.22*Disengagement
  Spirituality ~ 0.12*GRIC
  Spirituality ~ -0.06*MntlHlth
  Spirituality ~ -0.13*PhysHlth

  SocSupport ~ 0.46*Engagement
```

```

SocSupport ~ 0.26*Disengagement
SocSupport ~ 0.38*GRIC
SocSupport ~ -0.18*MntlHlth
SocSupport ~ -0.08*PhysHlth

Engagement ~ 0.37*Disengagement
Engagement ~ 0.08*GRIC
Engagement ~ -0.14*MntlHlth
Engagement ~ -0.06*PhysHlth

Disengagement ~ 0.05*GRIC
Disengagement ~ -0.54*MntlHlth
Disengagement ~ -0.28*PhysHlth

GRIC ~ -0.10*MntlHlth
GRIC ~ 0.14*PhysHlth

MntlHlth ~ 0.47*PhysHlth
'

set.seed(230925)
dfLewis <- lavaan::simulateData(model = Lewis_generating_model,
                                 model.type = "sem",
                                 meanstructure = T,
                                 sample.nobs=231,
                                 standardized=FALSE)

#used to retrieve column indices used in the rescaling script below
#col_index <- as.data.frame(colnames(dfLewis))

for(i in 1:ncol(dfLewis)){ # for loop to go through each column of the dataframe
  if(i >= 1 & i <= 25){ # apply only to GRMS variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 5))
  }
  if(i >= 26 & i <= 37){ # apply only to mental and physical health variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(0, 6))
  }
  if(i >= 38 & i <= 45){ # apply only to coping variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 4))
  }
  if(i >= 46 & i <= 55){ # apply only to GRIC variables
    dfLewis[,i] <- scales::rescale(dfLewis[,i], c(1, 7))
  }
}
}

#rounding to integers so that the data resembles that which was collected
library(tidyverse)

```

```
dfLewis <- dfLewis %>% round(0)

#quick check of my work
#psych::describe(dfLewis)
```

The script below allows you to store the simulated data as a file on your computer. This is optional – the entire lesson can be worked with the simulated data.

If you prefer the .rds format, use this script (remove the hashtags). The .rds format has the advantage of preserving any formatting of variables. A disadvantage is that you cannot open these files outside of the R environment.

Script to save the data to your computer as an .rds file.

```
# saveRDS(dfLewis, 'dfLewis.rds')
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfLewis<- readRDS('dfLewis.rds')
```

If you prefer the .csv format (think “Excel lite”) use this script (remove the hashtags). An advantage of the .csv format is that you can open the data outside of the R environment. A disadvantage is that it may not retain any formatting of variables

Script to save the data to your computer as a .csv file.

```
# write.table(dfLewis, file = 'dfLewis.csv', sep = ',',
# col.names=TRUE, row.names=FALSE)
```

Once saved, you could clean your environment and bring the data back in from its .csv format.

```
# dfLewis<- read.csv ('dfLewis.csv', header = TRUE)
```

11.5 Scrubbing, Scoring, and Data Diagnostics

Because the focus of this lesson is on the specific topic of specifying and evaluating a structural model for SEM and have used simulated data, we are skipping many of the steps in scrubbing, scoring and data diagnostics. If this were real, raw, data, it would be important to [scrub](#), if needed [score](#), and conduct [data diagnostics](#) to evaluate the suitability of the data for the proposed analyses.

11.6 Script for Specifying Models in *lavaan*

SEM in *lavaan* requires fluency with the R script. Below is a brief overview of the operators we use most frequently:

- Latent variables (factors) must be *defined* by their manifest or latent indicators.
 - the special operator ($=\sim$, *is measured/defined by*) is used for this
 - Example: $f1 =\sim y1 + y2 + y3$
- Regression equations use the single tilda (\sim , *is regressed on*)
 - place DV (y) on left of operator
 - place IVs, separate by + on the right
 - Example: $y \sim f1 + f2 + x1 + x2$
 - * f is a latent variable in this example
 - * y , $x1$, and $x2$ are observed variables in this example
 - An asterisk can affix a label in subsequent calculations and in interpreting output
- 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: $f1 \sim\sim f2 *Intercepts (\sim 1)$ 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. Readability of model syntax is improved by:

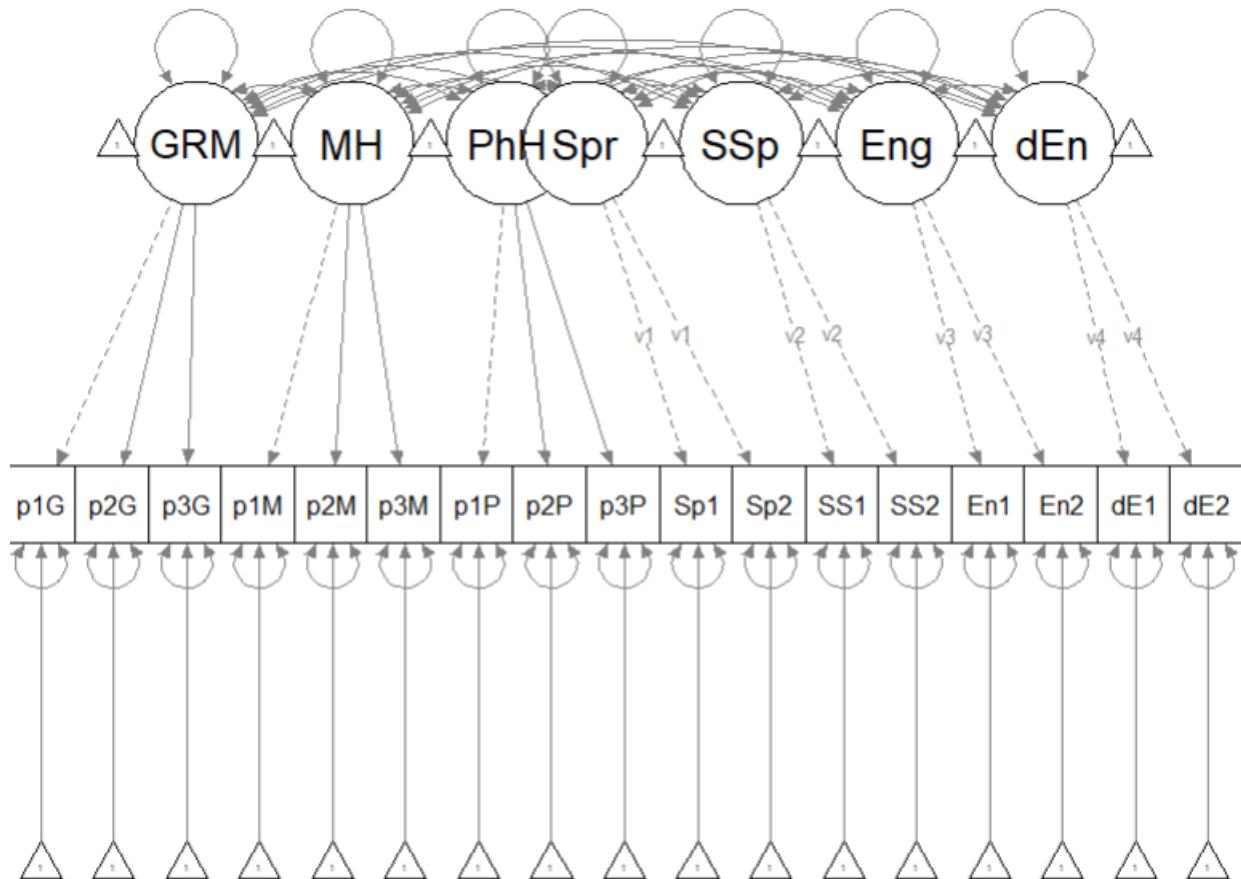
- splitting formulas over multiple lines
- using blank lines within single quote
- labeling with the hashtag

11.7 Quick Specification of the Measurement Model

Recall that the first step in establishing a structural model is to specify, evaluate, and if necessary re-specify the measurement model. For this data I have specified the measurement model as follows:

- Gendered Racial Microaggressions Scale (GRMS): randomly assign the 26 items to 3 parcels.
- Mental health (MH): randomly assign the 6 items to 3 parcels.
- Physical health (PhH): randomly assign the 6 items to 3 parcels.
- Coping strategies (spiritual/Spr, social support (SSp), engagement (Eng), disengagement (dEng)): Constrain the loadings for each of the two variables per construct to be equal.

For more information on establishing measurement models please visit the lesson on [establishing the measurement model](#). Here is a representation of the measurement model we are specifying.



To proceed with this approach, I first need to create parcels for the GRMS, MH, and PhH scales. This code randomly assigns the GRMS items to three parcels.

```
set.seed(230916)
items <- c("Ob1", "Ob2", "Ob3", "Ob4", "Ob5", "Ob6", "Ob7", "Ob8", "Ob9",
          "Ob10", "Ma1", "Ma2", "Ma3", "Ma4", "Ma5", "Ma6", "Ma7", "St1", "St2",
          "St3", "St4", "St5", "An1", "An2")
parcels <- c("GRMS_p1", "GRMS_p_2", "GRMS_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))
```

```
##      items    parcel
## 1     Ma3  GRMS_p1
## 2     Ob7  GRMS_p_2
## 3     Ob9  GRMS_p3
## 4     Ma7  GRMS_p1
## 5     Ma5  GRMS_p_2
## 6     Ob2  GRMS_p3
## 7     Ma1  GRMS_p1
## 8     An1  GRMS_p_2
## 9     Ob4  GRMS_p3
## 10    Ob1  GRMS_p1
## 11    St3  GRMS_p_2
```

```
## 12  Ob6  GRMS_p3
## 13  St2  GRMS_p1
## 14  Ob5  GRMS_p_2
## 15  Ob10 GRMS_p3
## 16  Ob8  GRMS_p1
## 17  Ma6  GRMS_p_2
## 18  St5  GRMS_p3
## 19  An2  GRMS_p1
## 20  Ma2  GRMS_p_2
## 21  Ob3  GRMS_p3
## 22  Ma4  GRMS_p1
## 23  St1  GRMS_p_2
## 24  St4  GRMS_p3
```

We can now create the parcels using the same scoring procedure as we did for the REMS and CMI instruments.

```
GRMS_p1_vars <- c("Ma3", "Ma7", "Ma1", "Ob1", "St2", "Ob8", "An2", "Ma4")
GRMS_p2_vars <- c("Ob7", "Ma5", "An1", "St3", "Ob5", "Ma6", "Ma2", "St1")
GRMS_p3_vars <- c("Ob9", "Ob2", "Ob4", "Ob6", "Ob10", "St5", "Ob3", "St4")

dfLewis$p1GRMS <- sjstats::mean_n(dfLewis[, GRMS_p1_vars], 0.75)
dfLewis$p2GRMS <- sjstats::mean_n(dfLewis[, GRMS_p2_vars], 0.75)
dfLewis$p3GRMS <- sjstats::mean_n(dfLewis[, GRMS_p3_vars], 0.75)

# If the scoring code above does not work for you, try the format
# below which involves inserting periods in front of the variable
# list. One example is provided. dfLewis$p3PWB <-
# sjstats::mean_n(dfLewis[, .PWB_p3_vars], .75)
```

This code randomly assigns the mental health items to three parcels.

```
set.seed(230916)
items <- c("MH1", "MH2", "MH3", "MH4", "MH5", "MH6")
parcels <- c("MH_p1", "MH_p2", "MH_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))

##   items parcel
## 1   MH5  MH_p1
## 2   MH1  MH_p2
## 3   MH4  MH_p3
## 4   MH6  MH_p1
## 5   MH3  MH_p2
## 6   MH2  MH_p3
```

This code provides means for each of the three REMS parcels.

```

MH_p1_vars <- c("MH5", "MH6")
MH_p2_vars <- c("MH1", "MH3")
MH_p3_vars <- c("MH4", "MH2")

dfLewis$p1MH <- sjstats::mean_n(dfLewis[, MH_p1_vars], 0.75)
dfLewis$p2MH <- sjstats::mean_n(dfLewis[, MH_p2_vars], 0.75)
dfLewis$p3MH <- sjstats::mean_n(dfLewis[, MH_p3_vars], 0.75)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfLewis$p3REMS <-
# sjstats::mean_n(dfLewis[, ..REMS_p3_vars], .80)

```

This code randomly assigns the physical health items to three parcels.

```

set.seed(230916)
items <- c("PhH1", "PhH2", "PhH3", "PhH4", "PhH5", "PhH6")
parcels <- c("PhH_p1", "PhH_p2", "PhH_p3")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))

##   items parcel
## 1 PhH5 PhH_p1
## 2 PhH1 PhH_p2
## 3 PhH4 PhH_p3
## 4 PhH6 PhH_p1
## 5 PhH3 PhH_p2
## 6 PhH2 PhH_p3

```

This code provides means for each of the three REMS parcels.

```

PhH_p1_vars <- c("PhH5", "PhH6")
PhH_p2_vars <- c("PhH1", "PhH3")
PhH_p3_vars <- c("PhH4", "PhH2")

dfLewis$p1PhH <- sjstats::mean_n(dfLewis[, PhH_p1_vars], 0.75)
dfLewis$p2PhH <- sjstats::mean_n(dfLewis[, PhH_p2_vars], 0.75)
dfLewis$p3PhH <- sjstats::mean_n(dfLewis[, PhH_p3_vars], 0.75)

# If the scoring code above does not work for you, try the format
# below which involves inserting two periods in front of the variable
# list. One example is provided. dfLewis$p3REMS <-
# sjstats::mean_n(dfLewis[, ..REMS_p3_vars], .80)

```

Below is code for specifying the measurement model. Each of the latent variables/factors (REMS, CMI, PWB) is identified by three parcels. Each of the latent variables is allowed to covary with the others.

```

msmt_mod <- "
  ##measurement model
  GRMS =~ p1GRMS + p2GRMS + p3GRMS
  MH =~ p1MH + p2MH + p3MH
  PhH =~ p1PhH + p2PhH + p3PhH
  Spr =~ v1*Spirit1 + v1*Spirit2
  SSp =~ v2*SocS1 + v2*SocS2
  Eng =~ v3*Eng1 + v3*Eng2
  dEng =~ v4*dEng1 + v4*dEng2

  # Covariances
  GRMS ~~ MH
  GRMS ~~ PhH
  GRMS ~~ Spr
  GRMS ~~ SSp
  GRMS ~~ Eng
  GRMS ~~ dEng
  MH ~~ PhH
  MH ~~ Spr
  MH ~~ SSp
  MH ~~ Eng
  MH ~~ dEng
  PhH ~~ Spr
  PhH ~~ SSp
  PhH ~~ Eng
  PhH ~~ dEng
  Spr ~~ SSp
  Spr ~~ Eng
  Spr ~~ dEng
  SSp ~~ Eng
  SSp ~~ dEng
  Eng ~~ dEng

  "

set.seed(230916)
msmt_fit <- lavaan::cfa(msmt_mod, data = dfLewis, missing = "fiml")
msmt_fit_sum <- lavaan::summary(msmt_fit, fit.measures = TRUE, standardized = TRUE)
msmt_fit_pEsts <- lavaan::parameterEstimates(msmt_fit, boot.ci.type = "bca.simple",
                                              standardized = TRUE)
msmt_fit_sum

## lavaan 0.6.16 ended normally after 106 iterations
##                                     ML
##   Estimator
##   Optimization method          NLMINB

```

```

## Number of model parameters          68
## 
## Number of observations             231
## Number of missing patterns        1
## 
## Model Test User Model:
## 
##   Test statistic                  94.122
##   Degrees of freedom              102
##   P-value (Chi-square)           0.698
## 
## Model Test Baseline Model:
## 
##   Test statistic                 2104.157
##   Degrees of freedom              136
##   P-value                         0.000
## 
## User Model versus Baseline Model:
## 
##   Comparative Fit Index (CFI)     1.000
##   Tucker-Lewis Index (TLI)         1.005
## 
##   Robust Comparative Fit Index (CFI) 1.000
##   Robust Tucker-Lewis Index (TLI)    1.005
## 
## Loglikelihood and Information Criteria:
## 
##   Loglikelihood user model (H0)    -3376.747
##   Loglikelihood unrestricted model (H1) -3329.686
## 
##   Akaike (AIC)                   6889.494
##   Bayesian (BIC)                  7123.579
##   Sample-size adjusted Bayesian (SABIC) 6908.057
## 
## Root Mean Square Error of Approximation:
## 
##   RMSEA                           0.000
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.028
##   P-value H_0: RMSEA <= 0.050      1.000
##   P-value H_0: RMSEA >= 0.080      0.000
## 
##   Robust RMSEA                     0.000
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.028
##   P-value H_0: Robust RMSEA <= 0.050 1.000
##   P-value H_0: Robust RMSEA >= 0.080 0.000
## 
```

```

## Standardized Root Mean Square Residual:
##
##      SRMR          0.034
##
## Parameter Estimates:
##
##      Standard errors           Standard
##      Information             Observed
##      Observed information based on   Hessian
##
## Latent Variables:
##      Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##      GRMS =~
##      p1GRMS     1.000
##      p2GRMS     1.014    0.030 34.037  0.000  0.726  0.956
##      p3GRMS     0.904    0.030 30.069  0.000  0.656  0.934
##      MH =~
##      p1MH       1.000
##      p2MH       1.237    0.128  9.696  0.000  0.773  0.728
##      p3MH       1.241    0.112 11.044  0.000  0.775  0.776
##      PhH =~
##      p1PhH      1.000
##      p2PhH      0.997    0.125  7.959  0.000  0.598  0.714
##      p3PhH      1.060    0.139  7.643  0.000  0.636  0.727
##      Spr =~
##      Spirit1    (v1)  1.000
##      Spirit2    (v1)  1.000
##      SSp =~
##      SocS1      (v2)  1.000
##      SocS2      (v2)  1.000
##      Eng =~
##      Eng1       (v3)  1.000
##      Eng2       (v3)  1.000
##      dEng =~
##      dEng1     (v4)  1.000
##      dEng2     (v4)  1.000
##
## Covariances:
##      Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##      GRMS ~~
##      MH        -0.290    0.042 -6.868  0.000 -0.639 -0.639
##      PhH       -0.185    0.039 -4.698  0.000 -0.425 -0.425
##      Spr        0.289    0.034  8.445  0.000  0.815  0.815
##      SSp        0.212    0.030  7.160  0.000  0.689  0.689
##      Eng        0.204    0.029  7.054  0.000  0.689  0.689
##      dEng       0.202    0.029  7.008  0.000  0.680  0.680
##      MH ~~
##      PhH        0.200    0.040  5.041  0.000  0.533  0.533

```

```

##   Spr      -0.155  0.030  -5.177  0.000  -0.507  -0.507
##   SSp      -0.113  0.027  -4.222  0.000  -0.427  -0.427
##   Eng      -0.090  0.025  -3.558  0.000  -0.353  -0.353
##   dEng     -0.168  0.028  -6.022  0.000  -0.656  -0.656
##   PhH ~~
##   Spr      -0.090  0.028  -3.205  0.001  -0.307  -0.307
##   SSp      -0.069  0.026  -2.699  0.007  -0.271  -0.271
##   Eng      -0.052  0.025  -2.111  0.035  -0.212  -0.212
##   dEng     -0.101  0.026  -3.830  0.000  -0.411  -0.411
##   Spr ~~
##   SSp      0.153  0.023  6.739  0.000  0.738  0.738
##   Eng      0.140  0.022  6.409  0.000  0.704  0.704
##   dEng     0.129  0.022  5.998  0.000  0.646  0.646
##   SSp ~~
##   Eng      0.076  0.019  4.005  0.000  0.439  0.439
##   dEng     0.070  0.019  3.693  0.000  0.400  0.400
##   Eng ~~
##   dEng     0.088  0.019  4.657  0.000  0.525  0.525
##
## Intercepts:
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1GRMS      2.592  0.050 51.897  0.000  2.592  3.415
## .p2GRMS      2.545  0.050 50.567  0.000  2.545  3.327
## .p3GRMS      2.579  0.046 55.829  0.000  2.579  3.673
## .p1MH        3.582  0.054 66.695  0.000  3.582  4.388
## .p2MH        2.866  0.070 41.068  0.000  2.866  2.702
## .p3MH        3.035  0.066 46.200  0.000  3.035  3.040
## .p1PhH       3.128  0.060 52.418  0.000  3.128  3.449
## .p2PhH       2.652  0.055 48.093  0.000  2.652  3.164
## .p3PhH       3.067  0.058 53.298  0.000  3.067  3.507
## .Spirit1     2.511  0.044 57.110  0.000  2.511  3.758
## .Spirit2     2.437  0.042 57.882  0.000  2.437  3.808
## .SocS1       2.550  0.040 64.254  0.000  2.550  4.228
## .SocS2       2.758  0.043 64.576  0.000  2.758  4.249
## .Eng1        2.437  0.044 55.427  0.000  2.437  3.647
## .Eng2        2.515  0.038 65.479  0.000  2.515  4.308
## .dEng1       2.502  0.039 63.518  0.000  2.502  4.179
## .dEng2       2.455  0.041 59.357  0.000  2.455  3.905
## GRMS        0.000                               0.000  0.000
## MH          0.000                               0.000  0.000
## PhH         0.000                               0.000  0.000
## Spr         0.000                               0.000  0.000
## SSp         0.000                               0.000  0.000
## Eng         0.000                               0.000  0.000
## dEng        0.000                               0.000  0.000
##
## Variances:
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all

```

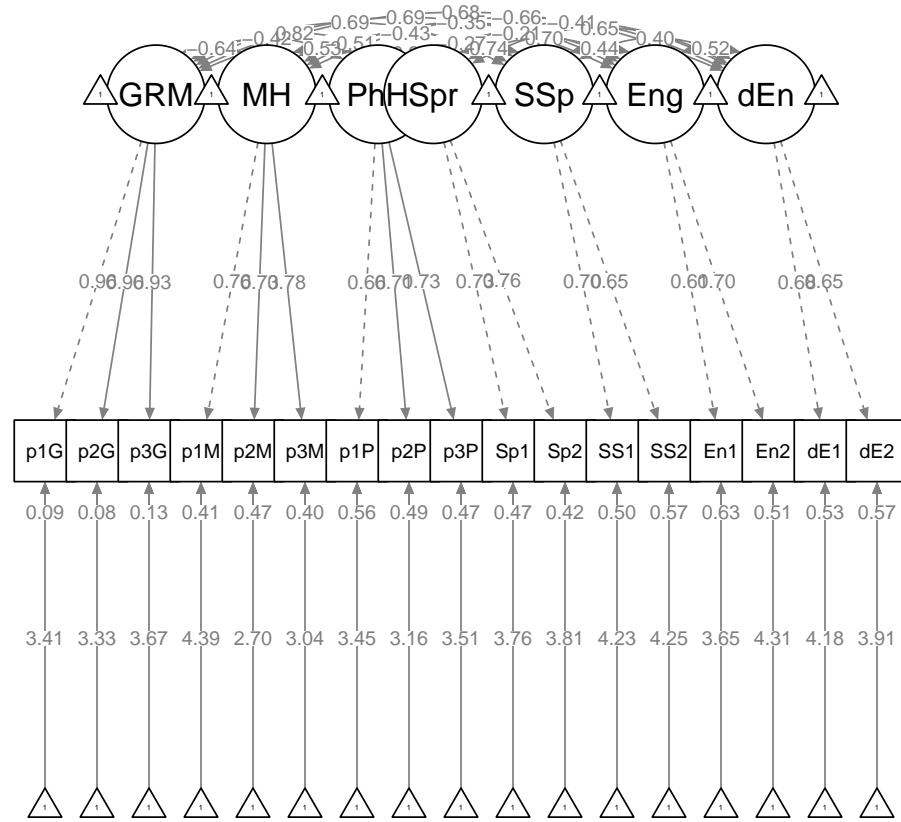
##	.p1GRMS	0.050	0.008	6.600	0.000	0.050	0.086
##	.p2GRMS	0.044	0.007	6.015	0.000	0.044	0.076
##	.p3GRMS	0.063	0.008	8.289	0.000	0.063	0.128
##	.p1MH	0.277	0.037	7.437	0.000	0.277	0.415
##	.p2MH	0.528	0.067	7.935	0.000	0.528	0.469
##	.p3MH	0.396	0.055	7.225	0.000	0.396	0.397
##	.p1PhH	0.463	0.058	7.996	0.000	0.463	0.562
##	.p2PhH	0.344	0.049	7.098	0.000	0.344	0.490
##	.p3PhH	0.360	0.054	6.723	0.000	0.360	0.471
##	.Spirit1	0.208	0.026	8.152	0.000	0.208	0.467
##	.Spirit2	0.171	0.023	7.433	0.000	0.171	0.419
##	.SocS1	0.183	0.025	7.225	0.000	0.183	0.504
##	.SocS2	0.241	0.029	8.235	0.000	0.241	0.572
##	.Eng1	0.279	0.032	8.627	0.000	0.279	0.625
##	.Eng2	0.173	0.025	6.893	0.000	0.173	0.509
##	.dEng1	0.191	0.025	7.482	0.000	0.191	0.532
##	.dEng2	0.227	0.028	8.127	0.000	0.227	0.575
##	GRMS	0.527	0.054	9.796	0.000	1.000	1.000
##	MH	0.390	0.062	6.241	0.000	1.000	1.000
##	PhH	0.360	0.074	4.854	0.000	1.000	1.000
##	Spr	0.238	0.032	7.412	0.000	1.000	1.000
##	SSp	0.180	0.028	6.378	0.000	1.000	1.000
##	Eng	0.167	0.028	6.042	0.000	1.000	1.000
##	dEng	0.168	0.027	6.208	0.000	1.000	1.000

```
# msmt_fit_pEsts #although creating the object is useful to export as
# a .csv I didn't ask it to print into the book
```

The factor loadings were all strong, statistically significant, and properly valenced. Further, global fit statistics were within acceptable thresholds ($\chi^2(102) = 94.122, p = 0.698, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.028], SRMR = 0.034$).

The figure below is an illustration of our measurement model with its results. It also conveys that each latent variable is indicated by three parcels and all of the latent variables are allowed to covary.

```
semPlot::semPaths(msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
  node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(5,
  5, 5, 5))
```



```
# semPlot::semPaths(msmt_fit) #ignore -- used to create a no-results
# figure earlier in the chapter
```

Below is script that will export the global fit indices (via `tidySEM::table_fit`) and the parameter estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```
# global fit indices
msmt_fitstats <- tidySEM::table_fit(msmt_fit)
```

```
## Registered S3 method overwritten by 'tidySEM':
##   method           from
##   predict.MxModel OpenMx
```

```
write.csv(msmt_fitstats, file = "msmt_fitstats.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(msmt_fit_pEsts, file = "msmt_pEsts.csv")
```

Here's how I might write an APA style summary of establishing the measurement model.

Analyzing our proposed multiple mediator model followed the two-step procedure of first evaluating a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index (CFI) ≥ 0.95 , root-mean-square error of approximation (RMSEA) ≤ 0.06 , and the standard root-mean-square residual (SRMR) ≤ 0.08 . To establish acceptable fit, we used CFI ≥ 0.90 , RMSEA ≤ 0.10 , and SRMR ≤ 0.10 [Weston and Gore, 2006].

To evaluate the measurement model we followed recommendations by Little et al. [Little et al., 2002, 2013]. Specifically, each latent variable with six or more indicators was represented by three parcels. Parcels were created by randomly assigning scale items to the parcels and then calculating the mean, if at least 65% of the items were non-missing. For the four latent variables with only two indicators each, we constrained the factor loadings to be equal. Factor loadings were all strong, statistically significant, and properly valenced. Global fit statistics were within acceptable thresholds ($\chi^2(102) = 94.122, p = 0.698, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.028], SRMR = 0.034$). Thus, we proceeded to testing the structural model.

Table 1

Factor Loadings for the Measurement Model				
Latent variable and indicator	est	SE	p	est_std
Gendered Racial Microaggressions				
Parcel 1	1.000	0.000		0.956
Parcel 2	1.014	0.030	<0.001	0.961
Parcel 3	1.904	0.030	<0.001	0.934
Mental Health				
Parcel 1	1.000	0.000		0.765
Parcel 2	1.237	0.128	<0.001	0.728
Parcel 3	1.241	0.112	<0.001	0.776
Physical Health				
Parcel 1	1.000	0.000		0.662
Parcel 2	0.997	0.125	<0.001	0.714
Parcel 3	1.060	0.139	<0.001	0.727

Spiritual Coping			
Item 1	1.000	0.000	0.730
Item 2	1.000	0.000	0.762
Social Support Coping			
Item 1	1.000	0.000	0.704
Item 2	1.000	0.000	0.654
Engagement Coping			
Item 1	1.000	0.000	0.612
Item 2	1.000	0.000	0.701
Disengagement Coping			
Item 1	1.000	0.000	0.684
Item 2	1.000	0.000	0.652

Having established that our measurement model is adequate, we are ready to replace the covariances between latent variables with the paths (directional) and covariances (bidirectional) we hypothesize. These paths and covariances are *soft* hypotheses. That is, we are “freeing” them to relate. In SEM, *hard* hypotheses are where no path/covariance exists and the relationship between these variables is “fixed” to zero. This is directly related to degrees of freedom and the identification status (just-identified, over-identified, underidentified) of the model.

11.8 Specifying and Evaluating the Hypothesized Structural Model

As described more completely in the [lesson on specifying and evaluating the structural model](#), the **structural model** evaluates the hypothesized relations between the latent variables. The structural model is typically more parsimonious (i.e., not saturated) than the measurement model and is characterized by directional paths (not covariances) between some (not all) of the variables.

Here’s a quick reminder of the hypothesized model we are testing. Using data simulated from the Lewis et al. [2017] article, we are evaluating a parallel mediation model, predicting mental and physical health directly from gendered racial microaggressions and indirectly through four approaches to coping (i.e., spiritual, social support, engagement, disengagement). This model is *hybrid* because it include measurement models (i.e., latent variables indicated by their parcels), plus the hypothesized paths.

11.8.1 Model Identification for the Hypothesized (Original) Model

In order to be evaluated, structural models need to be *just identified* ($df_M = 0$) or *overidentified* ($df_M > 0$). Computer programs are not (yet) good at estimating identification status because it is based on symbolism and not numbers. Therefore, we researchers must do the mental math to ensure that our *knowns* (measured/observed variables) are equal (just-identified) or greater than (overidentified) our *unknowns* (parameters that will be estimated). Model identification is described more completely in the [lesson on specifying and evaluating the structural model](#),

Knowns: $\frac{k(k+1)}{2}$ where k is the number of *constructs* (humoR: konstrukts?) in the model. In our case, we have seven constructs. Deploying the formula below, we learn that we have 21 knowns.

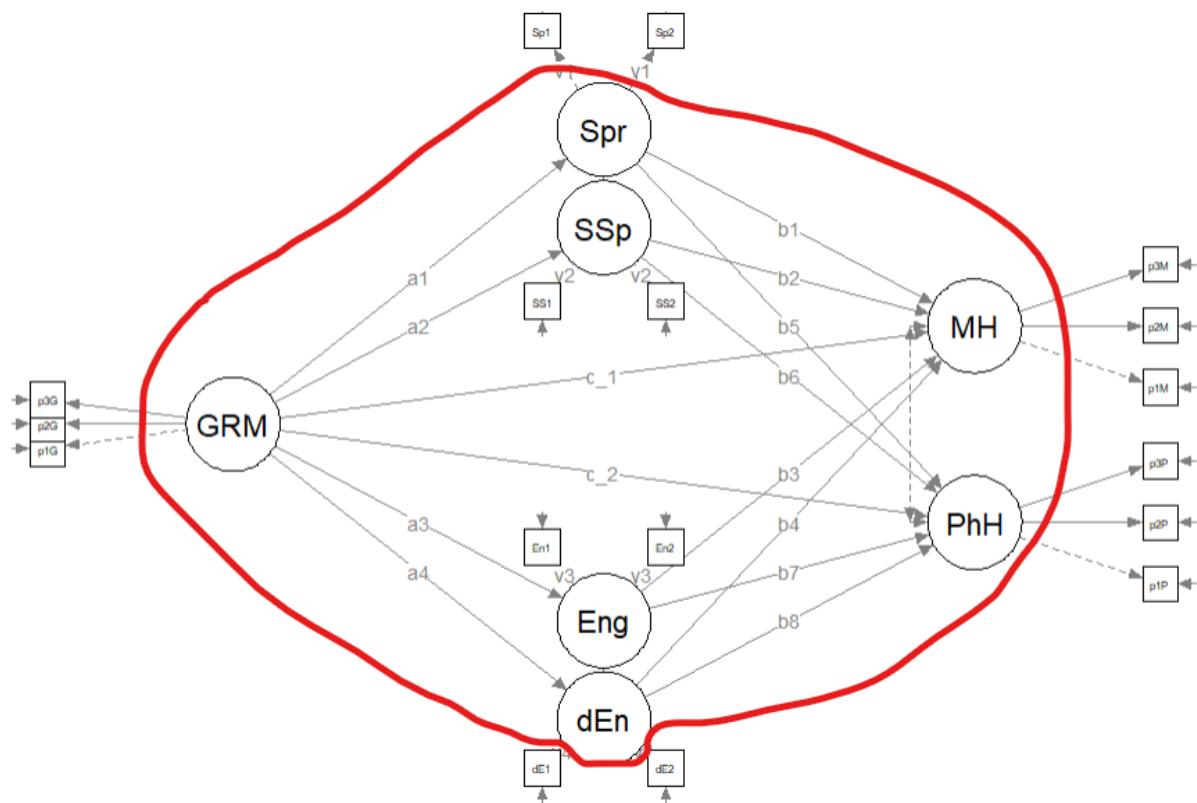


Figure 11.2: An image of the hypothesized (original) model being evaluated in this lesson; the structural model is outlined in red.

```
(7*(7+1))/2
```

```
## [1] 28
```

Unknowns: are calculated with the following

- Exogenous (predictor) variables (1 variance estimated for each): we have 1 (GRMS)
- Endogenous (predicted) variables (1 disturbance variance for each): we have 6 (Spr, SSp, Eng, dEng, MH, PhH)
- Correlations between variables (1 covariance for each pairing): we have 0
- Regression paths (arrows linking exogenous variables to endogenous variables): we have 14

With 28 knowns and 20 unknowns, we have 8 degrees of freedom in the structural portion of the model. This is an *over-identified* model.

11.8.1.1 Specifying and Evaluating the Structural Model

Specifying our structural model in *lavaan* includes script for the measurement model, the structural model, and any additional model parameters (e.g., indirect and total effects) that we might add. In the script below you will see each of these elements.

- the mediating variables (Spr, SSp, Eng, dEng) are predicted by the independent variable (GRMS),
- the dependent variables (MH, PhH) are predicted by the independent variable (GRMS) and the mediating variables (Spr, SSp, Eng, dEng),
- labels are assigned to represent the a , b , and c' paths
- calculations that use the labels will estimate the indirect, direct, and total paths

In the model specification below, there are more elements to note. I have chosen to specify the dependent variables with all of the variables that predict them in a single line of code. In the script below I wrote:

```
MH ~ c_p1*GRMS + b1*Spr + b2*SSp + b3*Eng + b4*dEng
```

It is equally acceptable to list them specify them from fewer predictors at a time. For example, I could have written

```
MH ~ c_p1*GRMS
MH ~ b1*Spr
MH ~ b2*SSp
MH ~ b3*Eng
MH ~ b4*dEng
```

Because *lavaan* has elements of randomness in its algorithms (particularly around its version of bias-corrected, bootstrapped confidence intervals), including a *set.seed* function will facilitate the reproducibility of results.

If the data contain missing values, the default behavior in *lavaan::sem* is listwise deletion. If we can presume that the missing mechanism is MCAR or MAR (e.g., there is no systematic missingness), we can specify a *full information maximum likelihood* (FIML) estimation procedure with the *missing = "fiml"* argument. Recall that we retained cases if they had 20% or less missing. Using the "fiml" option is part of the AIA approach [Parent, 2013].

In the *lavaan::summary* function, we will want to retrieve the global fit indices with the *fit.measures=TRUE*. Because SEM figures are often represented with standardized values, we will want *standardized = TRUE*. And if we wish to know the proportion of variance predicted in our endogenous variables, we will include *rsq = TRUE*.

In the *lavaan::parameterEstimates* we can obtain *lavaan*'s version of bias-corrected bootstrapped confidence intervals (they aren't quite the same) by including *boot.ci.type = "bca.simple"*.

```
struct_mod1 <- "
  ##measurement model
  GRMS =~ p1GRMS + p2GRMS + p3GRMS
  MH =~ p1MH + p2MH + p3MH
  PhH =~ p1PhH + p2PhH + p3PhH
  Spr =~ v1*Spirit1 + v1*Spirit2
  SSp =~ v2*SocS1 + v2*SocS2
  Eng =~ v3*Eng1 + v3*Eng2
  dEng =~ v4*dEng1 + v4*dEng2

  #structural model with labels for calculation of the indirect effect

  Eng ~ a1*GRMS
  Spr ~ a2*GRMS
  SSp ~ a3*GRMS
  dEng ~ a4*GRMS

  MH ~ c_p1*GRMS + b1*Eng + b2*Spr + b3*SSp + b4*dEng
  PhH ~ c_p2*GRMS + b5*Eng + b6*Spr + b7*SSp + b8*dEng

  #cov
  MH ~~ 0*PhH #prevents MH and PhD from correlating

  #calculations
  indirect.EngMH := a1*b1
  indirect.SprMH := a2*b2
  indirect.SSpMH := a3*b3
  indirect.dEngMH := a4*b4"
```

```

indirect.EngPhH := a1*b5
indirect.SprPhH := a2*b6
indirect.SSpPhH := a3*b7
indirect.dEngPhH := a4*b8
direct.MH := c_p1
direct.PhH := c_p2
total.MH := c_p1 + (a1*b1) + (a2*b2) + (a3*b3) + (a4*b4)
total.PhH := c_p2 + (a1*b5) + (a1*b6) + (a1*b7) + (a1*b8)
"

set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
struct_fit1 <- lavaan::sem(struct_mod1, data = dfLewis, missing = "fiml")
struct_summary1 <- lavaan::summary(struct_fit1, fit.measures = TRUE, standardized = TRUE,
                                     rsq = TRUE)
struct_pEsts1 <- lavaan::parameterEstimates(struct_fit1, boot.ci.type = "bca.simple",
                                             standardized = TRUE)
struct_summary1

## lavaan 0.6.16 ended normally after 117 iterations
##
## Estimator                               ML
## Optimization method                   NLMINB
## Number of model parameters           61
##
## Number of observations                231
## Number of missing patterns            1
##
## Model Test User Model:
##
## Test statistic                         120.324
## Degrees of freedom                     109
## P-value (Chi-square)                  0.216
##
## Model Test Baseline Model:
##
## Test statistic                         2104.157
## Degrees of freedom                      136
## P-value                                0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)           0.994
## Tucker-Lewis Index (TLI)              0.993
##
## Robust Comparative Fit Index (CFI)    0.994
## Robust Tucker-Lewis Index (TLI)       0.993
##
## Loglikelihood and Information Criteria:

```

```

## 
## Loglikelihood user model (H0)           -3389.848
## Loglikelihood unrestricted model (H1)   -3329.686
## 
## Akaike (AIC)                          6901.696
## Bayesian (BIC)                        7111.683
## Sample-size adjusted Bayesian (SABIC)  6918.348
## 
## Root Mean Square Error of Approximation:
## 
## RMSEA                               0.021
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.041
## P-value H_0: RMSEA <= 0.050          0.995
## P-value H_0: RMSEA >= 0.080          0.000
## 
## Robust RMSEA                         0.021
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.041
## P-value H_0: Robust RMSEA <= 0.050   0.995
## P-value H_0: Robust RMSEA >= 0.080   0.000
## 
## Standardized Root Mean Square Residual:
## 
## SRMR                                0.044
## 
## Parameter Estimates:
## 
## Standard errors                      Standard
## Information                           Observed
## Observed information based on        Hessian
## 
## Latent Variables:
## 
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## GRMS =~
##   p1GRMS      1.000
##   p2GRMS      1.015    0.030 34.097  0.000  0.725  0.955
##   p3GRMS      0.903    0.030 29.818  0.000  0.655  0.932
## MH =~
##   p1MH       1.000
##   p2MH       1.218    0.126  9.663  0.000  0.765  0.721
##   p3MH       1.246    0.113 11.045  0.000  0.783  0.783
## PhH =~
##   p1PhH      1.000
##   p2PhH      1.007    0.128  7.878  0.000  0.608  0.725
##   p3PhH      1.036    0.135  7.664  0.000  0.625  0.715
## Spr =~
##   Spirit1 (v1) 1.000
## 
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## Spirit1 (v1) 1.000
## 
```

```

##    Spirit2   (v1)    1.000
##    SSp      =~
##    SocS1   (v2)    1.000
##    SocS2   (v2)    1.000
##    Eng     =~
##    Eng1    (v3)    1.000
##    Eng2    (v3)    1.000
##    dEng   =~
##    dEng1   (v4)    1.000
##    dEng2   (v4)    1.000
##
## Regressions:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    Eng ~
##    GRMS  (a1)    0.392  0.041  9.506  0.000  0.701  0.701
##    Spr ~
##    GRMS  (a2)    0.554  0.040 13.869  0.000  0.824  0.824
##    SSp ~
##    GRMS  (a3)    0.406  0.042  9.684  0.000  0.694  0.694
##    dEng ~
##    GRMS  (a4)    0.386  0.041  9.400  0.000  0.707  0.707
##    MH ~
##    GRMS  (c_p1) -0.445  0.205 -2.178  0.029 -0.514 -0.514
##    Eng   (b1)    0.355  0.223  1.594  0.111  0.229  0.229
##    Spr   (b2)    0.155  0.219  0.705  0.481  0.120  0.120
##    SSp   (b3)   -0.012  0.187 -0.062  0.950 -0.008 -0.008
##    dEng  (b4)   -0.850  0.303 -2.801  0.005 -0.536 -0.536
##    PhH ~
##    GRMS  (c_p2) -0.369  0.224 -1.650  0.099 -0.444 -0.444
##    Eng   (b5)    0.330  0.250  1.322  0.186  0.222  0.222
##    Spr   (b6)    0.215  0.254  0.845  0.398  0.174  0.174
##    SSp   (b7)    0.016  0.215  0.076  0.939  0.012  0.012
##    dEng  (b8)   -0.624  0.295 -2.115  0.034 -0.409 -0.409
##
## Covariances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    .MH ~~
##    .PhH    0.000
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    .p1GRMS  2.592  0.050 51.897  0.000  2.592  3.415
##    .p2GRMS  2.545  0.050 50.567  0.000  2.545  3.327
##    .p3GRMS  2.579  0.046 55.829  0.000  2.579  3.673
##    .p1MH    3.582  0.054 66.615  0.000  3.582  4.383
##    .p2MH    2.866  0.070 41.024  0.000  2.866  2.699
##    .p3MH    3.035  0.066 46.142  0.000  3.035  3.036
##    .p1PhH   3.128  0.060 52.388  0.000  3.128  3.447

```

```

## .p2PhH      2.652   0.055  48.061   0.000   2.652   3.162
## .p3PhH      3.067   0.058  53.263   0.000   3.067   3.504
## .Spirit1    2.511   0.044  56.810   0.000   2.511   3.738
## .Spirit2    2.437   0.042  58.200   0.000   2.437   3.829
## .SocS1      2.550   0.040  64.175   0.000   2.550   4.222
## .SocS2      2.758   0.043  64.662   0.000   2.758   4.254
## .Eng1       2.437   0.044  55.380   0.000   2.437   3.644
## .Eng2       2.515   0.038  65.524   0.000   2.515   4.311
## .dEng1     2.502   0.040  63.179   0.000   2.502   4.157
## .dEng2     2.455   0.041  59.610   0.000   2.455   3.922
## GRMS        0.000           0.000   0.000   0.000
## .MH         0.000           0.000   0.000   0.000
## .PhH        0.000           0.000   0.000   0.000
## .Spr        0.000           0.000   0.000   0.000
## .SSp        0.000           0.000   0.000   0.000
## .Eng        0.000           0.000   0.000   0.000
## .dEng      0.000           0.000   0.000   0.000
##
## Variances:
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1GRMS    0.050   0.007  6.754   0.000   0.050   0.087
## .p2GRMS    0.044   0.007  6.098   0.000   0.044   0.075
## .p3GRMS    0.064   0.008  8.400   0.000   0.064   0.131
## .p1MH      0.273   0.037  7.381   0.000   0.273   0.409
## .p2MH      0.542   0.067  8.057   0.000   0.542   0.480
## .p3MH      0.387   0.055  7.082   0.000   0.387   0.387
## .p1PhH     0.459   0.058  7.914   0.000   0.459   0.557
## .p2PhH     0.334   0.049  6.817   0.000   0.334   0.474
## .p3PhH     0.375   0.054  6.994   0.000   0.375   0.489
## .Spirit1   0.214   0.026  8.135   0.000   0.214   0.473
## .Spirit2   0.167   0.023  7.223   0.000   0.167   0.413
## .SocS1     0.184   0.026  7.155   0.000   0.184   0.506
## .SocS2     0.240   0.029  8.140   0.000   0.240   0.571
## .Eng1      0.282   0.033  8.558   0.000   0.282   0.631
## .Eng2      0.175   0.026  6.694   0.000   0.175   0.515
## .dEng1    0.205   0.029  7.114   0.000   0.205   0.567
## .dEng2    0.235   0.030  7.701   0.000   0.235   0.599
## GRMS      0.526   0.054  9.787   0.000   1.000   1.000
## .MH        0.164   0.042  3.930   0.000   0.416   0.416
## .PhH      0.255   0.058  4.403   0.000   0.700   0.700
## .Spr        0.076   0.019  4.051   0.000   0.321   0.321
## .SSp        0.094   0.021  4.413   0.000   0.519   0.519
## .Eng        0.084   0.022  3.808   0.000   0.509   0.509
## .dEng      0.078   0.024  3.330   0.001   0.500   0.500
##
## R-Square:
##             Estimate
## p1GRMS      0.913

```

```

##      p2GRMS          0.925
##      p3GRMS          0.869
##      p1MH            0.591
##      p2MH            0.520
##      p3MH            0.613
##      p1PhH           0.443
##      p2PhH           0.526
##      p3PhH           0.511
##      Spirit1          0.527
##      Spirit2          0.587
##      SocS1            0.494
##      SocS2            0.429
##      Eng1             0.369
##      Eng2             0.485
##      dEng1            0.433
##      dEng2            0.401
##      MH               0.584
##      PhH              0.300
##      Spr              0.679
##      SSp              0.481
##      Eng              0.491
##      dEng             0.500
##
## Defined Parameters:
##                Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##      indirect.EngMH   0.139  0.089  1.564  0.118  0.161  0.161
##      indirect.SprMH   0.086  0.122  0.703  0.482  0.099  0.099
##      indirect.SSpMH  -0.005  0.076 -0.062  0.950 -0.005 -0.005
##      indirct.dEngMH  -0.328  0.123 -2.680  0.007 -0.379 -0.379
##      indirct.EngPhH   0.130  0.099  1.305  0.192  0.156  0.156
##      indirct.SprPhH   0.119  0.141  0.842  0.400  0.143  0.143
##      indirct.SSpPhH   0.007  0.087  0.076  0.939  0.008  0.008
##      indrct.dEngPhH  -0.241  0.117 -2.059  0.039 -0.289 -0.289
##      direct.MH        -0.445  0.205 -2.178  0.029 -0.514 -0.514
##      direct.PhH       -0.369  0.224 -1.650  0.099 -0.444 -0.444
##      total.MH         -0.554  0.062 -8.932  0.000 -0.639 -0.639
##      total.PhH        -0.394  0.083 -4.768  0.000 -0.445 -0.445

# struct_pEsts #although creating the object is useful to export as a
# .csv I didn't ask it to print into the book

```

Below is script that will export the global fit indices (via *tidySEM::table_fit*) and the parameter estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```

# global fit indices
struct_fitstats <- tidySEM::table_fit(struct_fit1)

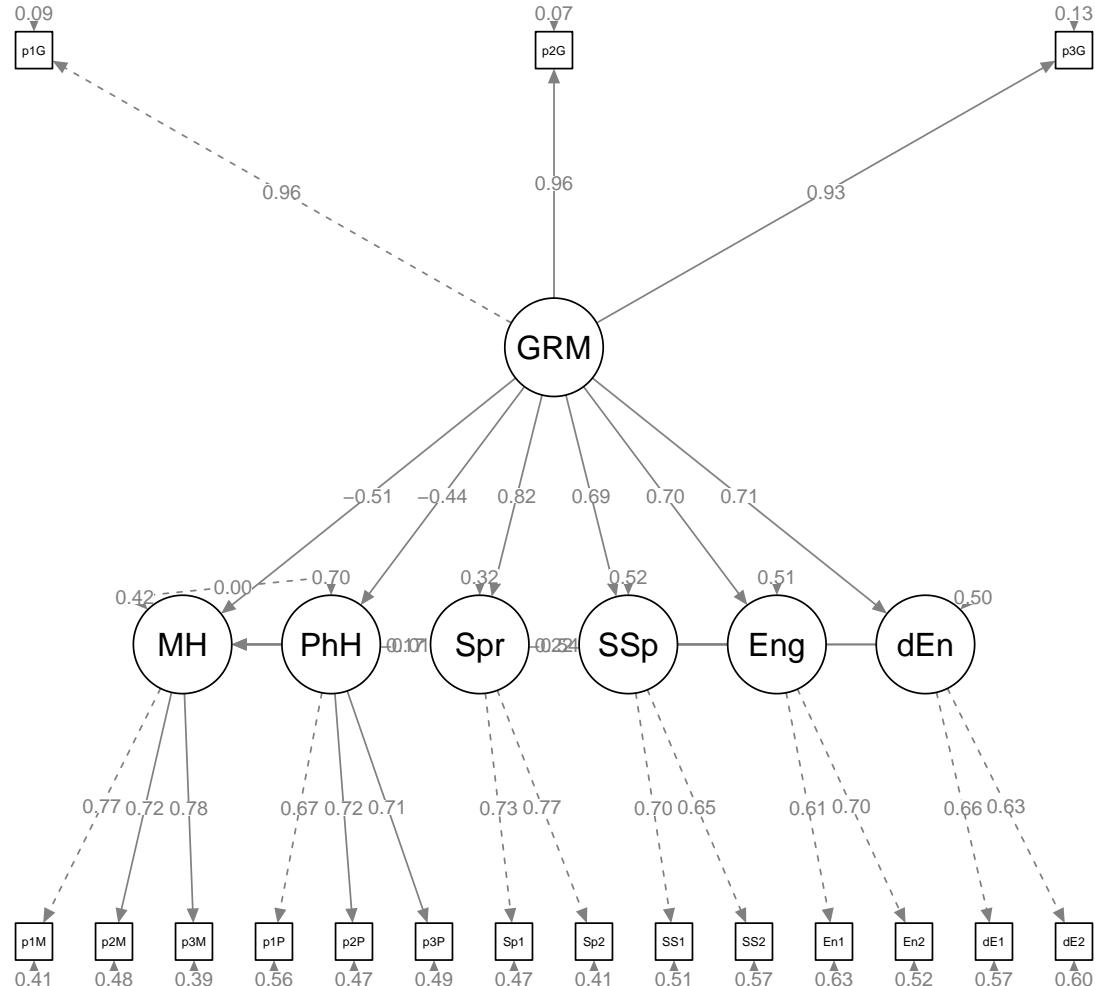
```

```
write.csv(struct_fitstats, file = "struct_fitstats.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(struct_pEsts1, file = "struct_pEsts1.csv")
```

11.8.1.2 Interpreting the Output

Plotting the results can be useful in checking our work and, if correct, understanding the relations between the variables. The `semPlot::semPaths` function will produce an initial guess at what we might like that can be further tweaked.

```
plot_struct1 <- semPlot::semPaths(struct_fit1, what = "col", whatLabels = "stand",
    sizeMan = 3, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
    mar = c(2, 2, 2, 2), structural = FALSE, curve = FALSE, intercepts = FALSE)
```



Although the code below may look daunting, I find it to be a fairly straightforward way to obtain figures that convey the model we are testing. We first start by identifying the desired location of

our latent variables, using numbers to represent their position by “(column, row)”. In the table below, I have mapped my variables.

Grid for Plotting semplot::sempath		
(1,1)	(1,2)	Eng
(2,1)	(2,2)	Spr
(3,1)	(3,2)	MH
(4,1) GRM	(4,2)	(4,3)
(5,1)	(5,2)	(5,3) PhH
(6,1)	(6,2)	Ssp
(7,1)	(7,2)	dEng
		(7,3)

We place these values along with the names of our latent variables in to the *semptools::layout_matrix* function.

```
# IMPORTANT: Must use the node names (take directly from the SemPlot)
# assigned by SemPlot You can change them as the last thing
m1_msmt <- semptools::layout_matrix(GRM = c(4, 1), Eng = c(1, 2), Spr = c(2,
2), SSp = c(6, 2), dEn = c(7, 2), MH = c(3, 3), PhH = c(5, 3))
```

Next we provide instruction on the direction (up, down, left, right) we want the indicator/observed variables to face. We identify the direction by the location of each of our latent variables. For example, in the code below we want the indicators for the REM variable (2,1) to face left.

```
# tell where you want the indicators to face
m1_point_to <- semptools::layout_matrix(left = c(4, 1), up = c(1, 2), down = c(2,
2), up = c(6, 2), down = c(7, 2), right = c(3, 3), right = c(5, 3))
```

The next two sets of code work together to specify the order of the observed variables for each factor. in the top set of code the variable names indicate the order in which they will appear (i.e., p1R, p2R, p3R). In the second set of code, the listing the variable name three times (i.e., REM, REM, REM) serves as a placeholder for each of the indicators.

It is critical to note that we need to use the abbreviated variable names assigned by *semTools::semPaths* and not necessarily the names that are in the dataframe.

```
# the next two codes -- indicator_order and indicator_factor are
# paired together, they specify the order of observed variables for
# each factor
m1_indicator_order <- c("p1G", "p2G", "p3G", "p1M", "p2M", "p3M", "p1P",
"p2P", "p3P", "Sp1", "Sp2", "SS1", "SS2", "En1", "En2", "dE1", "dE2")

m1_indicator_factor <- c("GRM", "GRM", "GRM", "MH", "MH", "MH", "PhH",
"PhH", "PhH", "Spr", "Spr", "SSp", "SSp", "Eng", "Eng", "dEn", "dEn")
```

The next two sets of codes provide some guidance about how far away the indicator (square/rectangular) variables should be away from the latent (oval/circular) variables. Subsequently, the next set of values indicate how far away each of the indicator (square/rectangular) variables should be spread apart.

```
#next set of code pushes the indicator variables away from the factor
m1_indicator_push <- c(GRM = .5,
                      MH = 1,
                      PhH = 1,
                      Spr = 2,
                      SSp = 1.5,
                      Eng = 1.5,
                      dEn = 2)

m1_indicator_spread <- c(GRM = .5, #spreading the boxes away from each other
                          MH = 2.5,
                          PhH = 2.5,
                          Spr = 1,
                          SSp = 1,
                          Eng = 1,
                          dEn = 1)
```

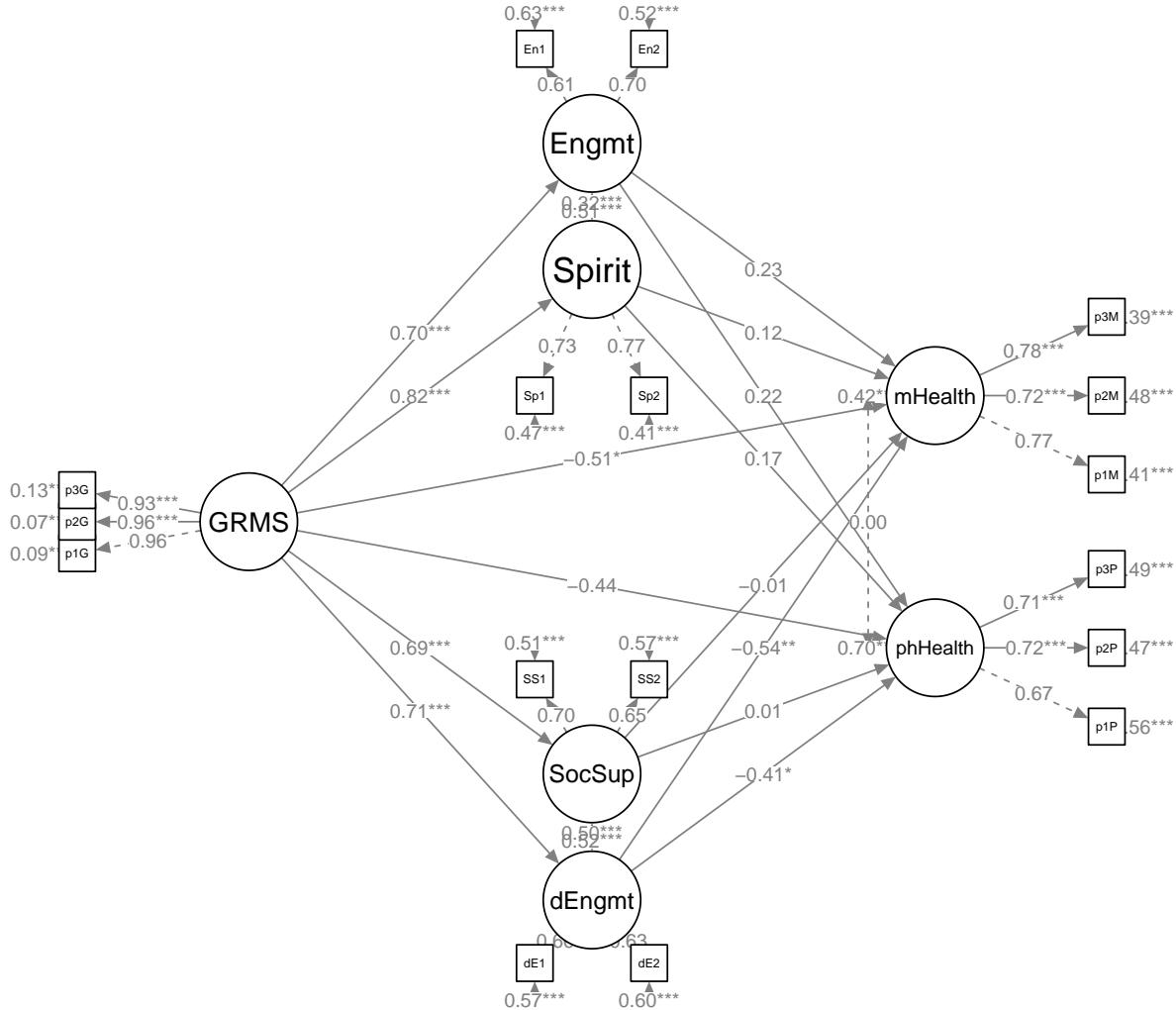
Finally, we can feed all of the objects that whole these instructions into the *semtools::sem_set_layout* function. If desired, we can use the *semtools::change_node_label* function to rename the latent variables. Again, make sure to use the variable names that *semPlot::semPaths* has assigned.

```
plot1 <- semtools::set_sem_layout(plot_struct1, indicator_order = m1_indicator_order,
                                   indicator_factor = m1_indicator_factor, factor_layout = m1_msmt, factor_point_to = m1_point_to,
                                   indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)

# changing node labels
plot1 <- semtools::change_node_label(plot1, c(GRM = "GRMS", MH = "mHealth",
                                               PhH = "phHealth", Eng = "Engmt", dEn = "dEngmt", Spr = "Spirit", SSp = "SocSup"),
                                         label.cex = 1.1)

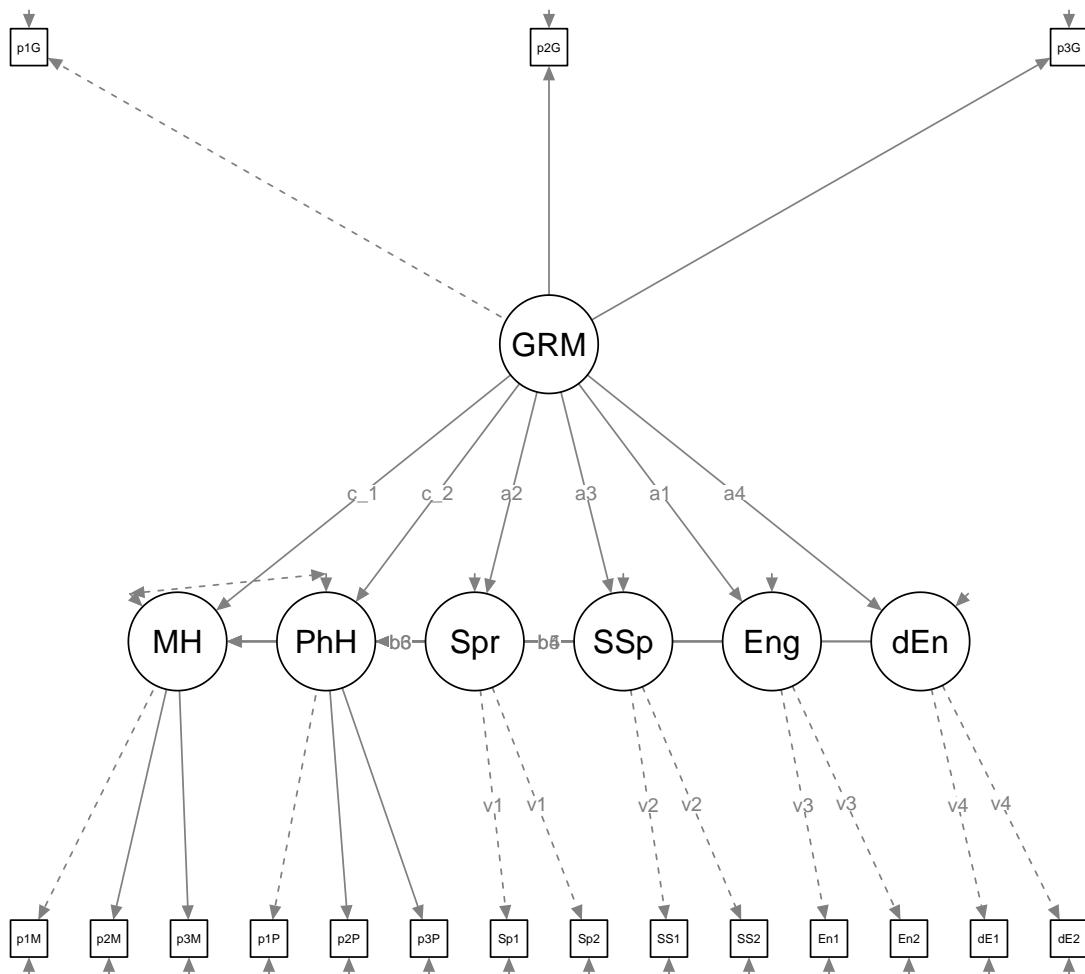
# adding stars to indicate significant paths
plot1 <- semtools::mark_sig(plot1, struct_fit1)

plot(plot1)
```



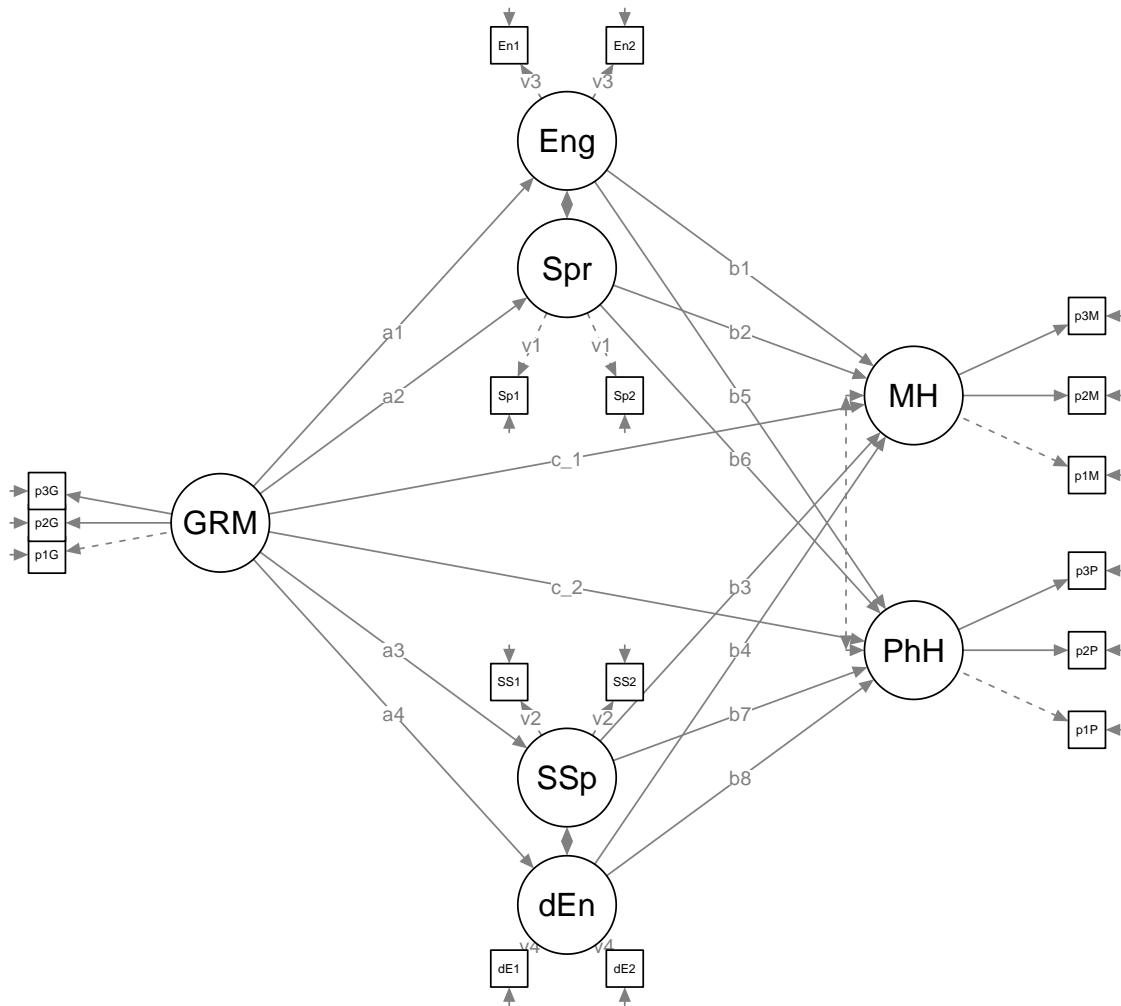
It can be useful to have a representation of the model without the results. This set of code produces those results. It does so by including only the name of the fitted object into the `semPlot::semPaths` function. Then it uses all the objects we just created as instructions for the figure's appearance.

```
# Code to plot the theoretical model (in case you don't want to print
# the results on the graph):
plot1_theoretical <- semPlot::semPaths(struct_fit1, sizeMan = 3, node.width = 1,
  edge.label.cex = 0.75, style = "lisrel", mar = c(2, 2, 2, 2), structural = FALSE,
  curve = FALSE, intercepts = FALSE)
```



```

plot1_theoretical <- semtools::set_sem_layout(plot1_theoretical, indicator_order = m1_indicator_order,
                                              indicator_factor = m1_indicator_factor, factor_layout = m1_msmt,
                                              factor_point_to = m1_point_to,
                                              indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)
plot(plot1_theoretical)
    
```



The statistical string for the global fit indices can be represented this way: $\chi^2(109) = 120.324, p = 0.216, CFI = 0.994, RMSEA = 0.021, 90CI[0.000, 0.041], SRMR = 0.044$.

Tabling the regression weights will assist us in understanding the relations between the variables. To be consistent with my figure, in this table I have included the standardized results (i.e., β).

Table 2

Model Coefficients Assessing the Effect of Gendered Racial Microaggressions on Mental and Physical Health through Coping Strategies

Predictor	B	SE _B	p	β	R ²
Engagement (M1)					.49
Gendered Racial Microaggressions (a ₁)	0.392	0.041	< 0.001	0.701	
Spirituality (M2)					.68

Predictor	<i>B</i>	<i>SE_B</i>	<i>p</i>	β	<i>R</i> ²
Gendered Racial Microaggressions (a_2)	0.554	0.040	< 0.001	0.824	
Social Support (M3)					.48
Gendered Racial Microaggressions (a_3)	0.406	0.042	< 0.001	0.694	
Disengagement (M4)					.50
Gendered Racial Microaggressions (a_4)	0.386	0.041	< 0.001	0.707	
Mental Health (DV1)					.58
Engagement (b_1)	0.355	0.223	0.111	0.229	
Spirituality (b_1)	0.155	0.219	0.481	0.120	
Social Support (b_3)	-0.012	0.187	0.950	-0.008	
Disengagement (b_4)	-0.850	0.303	0.005	-0.536	
Gendered Racial Microaggressions (c'_1)	-0.445	0.205	0.029	-0.514	
Physical Health (DV2)					.30
Engagement (b_5)	0.330	0.250	0.186	0.222	
Spirituality (b_6)	0.215	0.254	0.398	0.174	
Social Support (b_7)	0.016	0.215	0.939	0.012	
Disengagement (b_8)	-0.624	0.295	0.034	-0.409	
Gendered Racial Microaggressions (c'_2)	-0.369	0.224	0.099	-0.444	

Effects	<i>B</i>	<i>SE_B</i>	<i>p</i>	β	95% CI
Indirect 3($a_1 * b_1$)	0.139	0.089	0.118	0.161	-0.035, 0.314
Indirect 1($a_2 * b_2$)	0.086	0.122	0.482	0.099	-0.153, 0.325
Indirect 2($a_3 * b_3$)	-0.005	0.076	0.950	-0.005	-0.154, 0.144
Indirect 4($a_4 * b_4$)	-0.328	0.123	0.007	-0.379	-0.569, -0.088
Direct 1 (c'_1)	-0.445	0.205	0.029	-0.514	-0.846, -0.045
Total 1 (c_1)	-0.554	0.062	< 0.001	-0.639	-0.675, -0.432
Indirect 7($a_1 * b_5$)	0.130	0.099	0.192	0.156	-0.065, 0.324
Indirect 5($a_2 * b_6$)	0.119	0.141	0.400	0.143	-0.158, 0.396
Indirect 6($a_3 * b_7$)	0.007	0.087	0.939	0.008	-0.165, 0.178
Indirect 8($a_4 * b_8$)	-0.241	0.117	0.039	-0.289	-0.470, -0.012
Direct 2 (c'_2)	-0.369	0.224	0.099	-0.444	-0.808, 0.069
Total 2 (c_2)	-0.394	0.083	0.000	-0.445	-0.556, -0.232

Note. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

As we can see, our model accounts for 48-68% of the variance in the mediators and 58% and 30% of the variance in the mental and physical health variables, respectively. Only one indirect effect, gendered racial microaggressions to mental health, via disengagement coping. Inspection of the

specific paths may provide some insight into this. Specifically, all of the a paths are statistically significant (i.e., gendered racial microaggressions predicting each of the four coping approaches) but only two of the b paths (i.e., disengagement coping to mental and physical health, respectively).

Here's how I might write up this section of the analysis:

Our structural model was a parallel mediation, predicting mental and physical health directly from gendered racial microaggressions and indirectly through four coping strategies (i.e., spirituality, social support, engagement, disengagement). Results of the global fit indices suggested adequate fit $\chi^2(109) = 120.324, p = 0.216, CFI = 0.994, RMSEA = 0.021, 90CI[0.000, 0.041], SRMR = 0.044$. As shown in Table 2 only two of the eight indirect effects was statistically significant. For both, disengagement coping mediated the effect of gendered racial microaggressions on mental health ($B = -0.328, p = 0.007$) and physical health ($B = -0.241, p = 0.039$). The model accounted for 48-68% of the variance in the mediators and 58% and 30% of the variance in mental and physical health, respectively.

With such strong overall fit, if I were the researcher, I would be inclined to stop here. Given that this is a lecture on model building and trimming, let's proceed to see if respecification is indicated.

11.9 Model Building

Recall that in model building we start with a more parsimonious model and consider “adding paths.” This requires that our original model be *overidentified* with positive degrees of freedom. Earlier we determined that we had 8 degrees of freedom.

Modification indices are a tool that can help us determine where directional paths or bidirectional covariances might improve model fit. A modification index is produced for each element of the model that is constrained to be zero (i.e., does not have a path or covariance). These modification indices are presented in the metric of the chi-square test and will tell us “by how much the chi-square value will decrease” if the two elements are freed to relate (i.e., by a path or covariance). In a one-degree chi-square test, statistically significant change occurs when the modification index is greater than 3.841. Thus, when researchers use the *lavaan::modindices()* function, they typically set a minimum value of 4.0 (rounding up from 3.841).

When inspecting the results, we will look for the parameters with the highest values (that are clearly above 4.0) that would be theoretically defensible. Be aware that this procedure may “suggest” many nonsensible relations.

```
lavaan::modindices(struct_fit1, sort = TRUE, minimum.value = 4)
```

##	lhs	op	rhs	mi	epc	sepc.lv	sepc.all	sepc.nox
## 125	PhH	=~	p2MH	12.407	0.432	0.261	0.246	0.246
## 264	p2MH	~~	dEng2	8.859	-0.087	-0.087	-0.243	-0.243
## 362	SSp	~	Spr	8.339	0.468	0.537	0.537	0.537
## 345	Spr	~~	SSp	8.339	0.036	0.422	0.422	0.422
## 357	Spr	~	SSp	8.339	0.381	0.332	0.332	0.332

```

## 159    SSp =~ Spirit1  8.328  0.304   0.129   0.192   0.192
## 184    dEng =~ p2MH   7.683 -0.944  -0.374  -0.352  -0.352
## 303  Spirit1 ~~ Spirit2  7.539  2.239   2.239  11.841  11.841
## 325    Eng1 ~~ Eng2   7.538  0.635   0.635   2.853   2.853
## 372    PhH ~ MH      7.538  0.453   0.471   0.471   0.471
## 330    dEng1 ~~ dEng2  7.538  0.140   0.140   0.639   0.639
## 32     MH ~~ PhH     7.538  0.074   0.363   0.363   0.363
## 371     MH ~ PhH     7.538  0.292   0.280   0.280   0.280
## 133     PhH =~ dEng1  7.349  0.210   0.127   0.210   0.210
## 196    p1GRMS ~~ p3GRMS 7.107  0.022   0.022   0.384   0.384
## 245    p1MH ~~ Spirit1 6.205 -0.052  -0.052  -0.215  -0.215
## 119     MH =~ dEng1  6.170  0.186   0.117   0.194   0.194
## 241    p1MH ~~ p3MH   5.439  0.110   0.110   0.337   0.337
## 137     Spr =~ p3GRMS 5.237 -0.238  -0.116  -0.165  -0.165
## 185    dEng =~ p3MH   5.180  0.758   0.300   0.301   0.301
## 232    p3GRMS ~~ Spirit1 5.110 -0.022  -0.022  -0.188  -0.188
## 221    p2GRMS ~~ SocS2  5.098  0.022   0.022   0.213   0.213
## 351    Eng ~ Spr     4.830  0.351   0.421   0.421   0.421
## 346    Spr ~~ Eng    4.830  0.027   0.334   0.334   0.334
## 356    Spr ~ Eng    4.830  0.318   0.265   0.265   0.265
## 304  Spirit1 ~~ SocS1 4.473  0.036   0.036   0.183   0.183
## 16     dEng =~ dEng1  4.197 -0.290  -0.115  -0.191  -0.191
## 17     dEng =~ dEng2  4.197  0.290   0.115   0.183   0.183

```

Because we used the command, “sort=TRUE” the modification indices (in the “mi” column) are presented from highest to lowest. Having familiarity with *lavaan* syntax can be useful in deciphering what is being suggested.

The first modification index (12.40) relates to the factor loadings in the measurement model. We know this because of the “ $\sim\!$ ” operator. It suggests that we free *p2MH* (the second parcel for mental health) to *PhH* (the physical health latent variable). I would consider this “nonsensible.” While this might improve model fit, this suggestion relates to our measurement model – and our measurement model had terrific fit with no need for further improvement.

The second modification index (8.859) is accompanied by the “ $\sim\!$ ” operator. This means it is suggesting that we free the second parcel for mental health to covary (correlate) with item 2 for disengagement coping. This is not a sensible suggestion. In fact the *only* sensible suggested covariation is to free the dependent variables (*PhH* and *MH*) to covary (correlate).

The single tilda (~) provides suggestions for regression paths. Note that the third and fifth recommendations are to either predict spiritual coping from social support coping or the vice versa. Further, the fourth recommendation is to allow those two variables to covary. Either of these three options would reduce our chi-square by 8.339. Keep in mind that the *lavaan::modindices* package will produce all possible combinations of ways to improve model fit, even if they are redundant.

I am intrigued by this suggested relation between spiritual coping and social support. If we added this path, we could test for the significance of an serially mediated effect from gendered racial microaggressions, to spiritual support, through social support, to mental and physical health, respectively.

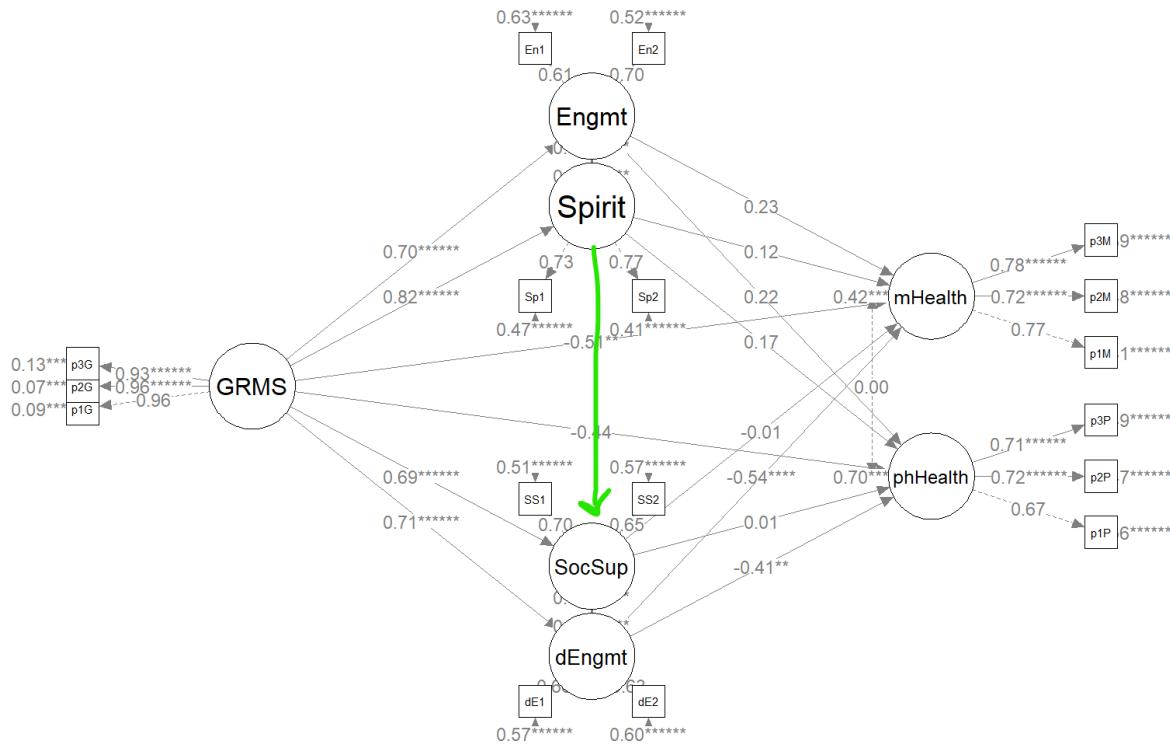


Figure 11.3: An image illustrating the most sensible path suggested by the modification indices

```

struct_mod2 <- "
##measurement model
GRMS =~ p1GRMS + p2GRMS + p3GRMS
MH =~ p1MH + p2MH + p3MH
PhH =~ p1PhH + p2PhH + p3PhH
Spr =~ v1*Spirit1 + v1*Spirit2
SSp =~ v2*SocS1 + v2*SocS2
Eng =~ v3*Eng1 + v3*Eng2
dEng =~ v4*dEng1 + v4*dEng2

#structural model with labels for calculation of the indirect effect

Eng ~ a1*GRMS
Spr ~ a2*GRMS
SSp ~ a3*GRMS
dEng ~ a4*GRMS

MH ~ c_p1*GRMS + b1*Eng + b2*Spr + b3*SSp + b4*dEng
PhH ~ c_p2*GRMS + b5*Eng + b6*Spr + b7*SSp + b8*dEng

SSp ~ d1*Spr

```

```

#cov
MH ~~ 0*PhH

#calculations
indirect.EngMH := a1*b1
indirect.SprMH := a2*b2
indirect.SSpMH := a3*b3
indirect.dEngMH := a4*b4

indirect.EngPhH := a1*b5
indirect.SprPhH := a2*b6
indirect.SSpPhH := a3*b7
indirect.dEngPhH := a4*b8

serial.MH := a2*d1*b3
serial.PH := a2*d1*b7

direct.MH := c_p1
direct.PhH := c_p2
total.MH := c_p1 + (a1*b1) + (a2*b2) + (a3*b3) + (a4*b4) + (a2*d1*b3)
total.PhH := c_p2 + (a1*b5) + (a1*b6) + (a1*b7) + (a1*b8) + (a2*d1*b7)

"
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
struct_fit2 <- lavaan::sem(struct_mod2, data = dfLewis, missing = "fiml")
struct_summary2 <- lavaan::summary(struct_fit2, fit.measures = TRUE, standardized = TRUE,
                                    rsq = TRUE)
struct_pEsts2 <- lavaan::parameterEstimates(struct_fit2, boot.ci.type = "bca.simple",
                                             standardized = TRUE)
struct_summary2

## lavaan 0.6.16 ended normally after 121 iterations
##
##      Estimator                               ML
## Optimization method                         NLMINB
## Number of model parameters                  62
## 
##      Number of observations                   231
##      Number of missing patterns                1
## 
##      Model Test User Model:
## 
##      Test statistic                          111.657
##      Degrees of freedom                      108
##      P-value (Chi-square)                    0.385
## 
```

```

## Model Test Baseline Model:
##
##   Test statistic                      2104.157
##   Degrees of freedom                  136
##   P-value                            0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)        0.998
##   Tucker-Lewis Index (TLI)          0.998
##
##   Robust Comparative Fit Index (CFI) 0.998
##   Robust Tucker-Lewis Index (TLI)    0.998
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)      -3385.514
##   Loglikelihood unrestricted model (H1) -3329.686
##
##   Akaike (AIC)                      6895.029
##   Bayesian (BIC)                     7108.458
##   Sample-size adjusted Bayesian (SABIC) 6911.954
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                           0.012
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.036
##   P-value H_0: RMSEA <= 0.050       0.999
##   P-value H_0: RMSEA >= 0.080       0.000
##
##   Robust RMSEA                     0.012
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.036
##   P-value H_0: Robust RMSEA <= 0.050 0.999
##   P-value H_0: Robust RMSEA >= 0.080 0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                            0.042
##
## Parameter Estimates:
##
##   Standard errors                  Standard
##   Information                      Observed
##   Observed information based on    Hessian
##
## Latent Variables:

```

##		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
##	GRMS =~						
##	p1GRMS	1.000				0.725	0.955
##	p2GRMS	1.014	0.030	34.076	0.000	0.736	0.962
##	p3GRMS	0.903	0.030	29.918	0.000	0.655	0.933
##	MH =~						
##	p1MH	1.000				0.628	0.769
##	p2MH	1.217	0.126	9.664	0.000	0.765	0.721
##	p3MH	1.245	0.113	11.045	0.000	0.783	0.783
##	PhH =~						
##	p1PhH	1.000				0.604	0.666
##	p2PhH	1.006	0.128	7.878	0.000	0.608	0.725
##	p3PhH	1.036	0.135	7.662	0.000	0.626	0.715
##	Spr =~						
##	Spirit1 (v1)	1.000				0.487	0.731
##	Spirit2 (v1)	1.000				0.487	0.759
##	SSp =~						
##	SocS1 (v2)	1.000				0.425	0.704
##	SocS2 (v2)	1.000				0.425	0.654
##	Eng =~						
##	Eng1 (v3)	1.000				0.406	0.607
##	Eng2 (v3)	1.000				0.406	0.696
##	dEng =~						
##	dEng1 (v4)	1.000				0.396	0.658
##	dEng2 (v4)	1.000				0.396	0.633
##							
##	Regressions:						
##		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
##	Eng ~						
##	GRMS (a1)	0.392	0.041	9.498	0.000	0.700	0.700
##	Spr ~						
##	GRMS (a2)	0.552	0.040	13.784	0.000	0.822	0.822
##	SSp ~						
##	GRMS (a3)	0.138	0.109	1.266	0.205	0.236	0.236
##	dEng ~						
##	GRMS (a4)	0.386	0.041	9.400	0.000	0.707	0.707
##	MH ~						
##	GRMS (c_p1)	-0.431	0.193	-2.236	0.025	-0.498	-0.498
##	Eng (b1)	0.351	0.222	1.579	0.114	0.227	0.227
##	Spr (b2)	0.185	0.277	0.668	0.504	0.144	0.144
##	SSp (b3)	-0.082	0.237	-0.347	0.729	-0.055	-0.055
##	dEng (b4)	-0.852	0.303	-2.815	0.005	-0.537	-0.537
##	PhH ~						
##	GRMS (c_p2)	-0.354	0.210	-1.690	0.091	-0.425	-0.425
##	Eng (b5)	0.325	0.249	1.303	0.192	0.219	0.219
##	Spr (b6)	0.257	0.322	0.797	0.425	0.207	0.207
##	SSp (b7)	-0.071	0.273	-0.259	0.795	-0.050	-0.050
##	dEng (b8)	-0.626	0.294	-2.129	0.033	-0.411	-0.411

```

##   SSp ~
##   Spr      (d1)    0.477    0.180    2.646    0.008    0.548    0.548
##
## Covariances:
##                               Estimate Std.Err  z-value P(>|z|) Std.lv Std.all
##   .MH ~~
##   .PhH      0.000                0.000    0.000
##
## Intercepts:
##                               Estimate Std.Err  z-value P(>|z|) Std.lv Std.all
##   .p1GRMS    2.592    0.050   51.897   0.000   2.592   3.415
##   .p2GRMS    2.545    0.050   50.567   0.000   2.545   3.327
##   .p3GRMS    2.579    0.046   55.829   0.000   2.579   3.673
##   .p1MH      3.582    0.054   66.611   0.000   3.582   4.383
##   .p2MH      2.866    0.070   41.023   0.000   2.866   2.699
##   .p3MH      3.035    0.066   46.140   0.000   3.035   3.036
##   .p1PhH     3.128    0.060   52.384   0.000   3.128   3.447
##   .p2PhH     2.652    0.055   48.056   0.000   2.652   3.162
##   .p3PhH     3.067    0.058   53.258   0.000   3.067   3.504
##   .Spirit1   2.511    0.044   57.292   0.000   2.511   3.770
##   .Spirit2   2.437    0.042   57.677   0.000   2.437   3.795
##   .SocS1     2.550    0.040   64.289   0.000   2.550   4.230
##   .SocS2     2.758    0.043   64.537   0.000   2.758   4.246
##   .Eng1      2.437    0.044   55.375   0.000   2.437   3.643
##   .Eng2      2.515    0.038   65.530   0.000   2.515   4.312
##   .dEng1    2.502    0.040   63.177   0.000   2.502   4.157
##   .dEng2    2.455    0.041   59.612   0.000   2.455   3.922
##   GRMS      0.000                0.000   0.000   0.000
##   .MH       0.000                0.000   0.000   0.000
##   .PhH      0.000                0.000   0.000   0.000
##   .Spr      0.000                0.000   0.000   0.000
##   .SSp      0.000                0.000   0.000   0.000
##   .Eng      0.000                0.000   0.000   0.000
##   .dEng    0.000                0.000   0.000   0.000
##
## Variances:
##                               Estimate Std.Err  z-value P(>|z|) Std.lv Std.all
##   .p1GRMS    0.050    0.007   6.695   0.000   0.050   0.087
##   .p2GRMS    0.044    0.007   6.058   0.000   0.044   0.075
##   .p3GRMS    0.064    0.008   8.355   0.000   0.064   0.129
##   .p1MH      0.273    0.037   7.377   0.000   0.273   0.409
##   .p2MH      0.542    0.067   8.062   0.000   0.542   0.481
##   .p3MH      0.387    0.055   7.083   0.000   0.387   0.387
##   .p1PhH     0.459    0.058   7.914   0.000   0.459   0.557
##   .p2PhH     0.334    0.049   6.824   0.000   0.334   0.475
##   .p3PhH     0.375    0.054   6.984   0.000   0.375   0.489
##   .Spirit1   0.206    0.026   8.020   0.000   0.206   0.465
##   .Spirit2   0.175    0.024   7.399   0.000   0.175   0.424

```

```

##   .SocS1          0.183  0.025  7.198  0.000  0.183  0.504
##   .SocS2          0.241  0.029  8.222  0.000  0.241  0.573
##   .Eng1           0.282  0.033  8.560  0.000  0.282  0.631
##   .Eng2           0.175  0.026  6.694  0.000  0.175  0.515
##   .dEng1          0.205  0.029  7.123  0.000  0.205  0.567
##   .dEng2          0.235  0.030  7.705  0.000  0.235  0.599
##   GRMS            0.526  0.054  9.790  0.000  1.000  1.000
##   .MH              0.164  0.042  3.926  0.000  0.416  0.416
##   .PhH             0.255  0.058  4.392  0.000  0.698  0.698
##   .Spr             0.077  0.019  4.044  0.000  0.324  0.324
##   .SSp             0.078  0.022  3.576  0.000  0.431  0.431
##   .Eng              0.084  0.022  3.819  0.000  0.510  0.510
##   .dEng            0.078  0.024  3.333  0.001  0.500  0.500
##
## R-Square:
##                               Estimate
##   p1GRMS            0.913
##   p2GRMS            0.925
##   p3GRMS            0.871
##   p1MH              0.591
##   p2MH              0.519
##   p3MH              0.613
##   p1PhH             0.443
##   p2PhH             0.525
##   p3PhH             0.511
##   Spirit1            0.535
##   Spirit2            0.576
##   SocS1              0.496
##   SocS2              0.427
##   Eng1               0.369
##   Eng2               0.485
##   dEng1              0.433
##   dEng2              0.401
##   MH                 0.584
##   PhH                0.302
##   Spr                0.676
##   SSp                0.569
##   Eng                0.490
##   dEng               0.500
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   indirect.EngMH      0.138  0.089  1.549  0.121  0.159  0.159
##   indirect.SprMH      0.102  0.154  0.667  0.505  0.118  0.118
##   indirect.SSpMH     -0.011  0.032 -0.350  0.726 -0.013 -0.013
##   indirct.dEngMH     -0.329  0.122 -2.691  0.007 -0.380 -0.380
##   indirct.EngPhH      0.127  0.099  1.287  0.198  0.153  0.153
##   indirct.SprPhH      0.142  0.178  0.794  0.427  0.170  0.170

```

```

##    indirct.SSpPhH   -0.010   0.037  -0.264   0.792  -0.012  -0.012
##    indrct.dEngPhH  -0.242   0.117  -2.072   0.038  -0.290  -0.290
##    serial.MH       -0.022   0.064  -0.336   0.737  -0.025  -0.025
##    serial.PH       -0.019   0.074  -0.253   0.800  -0.022  -0.022
##    direct.MH       -0.431   0.193  -2.236   0.025  -0.498  -0.498
##    direct.PhH      -0.354   0.210  -1.690   0.091  -0.425  -0.425
##    total.MH        -0.554   0.062  -8.934   0.000  -0.639  -0.639
##    total.PhH       -0.418   0.134  -3.120   0.002  -0.472  -0.472

```

```

# struct_pEsts2 #although creating the object is useful to export as
# a .csv I didn't ask it to print into the book

```

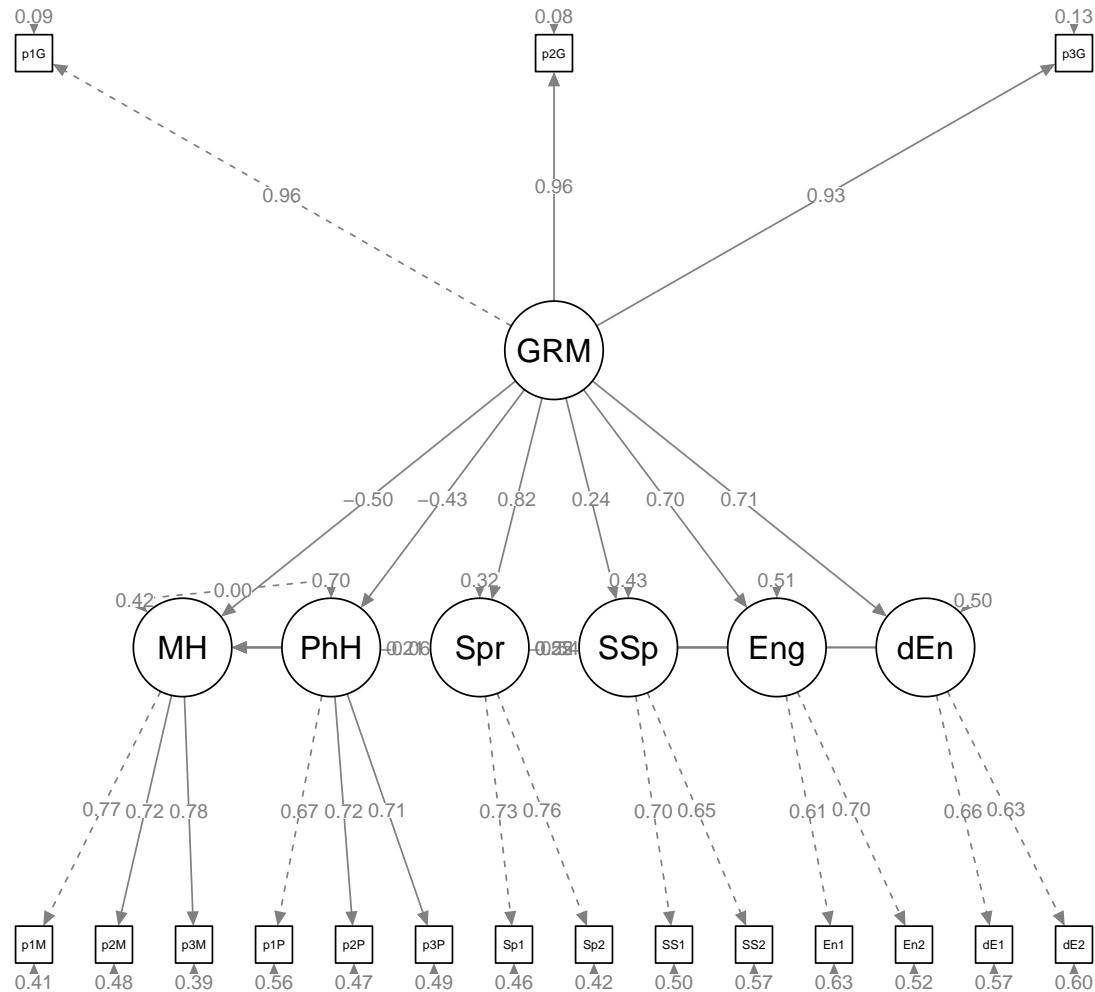
Creating a figure will help us with the conceptual understanding of what we've just done (and will also help us check our work).

Let's first create a fresh plot from se

```

plot_struct2 <- semPlot::semPaths(struct_fit2, what = "col", whatLabels = "stand",
  sizeMan = 3, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(2, 2, 2, 2), structural = FALSE, curve = FALSE, intercepts = FALSE)

```



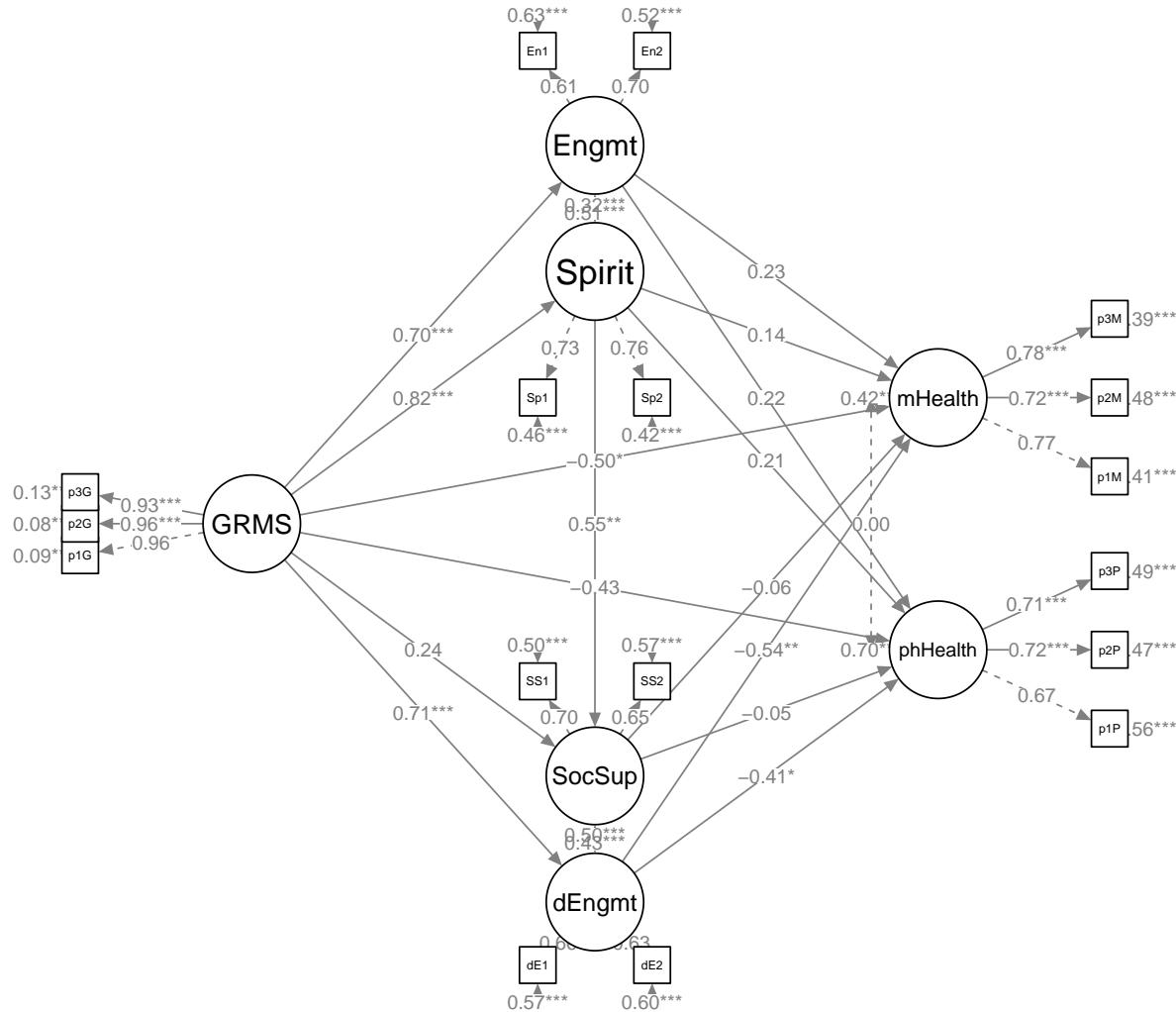
Because we haven't added or deleted variables (only 1 path) and we want them to stay in the same location with the same orientation of paths and covariances, we should be able to recycle some of the diagram code we created earlier

```
plot2 <- semtools::set_sem_layout(plot_struct2, indicator_order = m1_indicator_order,
  indicator_factor = m1_indicator_factor, factor_layout = m1_msmt, factor_point_to = m1_point
  indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)

# changing node labels
plot2 <- semtools::change_node_label(plot2, c(GRM = "GRMS", MH = "mHealth",
  PhH = "phHealth", Eng = "Engmt", dEn = "dEngmt", Spr = "Spirit", SSp = "SocSup"),
  label.cex = 1.1)

# adding stars to indicate significant paths
plot2 <- semtools::mark_sig(plot2, struct_fit2)

plot(plot2)
```



As predicted by the modification indices (and our knowledge that a model with fewer degrees of freedom will generally have improved fit), the global fit indices have improved: $\chi^2(108) = 111.657, p = 0.385, CFI = 0.998, RMSEA = 0.012, 90CI[0.000, 0.036], SRMR = 0.042$

We can formally compare these with the `lavaan::lavTestLRT()` function.

```
lavaan::lavTestLRT(struct_fit1, struct_fit2)
```

```
##
## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq Chisq diff     RMSEA Df diff Pr(>Chisq)
## struct_fit2 108 6895.0 7108.5 111.66
## struct_fit1 109 6901.7 7111.7 120.32      8.667 0.18218       1 0.00324 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The chi-square difference test can be used to compare nested models. Models are nested when

there is only the addition/deletion of paths or covariances. If the sample or variables change, they cannot be used. Our chi-square difference test was, $\Delta\chi^2(1) = 8.667, p = 0.032$. This suggests that there were statistically significant difference in our nested models, favoring the newer model.

Something about this should look suspiciously familiar. Do you notice the chi-square value itself? Earlier our modification indices told us that if we freed Spr and SSp to relate, the chi-square value would drop by 8.339. While not exact, the prediction was fairly spot on!

Additionally, the AIC and BIC can be used to compare non-nested models. Preferred models have lower values. In our case, the newer model appears to be preferred.

Is freeing this parameter an appropriate addition to our model, or is it *overparameterization*. We should also inspect the d_1 path and the serially mediated indirect effects. It does appear that regression weight from spirituality to social support (i.e., d_1) was statistically significant ($B = 0.477, p = 0.008$) but the serially mediated indirect effects was not. Specifically, when mental health was the outcome, $B = 0.022, p = 0.737$; when physical health was the outcome, $B = -0.019, p = 0.800$.

Adding this path feels, to me, like overparameterization. In the case of model building, I will stick with the originally hypothesized model.

11.10 Model Trimming

In model trimming, the researcher starts with a more general model or saturated (i.e., zero degrees of freedom in the structural model) and, on the basis of statistical criteria, trims non-significant (i.e., and low regression weight) paths from the model. Chou and Bentler [2002] termed this *backward searching*. We have a number of non-significant paths. Should we delete them? And if so, how many of them?

The inclusion of indirect effects in our model makes it a little tricky to know how and where we might delete paths. As we look at the regression weights of our original model, I note that

- all b paths (except those involving disengagement coping) are non-significant,
- all indirect effects (except those involving disengagement coping) are non-significant,
- the b paths involving engagement coping are closest to being statistically significant.

It is a best practice to trim paths one at a time and then re-check the results. Why? Because the constraint may free up a little power and statistical significance for a regression path may turn on elsewhere. Because this is a book chapter and constraining each non-significant effect one-at-a-time would (a) take forever and (b) make this chapter absolutely unwieldy, I will first delete the b paths associated with social support at the same time. So that I can keep track of what I'm doing in my code, I will hashtag them out. The elimination of these will also result in the elimination of the associated indirect effect; I will hashtag those out, also.

```
struct_mod3 <- "
##measurement model
GRMS =~ p1GRMS + p2GRMS + p3GRMS
MH =~ p1MH + p2MH + p3MH
PhH =~ p1PhH + p2PhH + p3PhH
```

```

Spr =~ v1*Spirit1 + v1*Spirit2
SSp =~ v2*SocS1 + v2*SocS2
Eng =~ v3*Eng1 + v3*Eng2
dEng =~ v4*dEng1 + v4*dEng2

#structural model with labels for calculation of the indirect effect

Eng ~ a1*GRMS
Spr ~ a2*GRMS
SSp ~ a3*GRMS
dEng ~ a4*GRMS

MH ~ c_p1*GRMS + b1*Eng + b2*Spr + b4*dEng #trimming + b3*SSp
PhH ~ c_p2*GRMS + b5*Eng + b6*Spr + b8*dEng #trimming + b7*SSp

#cov
MH ~~ 0*PhH #prevents MH and PhD from correlating

#calculations
indirect.EngMH := a1*b1
indirect.SprMH := a2*b2
#indirect.SSpMH := a3*b3 #trimmed after deleting b3, above
indirect.dEngMH := a4*b4

indirect.EngPhH := a1*b5
indirect.SprPhH := a2*b6
#indirect.SSpPhH := a3*b7 #trimmed after deleting b7, above
indirect.dEngPhH := a4*b8
direct.MH := c_p1
direct.PhH := c_p2
total.MH := c_p1 + (a1*b1) + (a2*b2) + (a4*b4) #trimmed + (a3*b3)
total.PhH := c_p2 + (a1*b5) + (a1*b6) + (a1*b8) #trimmed + (a1*b7)

"
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
struct_fit3 <- lavaan::sem(struct_mod3, data = dfLewis, missing = "fiml")
struct_summary3 <- lavaan::summary(struct_fit3, fit.measures = TRUE, standardized = TRUE,
                                    rsq = TRUE)
struct_pEsts3 <- lavaan::parameterEstimates(struct_fit3, boot.ci.type = "bca.simple",
                                             standardized = TRUE)
struct_summary3

## lavaan 0.6.16 ended normally after 113 iterations
##  

##   Estimator                               ML  

##   Optimization method                     NLMINB  

##   Number of model parameters               61

```

```

## 
##   Number of observations           231
##   Number of missing patterns        1
##
## Model Test User Model:
##
##   Test statistic                  120.319
##   Degrees of freedom                109
##   P-value (Chi-square)            0.216
##
## Model Test Baseline Model:
##
##   Test statistic                 2104.157
##   Degrees of freedom                  136
##   P-value                           0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)      0.994
##   Tucker-Lewis Index (TLI)          0.993
##
##   Robust Comparative Fit Index (CFI) 0.994
##   Robust Tucker-Lewis Index (TLI)    0.993
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)     -3389.845
##   Loglikelihood unrestricted model (H1) -3329.686
##
##   Akaike (AIC)                      6901.691
##   Bayesian (BIC)                     7111.678
##   Sample-size adjusted Bayesian (SABIC) 6918.343
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                            0.021
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.041
##   P-value H_0: RMSEA <= 0.050       0.995
##   P-value H_0: RMSEA >= 0.080       0.000
##
##   Robust RMSEA                      0.021
##   90 Percent confidence interval - lower 0.000
##   90 Percent confidence interval - upper 0.041
##   P-value H_0: Robust RMSEA <= 0.050 0.995
##   P-value H_0: Robust RMSEA >= 0.080 0.000
##
## Standardized Root Mean Square Residual:

```

```

##  

##      SRMR                               0.044  

##  

## Parameter Estimates:  

##  

##      Standard errors                   Standard  

##      Information                      Observed  

##      Observed information based on    Hessian  

##  

## Latent Variables:  

##  

##      Estimate   Std.Err  z-value  P(>|z|)  Std.lv  Std.all  

##  

##  GRMS =~  

##      p1GRMS      1.000  

##      p2GRMS      1.015  0.030  34.097  0.000  0.725  0.955  

##      p3GRMS      0.903  0.030  29.818  0.000  0.655  0.932  

##  MH =~  

##      p1MH        1.000  

##      p2MH        1.218  0.126  9.662  0.000  0.765  0.721  

##      p3MH        1.245  0.113  11.045 0.000  0.783  0.783  

##  PhH =~  

##      p1PhH       1.000  

##      p2PhH       1.007  0.128  7.878  0.000  0.608  0.725  

##      p3PhH       1.035  0.135  7.664  0.000  0.625  0.714  

##  Spr =~  

##      Spirit1     (v1)  1.000  

##      Spirit2     (v1)  1.000  

##  SSp =~  

##      SocS1       (v2)  1.000  

##      SocS2       (v2)  1.000  

##  Eng =~  

##      Eng1        (v3)  1.000  

##      Eng2        (v3)  1.000  

##  dEng =~  

##      dEng1       (v4)  1.000  

##      dEng2       (v4)  1.000  

##  

## Regressions:  

##  

##      Estimate   Std.Err  z-value  P(>|z|)  Std.lv  Std.all  

##  

##  Eng ~  

##      GRMS       (a1)  0.392  0.041  9.506  0.000  0.701  0.701  

##  Spr ~  

##      GRMS       (a2)  0.554  0.040  13.869 0.000  0.824  0.824  

##  SSp ~  

##      GRMS       (a3)  0.406  0.042  9.685  0.000  0.694  0.694  

##  dEng ~  

##      GRMS       (a4)  0.386  0.041  9.400  0.000  0.707  0.707  

##  MH ~  

##      GRMS       (c_p1) -0.450 0.193 -2.336 0.019 -0.520 -0.520

```

```

##   Eng      (b1)    0.355    0.222    1.594    0.111    0.229    0.229
##   Spr      (b2)    0.156    0.218    0.712    0.476    0.121    0.121
##   dEng     (b4)   -0.850    0.303   -2.807    0.005   -0.536   -0.536
## PhH ~
##   GRMS    (c_p2)   -0.363    0.209   -1.735    0.083   -0.435   -0.435
##   Eng      (b5)    0.330    0.249    1.325    0.185    0.222    0.222
##   Spr      (b6)    0.214    0.253    0.845    0.398    0.173    0.173
##   dEng     (b8)   -0.623    0.295   -2.115    0.034   -0.409   -0.409
##
## Covariances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .MH ~~
## .PhH        0.000
## .SSp       -0.002    0.017  -0.094    0.925  -0.013  -0.013
## .PhH ~~
## .SSp        0.002    0.019   0.104    0.917   0.013   0.013
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1GRMS    2.592    0.050  51.897    0.000   2.592   3.415
## .p2GRMS    2.545    0.050  50.567    0.000   2.545   3.327
## .p3GRMS    2.579    0.046  55.829    0.000   2.579   3.673
## .p1MH      3.582    0.054  66.613    0.000   3.582   4.383
## .p2MH      2.866    0.070  41.024    0.000   2.866   2.699
## .p3MH      3.035    0.066  46.141    0.000   3.035   3.036
## .p1PhH     3.128    0.060  52.389    0.000   3.128   3.447
## .p2PhH     2.652    0.055  48.062    0.000   2.652   3.162
## .p3PhH     3.067    0.058  53.264    0.000   3.067   3.504
## .Spirit1    2.511    0.044  56.810    0.000   2.511   3.738
## .Spirit2    2.437    0.042  58.201    0.000   2.437   3.829
## .SocS1      2.550    0.040  64.171    0.000   2.550   4.222
## .SocS2      2.758    0.043  64.666    0.000   2.758   4.255
## .Eng1       2.437    0.044  55.380    0.000   2.437   3.644
## .Eng2       2.515    0.038  65.524    0.000   2.515   4.311
## .dEng1     2.502    0.040  63.179    0.000   2.502   4.157
## .dEng2     2.455    0.041  59.610    0.000   2.455   3.922
## GRMS       0.000
## .MH        0.000
## .PhH        0.000
## .Spr        0.000
## .SSp        0.000
## .Eng        0.000
## .dEng       0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1GRMS    0.050    0.007   6.754    0.000   0.050   0.087
## .p2GRMS    0.044    0.007   6.098    0.000   0.044   0.075

```

```

##  .p3GRMS      0.064  0.008  8.400  0.000  0.064  0.131
##  .p1MH       0.273  0.037  7.378  0.000  0.273  0.409
##  .p2MH       0.542  0.067  8.059  0.000  0.542  0.481
##  .p3MH       0.387  0.055  7.083  0.000  0.387  0.387
##  .p1PhH      0.459  0.058  7.915  0.000  0.459  0.557
##  .p2PhH      0.333  0.049  6.816  0.000  0.333  0.474
##  .p3PhH      0.375  0.054  6.997  0.000  0.375  0.490
##  .Spirit1     0.214  0.026  8.135  0.000  0.214  0.473
##  .Spirit2     0.167  0.023  7.223  0.000  0.167  0.413
##  .SocS1       0.184  0.026  7.154  0.000  0.184  0.506
##  .SocS2       0.240  0.029  8.139  0.000  0.240  0.571
##  .Eng1        0.282  0.033  8.558  0.000  0.282  0.631
##  .Eng2        0.175  0.026  6.695  0.000  0.175  0.515
##  .dEng1       0.205  0.029  7.116  0.000  0.205  0.567
##  .dEng2       0.235  0.030  7.702  0.000  0.235  0.599
##  GRMS         0.526  0.054  9.787  0.000  1.000  1.000
##  .MH          0.164  0.042  3.929  0.000  0.416  0.416
##  .PhH          0.255  0.058  4.403  0.000  0.700  0.700
##  .Spr          0.076  0.019  4.051  0.000  0.321  0.321
##  .SSp          0.094  0.021  4.414  0.000  0.519  0.519
##  .Eng          0.084  0.022  3.809  0.000  0.509  0.509
##  .dEng         0.078  0.024  3.332  0.001  0.500  0.500
##
## R-Square:
##                               Estimate
##  p1GRMS      0.913
##  p2GRMS      0.925
##  p3GRMS      0.869
##  p1MH        0.591
##  p2MH        0.519
##  p3MH        0.613
##  p1PhH       0.443
##  p2PhH       0.526
##  p3PhH       0.510
##  Spirit1     0.527
##  Spirit2     0.587
##  SocS1       0.494
##  SocS2       0.429
##  Eng1        0.369
##  Eng2        0.485
##  dEng1       0.433
##  dEng2       0.401
##  MH          0.584
##  PhH          0.300
##  Spr          0.679
##  SSp          0.481
##  Eng          0.491
##  dEng         0.500

```

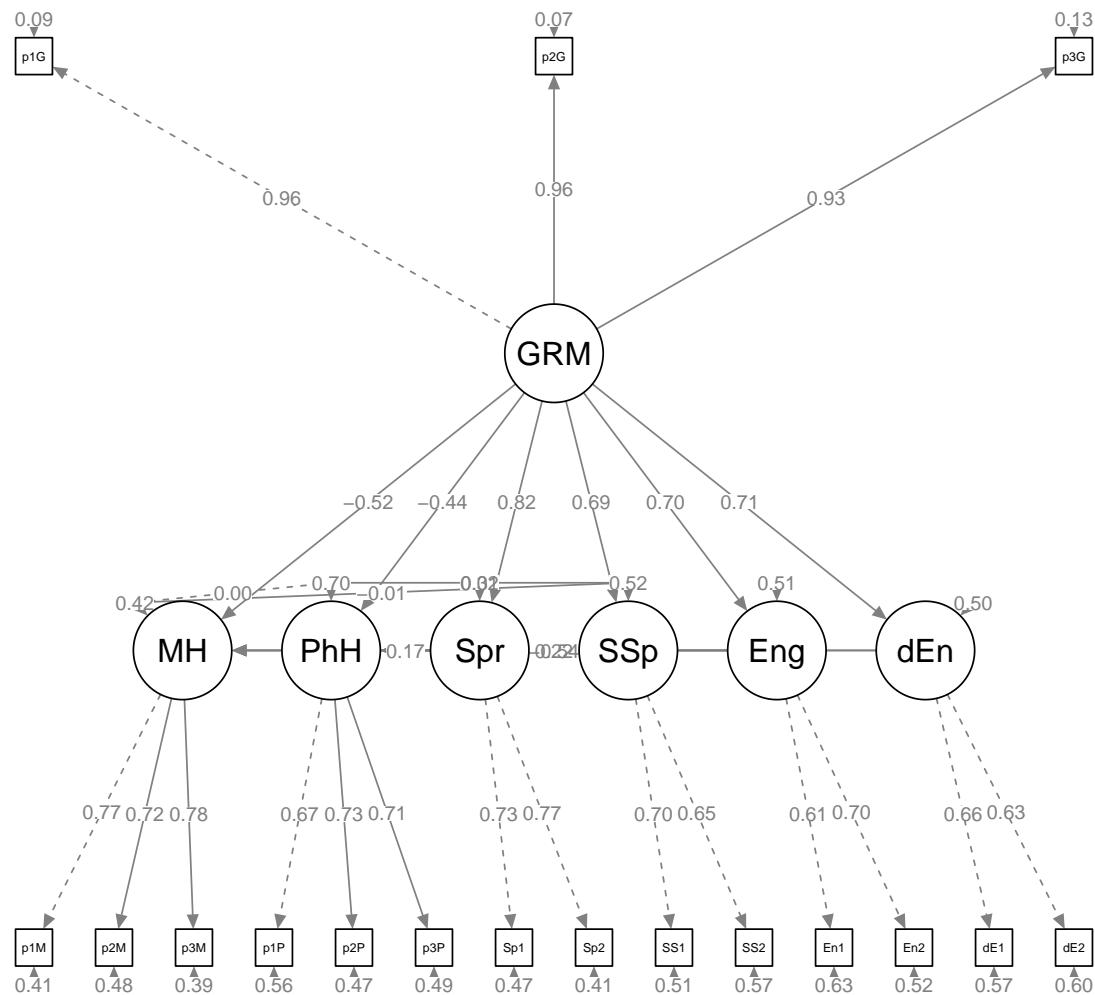
```
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## indirect.EngMH      0.139   0.089   1.564   0.118   0.161   0.161
## indirect.SprMH      0.086   0.121   0.710   0.478   0.099   0.099
## indirct.dEngMH     -0.329   0.122  -2.685   0.007  -0.379  -0.379
## indirct.EngPhH      0.130   0.099   1.307   0.191   0.156   0.156
## indirct.SprPhH      0.119   0.141   0.842   0.400   0.142   0.142
## indrct.dEngPhH     -0.241   0.117  -2.060   0.039  -0.289  -0.289
## direct.MH          -0.450   0.193  -2.336   0.019  -0.520  -0.520
## direct.PhH          -0.363   0.209  -1.735   0.083  -0.435  -0.435
## total.MH          -0.554   0.062  -8.932   0.000  -0.639  -0.639
## total.PhH          -0.394   0.083  -4.752   0.000  -0.445  -0.445
```

```
# struct_pEsts3 #although creating the object is useful to export as
# a .csv I didn't ask it to print into the book
```

Creating a figure will help us with the conceptual understanding of what we've just done (and will also help us check our work).

Let's first create a fresh plot from se

```
plot_struct3 <- semPlot::semPaths(struct_fit3, what = "col", whatLabels = "stand",
  sizeMan = 3, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(2, 2, 2, 2), structural = FALSE, curve = FALSE, intercepts = FALSE)
```



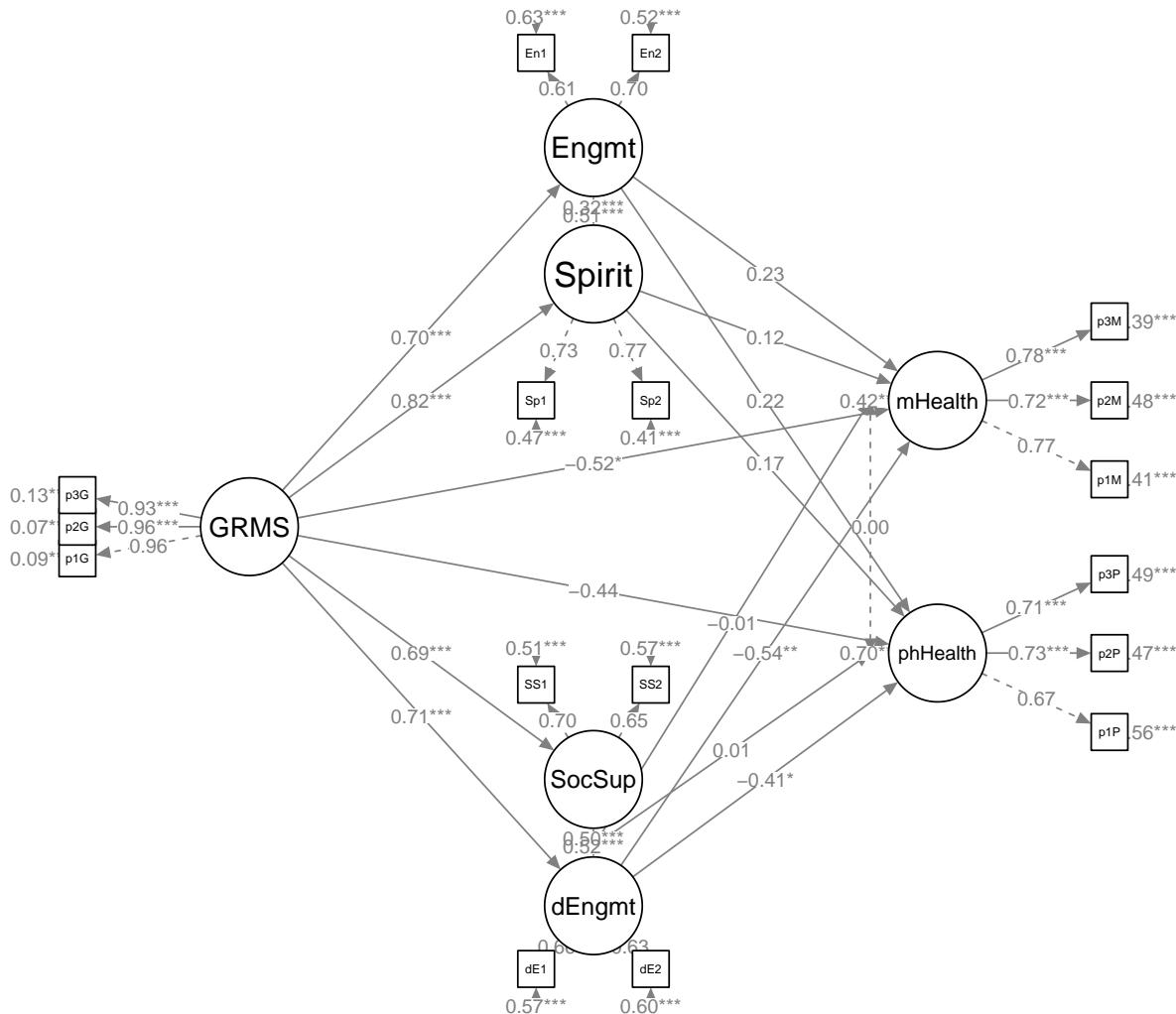
Because we haven't added or deleted variables (only 1 path) and we want them to stay in the same location with the same orientation of paths and covariances, we should be able to recycle some of the diagram code we created earlier

```
plot3 <- semtools::set_sem_layout(plot_struct3, indicator_order = m1_indicator_order,
  indicator_factor = m1_indicator_factor, factor_layout = m1_msmt, factor_point_to = m1_point
  indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)

# changing node labels
plot3 <- semtools::change_node_label(plot3, c(GRM = "GRMS", MH = "mHealth",
  PhH = "phHealth", Eng = "Engmt", dEn = "dEngmt", Spr = "Spirit", SSp = "SocSup"),
  label.cex = 1.1)

# adding stars to indicate significant paths
plot3 <- semtools::mark_sig(plot3, struct_fit3)

plot(plot3)
```



Close inspection shows that there are no longer b paths from social support to mental and physical health.

Curiously, even though we deleted two paths, our degrees of freedom was the same as the original model. This means that our fit indices will be identical: $\chi^2(109) = 120.319, p = 0.216, CFI = 0.994, RMSEA = 0.021, 90CI[0.000, 0.041], SRMR = 0.044$.

We can formally test differences in the two models with the `lavaan::lavTestLRT()` function.

```
lavaan::lavTestLRT(struct_fit1, struct_fit3)
```

```
## Warning in lavaan::lavTestLRT(struct_fit1, struct_fit3): lavaan WARNING:
##   Some restricted models fit better than less restricted models;
##   either these models are not nested, or the less restricted model
##   failed to reach a global optimum. Smallest difference =
##   -0.00500996567915024

## Warning in lavaan::lavTestLRT(struct_fit1, struct_fit3): lavaan WARNING: some
## models have the same degrees of freedom
```

```

##  

## Chi-Squared Difference Test  

##  

##          Df      AIC      BIC  Chisq Chisq diff RMSEA Df diff Pr(>Chisq)  

## struct_fit1 109 6901.7 7111.7 120.32  

## struct_fit3 109 6901.7 7111.7 120.32    -0.00501      0         0

```

As shown, this $df = 0$ test indicated that the models are not statistically significantly different from each other.

When researchers engage in model trimming, their (typical) hope is that the models are not statistically significantly different from each other. Why? This means that the more parsimonious model (i.e., the nested model, the one with fewer paths/covariances) fits similarly to the sample covariance matrix (i.e., where all variables freely covary).

11.10.0.1 Extreme Model Trimming

In the description of the Lewis et al. [2017], I noted that the researchers conducted eight simple mediations instead of two parallel mediations (for mental and physical health, respectively), or as we have done – a complex SEM modeling all four mediators with both dependent variables. Given the small regression weights and very non-significant p values for spirituality and social support, I’m curious what would happen if we respecified the model, retaining only disengagement and engagement variables. This level of respecification would render our model as *non-nested* so our comparison would be limited to the AIC/BIC (lower values indicate better fit) and the global fit indices, themselves (without formal comparison).

```

struct_mod4 <- "  

  ##measurement model  

  GRMS =~ p1GRMS + p2GRMS + p3GRMS  

  MH =~ p1MH + p2MH + p3MH  

  PhH =~ p1PhH + p2PhH + p3PhH  

  #Spr =~ v1*Spirit1 + v1*Spirit2 #trimmed altogether  

  #SSp =~ v2*SocS1 + v2*SocS2 #trimmed altogether  

  Eng =~ v3*Eng1 + v3*Eng2  

  dEng =~ v4*dEng1 + v4*dEng2  

  #structural model with labels for calculation of the indirect effect  

  Eng ~ a1*GRMS  

  #Spr ~ a2*GRMS #trimmed altogether  

  #SSp ~ a3*GRMS #trimmed altogether  

  dEng ~ a4*GRMS  

  MH ~ c_p1*GRMS + b1*Eng + b4*dEng #trimming + b2*Spr + b3*SSp  

  PhH ~ c_p2*GRMS + b5*Eng + b8*dEng #trimming + b6*Spr + b7*SSp  

  #cov

```

```

MH ~~ 0*PhH #prevents MH and PhD from correlating

#calculations
indirect.EngMH := a1*b1
#indirect.SprMH := a2*b2 #trimmed after deleting Spr
#indirect.SSpMH := a3*b3 #trimmed after deleting SSp
indirect.dEngMH := a4*b4

indirect.EngPhH := a1*b5
#indirect.SprPhH := a2*b6 #trimmed after deleting Spr
#indirect.SSpPhH := a3*b7 #trimmed after deleting Ssp
indirect.dEngPhH := a4*b8
direct.MH := c_p1
direct.PhH := c_p2
total.MH := c_p1 + (a1*b1) + (a4*b4) #trimmed + (a2*b2) + (a3*b3)
total.PhH := c_p2 + (a1*b5) + (a1*b8) #trimmed + (a1*b6) + (a1*b7)

"
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
struct_fit4 <- lavaan::sem(struct_mod4, data = dfLewis, missing = "fiml")
struct_summary4 <- lavaan::summary(struct_fit4, fit.measures = TRUE, standardized = TRUE,
                                    rsq = TRUE)
struct_pEsts4 <- lavaan::parameterEstimates(struct_fit4, boot.ci.type = "bca.simple",
                                             standardized = TRUE)
struct_summary4

## lavaan 0.6.16 ended normally after 64 iterations
##
##             ML
## Estimator
## Optimization method           NLMINB
## Number of model parameters    45
## 
##             231
## Number of observations
## Number of missing patterns    1
## 
## Model Test User Model:
## 
##             59.999
## Test statistic
## Degrees of freedom            59
## P-value (Chi-square)          0.439
## 
## Model Test Baseline Model:
## 
##             1675.582
## Test statistic
## Degrees of freedom            78
## P-value                         0.000
## 
```

```

## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)          0.999
## Tucker-Lewis Index (TLI)            0.999
##
## Robust Comparative Fit Index (CFI)    0.999
## Robust Tucker-Lewis Index (TLI)      0.999
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)        -2675.295
## Loglikelihood unrestricted model (H1) -2645.296
##
## Akaike (AIC)                         5440.591
## Bayesian (BIC)                        5595.500
## Sample-size adjusted Bayesian (SABIC)  5452.875
##
## Root Mean Square Error of Approximation:
##
## RMSEA                                0.009
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.042
## P-value H_0: RMSEA <= 0.050          0.989
## P-value H_0: RMSEA >= 0.080          0.000
##
## Robust RMSEA                           0.009
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.042
## P-value H_0: Robust RMSEA <= 0.050   0.989
## P-value H_0: Robust RMSEA >= 0.080   0.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                                 0.043
##
## Parameter Estimates:
##
## Standard errors                      Standard
## Information                            Observed
## Observed information based on        Hessian
##
## Latent Variables:
##                   Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## GRMS =~
##   p1GRMS      1.000
##   p2GRMS      1.011  0.030 33.674  0.000  0.726  0.956
##   p3GRMS      0.906  0.030 30.368  0.000  0.734  0.959
## MH =~

```

```

##      p1MH          1.000
##      p2MH          1.219  0.126  9.704  0.000  0.766  0.721
##      p3MH          1.245  0.112 11.078  0.000  0.782  0.782
## PhH =~
##      p1PhH         1.000
##      p2PhH         1.008  0.128  7.879  0.000  0.608  0.725
##      p3PhH         1.037  0.135  7.663  0.000  0.626  0.715
## Eng =~
##      Eng1        (v3)  1.000
##      Eng2        (v3)  1.000
## dEng =~
##      dEng1       (v4)  1.000
##      dEng2       (v4)  1.000
##
## Regressions:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## Eng ~
##      GRMS    (a1)  0.389  0.041  9.419  0.000  0.697  0.697
## dEng ~
##      GRMS    (a4)  0.385  0.041  9.351  0.000  0.705  0.705
## MH ~
##      GRMS    (c_p1) -0.374  0.161 -2.319  0.020 -0.432 -0.432
##      Eng     (b1)   0.373  0.220  1.692  0.091  0.241  0.241
##      dEng    (b4)  -0.841  0.302 -2.784  0.005 -0.531 -0.531
## PhH ~
##      GRMS    (c_p2) -0.259  0.168 -1.541  0.123 -0.312 -0.312
##      Eng     (b5)   0.360  0.247  1.456  0.146  0.242  0.242
##      dEng    (b8)  -0.611  0.294 -2.083  0.037 -0.401 -0.401
##
## Covariances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .MH ~~
##      .PhH      0.000
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1GRMS     2.592  0.050 51.897  0.000  2.592  3.415
## .p2GRMS     2.545  0.050 50.567  0.000  2.545  3.327
## .p3GRMS     2.579  0.046 55.829  0.000  2.579  3.673
## .p1MH       3.582  0.054 66.626  0.000  3.582  4.384
## .p2MH       2.866  0.070 41.030  0.000  2.866  2.700
## .p3MH       3.035  0.066 46.150  0.000  3.035  3.036
## .p1PhH      3.128  0.060 52.387  0.000  3.128  3.447
## .p2PhH      2.652  0.055 48.059  0.000  2.652  3.162
## .p3PhH      3.067  0.058 53.261  0.000  3.067  3.504
## .Eng1       2.437  0.044 55.381  0.000  2.437  3.644
## .Eng2       2.515  0.038 65.523  0.000  2.515  4.311
## dEng1      2.502  0.040 63.148  0.000  2.502  4.155

```

```

## .dEng2          2.455    0.041   59.634    0.000    2.455    3.924
## GRMS           0.000
## .MH            0.000
## .PhH            0.000
## .Eng            0.000
## .dEng           0.000
##
## Variances:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .p1GRMS          0.049    0.008   6.508    0.000    0.049    0.085
## .p2GRMS          0.047    0.008   6.191    0.000    0.047    0.080
## .p3GRMS          0.061    0.008   8.075    0.000    0.061    0.123
## .p1MH            0.273    0.037   7.431    0.000    0.273    0.410
## .p2MH            0.541    0.067   8.067    0.000    0.541    0.480
## .p3MH            0.388    0.055   7.098    0.000    0.388    0.388
## .p1PhH           0.459    0.058   7.923    0.000    0.459    0.558
## .p2PhH           0.333    0.049   6.821    0.000    0.333    0.474
## .p3PhH           0.374    0.054   6.993    0.000    0.374    0.489
## .Eng1            0.283    0.033   8.541    0.000    0.283    0.632
## .Eng2            0.176    0.026   6.668    0.000    0.176    0.517
## .dEng1           0.206    0.029   7.085    0.000    0.206    0.567
## .dEng2           0.234    0.031   7.671    0.000    0.234    0.599
## GRMS            0.527    0.054   9.803    0.000    1.000    1.000
## .MH              0.166    0.042   3.957    0.000    0.421    0.421
## .PhH             0.257    0.058   4.414    0.000    0.707    0.707
## .Eng             0.085    0.022   3.798    0.000    0.515    0.515
## .dEng            0.079    0.024   3.326    0.001    0.503    0.503
##
## R-Square:
##                               Estimate
## p1GRMS          0.915
## p2GRMS          0.920
## p3GRMS          0.877
## p1MH            0.590
## p2MH            0.520
## p3MH            0.612
## p1PhH           0.442
## p2PhH           0.526
## p3PhH           0.511
## Eng1            0.368
## Eng2            0.483
## dEng1           0.433
## dEng2           0.401
## MH              0.579
## PhH             0.293
## Eng              0.485
## dEng            0.497
##

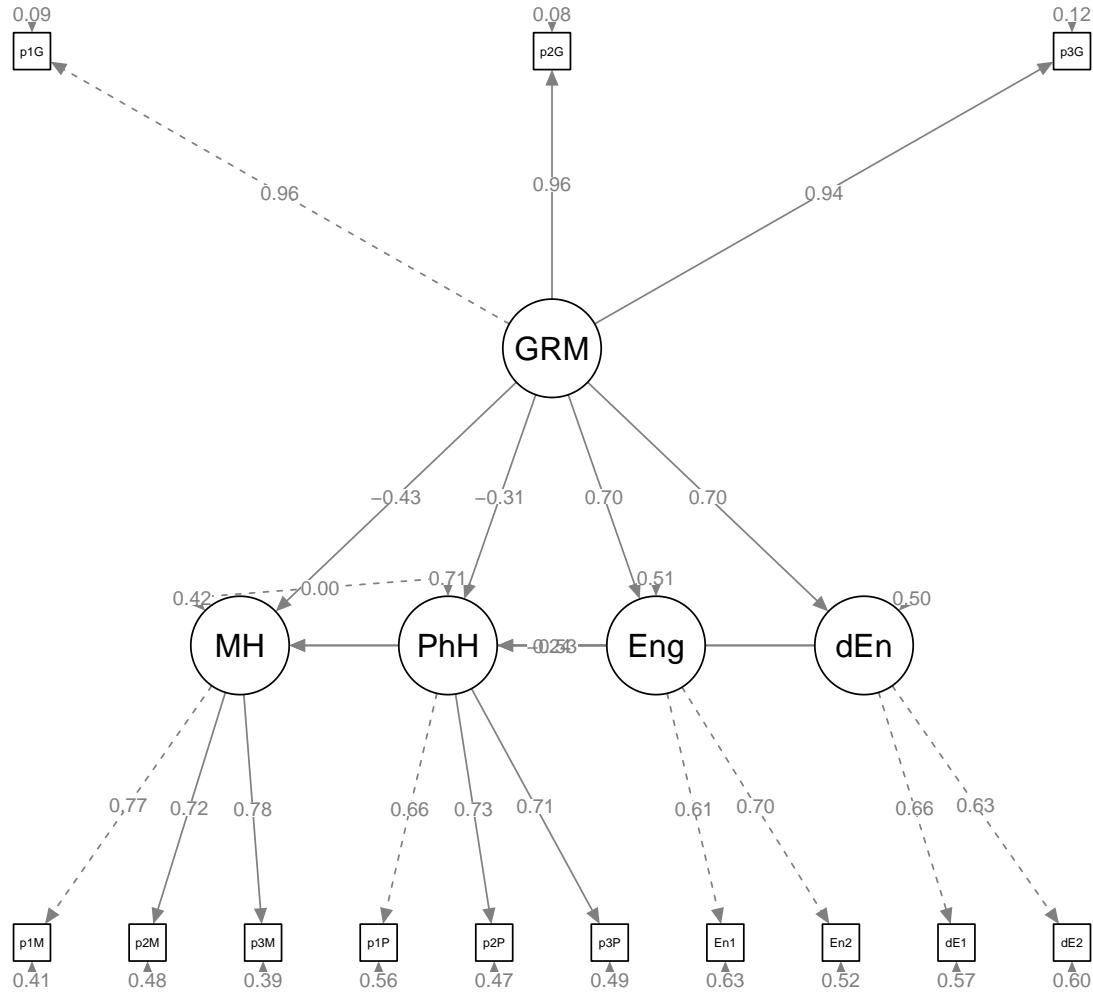
```

```
## Defined Parameters:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## indirect.EngMH   0.145  0.088   1.656   0.098   0.168   0.168
## indirct.dEngMH -0.324  0.122  -2.662   0.008  -0.374  -0.374
## indirct.EngPhH  0.140  0.098   1.433   0.152   0.169   0.169
## indrct.dEngPhH -0.235  0.116  -2.029   0.043  -0.283  -0.283
## direct.MH       -0.374  0.161  -2.319   0.020  -0.432  -0.432
## direct.PhH      -0.259  0.168  -1.541   0.123  -0.312  -0.312
## total.MH        -0.552  0.062  -8.930   0.000  -0.638  -0.638
## total.PhH       -0.357  0.071  -5.054   0.000  -0.423  -0.423
```

```
# struct_pEsts4 #although creating the object is useful to export as
# a .csv I didn't ask it to print into the book
```

Creating a figure to represent our analysis will require us to tweak the full set of code.

```
plot_struct4 <- semPlot::semPaths(struct_fit4, what = "col", whatLabels = "stand",
  sizeMan = 3, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(2, 2, 2, 2), structural = FALSE, curve = FALSE, intercepts = FALSE)
```



Although the code below may look daunting, I find it to be a fairly straightforward way to obtain figures that convey the model we are testing. We first start by identifying the desired location of our latent variables, using numbers to represent their position by “(column, row)”. In the table below, I have mapped my variables.

Grid for Plotting `semplot::sempath`

(1,1)	(1,2) Eng	(1,3)
(2,1)	(2,2)	(2,3) MH
(3,1) GRM	(3,2)	(3,3)
(4,1)	(4,2)	(4,3) PhH
(5,1)	(5,2) dEn	(5,3)

We place these values along with the names of our latent variables in to the `semp-tools::layout_matrix` function.

```

#IMPORTANT: Must use the node names (take directly from the SemPlot) assigned by SemPlot
#You can change them as the last thing
m4_msmt <- semptools::layout_matrix(GRM = c(3,1),
                                      Eng = c(1,2),
                                      dEn = c(5,2),
                                      MH = c(2,3),
                                      PhH = c(4,3))

#tell where you want the indicators to face
m4_point_to <- semptools::layout_matrix(left = c(3,1),
                                         up = c(1,2),
                                         down = c(5,2),
                                         right = c(2,3),
                                         right = c(4,3))

#the next two codes -- indicator_order and indicator_factor are paired together, they specify
m4_indicator_order <- c("p1G", "p2G", "p3G",
                        "p1M", "p2M", "p3M",
                        "p1P", "p2P", "p3P",
                        "En1", "En2",
                        "dE1", "dE2")

m4_indicator_factor <- c("GRM", "GRM", "GRM",
                         "MH", "MH", "MH",
                         "PhH", "PhH", "PhH",
                         "Eng", "Eng",
                         "dEn", "dEn")

#next set of code pushes the indicator variables away from the factor
m4_indicator_push <- c(GRM = .5,
                       MH = 1,
                       PhH = 1,
                       Eng = 1.5,
                       dEn = 2)

m4_indicator_spread <- c(GRM = .5, #spreading the boxes away from each other
                         MH = 1,
                         PhH = 1,
                         Eng = 1,
                         dEn = 1)

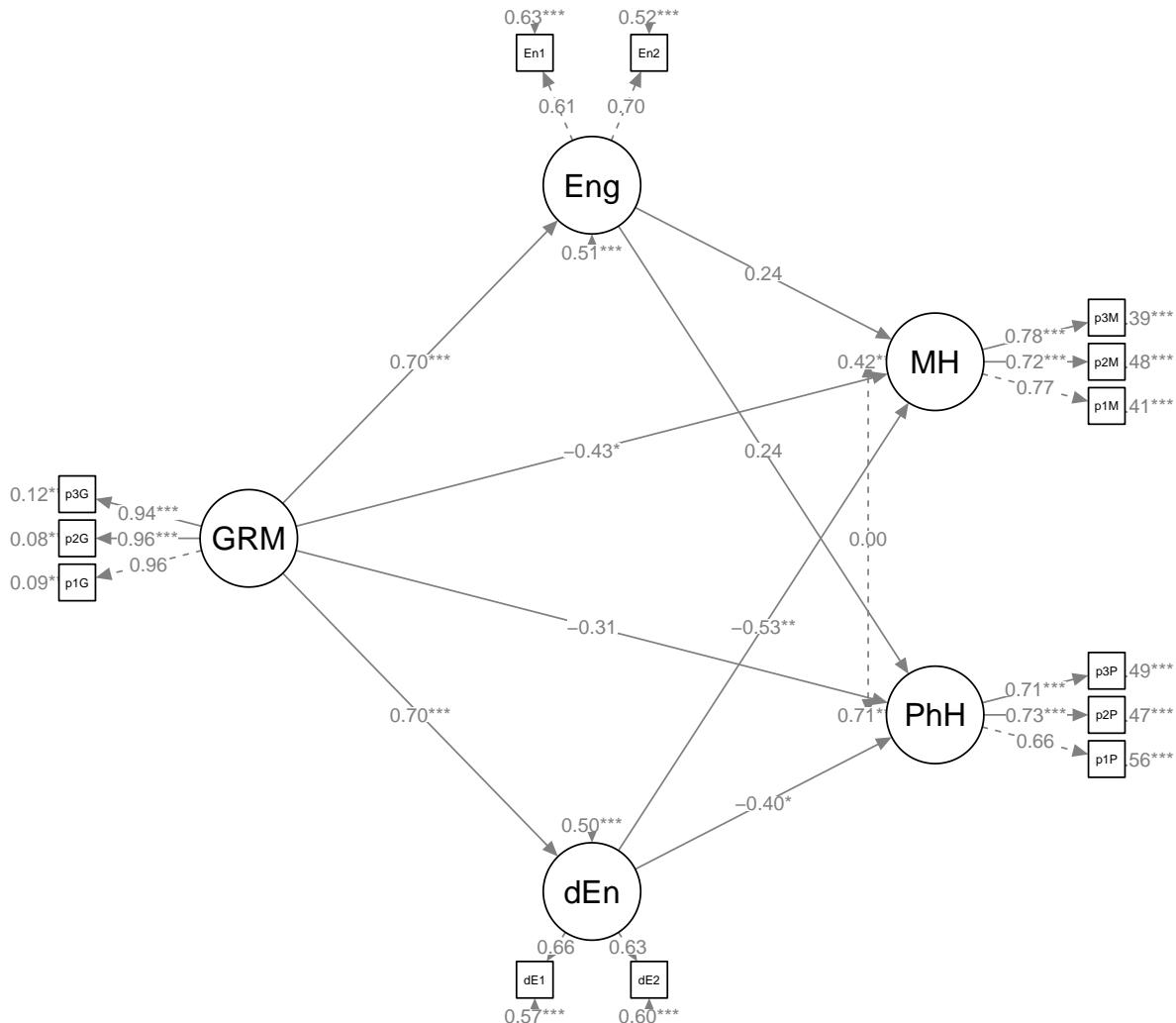
plot4 <- semptools::set_sem_layout(plot_struct4,
                                    indicator_order = m4_indicator_order,
                                    indicator_factor = m4_indicator_factor,
                                    factor_layout = m4_msmt,
                                    factor_point_to = m4_point_to,
                                    indicator_push = m4_indicator_push,
                                    indicator_spread = m4_indicator_spread)

```

```
#changing node labels -- throwing an error and I'm not sure why, it worked above
#plot4 <- semptools::change_node_label(plot4,
                                         #c(GRM = "GRMS",
                                         #MH = "mHealth",
                                         # PhH = "phHealth",
                                         #Eng = "Engmt",
                                         #dEn = "dEngmt",
                                         #label.cex = 1.1))

#adding stars to indicate significant paths
plot4 <- semptools::mark_sig(plot4, struct_fit4)

plot(plot4)
```



Our global fit indices were strong, $\chi^2(59) = 59.999, p = 0.439, CFI = 0.999, RMSEA = 0.009, 90CI[0.000, 0.042], SRMR = 0.043$. Yet the indirect effects for engagement remained

non-significant. When mental health was the dependent variable, $B = 0.145, p = 0.098$; when physical health was the dependent variable, $B = 0.140, p = 0.152$.

If we formally compare the results, we can only use the AIC and BIC with lower values indicating better fit.

```

lavaan:::lavTestLRT(struct_fit1, struct_fit4)

## Warning in lavaan:::lavTestLRT(struct_fit1, struct_fit4): lavaan WARNING: some
## models are based on a different set of observed variables

##
## Chi-Squared Difference Test
##
##          Df      AIC      BIC    Chisq Chisq diff      RMSEA Df diff Pr(>Chisq)
## struct_fit4 59 5440.6 5595.5  59.999
## struct_fit1 109 6901.7 7111.7 120.324      60.324 0.029898      50      0.1504

```

Remember – ignore the chi-square difference test. These are non-nested models. The AIC and BIC favor our trimmed model.

What would I do? It would be tempting to simply report the results of engagement and disengagement coping – yet the researchers were interested in the potential of all four coping strategies as potential mediators. Because our disengagement coping mediator remained statistically significant throughout all of our exploration, I would probably choose to report the full analysis because it provides useful information to future researchers and practitioners about the important contribution of disengagement coping (while controlling for the other mediators).

11.10.1 APA Style Write-up of the Results

Method/Analytic Strategy

We specified a parallel mediation, predicting mental and physical health directly from gendered racial microaggressions and indirectly through four coping strategies (i.e., spirituality, social support, engagement, disengagement). The primary analysis occurred in two stages. First, we specified and evaluated a measurement model. Data were analyzed with a maximum likelihood approach the package, *lavaan* (v. 0.6-16).

Results

Preliminary Analyses

- Missing data analyses and managing missing data>
- Bivariate correlations, means, SDs
- Distributional characteristics, assumptions, etc.
- Address limitations and concerns

Primary Analyses Analyzing our proposed multiple mediator model followed the two-step procedure of first establishing a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index (CFI) ≥ 0.95 , root-mean-square error of approximation (RMSEA) ≤ 0.06 , and the standard root-mean-square residual (SRMR) ≤ 0.08 . To establish acceptable fit, we used CFI ≥ 0.90 , RMSEA ≤ 0.10 , and SRMR ≤ 0.10 [Weston and Gore, 2006].

Following recommendations by Little et al. [Little et al., 2002, 2013], each latent variable with six or more indicators was represented by three parcels. Parcels were created by randomly assigning scale items to the parcels and then calculating the mean, if at least 65% of the items were non-missing. For the four latent variables with only two indicators each, we constrained the factor loadings to be equal. Factor loadings were all strong, statistically significant, and properly valenced. Global fit statistics were within acceptable thresholds ($\chi^2(102) = 94.122, p = 0.698, CFI = 1.000, RMSEA = 0.000, 90CI[0.000, 0.028], SRMR = 0.034$). Thus, we proceeded to testing the structural model.

Our structural model was a parallel mediation, predicting mental and physical health directly from gendered racial microaggressions and indirectly through four coping strategies (i.e., spirituality, social support, engagement, disengagement). Results of the global fit indices suggested good fit $\chi^2(109) = 120.324, p = 0.216, CFI = 0.994, RMSEA = 0.021, 90CI[0.000, 0.041], SRMR = 0.044$. As shown in Table 2 only two of the eight indirect effects was statistically significant. For both, disengagement coping mediated the effect of gendered racial microaggressions on mental health ($B = -0.328, p = 0.007$) and physical health ($B = -0.241, p = 0.039$). The model accounted for 48-68% of the variance in the mediators and 58% and 30% of the variance in mental and physical health, respectively.

Although the model fit was strong, we considered respecification. Modification indices suggested adding a path that would result in a serial mediation from gendered racial microaggressions, to spirituality, to social support and then to each of the dependent variables. While model fit ($\chi^2(108) = 111.657, p = 0.385, CFI = 0.998, RMSEA = 0.012, 90CI[0.000, 0.036], SRMR = 0.042$) was statistically significantly improved in favor of the serially mediated model ($\Delta\chi^2(1) = 8.667, p = 0.032$), the serially mediated indirect effects to mental ($B = -0.022, p = 0.737$) and physical health ($B = -0.019, p = 0.800$) were non-significant. Consequently, we retained our originally hypothesized model.

We also considered model trimming. Considering both theory plus regression weights and associated p values of the originally hypothesized model, we noted that spiritual and social support coping were not statistically significant with low regression weights. We further noted that engagement coping approached statistical significance. We trimmed the spiritual and social support mediators altogether. While the resulting model (with only engagement and disengagement coping) evidenced strong fit ($\chi^2(59) = 59.999, p =$

0.439 , $CFI = 0.999$, $RMSEA = 0.009$, $90CI[0.000, 0.042]$, $SRMR = 0.043$), the indirect effects to mental health ($B = 0.145$, $p = 0.098$) and physical health ($B = 0.140$, $p = 0.152$) through engagement coping remained non-significant. Because conveying information about significant and not significant paths are important to future researchers and practitioners, we retained the originally hypothesized model.

11.11 STAY TUNED

A section on power analysis is planned and coming soon! My apologies that it's not quite Ready.

11.12 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 start with a hypothesized model that has at least four variables and is over-identified. Overall you will (a) start with an established measurement model, (b) test a structural model, (c) use modification indices to add a path or covariance and evaluate the change to the model, (d) use strength and significance of regression weights to trim at least one path or covariance and evaluate the change to the model, (e) make a final decision about the model you retain, (f) provide an APA style representation of the results (with table[s] and figure[s]) .

11.12.1 Problem #1: Change the random seed

Simply change the random seed in the data simulation, then rework the problem evaluated in this chapter. It is possible to further simplify this model by deleting a pair of the mediators and/or one of the dependent variables.

11.12.2 Problem #2: Swap one or more of the variables

The Lewis et al. [2017] study included the additional variable of gendered racial identity centrality. Consider substituting it as an independent or dependent variable.

11.12.3 Problem #3: Try something entirely new.

Conduct a hybrid analysis 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).

Regardless of your choice(s) complete all the elements listed in the grading rubric.

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. Identify the structural model you will evaluate. It should have a minimum of four variables and could be one of the prior path-level models you already examined.	5	_____
2. Import the data and format the variables in the model.	5	_____
3. Specify and evaluate a <i>measurement</i> model that you have established.	10	_____
4. Specify and evaluate a <i>structural</i> model. For the purpose of this exercise, the structural model should be over-identified, that is, should have positive degrees of freedom. How many degrees of freedom does your structural model have?	10	_____
5. Use modification indices to add at least one path or covariance.	10	_____
6. Conduct a formal comparison of <i>global fit</i> between the original and respecified model.	5	_____
7. Using the strength and significance of regression weights as a guide, trim at least path or covariance.	10	_____
8. Conduct a formal comparison of <i>global fit</i> between the original (or built) and trimmed model.	5	_____
9. APA style results with table(s) and figure(s).	5	_____
10. Explanation to grader.	5	_____
Totals	65	_____

11.13 Homeworked Example

Screencast Link

For more information about the data used in this homeworked example, please refer to the description and codebook located at the end of the [introductory lesson](#) in [ReCentering Psych Stats](#). An .rds file which holds the data is located in the [Worked Examples](#) folder at the GitHub site the hosts the OER. The file name is *ReC.rds*.

The suggested practice problem for this chapter is to evaluate the measurement model that would precede the evaluation of a structural model. And actually, we will need to evaluate two measurement models – an “all items” on indicators model and a parceled model.

Identify the structural model you will evaluate

It should have a minimum of four variables and could be one of the prior path-level models you already examined.

X = Centering: explicit recentering (0 = precentered; 1 = recentered) M1 = TradPed: traditional pedagogy (continuously scaled with higher scores being more favorable) M2 = SRPed: socially

responsive pedagogy (continuously scaled with higher scores being more favorable) Y = Valued: valued by me (continuously scaled with higher scores being more favorable)

In this *parallel mediation*, I am hypothesizing that the perceived course value to the students is predicted by intentional recentering through their assessments of traditional and socially responsive pedagogy.

It helps me to make a quick sketch:

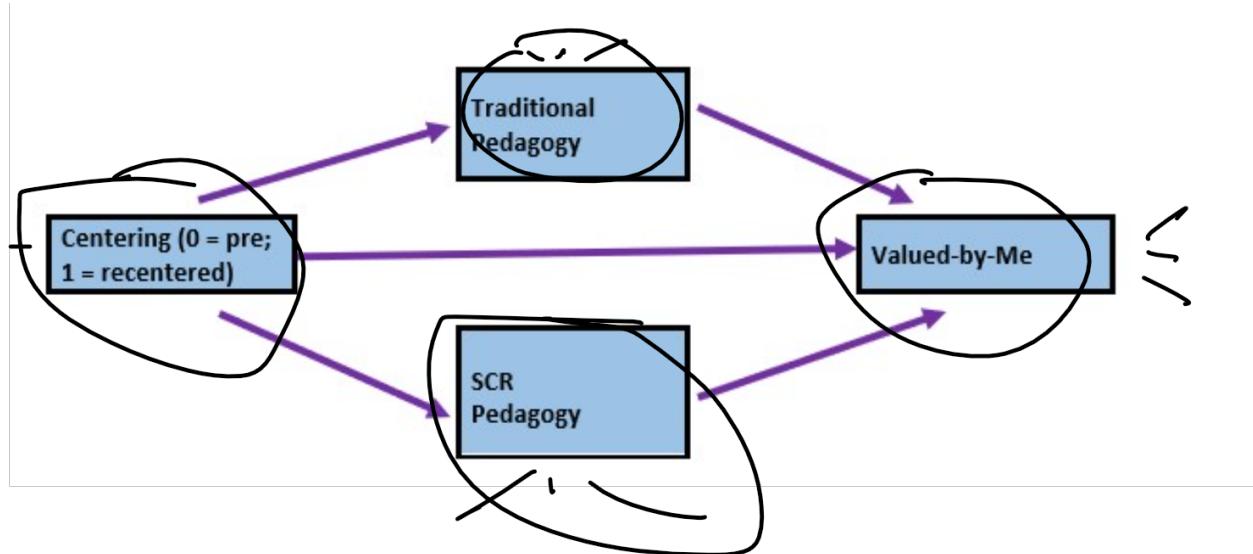


Figure 11.4: An image of the parallel mediation model for the homeworked example

Import the data and format the variables in the model.

```
raw <- readRDS("ReC.rds")
```

The approach we are taking to complex mediation does not allow dependency in the data. Therefore, we will include only those who took the multivariate class (i.e., excluding responses for the ANOVA and psychometrics courses).

```
raw <- (dplyr::filter(raw, Course == "Multivariate"))
```

Although this dataset is overall small, I will go ahead and make a babydf with the item-level variables.

```
babydf <- dplyr::select(raw, Centering, ClearResponsibilities, EffectiveAnswers,
  Feedback, ClearOrganization, ClearPresentation, ValObjectives, IncrUnderstanding,
  IncrInterest, InclusvClassrm, EquitableEval, MultPerspectives, DEIIntegration)
```

Let's check the structure of the variables:

```
str(babydf)
```

```
## Classes 'data.table' and 'data.frame': 84 obs. of 13 variables:
## $ Centering : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ ClearResponsibilities: int 4 5 5 5 4 3 5 5 3 5 ...
## $ EffectiveAnswers : int 4 5 5 4 4 3 5 5 4 4 ...
## $ Feedback : int 4 5 4 4 5 4 5 4 4 5 ...
## $ ClearOrganization : int 3 5 5 4 4 3 5 5 4 5 ...
## $ ClearPresentation : int 4 5 5 3 4 2 5 4 5 5 ...
## $ ValObjectives : int 5 5 5 4 4 5 5 4 4 5 ...
## $ IncrUnderstanding : int 4 5 5 3 4 3 5 4 5 5 ...
## $ IncrInterest : int 4 5 4 3 4 3 5 4 5 4 ...
## $ InclusvClassrm : int 5 5 5 5 4 5 5 5 5 ...
## $ EquitableEval : int 4 5 5 5 4 4 5 4 5 5 ...
## $ MultPerspectives : int 4 5 5 5 5 5 5 4 5 5 ...
## $ DEIintegration : int 5 5 5 5 5 5 5 5 5 5 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

All of the item-level variables are integers (i.e., numerical). This is fine.

The centering variable will need to be dummy coded as 0/1:

```
babydf$CEN <- as.numeric(babydf$Centering)
babydf$CEN <- (babydf$CEN - 1)
str(babydf)
```

```
## Classes 'data.table' and 'data.frame': 84 obs. of 14 variables:
## $ Centering : Factor w/ 2 levels "Pre","Re": 2 2 2 2 2 2 2 2 2 ...
## $ ClearResponsibilities: int 4 5 5 5 4 3 5 5 3 5 ...
## $ EffectiveAnswers : int 4 5 5 4 4 3 5 5 4 4 ...
## $ Feedback : int 4 5 4 4 5 4 5 4 4 5 ...
## $ ClearOrganization : int 3 5 5 4 4 3 5 5 4 5 ...
## $ ClearPresentation : int 4 5 5 3 4 2 5 4 5 5 ...
## $ ValObjectives : int 5 5 5 4 4 5 5 4 4 5 ...
## $ IncrUnderstanding : int 4 5 5 3 4 3 5 4 5 5 ...
## $ IncrInterest : int 4 5 4 3 4 3 5 4 5 4 ...
## $ InclusvClassrm : int 5 5 5 5 5 4 5 5 5 5 ...
## $ EquitableEval : int 4 5 5 5 4 4 5 4 5 5 ...
## $ MultPerspectives : int 4 5 5 5 5 5 5 4 5 5 ...
## $ DEIintegration : int 5 5 5 5 5 5 5 5 5 5 ...
## $ CEN : num 1 1 1 1 1 1 1 1 1 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

Specify and evaluate a *measurement* model that you have established.

As noted in the [homeworked example](#) establishing a measurement model for this dataset may seem tricky. That is, with five, four, and three items on each of the constructs, it seems odd to parcel.

Previous researchers have parceled scales that have more than three items, even when some parcels will have one item each [Spengler et al., 2023]. Correspondingly, I will randomly assign the scales with more than three items each to three parcels.

Here I assign the TradPed items to the 3 parcels.

```
set.seed(230916)
items <- c("ClearResponsibilities", "EffectiveAnswers", "Feedback", "ClearOrganization",
  "ClearPresentation")
parcels <- c("p1_TR", "p2_TR", "p3_TR")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))
```

```
##           items parcel
## 1      ClearPresentation p1_TR
## 2          Feedback    p2_TR
## 3 ClearResponsibilities p3_TR
## 4      EffectiveAnswers p1_TR
## 5      ClearOrganization p2_TR
```

I can now create the parcels using the traditional scoring procedure. Given that we will allow single-item representations, I will score the 2-item variables if at least one is present (i.e., .5).

As a variable, *ClearResponsibilities* will stand alone (i.e., the scoring mechanism won't work on a single variable). Therefore, I will

```
TRp1_vars <- c("ClearPresentation", "EffectiveAnswers")
TRp2_vars <- c("Feedback", "ClearOrganization")

babydf$p1T <- sjstats::mean_n(babydf[, ..TRp1_vars], 0.5)
babydf$p2T <- sjstats::mean_n(babydf[, ..TRp2_vars], 0.5)

# for consistency, I will create a third parcel from the
# ClearResponsibilities variable by duplicating and renaming it

babydf$p3T <- babydf$ClearResponsibilities

# If the scoring code above does not work for you, try the format
# below which involves removing the periods in front of the variable
# list. One example is provided. babydf$p2T <-
# sjstats::mean_n(babydf[, TRp1_vars], .5)
```

Here I assign the socially responsive pedagogy items to three parcels.

```
set.seed(230916)
items <- c("InclusvClassrm", "EquitableEval", "MultPerspectives", "DEIintegration")
parcels <- c("p1_SR", "p2_SR", "p3_SR")
data.frame(items = sample(items), parcel = rep(parcels, length = length(items)))
```

```
##           items parcel
## 1 InclusvClassrm  p1_SR
## 2 MultPerspectives p2_SR
## 3 DEIintegration  p3_SR
## 4 EquitableEval   p1_SR
```

Only parcel one needs to be scored; the remaining are the single items.

```
SRp1_vars <- c("InclusvClassrm", "EquitableEval")

babydf$p1S <- sjstats::mean_n(babydf[, ..SRp1_vars], 0.5)

# Here I create the second and third parcels from the individual
# items by duplicating and naming them

babydf$p2S <- babydf$MultPerspectives
babydf$p3S <- babydf$DEIintegration

# If the scoring code above does not work for you, try the format
# below which involves removing the periods in front of the variable
# list. One example is provided. babydf$p1S <-
# sjstats::mean_n(babydf[, SRp1_vars], .5)
```

I will create “parcels” for the three valued items by naming and duplicating.

```
babydf$p1V <- babydf$ValObjectives
babydf$p2V <- babydf$IncrUnderstanding
babydf$p3V <- babydf$IncrInterest

msmt_mod <- "
  ##measurement model
  CTR =~ CEN #this is a single item indicator, I had to add code below to set the variance
  TrP =~ p1T + p2T + p3T
  SRP =~ p1S + p2S + p3S
  Val =~ p1V + p2V + p3V

  # Variance of the single item indicator
  CTR ~~ 0*CEN

  # Covariances
  CTR ~~ TrP
  CTR ~~ SRP
  CTR ~~ Val
```

```

TrP ~~ SRP
TrP ~~ Val
SRP ~~ Val
"

set.seed(230916)
msmt_fit <- lavaan:::cfa(msmt_mod, data = babydf, missing = "fiml")
msmt_fit_sum <- lavaan:::summary(msmt_fit, fit.measures = TRUE, standardized = TRUE)
msmt_fit_sum

## lavaan 0.6.16 ended normally after 82 iterations
##
## Estimator                               ML
## Optimization method                    NLMINB
## Number of model parameters           35
##
## Number of observations                84
## Number of missing patterns            5
##
## Model Test User Model:
##
## Test statistic                         51.639
## Degrees of freedom                     30
## P-value (Chi-square)                  0.008
##
## Model Test Baseline Model:
##
## Test statistic                         561.645
## Degrees of freedom                     45
## P-value                                0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)          0.958
## Tucker-Lewis Index (TLI)              0.937
##
## Robust Comparative Fit Index (CFI)    0.951
## Robust Tucker-Lewis Index (TLI)       0.927
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)        -635.472
## Loglikelihood unrestricted model (H1) -609.652
##
## Akaike (AIC)                          1340.943
## Bayesian (BIC)                        1426.022
## Sample-size adjusted Bayesian (SABIC) 1315.614

```

```

## 
## Root Mean Square Error of Approximation:
## 

##    RMSEA                      0.093
##    90 Percent confidence interval - lower 0.047
##    90 Percent confidence interval - upper 0.134
##    P-value H_0: RMSEA <= 0.050      0.060
##    P-value H_0: RMSEA >= 0.080      0.709
## 
##    Robust RMSEA                  0.103
##    90 Percent confidence interval - lower 0.058
##    90 Percent confidence interval - upper 0.146
##    P-value H_0: Robust RMSEA <= 0.050 0.030
##    P-value H_0: Robust RMSEA >= 0.080 0.821
## 
## Standardized Root Mean Square Residual:
## 

##    SRMR                      0.052
## 

## Parameter Estimates:
## 

##    Standard errors             Standard
##    Information                 Observed
##    Observed information based on Hessian
## 
## Latent Variables:
## 

##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    CTR =~
##    CEN           1.000                   0.483   1.000
##    TrP =~
##    p1T           1.000                   0.691   0.912
##    p2T           0.963     0.090   10.727   0.000   0.666   0.850
##    p3T           0.952     0.108    8.848   0.000   0.658   0.757
##    SRP =~
##    p1S           1.000                   0.497   0.936
##    p2S           1.061     0.134    7.944   0.000   0.527   0.761
##    p3S           1.408     0.179    7.852   0.000   0.700   0.807
##    Val =~
##    p1V           1.000                   0.352   0.562
##    p2V           2.238     0.399    5.608   0.000   0.788   0.925
##    p3V           2.200     0.400    5.495   0.000   0.774   0.852
## 
## Covariances:
## 

##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    CTR ~~
##    .CEN           0.000                   0.000   NaN
##    TrP            0.075     0.039    1.907   0.056   0.225   0.225
##    SRP            0.068     0.029    2.357   0.018   0.283   0.283

```

```

##      Val          0.032    0.021    1.539    0.124    0.187    0.187
## TrP ~~
##      SRP          0.273    0.053    5.122    0.000    0.795    0.795
##      Val          0.219    0.053    4.132    0.000    0.902    0.902
## SRP ~~
##      Val          0.121    0.032    3.783    0.000    0.691    0.691
##
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CEN          0.369   0.053   7.009   0.000   0.369   0.765
## .p1T          4.321   0.083  52.220   0.000   4.321   5.698
## .p2T          4.196   0.085  49.115   0.000   4.196   5.359
## .p3T          4.478   0.095  47.081   0.000   4.478   5.152
## .p1S          4.637   0.058  80.049   0.000   4.637   8.734
## .p2S          4.411   0.076  58.169   0.000   4.411   6.367
## .p3S          4.360   0.099  43.986   0.000   4.360   5.029
## .p1V          4.464   0.068  65.385   0.000   4.464   7.134
## .p2V          4.190   0.093  45.087   0.000   4.190   4.919
## .p3V          3.986   0.099  40.142   0.000   3.986   4.389
## CTR          0.000
## TrP          0.000
## SRP          0.000
## Val          0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CEN          0.000
## .p1T          0.097   0.026   3.798   0.000   0.097   0.169
## .p2T          0.170   0.035   4.918   0.000   0.170   0.277
## .p3T          0.322   0.057   5.689   0.000   0.322   0.427
## .p1S          0.035   0.019   1.878   0.060   0.035   0.124
## .p2S          0.202   0.040   5.075   0.000   0.202   0.421
## .p3S          0.262   0.056   4.714   0.000   0.262   0.349
## .p1V          0.268   0.043   6.199   0.000   0.268   0.684
## .p2V          0.105   0.036   2.966   0.003   0.105   0.145
## .p3V          0.226   0.048   4.744   0.000   0.226   0.274
## CTR          0.233   0.036   6.481   0.000   1.000   1.000
## TrP          0.478   0.090   5.316   0.000   1.000   1.000
## SRP          0.247   0.047   5.291   0.000   1.000   1.000
## Val          0.124   0.046   2.700   0.007   1.000   1.000

msmt_fit_pEsts <- lavaan::parameterEstimates(msmt_fit, boot.ci.type = "bca.simple",
                                              standardized = TRUE)
# msmt_fit_pEsts #To reduce redundancy in the book, I did not print
# the parameter estimates. Their object is used in exporting a .csv
# file.

```

Below is script that will export the global fit indices (via `tidySEM::table_fit`) and the parameter

estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```
# global fit indices
msmt_globalfit <- tidySEM::table_fit(msmt_fit)
write.csv(msmt_globalfit, file = "msmt_globalfit.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(msmt_fit_pEsts, file = "msmt_fit_pEsts.csv")
```

Here is how I wrote up the results:

Analyzing our proposed parallel mediation followed the two-step procedure of first establishing a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index ($CFI \geq 0.95$), root-mean-square error of approximation ($RMSEA \leq 0.06$), and the standard root-mean-square residual ($SRMR \leq 0.08$). To establish acceptable fit, we used $CFI \geq 0.90$, $RMSEA \leq 0.10$, and $SRMR \leq 0.10$ [Weston and Gore, 2006].

We evaluated the measurement model by following recommendations by Little et al. [Little et al., 2002, 2013]. The three course evaluation scales representing traditional pedagogy (5 items), socially responsive pedagogy (4 items), and perceived value to the student (3) were each represented by a combination of parcels and single items that created just-identified factors (i.e., the five items of traditional pedagogy were randomly assigned across three parcels and mean scores were used). For the centering variable – a single item indicator – we constrained the variance to zero.

With the exception of the statistically significant chi-square, fit statistics evidenced a mix of good and acceptable thresholds ($\chi^2(30) = 51.639, p = 0.004, CFI = 0.958, RMSEA = 0.093, 90CI[0.047, 0.134], SRMR = 0.052$). Thus, we proceeded to testing the structural model. The strong, statistically significant, and properly balanced factor loadings are presented in Table 1.

Table 1

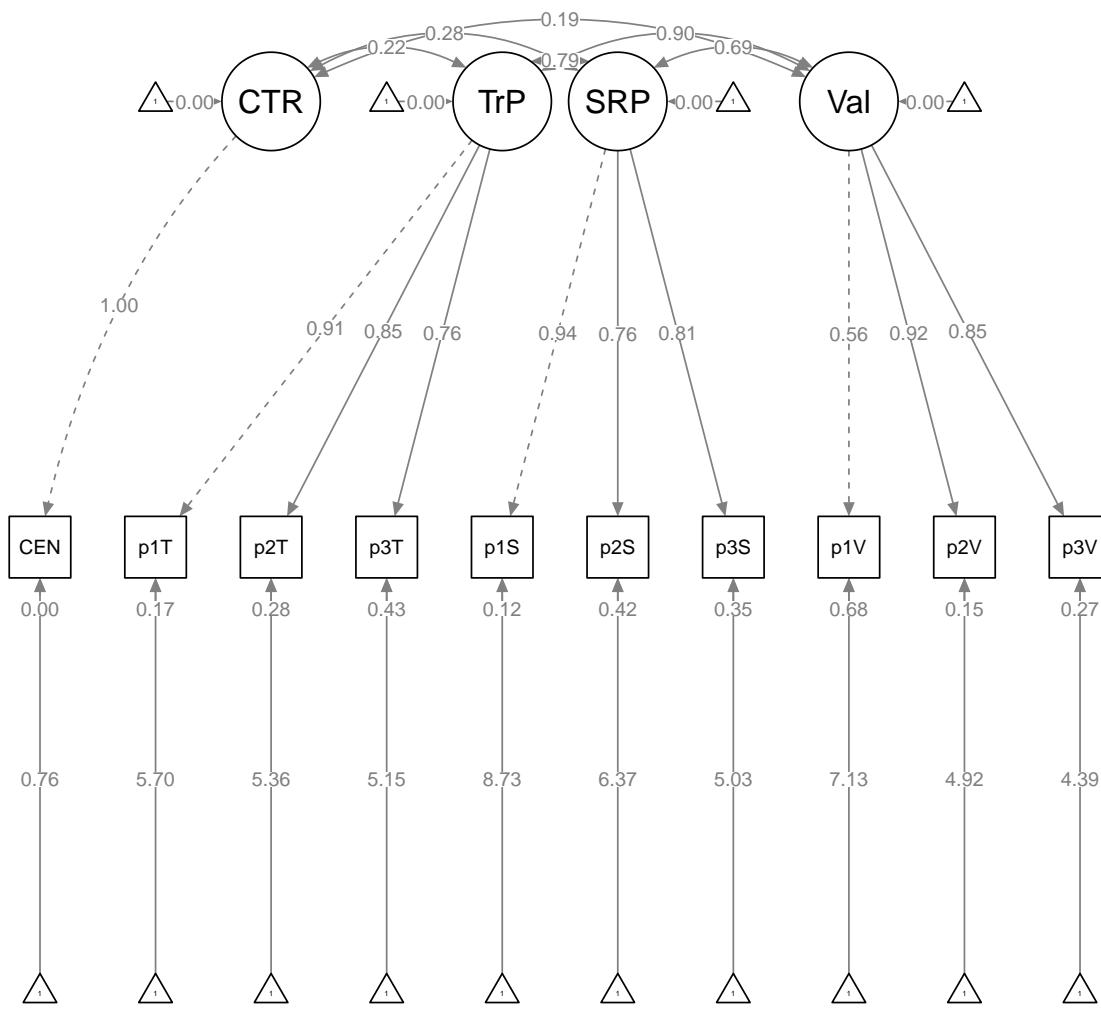
|Factor Loadings for the Measurement Model |:—————|

Latent variable and indicator	est	SE	p	est_std
Traditional Pedagogy				
Parcel 1	1.000	0.000		0.912
Parcel 2	0.963	0.090	<0.001	0.850
Parcel 3	0.952	0.108	<0.001	0.757
Socially Responsive Pedagogy				

Parcel 1	1.000	0.000		0.936
Parcel 2	1.061	0.134	<0.001	0.761
Parcel 3	1.408	0.179	<0.001	0.807
Perceived Value to the Student				
Item 1	1.000	0.000		0.562
Item 2	2.238	0.399	<0.001	0.925
Item 3	2.200	0.400	<0.001	0.852
CENTERING	1.000	0.000		1.000

Although it likely would not appear in an article (no space), here is a figure of my measurement model. We can use it to clarify our conceptual understanding of what we specified and check our work.

```
semPlot::semPaths(msmt_fit, what = "col", whatLabels = "stand", sizeMan = 5,
  node.width = 1, edge.label.cex = 0.75, style = "lisrel", mar = c(2,
  2, 2, 2))
```



Specify and evaluate a *structural* model

As a reminder, I am hypothesizing a *parallel mediation* where the perceived course value to the students is predicted by intentional recentering through their assessments of traditional and socially responsive pedagogy.

X = Centering: explicit recentering (0 = precentered; 1 = recentered) M1 = TradPed: traditional pedagogy (continuously scaled with higher scores being more favorable) M2 = SRPed: socially responsive pedagogy (continuously scaled with higher scores being more favorable) Y = Valued: valued by me (continuously scaled with higher scores being more favorable)

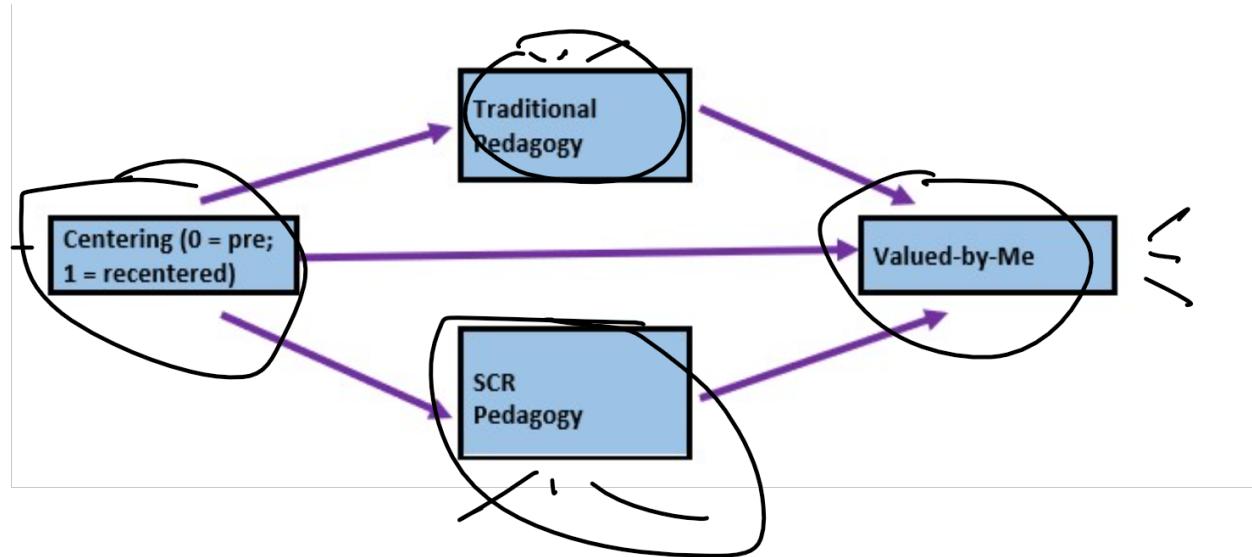


Figure 11.5: An image of the parallel mediation model for the homeworked example

For the purpose of this exercise, the structural model should be over-identified, that is, should have positive degrees of freedom. How many degrees of freedom does your structural model have?

**Knowns::: $\frac{k(k+1)}{2}$ where k is the number of constructs

```
(4*(4+1))/2
```

```
## [1] 10
```

There are 10 knowns.

Unknowns:

- Exogenous (predictor) variables (1 variance estimated for each): we have 1 (CTR)
- Endogenous (predicted) variables (1 disturbance variance for each): we have 3 (TrP, SRP, Val)
- Correlations between variables (1 covariance for each pairing): we have 0
- Regression paths (arrows linking exogenous variables to endogenous variables): we have 5

With 10 knowns and 9 unknowns, we have 1 degree of freedom in the structural portion of the model. This is an *over-identified* model.

```
ReC_struct_mod1 <- "
#measurement model
CTR =~ CEN #this is a single item indicator, I had to add code below to set the variance
TrP =~ p1T + p2T + p3T
SRP =~ p1S + p2S + p3S
Val =~ p1V + p2V + p3V

# Variance of the single item indicator
CTR ~~ 0*CEN

#structural model
Val ~ b1*TrP + b2*SRP + c_p*CTR
TrP ~ a1*CTR
SRP ~ a2*CTR

indirect1 := a1 * b1
indirect2 := a2 * b2
contrast := indirect1 - indirect2
total_indirects := indirect1 + indirect2
total_c     := c_p + (indirect1) + (indirect2)
direct := c_p

"
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
ReC_struct_fit1 <- lavaan::sem(ReC_struct_mod1, data = babydf, missing = "fiml",
    fixed.x = FALSE)
ReC_struct_summary1 <- lavaan::summary(ReC_struct_fit1, fit.measures = TRUE,
    standardized = TRUE, rsq = TRUE)
ReC_struct_pEsts1 <- lavaan::parameterEstimates(ReC_struct_fit1, boot.ci.type = "bca.simple",
    standardized = TRUE)
ReC_struct_summary1

## lavaan 0.6.16 ended normally after 72 iterations
##
##           Estimator          ML
##           Optimization method   NLMINB
##           Number of model parameters      34
##           Number of observations        84
##           Number of missing patterns       5
##
## Model Test User Model:
##
##           Test statistic      105.390
```

```

## Degrees of freedom                      31
## P-value (Chi-square)                   0.000
##
## Model Test Baseline Model:
##
## Test statistic                         561.645
## Degrees of freedom                     45
## P-value                                0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI)           0.856
## Tucker-Lewis Index (TLI)              0.791
##
## Robust Comparative Fit Index (CFI)    0.858
## Robust Tucker-Lewis Index (TLI)       0.794
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)        -662.347
## Loglikelihood unrestricted model (H1) -609.652
##
## Akaike (AIC)                          1392.694
## Bayesian (BIC)                        1475.342
## Sample-size adjusted Bayesian (SABIC) 1368.088
##
## Root Mean Square Error of Approximation:
##
## RMSEA                               0.169
## 90 Percent confidence interval - lower 0.134
## 90 Percent confidence interval - upper 0.205
## P-value H_0: RMSEA <= 0.050          0.000
## P-value H_0: RMSEA >= 0.080          1.000
##
## Robust RMSEA                         0.173
## 90 Percent confidence interval - lower 0.136
## 90 Percent confidence interval - upper 0.210
## P-value H_0: Robust RMSEA <= 0.050   0.000
## P-value H_0: Robust RMSEA >= 0.080   1.000
##
## Standardized Root Mean Square Residual:
##
## SRMR                                0.226
##
## Parameter Estimates:
##
## Standard errors                      Standard
## Information                           Observed

```

```

## Observed information based on Hessian
##
## Latent Variables:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   CTR     1.000
##   CEN     0.483  1.000
##   TrP     0.675  0.891
##   p1T     1.000
##   p2T     1.012  0.100 10.149  0.000  0.684  0.873
##   p3T     0.974  0.115  8.448  0.000  0.658  0.757
##   SRP     0.467  0.879
##   p1S     1.000
##   p2S     1.160  0.148  7.851  0.000  0.541  0.780
##   p3S     1.595  0.205  7.798  0.000  0.744  0.863
##   Val     0.335  0.544
##   p1V     1.000
##   p2V     2.220  0.394  5.633  0.000  0.743  0.913
##   p3V     2.197  0.398  5.527  0.000  0.736  0.842
##
## Regressions:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   Val ~
##   TrP (b1)    0.439  0.092  4.763  0.000  0.885  0.885
##   SRP (b2)    0.087  0.079  1.091  0.275  0.121  0.121
##   CTR (c_p)   -0.029  0.053 -0.543  0.587 -0.042 -0.042
##   TrP ~
##   CTR (a1)    0.319  0.158  2.017  0.044  0.228  0.228
##   SRP ~
##   CTR (a2)    0.311  0.109  2.865  0.004  0.321  0.321
##
## Covariances:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   CTR ~~
##   .CEN     0.000
##   .CTR     0.000  NaN
##
## Intercepts:
##          Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   .CEN    0.369  0.053  7.009  0.000  0.369  0.765
##   .p1T    4.321  0.083 52.220  0.000  4.321  5.698
##   .p2T    4.196  0.085 49.115  0.000  4.196  5.359
##   .p3T    4.478  0.095 47.068  0.000  4.478  5.151
##   .p1S    4.637  0.058 80.049  0.000  4.637  8.734
##   .p2S    4.413  0.076 58.132  0.000  4.413  6.363
##   .p3S    4.359  0.098 44.348  0.000  4.359  5.051
##   .p1V    4.464  0.067 66.486  0.000  4.464  7.254
##   .p2V    4.190  0.089 47.193  0.000  4.190  5.149
##   .p3V    3.986  0.096 41.739  0.000  3.986  4.564
##   CTR     0.000  0.000  0.000  0.000

```

```

## .TrP           0.000           0.000           0.000
## .SRP           0.000           0.000           0.000
## .Val           0.000           0.000           0.000
##
## Variances:
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CEN          0.000    0.000 0.000   0.000 0.000 0.000
## .p1T          0.119    0.032 3.701   0.000 0.119 0.207
## .p2T          0.146    0.036 4.075   0.000 0.146 0.237
## .p3T          0.323    0.058 5.539   0.000 0.323 0.427
## .p1S          0.064    0.022 2.961   0.003 0.064 0.227
## .p2S          0.188    0.039 4.816   0.000 0.188 0.391
## .p3S          0.191    0.057 3.321   0.001 0.191 0.256
## .p1V          0.267    0.043 6.193   0.000 0.267 0.704
## .p2V          0.110    0.036 3.076   0.002 0.110 0.166
## .p3V          0.222    0.047 4.682   0.000 0.222 0.290
## CTR          0.233    0.036 6.481   0.000 1.000 1.000
## .TrP          0.432    0.087 4.992   0.000 0.948 0.948
## .SRP          0.195    0.043 4.498   0.000 0.897 0.897
## .Val          0.023    0.011 2.141   0.032 0.205 0.205
##
## R-Square:
##             Estimate
## CEN          1.000
## p1T          0.793
## p2T          0.763
## p3T          0.573
## p1S          0.773
## p2S          0.609
## p3S          0.744
## p1V          0.296
## p2V          0.834
## p3V          0.710
## TrP          0.052
## SRP          0.103
## Val          0.795
##
## Defined Parameters:
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## indirect1     0.140    0.075 1.869   0.062 0.202 0.202
## indirect2     0.027    0.026 1.023   0.306 0.039 0.039
## contrast      0.113    0.082 1.386   0.166 0.163 0.163
## total_indircts 0.167    0.077 2.163   0.031 0.241 0.241
## total_c       0.138    0.083 1.670   0.095 0.199 0.199
## direct        -0.029   0.053 -0.543   0.587 -0.042 -0.042

```

```
# ReC_struct_pEsts1 #although creating the object is useful to export
# as a .csv I didn't ask it to print into the book
```

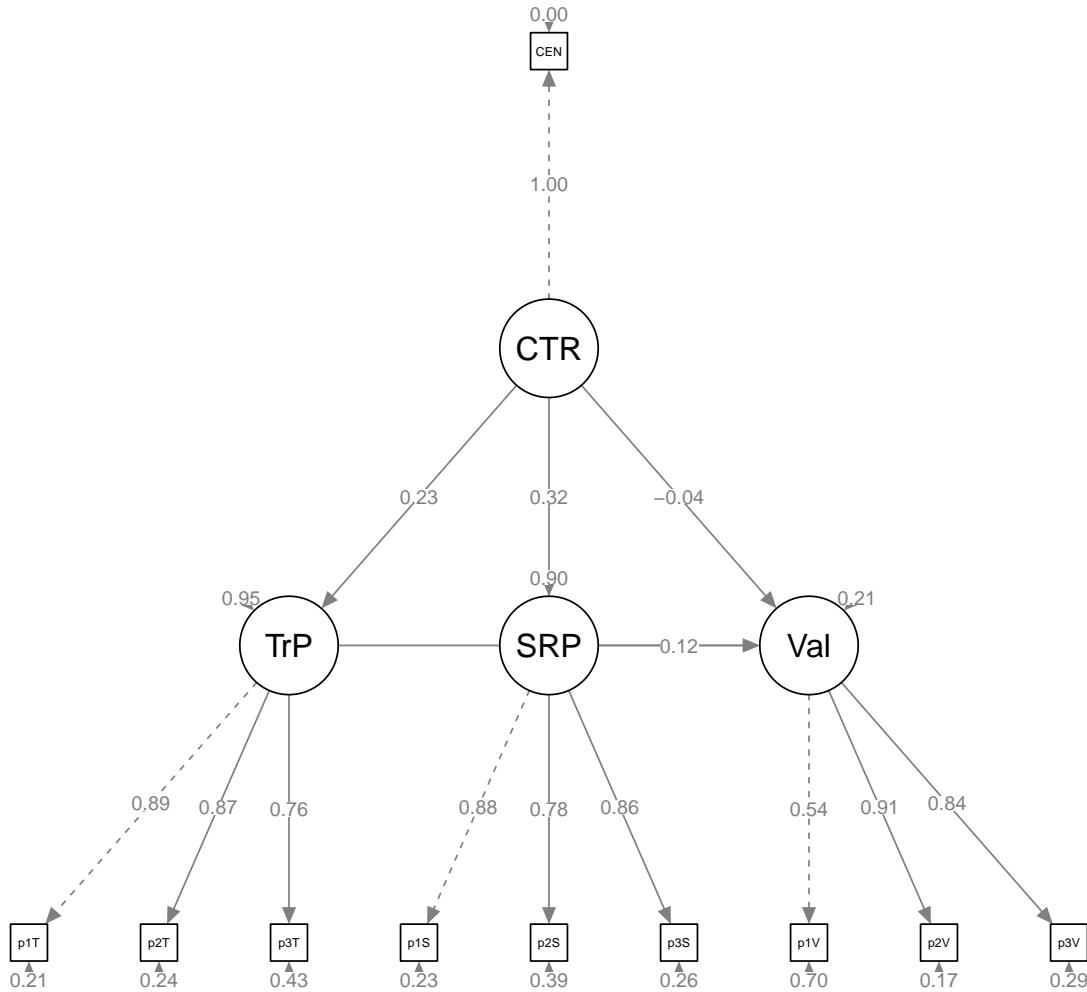
$(\chi^2(31) = 105.390, p < 0.001, CFI = 0.856, RMSEA = 0.169, 90CI[0.134, 0.205], SRMR = 0.226)$.

Below is script that will export the global fit indices (via *tidySEM::table_fit*) and the parameter estimates (e.g., factor loadings, structural regression weights, and parameters we requested such as the indirect effect) to .csv files that you can manipulate outside of R.

```
# global fit indices
ReC_globalfit1 <- tidySEM::table_fit(ReC_struct_fit1)
write.csv(ReC_globalfit1, file = "ReC_globalfit1.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(ReC_struct_pEsts1, file = "ReC_struct_pEsts1.csv")
```

Let's work up a figure

```
plot_ReC_struct1 <- semPlot::semPaths(ReC_struct_fit1, what = "col", whatLabels = "stand",
sizeMan = 3, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
mar = c(2, 2, 2, 2), structural = FALSE, curve = FALSE, intercepts = FALSE)
```

Grid for Plotting `semplot::semplot`

(1,1) empty	(1,2) TrP	(1,3) empty
(2,1) CTR	(2,2) empty	(2,3) Val
(3,1) empty	(3,2) SRP	(3,3) empty

We place these values along with the names of our latent variables in to the `semtools::layout_matrix` function.

```
# IMPORTANT: Must use the node names (take directly from the SemPlot)
# assigned by SemPlot You can change them as the last thing
m1_msmt <- semtools::layout_matrix(CTR = c(2, 1), TrP = c(1, 2), SRP = c(3,
2), Val = c(2, 3))
```

Next we provide instruction on the direction (up, down, left, right) we want the indicator/observed variables to face. We identify the direction by the location of each of our latent variables. For example, in the code below we want the indicators for the REM variable (2,1) to face left.

```
# tell where you want the indicators to face
m1_point_to <- semptools::layout_matrix(left = c(2, 1), up = c(1, 2), down = c(3,
2), right = c(2, 3))
```

The next two sets of code work together to specify the order of the observed variables for each factor. in the top set of code the variable names indicate the order in which they will appear (i.e., p1R, p2R, p3R). In the second set of code, the listing the variable name three times (i.e., REM, REM, REM) serves as a placeholder for each of the indicators.

It is critical to note that we need to use the abbreviated variable names assigned by *semTools::semPaths* and not necessarily the names that are in the dataframe.

```
# the next two codes -- indicator_order and indicator_factor are
# paired together, they specify the order of observed variables for
# each factor
m1_indicator_order <- c("CEN", "p1T", "p2T", "p3T", "p1S", "p2S", "p3S",
 "p1V", "p2V", "p3V")

m1_indicator_factor <- c("CTR", "TrP", "TrP", "TrP", "SRP", "SRP", "SRP",
 "Val", "Val", "Val")
```

The next two sets of codes provide some guidance about how far away the indicator (square/rectangular) variables should be away from the latent (oval/circular) variables. Subsequently, the next set of values indicate how far away each of the indicator (square/rectangular) variables should be spread apart.

```
# next set of code pushes the indicator variables away from the
# factor
m1_indicator_push <- c(CTR = 1, TrP = 1, SRP = 1, Val = 1)

# spreading the boxes away from each other
m1_indicator_spread <- c(CTR = 0.5, TrP = 2.5, SRP = 2.5, Val = 1)
```

Finally, we can feed all of the objects that whole these instructions into the *sempool::sem_set_layout* function. If desired, we can use the *sempool::change_node_label* function to rename the latent variables. Again, make sure to use the variable names that *semPlot::semPaths* has assigned.

```
plot1 <- sempool::set_sem_layout(plot_ReC_struct1, indicator_order = m1_indicator_order,
 indicator_factor = m1_indicator_factor, factor_layout = m1_msmt, factor_point_to = m1_point_to,
 indicator_push = m1_indicator_push, indicator_spread = m1_indicator_spread)

# changing node labels plot1 <- sempool::change_node_label(plot1,
# c(CTR = 'CTRing', TrP = 'TradPed', SRP = 'SRPed', Val = 'Valued'),
# label.cex = 1.1)

# adding stars to indicate significant paths
```

```
plot1 <- semtools::mark_sig(plot1, ReC_struct_fit1)
plot(plot1)
```

If we table the results, here's what we have:

Table 2

Model Coefficients Assessing the Effect of Perceived Value from Recentering through Socially Responsive and Traditional Pedagogies

Predictor	B	SE _B	p	β	R ²
Traditional Pedagogy (M1)					.05
Centering (a_1)	0.319	0.158	0.044	0.228	
Socially Responsive Pedagogy (M2)					.10
Centering (a_2)	0.311	0.109	0.004	0.321	
Perceived Value (DV)					.80
Traditional Pedagogy (b_1)	0.595	0.050	0.000	0.865	
Socially Responsive Pedagogy (b_2)	0.087	0.079	0.275	0.121	
Centering (c')	-0.029	0.053	0.587	-0.042	

Effects	B	SE _B	p	95% CI
Indirect($a_1 * b_1$)	0.140	0.075	0.062	-0.007, 0.287
Indirect($a_2 * b_2$)	0.027	0.026	0.306	0.039
Contrast	0.113	0.082	0.166	0.163
Total indirects	0.167	0.077	0.031	0.241
Total effect	0.138	0.083	0.095	0.199

Note. The significance of the indirect effects was calculated with bootstrapped, bias-corrected, confidence intervals (.95).

Our structural model was a parallel mediation, predicting perceived value to the student directly from centering and indirectly through traditional and socially responsive pedagogy. Results of the global fit indices all fell below the thresholds of acceptability ($\chi^2(31) = 105.390, p < 0.001, CFI = 0.856, RMSEA = 0.169, 90CI[0.134, 0.205], SRMR = 0.226$). As shown in Table 2, although the model accounted for 95% of the variance in perceived value, neither of the indirect effects were statistically significant. Only traditional and socially responsive pedagogy appeared to have significant influence on perceived value. Thus, we considered the possibility of forward searching to build the model.

Use modification indices to add at least one path or covariance

Normally, we would use the `lavaan::modindices()` function to retrieve the modification indices. Unfortunately, it is not working and I cannot understand why. This error does not make sense to me. I wonder if this error and the inability to plot the result is related to the dichotomous predictor variable? Who knows. If anyone discovers a solution, please let me know!

```
lavaan::modindices(ReC_struct_fit1, sort = TRUE, minimum.value = 4)
```

I planned this problem knowing that the the only logical path would be to add a path from traditional pedagogy to socially responsive pedagogy, turning this into a serial mediation. Let's do that!

Before even conducting the statistic, adding this path will make the structural model *just-identified*. As such, it will have the identical strong fit of the measurement model. This means we probably should not favor this model if the serial indirect effect is not statistically significant.

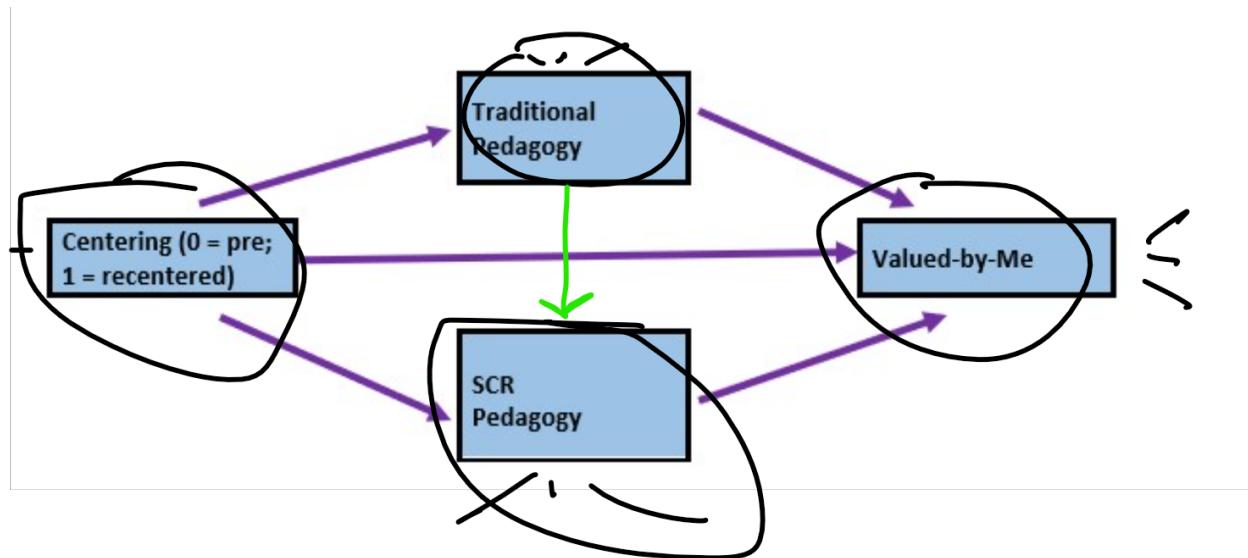


Figure 11.6: An image of the parallel mediation model for the homeworked exampl.

Conduct a formal comparison of *global fit* between the original and respecified model

```
ReC_struct_mod2 <- "
#measurement model
CTR =~ CEN #this is a single item indicator, I had to add code below to set the variance
TrP =~ p1T + p2T + p3T
SRP =~ p1S + p2S + p3S
Val =~ p1V + p2V + p3V

# Variance of the single item indicator
```

```

CTR ~~ 0*CEN

#structural model
Val ~ b1*TrP + b2*SRP + c_p*CTR
TrP ~ a1*CTR
SRP ~ a2*CTR + d1*TrP

indirect1 := a1 * b1
indirect2 := a2 * b2
indirect3 := a1 * d1 * b2
contrast1 := indirect1 - indirect2
contrast2 := indirect1 - indirect3
contrast3 := indirect2 - indirect3
total_indirects := indirect1 + indirect2 + indirect3
total_c      := c_p + (indirect1) + (indirect2) + (indirect3)
direct := c_p

"
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
ReC_struct_fit2 <- lavaan::sem(ReC_struct_mod2, data = babydf, missing = "fiml")
ReC_struct_summary2 <- lavaan::summary(ReC_struct_fit2, fit.measures = TRUE,
standardized = TRUE, rsq = TRUE)
ReC_struct_pEsts2 <- lavaan::parameterEstimates(ReC_struct_fit2, boot.ci.type = "bca.simple",
standardized = TRUE)
ReC_struct_summary2

## lavaan 0.6.16 ended normally after 75 iterations
##
##    Estimator                      ML
## Optimization method            NLMINB
## Number of model parameters    35
## 
##    Number of observations        84
##    Number of missing patterns     5
## 
## Model Test User Model:
##
##    Test statistic              51.639
##    Degrees of freedom           30
##    P-value (Chi-square)         0.008
## 
## Model Test Baseline Model:
##
##    Test statistic              561.645
##    Degrees of freedom            45
##    P-value                       0.000

```

```

## 
## User Model versus Baseline Model:
## 
## Comparative Fit Index (CFI)          0.958
## Tucker-Lewis Index (TLI)            0.937
## 
## Robust Comparative Fit Index (CFI)    0.951
## Robust Tucker-Lewis Index (TLI)      0.927
## 
## Loglikelihood and Information Criteria:
## 
## Loglikelihood user model (H0)        -635.472
## Loglikelihood unrestricted model (H1) -609.652
## 
## Akaike (AIC)                         1340.943
## Bayesian (BIC)                        1426.022
## Sample-size adjusted Bayesian (SABIC) 1315.614
## 
## Root Mean Square Error of Approximation:
## 
## RMSEA                                0.093
## 90 Percent confidence interval - lower 0.047
## 90 Percent confidence interval - upper 0.134
## P-value H_0: RMSEA <= 0.050          0.060
## P-value H_0: RMSEA >= 0.080          0.709
## 
## Robust RMSEA                           0.103
## 90 Percent confidence interval - lower 0.058
## 90 Percent confidence interval - upper 0.146
## P-value H_0: Robust RMSEA <= 0.050   0.030
## P-value H_0: Robust RMSEA >= 0.080   0.821
## 
## Standardized Root Mean Square Residual:
## 
## SRMR                                 0.052
## 
## Parameter Estimates:
## 
## Standard errors                      Standard
## Information                            Observed
## Observed information based on        Hessian
## 
## Latent Variables:
## 
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR =~
##     CEN           1.000
## TrP =~
##     p1T           1.000

```

```

##      p2T          0.963    0.090   10.727   0.000    0.666    0.850
##      p3T          0.952    0.108    8.848   0.000    0.658    0.757
## SRP =~
##      p1S          1.000
##      p2S          1.061    0.134   7.944   0.000    0.527    0.761
##      p3S          1.408    0.179   7.852   0.000    0.700    0.807
## Val =~
##      p1V          1.000
##      p2V          2.238    0.399   5.608   0.000    0.788    0.925
##      p3V          2.200    0.400   5.495   0.000    0.774    0.852
##
## Regressions:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## Val ~
## TrP     (b1)    0.487    0.113   4.308   0.000    0.957    0.957
## SRP     (b2)   -0.047    0.104  -0.456   0.649   -0.067  -0.067
## CTR     (c_p)   -0.007    0.053  -0.128   0.898  -0.009  -0.009
## TrP ~
## CTR     (a1)    0.322    0.161   1.996   0.046    0.225    0.225
## SRP ~
## CTR     (a2)    0.113    0.087   1.291   0.197    0.109    0.109
## TrP     (d1)    0.554    0.070   7.963   0.000    0.770    0.770
##
## Covariances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## CTR ~~
## .CEN      0.000
## 
## Intercepts:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CEN      0.369    0.053   7.009   0.000    0.369    0.765
## .p1T      4.321    0.083  52.220   0.000    4.321    5.698
## .p2T      4.196    0.085  49.115   0.000    4.196    5.359
## .p3T      4.478    0.095  47.081   0.000    4.478    5.152
## .p1S      4.637    0.058  80.049   0.000    4.637    8.734
## .p2S      4.411    0.076  58.169   0.000    4.411    6.367
## .p3S      4.360    0.099  43.986   0.000    4.360    5.029
## .p1V      4.464    0.068  65.385   0.000    4.464    7.134
## .p2V      4.190    0.093  45.087   0.000    4.190    4.919
## .p3V      3.986    0.099  40.142   0.000    3.986    4.389
## CTR      0.000
## TrP      0.000
## SRP      0.000
## Val      0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .CEN      0.000

```

```

##   .p1T          0.097  0.026  3.798  0.000  0.097  0.169
##   .p2T          0.170  0.035  4.918  0.000  0.170  0.277
##   .p3T          0.322  0.057  5.689  0.000  0.322  0.427
##   .p1S          0.035  0.019  1.878  0.060  0.035  0.124
##   .p2S          0.202  0.040  5.075  0.000  0.202  0.421
##   .p3S          0.262  0.056  4.714  0.000  0.262  0.349
##   .p1V          0.268  0.043  6.199  0.000  0.268  0.684
##   .p2V          0.105  0.036  2.966  0.003  0.105  0.145
##   .p3V          0.226  0.048  4.744  0.000  0.226  0.274
##   CTR           0.233  0.036  6.481  0.000  1.000  1.000
##   .TrP           0.454  0.086  5.280  0.000  0.949  0.949
##   .SRP           0.088  0.023  3.838  0.000  0.357  0.357
##   .Val           0.023  0.011  2.087  0.037  0.184  0.184
##
## R-Square:
##                               Estimate
##   CEN             1.000
##   p1T            0.831
##   p2T            0.723
##   p3T            0.573
##   p1S            0.876
##   p2S            0.579
##   p3S            0.651
##   p1V            0.316
##   p2V            0.855
##   p3V            0.726
##   TrP            0.051
##   SRP            0.643
##   Val            0.816
##
## Defined Parameters:
##                               Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   indirect1        0.157  0.086   1.817   0.069   0.215  0.215
##   indirect2       -0.005  0.012  -0.439   0.661  -0.007 -0.007
##   indirect3       -0.008  0.019  -0.439   0.661  -0.012 -0.012
##   contrast1        0.162  0.090   1.806   0.071   0.223  0.223
##   contrast2        0.165  0.097   1.708   0.088   0.227  0.227
##   contrast3        0.003  0.010   0.318   0.750   0.004  0.004
##   total_indircts   0.143  0.080   1.790   0.073   0.196  0.196
##   total_c          0.136  0.086   1.584   0.113   0.187  0.187
##   direct          -0.007  0.053  -0.128   0.898  -0.009 -0.009

# ReC_struct_pEsts1 #although creating the object is useful to export
# as a .csv I didn't ask it to print into the book

```

```

# global fit indices
ReC_globalfit2 <- tidySEM::table_fit(ReC_struct_fit2)

```

```
write.csv(ReC_globalfit2, file = "ReC_globalfit2.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(ReC_struct_pEsts2, file = "ReC_struct_pEsts2.csv")
```

Here is the global fit indices: $(\chi^2(30) = 51.639, p = 0.004, CFI = 0.958, RMSEA = 0.093, 90CI[0.047, 0.134], SRMR = 0.052)$. As I noted before we ran it, they are identical to those of the just-identified measurement model.

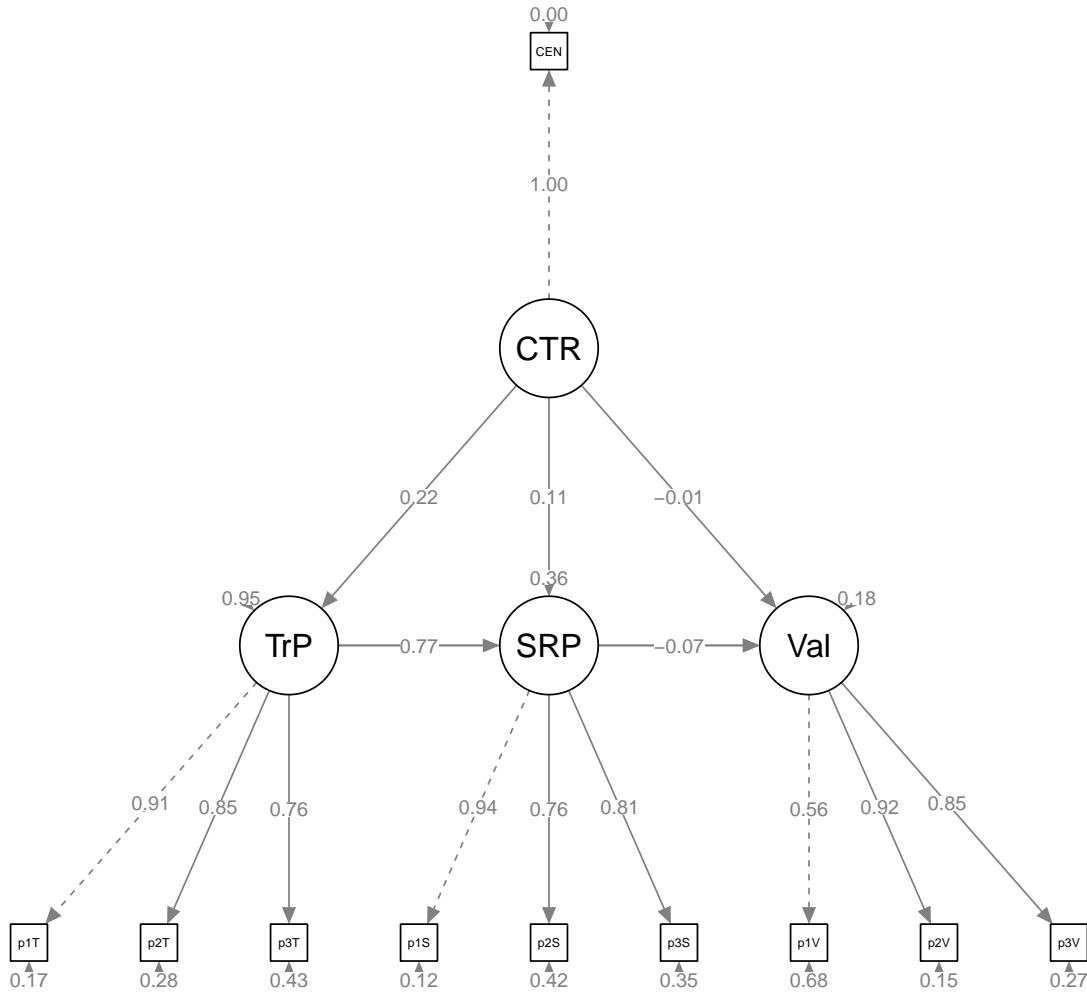
We can compare the difference between the originally hypothesized fit and this fit.

```
lavaan::lavTestLRT(ReC_struct_fit1, ReC_struct_fit2)
```

```
##
## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq Chisq diff    RMSEA Df diff
## ReC_struct_fit2 30 1340.9 1426.0  51.639
## ReC_struct_fit1 31 1392.7 1475.3 105.390      53.751 0.79246      1
##                      Pr(>Chisq)
## ReC_struct_fit2
## ReC_struct_fit1 0.000000000002276 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Given that these models are nested (i.e., same variables, same sample, only the paths change), we can use the $(\Delta\chi^2(1) = 53.751, p < 0.001)$ to see that there is a statistically significant difference favoring the model with the serial mediation. If we examine the regression weights and parameters, we see that none of the indirect effects, including the serial mediation ($B = -0.008, SE = 0.019, \beta = -0.012, p = 0.661, 95CI[-0.046, 0.029]$). are non-significant. Thus, I would favor retaining the originally hypothesized model.

```
plot_struct2 <- semPlot::semPaths(ReC_struct_fit2, what = "col", whatLabels = "stand",
  sizeMan = 3, node.width = 1, edge.label.cex = 0.75, style = "lisrel",
  mar = c(2, 2, 2, 2), structural = FALSE, curve = FALSE, intercepts = FALSE)
```



Using the strength and significance of regression weights as a guide, trim at least path or covariance

Given my rejection of the model that resulted from adding the serial mediation, I will inspect the strength, direction, and significance of my regression weights to see about deleting a path.

This is tricky, the “most-nonsignificant” path is the direct path from centering to perceived value ($B = 0.029, p = 0.587$). That path, however, is involved in the indirect effects (and I would sue like the indirect effect through traditional pedagogy [$B = 0.140, p = 0.061, 95CI(-0.007, 0.287)$] to pass into statistical significance). The next most non-significant path is b_2 , from socially responsive pedagogy to perceived value ($B = 0.087, p = 0.275$)). Let’s trim it.

```

ReC_struct_mod3 <- "
  #measurement model
  CTR =~ CEN #this is a single item indicator, I had to add code below to set the variance
  TrP =~ p1T + p2T + p3T
  SRP =~ p1S + p2S + p3S
  "
  
```

```

Val =~ p1V + p2V + p3V

# Variance of the single item indicator
CTR ~~ 0*CEN

#structural model
Val ~ b1*TrP + c_p*CTR #trimmed + b2*SRP
TrP ~ a1*CTR
SRP ~ a2*CTR

indirect1 := a1 * b1
#indirect2 := a2 * b2 #trimmed because the b2 path was trimmed
#contrast := indirect1 - indirect2
#total_indirects := indirect1 + indirect2
total_c    := c_p + (indirect1) #trimmed + (indirect2)
direct := c_p

"
set.seed(230916) #needed for reproducibility especially when specifying bootstrapped confidence intervals
ReC_struct_fit3 <- lavaan::sem(ReC_struct_mod3, data = babydf, missing = "fiml",
fixed.x = FALSE)
ReC_struct_summary3 <- lavaan::summary(ReC_struct_fit3, fit.measures = TRUE,
standardized = TRUE, rsq = TRUE)
ReC_struct_pEsts3 <- lavaan::parameterEstimates(ReC_struct_fit3, boot.ci.type = "bca.simple",
standardized = TRUE)
ReC_struct_summary3

## lavaan 0.6.16 ended normally after 71 iterations
##
##   Estimator                      ML
##   Optimization method            NLMINB
##   Number of model parameters    34
##   Number of observations        84
##   Number of missing patterns     5
##
## Model Test User Model:
##
##   Test statistic                 105.390
##   Degrees of freedom              31
##   P-value (Chi-square)           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                  561.645
##   Degrees of freedom                45
##   P-value                           0.000

```

```

## 
## User Model versus Baseline Model:
## 
##    Comparative Fit Index (CFI)          0.856
##    Tucker-Lewis Index (TLI)            0.791
## 
##    Robust Comparative Fit Index (CFI)    0.858
##    Robust Tucker-Lewis Index (TLI)      0.794
## 
## Loglikelihood and Information Criteria:
## 
##    Loglikelihood user model (H0)        -662.347
##    Loglikelihood unrestricted model (H1) -609.652
## 
##    Akaike (AIC)                      1392.694
##    Bayesian (BIC)                    1475.342
##    Sample-size adjusted Bayesian (SABIC) 1368.088
## 
## Root Mean Square Error of Approximation:
## 
##    RMSEA                           0.169
##    90 Percent confidence interval - lower 0.134
##    90 Percent confidence interval - upper 0.205
##    P-value H_0: RMSEA <= 0.050       0.000
##    P-value H_0: RMSEA >= 0.080       1.000
## 
##    Robust RMSEA                     0.173
##    90 Percent confidence interval - lower 0.136
##    90 Percent confidence interval - upper 0.210
##    P-value H_0: Robust RMSEA <= 0.050 0.000
##    P-value H_0: Robust RMSEA >= 0.080 1.000
## 
## Standardized Root Mean Square Residual:
## 
##    SRMR                            0.226
## 
## Parameter Estimates:
## 
##    Standard errors                  Standard
##    Information                      Observed
##    Observed information based on   Hessian
## 
## Latent Variables:
## 
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    CTR =~
##    CEN          1.000
##    TrP =~
##    p1T          1.000

```

```

##    p2T          1.012   0.100  10.149   0.000   0.684   0.873
##    p3T          0.974   0.115   8.448   0.000   0.658   0.757
##    SRP =~
##    p1S          1.000
##    p2S          1.160   0.148   7.851   0.000   0.541   0.780
##    p3S          1.595   0.205   7.798   0.000   0.744   0.863
##    Val =~
##    p1V          1.000
##    p2V          2.220   0.394   5.633   0.000   0.743   0.913
##    p3V          2.197   0.398   5.527   0.000   0.736   0.842
##
##    ## Regressions:
##    Val ~
##    TrP      (b1)  0.439   0.092   4.763   0.000   0.885   0.885
##    CTR      (c_p) -0.002   0.053  -0.037   0.970  -0.003  -0.003
##    TrP ~
##    CTR      (a1)  0.319   0.158   2.017   0.044   0.228   0.228
##    SRP ~
##    CTR      (a2)  0.311   0.109   2.865   0.004   0.321   0.321
##
##    ## Covariances:
##    CTR ~~
##    .CEN      0.000
##    .SRP ~~
##    .Val      0.017   0.016   1.075   0.282   0.244   0.244
##
##    ## Intercepts:
##    .CEN      0.369   0.053   7.009   0.000   0.369   0.765
##    .p1T      4.321   0.083  52.220   0.000   4.321   5.698
##    .p2T      4.196   0.085  49.115   0.000   4.196   5.359
##    .p3T      4.478   0.095  47.068   0.000   4.478   5.151
##    .p1S      4.637   0.058  80.049   0.000   4.637   8.734
##    .p2S      4.413   0.076  58.132   0.000   4.413   6.363
##    .p3S      4.359   0.098  44.348   0.000   4.359   5.051
##    .p1V      4.464   0.067  66.486   0.000   4.464   7.254
##    .p2V      4.190   0.089  47.193   0.000   4.190   5.149
##    .p3V      3.986   0.096  41.739   0.000   3.986   4.564
##    CTR      0.000
##    .TrP      0.000
##    .SRP      0.000
##    .Val      0.000
##
##    ## Variances:
##    .CEN      Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    .CEN      0.000   0.000   0.000   1.000   0.000   0.000

```

```

##   .p1T          0.119    0.032    3.701    0.000    0.119    0.207
##   .p2T          0.146    0.036    4.075    0.000    0.146    0.237
##   .p3T          0.323    0.058    5.539    0.000    0.323    0.427
##   .p1S          0.064    0.022    2.961    0.003    0.064    0.227
##   .p2S          0.188    0.039    4.816    0.000    0.188    0.391
##   .p3S          0.191    0.057    3.321    0.001    0.191    0.256
##   .p1V          0.267    0.043    6.193    0.000    0.267    0.704
##   .p2V          0.110    0.036    3.076    0.002    0.110    0.166
##   .p3V          0.222    0.047    4.682    0.000    0.222    0.290
##   CTR          0.233    0.036    6.481    0.000    1.000    1.000
##   .TrP          0.432    0.087    4.992    0.000    0.948    0.948
##   .SRP          0.195    0.043    4.498    0.000    0.897    0.897
##   .Val          0.024    0.012    2.115    0.034    0.218    0.218
##
## R-Square:
##             Estimate
##   CEN          1.000
##   p1T          0.793
##   p2T          0.763
##   p3T          0.573
##   p1S          0.773
##   p2S          0.609
##   p3S          0.744
##   p1V          0.296
##   p2V          0.834
##   p3V          0.710
##   TrP          0.052
##   SRP          0.103
##   Val          0.782
##
## Defined Parameters:
##             Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   indirect1     0.140   0.075   1.869   0.062   0.202   0.202
##   total_c       0.138   0.083   1.670   0.095   0.199   0.199
##   direct        -0.002  0.053  -0.037  0.970  -0.003  -0.003

```

```

# ReC_struct_pEsts1 #although creating the object is useful to export
# as a .csv I didn't ask it to print into the book

```

Exporting results to .csv files:

```

# global fit indices
ReC_globalfit3 <- tidySEM::table_fit(ReC_struct_fit3)
write.csv(ReC_globalfit3, file = "ReC_globalfit3.csv")
# the code below writes the parameter estimates into a .csv file
write.csv(ReC_struct_pEsts3, file = "ReC_struct_pEsts3.csv")

```

Curiously, the global fit indices are identical to the originally hypothesized model. Given that we trimmed a path, I'm a little confused by this: ($\chi^2(31) = 105.390, p < 0.001, CFI = 0.856, RMSEA = 0.169, 90CI[0.134, 0.205, SRMR = 0.226]$). Additionally, the indirect effect through traditional pedagogy is not statistically significant \$(B = 0.140, SE = 0.075, p = 0.062, 95CI[-0.007, 0.287]).

Conduct a formal comparison of *global* fit between the original (or built) and trimmed model

None-the-less, I can formally compare the two tests: ($\chi^2(1) = 53.751, p < 0.001$)

```
lavaan:::lavTestLRT(ReC_struct_fit1, ReC_struct_fit3)

## Warning in lavaan:::lavTestLRT(ReC_struct_fit1, ReC_struct_fit3): lavaan
## WARNING: some models have the same degrees of freedom

## 
## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq      Chisq diff RMSEA Df diff
## ReC_struct_fit1 31 1392.7 1475.3 105.39
## ReC_struct_fit3 31 1392.7 1475.3 105.39 -0.00000000055586      0      0
##          Pr(>Chisq)
## ReC_struct_fit1
## ReC_struct_fit3
```

APA style results with table(s) and figure(s)

Analyzing our proposed parallel mediation followed the two-step procedure of first establishing a measurement model with acceptable fit to the data and then proceeding to test the structural model. Given that different researchers recommend somewhat differing thresholds to determine the adequacy of fit, We used the following as evidence of good fit: comparative fit index ($CFI \geq 0.95$), root-mean-square error of approximation ($RMSEA \leq 0.06$), and the standard root-mean-square residual ($SRMR \leq 0.08$). To establish acceptable fit, we used $CFI \geq 0.90$, $RMSEA \leq 0.10$, and $SRMR \leq 0.10$ [Weston and Gore, 2006].

We evaluated the measurement model by following recommendations by Little et al. [Little et al., 2002, 2013]. The three course evaluation scales representing traditional pedagogy (5 items), socially responsive pedagogy (4 items), and perceived value to the student (3) were each represented by a combination of parcels and single items that created just-identified factors (i.e., the five items of traditional pedagogy were randomly assigned across three parcels and mean scores were used). For the centering variable – a single item indicator – we constrained the variance to zero.

With the exception of the statistically significant chi-square, fit statistics evidenced a mix of good and acceptable thresholds ($\chi^2(30) = 51.639, p = 0.004, CFI = 0.958, RMSEA = 0.093, 90CI[0.047, 0.134], SRMR = 0.052$). Thus, we proceeded to testing the structural model. The strong, statistically significant, and properly balanced factor loadings are presented in Table 1.

Our structural model was a parallel mediation, predicting perceived value to the student directly from centering and indirectly through traditional and socially responsive pedagogy. Results of the global fit indices all fell below the thresholds of acceptability ($\chi^2(31) = 105.390, p < 0.001, CFI = 0.856, RMSEA = 0.169, 90CI[0.134, 0.205], SRMR = 0.226$). As shown in Table 2, although the model accounted for 95% of the variance in perceived value, neither of the indirect effects were statistically significant. Only traditional and socially responsive pedagogy appeared to have significant influence on perceived value. Thus, we considered both building and trimming approaches.

I would have liked to use modification indices. However, R errors prevented me from doing so. None-the-less, the only available path to add to the model was from traditional pedagogy to socially responsive pedagogy, turning the parallel mediation into a serial mediation. Given that these models are nested (i.e., same variables, same sample, only the paths change), it was possible to test for a statistically significant differences. Results ($\Delta\chi^2(1) = 53.751, p < 0.001$) indicated a statistically significant difference favoring the model with the serial mediation. Inspection of the regression weights and parameters, suggested that none of the indirect effects, including the serial mediation ($B = -0.008, SE = 0.019, \beta = -0.012, p = 0.661, 95CI[-0.046, 0.029]$). were non-significant. Thus, I would favored retaining the originally hypothesized model.

Next, I engaged in backward searching and trimmed the b_2 path from socially responsive pedagogy to perceived value. Because the global fit indices were identical to the originally hypothesized model ($\chi^2(31) = 105.390, p < 0.001, CFI = 0.856, RMSEA = 0.169, 90CI[0.134, 0.205], SRMR = 0.226$), it was not possible to conduct a ($\Delta\chi^2$). Additionally, the indirect effect through traditional pedagogy was not statistically significant ($B = 0.140, SE = 0.075, p = 0.062, 95CI[-0.007, 0.287]$). Thus, I retained and interpreted the originally hypothesized parallel mediation.

Explanation to grader

REFERENCES

Bibliography

FACT SHEET: Anti-Asian Prejudice March 2020, a. URL <https://www.csusb.edu/sites/default/files/FACT%20SHEET-%20Anti-Asian%20Hate%202020%203.2.21.pdf>.

SEM with Latent Variables (David A. Kenny), b. URL <https://davidakenny.net/cm/sem.htm>.

STOP AAPI HATE, c. URL <https://stopaapihate.org/>.

Hector Y. Adames, Nayeli Y. Chavez-Dueñas, and Maryam M. Jernigan. The fallacy of a raceless Latinidad: Action guidelines for centering Blackness in Latinx psychology. *Journal of Latinx Psychology*, 9(1):26–44, February 2021. ISSN 2578-8086. doi: 10.1037/lat0000179. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2020-69785-001&site=ehost-live>. Publisher: Educational Publishing Foundation.

Kelsey L. Autin, Megan E. Herdt, Roberto G. Garcia, and Gabriel N. Ezema. Basic Psychological Need Satisfaction, Autonomous Motivation, and Meaningful Work: A Self-Determination Theory Perspective. *Journal of Career Assessment*, 30(1):78–93, February 2022. ISSN 1069-0727, 1552-4590. doi: 10.1177/10690727211018647. URL <http://journals.sagepub.com/doi/10.1177/10690727211018647>.

Reuben M Baron and David A Kenny. The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations. *Journal of Personality and Social Psychology*, 51(6):1173–1182, 1986. doi: 0022-3514/86. URL <https://www.sesp.org/files/The%20Moderator-Baron.pdf>.

Kenneth A. Bollen and Rick H. Hoyle. Perceived cohesion: A conceptual and empirical examination. *Social Forces*, 69(2):479–504, December 1990. ISSN 0037-7732. doi: 10.2307/2579670. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=1991-21226-001&site=ehost-live>. Publisher: University of North Carolina Press.

Barbara M. Byrne. Application 4: Testing the Validity of a Causal Structure. Chapter 6. In *Structural Equation Modeling with AMOS: Basic Concepts, Applications, and Programming, Third Edition*. Taylor & Francis Group, London, UNITED KINGDOM, 2016a. ISBN 978-1-317-63313-6. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4556523>.

Barbara M. Byrne. *Structural Equation Modeling with AMOS: Basic Concepts, Applications, and Programming, Third Edition*. Taylor & Francis Group, London, UNITED KINGDOM, 2016b. ISBN 978-1-317-63313-6. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4556523>.

- Barbara M. Byrne. *Structural equation modeling with AMOS: Basic concepts, applications, and programming*. Routledge, London, UNITED KINGDOM, 3rd edition, 2016c. ISBN 978-1-317-63313-6. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4556523>.
- Cristalis Capielo Rosario, Hector Y. Adames, Nayeli Y. Chavez-Dueñas, and Roberto Renteria. Acculturation Profiles of Central Florida Puerto Ricans: Examining the Influence of Skin Color, Perceived Ethnic-Racial Discrimination, and Neighborhood Ethnic-Racial Composition. *Journal of Cross-Cultural Psychology*, 50(4):556–576, May 2019. ISSN 0022-0221, 1552-5422. doi: 10.1177/0022022119835979. URL <http://journals.sagepub.com/doi/10.1177/0022022119835979>.
- Charles S. Carver. You want to measure coping but your protocol's too long: Consider the Brief COPE. *International Journal of Behavioral Medicine*, 4(1):92–100, 1997. ISSN 1070-5503. doi: 10.1207/s15327558ijbm0401_6. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=1999-13167-006&site=ehost-live>. Publisher: Lawrence Erlbaum.
- Chih-Ping Chou and P. M Bentler. Model modification in structural equation modeling by imposing constraints. *Computational Statistics & Data Analysis*, 41(2):271–287, December 2002. ISSN 0167-9473. doi: 10.1016/S0167-9473(02)00097-X. URL <https://www.sciencedirect.com/science/article/pii/S016794730200097X>.
- Jacob Cohen, P. Cohen, Stephen G. West, and Leona S. Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. LErlbaum Associates, Mahwah, N.J., 3rd ed. edition, 2003. ISBN 978-0-8058-2223-6.
- M. R. Cohen and E. Nagel. *An introduction to logic and scientific method*. Harcourt Brace, New York, 1934.
- Jose M. Cortina. What is coefficient alpha? An examination of theory and applications. *Journal of Applied Psychology*, 78(1):98–104, 1993. ISSN 0021-9010. doi: 10.1037/0021-9010.78.1.98. URL <http://doi.apa.org/getdoi.cfm?doi=10.1037/0021-9010.78.1.98>.
- Geoff Cumming. The New Statistics: Why and How. *Psychological Science*, 25(1):7–29, January 2014. ISSN 0956-7976. doi: 10.1177/0956797613504966. URL <https://doi.org/10.1177/0956797613504966>.
- Craig K. Enders. *Applied missing data analysis*. Guilford Press, New York, NY, 2010. ISBN 978-1-60623-639-0.
- Craig K. Enders. Multiple imputation as a flexible tool for missing data handling in clinical research. *Behaviour Research and Therapy*, 98:4–18, November 2017. ISSN 0005-7967. doi: 10.1016/j.brat.2016.11.008. Publisher: Elsevier Science.
- Andy P. Field. *Discovering statistics using R*. Sage, Thousand Oaks, California, 2012. ISBN 978-1-4462-0046-9.
- Samuel B. Green and Neil J. Salkind. *Using SPSS for Windows and Macintosh: analyzing and understanding data*. Pearson, Boston, eighth edition. edition, 2017. ISBN 978-0-13-431988-9.
- Andrew F. Hayes. *Introduction to Mediation, Moderation, and Conditional Process Analysis, Second Edition: A Regression-Based Approach*. Guilford Publications, New York, UNITED STATES, 2018. ISBN 978-1-4625-3467-8. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=5109647>.

- Andrew F. Hayes. *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach*. Guilford Publications, New York, UNITED STATES, 2022a. ISBN 978-1-4625-4905-4. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=6809031>.
- Andrew F. Hayes. More than one Mediator. In *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach*, pages 159–200. Guilford Publications, New York, UNITED STATES, 2022b. ISBN 978-1-4625-4905-4. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=6809031>.
- Li-tze Hu and Peter M. Bentler. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling*, 6(1):1–55, 1999. ISSN 1070-5511. doi: 10.1080/10705519909540118. URL <https://ezproxy.spu.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=1998-03102-001&site=ehost-live>. Publisher: Lawrence Erlbaum.
- Sylvia Hurtado. Linking Diversity with the Educational and Civic Missions of Higher Education. *Review of Higher Education: Journal of the Association for the Study of Higher Education*, 30(2):185–196, 2007. ISSN 0162-5748. doi: 10.1353/rhe.2006.0070. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2006-23268-004&site=ehost-live>. Publisher: Johns Hopkins University Press.
- Sylvia Hurtado and Deborah Faye Carter. Effects of college transition and perceptions of the campus racial climate on Latino college students' sense of belonging. *Sociology of Education*, 70:324–345, October 1997. ISSN 00380407. doi: 10.2307/2673270. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=eue&AN=507591795&site=ehost-live>.
- Juliette M. Iacovino and Sherman A. James. Retaining Students of Color in Higher Education: Expanding Our Focus to Psychosocial Adjustment and Mental Health. In *The Crisis of Race in Higher Education: A Day of Discovery and Dialogue*, volume 19 of *Diversity in Higher Education*, pages 61–84. Emerald Group Publishing Limited, January 2016. ISBN 978-1-78635-710-6 978-1-78635-709-0. doi: 10.1108/S1479-364420160000019004. URL <https://doi.org/10.1108/S1479-364420160000019004>.
- Rajiv S. Jhangiani, I.-Chant A. Chiang, Carrie Cuttler, and Dana C. Leighton. *Research Methods in Psychology*. August 2019. ISBN 978-1-9991981-0-7. doi: 10.17605/OSF.IO/HF7DQ. URL <https://kpu.pressbooks.pub/psychmethods4e/>.
- K. G. Joreskog. Testing structural equation models. In Kenneth A. Bollen and J. Scott Long, editors, *Testing Structural Equation Models*. SAGE, February 1993. ISBN 978-0-8039-4507-4. Google-Books-ID: FvIxxeYDLx4C.
- Aycan Katitas. Getting Started with Multiple Imputation in R, 2019. URL <https://uvastatlab.github.io/2019/05/01/getting-started-with-multiple-imputation-in-r/>. Library Catalog: uvastatlab.github.io.
- D. A. Kenny. SEM: Identification, 2012. URL http://davidakenny.net/cm/identify_formal.htm.

Paul Kim. Yes, Asians and Asian Americans experience racism, March 2021a. URL <https://www.seattletimes.com/opinion/yes-asians-and-asian-americans-experience-racism/>. Section: Opinion.

Paul Y. Kim. Guest Post: Anti-Asian Racism during the Pandemic: How Faculty on Christian Campuses Can Support Asian and Asian American Students, March 2021b. URL <https://christianscholars.com/guest-post-anti-asian-racism-during-the-pandemic-how-faculty-on-christian-campuses-can-support-asian-and-asian-american-students/>.

Paul Youngbin Kim, Dana L. Kendall, and Hee-Sun Cheon. Racial microaggressions, cultural mistrust, and mental health outcomes among asian american college students. *American Journal of Orthopsychiatry*, 87(6):663–670, 2017. ISSN 0002-9432. doi: 10.1037/ort0000203. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2016-39469-001&sit=ehost-live>. Publisher: Educational Publishing Foundation.

Rex B. Kline. The mediation myth. *Basic and Applied Social Psychology*, 37(4):202–213, July 2015. ISSN 0197-3533. doi: 10.1080/01973533.2015.1049349. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2015-36735-002&sit=ehost-live>. Publisher: Taylor & Francis.

Rex B. Kline. Data Preparation and Psychometrics Review (Chapter 4). In *Principles and practice of structural equation modeling*, pages 64–96. Guilford Publications, New York, UNITED STATES, 4th edition, 2016a. ISBN 978-1-4625-2336-8. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4000663>.

Rex B. Kline. *Principles and practice of structural equation modeling*. Guilford Publications, New York, UNITED STATES, 4th edition, 2016b. ISBN 978-1-4625-2336-8. URL <http://ebookcentral.proquest.com/lib/spu/detail.action?docID=4000663>.

Jioni A. Lewis and Helen A. Neville. Construction and initial validation of the Gendered Racial Microaggressions Scale for Black women. *Journal of Counseling Psychology*, 62(2):289–302, April 2015. ISSN 0022-0167. doi: 10.1037/cou0000062. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2015-15500-004&sit=ehost-live>. Publisher: American Psychological Association.

Jioni A. Lewis, Marlene G. Williams, Erica J. Peppers, and Cecile A. Gadson. Applying intersectionality to explore the relations between gendered racism and health among Black women. *Journal of Counseling Psychology*, 64(5):475–486, October 2017. ISSN 0022-0167. doi: 10.1037/cou0000231. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2017-46640-003&sit=ehost-live>. Publisher: American Psychological Association.

Kaleea R. Lewis and Payal P. Shah. Black students' narratives of diversity and inclusion initiatives and the campus racial climate: An interest-convergence analysis. *Journal of Diversity in Higher Education*, October 2019. ISSN 1938-8926. doi: 10.1037/dhe0000147. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2019-59378-001&sit=ehost-live>. Publisher: Educational Publishing Foundation.

- Roderick J. A. Little and Donald B. Rubin. *Statistical analysis with missing data*. Wiley, Hoboken, second edition. edition, 2002. ISBN 978-1-118-62586-6. URL <http://site.ebrary.com/id/10921256>.
- Todd D. Little, William A. Cunningham, Golan Shahar, and Keith F. Widaman. To parcel or not to parcel: Exploring the question, weighing the merits. *Structural Equation Modeling*, 9(2):151–173, April 2002. ISSN 1070-5511. doi: 10.1207/S15328007SEM0902_1.
- Todd D. Little, W. J. Howard, E. K. McConnell, and K. N. Stump. Missing data in large data projects: Two methods of missing data imputation when working with large data projects. *KUant Guides*, 011.3:10, 2008. URL <https://crmda.dept.ku.edu/guides/11.ImputationWithLargeDataSets/11.ImputationWithLargeDataSets.pdf>.
- Todd D. Little, Mijke Rhemtulla, Kimberly Gibson, and Alexander M. Schoemann. Why the items versus parcels controversy needn't be one. *Psychological Methods*, 18(3):285–300, 2013. ISSN 1939-1463, 1082-989X. doi: 10.1037/a0033266. URL <http://doi.apa.org/getdoi.cfm?doi=10.1037/a0033266>.
- R. Duncan Luce. Four tensions concerning mathematical modeling in psychology. *Annual Review of Psychology*, 46:1–26, 1995. ISSN 0066-4308. doi: 10.1146/annurev.ps.46.020195.000245. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=1995-19897-001&site=ehost-live>. Publisher: Annual Reviews.
- Brent Mallinckrodt and Meifen Wei. Attachment, Social Competencies, Social Support, and Psychological Distress. *Journal of Counseling Psychology*, 52(3):358–367, July 2005. ISSN 0022-0167. doi: 10.1037/0022-0167.52.3.358. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2005-08078-009&site=ehost-live>. Publisher: American Psychological Association.
- Brent Mallinckrodt, W. Todd Abraham, Meifen Wei, and Daniel W. Russell. Advances in testing the statistical significance of mediation effects. *Journal of Counseling Psychology*, 53(3):372–378, July 2006. ISSN 0022-0167. doi: 10.1037/0022-0167.53.3.372. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2006-08780-009&site=ehost-live>. Publisher: American Psychological Association.
- Brent Mallinckrodt, Joseph R. Miles, and Jacob J. Levy. The scientist-practitioner-advocate model: Addressing contemporary training needs for social justice advocacy. *Training and Education in Professional Psychology*, 8(4):303–311, November 2014. ISSN 1931-3918. doi: 10.1037/tep0000045. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2014-25072-001&site=ehost-live>. Publisher: Educational Publishing Foundation.
- Della V. Mosley, Helen A. Neville, Nayeli Y. Chavez-Dueñas, Hector Y. Adames, Jioni A. Lewis, and Bryana H. French. Radical hope in revolting times: Proposing a culturally relevant psychological framework. *Social and Personality Psychology Compass*, 14(1), January 2020. ISSN 1751-9004. doi: 10.1111/spc3.12512. Publisher: Wiley-Blackwell Publishing Ltd.
- Della V. Mosley, Candice N. Hargons, Carolyn Meiller, Blanka Angyal, Paris Wheeler, Candice Davis, and Danelle Stevens-Watkins. Critical consciousness of anti-Black

- racism: A practical model to prevent and resist racial trauma. *Journal of Counseling Psychology*, 68(1):1–16, January 2021. ISSN 0022-0167. doi: 10.1037/cou0000430. URL <https://ezproxy.spu.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2020-20397-001&site=ehost-live>. Publisher: American Psychological Association.
- In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100, February 2003. ISSN 00222496. doi: 10.1016/S0022-2496(02)00028-7. URL <https://linkinghub.elsevier.com/retrieve/pii/S0022249602000287>.
- Kevin L. Nadal. The Racial and Ethnic Microaggressions Scale (REMS): Construction, reliability, and validity. *Journal of Counseling Psychology*, 58(4):470–480, October 2011. ISSN 0022-0167. doi: 10.1037/a0025193. URL <https://ezproxy.spu.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2011-19058-001&site=ehost-live>. Publisher: American Psychological Association.
- E. D. Niemark and W. K. Estes. *Stimulus sampling theory*. Holden-Day, San Francisco, CA, 1967.
- Mike C. Parent. Handling item-level missing data: Simpler is just as good. *The Counseling Psychologist*, 41(4):568–600, May 2013. ISSN 0011-0000, 1552-3861. doi: 10.1177/0011000012445176. URL <http://journals.sagepub.com/doi/10.1177/0011000012445176>.
- Judea Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, Cambridge, U.K. ; New York, 2000. ISBN 978-1-139-64936-0.
- Joseph Lee Rodgers. The epistemology of mathematical and statistical modeling: A quiet methodological revolution. *American Psychologist*, 65(1):1–12, January 2010. ISSN 0003-066X. doi: 10.1037/a0018326. URL <https://ezproxy.spu.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2009-24989-001&site=ehost-live>.
- Yves Rosseel. The lavaan tutorial, January 2020. URL <http://lavaan.ugent.be/tutorial/tutorial.pdf>.
- Robert M Sellers, Stephanie A J Rowley, Tabbye M Chavous, J Nicole Shelton, and Mia A Smith. Multidimensional Inventory of Black Identity: A Preliminary Investigation of Reliability and Construct Validity. page 11.
- Klaas Sijtsma. On the Use, the Misuse, and the Very Limited Usefulness of Cronbach's Alpha. *Psychometrika*, 74(1):107–120, March 2009. ISSN 0033-3123. doi: 10.1007/s11336-008-9101-0. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2792363/>.
- Anneliese Singh. Building a Counseling Psychology of Liberation: The Path Behind Us, Under Us, and Before Us. *The Counseling Psychologist*, 48(8):1109–1130, November 2020. ISSN 0011-0000, 1552-3861. doi: 10.1177/0011000020959007. URL <http://journals.sagepub.com/doi/10.1177/0011000020959007>.
- Elliot S. Spengler, David Tierney, L. Christian Elledge, and Patrick R. Grzanka. Beyond the individual: Sexual minority help-seeking and the consequences of structural barriers. *Journal of Counseling Psychology*, 70(2):133–145, March 2023. ISSN 0022-0167. doi: 10.1037/cou0000657.supp. URL <https://ezproxy.spu.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=2023-36794-001&site=ehost-live>. Publisher: American Psychological Association.

- EF Stone-Romero and Patrick Rosopa. Research design options for testing mediation models and their implications for facets of validity. *Journal of Managerial Psychology*, 25:697–712, September 2010. doi: 10.1108/02683941011075256.
- Dawn M. Szymanski and Danielle Bissonette. Perceptions of the LGBTQ College Campus Climate Scale: Development and psychometric evaluation. *Journal of Homosexuality*, 67(10):1412–1428, August 2020. ISSN 0091-8369, 1540-3602. doi: 10.1080/00918369.2019.1591788. URL <https://www.tandfonline.com/doi/full/10.1080/00918369.2019.1591788>.
- Francis Terrell and Sandra L. Terrell. An inventory to measure cultural mistrust among Blacks. *The Western Journal of Black Studies*, 5(3):180–184, 1981. ISSN 0197-4327. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=1982-29464-001&site=ehost-live>. Publisher: Washington State University Press.
- Paolo Toffanin. Multiple-mediator analysis with lavaan, May 2017. URL <https://paolotoffanin.wordpress.com/2017/05/06/multiple-mediator-analysis-with-lavaan/>.
- David M. Tokar, Mark L. Savickas, and Kevin P. Kaut. A Test of the Career Construction Theory Model of Adaptation in Adult Workers With Chiari Malformation. *Journal of Career Assessment*, 28(3):381–401, August 2020. ISSN 1069-0727, 1552-4590. doi: 10.1177/1069072719867733. URL <http://journals.sagepub.com/doi/10.1177/1069072719867733>.
- Clairice T. Veit and John E. Ware. The structure of psychological distress and well-being in general populations. *Journal of Consulting and Clinical Psychology*, 51(5):730–742, October 1983. ISSN 0022-006X. doi: 10.1037/0022-006X.51.5.730. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=1984-02935-001&site=ehost-live>. Publisher: American Psychological Association.
- John E. Ware, Mark Kosinski, Martha S. Bayliss, and Colleen A. McHorney. Comparison of methods for the scoring and statistical analysis of SF-36 health profile and summary measures: Summary of results from the medical outcomes study. *Medical Care*, 33(4, Suppl):264–279, April 1995. ISSN 0025-7079. URL <https://ezproxy.spu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=psyh&AN=1995-35535-001&site=ehost-live>. Publisher: Lippincott Williams & Wilkins.
- Rebecca Weston and Paul A. Gore. A Brief Guide to Structural Equation Modeling. *The Counseling Psychologist*, 34(5):719–751, September 2006. ISSN 0011-0000, 1552-3861. doi: 10.1177/001100006286345. URL <http://journals.sagepub.com/doi/10.1177/001100006286345>.