Principles of Psychological Assessment: With Applied Examples in R

Isaac T. Petersen

2021-06-12

Contents

1 Preface **9**

1.1 Acknowledgments 9

2 Introduction **11**

2.1 About this Book 11

2.2 Why R? 11

2.3 About The Author 12

2.4 What is Assessment? 12

2.5 Why Should I Care about Assessment (and Science)? 12

2.6 Prerequisites 17

I Part I **19**

3 Reliability **21**

3.1 Classical Test Theory 21

3.2 Measurement Error 28

3.3 Overview of Reliability 32

3.4 Getting Started 36

3.5 Types of Reliability 38

3.6 Applied Examples 67

3.7 Standard Error of Measurement (SEM) 68

3.8 Influences of Measurement Error on Test–Retest Reliability . . . 69

3.9 Effect of Measurement Error on Associations 83

3.10 Method Bias 84
3.11 Generalizability (G-)Theory 89
3.12 Item Response Theory 94
3.13 The Problem of Low Reliability 94
3.14 Ways to Increase Reliability 94
3.15 Conclusion 96
3.16 Exercises 96

4 Validity 103

4.1 Overview of Validity 103
4.2 Getting Started 104
4.3 Types of Validity 105
4.4 Validity is a Process, Not an Outcome 140
4.5 Reliability versus Validity 141
4.6 Effect of Measurement Error on Associations 143
4.7 Generalizability Theory 148
4.8 Ways to Increase Validity 148
4.9 Conclusion 150
4.10 Exercises 150

5 Structural Equation Modeling 153

5.1 Overview of SEM 153
5.2 Getting Started 153
5.3 Types of Models 154
5.4 Types of Constructs 156
5.5 Practical issues 162
5.6 Additional Types of SEM 168
5.7 Correlation Matrix 168
5.8 Measurement Model 169
5.9 Confirmatory Factor Analysis (CFA) 179
5.10 Structural Equation Model (SEM) 203
5.11 Benefits of SEM 218

5.12 Power Analysis using Monte Carlo Simulation 220
5.13 Generalizability Theory 238
5.14 Conclusion 238
5.15 Exercises 239

6 Item Response Theory 245

6.1 Overview of IRT 245
6.2 Getting Started 269
6.3 Comparison of Scoring Approaches 269
6.4 Relation between IRT, SEM, and CFA 271
6.5 Rasch Model (1-Parameter Logistic) 271
6.6 Two-Parameter Logistic Model 274
6.7 Two-Parameter Multi-Dimensional Logistic Model 289
6.8 Three-Parameter Logistic Model 297
6.9 Four-Parameter Logistic Model 309
6.10 Graded Response Model 317
6.11 Conclusion 327
6.12 Exercises 327

7 Prediction 337

7.1 Overview of Prediction 337
7.2 Getting Started 364
7.3 Receiver Operating Characteristic (ROC) Curve 365
7.4 Prediction Accuracy Across Cutoffs 368
7.5 Prediction Accuracy at a Given Cutoff 395
7.6 Optimal Cutoff Specification 426
7.7 Accuracy at Every Possible Cutoff 431
7.8 Regression for Prediction of Continuous Outcomes 443
7.9 Ways to Improve Prediction Accuracy 446
7.10 Conclusion 448
7.11 Exercises 448

8 Clinical Judgment vs. Algorithmic Prediction 461

8.1 Approaches to Prediction 461

8.2 Errors in Clinical Judgment 462

8.3 Humans versus Computers 463

8.4 Accuracy of Different Statistical Models 464

8.5 Why Clinical Judgment is More Widely Used Than Statistical Formulas 464

8.6 Conclusion 464

9 General Issues in Clinical Assessment 465

9.1 Terminology 465

9.2 The Three S's: Signs, Symptoms, and Syndromes 465

9.3 Conclusion 466

10 Evidence-Based Assessment 467

10.1 Conclusion 471

11 Ethical Issues in Assessment 473

11.1 Conclusion 473

12 Factor Analysis and Principal Component Analysis 475

12.1 Getting Started 475

12.2 Descriptive Statistics and Correlations 476

12.3 Factor Analysis 478

12.4 Exploratory Factor Analysis (EFA) 478

12.5 Confirmatory Factor Analysis (CFA) 497

12.6 Principal Component Analysis (PCA) 659

12.7 Conclusion 673

12.8 Exercises 673

II Part II 679

13 Intellectual Assessment 681

13.1 Conclusion 683

14 Test Bias **687**
14.1 Getting Started 687
14.2 Examples of Unbiased Tests (in terms of predictive bias) 689
14.3 Predictive Bias: Different Regression Lines 693
14.4 Differential Item Functioning (DIF) 704
14.5 Measurement/Factorial Invariance 727
14.6 Conclusion 867
14.7 Exercises 868

15 The Interview and the DSM **869**
15.1 Conclusion 869

16 Objective Personality Testing **871**
16.1 Conclusion 871

17 Projective Personality Testing **873**
17.1 Conclusion 873

18 Psychophysiological Measures **875**
18.1 Conclusion 875

19 Computers and Adaptive Testing **877**
19.1 Getting Started 877
19.2 Example of Uni-dimensional CAT 877
19.3 Conclusion 888
19.4 Exercises 888

20 Behavioral Assessment **893**
20.1 Conclusion 893

21 Repeated Assessments across Time **895**
21.1 Types of Research Designs 895
21.2 Cross-sequential design 899
21.3 Using Sequential Designs To Make Developmental Inferences . . 900
21.4 Conclusion 901

22 Assessment of Cognition **903**

22.1 Conclusion 903

23 Cultural and Individual Diversity **905**

23.1 ADDRESSING 905

23.2 Conclusion 907

24 Reproducibility **909**

24.1 Session Info 909

Chapter 1

Preface

This is a book in progress—it is incomplete. I will continue to add to and update it as I am able. If you have data or analysis examples that are you willing to share and include in the book, feel free to contact me.

Figure 1.1: book cover

1.1 Acknowledgments

This book would not be possible without the help of others. Much of the content of this book was inspired by Richard Viken's course in psychological assessment that I took as a graduate student. I also thank W. Joel Schneider

who provided some of the data and examples that were adapted for this book. I also acknowledge my wife, Alyssa Varner, who helped design several of the graphics used in this book, in addition to all of her support throughout the process.

Chapter 2

Introduction

2.1 About this Book

First, let us discuss what this book is *not*. This book is not a guide on how to assess each psychological construct or disorder. This book is also not a comparative summary of the psychometrics of different measures. There already exist many resources that summarize and compare the reliability and validity of measures in psychology. Instead, this book is about the *principles* of psychological assessment [Buros Center for Testing, 2021]. This book was originally written for a graduate-level course. The goal of this book is to help researchers and clinicians learn to think critically about assessments so they can better develop, evaluate, administer, score, integrate, and interpret psychological assessments. Learning important principles of assessment will put you in a better position to learn any assessment device, and to develop better ones. In this book, I apply a scientific perspective to the principles of psychological assessment. The assessments used in a given situation—whether in research or practice—should be supported by the strongest available science (or should be used cautiously while undergoing development and study). In addition to discussing principles, however, I also provide analysis scripts in the software R [R Core Team, 2020], so that you are able to apply the principles discussed in this book.

2.2 Why R?

R is free, open source, open platform, and widely used. Unlike proprietary software used for data analysis, R is not a black box. You can examine the code for any function or computation you perform. You can even modify and improve these functions by changing the code, and can create your own functions. R also has advanced capabilities for data wrangling, and has many packages available

for advanced statistical analysis and graphing. In addition, there are strong resources available for creating your analyses in R so they are reproducible by others [Gandrud, 2020].

2.3 About The Author

Regarding my background, I am a licensed clinical psychologist. My research examines how children develop behavior problems. I am also a trained clinician, and I supervise training clinicians in assessment and therapy, particularly assessment and treatment of children's disruptive behavior. Given my expertise, many of the examples in the book deal with topics in clinical psychology, but many of the assessment principles discussed are relevant to all areas of psychology—and science more broadly—and are often overlooked in research and practice. As a card-carrying clinical scientist, my perspective is that the scientific epistemology is the strongest approach to knowledge, and that assessment should be led first and foremost by the epistemology of science, regardless of whether one is doing research or practice.

2.4 What is Assessment?

Assessment is the gathering of information about a person, group, setting, or context. In psychological assessment, we are interested in gathering information about people's psychological functioning, including their thoughts, emotions, and behaviors. Psychological assessment can also consider biological and physiological processes that are linked to people's thoughts, emotions, and behaviors. Many assessment approaches can be used to assess people's thoughts, emotions, and behaviors, including self-report questionnaires, questionnaires reported by others (e.g., a spouse, parent, teacher, or friend), interviews, observations, biopsychological assessments (e.g., cortisol, heart rate, brain imaging), performance-based assessments, archival approaches (e.g., chart review), and combinations of these.

2.5 Why Should I Care about Assessment (and Science)?

In research, assessments are conducted to advance knowledge, such as improved prediction or understanding. For example, in my research, I use assessments to understand what processes influence children's development of disruptive behavior. In society, assessments are conducted to improve decision making. For instance, assessments are conducted to determine whether to hire a job candidate or promote an employee. In a clinical context, assessments are conducted

to improve treatment and the client's outcomes. For instance, assessments are conducted to determine which treatment would be most effective for a person suffering from depression. Assessments can be valuable both to understanding current functioning as well as making predictions. To best answer these questions and address these goals, we need to have confidence that our devices yield accurate answers for these purposes for the assessed individuals. Science is crucial for knowing how much (or how little) confidence we have in a given assessment for a given purpose and population. Effective treatment often depends on accurate assessment. Thus, knowing how to conduct and critically evaluate science will make you more effective at selecting, administering, and interpreting assessments.

Decisions resulting from assessments can have important life-altering consequences. High-stakes decisions based on assessments include decisions about whether a person is hospitalized, whether a child is removed from their abusive home, whether a person is deemed competent to stand trial, whether a prisoner is released on parole, and whether an applicant is admitted to graduate school. These important assessment-related decisions should be made using the best available science.

The problem is that there has been a proliferation of pseudoscience in assessment and treatment. There are widely used psychological assessments and treatments that we know are inaccurate, do not work, or in some cases, are known to be harmful. Lists of harmful psychological treatments [e.g., Lilienfeld, 2007] and inaccurate assessments [e.g., Hunsley et al., 2015] have been published, but these treatments and assessments are still used by professional providers to this day. Practice using such techniques violates the aphorism "first, do no harm." This would be inconceivable for other applied sciences, such as chemistry, engineering, and medicine. For instance, the prescription of a particular medication for a particular purpose requires approval by the U.S. Food and Drug Administration. Psychological assessments and treatments do not have the same level of oversight.

The gap between what we know based on science and what is implemented in practice (the science–practice gap) motivated McFall's [1991] "Manifesto for a Science of Clinical Psychology," which he later expanded [McFall, 2000]. The Manifesto has one cardinal principle and four corollaries:

> Cardinal Principle: Scientific clinical psychology is the only legitimate and acceptable form of clinical psychology.
> First Corollary: Psychological services should not be administered to the public (except under strict experimental control) until they have satisfied these four minimal criteria:
>
> 1. The exact nature of the service must be described clearly.
> 2. The claimed benefits of the service must be stated explicitly.
> 3. These claimed benefits must be validated scientifically.

4. Possible negative side effects that might outweigh any benefits must be ruled out empirically.

Second Corollary: The primary and overriding objective of doctoral training programs in clinical psychology must be to produce the most competent clinical scientists possible.
Third Corollary: A scientific epistemology differentiates science from pseudoscience.
Fourth Corollary: The most caring and humane psychological services are those that have been shown empirically to be the most effective, efficient, and safe.

The Manifesto orients you to the scientific perspective from which we will be examining psychological assessment techniques in this book.

2.5.1 Assessment and the Replication Crisis in Science

Assessment is also crucial to advancing knowledge in research, as summarized in the maxim, "What we know depends on how we know it." Findings from studies boil down to the methods that were used to obtain them—everything comes down to methods.

Many domains of science, particularly social science, have struggled with a replication crisis, such that a large proportion of findings fail to replicate when independent investigators attempt to replicate the original findings [Duncan et al., 2014, Freese and Peterson, 2017, Larson and Carbine, 2017, Lilienfeld, 2017, Open Science Collaboration, 2015, Shrout and Rodgers, 2018, Tackett et al., 2019a]. There is considerable speculation on what factors may best account for the replication crisis. For instance, one possible factor is the researcher degrees of freedom, which are unacknowledged choices in how researchers prepare, analyze, and report their data that can lead to detecting significance in the absence of real effects [Loken and Gelman, 2017]. This is similar to Gelman and Loken [2013]'s description of research as the garden of forking paths, where different decisions along the way can lead to different outcomes [**include graphic**]. A second possibility for the replication crisis is that some replication studies have had limited statistical power (e.g., insufficiently large sample sizes). A third possibility may be that there is publication bias such that researchers tend to publish only significant findings, which is known as the file-drawer effect. A fourth possibility is that researchers may engage in questionable research practices. As described by John et al. [2012], questionable research practices may include:

- not reporting all dependent variables
- deciding to collect more data after looking at the data to see whether the results were significant

- not reporting all of a study's conditions
- stopping collecting data earlier than planned because the expected result is detected
- rounding down a p-value
- selectively reporting studies that worked
- deciding whether to exclude data after looking at the impact of doing so on the results
- reporting an unxpected finding as having been predicted from the start (known as HARK-ing: **h**ypothesizing **a**fter the **r**esults are **k**nown)
- claiming that results are unaffected by demographic variables when one is actually unsure
- falsifying data
- p-Hacking: multiple testing, selective reporting, and misreporting of the true effect size

However, difficulties with replication could exist even if researchers have the best of intentions, engage in ethical research practices, and are transparent about all of the methods they used and decisions they made. The replication crisis could owe, in part, to noisy (imprecise and inaccurate) measures. As Loken and Gelman [2017] demonstrated, when measures are less noisy, measurement error weakens the association between the measures. But when using noisy measures and selecting what to publish based on statistical significance, measurement error can make the association appear stronger than it is (see Figure **provide figure**). This is what Loken and Gelman [2017] describe as the statistical significance filter: in a study with a noise measures and a small or moderate sample size, statistically significant estimates are likely to have a stronger effect size than the actual effect size—the "true" underlying effects could be small or nonexistent. However, the effect of noise (measurement error) diminishes as the sample size increases. So, the goal should be to use less noisy measures with larger sample sizes. And, as discussed in Chapter 11 on ethical considerations in psychological assessment, the use of pre-registration could be useful to control researcher degrees of freedom.

The lack of replicability of findings has the potential to negatively impact the people we study through misinformed assessment, treatment, and policy decisions. Therefore, it is crucial to use assessments with strong psychometric properties and/or to develop better assessments. By psychometrics, I refer to the reliability and validity of measures. These concepts will be described in greater detail in Chapters 3 and 4, but for now, think about reliability as *consistency* of measurement and *validity* as accuracy of measurement.

2.5.2 Science vs. pseudoscience in assessment

Science is the best system of epistemology we have to pursue truth. Science is a process, not a set of facts. It helps us overcome blind spots. The system

is revisionary and self-correcting. Science is the epistemology that is the least susceptible to error due to authority, belief, intuition, bias, and preference, etc. Clients are in a vulnerable position, and deserve to receive services consistent with the strongest available evidence. By providing a client a service, you are implicitly making a claim and prediction. As a psychologist, you are claiming to have expert knowledge and competence. You are making a prediction that the client will improve because of your services. Ethically, you should be making these predictions based on science and a risk-benefit analysis. It is also important to make sure the client knows when services are unproven so they can provide fully informed consent. Otherwise, because of your position as a psychologist, they may believe that you are using a evidence-based approach when you are not.

We will be examining psychological assessment from a scientific perspective. Here are characteristics of science that distinguish it from pseudoscience:

1. Risky hypotheses are posed that are falsifiable. The hypotheses can be shown to be wrong.
2. Findings can be replicated independently by different research groups and different methods. Evidence converges across studies and methods.
3. Potential alternative explanations for findings are specified and examined empirically (with data).
4. Steps are taken to guard against the undue influence of personal beliefs and biases.
5. The strength of claims reflects the strength of evidence. Findings and the ability to make judgments or predictions are not overstated. For instance, it is important to present the degree of uncertainty from assessments with error bars or confidence intervals.
6. Scientifically supported measurement strategies are used based on their psychometrics, including reliability and validity.

Science does not progress without advances in measurement, including:

- more efficient measurement
- more precise measurement
- more accurate measurement
- more sophisticated modeling
- more sophisticated biopsychological (e.g., cognitive neuroscience) techniques, as opposed to self-report and neuropsychological techniques
- considerations of cultural and individual diversity
- ethical considerations

These considerations serve as the focus of this book.

2.6 Prerequisites

This book was written in `Markdown` using the `bookdown` package [Xie, 2020] in `R` [R Core Team, 2020]. The `bookdown` package was built on top of the `rmarkdown` [Allaire et al., 2020] and `knitr` [Xie, 2021] packages [Xie, 2015].

If you want to compile the book, you will need the `bookdown` package. The `bookdown` package can be installed from CRAN:

```
install.packages("bookdown")
```

Part I

Part I

Chapter 3

Reliability

3.1 Classical Test Theory

To understand the concept of reliability, it is helpful to understand *classical test theory* (CTT), which is also known as "true score theory." Classical test theory is one of various measurement theories, in addition to item response theory and generalizability theory. CTT has been the predominant measurement theory through the history of psychology. CTT is a theory of how test scores relate to a construct. A *construct* is the concept or characteristic that a measure is intended to assess.

Let us assume you take a measurement of the same object multiple times (i.e., repeated measure). For example, you assess the mass of the same rock multiple times. However, you obtain different estimates of the rock's mass each time. There are multiple possible explanations for this observation. One possible explanation could be that the rock is changing in its mass, which would be consistent with the idea proposed by the Greek philosopher Heraclitus that nothing is stable and the world is in flux. An alternative possibility, however, is that the rock is stable in its mass but the measurements are jittery—that is, they include error.

Based on the possibility that the rock is stable, and that differences in scores across time reflect measurement error, CTT proposes the following formula:

$$X = T + e$$
$$\text{observed score} = \text{true score} + \text{measurement error}$$

X is the observed measurement or score, T is the classical (or psychometric) "true score" and e is the measurement error.

This formula is depicted visually in the form of a path diagram in Figure 3.1.

Figure 3.1: Classical test theory formula in a path diagram.

It is important to distinguish between the classical true score and the Platonic true score. The *Platonic true score* is the truth, and it does not depend of measurement. The Platonic true score is an abstract notion because it is not directly observable and is based on Platonic ideals and theories of the construct and what a person's "true" level is on the construct. In CTT, we attempt to approximate the Platonic true score with the classical true score, (T). If we took infinite repeated observations (and the measurements had no carryover effect), the average score approaches the classical true score, T. That is, $\overline{X} = T$ as number of observations $\rightarrow \infty$. CTT attempts to partition variance into different sources. *Variance* is an index of scores' variability, i.e., the degree to which scores differ. Variance is defined as the average squared deviation from the mean:

$$\sigma_X^2 = E[(X - \mu^2)]$$

According to CTT, any observed measurement includes both true score (construct) variance (σ_T^2) and error variance (σ_e^2). Given the true score formula ($X = T + e$), this means that their variance is:

$$\sigma_X^2 = \sigma_T^2 + \sigma_e^2$$
$$\text{observed score variance} = \text{true score variance} + \text{error variance}$$

Nevertheless, The *classical true score*, T, is the expected value and is not necessarily the same thing as the Platonic true score [Borsboom, 2003, Klein and Cleary, 1969], because the expected value would need to be entirely valid/accurate (i.e., it would need to be the construct score) for it to be the Platonic true score. The expected score could also be influenced by systematic sources of error such as other constructs, which would not fall into the error portion of the CTT formula because, as described below, CTT assumes that error is random (not systematic).

The true score formula is theoretically useful, but it is not practically useful because it is an under-identified equation and we do not know the values of T or e based on knowing the observed score (X). For instance, if we obtain an observed score of 10, our formula is $10 = T + e$, and we do not know what the true score or error is. As a result, CTT makes several simplifying assumptions so we can estimate how stable (reliable) or noisy the measure is, and what proportion of the observed score reflects true score versus measurement error.

1. $E(e) = 0$

   The first assumption of CTT is that the expected value of the error is zero. Basically, the errors are expected to be random with a mean of zero. The likelihood that the observed score is an overestimate of T is assumed to be the same as the likelihood that the observed score is an underestimate of T. In other words, the distribution of errors above T is assumed to be

the same as the distribution of errors below T. In reality, this assumption is likely false in many situations. For instance, social desirability bias is a systematic error where people rate themselves as better than they actually are; social desirability biases scores in a particular direction across respondents, so errors would not be entirely random. But using the assumption that the expected value of e is zero, also informs that the expected value of the observed score (X) equals the expected value of the true score (T):

$$\begin{aligned} E(X) &= E(T + e) \\ &= E(T) + E(e) \\ &= E(T) + 0 \\ &= E(T) \end{aligned}$$

2. $r_{T,e} = 0$

   The second assumption is that the correlation between T and e is zero—that is, true scores and errors are uncorrelated. However, this assumption is likely false in many situations. For instance, one can imagine that, on a a paper-and-pencil intelligence test, scores may have greater error for respondents with lower true scores and may have less error for respondents with higher true scores.

3. $r_{e_1,e_2} = 0$

   The third assumption is that errors at time 1 (e_1) are not associated with errors at time 2 (e_2). However, this assumption is also likely false in many situations. For example, if some people have a high social desirability bias at time 1, they are likely to also have a high social desirability bias at time 2. That is, the errors are likely to be related across time.

These three assumptions are depicted in the path analytic diagram in Figure 3.2, which depicts the CTT approach to understanding reliability of a measure across two time points.

In path analytic (and structural equation modeling) language, rectangles represent variables we observe, and circles represent latent (i.e., unobserved) variables. The observed scores at time 1 (X_1) and time 2 (X_2) are entities we observe, so they are represented by rectangles. We do not directly observe the true scores (T_1, T_2) and errors (e_1, e_2), so they are considered latent entities and are represented by circles. Single-headed arrows indicate regression paths, where conceptually, one variable is thought to influence another variable. As the model depicts, the observed scores are thought to be influenced both by true scores and by error. We also expect the true scores at time 1 (T_1) and time 2 (T_2) to be correlated, so we have a covariance path, as indicated by a double-headed arrow. A *covariance* is an unstandardized index of the strength of association between two variables. Because a covariance is unstandardized,

Figure 3.2: Reliability of a measure across two time points, as depicted in a path diagram.

its scale depends on the scale of the variables. The covariance between two variables is the average product of their deviations from their respective means:

$$\sigma_{XY} = E[(X - \mu_X)(X - \mu_Y)]$$

The covariance of a variable with itself is equivalent to its variance:

$$\begin{aligned}\sigma_{XX} &= E[(X - \mu_X)(X - \mu_X)] \\ &= E[(X - \mu_X)^2] \\ &= \sigma_X^2\end{aligned}$$

By contrast, a *correlation* is a standardized index of the strength of association between two variables. Because a correlation is standardized (fixed between [-1,1]), its scale does not depend on the scales of the variables. In this figure, no other parameters (regressions, covariances, or means) are specified, so implicit in the diagram:

- $E(e) = 0$
- $r_{T,e} = 0$
- $r_{e_1,e_2} = 0$

The factor loadings reflect the magnitude that the latent factor influences the observed variable. In this case, the true scores influence the observed scores with a magnitude of $\sqrt{r_{xx}}$, which is known as the *index of reliability*. The index of reliability is the theoretical estimate of the correlation between the true scores and the observed scores. This is depicted in Figure 3.3.

We can use path tracing rules to estimate the reliability of the measure, where the reliability of the measure, i.e., the *coefficient of reliability* (r_{xx}), is estimated as the correlation between x_1 and x_2. According to path tracing rules [Pearl, 2013], the correlation between x_1 and x_2 is equal to the sum of the standardized coefficients of all the routes through which x_1 and x_2 are connected. The contribution of a given route to the correlation between x_1 and x_2 is equal to the product of all standardized coefficients on that route that link x_1 and x_2 that move in the following directions: (a) forward (e.g., T_1 to x_1), or (b) backward once and then forward (x_1 to T_1 to T_2 to x_2). Path tracing does not allow moving forward and then backward—that is, it does not allow retracing (e.g., e to x_1 to T_1) in the same route. It also does not allow passing through through more than one curved arrow (covariance path) or through the same variable twice in the same route. Once you know the contribution of each route to the correlation, you can calculate the total correlation between the two variables as the sum of the contribution of each route. Therefore, using one route, we can calculate the association between x_1 and x_2 as:

Figure 3.3: Reliability of a measure across two time points, as depicted in a path diagram.

$$\begin{aligned} r_{x_1,x_2} &= \sqrt{r_{xx}} \times r_{T_1,T_2} \times \sqrt{r_{xx}} \\ &= r_{T_1,T_2} \times r_{xx} \\ &= \text{correlation of true scores across time} \times \text{reliability} \end{aligned}$$

When dealing with a stable construct, we would assume that the correlation between true scores across time is 1.0: $r_{T_1,T_2} = 1.0$, as depicted in Figure 3.4.

Then, to calculate the association between x_1 and x_2 of a stable construct using path tracing rules:

$$\begin{aligned} r_{x_1,x_2} &= \sqrt{r_{xx}} \times r_{T_1,T_2} \times \sqrt{r_{xx}} \\ &= \sqrt{r_{xx}} \times 1 \times \sqrt{r_{xx}} \\ &= r_{xx} \\ &= \text{coefficient of reliability} \end{aligned}$$

That is, for a stable construct (i.e., whose true scores are perfectly correlated across time; $r_{T_1,T_2} = 1.0$), we estimate reliability as the correlation between the observed scores at time 1 (x_1) and the observed scores at time 2 (x_2). This is known as *test–retest reliability*. We therefore assume that the extent to which the correlation between x_1 and x_2 is less than one reflects measurement error (an unstable measure), rather than people's changes in their true score on the construct (an unstable construct).

As described above, the reliability coefficient (r_{xx}) is the association between a measure and itself over time, or with another measure in the domain. By contrast, the *reliability index* (r_{xT}) is the correlation between observed scores on a measure and the true scores [Nunnally and Bernstein, 1994]. It is the square root of the reliability coefficient.

$$\begin{aligned} r_{xT} &= \\ &= \sqrt{r_{xx}} \\ &= \text{index of reliability} \end{aligned}$$

3.2 Measurement Error

Measurement error is the difference between the measured (observed) value and the true value. All measurements come with uncertainty and measurement error. Even a measure of something as simple as whether someone is dead has error.

There are two main types of measurement error: systematic (nonrandom) errors and unsystematic (random) errors.

Figure 3.4: Reliability of a measure of a stable construct across two time points, as depicted in a path diagram.

3.2.1 Systematic (nonrandom) error

Systematic errors are depicted in Figure 3.5. Systematic errors are errors than influence across the sample. They are errors that always occur, with the same value, when using the measure in the same way and in the same case. An example of a systematic error is a measure that consistently assesses constructs other than the construct the measure was designed to assess. For instance, if a test written in English to assess math skills is administered in a non-native English-speaking country, some portion of the scores will reflect variance attributable to English reading skills rather than the construct of interest (math skills). Other examples of systematic errors include response styles or subjective, idiosyncratic judgments by a rater—for instance, if the rater's judgments are systematically harsh or lenient. Systematic errors affect the average score (i.e., they result in bias) which makes the group-level estimates less accurate, and they make the measurements for an individual less accurate.

As depicted in Figure 3.5, systematic error does not affect the variability of the scores but it does affect the mean of the scores, so the group-level mean is less accurate. In other words, the systematic errors leads to a biased estimate of the average.

Figure 3.5: Correlation coefficients

3.2.2 Unsytematic (random) error

Unsystematic (random) errors are depicted in Figure 3.5. Random errors occur due to chance. For instance, they could arise from a participant being fatigued on a particular testing day, or from a participant getting luck in guessing the correct answer. Random errors do not have consistent effects across the sample, and they may vary from one observation to another. Random errors do not have the average, i.e., the group-level estimate—they only affect the variability around the average (noise). However, random errors make measurements for an individual less accurate. A large number of observations of the same construct cancels out random errors, but does not cancel out systematic errors.

As depicted in Figure 3.6, random error does not affect the mean of the scores but it does increase the variability of the scores. In other words, the group-level mean is still accurate, but individuals' scores are less precise.

Figure 3.6: Correlation coefficients

3.2.3 Summary

In sum, all measurement error (whether systematic or random) leads to less accurate scores for an individual. However, systematic and random error have different effects on accuracy at the group-level.

CTT assumes that all errors are random. According to CTT, as the number of measurements approaches infinity, the mean of the measurements gets closer to the true score. With more measurements, we reduce our uncertainty and increase our precision. According to CTT, if we take many measurements and the average of the measurements is 10, we have some confidence that $T \approx 10$.

3.3 Overview of Reliability

The "Standards for Educational and Psychological Testing" set the standard for educational and psychological assessment, and are jointly published by the American Educational Research Association, American Psychological Association, and National Council on Measurement in Education. According to the "Standards" [American Educational Research Association et al., 2014, p. 33], reliability is the "consistency of the scores across instances of the testing procedure". In this book, I define reliability as how much repeatability, consistency, and precision the scores from a measure have.

Reliability (r_{xx}) has been defined mathematically as the proportion of observed score variance (σ_X^2) that is attributable to true score variance (σ_T^2):

$$\begin{aligned} r_{xx} &= \frac{\sigma_T^2}{\sigma_T^2 + \sigma_e^2} \\ &= \frac{\sigma_T^2}{\sigma_X^2} \\ &= \frac{\text{true score variance}}{\text{observed score variance}} \end{aligned}$$

An alternative formulation is that reliability (r_{xx}) is the the lack of error variance or the degree to which observed scores are uncorrelated with error scores. In CTT, reliability can be conceptualized in four primary ways, as depicted in Figure 3.7 [Furr, 2017].

		Conceptual basis of reliability: Observed scores in relation to...	
		True Scores	**Measurement Error**
Statistical basis of reliability in terms of...	**Proportions of Variance**	Ratio of true score variance to observed score variance: $r_{xx} = \frac{\sigma_T^2}{\sigma_X^2}$	Lack of error variance: $r_{xx} = 1 - \frac{\sigma_e^2}{\sigma_X^2}$
	Correlations	Squared correlation between observed scores and true scores: $r_{xx} = r_{xT}^2$	Lack of correlation between observed scores and error scores: $r_{xx} = 1 - r_{xe}^2$

Figure 3.7: Four different ways of conceptualizations reliability.

However, we cannot *calculate* reliability because we cannot measure the true score component of an observation. Therefore, we *estimate* reliability (the coefficient of reliability) based on the relation between two observations of the same measure (for test–retest reliability), or using other various estimates of reliability.

The coefficient of reliability can depend on several factors. Reliability is inversely related to the amount of measurement error. The coefficient of reliability, like correlation, also depends on the degree of spread (variance) of the scores. If the scores at one or both time points show restricted range, the scores will show a weaker association and coefficient of reliability, as shown in Figure 3.12. Instead of taking one person and repeatedly measuring them many times to get an estimate of their true score and the reliability of the measure, we typically estimate reliability by taking many people and doing repeated measures twice. This is a shortcut to estimate reliability, but even that is not often done. Reliability can also be related to the number of items in the measure. In general, the greater the number of items, the more reliable the measure (assuming the items assess the same construct), because we are averaging out random error.

We never see the true scores or the error scores, so we cannot compute reliability—we can only *estimate* it from the observed scores. This estimate of reliability gives a probabilistic answer of the reliability of the measure, rather than an absolute answer.

3.3.1 How Reliable is Reliable Enough?

As described by Nunnally and Bernstein [1994], how reliable a measure should be depends on the proposed uses. If it is early in the research process, and the focus is on group-level inferences (e.g., associations or group differences), modest reliability (e.g., .70) may be sufficient and save time and money. Then, the researcher can see what the associations would be when disattenuated for unreliability, as described in Section 4.6 of the chapter on validity. If the disattenuated associations are promising, it may be worth increasing the reliability of the measure. Associations are only weakly attenuated above a reliability of .80, so this may be an appropriate target for basic research.

However, when making decisions about individual people from their score on a measure, reliability and precision are more important because small difference in scores can lead to different decisions. Nunnally and Bernstein [1994] recommend that measures have at least a reliability of .90, and preferably a reliability of .95 or higher when making important decisions about individual people. Nevertheless, they also note that one should not switch to a less valid measure merely because it is more reliable.

3.3.2 Standard Error of Measurement (SEM)

The estimate of reliability gives a general idea of the degree of uncertainty you have of a person's true score given their observed score. From this, we can estimate the standard error of measurement, which estimates the extent to which an observed score deviates from a true score. The standard error of measurement indicates the typical distance of the observed score from the true score.

$$\text{standard error of measurement (SEM)} = \sigma_x \sqrt{1 - r_{xx}}$$

where σ_x represents the standard deviation of scores. Thus, the standard error of measurement is directly related to the reliability of the measure. The higher the reliability, the lower the standard error of measurement.

Figure 3.8: Standard error of measurement as a function of reliability

Here is the derivation of the SEM (from W. Joel Schneider):

$$
\begin{aligned}
\text{Remember, based on } X = T + e: \quad & \sigma_X^2 = \sigma_T^2 + \sigma_e^2 \\
\text{Solve for } \sigma_T^2: \quad & \sigma_T^2 = \sigma_X^2 - \sigma_e^2 \\
\text{Remember:} \quad & r_{xx} = \frac{\sigma_T^2}{\sigma_X^2} \\
\text{Substitute for } \sigma_T^2: \quad & = \frac{\sigma_X^2 - \sigma_e^2}{\sigma_X^2} \\
\text{Multiply by } \sigma_X^2: \quad & \sigma_X^2 \cdot r_{xx} = \sigma_X^2 - \sigma_e^2 \\
\text{Solve for } \sigma_e^2: \quad & \sigma_e^2 = \sigma_X^2 - \sigma_X^2 \cdot r_{xx} \\
\text{Factor out } \sigma_X^2: \quad & \sigma_e^2 = \sigma_X^2(1 - r_{xx}) \\
\text{Take the square root:} \quad & \sigma_e = \sigma_X \sqrt{1 - r_{xx}}
\end{aligned}
$$

The SEM is equivalent to the standard deviation of measurement error (e) [Lek and Van De Schoot, 2018]:

$$
\begin{aligned}
\text{standard error of measurement (SEM)} &= \sigma_x \sqrt{1 - r_{xx}} \\
&= \sqrt{\sigma_x^2} \sqrt{1 - r_{xx}} \\
&= \sqrt{\sigma_x^2 (1 - r_{xx})} \\
&= \sqrt{\sigma_x^2 - \sigma_x^2 \cdot r_{xx}} \\
&= \sqrt{\sigma_x^2 - \sigma_x^2 \frac{\sigma_T^2}{\sigma_x^2}} \\
&= \sqrt{\sigma_x^2 - \sigma_T^2} \\
&= \sqrt{\sigma_e^2} \\
&= \sigma_e
\end{aligned}
$$

Around 95% of scores would be expected to fall within ± 2 SEMs of the true score (or, more precisely, within ± 1.959964 SEMs of the true score). In other words, 95% of the time, the true score is expected to fall within ± 2 SEMs of the observed score. Given an observed score of $X = 15$ and SEM $= 2$, the 95% confidence interval of the true score is [11, 19]. So, if a person's get a score of 15 on the measure, 95% of the time, their true score is expected to fall within 11–19. I provide an empirical example below of estimating the SEM in Section 3.3.2.

Based on the preceding discussion, consider the characteristics of measures that make them more useful from a reliability perspective. A useful measure would show wide variation across people (individual differences), so we can more accurately estimate its reliability. And we would expect a useful measure to show consistency, stability, precision, and reliability of scores.

3.4 Getting Started

Now that we have discussed some introductory concepts involving reliability, applied examples in R are provided below. Each chapter that has R examples in this book has a section on "Getting Started" which provides the code to load relevant libraries, load data files, simulate data, add missing data, conduct calculations, and more.

3.4.1 Load Libraries

```
library("psych")
library("blandr")
library("MBESS")
library("lavaan")
library("psychmeta")
library("gtheory")
library("tidyverse")
library("tinytex")
library("knitr")
library("rmarkdown")
library("bookdown")
```

3.4.2 Prepare Data

3.4.2.1 Simulate Data

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
sampleSize <- 100

set.seed(52242)

rater1continuous <- rnorm(n = sampleSize, mean = 50, sd = 10)
rater2continuous <- rater1continuous + rnorm(n = sampleSize, mean = 0, sd = 4)
rater3continuous <- rater2continuous + rnorm(n = sampleSize, mean = 0, sd = 8)

rater1categorical <- sample(c(0,1), size = sampleSize, replace = TRUE)
rater2categorical <- rater1categorical
rater3categorical <- rater1categorical

rater2categorical[sample(1:length(rater2categorical), size = 10, replace = FALSE)] <- (
```

```
rater3categorical[sample(1:length(rater3categorical), size = 10, replace = FALSE)] <- 1

time1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
time2 <- time1 + rnorm(n = sampleSize, mean = 0, sd = 4)
time3 <- time2 + rnorm(n = sampleSize, mean = 0, sd = 8)

item1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
item2 <- item1 + rnorm(n = sampleSize, mean = 0, sd = 4)
item3 <- item2 + rnorm(n = sampleSize, mean = 0, sd = 8)
item4 <- item3 + rnorm(n = sampleSize, mean = 0, sd = 12)

Person <- as.factor(rep(1:6, each = 8))
Occasion <- Rater <- as.factor(rep(1:2, each = 4, times = 6))
Item <- as.factor(rep(1:4, times = 12))
Score <- c(9,9,7,4,9,8,5,5,9,8,4,6,
           6,5,3,3,8,8,6,2,8,7,3,2,
           9,8,6,3,9,6,6,2,10,9,8,7,
           8,8,9,7,6,4,5,1,3,2,3,2)
```

3.4.2.2 Add Missing Data

Adding missing data to dataframes helps make examples more realistic to real-life data, and helps you get in the habit of programming to account for missing data.

```
rater1continuous[c(5,10)] <- NA
rater2continuous[c(10,15)] <- NA
rater3continuous[c(10)] <- NA

rater1categorical[c(5,10)] <- NA
rater2categorical[c(10,15)] <- NA
rater3categorical[c(10)] <- NA

time1[c(5,10)] <- NA
time2[c(10,15)] <- NA
time3[c(10)] <- NA

item1[c(5,10)] <- NA
item2[c(10,15)] <- NA
item3[c(10)] <- NA
item4[c(10)] <- NA
```

3.4.2.3 Combine data into data frame

```
mydata <- data.frame(rater1continuous, rater2continuous, rater3continuous,
                     rater1categorical, rater2categorical, rater3categorical,
                     time1, time2, time3,
                     item1, item2, item3, item4)

pio_cross_dat <- data.frame(Person, Item, Score, Occasion)
```

3.5 Types of Reliability

Reliability is not one thing. There are several types of reliability. In this book, I focus on test–retest, inter-rater, intra-rater, parallel-forms, and internal consistency reliability.

3.5.1 Test–Retest Reliability

Test–retest reliability is defined as the consistency of scores across time. Typically, this is based on a two-week retest interval. The intent of a two-week interval between the original testing and the retest is to provide adequate time to pass to reduce any carryover effects from the original testing while not allowing too much time to pass such that the person's level on the construct (i.e., true scores) would change. [**DEFINE CARRYOVER effects**] Examples of carryover effects resulting from repeated measurement can include fatigue, boredom, learning (practice effects), etc. Another potential issue is that measurement errors can be correlated across the two measurements.

Test–retest reliability controls for transient error and random response error. If the construct is not stable across time (i.e., people's true scores change), test–retest reliability is not relevant because the CTT approach to estimating reliability assumes that the true scores are perfectly correlated across time (see Section 3.1).

The length of the optimal retest interval depends on the construct of interest. For a construct in which people's levels change rapidly, a shorter retest interval may be appropriate. But one should pay attention to ways to reduce potential carryover effects. By contrast, if the retest interval is too long, people's levels on the construct may change during that span. If people's levels on the construct change from test to retest, we can no longer assume that the true scores are perfectly correlated across time, which would violate CTT assumptions for estimating test–retest reliability of a measure. The longer the retest interval, the smaller the observed association between scores across time will tend to be. For weak associations obtained from a lengthy retest interval, it can be difficult to

determine how much of this weak association reflects measurement unreliability versus people's change in their levels on the construct. Thus, when conducting studies to evaluate test–retest reliability, it is important to consider the length of the retest interval and ways to reduce carryover effects.

3.5.1.1 Coefficient of Stability (and Coefficient of Dependability)

The coefficient of stability is the most widely used when reporting the test–retest reliability of a measure. It is estimated using a Pearson correlation of the scores at time 1 with the score at time 2. That is, the coefficient of stability assesses the stability of individual differences. The Pearson correlation is called the coefficient of stability when the length of the retest interval (the delay between test and retest) is on the order of days or weeks. If the retest occurs almost at the same time as the original test (e.g., a 45-minute delay), the Pearson correlation is called the *coefficient of dependability* [Revelle and Condon, 2019].

I estimate the coefficient of stability below:

```
cor.test(x = mydata$time1, y = mydata$time2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydata$time1 and mydata$time2
## t = 25.009, df = 95, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8994368 0.9539169
## sample estimates:
##       cor
## 0.9317389
```

```
cor(mydata[,c("time1","time2","time3")], use = "pairwise.complete.obs")
```

```
##           time1     time2     time3
## time1 1.0000000 0.9317389 0.8290914
## time2 0.9317389 1.0000000 0.8425997
## time3 0.8290914 0.8425997 1.0000000
```

Below is a scatterplot of the time 1 scores on the x-axis and time 2 scores on the y-axis. The black line is the best-fitting linear line. The red line is a locally estimated scatterplot smoothing (LOESS) line, which uses non-parametric estimation of the best fit.

```
plot(mydata$time1, mydata$time2,
     main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x
abline(lm(time2 ~ time1, data = mydata), col = "black")
mydataNoMissing <- na.omit(mydata[,c("time1","time2")])
lines(lowess(mydataNoMissing$time1, mydataNoMissing$time2), col = "red")
```

3.5.1.1.1 Considerations about the correlation coefficient The correlation coefficient ranges from -1.0 to +1.0. The correlation coefficient (r) tells you two things: (1) the direction of the association (positive or negative), and (2) the magnitude of the association. If the correlation coefficient is positive, the association is positive. If the correlation coefficient is negative, the association is negative. If the association is positive, as X increases, Y increases (or conversely, as X decreases, Y decreases). If the association is negative, as X increases, Y decreases (or conversely, as X decreases, Y increases). The smaller the absolute value of the correlation coefficient (i.e., the closer the r value is to zero), the weaker the association, and the flatter the slope of the best-fit line in a scatterplot. The larger the absolute value of the correlation coefficient (i.e., the closer the absolute value of the r value is to one), the stronger the association, and the steeper the slope of the best-fit line in a scatterplot. See Figure 3.9 for a range of different correlation coefficients and what some example data may look like for each direction and strength of association.

Keep in mind that the Pearson correlation examines the strength of the *linear* association between two variables. If the association between two variables is

Figure 3.9: Correlation coefficients

non-linear, the Pearson correlation provides the strength of the linear trend and may not provide a meaningful index of the strength of the association between the variables. For instance, Anscombe's quartet includes four sets of data that have nearly identical basic descriptive statistics, including the same bivariate correlation, yet have very different distributions and whose association takes very different forms (see Figure 3.10).

Figure 3.10: Anscombe's Quartet

The coefficient of stability (and dependability) has important weaknesses. It is a form of *relative* reliability rather than *absolute* reliability: it examines the consistency of scores across time relative to variation across people. Higher stability coefficients reflect greater stability of individual differences—not greater stability in people's absolute level on the measure. This is a major limitation. Figure 3.11 depicts two example data sets that show strong relative reliability (strong

coefficient of stability) but poor absolute reliability based on inconsistency in people's absolute level across time.

Another limitation of the coefficient of stability (and dependability) is that it is sensitive to outliers. Additionally, if there are little or no individual differences in scores on a given measure, the coefficient of stability will be low relative to the true reliability because correlation coefficients tend to be attenuated in the presence of restricted range, and it may not be a useful index of reliability depending on the purpose. [**See section on cognitive reliability paradox**]. See Figure 3.12 for an example of how a correlation coefficient tends to be attenuated when the range of one or both of the variables has restriction of range.

Here is the correlation without and with restriction of range on x (i.e., x is restricted to values between 55 and 65):

```
par(mfrow = c(1,2))
par(mar = c(4, 0, 3, 0), oma = c(0, 4, 0, 0))

#Plot with no restricted range
plot(mydata$time1, mydata$time3,
     main = substitute(paste("No range restriction: ", italic(r), " = ", x, sep = ""), list(x = r
     xlab = "time1",
     ylab = "time3",
     xlim = c(25,85),
     ylim = c(25,85))
points(mydata$time1[which(mydata$time1 > 55 & mydata$time1 < 65)], mydata$time3[which(mydata$time
abline(lm(time3 ~ time1, data = mydata), col = "black")
mydataNoMissing <- na.omit(mydata[,c("time1","time3")])
lines(lowess(mydataNoMissing$time1, mydataNoMissing$time3), col = "red")

#Plot with restricted range
plot(mydata$time1[which(mydata$time1 > 55 & mydata$time1 < 65)], mydata$time3[which(mydata$time1
     main = substitute(paste("Restricted range for time1 (55-65): ", italic(r), " = ", x, sep = '
     xlab = "time1",
     ylab = "time3",
     xlim = c(25,85),
     ylim = c(25,85),
     yaxt = "n",
     pch = 19)
abline(lm(time3 ~ time1, data = mydata[which(mydata$time1 > 55 & mydata$time1 < 65),]), col = "bl
mydataNoMissing <- na.omit(mydata[,c("time1","time3")])
lines(lowess(mydataNoMissing$time1[which(mydataNoMissing$time1 > 55 & mydataNoMissing$time1 < 65)

mtext("time3", side = 2, outer = TRUE, line = 2.7, adj = 0.6)
```

The observed correlation became much weaker due to restriction of range. Thus,

Figure 3.11: Hypothetical data demonstrating good relative reliability despite poor absolute reliability. The figure depicts two fictional data sets (black and red circles) which both exhibit a similar linear association. The line of best fit is the solid line in the graph and is the same for both data sets, but the black circles sit much closer to the line than the red circles, leading to a much higher coefficient of stability (r = .99 and .84, respectively). However, neither sets of circles are on the line of complete agreement represented by the dashed line in the graph. If the circles fall on the line of complete agreement, it indicates that the measure's scores show consistency in absolute scores across time. Thus, although the measures show strong relative reliability despite poor absolute reliability. Figure reprinted from @Vaz2013, Figure 1, p. 3. Vaz, S., Falkmer, T., Passmore, A. E., Parsons, R., & Andreou, P. (2013). The case for using the repeatability coefficient when calculating test–retest reliability. *PLoS ONE*, *8*(9), e73990. https://doi.org/10.1371/journal.pone.0073990

Figure 3.12: Example of correlation without and with range restriction. Filled black points represent the points in common across the two scatterplots.

when developing measures, it is important to ensure there is adequate variability of scores (i.e., individual differences), and that scores are not truncated due to ceiling or floor effects. Moreover, when interpreting associations with truncated variability, it is important to keep in mind that the true association is likely to be even stronger than what was observed if the measures did not have restricted range.

To address these limitations of the coefficient of stability, it is important to consider additional indices of test–retest reliability, such as the coefficient of repeatability and bias that are depicted in Bland-Altman plots, as described below.

3.5.1.2 Coefficient of Repeatability

The coefficient of repeatability (CR), also known as the "smallest real difference", is a form of *absolute* reliability. That is, it reflects the consistency of scores across time within a person. It is calculated as ~1.96 times the standard deviation of the mean differences between the two measurements (d_1 and d_2): https://www.medcalc.org/manual/blandaltman.php [Bland and Altman, 1986, 1999]. The CR is used for determining the lower and upper limits of agreement because it defines the range within which 95% of differences between the two measurement methods are expected to be. The precise value is 1.959964 (not 1.96) times the standard deviation of the within-subject differences between the two measurements. This precise value is based on the fact that 95% of the area under a standard normal distribution lies within ± 1.959964 standard deviations of the mean. This is based on the quantile of the 2.5th (-1.959964) and 97.5th

(1.959964) percentile of the standard normal distribution (the 95% confidence interval spans the 92.5th percentile to the 97.5th percentile). However, some researchers have rounded 1.959964 to 1.96 or to 2 for simplicity.

The smaller the coefficient of repeatability, the more consistent the scores are across time within a person. The coefficient of repeatability is tied to the units of the measure, so how small is considered "good" absolute reliability depends on the measure. In general, however, given the same units of measurement, a smaller coefficient of repeatability indicates stronger absolute test–retest reliability.

$$CR = 1.959964 \times \sqrt{\frac{\sum(d_1 - d_2)^2}{n}}$$

```
coefficientOfRepeatability <- function(measure1, measure2){
  value <- qnorm(.975) * sd((measure1 - measure2), na.rm = TRUE)

  return(value)
}

coefficientOfRepeatability(measure1 = mydata$time1, measure2 = mydata$time2)
```

```
## [1] 7.694052
```

3.5.1.2.1 Bland-Altman Plot The Bland-Altman plot can be useful for visualizing the degree of (in)consistency of people's absolute scores on a measure across two time points [Bland and Altman, 1986, 1999].

In a Bland-Altman plot, the x-axis represents the mean of a person's score across the two time points, and the y-axis represents the within-subject differences across time points. The horizontal line at $y = 0$ is called the line of identity. Points would fall on the line of identity if people's scores showed perfect consistency in absolute level across time points. Another line reflects the bias (i.e., mean difference from the measure across the two points). The bias plus or minus the coefficient of repeatability is the upper or lower limit of agreement (LOA), respectively. The top dashed line reflects the upper LOA. The bottom dashed line reflects the lower LOA. The Bland-Altman plot was created using the `blandr` package [Datta, 2018].

```
BAstats <- blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)
BAplotlimits <- blandr.plot.limits(BAstats)
```

```
blandr.plot.rplot(BAstats, plot.limits = BAplotlimits, annotate = TRUE, method1name = '
```

Figure 3.13: Example of a Bland-Altman plot. Figure reprinted from @Vaz2013, Figure 2, p. 4. Vaz, S., Falkmer, T., Passmore, A. E., Parsons, R., & Andreou, P. (2013). The case for using the repeatability coefficient when calculating test–retest reliability. *PLoS ONE*, *8*(9), e73990. https://doi.org/10.1371/journal.pone.0073990

Figure 3.14: Bland-Altman Plot

3.5.1.2.2 Bias Bias (also known as mean error in the context of assessing accuracy against a criterion, see Section 7.4.1.1.1) is the average of the within-subject differences between the two measurements. Bias values closer to zero reflect stronger absolute test–retest reliability at the group-level. However, bias values can be closer to zero and individuals can still show large differences because large positive and negative differences could cancel each other out. As a result, it is also important to examine other indices of test–retest reliability, including the coefficient of stability and the coefficient of repeatability.

```
bias <- function(measure1, measure2){
  value <- mean((measure1 - measure2), na.rm = TRUE)

  return(value)
}

bias(measure1 = mydata$time1, measure2 = mydata$time2)
```

```
## [1] -0.3956087
```

3.5.1.2.3 95% Limits of Agreement

$$\text{Lower Limit of Agreement} = \text{Bias} - \text{Coefficient of Repeatability}$$

$$\text{Upper Limit of Agreement} = \text{Bias} + \text{Coefficient of Repeatability}$$

It is expected that the 95% limits of agreement (LOA) include 95% of differences between the two measurement methods (assuming the scores are normally distributed), based on how the coefficient of repeatability is calculated.

```
lowerLOA <- function(measure1, measure2){
  value <- bias(measure1, measure2) - coefficientOfRepeatability(measure1, measure2)

  return(value)
}

upperLOA <- function(measure1, measure2){
  value <- bias(measure1, measure2) + coefficientOfRepeatability(measure1, measure2)

  return(value)
}

lowerLOA(measure1 = mydata$time1, measure2 = mydata$time2)
```

```
## [1] -8.08966
```

```
upperLOA(measure1 = mydata$time1, measure2 = mydata$time2)
```

```
## [1] 7.298443
```

```
BAstats$regression.fixed.slope
```

3.5.1.2.4 Regression Slope

```
##  means
## -0.029
```

```
BAstats
```

3.5.1.2.5 Bland-Altman Statistics

```
## $means
##  [1] 45.41616 29.38352 52.83203 59.09428 40.81583 33.92801 44.04557 52.70680
##  [9] 50.69188 61.13508 70.10419 30.61419 51.06184 44.05256 56.42619 64.08919
## [17] 58.65712 44.11402 54.50209 49.15502 30.89027 41.80044 41.81681 47.21863
## [25] 73.31753 50.41762 48.49486 60.69893 48.78386 49.15547 43.90476 42.19101
## [33] 51.70235 62.21811 56.93890 41.84110 38.67391 50.27279 49.08995 53.13674
## [41] 60.21733 46.72160 45.54500 41.76403 64.76012 47.89258 53.22552 41.87576
## [49] 63.96757 82.95511 46.35012 42.05866 43.52036 29.87431 43.55178 58.83918
## [57] 73.75502 39.63434 53.81664 56.06827 63.31167 46.01927 45.15406 49.08199
## [65] 31.79205 62.95346 52.53782 57.54961 57.71031 50.58533 50.30490 52.04691
## [73] 62.87564 59.25578 37.97333 46.80365 54.84389 44.47204 59.39434 49.33871
## [81] 38.02790 30.65278 58.71295 45.42564 47.57603 49.65851 53.79797 53.24392
## [89] 36.97624 63.91611 67.82623 61.59119 61.52289 61.41113 60.83455 65.03694
## [97] 51.83620
##
## $differences
##  [1]   1.245020032  -2.377867165  -2.785315000  -1.289014150   4.301847437
##  [6]  -5.731637566   6.149992448  -6.494610948   2.457557157   0.774819695
## [11]  -3.554082677   0.186244917  -2.111806973   1.588909002   2.384188410
## [16]  -5.057180121  -0.647170814  -0.006229196  -5.598211508   0.281610645
## [21]  -1.694406962  -1.470900094  -5.036628171  -0.153792899   1.456101018
## [26]   2.990401115   5.388719805   0.816794054   6.119865945  -5.546951849
## [31]   6.373022139  -2.715330615   1.231221443   2.246139472   0.496967958
## [36]  -3.038138268  10.000195202   3.217102906  -7.473901982   3.011984913
## [41]  -1.555445860  -2.709198795  -4.827130509  -4.955453611   1.197794395
## [46]   5.118462434  -2.936569523   2.879365916   2.801251412   3.102180087
## [51]  -5.835451910   3.689356630   4.931280445   5.197238390   3.656180485
## [56]  -1.026930818  -3.197722284  -3.820555994  -3.729365286   5.206846348
## [61]   3.801191058  -3.376888132   1.035927654  -1.390297678  -3.174774919
## [66]  -9.122742915  -0.583252760 -11.644788772  -3.201798975  -5.349772139
## [71]   5.415966552  -2.719797446  -0.613921086  -3.850482165   4.375526613
## [76]   2.683634845  -2.072588509  -0.516727657  -1.558234634  -7.103018399
## [81]  -2.581420413   3.801986536  -2.187599244   0.509391773  -3.629211508
## [86]  -1.979410597   4.614920216   0.158893719  -3.296571501   4.697019503
## [91]  -6.197827372  -0.522025764  -0.949234099   2.507778726   1.593524369
## [96]   0.983768603  -0.052845246
##
## $method1
##   [1] 46.03867 28.19459 51.43937 58.44978       NA 42.96676 31.06219 47.12057
##   [9] 49.45949       NA 51.92066 61.52249 68.32715 30.70731 57.12481 50.00594
##  [17] 44.84701 57.61828 61.56060 58.33353 44.11091 51.70299 49.29583 30.04307
##  [25] 41.06499 39.29850 47.14174 74.04558 51.91283 51.18922 61.10732 51.84379
##  [33] 46.38200 47.09127 40.83335 52.31796 63.34118 57.18739 40.32203 43.67401
##  [41] 51.88135 45.35300 54.64273 59.43961 45.36700 43.13144 39.28630 65.35902
##  [49] 50.45181 51.75724 43.31545 65.36819 84.50620 43.43240 43.90334 45.98600
##  [57] 32.47293 45.37987 58.32571 72.15616 37.72407 51.95196 58.67169 65.21227
```

```
##  [65] 44.33083 45.67203 48.38684 30.20466 58.39209 52.24619 51.72721 56.10941
##  [73] 47.91044 53.01289 50.68701 62.56868 57.33054 40.16109 48.14547 53.80759
##  [81] 44.21368 58.61522 45.78720 36.73719 32.55377 57.61915 45.68033 45.76143
##  [89] 48.66880 56.10543 53.32337 35.32796 66.26462 64.72732 61.33018 61.04828
##  [97] 62.66501 61.63131 65.52882 51.80978
##
## $method2
##   [1] 44.79365 30.57245 54.22468 59.73879 58.54114 38.66491 36.79383 40.97057
##   [9] 55.95410       NA 49.46310 60.74767 71.88123 30.52107       NA 52.11775
##  [17] 43.25810 55.23409 66.61778 58.98071 44.11714 57.30120 49.01422 31.73748
##  [25] 42.53589 44.33512 47.29553 72.58948 48.92242 45.80050 60.29053 45.72393
##  [33] 51.92895 40.71825 43.54868 51.08674 61.09504 56.69042 43.36017 33.67381
##  [41] 48.66424 52.82690 51.63074 60.99505 48.07620 47.95857 44.24175 64.16122
##  [49] 45.33335 54.69381 40.43608 62.56694 81.40402 49.26785 40.21398 41.05472
##  [57] 27.27569 41.72369 59.35264 75.35388 41.54462 55.68133 53.46485 61.41108
##  [65] 47.70771 44.63610 49.77713 33.37944 67.51483 52.82944 63.37200 59.31121
##  [73] 53.26021 47.59692 53.40681 63.18260 61.18103 35.78557 45.46184 55.88018
##  [81] 44.73041 60.17346 52.89022 39.31861 28.75178 59.80675 45.17094 49.39064
##  [89] 50.64821 51.49051 53.16448 38.62453 61.56760 70.92515 61.85220 61.99751
##  [97] 60.15724 60.03778 64.54505 51.86262
##
## $sig.level
## [1] 0.95
##
## $sig.level.convert.to.z
## [1] 1.959964
##
## $bias
## [1] -0.3956087
##
## $biasUpperCI
## [1] 0.3955767
##
## $biasLowerCI
## [1] -1.186794
##
## $biasStdDev
## [1] 3.925609
##
## $biasSEM
## [1] 0.3985852
##
## $LOA_SEM
## [1] 0.6835169
##
## $upperLOA
```

```
## [1] 7.298585
##
## $upperLOA_upperCI
## [1] 8.655355
##
## $upperLOA_lowerCI
## [1] 5.941814
##
## $lowerLOA
## [1] -8.089802
##
## $lowerLOA_upperCI
## [1] -6.733032
##
## $lowerLOA_lowerCI
## [1] -9.446572
##
## $proportion
##  [1]   2.74135917  -8.09251988  -5.27202009  -2.18128396  10.53965340
##  [6] -16.89352667  13.96279479 -12.32215096   4.84802942   1.26738969
## [11]  -5.06971501   0.60836136  -4.13578293   3.60684842   4.22532252
## [16]  -7.89084742  -1.10331161  -0.01412067 -10.27155292   0.57290308
## [21]  -5.48524462  -3.51886261 -12.04450589  -0.32570384   1.98602032
## [26]   5.93126139  11.11194116   1.34564825  12.54485846 -11.28450470
## [31]  14.51556067  -6.43580309   2.38136471   3.61010561   0.87280915
## [36]  -7.26113318  25.85772864   6.39929192 -15.22491193   5.66836633
## [41]  -2.58305350  -5.79860030 -10.59859539 -11.86536345   1.84958638
## [46]  10.68738101  -5.51722061   6.87597227   4.37917439   3.73958912
## [51] -12.58993884   8.77193161  11.33097459  17.39701432   8.39502053
## [56]  -1.74531816  -4.33559963  -9.63950875  -6.92976203   9.28661851
## [61]   6.00393402  -7.33798726   2.29420701  -2.83260277  -9.98606605
## [66] -14.49124950  -1.11015795 -20.23434918  -5.54805396 -10.57573923
## [71]  10.76627940  -5.22566581  -0.97640533  -6.49806969  11.52263059
## [76]   5.73381499  -3.77906932  -1.16191572  -2.62354062 -14.39644225
## [81]  -6.78822746  12.40339952  -3.72592285   1.12137513  -7.62823504
## [86]  -3.98604547   8.57824215   0.29842602  -8.91537739   7.34872606
## [91]  -9.13780276  -0.84756563  -1.54289575   4.08359024   2.61943989
## [96]   1.51263059  -0.10194660
##
## $no.of.observations
## [1] 97
##
## $regression.equation
## [1] "y(differences) = -0.029 x(means) + 1.1"
##
## $regression.fixed.slope
```

```
##  means
## -0.029
##
## $regression.fixed.intercept
## (Intercept)
##         1.1
```

3.5.2 Inter-Rater Reliability

Inter-rater reliability is the consistency of scores across raters. For instance, two different raters may make a diagnostic decision for the same client, two different people may provide observational ratings of the same participant, or two different researchers may process psychophysiological data (e.g., electroencephalography, electrocardiography) from the same person. Consistency of scores can also be assessed within the same rater, which is known as *intra*-rater reliability and is described in Section 3.5.3. For assessing inter-rater reliability, intraclass correlation (ICC) is recommended for continuous data (e.g., ratings of a participant's mood), whereas Cohen's kappa (κ) is recommended for categorical variables (e.g., making a diagnostic decision whether to give someone a diagnosis of depression or not).

3.5.2.1 Intraclass Correlation (ICC)

Use intraclass correlation (ICC) for estimating inter-rater reliability of continuous data. ICC was estimated using the `psych` package [Revelle, 2020].

```
ICC(mydata[,c("rater1continuous","rater2continuous","rater3continuous")], missing = FALSE)
```

```
## Call: ICC(x = mydata[, c("rater1continuous", "rater2continuous", "rater3continuous")],
##     missing = FALSE)
##
## Intraclass correlation coefficients
##                          type  ICC  F df1 df2       p lower bound upper bound
## Single_raters_absolute   ICC1 0.81 14  99 200 2.7e-54        0.76        0.86
## Single_random_raters     ICC2 0.81 14  99 198 6.4e-54        0.76        0.86
## Single_fixed_raters      ICC3 0.81 14  99 198 6.4e-54        0.76        0.86
## Average_raters_absolute ICC1k 0.93 14  99 200 2.7e-54        0.91        0.95
## Average_random_raters   ICC2k 0.93 14  99 198 6.4e-54        0.91        0.95
## Average_fixed_raters    ICC3k 0.93 14  99 198 6.4e-54        0.91        0.95
##
##  Number of subjects = 100     Number of Judges =  3
```

3.5.2.2 Cohen's Kappa

Use Cohen's kappa for estimating inter-rater reliability of categorical data [Bakeman and Goodman, 2020]. Cohen's kappa was estimated using the `psych` package [Revelle, 2020].

```
cohen.kappa(mydata[,c("rater1categorical","rater2categorical","rater3categorical")])
```

```
##
## Cohen Kappa (below the diagonal) and Weighted Kappa (above the diagonal)
## For confidence intervals and detail print with all=TRUE
##                   rater1categorical rater2categorical rater3categorical
## rater1categorical              1.00              0.86              0.90
## rater2categorical              0.86              1.00              0.76
## rater3categorical              0.90              0.76              1.00
##
## Average Cohen kappa for all raters  0.84
## Average weighted kappa for all raters  0.84
```

3.5.3 Intra-Rater Reliability

Principles of intra-rater are similar to principles of inter-rater reliability (described in Section 3.5.2), except the *same* rater provides ratings on two occasions, rather than using multiple raters.

3.5.4 Parallel- (or Alternate-) Forms Reliability

Parallel-forms reliability (also known as alternate-forms reliability) is the consistency of scores across two parallel forms. Parallel forms are two equivalent measures of the same construct that differ in content or format. The two measures can be administered in the same occasion, known as *immediate parallel-forms reliability*, or they can be administered separated by a delay, known as *delayed parallel-forms reliability*. Similar considerations of length of delay duration and ways to reduce carryover effects are relevant to parallel-forms reliability as they are to test–retest reliability.

Parallel-forms reliability is often estimated with the Pearson correlation, which is known as the *coefficient of equivalence*. The coefficient of equivalence is the consistency of scores across two parallel forms relative to variation across people. It is interpreted as the ratio of true score variance to observed score variance, consistent with the formal definition of reliability.

An example of parallel-forms reliability would be if we want to assess a participant twice with different measures of the same construct to avoid practice

effects. This approach is often conducted in academic achievement testing, and it could involve retesting that is immediate or delayed. For instance, we could administer item set A at time 1 and item set B at time 2 for the same participant to get rid of any possibility that the person's score differed across time because of improved performance that resulted merely from prior exposure to the same items. Many standardized academic achievement and aptitude tests have developed parallel forms, including the the SAT, ACT, and GRE.

Parallel-forms reliability controls for specific error, i.e., error that is particular to a specific measure. However, parallel-forms reliability has the limitation that it assumes that the parallel forms are equivalent, such that it makes no difference which test you use. More specifically, two forms are considered equivalent when they assess the same construct, have the same mean and variance of scores, and have the same inter-correlations with external criteria. Though, technically, two instruments are only truly equivalent if they have the same true scores and variability of errors. Given these constraints, it can be difficult to assume that different forms are equivalent. Nevertheless, integrative data analysis and measurement harmonization may help make the assumption more tenable [Hussong et al., 2020, 2013]. In general, parallel forms can be difficult to create and they can more than double the time it takes to develop a single measure, often with limited benefit, depending on the intended use.

3.5.5 Internal Consistency Reliability

Internal consistency reliability is the consistency of scores across items, that is, their inter-relatedness. Internal consistency is necessary, but insufficient, for establishing the homogeneity or unidimensionality of the measure, i.e., that the items on the measure assess the same construct.

There are many examples of internal consistency estimates, including Cronbach's coefficient alpha (α), Kuder-Richardson Formula 20 (which is a special case of Cronbach's alpha with dichotomous items), omega coefficients, greatest lower bound, coefficient H, split-half reliability, average inter–item correlation, and average item–total correlation.

Internal consistency can be affected by many factors, such as:

- Wording of the item
- Participant's comprehension of the item, which could be impacted if using items with dated language or double negatives
- Interpretation
- Attention of the participant
- Compliance of the participant

Although establishing internal consistency can help ensure that items in the measure are inter-related, you do not want to select items just based on internal

consistency. If you did, you would end up with items that are too similar. For example, you could end up with only with items that have similar item stems and are coded in the same direction. Therefore, selecting items solely based on internal consistency would likely not assess the breadth of the construct, and would reflect strong method variance of the wording of the particular items. In other words, internal consistency can be *too* high. Internal consistency is helpful to a point, but you do not want items to be too homogeneous or redundant.

3.5.5.1 Average Inter-Item Correlation

One estimate of internal consistency is the average (mean or median) inter-item correlaton.

```
interItemCorrelations <- cor(mydata[,c("item1","item2","item3","item4")], use = "pairwi
diag(interItemCorrelations) <- NA
```

Here is the mean inter-item correlation:

```
meanInterItemCorrelations <- colMeans(interItemCorrelations, na.rm = TRUE)
meanInterItemCorrelations
```

```
##     item1     item2     item3     item4
## 0.7748087 0.7787348 0.7538928 0.6594609
```

```
mean(meanInterItemCorrelations)
```

```
## [1] 0.7417243
```

```
psych::alpha(mydata[,c("item1","item2","item3","item4")])$total$average_r
```

```
## [1] 0.7452872
```

```
hist(meanInterItemCorrelations)
abline(v = mean(meanInterItemCorrelations), lwd = 2)
```

Here is the median inter-item correlation:

```
medianInterItemCorrelations <- apply(interItemCorrelations, 2, function(x) median(x, na.rm = TRUE
medianInterItemCorrelations
```

```
##     item1     item2     item3     item4
## 0.7577984 0.7791957 0.7577984 0.6316587
```

```
median(interItemCorrelations[lower.tri(interItemCorrelations)])
```

```
## [1] 0.7412414
```

```
psych::alpha(mydata[,c("item1","item2","item3","item4")])$total$median_r
```

```
## [1] 0.7429489
```

3.5.5.2 Average Item–Total Correlation

Another estimate of internal consistency is the average item–total correlation.

```
itemTotalCorrelations <- psych::alpha(mydata[,c("item1","item2","item3","item4")])$item.stats["ra

mean(itemTotalCorrelations$raw.r)
```

```
## [1] 0.8940236
```

When examining the item–total correlation for a given item, it is preferable to exclude the item of interest from the total score, so that the item is associated with the total score that is calculated from the remaining items. This reduces the extent to which the association reflects that the item is associated with itself.

```
itemTotalCorrelationsDrop <- psych::alpha(mydata[,c("item1","item2","item3","item4")])$

mean(itemTotalCorrelationsDrop$r.drop)
```

```
## [1] 0.8053408
```

3.5.5.3 Split-half Reliability

When estimating split-half reliability, we randomly take half of the items on the measure, and relate scores on these items to the score on the other half of items. One challenge to split-half reliability is that, according to classical test theory, shorter tests tend to be less reliable than longer tests, so this would be an underestimate of the test's reliability. In general, the greater the number of items, the more reliable the test. To overcome this, the Spearman-Brown prediction (or prophecy) formula predicts the reliability of a test after changing the test length. Thus, the Spearman-Brown prediction formula accounts for the shorter test in split-half reliability in predicting what the reliability of the full-length test would be.

Here is the Spearman-Brown prediction formula:

$$\text{predicted test reliability, } r_{xx} = \frac{n \cdot r_{xx'}}{1 + (n-1)r_{xx'}}$$

where n = the number of tests combined (i.e., the ratio of the number of new items to the number of old items), and $r_{xx'}$ = the reliability of the original test.

Another challenge to split-half reliability is that the reliability differs depending on which items go into which half. Modern computational software allows for estimating the mean reliability estimate of all possible split-halves [Revelle and Condon, 2019], while correcting for test length, as seen below. Split-half reliability was estimated using the `psych` package [Revelle, 2020].

```
splitHalf(mydata[,c("item1","item2","item3","item4")], brute = TRUE)
```

```
## Split half reliabilities
```

```
## Call: splitHalf(r = mydata[, c("item1", "item2", "item3", "item4")],
##     brute = TRUE)
##
## Maximum split half reliability (lambda 4) =  0.95
## Guttman lambda 6                          =  0.93
## Average split half reliability            =  0.92
## Guttman lambda 3 (alpha)                  =  0.92
## Guttman lambda 2                          =  0.92
## Minimum split half reliability  (beta)    =  0.87
## Average interitem r =  0.75  with median =  0.74
```

3.5.5.4 Cronbach's Alpha (Coefficient Alpha)

Cronbach's alpha (α), also known as coefficient alpha, is approximately equal to the mean reliability estimate of all split-halves, assuming that item standard deviations are equal. Alpha considers the inter-relatedness of a total set of items—giving us information about the extent to which each item in a set of items correlates with all other items. Alpha is the most commonly used for internal consistency rekiability, but it has many problems.

Cronbach's alpha assumes that the items are: - uni-dimensional: that there is one predominant dimension that explains the variance among the items - tau (τ) equivalent: that the items are equally related to the construct

Cronbach's alpha is affected by:

- The number of items: the more items, the more inflated alpha is, even if correlations among the items are low.
- Variance of the item scores: if the variances of item scores are low (e.g., dichotomous scores), alphas tend to be an underestimate of internal consistency reliability.
- Violations of assumptions:
  - Cronbach's alpha assumes tau equivalence—that the items are equally related to the construct. However, this is an unrealistic assumption.
  - Cronbach's alpha depends on the dimensionality of the scale: alpha is only valid when the scale is uni-dimensional. If you use it with multidimensional scales, you know the extent to which items correlate with another, but nothing about whether they are actually measuring the same constructs.

Cronbach's alpha is appropriate only in situations where a scale is unidimensional, where the items equally contribute to the scale, and you would like to determine whether its items are inter-related. However, Cronbach's alpha does

not assess the dimensionality of the scale or any type construct validity—it can only tell us how strongly the items hang together.

Cronbach's alpha (α) was estimated using the `psych` package [Revelle, 2020].

```
psych::alpha(mydata[,c("item1","item2","item3","item4")])$total$raw_alpha
```

```
## [1] 0.8923861
```

Standardized alpha (if the items are standardized before computing the scale score).

```
psych::alpha(mydata[,c("item1","item2","item3","item4")])$total$std.alpha
```

```
## [1] 0.9212844
```

Item statistics:

```
psych::alpha(mydata[,c("item1","item2","item3","item4")])
```

```
##
## Reliability analysis
## Call: psych::alpha(x = mydata[, c("item1", "item2", "item3", "item4")])
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean sd median_r
##       0.89      0.92    0.93      0.75  12 0.017   49 11     0.74
##
##  lower alpha upper     95% confidence boundaries
## 0.86 0.89 0.93
##
##  Reliability if an item is dropped:
##       raw_alpha std.alpha G6(smc) average_r  S/N alpha se  var.r med.r
## item1      0.85      0.88    0.84      0.71  7.4    0.024 0.0064  0.72
## item2      0.85      0.88    0.84      0.71  7.2    0.024 0.0042  0.72
## item3      0.83      0.89    0.89      0.73  8.3    0.029 0.0331  0.63
## item4      0.93      0.94    0.93      0.83 14.6    0.014 0.0101  0.78
##
##  Item statistics
##        n raw.r std.r r.cor r.drop mean sd
## item1 98  0.89  0.93  0.93   0.84   49 10
## item2 98  0.90  0.93  0.94   0.84   50 11
## item3 99  0.91  0.91  0.86   0.83   50 13
## item4 99  0.87  0.83  0.73   0.71   46 18
```

However, because of the many weaknesses of Cronbach's alpha [Cortina, 1993; Dunn et al., 2014; Flora, 2020; Green and Yang, 2015; Hayes and Coutts, 2020; Kelley and Pornprasertmanit, 2016; McNeish, 2018; Peters, 2014; Raykov, 2001; Revelle and Condon, 2019; Sijtsma, 2008; but see Raykov and Marcoulides, 2019], current recommendations are to use coefficient omega (ω) instead, as described next.

3.5.5.5 Coefficient Omega (ω)

Current recommendations are to use McDonald's coefficient omega (ω) instead of Cronbach's alpha (α) for internal consistency reliability. Specifically, it is recommended to use omega total (ω_t) for continuous items that are unidimensional, omega hierarchical (ω_h) for continuous items that are multi-dimensional, and omega categorical (ω_C) for categorical items (i.e., items with fewer than five ordinal categories), with confidence intervals calculated from a bias-corrected bootstrap [Flora, 2020, Kelley and Pornprasertmanit, 2016].

3.5.5.5.1 Omega Total (ω_t) Omega total (ω_t) (or simply omega) is the proportion of the total variance that is accounted for by the common factor [Flora, 2020]. Omega total, like Cronbach's alpha, assumes that the items are uni-dimensional. Omega total (ω_t) was calculated using the `MBESS` package [Kelley, 2020]. This code quickly calculates a point estimate for omega total:

```
ci.reliability(mydata[,c("item1","item2","item3","item4")], type = "omega", interval.type = "none
```

```
## [1] 0.8760799
```

If you want to calculate a confidence interval for omega, you will need to specify an `interval.type` and specify how many bootstrap replications.

```
bootstrapReplications <- 1000
```

The following code calculates a confidence interval from a bias-corrected bootstrap. For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed. **Warning**: this code takes a while to run based on 1000 bootstrap replications. You can reduce the number of replications to be faster. However, the number of bootstrap replications must be larger than the number of rows in the data.

```
set.seed(52242)
ci.reliability(mydata[,c("item1","item2","item3","item4")], type = "omega", interval.type = "bca"
```

```
## $est
## [1] 0.8760799
##
## $se
## [1] 0.02461448
##
## $ci.lower
## [1] 0.81431
##
## $ci.upper
## [1] 0.9149732
##
## $conf.level
## [1] 0.95
##
## $type
## [1] "omega"
##
## $interval.type
## [1] "bca bootstrap"
```

3.5.5.5.2 Omega Hierarchical (ω_h) If the items are multi-dimensional and follow a bifactor structure, omega hierarchical (ω_h) is preferred over omega total. Omega hierarchical is the proportion of the total variance that is accounted for by the general factor in set of multi-dimensional items [Flora, 2020]. Use omega hierarchical for estimating internal consistency of continuous items that are multi-dimensional and follow a bifactor structure. If the multi-dimensional items follow a higher-order structure, use omega higher-order [ω_{ho}; Flora [2020]].

Omega hierarchical was calculated using the `MBESS` package [Kelley, 2020].

```
ci.reliability(mydata[,c("item1","item2","item3","item4")], type = "hierarchical", int
```

```
## [1] 0.8453203
```

The following code calculates a confidence interval from a bias-corrected bootstrap. For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed. **Warning**: this code takes a while to run based on 1000 bootstrap replications. You can reduce the number of replications to be faster. However, the number of bootstrap replications must be larger than the number of rows in the data.

```
set.seed(52242)
ci.reliability(mydata[,c("item1","item2","item3","item4")], type = "hierarchical", interval.type
```

```
## $est
## [1] 0.8453203
##
## $se
## [1] 0.03381731
##
## $ci.lower
## [1] 0.7641922
##
## $ci.upper
## [1] 0.8972227
##
## $conf.level
## [1] 0.95
##
## $type
## [1] "hierarchical omega"
##
## $interval.type
## [1] "bca bootstrap"
```

3.5.5.5.3 Omega Categorical (ω_C) Use omega categorical (ω_C) for categorical items that are uni-dimensional. To handle categorical data, omega categorical is calculated using a polychoric correlation matrix instead of a Pearson correlation matrix. If the categorical items are multi-dimensional items, and they follow a bifactor structure, use omega hierarchical categorical [$\omega_{h\text{-cat}}$; Flora [2020]]. If the categorical items are multi-dimensional, and they follow a higher-order structure, use omega higher-order categorical [$\omega_{ho\text{-cat}}$; Flora [2020]].

Omega categorical was calculated using the `MBESS` package [Kelley, 2020].

```
ci.reliability(mydata[,c("item1","item2","item3","item4")], type = "categorical", interval.type =
```

```
## [1] NA
```

The following code calculates a confidence interval from a bias-corrected bootstrap. For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed. **Warning**: this code takes a while to run based on 1000 bootstrap replications. You can reduce the number of replications to be faster. However, the number of bootstrap replications must be larger than the number of rows in the data.

```
set.seed(52242)
ci.reliability(mydata[,c("rater1categorical","rater2categorical","rater3categorical")]
```

```
## $est
## [1] 0.9600709
##
## $se
## [1] 0.009518443
##
## $ci.lower
## [1] 0.9342998
##
## $ci.upper
## [1] 0.9742486
##
## $conf.level
## [1] 0.95
##
## $type
## [1] "categorical omega"
##
## $interval.type
## [1] "bca bootstrap"
```

3.5.6 Comparison

All of types of reliability described in Section 3.5 are aspects of reliability. Even thought they are all called "reliability" they are very different from each other. Test–retest reliability and inter-rater reliability tend to be lower than immediate parallel-forms reliability and internal consistency reliability because they involve administering measures at different times or with different raters. However, delayed parallel-forms reliability tends to be lower than test–retest reliability because it involves assessing the construct at a different time point *and* with different items. Thus, if we put the different types of reliability in order from low to high, they tend to be (but not always) arranged like this:

$$\text{delayed parallel-forms} < \text{test–retest}, \text{inter-rater} < \text{immediate parallel-forms}, \text{internal consistency}$$

3.5.7 Reliability of a Linear Composite Score

A linear composite score is a linear combination of variables, each of which has a given weight in the linear composite. Unit weights (i.e., weights of 1) would reflect an unweighted composite.

We first specify the variables used to generate the correlation matrix, and the reliability, standard deviation, and weight of each variable. We specify items 2 and 3 to have twice the weight as item 1 in the composite, and item 4 to have 1.5 times the weight as item 1 in the composite.

```
rxx <- c(0.70, 0.75, 0.80, 0.85) #reliability coefficients
rho <- cor(mydata[,c("item1","item2","item3","item4")], use = "pairwise.complete.obs")
sigma <- apply(mydata[,c("item1","item2","item3","item4")], 2, function(x) sd(x, na.rm = TRUE)) #
weights <- c(1, 2, 2, 1.5) #weights of each variable in the linear combination
```

Here is the reliability of an unweighted composite:

```
composite_rel_matrix(rel_vec = rxx, r_mat = rho, sd_vec = sigma)
```

```
## [1] 0.9337237
```

Here is the reliability of a weighted composite:

```
composite_rel_matrix(rel_vec = rxx, r_mat = rho, sd_vec = sigma, wt_vec = weights)
```

```
## [1] 0.931709
```

3.5.8 Reliability of a Difference Score

Difference scores tend to be lower in reliability than the individual indices that compose the difference score. This is especially the case when the two indices are correlated. If the correlation between the indices is equal to their reliability of the indices, the reliability of the difference score will be zero [Revelle and Condon, 2019]. If the correlation between the two indices is large, it requires a very high reliability for the difference score to show a strong reliability [Trafimow, 2015]. To the extent that the two indices have unequal variances (e.g., if one index has a standard deviation that is four times the standard deviation of the second index), the reliabilities do not need to be quite as high for the difference score to be reliable [Trafimow, 2015].

The reliability of a difference score can be calculated using the following formula [Trafimow, 2015]:

$$r_{x-y,x-y} = \frac{\sigma_x^2 r_{xx} + \sigma_y^2 r_{yy} - 2r_{xy}\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 - 2r_{xy}\sigma_x\sigma_y}$$

Here is a function to calculate the reliability of a difference score:

```
reliabilityOfDifferenceScore <- function(x, y, reliabilityX, reliabilityY){
  rxy <- as.numeric(cor.test(x = x, y = y)$estimate)
  sigmaX <- sd(x, na.rm = TRUE)
  sigmaY <- sd(y, na.rm = TRUE)

  reliability <- (sigmaX^2 * reliabilityX + sigmaY^2 * reliabilityY - (2 * rxy * sigma

  return(reliability)
}

reliabilityOfDifferenceScore(x = mydata$item1, y = mydata$item2, reliabilityX = .95, re
```

```
## [1] 0.2552898
```

For instance, in this case, the reliability of the difference score is .26 even though the reliability of both indices is .95.

In Figures 3.15 and 3.16, I present the reliability of a difference score as a function of (a) the reliability of the indices that compose the difference score and (b) the correlation between them, assuming that both indices have equal variances.

Figure 3.15: Reliability of Difference Score as a Function of Reliability of Indices and the Correlation between Them

Figure 3.16: Reliability of Difference Score as a Function of Correlation between Indices and Reliability of Indices

3.6 Applied Examples

Although reliability is an important consideration for measures, stronger reliability is not always better. Reliability is not one thing; it is not monolithic. There are multiple aspects of reliability, and which aspect(s) of reliability are most important depend on the construct of interest. That is, we want different kinds of reliability for measures of different constructs. In general, we want our measures to show strong inter-rater, intra-rater, immediate parallel-forms, and internal consistency reliability. However, the extent to which our measures should show test–retest and delayed parallel-forms reliability depends on the stability of the construct and the time lag.

Consider the construct of reactive dysphoria. Let us assume that we assess people's reactive dysphoria with observations and several questionnaires today and one month from now. However, test–retest reliability is not meaningful because the phenomenon is not meant to be stable across time—it is supposed to be reactive to the situation, not a trait. Instead, we would expect that a strong measure of reactive dysphoria has high internal consistency, a high coefficient of equivalence from immediate parallel-forms reliability, and high inter-/intra-rater reliability.

Consider the example of personality and intelligence. Personality is defined such that personality "traits" are thought to be stable across time and situa-

tion. Likewise, individual differences in intelligence are thought to be relatively stable (not in absolute level). Thus, strong measures of personality and intelligence should show high test–retest reliability, parallel-forms reliability, internal consistency, and inter-/intra-rater reliability.

Just because one aspect of reliability is strong does not necessarily mean that other other aspects of reliability will be strong. Measures of some constructs (e.g., depression) could be expected to show low test–retest reliability but high internal consistency. You can also have measures that have low internal consistency but that have high test–retest reliability (e.g., if our nonsensical measure averaged together z-scores of height and blood type).

Consider a questionnaire assessing people's extraversion. We would expect high stability of construct, and therefore a high coefficient of stability, but test and retest scores might not be equal (somewhat lower repeatability coefficient) because of random error—on a given day, they may be sick, stressed, or tired, they may have recently got into an argument, or they may be impacted by the season, the weather, or differneces in the context (e.g. lab or online), just to name a few sources of random error.

3.7 Standard Error of Measurement (SEM)

We presented and discussed the formula for calculating the standard error of measurement earlier in Equation (3.3.2). It is also below:

$$\text{standard error of measurement (SEM)} = \sigma_x \sqrt{1 - r_{xx}}$$

where σ_x = standard deviation of scores on measure x and r_{xx} = reliability of measure x

The reliability of the measure was estimated here based on the test–retest reliability of the measure at T1 with the measure at T2.

```
reliabilityOfMeasure <- as.numeric(cor.test(x = mydata$time1, y = mydata$time2)$estima
sem <- sd(mydata$time1, na.rm = TRUE) * sqrt(1 - reliabilityOfMeasure)
sem
```

```
## [1] 2.720493
```

95% of the time, the true score is expected to fall within ± 1.959964 SEMs of the observed score.

```
lower95CI <- mean(mydata$time1, na.rm = TRUE) - qnorm(.975)*sem
upper95CI <- mean(mydata$time1, na.rm = TRUE) + qnorm(.975)*sem

lower95CI
```

```
## [1] 45.60439
```

```
upper95CI
```

```
## [1] 56.26852
```

The 95% CI is: $[45.6, 56.27]$.

3.8 Influences of Measurement Error on Test–Retest Reliability

This section describes the influence of measurement error on reliability, specifically test–retest reliability. There are different types of measurement error, and each can have different impacts on reliability and, ultimately, your findings. As discussed in Section 3.2, errors can be either systematic or random. In addition, error can be within-person, between-person, or both. Let us consider two data columns, one column for participants' scores at time 1 and another column for participants' scores at time 2. Adding within-person error would mean adding noise (e) within the given row (or rows) for the relevant participant(s). Adding between-person error would mean adding noise (e) across the rows within the column. All types of error make each individual measurement for a person less accurate. But, different kinds of errors have different implications.

Types of measurement error

[Show Figure from Willett et al. (2012)]

Figure 3.17

3.8.1 No Error

This section provides the coefficient of stability, repeatability, and bias that are used as a comparison to when measurement error is added. For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
#Simulate Data
set.seed(52242)

time1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
time2 <- time1 + rnorm(n = sampleSize, mean = 0, sd = 4)
time3 <- time2 + rnorm(n = sampleSize, mean = 0, sd = 8)
```

Figure 3.17: Types of measurement error.

```
#Add Missing Data
time1[c(5,10)] <- NA
time2[c(10,15)] <- NA
time3[c(10)] <- NA

#Combine data into data frame
mydata <- data.frame(rater1continuous, rater2continuous, rater3continuous,
                     rater1categorical, rater2categorical, rater3categorical,
                     time1, time2, time3,
                     item1, item2, item3, item4)
```

3.8.1.1 Coefficient of stability

```
cor.test(x = mydata$time1, y = mydata$time2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydata$time1 and mydata$time2
## t = 28.184, df = 95, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.9188254 0.9630063
## sample estimates:
##       cor
## 0.9450797
```

3.8.1.2 Coefficient of repeatability

```
qnorm(.975) * sd(blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$differences, r
```

```
## [1] 6.588758
```

3.8.1.3 Bias

```
blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$bias
```

```
## [1] 0.335823
```

3.8.2 Add within-person random error

Adding within-person random error would involve adding random noise (e) to the given row (or rows) for the relevant participant(s). This could reflect momentary fluctuations in the assessment for a specific person. When adding within-person random error, the person's and group's measurements show no bias, i.e., there is no consistent increase or decrease in the scores from time 1 to time 2 (at least with a sample size large enough to cancel out the random error, according to the law of large numbers). A person's average approximates their true score if many repeated measurements are taken. A group's average approximates the sample mean's true score, especially when averaging the repeated measures across time.

Figure 3.18

In the example below, I added within-person random error. Specifically, I added error that is randomly distributed around a mean of 0 and a standard deviation of 3 to all participants' scores at all timepoints.

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

						Person-Level			
	Participant	True Score	T1	T2	Influence	Bias at T1	Bias at T2	Mean of T1 and T2	Mean-Level Bias
	1	70	80	60	Add random error (+10 to T1, -10 to T2)	10	-10	70	0
	2	70	70	70		0	0	70	0
	3	80	80	80		0	0	80	0
	4	80	80	80		0	0	80	0
	5	90	90	90		0	0	90	0
	6	90	90	90		0	0	90	0
	7	100	100	100		0	0	100	0
	8	100	100	100		0	0	100	0
	9	110	110	110		0	0	110	0
	10	110	110	110		0	0	110	0
Group-Level	Mean	90	91	89		1	-1	90	0
	Variance	222.22	187.78	276.67				222.22	

Figure 3.18: Within-person random error.

```
#Simulate Data
set.seed(52242)

time1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
time2 <- time1 + rnorm(n = sampleSize, mean = 0, sd = 4)
time3 <- time2 + rnorm(n = sampleSize, mean = 0, sd = 8)

#Add Missing Data
time1[c(5,10)] <- NA
time2[c(10,15)] <- NA
time3[c(10)] <- NA

#Combine data into data frame
mydata <- data.frame(rater1continuous, rater2continuous, rater3continuous,
                     rater1categorical, rater2categorical, rater3categorical,
                     time1, time2, time3,
                     item1, item2, item3, item4)

#Add error
set.seed(12345)
for(i in 1:nrow(mydata)){
  mydata[i, c("time1","time2","time3")] <- mydata[i, c("time1","time2","time3")] + rno
}
```

3.8.2.1 Coefficient of stability

```
cor.test(x = mydata$time1, y = mydata$time2)
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  mydata$time1 and mydata$time2
## t = 18.782, df = 95, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8361773 0.9235489
## sample estimates:
##       cor
## 0.8875967
```

3.8.2.2 Coefficient of repeatability

```
qnorm(.975) * sd(blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$differences, r
```

```
## [1] 10.03618
```

3.8.2.3 Bias

```
blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$bias
```

```
## [1] 0.5901701
```

3.8.2.4 Findings

Adding within-person random error led to:

- weaker (lower) stability coefficient
- weaker (higher) repeatability coefficient

3.8.3 Add between-person random error at T2

Adding between-person random error at time 2 would involve adding random noise (e) across the rows, within the column. Between-person random error would result in bias at the person-level but not at the group-level. It results in over-estimates of the person's true score for some people and under-estimates for other people, i.e., there is no consistent pattern across the sample. Thus, the group average approximates the sample's mean true score (at least with a sample size large enough to cancel out the random error, according to the law of large numbers). However, the group's variance is inflated.

						Person-Level			
	Participant	True Score	T1	T2	Influence	Bias at T1	Bias at T2	Mean of T1 and T2	Mean-Level Bias
	1	70	70	75	Add random error at T2 (5)	0	5	72.50	2.50
	2	70	70	65	Add random error at T2 (-5)	0	-5	67.50	-2.50
	3	80	80	83	Add random error at T2 (3)	0	3	81.50	1.50
	4	80	80	77	Add random error at T2 (-3)	0	-3	78.50	-1.50
	5	90	90	96	Add random error at T2 (6)	0	6	93	3
	6	90	90	84	Add random error at T2 (-6)	0	-6	87	-3
	7	100	100	110	Add random error at T2 (10)	0	10	105	5
	8	100	100	90	Add random error at T2 (-10)	0	-10	95	-5
	9	110	110	112	Add random error at T2 (2)	0	2	111	1
	10	110	110	108	Add random error at T2 (-2)	0	-2	109	-1
Group-Level	Mean	90	90	90		0	0	90	0
	Variance	222.22	222.22	260.89				231.89	

Figure 3.19: Between-person random error.

Figure 3.19

In the example below, I added between-person random error. Specifically, I added error that is randomly distributed around a mean of 0 and a standard deviation of 3 to all participants' scores at time 2:

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
#Simulate Data
set.seed(52242)

time1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
time2 <- time1 + rnorm(n = sampleSize, mean = 0, sd = 4)
time3 <- time2 + rnorm(n = sampleSize, mean = 0, sd = 8)

#Add Missing Data
time1[c(5,10)] <- NA
time2[c(10,15)] <- NA
time3[c(10)] <- NA

#Combine data into data frame
mydata <- data.frame(rater1continuous, rater2continuous, rater3continuous,
                     rater1categorical, rater2categorical, rater3categorical,
                     time1, time2, time3,
                     item1, item2, item3, item4)

#Add error
set.seed(12345)
mydata$time2 <- mydata$time2 + rnorm(sampleSize, mean = 0, sd = 3)
```

3.8.3.1 Coefficient of stability

```
cor.test(x = mydata$time1, y = mydata$time2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydata$time1 and mydata$time2
## t = 19.367, df = 95, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8442114 0.9274674
## sample estimates:
##       cor
## 0.8932557
```

3.8.3.2 Coefficient of repeatability

```
qnorm(.975) * sd(blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$differences, r
```

```
## [1] 9.288859
```

3.8.3.3 Bias

```
blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$bias
```

```
## [1] -0.4554251
```

3.8.3.4 Findings

Adding between-person random error led to:

- weaker (lower) stability coefficient
- weaker (higher) repeatability coefficient
- bias at the person-level, but this bias would go undetected by the estimate of group-level bias in the Bland-Altman plot

3.8.4 Add within-person systematic error

Adding within-person systematic error would involve adding systematic noise (e) (the same variance across columns) to the given row (or rows), reflecting the relevant participant(s). These are within-person effects that are consistent across time. For example, social desirability bias is high for some people and low for others. Another instance in which within-person systematic error could have is of one or more people consistently misinterpret a particular question. Within-person systematic error increases person-level bias because the person's mean shows a greater difference from their true score.

Figure 3.20

						Person-Level			
	Participant	True Score	T1	T2	Influence	Bias at T1	Bias at T2	Mean of T1 and T2	Mean-Level Bias
	1	70	80	80	Add systematic error (+10 to T1, +10 to T2)	10	10	80	10
	2	70	70	70		0	0	70	0
	3	80	80	80		0	0	80	0
	4	80	80	80		0	0	80	0
	5	90	90	90		0	0	90	0
	6	90	90	90		0	0	90	0
	7	100	100	100		0	0	100	0
	8	100	100	100		0	0	100	0
	9	110	110	110		0	0	110	0
	10	110	110	110		0	0	110	0
Group-Level	Mean	90	91	91		1	1	91	1
	Variance	222.22	187.78	187.78				187.78	

Figure 3.20: Within-person systematic error.

In the example below, I added within-person systematic error. Specifically, to all participants, I added error that is consistent across time for a given participant: pulled from a distribution with a mean of 5 and a standard deviation of 3.

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
#Simulate Data
set.seed(52242)

time1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
time2 <- time1 + rnorm(n = sampleSize, mean = 0, sd = 4)
time3 <- time2 + rnorm(n = sampleSize, mean = 0, sd = 8)

#Add Missing Data
time1[c(5,10)] <- NA
time2[c(10,15)] <- NA
time3[c(10)] <- NA

#Combine data into data frame
mydata <- data.frame(rater1continuous, rater2continuous, rater3continuous,
                     rater1categorical, rater2categorical, rater3categorical,
```

```
                      time1, time2, time3,
                      item1, item2, item3, item4)

#Add error
set.seed(12345)
for(i in 1:nrow(mydata)){
  mydata[i, c("time1","time2","time3")] <- mydata[i, c("time1","time2","time3")] + rnorm(1, mean
}
```

3.8.4.1 Coefficient of stability

```
cor.test(x = mydata$time1, y = mydata$time2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydata$time1 and mydata$time2
## t = 29.005, df = 95, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9229555 0.9649296
## sample estimates:
##       cor
## 0.9479103
```

3.8.4.2 Coefficient of repeatability

```
qnorm(.975) * sd(blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$differences, r
```

```
## [1] 6.588758
```

3.8.4.3 Bias

```
blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$bias
```

```
## [1] 0.335823
```

3.8.4.4 Findings

Adding within-person systematic error led to:

- artificially stronger (higher) stability coefficient
- no effect on repeatability coefficient
- bias at the person-level, but this bias would go undetected by the estimate of group-level bias in the Bland-Altman plot

3.8.5 Add between-person systematic error at T2

Adding between-person systematic error at time 2 would involve adding systematic noise (e) across the rows, within the column. Between person systematic errors result from within-person errors that tend to be negative or positive across participants. For instance, this could reflect an influence with a shared effect across subjects. For example, social desirability leads to a positive group-level bias for rating their socially desirable attributes. Another example would be when a research assistant enters values wrong at time 2 (e.g., adding 10 to all participants' scores). Between-person systematic error increases bias because it results in a greater group mean difference from the group's mean true score.

Figure 3.21

						Person-Level			
	Participant	True Score	T1	T2	Influence	Bias at T1	Bias at T2	Mean of T1 and T2	Mean-Level Bias
	1	70	70	75	Add systematic error at T2 (5)	0	5	72.50	2.50
	2	70	70	75	Add systematic error at T2 (5)	0	5	72.50	2.50
	3	80	80	85	Add systematic error at T2 (5)	0	5	82.50	2.50
	4	80	80	85	Add systematic error at T2 (5)	0	5	82.50	2.50
	5	90	90	95	Add systematic error at T2 (5)	0	5	92.50	2.50
	6	90	90	95	Add systematic error at T2 (5)	0	5	92.50	2.50
	7	100	100	105	Add systematic error at T2 (5)	0	5	102.50	2.50
	8	100	100	105	Add systematic error at T2 (5)	0	5	102.50	2.50
	9	110	110	115	Add systematic error at T2 (5)	0	5	112.50	2.50
	10	110	110	115	Add systematic error at T2 (5)	0	5	112.50	2.50
Group-Level	Mean	90	90	95		0	5	92.50	2.50
	Variance	222.22	222.22	222.22				222.22	

Figure 3.21: Between-person systematic error.

In the example below, I added between-person systematic error. Specifically, I added constant variance of 10 to all participants' scores at T2

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
#Simulate Data
set.seed(52242)

time1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
```

```
time2 <- time1 + rnorm(n = sampleSize, mean = 0, sd = 4)
time3 <- time2 + rnorm(n = sampleSize, mean = 0, sd = 8)

#Add Missing Data
time1[c(5,10)] <- NA
time2[c(10,15)] <- NA
time3[c(10)] <- NA

#Combine data into data frame
mydata <- data.frame(rater1continuous, rater2continuous, rater3continuous,
                     rater1categorical, rater2categorical, rater3categorical,
                     time1, time2, time3,
                     item1, item2, item3, item4)

#Add error
mydata$time2 <- mydata$time2 + 10
```

3.8.5.1 Coefficient of stability

```
cor.test(x = mydata$time1, y = mydata$time2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydata$time1 and mydata$time2
## t = 28.184, df = 95, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9188254 0.9630063
## sample estimates:
##       cor
## 0.9450797
```

3.8.5.2 Coefficient of repeatability

```
qnorm(.975) * sd(blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$differences, r
```

```
## [1] 6.588758
```

3.8.5.3 Bias

```
blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$bias
```

```
## [1] -9.664177
```

3.8.5.4 Findings

Adding between-person systematic error led to:

- no effect on stability coefficient
- no effect on repeatability coefficient
- bias at the group-level (detected by group-level estimate of bias in Bland-Altman plot)

3.8.6 Add constant variance of 10 to all participants at all timepoints

In the example below, I added constant variance to all participants' scores at all timepoints. Specifically I added 10 to all participants' scores at all timepoints.

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
#Simulate Data
set.seed(52242)

time1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
time2 <- time1 + rnorm(n = sampleSize, mean = 0, sd = 4)
time3 <- time2 + rnorm(n = sampleSize, mean = 0, sd = 8)

#Add Missing Data
time1[c(5,10)] <- NA
time2[c(10,15)] <- NA
time3[c(10)] <- NA

#Combine data into data frame
mydata <- data.frame(rater1continuous, rater2continuous, rater3continuous,
                     rater1categorical, rater2categorical, rater3categorical,
                     time1, time2, time3,
                     item1, item2, item3, item4)
```

```
#Add error
mydata$time1 <- mydata$time1 + 10
mydata$time2 <- mydata$time2 + 10
mydata$time3 <- mydata$time3 + 10
```

3.8.6.1 Coefficient of stability

```
cor.test(x = mydata$time1, y = mydata$time2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydata$time1 and mydata$time2
## t = 28.184, df = 95, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9188254 0.9630063
## sample estimates:
##       cor
## 0.9450797
```

3.8.6.2 Coefficient of repeatability

```
qnorm(.975) * sd(blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$differences, r
```

```
## [1] 6.588758
```

3.8.6.3 Bias

```
blandr.statistics(method1 = mydata$time1, method2 = mydata$time2)$bias
```

```
## [1] 0.335823
```

3.8.6.4 Findings

Adding constant variance to all participants's scores at all time points led to:

- no effect on stability coefficient
- no effect on repeatability coefficient
- bias at the person-level, but this bias would go undetected by the estimate of group-level bias in the Bland-Altman plot

3.8.7 Summary

To summarize the effects of measurement error, random error (both within- and between-person) results in (a) a worse (lower) stability coefficient because a smaller proportion of the observed score variance is attributable to true score variance, and (b) a worse (higher) repeatability coefficient because the standard deviation of the difference between the two measurements is greater. Neither type of random error leads to group-level bias. However, between-person random error leads to bias at the person-level.

In terms of systematic error, all types of systematic error lead to increased bias, i.e., reduced accuracy for the person-level average and/or group-level average. Though multiple biases can co-exist and can work in opposite directions, which can obscure bias even though it exists. In terms of Within-person systematic error, it (a) artificially increases the stability coefficient, even though the measure is not actually more reliable, (b) has no effect on the repeatability coefficient because it is not changing the difference between the two measurements for a person, and (c) results in bias at the person-level because the person's mean systematically differs from their true score.

In terms of between-person systematic error at time 2 (an example of adding constant variance to all participants), it (a) had no effect on the stability coefficient because the rank order of individual differences stays the same, (b) had no effect on the repeatability coefficient because the standard deviation of the difference between the two measurements is not changed, (c) resulted in bias at the group-level that would be detected in a Bland-Altman plot because the group-level average deviates from group's average true score.

In sum, just because a measure has high estimates of test–retest reliability does not mean that the measure is actually reliable because systematic error does not reduce the stability and repeatability coefficients. Indeed, within-person systematic error actually (and artificially) *increases* the coefficient of stability. The coefficient of stability index of test–retest reliability is affected by the measure *and* variance across people, and it can be thrown off by outliers or by range restriction because the Pearson correlation strongly depends on these. And sample variation also has a strong influence, so it is important to cross-validate estimates of reliabilty in independent samples.

3.9 Effect of Measurement Error on Associations

As we describe in greater detail in the chapter on validity in Section 4.6, (random) measurement error weakens (or attenuates) the association between variables. The greater the random measurement error, the weaker the association. In the context of examining the association between test and retest (i.e., test–retest reliability), the estimate of the stability of the construct (i.e., true test–retest association) is attenuated to the extent that the measure is unreliable, as described in the following formula:

3.9.1 Attenuation of True Test–Retest Reliability due to Measurement Error

$$\begin{aligned} r_{x_1 x_2} &= r_{x_{1_t} x_{2_t}} \sqrt{r_{xx} r_{xx}} \\ &= r_{x_{1_t} x_{2_t}} \sqrt{{r_{xx}}^2} \\ &= r_{x_{1_t} x_{2_t}} r_{xx} \\ \text{test–retest correlation of measure} &= (\text{true correlation of construct across two timepoints}) \times \\ &\quad (\text{reliability of measure}) \end{aligned}$$

Find the observed test–retest association if the true test–retest association (i.e., the stability of the construct) is .7 and the reliability of the measure is .9:

```
trueAssociation <- 0.7
reliabilityOfMeasure <- 0.9

trueAssociation * reliabilityOfMeasure
```

```
## [1] 0.63
```

3.9.2 Disattenuation of Observed Test–Retest Reliability due to Measurement Error

Then, to disattenuate the observed test–retest reliability due to measurement error, we can rearrange the terms in the formula above to estimate what the true stability of the constuct is, according to the following formula:

$$\begin{aligned} r_{x_{1_t} x_{2_t}} &= \frac{r_{x_1 x_2}}{r_{xx}} \\ \text{true correlation of construct across two timepoints} &= \frac{\text{test–retest correlation of measure}}{\text{reliability of measure}} \end{aligned}$$

Find the true correlation of the construct across two timepoints given an observed correlation if the observed reliability of the measure is .9:

```
reliabilityOfMeasure <- 0.9

as.numeric(cor.test(x = mydata$time2, y = mydata$time3)$estimate)/reliabilityOfMeasure
```

```
## [1] 0.8974584
```

In the next section, I discuss method biases, including what they are, why they are important, and how to account for them.

3.10 Method Bias

3.10.1 What Method Biases are and Why They are Important

Method bias is the influence of measurement on a person's score that is not due to the person's level on the construct. Method bias is a form of systematic, nonrandom measurement error because it involves errors/biases that are shared/common to/correlated for two measures.

Method bias is problematic because it can bias estimates of a measure's reliability and validity, and its association with other measures/constructs. There is a biasing effect of assessing two or more constructs with the same method on estimates of the association between them—that is, some of the observed covariance may be due to the fact that they share the same method of measurement.

Although estimates vary widely from study-to-study, construct-to-construct, and measure-to-measure, estimates of method bias tend to be substantial. Meta-analyses suggest that around 20–40% of a given measure's variance reflects method variance [for a review, see Podsakoff et al., 2012]. According to Podsakoff et al. [2012], response styles inflate the magnitude of observed correlations by 54% if there is a positive correlation between the constructs and a positive correlation between the response styles. response styles inflate the magnitude of observed correlations by even more (67%) if there is a negative correlation between the constructs and a negative correlation between the response styles. Moreover, the method biases can also weaken the association between measures. If the direction of correlation between the constructs (e.g., positive) is incongruent with the direction of the correlation between the response styles (e.g., negative), the association between the measures is attenuated.

3.10.2 Types of Method Biases

There are many different types of methods biases, including the effects of:

- Same source: whether ratings or information comes from the same source can lead to systematic method variance.
- Response styles: response styles are systematic ways of responding to items/questions that are not due to the construct of interest. Different types of response styles can lead to systematic method variance across constructs. Examples of response styles include:
  - Acquiescence: the tendency to agree with items (both positively and negatively worded), regardless of content
  - Disacquiescence: the tendency to disagree with items (both positively and negatively worded), regardless of content
  - Extreme: the tendency to endorse items on the extremes (positive or negative) of the response scale, regardless of content
  - Midpoint: the tendency to endorse items on the middle scale category, regardless of content
  - Noncontingent: the tendency to respond to items carelessly, randomly, or nonpurposefully
- Proximity: items (of unrelated constructs) that are in closer proximity to one another in a measure are more likely to be correlated than items that are further apart. The associations of items assessing unrelated constructs is weakest when 6 or more items apart. There is an exception for reversed items. For reversed items (e.g., reverse scored), the further apart the reversed item is placed from another item, the higher the negative correlation.
- Item wording: whether items are positively or negatively worded can influence the association between items (like-worded items go with like)
- Item context: manipulating the context of a measure can have an effect on the perceived association. You can observe a different association between constructs when you put the measure in a negative context versus in a positive context.
- Item ambiguity: when respondents are less certain of how to accurately answer a question, it increases the likelihood that they will rely on response styles or succumb to the effect of context when responding, which biases scores
- Motivational factors: researchers need to motivate participants to put in the cognitive effort to answer questions correctly, to decrease the likelihood that they will succumb to responding based on response style only. Also, researchers should decrease pressures for "optimal responding" on respondents. That is, researchers should attempt to reduce social desirability effects.
- The ability of the respondent: respondents with low cognitive abilities, less education, more fatigue (after lots of questionnaires), less experience

with the topic being assessed, and more uncertainty about how to respond are more likely to succumb to biasing factors.

3.10.3 Potential Remedies

There are numerous potential remedies to help reduce the effects of method bias, including:

- Obtain measures of the predictor and criterion from multiple or different sources
- Create temporal, physical, and/or psychological separation between the predictor and criterion
- Eliminate common scale properties; vary the scale types and anchor labels
- Remove ambiguity in items—keep questions simple, specific, and concise
- Reduce social desirability in item wording
- Reverse score some items to balancing positive and negatively worded (i.e., reverse-scored) items
- Separate items on the questionnaire to eliminate proximity effects
- Maximize respondent motivation to give an accurate response
- Developing a good cover story and instructions
- Match the measure difficulty to the respondent's ability
- Use confirmatory factor analysis or structural equation modeling, as described in Section 5.11.1
- Use mixed models, as described below

3.10.3.1 Mixed Models

One approach to handle method biases is in mixed models, which are also called mixed-effects models, multilevel models, and hierarchical linear models (HLM). Mixed models are similar to multiple regression in that they allow fitting a model that examines multiple predictor variables in relation to one outcome variable. One of the assumptions of multiple regression is that the data observations are independent from each other, as evidenced by the residuals being uncorrelated with each other. Mixed models are designed to handle non-independent observations, that is, data that were collected from non-independent units. Thus, mixed models can be fit to data that do not meet the non-indendence assumption of multiple regression.

Examples of data from non-independent units include nested data, multiple membership data, and cross-classified data.

One example of non-independent data is if your data are *nested*. When data are collected from multiple lower-level units (e.g., people) in an upper-level unit (e.g., married couple, family, classroom, school, neighborhood), the data from the lower-level units are considered nested within the upper-level unit.

For example, multiple participants in your sample may come from the same classroom. Data from multiple people can be nested within the same family, classroom, school, etc. Longitudinal data can also be considered nested data, in which time points are nested within the participant (i.e., the same participant provides an observation across multiple time points). And if you have multiple informants of a child's behavior (e.g., parent-, teacher-, friend-, and self-report), the ratings could also be nested within the participant (i.e., there are ratings from multiple informants of the same child).

Another form of non-independent data is when data involve *multiple membership*. Data involve multiple membership when the lower-level units belong to more than one upper-level unit simultaneously. As an example of multiple membership, children may have more than one teacher, and therefore, in a modeling sense, children "belong" to more than one teacher cluster.

Another form of non-independent data is when data are *cross-classified* (also called crossed). Data are cross-classified when the lower-level units are classified by two or more upper-level units, but the upper-level units are not hierarchical or nested within one another. For instance, children may be nested within the crossing of schools and neighborhoods. That is, children are nested within schools, and children are also nested within neighborhoods; however, children attending the same school may not necessarily be from the same neighborhood. The data would still have a two-level hierarchy, but the data would be nested in specific school-neighborhood combinations.

The general form of multiple regression is:

$$y = b_0 + b_1 x + e$$

where b_0 is the intercept, e is the error term, and b_1 is the slope of the predictor, x, that states the relation between the predictor (x) and the outcome (y). However, the slope can exist at multiple levels when data are nested within groups or time points are nested within people. For example, you can observe different associations of a predictor with an outcome at the between-individual level compared to the within-individual level. That is, the association between the predictor and outcome can differ when comparing across individuals versus when looking at the association between the predictor and outcome based on fluctuations within the individual. When the associations differ at multiple levels, this is known as *Simpson's paradox*. For an example of Simpson's paradox, see Figure 3.22 [Kievit et al., 2013]. Thus, it is important to consider that the association between variables may differ at different levels (person-level, group-level, population-level, etc.).

To account for method biases in a mixed model, you could use scores from multiple measures and methods as the outcome variable. For instance, your mixed model could examine the association between income and optimism, where optimism is assessed by self-report, parent-report, and a performance-based measure. In the model, you could include, as a *fixed effect*, a factor that indicates the

Figure 3.22: Example of Simpson's paradox, where the association between the predictor and outcome differs at different levels. In this case, the association between dosage and recovery probability is positive at the population-level (upper-level unit), but the association is negative among men and women separately (lower-level unit). Figure reprinted from @Kievit2013, Figure 1, p. 2. Kievit, R., Frankenhuis, W., Waldorp, L., & Borsboom, D. (2013). Simpson's paradox in psychological science: a practical guide. *Frontiers in Psychology*, *4*(513). https://doi.org/10.3389/fpsyg.2013.00513

method of the measure, in addition to its interaction with income, to account for systematic ways in which the measurement method influences scores on the measures across the participants as a whole. You could also include a *random effect* of measurement method if you wanted to allow the effect of measurement method on the scores to differ from person to person.

[*INCLUDE EXAMPLE*]

3.11 Generalizability (G-)Theory

Up to this point, we have discussed reliability from the perspective of CTT. However, as we discussed, CTT makes several assumptions that are unrealistic (e.g., that all error is random). There are other measurement theories that conceptualize reliability differently than the way that CTT conceptualizes reliability. One such measurement theory is generalizability theory [Brennan, 1992], also known as G-theory and domain sampling theory. G-theory is also discussed in the chapters on validity (Chapter 4, Section 4.7) and structural equation modeling (Chapter 5, Section 5.13).

In CTT, there is an assumption that there exists a true score that is an accurate measure of the trait under a *specific* set of conditions. G-theory is an alternative measurement theory that does not treat all measurement differences across time, rater, or situation as "error" but rather as a phenomenon of interest [Wiggins, 1973]. G-theory is a measurement theory that is used to examine the extent to which scores are consistent across a specific set of conditions. In G-theory, the true score is conceived of as a person's *universe score*—the mean of all observations for a person over all conditions in the universe—this allows us to estimate and recognize the magnitude of multiple influences on test performance. These multiple influences on test performance are called *facets*. G-theory asserts that the reliability of a test does not reside within the test itself; a test's reliability depends on the circumstances under which it is developed, administered, and interpreted. A person's test scores vary from testing to testing because of (many) variables in the testing situation.

Instead of conceiving of all variability in a person's scores as error, G-theory argues that we should describe the details of the particular test situation (universe) that lead to a specific test score. The universe is described in terms of its facets:

- settings
- observers (e.g., amount of training they had)
- instruments (e.g., number of items in test)
- occasions (time points)
- attributes (i.e., what we are assessing; the purpose of test administration)

Table 3.1: Percent of variance from different sources in generalizability theory model with three facets: person, item, and occasion (and their interactions). Adapted from @Webb2005

Source	Variance Accounted For (%)
Person (p)	30
Item (i)	5
Occasion (o)	3
p x i	25
p x o	5
i x o	2
p x i x o	10
residual	20

By assessing a person in multiple facets of the universe, this allows us to estimate and recognize the magnitude of multiple sources of measurement error. Such measurement error includes:

- day-to-day variation in performance (stability of the construct, test–retest reliability)
- variance in the item sampling (coefficient of internal consistency)
- variance due to both day-to-day and item sampling (coefficient of equivalence from parallel-forms reliability, or convergent validity, as will be discussed in Section 4.7)

In G-theory, all sources of measurement error (facets) are considered simultaneously—something CTT cannot achieve [Shavelson et al., 1989]. This occurs through specifying many different variance facets in the estimation of the true score, rather than just one source of error variance as in CTT. This specification allows us to take into consideration variance due to occasion effects, item effects, and occasion × item effects (i.e., main effects of the facets in addition to their interaction).

A score's usefulness largely depends on the extent to which it allows us to generalize accurately to behavior in a wider set of situations (a universe of generalization). According to G-theory, given the exact same conditions of all the facets in the universe, the exact same test score should be obtained. This is the universe score, which is analogous to the true score in CTT.

In G-theory, the goal is to conduct a *generalizability study*. A generalizability study examines how generalizable scores from a particular test are if the test is administered in different situations. Traditionally, a generalizability study would test this using a factorial analysis of variance [ANOVA; Brennan [1992]], as exemplified in Section 3.11.2.1, (as opposed to simple ANOVA in CTT). The

researcher must specify and define the universe (set of conditions) to which they would like to generalize their observations and in which they would like to study the reliability of the measure. For instance, it might involve randomly sampling from within that universe (in terms of people, items, observers, conditions, etc.). In G-theory, reliability is estimated with the *generalizability coefficient*. However, ANOVA is limiting—it works best with balanced designs, such as with the same sample size in each condition condition/facet; but in most real-world applications, things are not actually balanced. So, it is better to fit G-theory models in a mixed model framework, as exemplified in Section 3.11.2.2.

G-theory is strong theoretically, but it has not been widely implemented. G-theory can be challenging because the researcher must specify, define, and assess the universe to which they would like to generalize their observations and to understand the reliability of the measure. But, G-theory provides an important reminder that reliability is not one thing. You cannot just say that a test "is reliable"; it is important to specify the facets across which the reliability has been established (e.g., times, raters, items, groups, instruments).

Below are examples implementing G-theory. The `pio_cross_dat` data file for these examples come from the `gtheory` package [Moore, 2016]. The examples are adapted from Huebner and Lucht [2019].

3.11.1 Universe score for each person

```
pio_cross_dat %>%
  group_by(Person) %>%
  summarise(universeScore = mean(Score, na.rm = TRUE), .groups = "drop")
```

```
## # A tibble: 6 x 2
##   Person universeScore
##   <fct>          <dbl>
## 1 1               7
## 2 2               5.5
## 3 3               5.5
## 4 4               6.12
## 5 5               8.25
## 6 6               3.25
```

3.11.2 Generalizability (G) Study

3.11.2.1 ANOVA framework

```
summary(aov(Score ~ Person*Item*Occasion,
            data = pio_cross_dat))
```

```
##                      Df Sum Sq Mean Sq
## Person                5 112.94   22.59
## Item                  3 117.90   39.30
## Occasion              1  15.19   15.19
## Person:Item          15  35.48    2.37
## Person:Occasion       5   5.94    1.19
## Item:Occasion         3   2.90    0.97
## Person:Item:Occasion 15  12.48    0.83
```

3.11.2.2 Mixed model framework

The mixed model framework for estimating generalizability is described by Jiang [2018].

```
summary(lmer(Score ~ 1 + (1|Person) + (1|Item) + (1|Occasion) + (1|Person:Occasion) +
             data = pio_cross_dat))
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Score ~ 1 + (1 | Person) + (1 | Item) + (1 | Occasion) + (1 |
##     Person:Occasion) + (1 | Person:Item) + (1 | Occasion:Item)
##    Data: pio_cross_dat
##
## REML criterion at convergence: 177.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.29785 -0.65345 -0.09366  0.67122  1.41025
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  Person:Item     (Intercept) 0.76666  0.8756
##  Person:Occasion (Intercept) 0.08889  0.2981
##  Occasion:Item   (Intercept) 0.02227  0.1492
##  Person          (Intercept) 2.48374  1.5760
##  Item            (Intercept) 3.06732  1.7514
##  Occasion        (Intercept) 0.57787  0.7602
##  Residual                    0.83187  0.9121
## Number of obs: 48, groups:
## Person:Item, 24; Person:Occasion, 12; Occasion:Item, 8; Person, 6; Item, 4; Occasion
##
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    5.937      1.237   4.801
```

```
PxIxO <- gstudy(data = pio_cross_dat,
                Score ~ (1|Person) + (1|Item) + (1|Occasion) + (1|Person:Item) + (1|Person:Occasi

PxIxO
```

```
## $components
##            source        var percent n
## 1     Person:Item 0.76665688     9.8 1
## 2 Person:Occasion 0.08889285     1.1 1
## 3   Occasion:Item 0.02227409     0.3 1
## 4          Person 2.48374310    31.7 1
## 5            Item 3.06732106    39.1 1
## 6        Occasion 0.57787462     7.4 1
## 7        Residual 0.83186863    10.6 1
##
## attr(,"class")
## [1] "gstudy" "list"
```

3.11.3 Decision (D) Study

The decision (D) study and generalizability (G) study, from which the generalizability and dependendability coefficients can be estimated, were analyzed using the gtheory package [Moore, 2016].

```
decisionStudy <- dstudy(PxIxO, colname.objects = "Person", data = pio_cross_dat, colname.scores =
```

3.11.4 Generalizability Coefficient

The generalizability coefficient is analogous to the reliability coefficient in CTT. It divides the estimated person variance component (the universe score variance) by the estimated observed-score variance. It uses relative error variance, so it is relevant when decisions are made based on relative (norm-referenced) scores (e.g., percentiles).

```
decisionStudy$generalizability
```

```
## [1] 0.8795631
```

3.11.5 Dependability Coefficient

The dependability coefficient is analogous to the reliability coefficient in CTT. It uses absolute error variance, so it is relevant when decisions are made based on absolute (e.g., raw) scores.

```
decisionStudy$dependability
```

```
## [1] 0.639746
```

3.12 Item Response Theory

In addition to CTT and G-theory, item response theory (IRT) is another measurement theory. IRT also estimates reliability in a different way compared to CTT. We discuss IRT in Chapter (6).

3.13 The Problem of Low Reliability

Using measures that have low reliability can lead to several problems, including:

1. over- or under-estimation of the effects of interest. When sample sizes are large, measurement error tends to weaken effect sizes. However, when samples sizes are small-to-modest, measurement error can actually lead to over-estimation of effect sizes as well as inaccurate estimates of the sign of the effect [positive or negative; Loken and Gelman [2017]].
2. reduced statistical power to detect the effects of interest
3. failure to replicate findings
4. invalid measurement for the proposed uses. As we described in the chapter on validity in Section 4.5, reliability constrains validity.

These concerns are especially important for individual differences research (e.g., correlational designs) or for when making decisions about individual people based on their scores on measure. However, these concerns are also relevant for experimental designs.

3.14 Ways to Increase Reliability

Here are 13 potential ways to increase reliability of measurement:

1. Improve the clarity and homogeneity of items.

2. Use a structured response set and scoring system rather than unstructured ones.
3. Make sure participants know how to complete the task or items. For example, make sure they have appropriate instructions that are easily understood.
4. Adhere closely to the standardized procedures for administering and scoring a measure.
5. Make scoring rules as explicit as possible.
6. Train raters to some criterion of inter-rater reliability.
7. Use a more precise response scale. The smaller the number of response options, the greater the attenuation of a correlation [Rigdon, 2010].
8. Have a sample with a full range of variation in the items and measure.
9. If possible, do not use a difference score. This is discussed in greater detail in Sections 3.5.8 and [Section on performance-based cognitive assessments].
10. Remove items that reduce reliability. This may include items that have low variability because they are too easy or too difficult, in addition to items that have a low item–total correlation.
11. Collect more information across people and time.
12. Aggregate: add items, measures, observers, raters, and time points.

    a. Having more items that assess the same construct in the measure tends to increase reliability unless the additional items cause fatigue because the instrument is too long. Aggregation of like things reduces measurement error, because averaging together more items helps cancel out random error.
    b. Having multiple measures that assess the same construct can be beneficial, especially measures from different methods that do not share the same systematic errors (e.g., observation in addition to parent report).
    c. Items that aggregate a large amount of information tend to have strong predictive validity. For example, when hot dog sales are high, there are more wins at Yankees games. Hot dog sales have no theoretical influence on the Yankees, but it predicts well. In general, for predictive accuracy, it is often better to assess more things less well than to assess one thing extraordinarily well. Aggregation is a double-edged sword: it is your friend as a researcher, because of increased predictive utility, but it is your enemy as an interpreter because it is hard to interpret the theoretical meaning of predictive associations based on an aggregated variable because the measure is more general and nonspecific. If aggregation increases error, test–retest reliability will decrease. If error stays the same when aggregating, test–retest reliability will increase.

13. Use a latent variable approach to aggregate the multiple items, measure, items, measures, observers, raters, or time points. Latent variable approaches include structural equation modeling and item response theory.

Estimating a latent variable allows identifying the variance that is common to all of the measures (i.e., a better approximate of true score variance), and discards variance that is not shared by all measures (i.e., error variance). Thus, latent variables have the potential to get purer estimates of the constructs by reducing measurement error.

3.15 Conclusion

Reliability is how much repeatability, consistency, and precision a measure's scores have. Reliability is not one thing. There are multiple aspects of reliability, and the extent to which a given aspect of reliability is important depends on the construct of interest. In general, we want our measures to show strong inter-rater, intra-rater, immediate parallel-forms, and internal consistency reliability. However, the extent to which our measures should show test–retest and delayed parallel-forms reliability depends on the stability of the construct and the time lag. Moreover, although we want our measures to show strong internal consistency reliability, the internal consistency reliability for a measure can be *too* high. In addition, reliability is not a characteristic that resides in a test. The reliability of a measure's scores reflects an interaction of the properties of the test with the population for whom it is designed, and the sample, situation, and context in which it is administered. Thus, when reporting reliability in papers, it is important to adequately describe the aspects of reliability that have been considered, and the population, sample, and context in which the measure is assessed. However, it is not enough for measures to be consistent or reliable; our interpretation of the measures' scores should also be accurate, meaningful, and useful for the intended uses. In other words, our interpretation of the measures' scores should be valid for the intended uses. In the next chapter (Chapter 4), we discuss validity of measurement.

3.16 Exercises

3.16.1 Questions

Note: several of the following questions use data from the Children of the National Longitudinal Survey of Youth Survey (CNLSY). The CNLSY is a publicly available longitudinal dataset provided by the Bureau of Labor Statistics (https://www.bls.gov/nls/nlsy79-children.htm#topical-guide). The CNLSY data file for these exercises is located on the course's page of the Open Science Framework (https://osf.io/3pwza). Children's behavior problems were rated in 1988 (time 1: T1) and then again in 1990 (time 2: T2) on the Behavior Problems Index (BPI).

Table 3.2: Exercise 12: Table of diagnostic decisions for assessing inter-rater reliability.

Client	Your Diagnosis of the Client	Your Colleague's Diagnosis of the Client
1	Yes	Yes
2	No	No
3	Yes	No
4	No	Yes
5	No	No

1. What is the test–retest reliability of the Antisocial Behavior subscale of the BPI? Provide the coefficient of stability and coefficient of repeatability, and interpret what they indicate about the Antisocial Behavior subscale of the BPI.
2. What was the estimate of bias of the Antisocial Behavior subscale of the BPI comparing the measurements at T1 and T2, and what does this indicate about the Antisocial Behavior subscale of the BPI and the construct?
3. If a person had a score of 6 on the Antisocial Behavior subscale of the BPI at T1, within what range would we expect their score to be at T2?
4. What interval is often considered the "standard" for evaluating test–retest reliability of measures in psychology? What is a potential downside of having shorter intervals? What is a potential downside of having longer intervals? How would you expect the test–retest reliability and bias to differ had the BPI been re-assessed at that ("standard") interval?
5. For each of the following, what would the coefficient of stability, coefficient of repeatability, and bias be if the following measurement error were added. Also summarize the effects of measurement error in each case:

a. 3 points to all scores at only T2
b. 3 points to all scores at both T1 and T2
c. 3 points to the scores for the first 500 participants at only T1
d. 3 points to the scores for the first 500 participants at both T1 and T2

6. You and a colleague both conduct diagnostic interviews on the same five clients. The data are in Table 3.2 below. What is your inter-rater reliability with your colleague?

7. You and a colleague both rate the same five clients in the extent to which they are likely to be suicidal, where 1 = very low likelihood and 5 = very high likelihood. The data are in Table 3.3 below. What is your inter-rater reliability with your colleague?

Table 3.3: Exercise 12: Table of ratings for assessing inter-rater reliability.

Client	Your Rating of the Client	Your Colleague's Rating of the Client
1	5	4
2	3	2
3	4	3
4	2	2
5	1	2

8. What was the internal consistency of the items on the Antisocial Behavior subscale of the BPI at T1? Provide Cronbach's alpha (α), omega total (ω_t), omega hierarchical (ω_h), and the average of all possible split-half reliabilities. What do the estimates of internal consistency indicate about the BPI subscale?
9. What is the standard error of measurement of the Antisocial Behavior subscale of the BPI? If a child's observed score is 6, within what range would we expect their true score to be? How confident are we about a child's true score on the construct (antisocial behavior) if we know their observed score on the measure (Antisocial Behavior subscale of the BPI)?
10. For many ways of estimating test–retest reliability, we assume that the true association between the construct at T1 and the construct at T2 is 1.0 (that is, the association of the true scores is 1.0). According to the attenuation formula (that attenuates the association due to measurement error), if the true association between the construct at T1 and the construct at T2 was 1.0 and the reliability of the measure was .8, the observed test–retest correlation would be $r = .80$. According to the attenuation formula, what would the observed test–retest correlation be if the true association was .9 and the reliability of the measure was .8?
11. According to the disattenuation formula (to correct for attenuation of the association due to measurement error), what would be the true test-rest association of the construct of antisocial behavior if the reliability of the Antisocial Behavior subscale was $r = .6299494$?
12. You want to see how generalizable the Antisocial Behavior subscale of the BPI is. You conduct a generalizability study ("G study") to see how generalizable the scores are across participants ($N = 3$), two measurement occasions, and three raters. What is the universe score (estimate of true score) for each participant? What percent of variance is attributable to: (a) individual differences among participants, (b) different raters, (c) the measurement occasions, and (d) the interactive effect of raters and measurement occasions? Using a decision study ("D study"), what are the generalizability and dependability coefficients? Is the measure reliable across the universe (range of factors) we examined it in? Interpret the results from the G study and D study. The data from your study are in

Table 3.4: Exercise 12: Table of example data for evaluating generalizability of scores across participants, occasions, and raters.

Participant	Occasion	Rater	Score
1	1	1	7
2	1	1	2
3	1	1	0
1	2	1	10
2	2	1	5
3	2	1	1
1	1	2	8
2	1	2	3
3	1	2	1
1	2	2	11
2	2	2	6
3	2	2	1
1	1	3	6
2	1	3	2
3	1	3	4
1	2	3	12
2	2	3	6
3	2	3	5

Table 3.4 below:

3.16.2 Answers

1. The coefficient of stability ($r = .50$) indicates that the Antisocial subscale of the BPI shows moderate rank-order stability over a two-year interval at these ages in this sample. The coefficient of repeatability (4.45) indicates that 95% of repeated assessments on the BPI over a two-year interval would be expected to fall within 4.45 points of the score at the first measurement occasion (after accounting for any bias). Given that the total score ranges from 0–14, a repeatability coefficient of 4.45 indicates that individual children showed considerable differences from T1 to T2.
2. The estimate of bias was 0.10, indicating that children's scores were slightly (0.10 points) higher, on average, at T1 compared to T2. This may indicate developmental change such that children tend to decrease in antisocial behavior as they get older.
3. Their score would at T2 expected to be within the range of [1.65, 10.55].

4. A two-week interval is typically considered standard for evaluating test–retest reliability. Shorter intervals have the potential problem that there may be carryover effects from the original testing due to the participant's memory of their answers from the first testing (thus artificially increasing the estimate of test–retest reliability). Longer intervals have the potential problem that real developmental change could occur in between the measurement occasions, thus artificially decreasing the estimate of test–retest reliability. The BPI was assessed at a two-year interval. Had it been assessed at a two-week interval, we would expect test-rest reliability to be higher (higher coefficient of stability and lower coefficient of repeatability) and bias to be lower.
5. The estimates are as follows:
    a. Stability coefficient: $r = .50$; repeatability coefficient = 4.45; bias = -2.90
        i. increased bias (greater absolute value); no detectable effect on stability or repeatability coefficient
    b. Stability coefficient: $r = .50$; repeatability coefficient = 4.45; bias = 0.10
        i. no detectable effect on stability coefficient, repeatability coefficient, or bias
    c. Stability coefficient: $r = .48$; repeatability coefficient = 4.64; bias = -0.01
        i. lower/worse stability coefficient; higher/worse repeatability coefficient; contributed bias (but in the opposite direction of the bias before the score adjustment)
    d. Stability coefficient: $r = .54$; repeatability coefficient = 4.45; bias = 0.10
        i. artificially higher/stronger stability coefficient; no detectable effect on repeatability coefficient or bias
6. Your inter-rater reliability was very poor (Cohen's $\kappa = .17$).
7. Your inter-rater reliability was good. If you used only one judge's ratings in analysis, the scores would have an intra-class correlation of $\text{ICC}(2, 1) = .76$. If, however, you used the average of the judge's ratings in analysis, the scores would have an intra-class correlation of $\text{ICC}(2, k) = .86$.
8. The estimates of internal consistency (Cronbach's alpha $[\alpha] = .70$, omega total $[\omega_t] = .70$, omega hierarchical $[\omega_h] = .71$, average split-half reliability $= .73$) indicate that the ratings show moderate consistency across items on the Antisocial subscale (and that the items are somewhat similar to one another), but that the items on the Antisocial subscale are also somewhat distinct from one another.
9. The standard error of measurement is calculated based on the reliability of the measure. Reliability can be evaluated in different ways, and there is no single index of reliability. If we were to take infinite observations of a

child's score and average them, that average would approximate the child's true score. Then, the test–retest reliability of the measure would help tell us about how (im)precise our estimate was at any given occasion compared to the true score. For the Antisocial Behavior subscale of the BPI, the test–retest reliability was .50 as indexed by the coefficient of stability. If we used this as our estimate of the measure's reliability (though the stability coefficient would likely be higher if we used a re-test interval of two weeks instead of two years), the standard error of measurement would be 1.64. Using this as our estimate of the standard error of measurement, we would expect the child's true score to fall within the range of $[2.79, 9.21]$. That's a pretty wide range of uncertainty for a scale that ranges from 0–14. So, we are not very confident of a child's true score on the construct from their observed score on this measure.

10. The observed test–retest correlation (coefficient of stability) would be $r = .72$.
11. The true test–retest association would be $r = .80$.
12. The universe score is 9, 4, and 2 for participants 1, 2, and 3, respectively. The percent of variance attributable to each of those factors is: (a) participant: 64.6%, (b) rater: 0.2%, (c) the measurement occasion: 16.2%, and (d) the interactive effect of rater and measurement occasion: 1.5%. The generalizability coefficient is .90, and the dependability coefficient is .81. The measure is fairly reliable across the universe examined, both in terms of relative differences (generalizability coefficient) and absolute differences (dependability coefficient). However, when using the measure in future work, it would be important to assess many participants (due to the strong presence of individual differences) across multiple occasions (due to the strong effect of occasion) and multiple raters (due to the moderated effect of rater as a function of participant). It will be important to account for the considerable sources of variance in scores on the measure in future work including: participant, occasion, participant $\times$ rater, and participant $\times$ occasion. By accounting for these sources of variance, we can get a purer estimate of each participant's true score and the population's true score.

Chapter 4

Validity

"What we know depends on how we know it."

4.1 Overview of Validity

According to the "Standards for Educational and Psychological Testing" [American Educational Research Association et al., 2014, p. 11], measurement validity is "the degree to which evidence and theory support the interpretations of test scores for proposed uses of tests." I summarized reliability with three words: repeatability, consistency, and precision. My summary of validity in three words is: accuracy, utility, and meaningfulness (of the interpretations of a measure's scores for the the proposed uses). Validity is tied to the *interpretation* of a measure's scores for the *proposed uses*, not to the measure itself. The same set of scores can have different degrees of validity for different purposes. For instance, a measure's scores may have stronger validity for making a diagnostic decision than for making a prediction about future behavior. Thus, as "the Standards" indicate, it is incorrect to use the unqualified phrase "the validity of the measure" or "the measure is (in)valid" because these phrases do not specify which scores were used from the test, what the use is (e.g., predicting whether a person will succeed in a given job), and what interpretation was made of the test scores for this purpose.[1]

Below, we prepare the data to provide some validity-related examples throughout the rest of the chapter.

[1]For shorthand, I sometimes refer to the validity of the measure, but in such instances, I am actually referring to the validity of the interpretation of a measure's scores for a given use.

4.2 Getting Started

4.2.1 Load Libraries

```
library("lavaan")
library("semPlot")
library("rockchalk")
library("semTools")
library("kableExtra")
library("MASS")
library("psych")
library("tidyverse")
library("here")
library("tinytex")
library("rmarkdown")
library("DT")
```

4.2.2 Prepare Data

4.2.2.1 Load Data

The `MTMM` data file comes from W. Joel Schneider: http://my.ilstu.edu/~wjschne/442/MTMM.csv

```
MTMM <- read_csv(here("Data", "MTMM.csv")) #read_csv("../Data/MTMM.csv")
```

4.2.2.2 Simulate Data

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
sampleSize <- 1000

set.seed(52242)

means <- c(50, 100)
standardDeviations <- c(10, 15)

correlationMatrix <- matrix(.7, nrow = 2, ncol = 2)
diag(correlationMatrix) <- 1
rownames(correlationMatrix) <- colnames(correlationMatrix) <- c("predictor","criterion"
```

```
covarianceMatrix <- psych::cor2cov(correlationMatrix, sigma = standardDeviations)

mydataValidity <- as.data.frame(mvrnorm(n = sampleSize, mu = means, Sigma = covarianceMatrix, emp

errorToAddToPredictor <- 3.20
errorToAddToCriterion <- 6.15

mydataValidity$predictorWithMeasurementErrorT1 <- mydataValidity$predictor + rnorm(n = sampleSize
mydataValidity$predictorWithMeasurementErrorT2 <- mydataValidity$predictor + rnorm(n = sampleSize

mydataValidity$criterionWithMeasurementErrorT1 <- mydataValidity$criterion + rnorm(n = sampleSize
mydataValidity$criterionWithMeasurementErrorT2 <- mydataValidity$criterion + rnorm(n = sampleSize

mydataValidity$oldpredictor <- mydataValidity$criterion + rnorm(n = sampleSize, mean = 0, sd = 7.
```

4.2.2.3 Add Missing Data

Adding missing data to dataframes helps make examples more realistic to real-life data, and helps you get in the habit of programming to account for missing data.

```
missingValuesPredictor <- sample(1:sampleSize, size = 50, replace = FALSE)
missingValuesCriterion <- sample(1:sampleSize, size = 50, replace = FALSE)

mydataValidity$predictor[missingValuesPredictor] <- NA
mydataValidity$predictorWithMeasurementErrorT1[missingValuesPredictor] <- NA
mydataValidity$predictorWithMeasurementErrorT2[missingValuesPredictor] <- NA

mydataValidity$criterion[missingValuesCriterion] <- NA
mydataValidity$criterionWithMeasurementErrorT1[missingValuesCriterion] <- NA
mydataValidity$criterionWithMeasurementErrorT2[missingValuesCriterion] <- NA

mydataValidity$oldpredictor[missingValuesPredictor] <- NA
```

4.3 Types of Validity

Like reliability, validity is not one thing. There are many types of validity. In this book, I discuss the following types of validity:

- face validity
- content validity
- criterion-related validity

- concurrent validity
- predictive validity
- construct validity
- convergent validity
- discriminant (divergent) validity
- incremental validity
- treatment utility of assessment
- discriminative validity
- elaborative validity
- consequential validity
- representational validity
- factorial (structural) validity
- ecological validity
- process-focused validity
- internal validity
- external validity
- (statistical) conclusion validity

I arrange these types of validity into two broader categories: measurement validity and research design validity.

4.3.1 Measurement Validity

Aspects of *measurement validity* involve the validity of a particular measure, or more specifically, the validity of interpretations of scores from that measure for the proposed uses.

4.3.1.1 Face Validity

The interpretation of a measure's scores has *face validity* (for a given construct and use) if a typical person—a non-expert—who looks at the content of each item will believe that the item belongs in the scale for this construct and use. The measure, and each item, looks "on its face" like it assesses the target construct. There are several advantages of a measure having face validity. First, outside groups will be less likely to be critical of the measure because it is intuitive. Second, use of a face valid measure is rarely objected to on ethical and bias charges for selling the measure to the public or clinicians. Third, face validity can be helpful for dissemination because more people may be receptive to it.

However, face validity also has important disadvantages. First, judgments of face validity are not based on theory. Second, face validity is based on subjective judgment, which can be inaccurate. Third, these subjective judgments are made by laypeople whose judgments may be inaccurate because of biases and lack of awareness of scientific knowledge. Fourth, a face valid measure may be too

simple because anybody can understand the questions and what questions are intended to assess, so (presumably) respondents can easily fake responses to achieve their goals. Faking of responses may be a more of a concern in situations when there is an incentive for the respondent to achieve a particular outcome (e.g., be deemed competent to stand trial, be judged competent to obtain a job or custody of child, be judged to have a disorder to receive accommodations or disability benefits).

It is disputed whether having face validity is good or bad. Whether face validity is important to a given measure depends on the construct that is intended to be assessed, the context in which the assessment will occur, who will be paying for and/or administering the assessment, whether the respondents have incentives to achieve particular scores, and the goals of the assessment (i.e., how the assessment will be used). There is also controversy about whether face validity is a true form of validity; many researchers have argued that it not a true psychometric form of validity [**insert citation**].

4.3.1.2 Content Validity

Content validity involves a judgment about whether or not the content (items) of the measure theoretically matches the construct that is intended to be assessed, that is, whether the operationalization accurately reflects the construct. Content validity is developed based on items generated and selected by experts of the construct and based on the subjective determination that the measure adequately assesses and covers the construct of interest. Content validity differs from face validity in that, for face validity, a *lay person* determines whether or not the measure seems to assess the construct of interest. By contrast, for content validity, an *expert* determines whether or not the measure adheres to the construct of interest.

For a measure to have content validity, its items should span the breadth of the construct. For instance, the construct of depression has many facets, such as sleep disturbances, weight/appetite changes, low mood, suicidality, etc., as depicted in Figure 4.1.

For a measure to have content validity, there should be no *gaps*—facets of the construct that are not assessed by the measure—and there should be no *intrusions*—facets of different constructs that are assessed by the measure. Consider the construct of depression. If theory states the construct includes various facets such as sadness, loss of interest in activities, sleep disturbances, lack of energy, weight/appetite change, and suicidal thoughts, then a content valid measure should assess all of those facets. If the measure does not assess sleep disturbances (a gap), the measure would lack content validity. If the measure assessed facets of other constructs, such as impulsivity (an intrusion), the measure would lack content validity.

With content validity, it is important to consider the population of interest. The

Figure 4.1: Content facets of the construct of depression.

same construct may look different in different populations and may require different content to assess it. For instance, it is important to consider the cultural relativity of constructs. The content of a construct may depend on the culture, such as in the case of culture-bound syndromes. Culture-bound syndromes are syndromes that are limited to particular cultures. An example of a culture-bound syndrome among Korean women is *hwa-byung*, which is the feeling of an uncomfortable abdominal mass in response to emotional distress. Another important dimension to consider is development. Constructs can manifest differently at different points in development, known as heterotypic continuity, which is discussed in Section [**insert section**]. When considering the different dimensions of your population, it can be helpful to remember the acronym *ADDRESSING*.

4.3.1.3 Criterion-Related Validity

Criterion-related validity examines whether a measure behaves the way it should given your theory of the construct. This is quantified by the correlation between a measure's scores and some (hopefully universally accepted) criterion we select. For instance, a criterion could be a diagnosis, a child's achievement in school, an employee's performance in a job, etc.

Below, I provide an example of criterion-related validity by examining the Pearson correlation between a predictor and a criterion.

```
cor.test(x = mydataValidity$predictor, y = mydataValidity$criterion)
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  mydataValidity$predictor and mydataValidity$criterion
## t = 29.136, df = 898, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6619034 0.7292278
## sample estimates:
##      cor
## 0.697099
```

There are two types of criterion-related validity: concurrent validity and predictive validity.

4.3.1.3.1 Concurrent Validity *Concurrent validity* considers the *concurrent* association between the chosen measure and the criterion. That is, both the measure and the criterion are assessed at the same point in time. An example of concurrent validity would be examining self-report of court involvement in relation to current court records.

4.3.1.3.2 Predictive Validity *Predictive validity* considers the association between the chosen measure and the criterion at a *later* time point. An example of predictive validity would be examining the association between children's scores on an academic achievement test in first grade predicting their eventual academic outcomes five years later.

4.3.1.3.3 Empiricism and Theory Criterion-related validity arose out of a movement known as *radical operationalism*. Radical operationalism was a pushback against psychoanalysis. Psychoanalysis focused on grand theoretical accounts for how constructs relate. The goal on radical operationalism was to clarify concepts from a behavioristic perspective to allow predicting and changing behavior more successfully. "An"operation" in radical operationalism refers to a fully described measurement.

Proponents of radical operationalism argued that all constructs in psychology that could not be operationally defined should be excluded from the field as "non-scientific." They asserted that operations should be well-defined enough to be able to replicate the findings. So, constructs had to be defined precisely, according to this perspective, but how precisely? You could go on forever trying to more precisely describe a behavior in terms of its form, frequency, duration, intensity, situation, antecedents, consequences, biological substrates, etc. So, radical operationalists asserted that we should use theory of the construct to determine what is essential and what is not.

Radical operationalism was also related to *radical behavioralism*, which was espoused by B.F. Skinner. Skinner famously used a box (the "Skinner Box") to

more directly control, describe, and assess behaviors. Skinner noted the major role that the environment played in influencing behavior. Skinner proposed a theory of implicit learning about a behavior or stimulus based on its consequences, a theory known as operant conditioning. According to operant conditioning, something that increases the frequency of a given behavior is called a reinforcer (e.g., praise), and something that decreases the frequency of a behavior is called a punisher (e.g., loss of a privilege). Through this work, Skinner came to view every thing an organism does (e.g., action, thought, feeling) as a behavior.

Related to these historical perspectives was a perspective known as *dustbowl empiricism*. Dustbowl empiricism focused on the empirical connections between things—how things were associated using data. It was a completely atheoretical perspective in which interpretation was entirely data-driven. An example of dustbowl empiricism is the approach that was used to develop the first version of the Minnesota Multiphasic Personality Inventory (MMPI). The MMPI was developed using an approach known as empirical-criterion keying, where items were selected for the scale for no reason other than the items demonstrate an association with the criterion. That is, an item was selected if it showed a strong ability to discriminate (differentiate) between clinical and control groups. Using this method and hundreds of items, the MMPI developed 10 clinical scales, which involved operational rules based on previously collected empirical evidence.

But what do you know with this abundance of correlations? You can use data reduction methods to reduce the many variables, based on their inter-correlations, down to a more parsimonious set of factors. But, how do you name each factor, which is composed of many items? The developers originally numbered the MMPI clinical scales from 1 to 10. But numbered scales are not useful for other people, so the factors were eventually given labels (e.g., Paranoia). And if a client received an elevated score on a factor, many people would label the clients as _____ [the name of the factor], such as "paranoid." The MMPI is discussed in further detail in the chapter on objective personality testing.

The idea of dustbowl empiricism was to develop a strong empirical base that would provide a strong foundation that would help us build up to a broad understanding that was integrated, coherent, and systematic. This process was unclear when there was only a table of correlations. Radical operationalists were opposed to content validity because it allows intrusion of our flawed thinking because, according to operationalists, there are no experts. According to this perspective, the content does not matter; we just need enough data to bootstrap ourselves to a better understanding of the constructs.

Although the atheoretical approach can perform reasonably well, it can be improved by making better use of theory. An empirical result (e.g., a correlation) might not necessarily have a lot of meaning associated with it. As the maxim goes, correlation does not imply causation. Just because X is associated with Y does not mean that X causes Y. Consider you find an association between

variables X and Y, consistent with your hypothesis, as depicted in Figure 4.2.

Figure 4.2: Hypothesized causal effect based on an observed association between X and Y, such that X causes Y.

There are three primary reasons that an observed association between X and Y does not necessarily mean that X causes Y. First, the association could reflect the opposite direction of effect, where Y actually causes X, as depicted in Figure 4.3.

Figure 4.3: Reverse (opposite) direction of effect from the hypothesized effect, where Y causes X.

Second, the association could reflect the influence of a third variable. If a third variable is a common cause of each and accounts for their association, it is a *confound*. An observed association between X and Y could reflect a confound—i.e., a cause (Z) that influences both X and Y, which explains why X and Y are correlated even though they are not causally related. A third variable confound that is a common cause of both X and Y is depicted in Figure 4.4.

Third, the association might be spurious. It might just reflect random variation (i.e., chance), and that when tested on an independent sample, what appeared as an association may not hold when testing whether the association generalizes.

However, even if the association between X and Y reflects a causal effect and that X causes Y, it does not necessarily mean that the effect is clinically actionable or useful. An association may reflect a static or unmodifiable predictor that is not practically useful as a treatment target.

As Silver [2012] (p. 9) notes, "The numbers have no way of speaking for themselves. We speak for them. We imbue them with meaning." If we *understand* the variables in the system and how they influence each other, we can predict things more accurately than predicting for the sake of predicting. For instance, we have made great strides in the last decades in making more accurate weather forecasts, including of extreme weather events like hurricanes. These great strides

Figure 4.4: Confounded association between X and Y due to a common cause, Z.

have more to do with a better causal understanding of the weather system and the ability to conduct simulations of the atmosphere than merely because of big data [Silver, 2012]. By contrast, other events are still incredibly difficult to predict, including earthquakes, in large part because we do not have a strong understanding of the system [and because we do not have ways of precisely measuring those causes because they occur at a depth below which we are realistically able to drill; Silver [2012]]. Statistical models applied to big data (i.e., lots of variables and lots of samples) can *overfit* the data, which means that the statistical model accounts for error variance (an overly specific prediction), which will not generalize to future samples. So, even though an over-fitting statistical model appears to be accurate, it is not actually that accurate—it will predict new data less accurately than how accurately it accounts for the data with which the model was built.

In Figure 4.5, the blue line represents the true distribution of the data, and the red line is an over-fitting model:

An important issue in predictive validity is the criterion problem—finding the right criterion. It is important to avoid *criterion contamination*, which is artificial commonality between the measure and the criterion. The criterion is not always a well-measured clear criterion to predict (like predicting death in the medical field). And you may not have access to a predictive criterion until a long time from now. So, what researchers often do is adopt intermediate assessments, which are not actually what they are interested in, but it is related

Figure 4.5: Overfitting model

to the criterion of interest, and it is in a window of time that allows for some meaningful prediction. For instance, intermediate graduate school markers of whether a graduate student will go on to have a successful career could include their grades in graduate school, whether they completed the dissertation, their performance in comprehensive/qualifying exams, etc. However, these intermediate assessments do not always indicate whether or not a student will go on to have a successful career (i.e., they are not always correlated with the real criterion of interest).

So, empiricism is often not enough. It is important to use theory to guide selection of an intermediate criterion that will relate to the real criterion of interest. In psychology, even our long-term criteria are not well defined relative to other sciences. In clinical psychology, for example, we are often predicting to diagnosis, which is not that much more valid compared to our measure/predictor.

At the same time, in psychology, our theories of the causal processes that influence outcomes are not yet very strong. Indeed, I have misgivings calling them theories because they do not meet the traditional scientific standard for a theory. A scientific theory is an explanation of the natural world that is testable and falsifiable, and that has withstood rigorous scientific testing and scrutiny. In psychology, our "theories" are more like conceptual frameworks. And these conceptual frameworks are often vague, do not make specific predictions of effects *and* non-effects, and do not hold up consistently when rigorously tested. As described by Meehl [1978] (pp. 806–807):

> "I consider it unnecessary to persuade you that most so-called 'theories' in the soft areas of psychology (clinical, counseling, social, personality, community, and school psychology) are scientifically unimpressive and technologically worthless…Perhaps the easiest way to convince yourself is by scanning the literature of soft psychology over the last 30 years and noticing what happens to theories. Most of them suffer the fate that General MacArthur ascribed to old generals—They never die, they just slowly fade away. In the developed sciences, theories tend either to become widely accepted and built into the larger edifice of well-tested human knowledge or else they suffer destruction in the face of recalcitrant facts and are abandoned, perhaps regretfully as a 'nice try.' But in fields like personology and social psychology, this seems not to happen. There is a period of enthusiasm about a new theory, a period of attempted application to several fact domains, a period of disillusionment as the negative data come in, a growing bafflement about inconsistent and unreplicable empirical results, multiple resort to ad hoc excuses, and then finally people just sort of lose interest in the thing and pursue other endeavors."

Even if we had strong theoretical understanding of the causal system that influences behavior, we would likely still have difficulty making accurate predictions

because the field has largely relied on relatively crude instruments. According to one philosophical perspective known as LaPlace's demon, if we were able to know the exact conditions of everything in the universe, we would be able to know how the conditions would be in the future. This is an example of scientific determinism, where if you know the initial conditions, you also know the future. Other perspectives, such as quantum mechanics and chaos theory, would say that, even if we knew the initial conditions with 100% certainty, there would still be uncertainty in our understanding of the future. But assume, for a moment, that LaPlace's demon is true. The challenge in psychology is that we have a relatively poor understanding of the initial conditions of the universe. Thus, our predictions would necessarily be probabilistic, similar to weather forecasts. Despite having a strong understanding of how weather systems behave, we have imperfect understanding of the initial conditions (e.g., the position and movement of all molecules) [Silver, 2012].

4.3.1.3.3.1 Psychoanalysis versus Empiricism If we consider "The Hitchhiker's Guide to the Galaxy" according to psychoanalysts, when they visit the new galaxy and they discover the only thing left is cheesecake, but everything is connected to everything else because of grand theories. Psychoanalysts try to reconstruct the entire universe from a sparse bit of data with supposedly strong theoretical understanding (when, in reality, our theories are not so strong). Their "theories" make grand conceptual claims.

Let us contrast psychoanalysts with empiricists/radical operationalism. Figure 4.6 presents a depiction of empiricism. Empiricists evaluate how an observed predictor relates to an observed outcome. The rectangles in the figure represent entities that can be observed. For instance, an empiricism might examine the extent to which a person's blood pressure is related to the the number of words they speak per minute.

Figure 4.6: Depiction of empiricism.

Contrast the empirical approach with psychoanalysis, as depicted in Figure 4.7.

Figure 4.7: Depiction of psychoanalysis.

Circles represent unobserved, latent entities. For instance a psychoanalyst may make a conceptual claim that one observed variable influences another observed variable through a complex chain of intervening processes that are unobservable.

There is a classic and hilarious Monty Python video that is an absurd example akin to radical operationalism taken to the extreme. In the video, the researchers make lots of measurements and predictions. The researchers identify interspecies differences in intelligence where humans show better performance on the English-based intelligence test (who got an IQ score of 100) than the penguins, who got an IQ score of 2. But the penguins did not speak English and were unable to provide answers to the English-based intelligence test. So, the researchers also assessed a group of non-English speaking humans as an attempt to control for language ability. They found that the penguins' scores were equal to the scores of the non-English speaking humans. They argued that, based on their smaller brain and equal IQ when controlling for language ability, that penguins are smarter than humans. However, the researchers clearly made mistakes about confounding variables. And inclusion of a control group of non-English speaking humans does not solve the problem of validity or bias; it just obscures the problem. In summary, radical operationalism provides rich lower-level information, but lacks the broader picture. So, it seems, that we need both theory *and* empiricism. Theory and empiricism can inform each other.

4.3.1.4 Construct validity

[**Furr and S. Heuckeroth CITATION**]

Construct validity is the extent to which a measure accurately assesses a target construct. Construct validity is not quantified by a single index, but rather consists of a "web of evidence" (the totality of evidence), which reflect the sum of inferences from multiple aspects of validity. Construct validity deals with the association between a measure and an unobservable criterion, i.e., a latent construct. By contrast, criterion-related validity deals with an observable criterion.

Construct validity encompasses all other types of measurement validity (e.g., content and criterion-related validity), in addition to:

- Scores on the measure show homogeneity (scores on measure assess a single construct)
- Scores on the measure show theoretically expected developmental changes
- Scores on the measure show theoretically expected group differences
- Scores on the measure show theoretically expected intervention effects
- Establishing the *nomological network* of a construct

4.3.1.4.1 Nomological Network A *nomological network* is [**DEFINE**]. [**Give visual example.**]

With construct validity, we can judge the quality of our measures by how well or how sensibly it fits in nomological network. Latent constructs and observed measures improve each other step-by-step. But there is no established way to evaluate the process.

Historically, construct validity became a way for some researchers to skip out on other types of validity. People found a correlation between a measure's scores and some group membership and argued, therefore, that the measure has construct validity, because there was a theoretically expected correlation between some measure and _____. People started finding a correlation of a measure with other measures, asserting that it provides evidence of construct validity, and saying "that's my nomological network." But, a theoretically expected association is not enough! For example, consider a measure of how quickly someone can count backwards by 7. Performance is impaired in those with schizophrenia, anxiety (due to greater distractibility), and depression (due to slowing and concentration difficulties). Therefore, it is a low-quality claim that counting backwards is part of the phenomenology of these disorders because it lacks differential deficit or discriminant validity [Campbell and Fiske, 1959]. This is related to the "glop problem", which asserts that every bad thing is associated with every other bad thing—there is high comorbidity. Therefore, researchers needed some way to distinguish method variance from trait variance. This led to the development of the multi-trait multi-method matrix.

4.3.1.4.2 Multi-Trait Multi-Method Matrix (MTMM) The *mult-trait multi-method matrix* (MTMM), as proposed by Campbell and Fiske [1959], is a concrete way to evaluate the validity of a measure. The MTMM allows you to split the variance of measures' scores into variance that is due to the method (i.e., method variance or method bias) and variance that is due to the construct (trait) of interest (i.e., construct variance). To create a MTMM, you need at least two methods and at least two constructs. For example, A MTMM could include self-report and observation of depression and introversion. You would then examine the correlations across combinations of construct and method.

For an example of a MTMM, see Figure 4.8. Several aspects of psychometrics can be evaluated with a MTMM, including reliability, convergent validity, and discriminant validity. The reliability diagonal of a MTMM is the correlation of variable with itself, i.e., the test-retest reliability or monotrait-monomethod correlations. The reliability coefficients should be the highest values in the matrix because each measure should be more correlated with itself than with anything else.

4.3.1.4.2.1 Convergent Validity *Convergent validity* is the extent to which a measure is associated with other measures of the same target construct. In a MTMM, convergent validity evaluates whether measures targeting the same construct, but using different methods, converge upon the the same

A Synthetic Multitrait-Multimethod Matrix

		Method 1			Method 2			Method 3		
	Traits	A_1	B_1	C_1	A_2	B_2	C_2	A_3	B_3	C_3
Method 1	A_1	(.89)								
	B_1	.51	(.89)							
	C_1	.38	.37	(.76)						
Method 2	A_2	.57	.22	.09	(.93)					
	B_2	.22	.57	.10	.68	(.94)				
	C_2	.11	.11	.46	.59	.58	(.84)			
Method 3	A_3	.56	.22	.11	.67	.42	.33	(.94)		
	B_3	.23	.58	.12	.43	.66	.34	.67	(.92)	
	C_3	.11	.11	.45	.34	.32	.58	.58	.60	(.85)

Note.—The validity diagonals are the three sets of italicized values. The reliability diagonals are the three sets of values in parentheses. Each heterotrait-monomethod triangle is enclosed by a solid line. Each heterotrait-heteromethod triangle is enclosed by a broken line.

Figure 4.8: Multi-trait multi-method matrix. Figure reprinted from @Campbell1959, Table 1, p. 82. Campbell, D. T., & Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin*, *56*, 81-105. https://doi.org/10.1037/h0046016

construct. These are observed in the validity diagonals, also known as the convergent correlations or the monotrait-heteromethod correlations. For strong convergent validity, we would expect the values in the validity diagonals to be significant and high-ish.

4.3.1.4.2.2 Discriminant (Divergent) Validity You can also evaluate the *discriminant validity*, also called divergent validity, of a measure in the context of a MTMM. Discriminant validity is the extent to which a measure is *not* associated with measures of different constructs that are not theoretically expected to be related. In a MTMM, discriminant validity determine the extent to which a measure does not correlate with measures that share a method but assess different constructs. For strong discriminant validity, we would expect the discriminant correlations (heterotrait monomethod) should be low.

According to Campbell and Fiske [1959], discriminant validity of measures can be established when three criteria are met:

1. The convergent (monotrait-heteromethod) correlations are stronger than heterotrait-heteromethod correlations. This provides the weakest evidence for discriminant validity. That is, the convergent correlations are higher than the values in the same column or row in the same heteromethod block. This can be evaluated using the heterotrait-monotrait ratio (described below).
2. The convergent (monotrait-heteromethod) correlations are stronger than discriminant correlations (monomethod-heterotrait). This provides stronger evidence of discriminant validity.
3. The patterns of intercorrelations between constructs are the same, regardless of which measurement method is used. That is, the pattern of inter-trait associations is the same in all triangles. For example, if extraversion and anxiety are moderately correlated with each other but uncorrelated with achievement, we would expect this pattern of interrelationships between constructs would hold, regardless of which method was used to assess the construct.

You can estimate a measure's degree of discriminant validity based on the *heterotrait-monotrait ratio* [HTMT; Henseler et al. [2015]]. HTMT is the average of the heterotrait-heteromethod correlations (i.e., the correlations of measures from different measurement methods that assess different constructs), relative to the average of the monotrait-heteromethod correlations (i.e., the correlations of measures from different measurement methods that assess the same construct). As described here (http://www.henseler.com/htmt.html) based on evidence from Voorhees et al. [2016], "If the HTMT is clearly smaller than one, discriminant validity can be regarded as established. In many practical situations, a threshold of 0.85 reliably distinguishes between those pairs of latent variables that are discriminant valid and those that are not." HTMT was calculated using the `semTools` package [Jorgensen et al., 2021].

```
modelCFA <- '
  visual  =~ x1 + x2 + x3
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9
'

htmt(modelCFA,
     data = HolzingerSwineford1939,
     missing = "ml")
```

```
##         visual textul speed
## visual  1.000
## textual 0.424  1.000
## speed   0.467  0.290  1.000
```

Another index estimates the degree of correspondence of convergent and discriminant validity correlations with a priori hypotheses as an index of construct validity [Furr and Heuckeroth, 2019].

Using a MTMM, a researcher can learn a lot about the quality of a measure and build a nomological network. For instance, we can estimate the extent of method variance, or variance that is attributable to the measurement method rather than the construct of interest. Method variance is estimated by the difference between monomethod vs. heteromethod blocks.

A poor example of a MTMM would be having two constructs such as height and depression, and two methods such as Likert and true/false. This would be a poor MTMM for two reasons: 1) there are trivial differences in the methods, which would lead to obvious convergence, and 2) the differences in between constructs are not important—they are obviously discriminant. Using such a MTMM would find strong convergent associations, because the methods are maximally similar, and weak discriminant associations, because the constructs are maximally different. It would be better to use maximally different measurement methods (e.g., self-report and performance-based measures), and to use constructs that are important to distinguish (e.g., depression and anxiey).

The paper by Campbell and Fiske [1959] that introduced the MTMM is one of the most widely cited papers in psychology of all time. *Psychological Bulletin* published a more recent paper by Fiske and Campbell [1992], entitled "Citations Do Not Solve Problems." They noted how their original paper was the most widely cited paper in the history of *Psychological Bulletin*, but they argued that nothing came of their paper. The MTMM matrices published today show little-to-no improvement compared to the ones they published in 1959. In part, this may be because we need better measures (i.e., a higher ratio of construct to method variance).

4.3.1.4.2.3 Multi-Trait Multi-Method Correlation Matrix A multi-trait multi-method correlation matrix is below.

```
paged_table(data.frame(round(cor(MTMM), 2)))
```

```
kable(data.frame(round(cor(MTMM), 2)),
      booktabs = TRUE)
```

	VO1	VO2	VO3	VW1	VW2	VW3	VM1	VM2	VM3	SO1	SO2	SO3	SW1	SW2	S
VO1	1.00	0.55	0.54	0.48	0.48	0.50	0.50	0.51	0.49	0.32	0.34	0.36	0.30	0.27	0
VO2	0.55	1.00	0.57	0.48	0.52	0.50	0.53	0.53	0.50	0.36	0.39	0.37	0.33	0.30	0
VO3	0.54	0.57	1.00	0.49	0.49	0.48	0.48	0.52	0.50	0.35	0.38	0.37	0.30	0.31	0
VW1	0.48	0.48	0.49	1.00	0.61	0.57	0.48	0.48	0.50	0.21	0.28	0.25	0.35	0.37	0
VW2	0.48	0.52	0.49	0.61	1.00	0.58	0.52	0.51	0.54	0.26	0.33	0.30	0.42	0.40	0
VW3	0.50	0.50	0.48	0.57	0.58	1.00	0.51	0.48	0.51	0.27	0.28	0.28	0.36	0.38	0
VM1	0.50	0.53	0.48	0.48	0.52	0.51	1.00	0.54	0.57	0.25	0.29	0.29	0.29	0.28	0
VM2	0.51	0.53	0.52	0.48	0.51	0.48	0.54	1.00	0.57	0.29	0.29	0.33	0.31	0.29	0
VM3	0.49	0.50	0.50	0.50	0.54	0.51	0.57	0.57	1.00	0.25	0.31	0.29	0.29	0.30	0
SO1	0.32	0.36	0.35	0.21	0.26	0.27	0.25	0.29	0.25	1.00	0.57	0.56	0.48	0.48	0
SO2	0.34	0.39	0.38	0.28	0.33	0.28	0.29	0.29	0.31	0.57	1.00	0.58	0.51	0.53	0
SO3	0.36	0.37	0.37	0.25	0.30	0.28	0.29	0.33	0.29	0.56	0.58	1.00	0.47	0.47	0
SW1	0.30	0.33	0.30	0.35	0.42	0.36	0.29	0.31	0.29	0.48	0.51	0.47	1.00	0.59	0
SW2	0.27	0.30	0.31	0.37	0.40	0.38	0.28	0.29	0.30	0.48	0.53	0.47	0.59	1.00	0
SW3	0.24	0.30	0.28	0.31	0.41	0.37	0.28	0.28	0.25	0.51	0.51	0.48	0.61	0.61	1
SM1	0.24	0.30	0.31	0.26	0.28	0.23	0.30	0.34	0.30	0.47	0.51	0.47	0.49	0.51	0
SM2	0.28	0.33	0.30	0.27	0.31	0.26	0.32	0.36	0.34	0.47	0.49	0.48	0.50	0.47	0
SM3	0.30	0.32	0.31	0.27	0.32	0.31	0.37	0.37	0.34	0.50	0.51	0.50	0.49	0.51	0
QO1	0.43	0.47	0.42	0.34	0.35	0.34	0.34	0.36	0.34	0.34	0.40	0.35	0.34	0.33	0
QO2	0.41	0.45	0.43	0.31	0.33	0.30	0.32	0.34	0.37	0.36	0.39	0.36	0.32	0.35	0
QO3	0.44	0.44	0.45	0.33	0.35	0.30	0.37	0.39	0.39	0.34	0.40	0.38	0.27	0.31	0
QW1	0.31	0.33	0.33	0.43	0.43	0.39	0.33	0.35	0.36	0.25	0.30	0.28	0.37	0.42	0
QW2	0.35	0.36	0.34	0.43	0.41	0.40	0.35	0.34	0.39	0.27	0.34	0.31	0.39	0.43	0
QW3	0.32	0.35	0.33	0.43	0.44	0.38	0.34	0.33	0.36	0.27	0.30	0.30	0.37	0.45	0
QM1	0.34	0.39	0.36	0.33	0.35	0.32	0.42	0.41	0.43	0.30	0.36	0.31	0.30	0.35	0
QM2	0.36	0.39	0.38	0.32	0.34	0.32	0.41	0.39	0.41	0.30	0.36	0.33	0.29	0.35	0
QM3	0.35	0.40	0.31	0.35	0.34	0.32	0.42	0.38	0.42	0.31	0.34	0.31	0.29	0.35	0

```
kable(data.frame(round(cor(MTMM), 2)),
      booktabs = TRUE)
```

4.3.1.4.2.4 Construct validity beyond Campbell and Fiske There are a number of other approaches that can be helpful for establishing construct

validity in ways that go beyond the approaches proposed by Campbell and Fiske [1959]. One way is known as triangulation. Triangulation is conceptually depicted in Figure 4.9. Triangulation involves testing a hypothesis with multiple measures and/or methods to see if the findings are consistent, that is, whether the findings triangulate.

Figure 4.9: Using triangulation to arrive a closer estimate of the construct using multiple measures and/or methods.

MTMM in Confirmatory Factor Analysis (CFA)

Using modern modeling approaches, there are even more advanced ways of examining a MTMM. For instance, you can use structural equation modeling or confirmatory factor analysis to derive a latent variable of a construct from multiple methods to be free from method-related error variance to generate purer assessments of the construct to see how it relates to other constructs. For an example of a MTMM in confirmatory factor analysis, see Figure 4.10.

This example comes from W. Joel Schneider: http://my.ilstu.edu/~wjschne/442/CFA.html#(15). The confirmatory factor analysis (CFA) model was fit in the `lavaan` package [Rosseel et al., 2020].

```
modelMTMM <- '
  g =~ Verbal + Spatial + Quant
  Verbal =~
    VO1 + VO2 + VO3 +
    VW1 + VW2 + VW3 +
    VM1 + VM2 + VM3
  Spatial =~
    SO1 + SO2 + SO3 +
    SW1 + SW2 + SW3 +
    SM1 + SM2 + SM3
  Quant =~
    QO1 + QO2 + QO3 +
```

Figure 4.10: Multi-trait multi-method model in confirmatory factor analysis with three constructs (internalizing, externalizing, and thought-disordered problems) and three methods (mother-, father-, and teacher-report).

```
    QW1 + QW2 + QW3 +
    QM1 + QM2 + QM3
  Oral =~
    VO1 + VO2 + VO3 +
    SO1 + SO2 + SO3 +
    QO1 + QO2 + QO3
  Written =~
    VW1 + VW2 + VW3 +
    SW1 + SW2 + SW3 +
    QW1 + QW2 + QW3
  Manipulative =~
    VM1 + VM2 + VM3 +
    SM1 + SM2 + SM3 +
    QM1 + QM2 + QM3
'
MTMM.fit <- cfa(modelMTMM,
                data = MTMM,
                orthogonal = TRUE,
                missing = "ml")

summary(MTMM.fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 107 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                       111
##
##   Number of observations                          1000
##   Number of missing patterns                         1
##
## Model Test User Model:
##
##   Test statistic                               321.321
##   Degrees of freedom                               294
##   P-value (Chi-square)                           0.131
##
## Model Test Baseline Model:
##
##   Test statistic                             14923.912
##   Degrees of freedom                               351
##   P-value                                        0.000
```

```
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.998
##   Tucker-Lewis Index (TLI)                       0.998
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -31375.377
##   Loglikelihood unrestricted model (H1)     -31214.716
##
##   Akaike (AIC)                               62972.753
##   Bayesian (BIC)                             63517.514
##   Sample-size adjusted Bayesian (BIC)        63164.972
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.010
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.016
##   P-value RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.020
##
## 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
##   g =~
##     Verbal            1.000                               0.805    0.805
##     Spatial           0.890    0.062   14.245    0.000    0.720    0.720
##     Quant             1.058    0.072   14.791    0.000    0.862    0.862
##   Verbal =~
##     VO1               1.000                               0.704    0.692
##     VO2               1.053    0.048   21.724    0.000    0.741    0.721
##     VO3               0.996    0.047   21.190    0.000    0.701    0.693
##     VW1               0.965    0.048   20.224    0.000    0.680    0.691
##     VW2               1.046    0.050   21.017    0.000    0.736    0.721
##     VW3               1.025    0.051   20.254    0.000    0.721    0.693
##     VM1               1.006    0.050   20.126    0.000    0.708    0.715
```

```
##     VM2              1.040    0.051   20.267    0.000    0.732    0.717
##     VM3              1.064    0.052   20.325    0.000    0.749    0.729
##   Spatial =~
##     SO1              1.000                              0.700    0.692
##     SO2              1.039    0.047   21.902    0.000    0.728    0.729
##     SO3              0.998    0.048   20.871    0.000    0.699    0.689
##     SW1              1.000    0.049   20.258    0.000    0.701    0.704
##     SW2              1.035    0.050   20.558    0.000    0.725    0.712
##     SW3              1.029    0.050   20.670    0.000    0.721    0.706
##     SM1              1.010    0.052   19.389    0.000    0.707    0.696
##     SM2              0.995    0.052   19.210    0.000    0.697    0.690
##     SM3              1.022    0.051   19.899    0.000    0.716    0.711
##   Quant =~
##     QO1              1.000                              0.695    0.706
##     QO2              0.999    0.046   21.830    0.000    0.694    0.696
##     QO3              1.086    0.049   22.236    0.000    0.755    0.708
##     QW1              0.980    0.049   20.082    0.000    0.681    0.680
##     QW2              1.047    0.050   21.061    0.000    0.728    0.715
##     QW3              1.046    0.050   20.753    0.000    0.728    0.698
##     QM1              1.062    0.051   20.770    0.000    0.739    0.734
##     QM2              1.050    0.052   20.209    0.000    0.730    0.716
##     QM3              1.058    0.052   20.180    0.000    0.736    0.715
##   Oral =~
##     VO1              1.000                              0.255    0.251
##     VO2              0.978    0.194    5.053    0.000    0.250    0.243
##     VO3              1.133    0.210    5.401    0.000    0.289    0.286
##     SO1              0.985    0.204    4.830    0.000    0.252    0.249
##     SO2              0.937    0.196    4.769    0.000    0.239    0.240
##     SO3              0.996    0.204    4.882    0.000    0.254    0.251
##     QO1              0.968    0.188    5.145    0.000    0.247    0.251
##     QO2              1.236    0.229    5.395    0.000    0.316    0.316
##     QO3              1.346    0.245    5.493    0.000    0.344    0.322
##   Written =~
##     VW1              1.000                              0.310    0.315
##     VW2              1.107    0.139    7.958    0.000    0.343    0.336
##     VW3              0.940    0.134    7.021    0.000    0.292    0.280
##     SW1              0.934    0.141    6.636    0.000    0.290    0.291
##     SW2              1.109    0.152    7.315    0.000    0.344    0.338
##     SW3              1.145    0.165    6.955    0.000    0.355    0.348
##     QW1              1.042    0.144    7.258    0.000    0.323    0.323
##     QW2              0.871    0.131    6.668    0.000    0.270    0.265
##     QW3              1.154    0.154    7.488    0.000    0.358    0.344
##   Manipulative =~
##     VM1              1.000                              0.217    0.219
##     VM2              0.927    0.296    3.137    0.002    0.202    0.197
##     VM3              0.933    0.268    3.480    0.001    0.203    0.198
```

```
##     SM1               1.154    0.376    3.069    0.002    0.251    0.247
##     SM2               0.934    0.324    2.880    0.004    0.203    0.201
##     SM3               1.147    0.329    3.487    0.000    0.249    0.247
##     QM1               0.944    0.273    3.457    0.001    0.205    0.204
##     QM2               0.891    0.271    3.286    0.001    0.194    0.190
##     QM3               0.892    0.263    3.390    0.001    0.194    0.188
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   g ~~
##     Oral              0.000                               0.000    0.000
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##   Oral ~~
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##   Written ~~
##     Manipulative      0.000                               0.000    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .VO1               0.040    0.032    1.256    0.209    0.040    0.040
##    .VO2               0.023    0.033    0.694    0.487    0.023    0.022
##    .VO3               0.023    0.032    0.712    0.477    0.023    0.023
##    .VW1              -0.003    0.031   -0.082    0.934   -0.003   -0.003
##    .VW2               0.016    0.032    0.493    0.622    0.016    0.016
##    .VW3               0.034    0.033    1.046    0.295    0.034    0.033
##    .VM1               0.064    0.031    2.037    0.042    0.064    0.064
##    .VM2               0.015    0.032    0.476    0.634    0.015    0.015
##    .VM3               0.061    0.032    1.882    0.060    0.061    0.060
##    .SO1               0.013    0.032    0.408    0.683    0.013    0.013
##    .SO2              -0.021    0.032   -0.653    0.514   -0.021   -0.021
##    .SO3               0.004    0.032    0.110    0.913    0.004    0.003
##    .SW1               0.002    0.031    0.070    0.944    0.002    0.002
##    .SW2               0.006    0.032    0.196    0.845    0.006    0.006
##    .SW3               0.013    0.032    0.402    0.688    0.013    0.013
##    .SM1               0.066    0.032    2.044    0.041    0.066    0.065
##    .SM2               0.008    0.032    0.241    0.810    0.008    0.008
##    .SM3               0.004    0.032    0.118    0.906    0.004    0.004
##    .QO1               0.046    0.031    1.476    0.140    0.046    0.047
##    .QO2               0.025    0.032    0.782    0.434    0.025    0.025
##    .QO3               0.023    0.034    0.693    0.489    0.023    0.022
##    .QW1               0.047    0.032    1.486    0.137    0.047    0.047
##    .QW2               0.044    0.032    1.381    0.167    0.044    0.044
##    .QW3               0.006    0.033    0.173    0.862    0.006    0.005
##    .QM1               0.068    0.032    2.146    0.032    0.068    0.068
```

```
##    .QM2               0.011    0.032    0.328    0.743    0.011    0.010
##    .QM3               0.045    0.033    1.388    0.165    0.045    0.044
##     g                 0.000                               0.000    0.000
##    .Verbal            0.000                               0.000    0.000
##    .Spatial           0.000                               0.000    0.000
##    .Quant             0.000                               0.000    0.000
##     Oral              0.000                               0.000    0.000
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .VO1               0.475    0.025   18.967    0.000    0.475    0.459
##    .VO2               0.445    0.024   18.614    0.000    0.445    0.421
##    .VO3               0.448    0.025   18.202    0.000    0.448    0.438
##    .VW1               0.410    0.022   18.565    0.000    0.410    0.424
##    .VW2               0.381    0.022   17.473    0.000    0.381    0.366
##    .VW3               0.479    0.025   19.164    0.000    0.479    0.441
##    .VM1               0.434    0.024   18.027    0.000    0.434    0.441
##    .VM2               0.466    0.025   18.680    0.000    0.466    0.447
##    .VM3               0.452    0.024   18.520    0.000    0.452    0.429
##    .SO1               0.470    0.025   18.868    0.000    0.470    0.459
##    .SO2               0.410    0.022   18.393    0.000    0.410    0.411
##    .SO3               0.476    0.025   18.894    0.000    0.476    0.463
##    .SW1               0.415    0.022   18.855    0.000    0.415    0.419
##    .SW2               0.394    0.022   17.741    0.000    0.394    0.380
##    .SW3               0.396    0.023   17.517    0.000    0.396    0.380
##    .SM1               0.468    0.027   17.530    0.000    0.468    0.454
##    .SM2               0.492    0.026   19.023    0.000    0.492    0.483
##    .SM3               0.440    0.025   17.359    0.000    0.440    0.433
##    .QO1               0.426    0.022   18.961    0.000    0.426    0.439
##    .QO2               0.415    0.024   17.581    0.000    0.415    0.416
##    .QO3               0.450    0.026   17.092    0.000    0.450    0.395
##    .QW1               0.434    0.023   18.613    0.000    0.434    0.433
##    .QW2               0.433    0.023   19.013    0.000    0.433    0.418
##    .QW3               0.428    0.024   17.779    0.000    0.428    0.394
##    .QM1               0.426    0.023   18.231    0.000    0.426    0.420
##    .QM2               0.469    0.025   18.886    0.000    0.469    0.451
##    .QM3               0.480    0.025   18.884    0.000    0.480    0.453
##     g                 0.321    0.034    9.368    0.000    1.000    1.000
##    .Verbal            0.175    0.020    8.707    0.000    0.353    0.353
##    .Spatial           0.237    0.023   10.293    0.000    0.482    0.482
##    .Quant             0.124    0.018    6.804    0.000    0.257    0.257
##     Oral              0.065    0.019    3.399    0.001    1.000    1.000
##     Written           0.096    0.020    4.788    0.000    1.000    1.000
##     Manipulative      0.047    0.021    2.291    0.022    1.000    1.000
```

```
##
## R-Square:
##                    Estimate
##     VO1               0.541
##     VO2               0.579
##     VO3               0.562
##     VW1               0.576
##     VW2               0.634
##     VW3               0.559
##     VM1               0.559
##     VM2               0.553
##     VM3               0.571
##     SO1               0.541
##     SO2               0.589
##     SO3               0.537
##     SW1               0.581
##     SW2               0.620
##     SW3               0.620
##     SM1               0.546
##     SM2               0.517
##     SM3               0.567
##     QO1               0.561
##     QO2               0.584
##     QO3               0.605
##     QW1               0.567
##     QW2               0.582
##     QW3               0.606
##     QM1               0.580
##     QM2               0.549
##     QM3               0.547
##     Verbal            0.647
##     Spatial           0.518
##     Quant             0.743
```

Path Diagram:

A path diagram of the model is depicted using the `semPlot` package [Epskamp, 2019].

```
semPaths(MTMM.fit,
         what = "std",
         layout = "tree3",
         optimizeLatRes = FALSE,
         bifactor = c("Verbal","Spatial","Quant","g"),
         exoVar = FALSE)
```

Figure 4.11: Multi-trait multi-method model in confirmatory factor analysis

4.3.1.5 Incremental Validity

Accuracy is not enough for a measure to be useful. Proposed by Sechrest [1963], measures should also be judged by the extent to which they provide an increment in predictive efficiency over the information otherwise easily and cheaply available. *Incremental validity* deals with the incremental value or utility over a measure beyond other sources of information. It must be demonstrated that the addition of a measure will produce better predictions than are made on the basis of information ordinarily available. It is not enough to show that the measure is better than chance, and the measure should not just be capitalizing on shared method variance with criterion or on increased reliability of the measure. That is, the measure should explain truly unique variance—variance that was not explained before. Incremental validity demonstrates added value, unless the other measure is cheaper or less time consuming. Incremental validity is a specific kind of criterion-related validity: significantly increased R^2 in hierarchical regression. The incremental validity of a measure can be evaluated by examining whether the measure explains significant unique variance in the criterion when accounting for other information, such as easily accessible information, traditionally available measures, or the current gold-standard measure. The extent of incremental validity of a measure can be quantified with the change in the coefficient of determination (ΔR^2) that compares (a) the model that includes the old predictor(s) to (b) the model that includes includes old predictors and the new predictor (measure).

```
model1 <- lm(criterion ~ oldpredictor, data = na.omit(mydataValidity))
model2 <- lm(criterion ~ oldpredictor + predictor, data = na.omit(mydataValidity))

summary(model1)
```

```
##
## Call:
## lm(formula = criterion ~ oldpredictor, data = na.omit(mydataValidity))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.1340  -4.5455  -0.0465   4.5769  24.3385
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.20333    1.38483   15.31   <2e-16 ***
## oldpredictor  0.78767    0.01367   57.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.906 on 898 degrees of freedom
```

```
## Multiple R-squared:  0.7872, Adjusted R-squared:  0.787
## F-statistic:  3322 on 1 and 898 DF,  p-value: < 2.2e-16
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = criterion ~ oldpredictor + predictor, data = na.omit(mydataValidity))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.7515  -4.3807  -0.1707   4.0494  23.1528
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.96982    1.32330   12.82   <2e-16 ***
## oldpredictor  0.65974    0.01626   40.57   <2e-16 ***
## predictor     0.34027    0.02727   12.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.378 on 897 degrees of freedom
## Multiple R-squared:  0.8187, Adjusted R-squared:  0.8183
## F-statistic:  2025 on 2 and 897 DF,  p-value: < 2.2e-16
```

```
model1Rsquare <- summary(model1)$r.squared
model2Rsquare <- summary(model2)$r.squared

model1RsquareAdj <- summary(model1)$adj.r.squared
model2RsquareAdj <- summary(model2)$adj.r.squared

deltaRsquare <- getDeltaRsquare(model2)[["predictor"]]
```

```
## The deltaR-square values: the change in the R-square
##       observed when a single term is removed.
## Same as the square of the 'semi-partial correlation coefficient'
##              deltaRsquare
## oldpredictor   0.33272659
## predictor      0.03146744
```

```
deltaRsquareAdj <- model2RsquareAdj - model1RsquareAdj

anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: criterion ~ oldpredictor
## Model 2: criterion ~ oldpredictor + predictor
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    898 42825
## 2    897 36492  1    6332.9 155.67 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
getDeltaRsquare(model2)
```

```
## The deltaR-square values: the change in the R-square
##       observed when a single term is removed.
## Same as the square of the 'semi-partial correlation coefficient'
##              deltaRsquare
## oldpredictor   0.33272659
## predictor      0.03146744
```

ΔR^2 was calculated using the `rockchalk` package [Johnson, 2019]. The predictor shows significant incremental validity above the old predictor in predicting the criterion. `Model 1` explains 78.72% of the variance ($R^2 = 0.79$), and `Model 2` explains 81.87% of the variance ($R^2 = 0.82$). Thus, the predictor explains only 3.15% additional variance in the criterion above the variance explained by the old predictor (ΔR^2). Based on adjusted R^2 (R^2_{adj}), the predictor explains only 3.13% additional variance in the criterion above the variance explained by the old predictor (ΔR^2_{adj}).

4.3.1.6 Treatment Utility of Assessment

Hayes et al. [1987] argued that it is not enough for measures to be reliable and valid. The authors raised another important consideration for the validity of a measure: its treatment utility or usefulness. They asked the question: what goal do assessments accomplish in clinical psychology? In clinical psychology, the goal of assessment is to lead to better treatment outcomes. Therefore, the *treatment utility* of a measure is the extent to which a measure is shown to contribute to beneficial treatment outcomes. That is, if a clinician has the information results from this having administered this measure, do the clients have better outcomes? The treatment utility of assessment is a specific kind of criterion-related validity, with the idea that "all criteria are not created equal." And the criterion that is most important to optimize, from this perspective, when developing and selecting measures is a client's treatment outcomes.

Hayes et al. [1987] described different approaches to evaluating the extent to which a measure shows treatment utility. These are a priori group comparison approaches that examine whether a specific difference in the assessment

approach relates to treatment outcome. They distinguished between (a) manipulated assessment and (b) manipulated use.

In *manipulated assessment*, a single group of subjects is randomly divided into two subgroups and either the collection or availability of assessment data is varied systematically. Therapists then design or implement treatment in accord with the data available. As an example, the measure of interest may be administered in one condition, and the standard measure may be administered in the other condition. Then the treatment outcomes would be compared across the two conditions. An advantage of manipulated assessment is that this type of design is more realistic than manipulated use. When making the assessment data available but not assigning a certain treatment based on the assessment outcomes better simulates a realistic clinical environment. Also, because the control group has no access to the data, it might serve as a stronger control. A disadvantage of manipulated assessment is that it depends on whether and how clinicians use the measure, which depends on how positively the measure was received by the clinicians.

In *manipulated use*, the same assessment information is available for all subjects, but the researcher manipulates the way in which the assessment information is used. For example, one group gets a treatment matched to assessment outcomes, and the other group gets a cross-matched treatment that does not not target the problem identified by the assessment. So, in one group, the assessment information is used to match the treatment to the client based on their assessment results, whereas the other group receives a standard treatment regardless of their assessment results. An advantage of this design is that you can be certain that the treatment decisions are explicitly informed by the assessment results because the researcher ensures this, whereas the decision of how the assessment information is used is up to the clinician in manipulated assessment.

4.3.1.7 Discriminative Validity

Discriminative validity is the degree to which a measure accurately identifies persons placed into groups on the basis of another measure. Discriminative validity is not to be confused with discriminant (divergent) validity and discriminant analysis. A measure shows discriminant validity if it does not correlate with things that it would not be theoretically expected to correlated with. A measure shows discriminative validity, by contrast, if the measure is able to accurately differentiate things (e.g., two groups such as men and women). Discriminant analysis is a model that combines predictors to differentiate between multiple groups.

4.3.1.8 Elaborative Validity

Elaborative validity involves the extent to which a measure increases our theoretical understanding of the target construct or of neighboring constructs. It

deals with a measure's meaningfulness. Elaborative validity is a type of incremental theoretical validity. It is a combination of criterion-related validity and construct validity that examines how much it increases our understanding of a construct's nomological network. However, I am unaware of strong examples of measures that show strong elaborative validity in psychology.

4.3.1.9 Consequential Validity

Consequential validity is a form of validity that differs from evidential validity, or a test's potential to provide accurate, useful information based on research. Consequential validity takes a more macro view and deals with the community, sociological, and public policy perspective. *Consequential validity* evaluates the consequences of our measures beyond the circumstances of their development, based on their impact. Measures can have positive, negative, or mixed effects on society. An example of consequential validity would be asking what the impacts of aptitude testing are on society—how aptitude testing affects society in the long run. Some would argue that, even in cases where the aptitude tests have some degree of evidential validity (i.e., they accurately assess to some degree what they tend to assess), they are consequentially invalid—that is, they have had a net negative effect on society and, due to their low value to society, are invalid. The tests themselves may be fine in terms of their accuracy, but consequential validity says that their validity depends on what we do with the test, i.e., how the test is used.

Another example of consequential validity is when the validity of a measure changes over time due to changes in people's or society's response, as depicted in Figure 4.12. Judgments and predictions can change the way society reacts and the way people behave so that the predictions become either more or less accurate. A prediction that becomes more accurate as a result of people's response to a prediction is a self-fulfilling prediction or prophecy. For instance, if school staff make a prediction that a child will be held back a grade, the teacher may not provide adequate opportunities for the child to learn, which may lead to the child being more likely to be held back.

A prediction that becomes less accurate as a result of people's response to a prediction is a self-canceling prediction. The most effective prediction about a flu outbreak might be one that leads people to safer behavior, therefore, lower flu rates which do not correspond to the initial prediction [Silver, 2012]. Society's response to a measure can invalidate the measure. For example, consider that an organization rates cities based on quality-of-life measures. If quality-of-life indices include the percent of solved cases by the city's police (i.e., the clearance rate), cities may try to improve their ratings of "quality-of-life" by increasing their clearance rate either by increasing the number of cases they mark as solved (such as by marking false resolutions) or by decreasing the number of cases (such as by investigating fewer cases). That is, cities may "game" the system based on the quality-of-life indices used by various organizations. In this case, the

Figure 4.12: Invalidation of a measure due to society's response to the use of the measure.

clearance rate becomes a less accurate measure of a city's quality of life.

Another example of a measure that becomes less valid due to society's response is the use of alumni donations as an indicator of the strength of a university that is used for generating university rankings. Such an indicator could lead schools to admit wealthier students who give the most donations, and students whose parents were alumni and provide lavish donations to the university. Yet another example could be standardized testing, where instructors may "teach to the test" such that better performance might not necessarily reflect better underlying competence.

4.3.1.10 Representational Validity

Representational validity examines the extent to which the items or content of a measure "flesh out" and mirror the true nature and mechanisms of the construct. There are many different types of validity, but many of them are overlapping and can be considered types of other forms of validity. For instance, representational validity, is a type of elaborative validity and content validity.

4.3.1.11 Factorial (Structural) Validity

Factorial validity is examined in Section 12.3 on factor analysis. *Factorial validity* considers whether the factor structure of the measure(s) are consistent with the construct. According to factorial validity, if you claim that the measure is unidimensional with four items, factor analysis should support the unidimensionality. Thus, it involves testing empirically whether the measure has the same structure as would be expected based on theory. It is a type of construct validity.

4.3.1.12 Ecological Validity

Ecological validity examines the extent to which a measure provides scores that are indicative of the behavior of a person in the natural environment.

4.3.1.13 Process-Focused Validity

Process-focused validity attempts to get closer to the underlying mechanisms. *Process-focused validity* examines the degree to which respondents engage in a predictable set of psychological processes (which are specified a priori) when completing the measure [Bornstein, 2011, Furr, 2017]. These psychological processes include effects of the instrument (e.g., observer versus third-party report versus projective test) as well as effects of the context (e.g., assessment setting, assessment instructions, affect state of the participant, etc.). To determine whether a test is valid in the process-focused technique, one experimentally manipulates variables that moderate the test score-criterion association—to better understand the underlying processes. The ideal outcome is a measure that both (1) has an adequate outcome (correlations where expected) as well as (2) adequate process validity—the psychological processes that one engages in are well hypothesized.

The idea of process-focused validity is that if a measure does not inform us about process or mechanisms, it is not worth doing. Process-focused validity is a type of elaborative validity and construct validity. For instance, consider the common finding that low socioeconomic status is associated with poorer outcomes. To make an impact, process-focused validity would argue that we need to know the mechanisms that underlie this association, and it involves how a measure helps us understand process.

4.3.2 Research Design (Experimental) Validity

Aspects of *research design validity*, also called experimental validity, involve the validity of a research design for making various interpretations. Research design validity includes internal validity, external validity, and conclusion validity.

4.3.2.1 Internal Validity

Internal validity is the extent to which causal inference is justified from the research design. This encompasses multiple features including:

- temporal precedence—does the cause occur before the effect
- covariation of cause and effect—correlation is necessary (even if insufficient) for causality

- no plausible alternative explanations—such as third variables that could influence both variables and explain their covariation

There are number of potential threats to internal validity, which are important to consider when designing studies and interpreting findings. [**LIST EXAMPLES**]

Research designs differ in the extent to which they show internal versus external validity. An experiment is a research design in which one or more variables (independent variables) are manipulated to observe how the manipulation influences the dependent variable. In an experiment, the researcher has greater control over the variables and attempts to hold everything else constant (e.g., by standardization and random assignment). In correlational designs, however, the researcher has less control over the variables. They may be able to statistically account for potential confounds using covariates or the reverse direction of effect using longitudinal designs. Nevertheless, we can have greater confidence about whether a variable influences another variable in a experiment. Thus, experiments tend to have higher internal validity than correlational designs.

4.3.2.2 External Validity

External validity is the extent to which the findings can be generalized to the broader population and the real world. External validity is crucial to consider for studies that intend to make inferences to people outside of those who were assessed. For instance, norm-referenced assessments attempt to identify the distribution of scores for a given population from a sample within that population. The validity of the norms of a norm-referenced assessment are important to consider [Achenbach, 2001]. The validity of norms and external validity, more broadly, can depend highly on how representative the sample is of the target population and how appropriate this population is to a given participant. Some measures are known to have poor norms, including the Exner Comprehensive System [**CITATION**] for administering and scoring the Rorschach inkblot test, which has been known to over-pathologize [**CITATION**]. The Rorschach is discussed in greater detail in Chapter 17.

As the internal validity of a study's design increases, its external validity tends to decrease. The greater control we have over variables (and, therefore, have greater confidence about causal inferences), the lower the likelihood that the findings reflect what happens in the real world, because it is studying things in a metaphorical vacuum.

4.3.2.2.1 Tradeoffs of internal validity and external validity It is important to note that there is a tradeoff between internal and external validity—there is not a single research design cannot have both high internal and high external validity. Some research designs are better suited for making causal

inferences, whereas other designs tend to be better suited for making inferences that generalize to the real world. The research design that is best suited to making causal inferences is an experiment, where the researcher manipulates one variable (the independent variable) and holds all other variables constant to see how a change in the independent variable influences the outcome (dependent variable). Thus, experiments tend to have higher internal validity than other research designs. However, by manipulating one variable and holding everything else constant, the research takes place in a very standardized fashion that can become like studying a process in a vacuum. So, even if a process is theoretically causal in a vacuum, it may act very differently in the real world when it interacts with other processes. So, observational designs have greater capacity for external validity than experimental designs because people can be observed in their natural environments to see how variables are related in the real world. However, the greater external validity comes at a cost of lower internal validity. Observational designs are not well-positioned to make causal inferences because they have multiple threats to internal validity, including issues of temporal precedence in cross-sectional observational designs, and they are numerous potential third variables (i.e., confounds) that could act as a common cause of both the predictor and outcome variables. Thus, just because two variables are associated does not necessarily mean that they are causally related.

Because no single research design can have both high internal and high external validity, science needs a combination of many different research designs so we can be more confident in our inferences—experimental designs for making causal inferences and observational designs for making inferences that are more likely to reflect the real world.

Case studies, because they have smaller sample sizes, tend to have lower external validity than both experimental and observational studies. They also tend to have more threats to internal validity because they less potential to control for threats to internal validity, such as potential confounds or temporal precedence, so they also tend to have lower internal validity. Nevertheless, case studies can still be useful for generating hypotheses that can then be tested empirically with a larger sample in experimental or observational studies.

4.3.2.3 (Statistical) Conclusion Validity

Conclusion validity, also called statistical conclusion validity, considers the extent to which conclusions about the association among variables based on the data are reasonable. That is, were the correct statistical analyses performed, and are are the interpretations of the findings from those analyses correct?

4.3.3 Putting it Altogether: An Organizational Framework

There are many types of measurement validity, but the central psychometric aspect of measurement validity is construct validity. That is, whether the measure accurately assesses the target construct is the most important consideration of measurement validity. As discussed earlier, construct validity includes the nomological network of the construct. Construct validity also subsumes other key types of measurement validity, including:

- Convergent validity
- Discriminant (divergent) validity
- Criterion-related validity
  - Concurrent validity
  - Predictive validity
- Content validity

Show Figure of Organization of Types of Validity

Moreover, many different types of reliability and validity can be viewed through the lens of construct validity:

- Internal consistency: can be calculated as the coefficient of internal consistency, where the criterion for criterion-related validity is other items on same measure
- Test–retest reliability: can be calculated as the coefficient of stability, where the criterion for criterion-related validity is the same measure at another time point
- Parallel-forms reliability or convergent validity: can be calculated as the coefficient of equivalence, where the criterion for criterion-related validity is the parallel form

4.4 Validity is a Process, Not an Outcome

Validity (and validation) is a continual process, not an outcome. Validation is never complete. When evaluating the validity of a measure, we must ask validity for what and to what degree? We would not just say that a measure is or is not valid. We would express the strength of validity evidence across the different types of validity for a particular purpose, with a particular population, in a particular context (consistent with generalizability theory).

4.5 Reliability versus Validity

Reliability and validity are not the same thing. [**Insert Figure that depicts the difference between reliability and validity with targets; include a separate figure that differentiates accuracy at group-level (no bias) versus accuracy at person-level; Figure 1: a) low reliability, low validity; b) high reliability, low validity; c) high reliability, high validity. Figure 2: a) low reliability, low accuracy at person-level, low accuracy at group-level; b) low reliability, low accuracy at person-level, high accuracy at group-level (assuming representative sample and large enough sample size); c) high reliability, low accuracy at person-level, low accuracy at group-level; d) high reliability, high accuracy at person-level, high accuracy at group-level**] **See figure in Silver 2012 on p. 46.**

Figure 4.13

Figure 4.13: Traditional depiction of reliability (consistency) versus validity (accuracy).

Figure 4.14

Reliability deals with consistency whereas validity deals with accuracy. A measure can be consistent but not accurate (see Figure [insert]). That is, a measure can be reliable but not valid. However, a measure cannot be accurate if it is inconsistent; that is, a measure cannot be valid and unreliable.

Even though reliability and validity are not the same thing, there is a relation between reliability and validity. Validity depends on reliability. Reliability is necessary but insufficient for validity. However, measurement error (unreliability) can be systematic or random. If measurement error is systematic, it reduces the validity of the measure. If measurement error is random, it reduces the precision of an individual's score on a measure, but the measure could still be a valid measure of the construct at the group-level. However, random error would make it more difficult to make an accurate prediction at the individual person-level.

Reliability places the upper bound on validity because a measure can be no more valid than it is reliable. In other words, a measure should not correlate

Figure 4.14: Depiction of reliability versus validity, while distinguishing between validity (accuracy) at person- versus group-level.

more highly with another variable than it correlates with itself. Therefore, the maximum validity coefficient is the square root of the product of the reliability of each measure:

$$r_{xy_{\max}} = \sqrt{r_{xx} r_{yy}}$$
$$\text{maximum association between } x \text{ and } y = \sqrt{\text{reliability of } x \text{ and } y}$$

So, the maximum validity coefficient is based on the reliability of each measure. To the extent that one of the measures is unreliable, the validity coefficient will be attenuated relative to the true validity (i.e., the true strength of association of the constructs), as I describe below.

4.6 Effect of Measurement Error on Associations

Figure 4.15 depicts the CTT approach to understanding the validity of a measure, i.e., its association with another measure, which is the the validity coefficient (r_{xy}).

As described above, (random) measurement error weakens (or attenuates) the association between variables [Goodwin and Leech, 2006]. The greater the random measurement error, the weaker the association. Thus, the correlation between x and y depends on both the true correlation of x and y ($r_{x_t y_t}$) and the reliabilities of x (r_{xx}) and y (r_{yy}). So, measurement error in x and y can reduce the observed correlation below the true correlation. This is known as the *attenuation formula*:

$$r_{xy} = r_{x_t y_t} \sqrt{r_{xx} r_{yy}}$$
$$\text{observed association between } x \text{ and } y = (\text{true association of constructs}) \times \sqrt{\text{reliability of } x \text{ and } y}$$

The lower the reliability, the greater the attenuation of the validity coefficient relative to the true association between the constructs.

All of these r values (excluding the true correlation) are just estimates unless the sample size is infinite—so, the observed association is going to be an imperfect measure. Hence, we need a correction for this attenuation. This correction for the attenuation of an association due to measurement error (unreliability) is known as the disattenuation of a correlation, i.e., correction for the attenuation of an association due to measurement error to get a more accurate estimate of the true association between constructs. Rearranging the terms from the attenuation formula, the formula for disattenuation of a correlation (i.e., the *disattenuation formula*) is as follows:

Figure 4.15: Reliability of a measure across two time points, as depicted in a path diagram.

$$r_{x_t y_t} = \frac{r_{xy}}{\sqrt{r_{xx} r_{yy}}}$$

$$\text{true association of constructs} = \frac{\text{observed association between } x \text{ and } y}{\sqrt{\text{reliability of } x \text{ and } y}}$$

All of this is implied in the path diagram (see Figure 4.15). The attenuation and disattenuation formulas are based on classical test theory, and therefore assume that the all measurement error is random, that errors are uncorrelated, etc. Nevertheless, the attenuation formula can be informative for understanding how imperiled your research is when your measures have low reliability (i.e., when there is instability in the measure). Researchers recommend accounting for measurement reliability, to better estimate the association between constructs, either with the disattenuation formula [Schmidt and Hunter, 1996] or with structural equation modeling, as described in the next chapter.

4.6.1 Test with Simulated Data

4.6.1.1 Reliability of Predictor

```
cor.test(x = mydataValidity$predictorWithMeasurementErrorT1, y = mydataValidity$predictorWithMeas
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydataValidity$predictorWithMeasurementErrorT1 and mydataValidity$predictorWithMeasurem
## t = 63.458, df = 948, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8868366 0.9111563
## sample estimates:
##       cor
## 0.8996923
```

4.6.1.2 Reliability of Criterion

```
cor.test(x = mydataValidity$criterionWithMeasurementErrorT1, y = mydataValidity$criterionWithMeas
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  mydataValidity$criterionWithMeasurementErrorT1 and mydataValidity$criterionW
## t = 49.696, df = 948, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8314187 0.8668075
## sample estimates:
##       cor
## 0.8500698
```

4.6.1.3 True Association

```
cor.test(x = mydataValidity$predictor, y = mydataValidity$criterion)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydataValidity$predictor and mydataValidity$criterion
## t = 29.136, df = 898, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6619034 0.7292278
## sample estimates:
##      cor
## 0.697099
```

4.6.1.4 Observed Association (after adding measurement error)

```
cor.test(x = mydataValidity$predictorWithMeasurementErrorT1, y = mydataValidity$criteri
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydataValidity$predictorWithMeasurementErrorT1 and mydataValidity$criterionW
## t = 23.143, df = 898, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5686020 0.6505992
## sample estimates:
##       cor
## 0.6112382
```

4.6.2 Attenuation of True Correlation due to Measurement Error

The attenuation formula is presented in Equation (4.6). I extend it to a specific example below:

$$r_{xy} = r_{x_t y_t}\sqrt{r_{xx} r_{yy}}$$

$$\text{observed correlation between } x \text{ and } y = (\text{true association between construct } A \text{ and construct } B)\times \sqrt{\text{reliability of } x \times \text{reliability of } y}$$

where x = measure of construct A; y = measure of construct B

Find the observed correlation between the predictor and criterion if the true association between the constructs (i.e., correlation between true scores of constructs) is .7, the reliability of the predictor is .9, and the reliability of the criterion is .85:

```
trueAssociation <- .7
reliabilityOfPredictor <- 0.9
reliabilityOfCriterion <- 0.85

trueAssociation * sqrt(reliabilityOfPredictor * reliabilityOfCriterion)
```

```
## [1] 0.6122499
```

4.6.3 Disattenuation of Observed Correlation due to Measurement Error

The disattenuation formula is presented in Equation (4.6). I extend it to a specific example below:

$$r_{x_t y_t} = \frac{r_{xy}}{\sqrt{r_{xx} r_{yy}}}$$

$$\text{true association between construct } A \text{ and construct } B = \frac{\text{observed correlation between } x \text{ and } y}{\sqrt{\text{reliability of } x \times \text{reliability of } y}}$$

where x = measure of construct A; y = measure of construct B

Find the true association between the construct assessed by the predictor and the construct assessed by the criterion given an observed association if the reliability of the predictor is .9, and the reliability of the criterion is .85:

```
reliabilityOfPredictor <- 0.9
reliabilityOfCriterion <- 0.85

as.numeric(cor.test(x = mydataValidity$predictor, y = mydataValidity$criterion)$estima
```

```
## [1] 0.79701
```

4.7 Generalizability Theory

Generalizability theory is discussed in greater detail in the chapter on reliability in Section 3.11. As a brief reminder, G-theory is a measurement theory that, unlike CTT, does not treat all measurement differences across time, rater, or situation as "error" but rather as a phenomenon of interest. G-theory can simultaneously consider multiple aspects of reliability and validity in the same model. For instance, internal consistency reliability, test–retest reliability, inter-rater reliability, parallel-forms reliability, and convergent validity [in the Campbell and Fiske, 1959, sense of the same construct assessed by a different method] can all be incorporated into a G-theory model.

For example, a G-theory model could assess each participant across the following facets:

- time: e.g., T1 and T2 (test–retest reliability)
- items: e.g., questions within the same instrument (internal consistency reliability) and questions across different instruments (parallel-forms reliability)
- rater: e.g., self-report and other-report (inter-rater reliability)
- method: e.g., questionnaire and observation (convergent validity)

[**Provide Example Data Matrix**]

Using such a G-theory model, we can determine the extent to which scores on a measure generalize to other conditions, measures, etc. In sum, G-theory can be a useful way of estimating the degree of reliability and validity of a measure's scores in the same model.

4.8 Ways to Increase Validity

Here are potential ways to increase the validity of the interpretation of a measure's scores for a particular purpose:

- Make sure the measure's scores are reliable. For potential ways to increase the reliability of measurement, see Section 3.14. But, do not switch to a less valid measure or to items that are less valid merely because they are more reliable.
- Design the measure with a particular population and purpose in mind. When describing the measure in papers or in public spheres, make it clear to others what the population and intended purposes are, and what they are not.
- Make sure each item's scores are valid, based on theory and/or empiricism, for the particular population and purpose. For instance, the items should show content validity—the items should assess facets of the target construct for the target population as defined by experts, without item intrusions from other constructs. The items' scores should show convergent validity—the items' scores should be related to other measures of the construct, within the population of interest. The items' scores should show discriminant validity—the items' scores be more strongly related to measures that are intended to assess the same construct than they are to measures that are intended to assess other constructs.
- Obtain samples that are as representative of the population as possible, paying attention to including people who are traditionally underrepresented in research (if such groups are part of the target population).
- Make sure that people in the population can understand, interpret, and respond to each item in a meaningful and comparable way.
- Make sure the measure and its items are not biased against any subgroup within the population of interest. Test bias is discussed in Chapter 14.
- Be careful to administer the measure to the population of interest under the conditions in which it is designed. If the measure must be administered to people from a different population or under different conditions from which it was designed, be careful to (a) note that the measure was not designed to be administered for these other populations or conditions, and (b) note that individuals' scores may not accurately reflect their level on the construct. If interpretations are made based on these scores, make them cautiously and say how the differences in population or condition may have influenced the scores.
- Continue to monitor the validity of the measure's scores for the given population and purpose. The validity of measures' scores can change over time for a number of reasons. Cohort effects can lead items to become obsolete over time. If people or organizations change their behavior in response to a measure, this can invalidate a measure's scores for the intended purpose, as described in Section 4.3.1.9 when discussing consequential validity.

4.9 Conclusion

Validity is how much accuracy, utility, and meaningfulness the interpretation of a measure's scores have for a particular purpose. Like reliability, validity is not one thing. There are multiple aspects of validity. Validity is also not a characteristic that resides in a test. The validity of a measure's scores reflect an interaction of the properties of the test with the population for whom it is designed, and the sample and context in which it is administered. Thus, when reporting validity in papers, it is important to adequately describe the aspects of validity that have been considered, and the population, sample, and context in which the measure is assessed.

4.10 Exercises

4.10.1 Questions

Note: several of the following questions use data from the Children of the National Longitudinal Survey of Youth Survey (CNLSY). The CNLSY is a publicly available longitudinal dataset provided by the Bureau of Labor Statistics (https://www.bls.gov/nls/nlsy79-children.htm#topical-guide). The CNLSY data file for these exercises is located on the course's page of the Open Science Framework (https://osf.io/3pwza). Children's behavior problems were rated in 1988 (time 1: T1) and then again in 1990 (time 2: T2) on the Behavior Problems Index (BPI).

1. What is the criterion-related validity of the Antisocial Behavior subscale of the BPI in relation to the Hyperactive subscale of the BPI?
2. Assume the true correlation between two constructs, antisocial behavior and hyperactivity, is $r = .8$. And assume the reliability of the measure of antisocial behavior is $r = .7$, and the reliability of the measure of hyperactivity is $r = .7090234$. According to the attenuation formula (that attenuates the association due to measurement error), what would be the correlation between measures of antisocial behavior and hyperactivity that we would actually observe?
3. Assume the true correlation between two constructs, antisocial behavior and hyperactivity, is $r = .8$, the reliability of the Antisocial Behavior subscale of the BPI is $r = .7$, and the reliability of the measure of Hyperactive subscale of the BPI is $r = .8$. According to the disattenuation formula (to correct for attenuation of the association due to measurement error), what would be the true association between the constructs of antisocial behavior and hyperactivity?
4. You are interested in whether a child's levels of anxiety/depression can explain unique variance in children's hyperactivity above and beyond their

level of antisocial behavior. Is the child's level of anxiety/depression (as rated on the Anxiety/Depression subscale of the BPI) significantly associated with the child's level of hyperactivity (as rated on the Hyperactive subscale of the BPI) above and beyond the variance accounted for by the child's level of antisocial behavior (as rated on the Antisocial Behavior subscale of the BPI). How much unique variance in hyperactivity is accounted for by their anxiety/depression?

5. The course's OSF page has a data file titled `MTMM.csv` that includes data on participants' verbal, spatial, and quantitative abilities, each assessed in three subtests in each of three methods: oral, written, and manipulative.

   a. Provide a multitrait-multimethod matrix of the data from the first subtest of each trait-by-method combination (`VO1`, `SO1`, `QO1`, `VW1`, `SW1`, `QW1`, `VM1`, `SM1`, `QM1`). Assume the reliability of the variables assessed orally is .7, the reliability of the variables assessed using the written method is .8, and the reliability of variables assessed using the manipulative method is .9.
   b. Interpret the multitrait-multimethod matrix you just created.
   c. What is the Heterotrait-Monotrait (HTMT) ratio for the measures of the verbal (`VO1`, `VW1`, `VM1`) and spatial (`SO1`, `SW1`, `SM1`) constructs? What does this indicate?

4.10.2 Answers

1. The criterion-related validity is $r = .56$.

2. The observed correlation would be $r = .56$.

3. The true association would be $r = .75$.

4. Yes, the child's level of anxiety/depression is significantly associated with their hyperactivity above and beyond their antisocial behavior ($F[df = 1] = 217.25, p < .001$). The child's level of anxiety/depression accounted for 6.02% unique variance in hyperactivity above and beyond their antisocial behavior.

5. a. The multitrait-multimethod matrix is below:

5. b. The convergent correlations (green cells) are statistically significant (ps $< .05$) and moderate in magnitude ($.45 < r$s $< .53$), supporting the convergent validity of the measures. [The reliabilities of these measures are not provided, so we are not able to compare the magnitude of the convergent validities to the magnitude of reliability to see the extent to which the convergent validities may be attenuated due to measurement unreliability]. Evidence of discriminant validity is supported by three findings: First, the convergent correlations (green

Method	Trait	Variable	Oral			Written			Manipulative		
			Verbal	Spatial	Quant.	Verbal	Spatial	Quant.	Verbal	Spatial	Quant.
			VO1	SO1	QO1	VW1	SW1	QW1	VM1	SM1	QM1
Oral	Verbal	VO1	.70								
	Spatial	SO1	.32	.70							
	Quantitative	QO1	.43	.34	.70						
Written	Verbal	VW1	.48	.21	.34	.80					
	Spatial	SW1	.30	.48	.34	.35	.80				
	Quantitative	QW1	.31	.25	.46	.43	.37	.80			
Manip.	Verbal	VM1	.50	.25	.34	.48	.29	.33	.90		
	Spatial	SM1	.24	.47	.30	.26	.49	.32	.30	.90	
	Quantitative	QM1	.34	.30	.52	.33	.30	.51	.42	.37	.90

Key
- convergent validity diagonals (monotrait heteromethod)
- reliability diagonal (monotrait monomethod)
- heterotrait heteromethod
- heterotrait monomethod

Figure 4.16: Multi-trait multi-method matrix

cells: rs = .46–.53) are stronger than the heteromethod-heterotrait correlations (pink cells: rs = .24–.34). Second, the convergent correlations (green cells: rs = .46–.53) are stronger than the discriminant correlations (orange cells: rs = .30–.43). Third, the patterns of intercorrelations between traits are the same, regardless of which measurement method is used. Verbal, spatial, and quantitative skills are inter-correlated for every measurement method used.

c. The HTMT ratio for the measures of the verbal and spatial constructs is 0.578. The HTMT ratio is the average of the heterotrait-heteromethod correlations, relative to the average of the monotrait-heteromethod correlations. Given that the HTMT ratio is considerably less than 1 (and less than the common cutoff of .85), it indicates that the monotrait-heteromethod correlations are considerably larger than the heterotrait-heteromethod correlations. Thus, the HTMT provides evidence that the measures of verbal and spatial constructs show discriminant validity.

Chapter 5

Structural Equation Modeling

> "All models are wrong, but some are useful."
> — George Box [1979, p. 202]

5.1 Overview of SEM

Structural equation modeling is an advanced modeling approach that allows estimating latent variables to account for measurement error and to get purer estimates of constructs.

5.2 Getting Started

5.2.1 Load Libraries

```
library("lavaan")
library("semTools")
library("semPlot")
library("simsem")
library("snow")
library("mice")
library("tidyverse")
library("here")
library("tinytex")
```

5.2.2 Prepare Data

5.2.2.1 Simulate Data

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
sampleSize <- 300

set.seed(52242)

measure1 <- rnorm(n = sampleSize, mean = 50, sd = 10)
measure2 <- measure1 + rnorm(n = sampleSize, mean = 0, sd = 15)
measure3 <- measure1 + measure2 + rnorm(n = sampleSize, mean = 0, sd = 15)
```

5.2.2.2 Add Missing Data

Adding missing data to dataframes helps make examples more realistic to real-life data, and helps you get in the habit of programming to account for missing data.

```
measure1[c(5,10)] <- NA
measure2[c(10,15)] <- NA
measure3[c(10)] <- NA
PoliticalDemocracy <- as.data.frame(lapply(PoliticalDemocracy, function(cc) cc[ sample
```

5.2.2.3 Combine data into data frame

```
mydataSEM <- data.frame(measure1, measure2, measure3)
```

5.3 Types of Models

5.3.1 Path Analysis Model

To understand structural equation modeling (SEM), let us first discuss path analysis. Path analysis is similar to multiple regression. It allows examining the association between multiple predictor variables (or independent variables) in relation to an outcome variable (or dependent variable). Unlike multiple regression, however, path analysis also allows inclusion of multiple dependent variables in the same model. SEM is path analysis, but with latent (unobserved) variables. By contrast, path analysis uses only manifest (observed) variables, not latent variables.

5.3.2 Measurement Model

The measurement model is a crucial sub-component of any SEM model. A SEM model consists of two components: a measurement model and a structural model. The structural component of a SEM model includes the regression paths that specify the hypothesized causal relations among the latent variables. The measurement model is a confirmatory factor analysis (CFA) model that identifies how many latent factors are estimated, and which items load onto which latent factor. The measurement model can also specify correlated residuals. Basically, the measurement model specifies your best understanding of the structure of the latent construct(s) given how they were assessed. Before fitting the structural component of a SEM, it is important to have a well-fitting measurement model for each construct in the model. In Section 5.3.2, I present an example of a measurement model.

5.3.3 Confirmatory Factor Analysis Model

Confirmatory Factor Analysis (CFA) is a subset of SEM. CFA includes the measurement model but not the structural component of the model. In Section 5.9, I present an example of a CFA model. I discuss CFA models in greater depth in Section 12.5 of the chapter on Objective Personality Testing because they are often used for developing questionnaire measures.

5.3.4 Structural Equation Model

SEM is CFA, but it adds regression paths that specify hypothesized causal relations between the latent variables, which is called the structural component of the model. The structural model includes the hypothesized causal relations between latent variables. A SEM model includes both the measurement model and the structural model [see Figure 5.1, Civelek, 2018]. SEM fits a model to observed data, or the variance-covariance matrix, and evaluates the degree of model misfit. That is, fit indices evaluate how likely it is that a given model gave rise to the observed data. In Section 5.10, I present an example of a SEM model.

SEM is flexible in allowing you to specify measurement error and correlated errors. Thus, you do not need the same assumptions as in classical test theory, which assumes that errors are random and uncorrelated. But, the flexibility of SEM also poses challenges because you must explicitly decide what to include—and not include—in your model. If the model fit is unacceptable, you can try fitting a different model to see which fits better. Nevertheless, it is important to use theory as a guide when specifying and comparing competing models, and not just rely solely on model fit comparison. For example, the model you fit should depend on how you conceptualize each construct: as reflective or formative.

Figure 5.1: Demarcation between measurement model and structural model. Figure reprinted from @Civelek2018, Figure 1, p. 7. Civelek, M. E. (2018). *Essentials of structural equation modeling*. Zea E-Books. https://doi.org/10.13014/K2SJ1HR5

5.4 Types of Constructs

A *construct* is a concept. A construct is generally considered a latent, unobservable idea or phenomenon. For example, depression could be considered a construct because we cannot directly measure someone's level of depression. Rather, we infer a person's level on the construct of depression by taking measurements of many indirect proxies that we think are *indicators* of the construct. An indicator is a measurement, and it could be thought of as a behavior or questionnaire item. Potential indicators of depression could include behaviors/items such as whether the person has low mood, low energy, sleep difficulties, loss of interest in formerly enjoyable activities, changes in weight or appetite, etc.

There are two main types of constructs: reflective constructs and formative constructs.

5.4.1 Reflective Construct

With a *reflective construct* (or reflective model), the construct is the cause of the measures, and the construct is reflected by the indicators, see Figure 5.2 [Bollen and Lennox, 1991].

Figure 5.2: Reflective and formative constructs in structural equation modeling.

An example of reflective construct is extraversion, see Figure 5.3. We conceptualize extraversion as a latent concept that, although not directly observable, influences the measurements of indicators of extraversion, such as whether the person enjoys talking to strangers, goes to lots of parties, is energetic, and is happy. In a reflective model, the answers that answers people give on the items are thought to reflect a certain disposition, in this case, extraversion. None of these indicators, in isolation, is considered a direct measure of extraversion. Nevertheless, we feel that we can get better estimate of the concept of extraversion by capturing the variance that covaries across these indicators, i.e., the common (systematic) variance among the indicators. Thus, the latent factor is estimated to reflect the common variance among the indicators. Because a reflective construct influences the indicators, the indicators are called *effect indicators*. What we observe, theoretically, are the effects of the underlying construct, which influences score on each individual measure.

With a reflective construct, we would expect that the items would show strong internal consistency reliability. That is, we would expect that the indicators would all be correlated with each other, because all indicators are thought to be a reflection of the underlying construct. But in reality, there is often still residual correlation between items even after accounting for the latent factor.

In a measurement model of a reflective construct (see Figure 5.2), the lambda (λ) values are called *factor loadings* that represent regression coefficients from observable indicators to latent variables. A factor loading is the relative weight that the item is given in the estimation of the latent factor, similar to the correlation between the construct and the measure. The stronger an item's factor

Figure 5.3: Extraversion as a reflective construct.

loading on the construct, the more relative weight a given item has in the estimation of the latent factor. The epsilon (ϵ) term reflects measurement error. SEM is an application of the same way we have been thinking about measurement according to classical test theory, but in a latent variable framework. As a reminder, according to classical test theory, the observed score is thought to be some combination of true score and error (observed score = true score + error; $X = T + e$). Or, in a multiple regression framework: extraversion = $\lambda_1 \cdot$ strangers + $\lambda_2 \cdot$ parties + $\lambda_3 \cdot$ happy + errors. That is, the true score influences the observed score on each measure. If a person's level on the construct (extraversion) changes, their score on each measure/indicator/item is expected to change [Edwards and Bagozzi, 2000].

5.4.2 Formative Construct

Formative constructs are the other fundamental way to understand the relation between a construct and measure. With a *formative construct* (or formative model), the measures cause the construct, see Figure 5.2 [Bollen and Lennox, 1991]. That is, the construct is created by the measures. Because the indicators influence the formative construct, the indicators are called *causal indicators*.

Examples of a formative constructs include constructs such as socioeconomic status (SES), stress, and risk for cardiovascular disease. For instance, we may define SES such that it is the linear combination of a person's educational attainment, occupational prestige, and income, as in Figure 5.4. With a formative construct, if a person changes in their level on the construct, we might not expect that their scores on all of the measures/indicators/items would change [Edwards and Bagozzi, 2000]. If a person's SES changes, you would not necessarily expect the person's score on all of these measures/items to change. For instance, if a person's SES increases from T1 to T2, you wouldn't necessarily expect that their educational attainment increased; it could be instead that their occupational prestige or income increased.

In a structural equation model, the gamma (γ) values are regression coefficients from exogenous variables to endogenous variables. Endogenous variables are variables whose vales are determined by the model, whereas exogenous variables are variables whose values are not determined by the model. In a measurement model of a formative construct (see Figure 5.2), the gamma values reflect the relative weight that the item is given in the estimation of the latent factor, similar to the correlation between the construct and the measure. The stronger an item's factor loading on the construct, the more relative weight a given item has in the estimation of the latent factor.

With a formative construct, the disturbance term, zeta (ζ), represents the part of the construct that is not explained by the measures, i.e., the measurement error. However, the measures are thought (or at least assumed) to be error-free causes of the construct. The covariances between the indicators are permissive

Figure 5.4: Socioeconomic status as a formative construct.

of item correlations, but unlike the reflective model, the formative model does not require that the items are correlated. In other words, for a reflective model, it is not necessary that the items are correlated, even though the items could be correlated.

A formative construct is determined by the measures, so if you change the measures, you change the construct. A formative model requires that you include all facets of the construct. This is not true in a reflective model because all items of a unidimensional reflective construct are interchangeable as long as they are equally reliable because the construct influences the measure scores. With a reflective construct, all you need to do is find the study that reflects the underlying construct, and you could create parallel measures of the construct. This is impossible to do in a formative model.

Althought formative conceptualizations of constructs may be useful in some cases, formative constructs can be controversial [Edwards, 2011; Hardin et al., 2011; Howell et al., 2007; Markus, 2018; but see Bollen and Lennox, 1991; Bollen and Bauldry, 2011; Bollen and Diamantopoulos, 2017; Diamantopoulos et al., 2008]. It is important to think carefully about the construct of interest, as guided by theory, and how best to assess and estimate it.

5.4.2.1 Differences in measurement expectations

There are several differences in our measurement expectations based on the type of construct (formative or reflective) or indicator (effect or causal), as described by Bollen and Lennox [1991].

One difference we would expect between effect versus causal indicators is the actual correlations between the indicators. Causal indicators may not be correlated whereas effect indicators must be correlated. So, when we are assessing internal consistency reliability of a measure, and assume that higher correlations between items suggests higher internal consistency, we are assuming that indicators of the construct have an effects relationship. A scale assessing causal indicators would likely have very low internal consistency.

A second difference we would expect between effect versus causal indicators is whether to use a sample or a census of indicators. The suggestions that we sample all facets of a construct when determining indicators is necessary only for causal indicators, for which a failure to assess a certain formative indicator changes the overall meaning of the construct. For effect indicators of a unidimensional construct, equally reliable measures are interchangeable.

A third difference we would expect between effect versus causal indicators is the optimal correlations between the indicators. For effect indicators, high correlations are desirable. For causal indicators, too high of a correlation between indicators is likely to introduce the problem of multi-collinearity, so lower correlations are optimal. Multi-collinearity is when two or more predictors are

correlated such that their regression coefficients with the outcome variable (in this case, the factor loadings on the latent factor, i.e., λ) are erratic.

5.5 Practical issues

There are important practical issues to consider with both reflective and formative models. An important practical issue is model identification—adding enough constraints so that there is only one, best answer. The model is identified when each of the estimated parameters has a unique solution. Based on the number of data points compared to the number of estimated parameters, a model can be considered either just identified, under-identified, or over-identified. The number of data points in a SEM model is the number of variances and covariances in the variance/covariance matrix in addition to the number of means, which can be calculated as: $\frac{m(m+1)}{2} + m$, where $m =$ the number of indicators. A *just identified model* is a model in which the number of data points is equal to the number of parameters to be estimated (degrees of freedom $= 0$). An *under-identified model* is a model in which the number of data points is less than the number of parameters to be estimated (degrees of freedom < 0). An *over-identified model* is a model in which the number of number of data points is greater than the number of parameters to be estimated (degrees of freedom > 0).

As an example, there are 14 data points for a model with 4 indicators ($\frac{4(4+1)}{2} + 4 = 14$): 4 variances, 6 covariances, and 4 means.

Here is the variance-covariance matrix:

```
vcovMatrix4measures <- cov(PoliticalDemocracy[,c("y1","y2","y3","y4")], use = "pairwise

vcovMatrix4measures[upper.tri(vcovMatrix4measures)] <- NA

vcovMatrix4measures
```

```
##          y1        y2        y3       y4
## y1 7.129306        NA        NA       NA
## y2 6.602199 15.514625        NA       NA
## y3 6.188934  5.710225 11.183450       NA
## y4 5.375537  9.719346  6.557986 10.95935
```

Here are the variances:

```
variances4measures <- diag(vcovMatrix4measures)

variances4measures
```

```
##        y1        y2        y3        y4
##  7.129306 15.514625 11.183450 10.959347
```

Here are the covariances:

```
covariances4measures <- vcovMatrix4measures[lower.tri(vcovMatrix4measures)]

covariances4measures
```

```
## [1] 6.602199 6.188934 5.375537 5.710225 9.719346 6.557986
```

Here are the means:

```
means4Measures <- apply(PoliticalDemocracy[,c("y1","y2","y3","y4")], 2, mean, na.rm = TRUE)

means4Measures
```

```
##       y1       y2       y3       y4
## 5.479545 4.259046 6.529901 4.236372
```

For a reflective model with 4 indicators, we would need to estimate 12 parameters: a factor loading, error term, and intercept for each of the 4 indicators. Here are the parameters estimated:

```
reflectiveModel_syntax <- '
 #Reflective model factor loadings
 dem60 =~ y1 + y2 + y3 + y4
'

reflectiveModelFit <- sem(reflectiveModel_syntax,
                          data = PoliticalDemocracy,
                          missing = "ML",
                          estimator = "MLR",
                          std.lv = TRUE)

reflectiveModelParameters <- parameterEstimates(reflectiveModelFit)[!is.na(parameterEstimates(ref

row.names(reflectiveModelParameters) <- NULL

reflectiveModelParameters
```

```
##      lhs op rhs   est    se      z pvalue ci.lower ci.upper
## 1  dem60 =~  y1 2.233 0.245  9.111  0.000    1.753    2.714
```

```
## 2  dem60 =~  y2 3.034 0.291 10.422  0.000    2.463    3.605
## 3  dem60 =~  y3 2.437 0.361  6.749  0.000    1.730    3.145
## 4  dem60 =~  y4 2.933 0.261 11.222  0.000    2.421    3.445
## 5     y1 ~~  y1 2.281 0.646  3.531  0.000    1.015    3.547
## 6     y2 ~~  y2 5.903 1.410  4.187  0.000    3.139    8.666
## 7     y3 ~~  y3 5.275 1.315  4.012  0.000    2.698    7.851
## 8     y4 ~~  y4 2.483 0.798  3.111  0.002    0.919    4.048
## 9     y1 ~1     5.562 0.319 17.451  0.000    4.937    6.187
## 10    y2 ~1     4.325 0.459  9.420  0.000    3.425    5.225
## 11    y3 ~1     6.613 0.393 16.816  0.000    5.842    7.384
## 12    y4 ~1     4.407 0.396 11.142  0.000    3.632    5.182
```

Here are the degrees of freedom:

```
fitMeasures(reflectiveModelFit, "df")
```

```
## df
##  2
```

Here is a model diagram:

```
semPaths(reflectiveModelFit,
         what = "Std.all",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 0.8)
```

Thus, for a reflective model, We only have to estimate a small number of parameters to specify what is happening in our model, so the model is parsimonious. With 4 indicators, the number of data points (14) is greater than the number of parameters (12). We have two degrees of freedom ($14 - 12 = 2$). Because the degrees of freedom is greater than zero, it is easy to identify the model—the model is over-identified. A reflective model with 3 indicators would have 9 data points ($\frac{3(3+1)}{2} + 3 = 9$), 9 parameters (3 factor loadings, 3 error terms, 3 intercepts), and 0 degrees of freedom, and it would be identifiable because it would be just-identified.

However, for a formative model, we must specify more parameters: a factor loading, intercept, and variance for each of the 4 indicators, all 6 permissive correlations, and 1 error term for the latent variable, for a total of 19 parameters.

```
formativeModel_syntax <- '
 #Formative model factor loadings
 dem60 <~ y1 + y2 + y3 + y4

 dem60 ~~ dem60
'

formativeModelFit <- sem(formativeModel_syntax,
                         data = PoliticalDemocracy,
                         missing = "ML",
                         estimator = "MLR")
```

```
formativeModelParameters <- parameterEstimates(formativeModelFit)

formativeModelParameters
```

```
##      lhs op   rhs    est se  z pvalue ci.lower ci.upper
## 1  dem60 <~    y1  0.000 NA NA     NA       NA       NA
## 2  dem60 <~    y2  0.000 NA NA     NA       NA       NA
## 3  dem60 <~    y3  0.000 NA NA     NA       NA       NA
## 4  dem60 <~    y4  0.000 NA NA     NA       NA       NA
## 5  dem60 ~~ dem60  0.050 NA NA     NA       NA       NA
## 6     y1 ~~    y1  6.884  0 NA     NA    6.884    6.884
## 7     y1 ~~    y2  6.011  0 NA     NA    6.011    6.011
## 8     y1 ~~    y3  5.877  0 NA     NA    5.877    5.877
## 9     y1 ~~    y4  5.274  0 NA     NA    5.274    5.274
## 10    y2 ~~    y2 15.176  0 NA     NA   15.176   15.176
## 11    y2 ~~    y3  4.746  0 NA     NA    4.746    4.746
## 12    y2 ~~    y4  8.626  0 NA     NA    8.626    8.626
## 13    y3 ~~    y3 11.368  0 NA     NA   11.368   11.368
## 14    y3 ~~    y4  5.984  0 NA     NA    5.984    5.984
## 15    y4 ~~    y4 10.341  0 NA     NA   10.341   10.341
## 16    y1 ~1        5.163  0 NA     NA    5.163    5.163
## 17    y2 ~1        3.849  0 NA     NA    3.849    3.849
## 18    y3 ~1        6.445  0 NA     NA    6.445    6.445
## 19    y4 ~1        4.042  0 NA     NA    4.042    4.042
## 20 dem60 ~1        0.000  0 NA     NA    0.000    0.000
```

Here are the degrees of freedom:

```
formativeModelFit
```

```
## lavaan 0.6-8 ended normally after 1 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         5
##
##                                                   Used       Total
##   Number of observations                            51          75
##   Number of missing patterns                         1
##
## Model Test User Model:
##
##   Test statistic                                    NA
##   Degrees of freedom                                -5
##   P-value (Unknown)                                 NA
```

Here is a model diagram:

```
semPaths(formativeModelFit,
         what = "Std.all",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 0.8)
```

For a formative model with 4 measures, the number of data points (14) is less than the number of parameters (19). The number of degrees of freedom is negative ($14 - 19 = -5$), thus the model is not able to be identified—the model is under-identified. Here are the parameters estimated:

Thus, for a formative model, we need more parameters than we have data—the model is under-identified. Therefore, to estimate a formative model with 4 indicators, we must add assumptions and other variables that are consequences of the formative construct. Options for identifying a formative construct are described by Treiblmaier et al. [2011]. Formative models are challenging to use, and must be used in the context of a model that allows some constraints. But, one can easily estimate formative constructs outside of a SEM context, for example, with an average or weighted average composite, similar to principal component analysis. However, linear composite approaches (e.g., mean or sum scores) do not account for measurement error. That is, they do not include a disturbance/error term. Therefore, they are not the same as a latent variable.

If no disturbance term is included, the model is equivalent to a linear composite approach. Linear composite approaches are likely more defensible for formative constructs than reflective constructs.

5.6 Additional Types of SEM

Up to this point, we have discussed SEM with continuous constructs. It also worth knowing about additional types of SEM models, including latent class models and mixture models, that handle categorical constructs.

5.6.1 Latent Class Models

In *latent class models*, the construct is not continuous, but rather categorical. The categorical constructs are latent classifications and are called latent classes. For instance, the construct could be a diagnosis that influences scores on the measures. Latent class models examine qualitative differences in kind, rather than quantitative differences in degree.

5.6.2 Mixture Models

Mixture models allow for a combination of latent categorical constructs (classes) and latent continuous constructs. That is, it allows for both qualitative and quantitative differences. However, this additional model complexity also necessitates a larger sample size for estimation. SEM generally requires a 3-digit sample size ($N = 100+$), whereas mixture models typically require a 4- or 5-digit sample size ($N = 1,000+$).

5.7 Correlation Matrix

```
cor(mydataSEM, use = "pairwise.complete.obs")
```

```
##           measure1  measure2  measure3
## measure1 1.0000000 0.4852939 0.6877718
## measure2 0.4852939 1.0000000 0.7900745
## measure3 0.6877718 0.7900745 1.0000000
```

5.8 Measurement Model

Even though measurement models are CFA models, I provide separate examples of a measurement model and CFA models in my examples because CFA is often used to test competing factor structures. For instance, you could use CFA to test whether the variance in several measures' scores is best explained with one factor or two factors. In the measurement model below, I present a simple one-factor model with three measures. The measurement model is what we settle on as the estimation of each construct before we add the structural component to estimate the relations among latent variables. Basically, we add the structural component onto the measurement model. In Section 5.3.3, I present a CFA model with multiple latent factors.

The measurement models were fit in the `lavaan` package [Rosseel et al., 2020].

5.8.1 Specify the model

```
measurementModel_syntax <- '
 #Factor loadings
 latentFactor =~ measure1 + measure2 + measure3
'

measurementModel_fullSyntax <- '
 #Factor loadings (free the factor loading of the first indicator)
 latentFactor =~ NA*measure1 + measure2 + measure3

 #Fix latent mean to zero
 latentFactor ~ 0

 #Fix latent variance to one
 latentFactor ~~ 1*latentFactor

 #Estimate covariances among latent variables (not applicable because there is only one latent va

 #Estimate residual variances of manifest variables
 measure1 ~~ measure1
 measure2 ~~ measure2
 measure3 ~~ measure3

 #Free intercepts of manifest variables
 measure1 ~ int1*1
 measure2 ~ int2*1
 measure3 ~ int3*1
'
```

5.8.1.1 Summary of Model Features

```
summary(measurementModel_syntax)
```

```
##    Length     Class      Mode
##         1 character character
```

```
summary(measurementModel_fullSyntax)
```

```
##    Length     Class      Mode
##         1 character character
```

5.8.1.2 Model Syntax in Table Form:

```
lavaanify(measurementModel_syntax)
```

```
##   id          lhs op          rhs user block group free ustart exo label plabel
## 1  1 latentFactor =~     measure1    1     1     1    1     NA   0         .p1.
## 2  2 latentFactor =~     measure2    1     1     1    2     NA   0         .p2.
## 3  3 latentFactor =~     measure3    1     1     1    3     NA   0         .p3.
## 4  4     measure1 ~~     measure1    0     1     1    0      0   0         .p4.
## 5  5     measure2 ~~     measure2    0     1     1    0      0   0         .p5.
## 6  6     measure3 ~~     measure3    0     1     1    0      0   0         .p6.
## 7  7 latentFactor ~~ latentFactor    0     1     1    0      0   0         .p7.
```

```
lavaanify(measurementModel_fullSyntax)
```

```
##    id          lhs op          rhs user block group free ustart exo label
## 1   1 latentFactor =~     measure1    1     1     1    1     NA   0
## 2   2 latentFactor =~     measure2    1     1     1    2     NA   0
## 3   3 latentFactor =~     measure3    1     1     1    3     NA   0
## 4   4 latentFactor ~1                 1     1     1    0      0   0
## 5   5 latentFactor ~~ latentFactor    1     1     1    0      1   0
## 6   6     measure1 ~~     measure1    1     1     1    4     NA   0
## 7   7     measure2 ~~     measure2    1     1     1    5     NA   0
## 8   8     measure3 ~~     measure3    1     1     1    6     NA   0
## 9   9     measure1 ~1                 1     1     1    7     NA   0  int1
## 10 10     measure2 ~1                 1     1     1    8     NA   0  int2
## 11 11     measure3 ~1                 1     1     1    9     NA   0  int3
##    plabel
```

```
## 1    .p1.
## 2    .p2.
## 3    .p3.
## 4    .p4.
## 5    .p5.
## 6    .p6.
## 7    .p7.
## 8    .p8.
## 9    .p9.
## 10  .p10.
## 11  .p11.
```

5.8.2 Fit the model

```
measurementModelFit <- cfa(measurementModel_syntax,
                           data = mydataSEM,
                           missing = "ML",
                           estimator = "MLR",
                           std.lv = TRUE)

measurementModelFit_full <- lavaan(measurementModel_fullSyntax,
                                   data = mydataSEM,
                                   missing = "ML",
                                   estimator = "MLR")
```

5.8.3 Display summary output

```
summary(measurementModelFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 26 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         9
##
##                                                   Used       Total
##   Number of observations                           299         300
##   Number of missing patterns                         3
```

```
##
## Model Test User Model:
##                                              Standard      Robust
##   Test Statistic                                0.000       0.000
##   Degrees of freedom                                0           0
##
## Model Test Baseline Model:
##
##   Test statistic                               487.311     470.630
##   Degrees of freedom                                 3           3
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.035
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000       1.000
##   Tucker-Lewis Index (TLI)                       1.000       1.000
##
##   Robust Comparative Fit Index (CFI)                         1.000
##   Robust Tucker-Lewis Index (TLI)                            1.000
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3544.291   -3544.291
##   Loglikelihood unrestricted model (H1)      -3544.291   -3544.291
##
##   Akaike (AIC)                                7106.582    7106.582
##   Bayesian (BIC)                              7139.886    7139.886
##   Sample-size adjusted Bayesian (BIC)         7111.344    7111.344
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000       0.000
##   90 Percent confidence interval - lower         0.000       0.000
##   90 Percent confidence interval - upper         0.000       0.000
##   P-value RMSEA <= 0.05                             NA          NA
##
##   Robust RMSEA                                               0.000
##   90 Percent confidence interval - lower                     0.000
##   90 Percent confidence interval - upper                     0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000       0.000
##
## Parameter Estimates:
```

```
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   latentFactor =~
##     measure1          5.906    0.508   11.620    0.000    5.906    0.652
##     measure2         13.535    0.940   14.400    0.000   13.535    0.748
##     measure3         29.796    1.308   22.778    0.000   29.796    1.056
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .measure1         49.245    0.525   93.882    0.000   49.245    5.439
##    .measure2         49.570    1.047   47.331    0.000   49.570    2.739
##    .measure3         99.139    1.632   60.747    0.000   99.139    3.513
##     latentFactor      0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .measure1         47.097    4.673   10.079    0.000   47.097    0.574
##    .measure2        144.287   14.791    9.755    0.000  144.287    0.441
##    .measure3        -91.453   48.220   -1.897    0.058  -91.453   -0.115
##     latentFactor      1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     measure1          0.426
##     measure2          0.559
##     measure3             NA
```

```
summary(measurementModelFit_full,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 26 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         9
##
##                                                   Used       Total
##   Number of observations                           299         300
```

```
##   Number of missing patterns                        3
##
## Model Test User Model:
##                                              Standard      Robust
##   Test Statistic                                0.000       0.000
##   Degrees of freedom                                0           0
##
## Model Test Baseline Model:
##
##   Test statistic                              487.311     470.630
##   Degrees of freedom                                3           3
##   P-value                                       0.000       0.000
##   Scaling correction factor                                 1.035
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                   1.000       1.000
##   Tucker-Lewis Index (TLI)                      1.000       1.000
##
##   Robust Comparative Fit Index (CFI)                        1.000
##   Robust Tucker-Lewis Index (TLI)                           1.000
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -3544.291   -3544.291
##   Loglikelihood unrestricted model (H1)     -3544.291   -3544.291
##
##   Akaike (AIC)                               7106.582    7106.582
##   Bayesian (BIC)                             7139.886    7139.886
##   Sample-size adjusted Bayesian (BIC)        7111.344    7111.344
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.000       0.000
##   90 Percent confidence interval - lower        0.000       0.000
##   90 Percent confidence interval - upper        0.000       0.000
##   P-value RMSEA <= 0.05                            NA          NA
##
##   Robust RMSEA                                              0.000
##   90 Percent confidence interval - lower                    0.000
##   90 Percent confidence interval - upper                    0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.000       0.000
##
```

```
## Parameter Estimates:
##
##   Standard errors                              Sandwich
##   Information bread                            Observed
##   Observed information based on                 Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   latentFactor =~
##     measure1          5.906    0.508   11.620    0.000    5.906    0.652
##     measure2         13.535    0.940   14.400    0.000   13.535    0.748
##     measure3         29.796    1.308   22.778    0.000   29.796    1.056
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     ltntFct           0.000                               0.000    0.000
##    .measur1 (int1)   49.245    0.525   93.882    0.000   49.245    5.439
##    .measur2 (int2)   49.570    1.047   47.331    0.000   49.570    2.739
##    .measur3 (int3)   99.139    1.632   60.747    0.000   99.139    3.513
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     latentFactor      1.000                               1.000    1.000
##    .measure1         47.097    4.673   10.079    0.000   47.097    0.574
##    .measure2        144.287   14.791    9.755    0.000  144.287    0.441
##    .measure3        -91.453   48.220   -1.897    0.058  -91.453   -0.115
##
## R-Square:
##                    Estimate
##     measure1          0.426
##     measure2          0.559
##     measure3             NA
```

5.8.4 Estimates of model fit

```
fitMeasures(measurementModelFit, fit.measures = c("chisq", "df", "pvalue",
                                                  "chisq.scaled", "df.scaled", "pvalue.scaled",
                                                  "chisq.scaling.factor",
                                                  "rmsea", "cfi", "tli", "srmr",
                                                  "rmsea.robust", "cfi.robust", "tli.robust"))
```

```
##                chisq                   df               pvalue
##                    0                    0                   NA
```

```
##         chisq.scaled            df.scaled        pvalue.scaled
##                    0                    0                   NA
## chisq.scaling.factor               rmsea                  cfi
##                   NA                    0                    1
##                  tli                 srmr         rmsea.robust
##                    1                    0                    0
##           cfi.robust           tli.robust
##                    1                    1
```

5.8.5 Residuals

```
residuals(measurementModelFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##          measr1 measr2 measr3
## measure1 0
## measure2 0      0
## measure3 0      0      0
##
## $mean
## measure1 measure2 measure3
##        0        0        0
```

5.8.6 Modification indices

```
modificationindices(measurementModelFit)
```

```
## [1] lhs      op       rhs      mi       epc      sepc.lv  sepc.all sepc.nox
## <0 rows> (or 0-length row.names)
```

5.8.7 Factor scores

```
as_tibble(predict(measurementModelFit))
```

```
## # A tibble: 300 x 1
```

```
##    latentFactor
##    <lvn.mtrx>
##  1 -1.0963978
##  2 -1.1522043
##  3  0.6638866
##  4 -1.3214658
##  5  1.7245583
##  6  0.8962471
##  7  1.5769833
##  8 -1.4425620
##  9  0.3240618
## 10         NA
## # ... with 290 more rows
```

5.8.8 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(measurementModelFit)
```

```
##        latentFactor
## alpha     0.7834550
## omega     0.9604108
## omega2    0.9604108
## omega3    0.9604108
## avevar    0.9171249
```

5.8.9 Path diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(measurementModelFit,
         what = "Std.all",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 2)
```

Figure 5.5: Measurement model

5.9 Confirmatory Factor Analysis (CFA)

The confirmatory factor analysis (CFA) models were fit in the `lavaan` package [Rosseel et al., 2020]. The examples were adapted from the `lavaan` documentation: http://lavaan.ugent.be/tutorial/cfa.html

5.9.1 Specify the model

```
cfaModel_syntax <- '
 #Factor loadings
 visual  =~ x1 + x2 + x3
 textual =~ x4 + x5 + x6
 speed   =~ x7 + x8 + x9
'

cfaModel_fullSyntax <- '
 #Factor loadings (free the factor loading of the first indicator)
 visual  =~ NA*x1 + x2 + x3
 textual =~ NA*x4 + x5 + x6
 speed   =~ NA*x7 + x8 + x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one
 visual ~~ 1*visual
 textual ~~ 1*textual
 speed ~~ 1*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
```

```
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Free intercepts of manifest variables
 x1 ~ int1*1
 x2 ~ int2*1
 x3 ~ int3*1
 x4 ~ int4*1
 x5 ~ int5*1
 x6 ~ int6*1
 x7 ~ int7*1
 x8 ~ int8*1
 x9 ~ int9*1
'
```

5.9.1.1 Model Syntax in Table Form:

```
lavaanify(cfaModel_syntax)
```

```
##    id     lhs op     rhs user block group free ustart exo label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0         .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0         .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0         .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0         .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0         .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0         .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0         .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0         .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0         .p9.
## 10 10      x1 ~~      x1    0     1     1    0      0   0        .p10.
## 11 11      x2 ~~      x2    0     1     1    0      0   0        .p11.
## 12 12      x3 ~~      x3    0     1     1    0      0   0        .p12.
## 13 13      x4 ~~      x4    0     1     1    0      0   0        .p13.
## 14 14      x5 ~~      x5    0     1     1    0      0   0        .p14.
## 15 15      x6 ~~      x6    0     1     1    0      0   0        .p15.
## 16 16      x7 ~~      x7    0     1     1    0      0   0        .p16.
## 17 17      x8 ~~      x8    0     1     1    0      0   0        .p17.
## 18 18      x9 ~~      x9    0     1     1    0      0   0        .p18.
## 19 19  visual ~~  visual    0     1     1    0      0   0        .p19.
## 20 20 textual ~~ textual    0     1     1    0      0   0        .p20.
## 21 21   speed ~~   speed    0     1     1    0      0   0        .p21.
```

```
lavaanify(cfaModel_fullSyntax)
```

```
##    id     lhs op     rhs user block group free ustart exo label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0         .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0         .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0         .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0         .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0         .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0         .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0         .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0         .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0         .p9.
## 10 10  visual ~1            1     1     1    0      0   0        .p10.
## 11 11 textual ~1            1     1     1    0      0   0        .p11.
## 12 12   speed ~1            1     1     1    0      0   0        .p12.
## 13 13  visual ~~  visual    1     1     1    0      1   0        .p13.
## 14 14 textual ~~ textual    1     1     1    0      1   0        .p14.
## 15 15   speed ~~   speed    1     1     1    0      1   0        .p15.
## 16 16  visual ~~ textual    1     1     1   10     NA   0        .p16.
## 17 17  visual ~~   speed    1     1     1   11     NA   0        .p17.
## 18 18 textual ~~   speed    1     1     1   12     NA   0        .p18.
## 19 19      x1 ~~      x1    1     1     1   13     NA   0        .p19.
## 20 20      x2 ~~      x2    1     1     1   14     NA   0        .p20.
## 21 21      x3 ~~      x3    1     1     1   15     NA   0        .p21.
## 22 22      x4 ~~      x4    1     1     1   16     NA   0        .p22.
## 23 23      x5 ~~      x5    1     1     1   17     NA   0        .p23.
## 24 24      x6 ~~      x6    1     1     1   18     NA   0        .p24.
## 25 25      x7 ~~      x7    1     1     1   19     NA   0        .p25.
## 26 26      x8 ~~      x8    1     1     1   20     NA   0        .p26.
## 27 27      x9 ~~      x9    1     1     1   21     NA   0        .p27.
## 28 28      x1 ~1            1     1     1   22     NA   0  int1  .p28.
## 29 29      x2 ~1            1     1     1   23     NA   0  int2  .p29.
## 30 30      x3 ~1            1     1     1   24     NA   0  int3  .p30.
## 31 31      x4 ~1            1     1     1   25     NA   0  int4  .p31.
## 32 32      x5 ~1            1     1     1   26     NA   0  int5  .p32.
## 33 33      x6 ~1            1     1     1   27     NA   0  int6  .p33.
## 34 34      x7 ~1            1     1     1   28     NA   0  int7  .p34.
## 35 35      x8 ~1            1     1     1   29     NA   0  int8  .p35.
## 36 36      x9 ~1            1     1     1   30     NA   0  int9  .p36.
```

5.9.2 Fit the model

```
cfaModelFit <- cfa(cfaModel_syntax,
                   data = HolzingerSwineford1939,
                   missing = "ML",
                   estimator = "MLR",
                   std.lv = TRUE)

cfaModelFit_full <- lavaan(cfaModel_fullSyntax,
                           data = HolzingerSwineford1939,
                           missing = "ML",
                           estimator = "MLR")
```

5.9.3 Display summary output

```
summary(cfaModelFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 42 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        30
##
##   Number of observations                           301
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                85.306      87.132
##   Degrees of freedom                                24          24
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.979
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               918.852     880.082
##   Degrees of freedom                                36          36
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.044
##
```

```
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.931       0.925
##   Tucker-Lewis Index (TLI)                       0.896       0.888
##
##   Robust Comparative Fit Index (CFI)                         0.930
##   Robust Tucker-Lewis Index (TLI)                            0.895
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3737.745   -3737.745
##   Scaling correction factor                                  1.093
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -3695.092   -3695.092
##   Scaling correction factor                                  1.043
##       for the MLR correction
##
##   Akaike (AIC)                                7535.490    7535.490
##   Bayesian (BIC)                              7646.703    7646.703
##   Sample-size adjusted Bayesian (BIC)         7551.560    7551.560
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.092       0.093
##   90 Percent confidence interval - lower         0.071       0.073
##   90 Percent confidence interval - upper         0.114       0.115
##   P-value RMSEA <= 0.05                          0.001       0.001
##
##   Robust RMSEA                                               0.092
##   90 Percent confidence interval - lower                     0.072
##   90 Percent confidence interval - upper                     0.114
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.060       0.060
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1                0.900    0.100    8.973    0.000    0.900    0.772
```

```
##     x2                0.498    0.088    5.681    0.000    0.498    0.424
##     x3                0.656    0.080    8.151    0.000    0.656    0.581
##   textual =~
##     x4                0.990    0.061   16.150    0.000    0.990    0.852
##     x5                1.102    0.055   20.146    0.000    1.102    0.855
##     x6                0.917    0.058   15.767    0.000    0.917    0.838
##   speed =~
##     x7                0.619    0.086    7.193    0.000    0.619    0.570
##     x8                0.731    0.093    7.875    0.000    0.731    0.723
##     x9                0.670    0.099    6.761    0.000    0.670    0.665
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.459    0.073    6.258    0.000    0.459    0.459
##     speed             0.471    0.119    3.954    0.000    0.471    0.471
##   textual ~~
##     speed             0.283    0.085    3.311    0.001    0.283    0.283
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                4.936    0.067   73.473    0.000    4.936    4.235
##    .x2                6.088    0.068   89.855    0.000    6.088    5.179
##    .x3                2.250    0.065   34.579    0.000    2.250    1.993
##    .x4                3.061    0.067   45.694    0.000    3.061    2.634
##    .x5                4.341    0.074   58.452    0.000    4.341    3.369
##    .x6                2.186    0.063   34.667    0.000    2.186    1.998
##    .x7                4.186    0.063   66.766    0.000    4.186    3.848
##    .x8                5.527    0.058   94.854    0.000    5.527    5.467
##    .x9                5.374    0.058   92.546    0.000    5.374    5.334
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                0.549    0.156    3.509    0.000    0.549    0.404
##    .x2                1.134    0.112   10.135    0.000    1.134    0.821
##    .x3                0.844    0.100    8.419    0.000    0.844    0.662
##    .x4                0.371    0.050    7.382    0.000    0.371    0.275
##    .x5                0.446    0.057    7.870    0.000    0.446    0.269
##    .x6                0.356    0.047    7.658    0.000    0.356    0.298
##    .x7                0.799    0.097    8.222    0.000    0.799    0.676
##    .x8                0.488    0.120    4.080    0.000    0.488    0.477
##    .x9                0.566    0.119    4.768    0.000    0.566    0.558
##     visual            1.000                               1.000    1.000
```

```
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.596
##     x2                0.179
##     x3                0.338
##     x4                0.725
##     x5                0.731
##     x6                0.702
##     x7                0.324
##     x8                0.523
##     x9                0.442
```

```
summary(cfaModelFit_full,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 42 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        30
##
##   Number of observations                           301
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                85.306      87.132
##   Degrees of freedom                                24          24
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.979
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               918.852     880.082
##   Degrees of freedom                                36          36
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.044
##
## User Model versus Baseline Model:
```

```
##
##   Comparative Fit Index (CFI)                    0.931       0.925
##   Tucker-Lewis Index (TLI)                       0.896       0.888
##
##   Robust Comparative Fit Index (CFI)                         0.930
##   Robust Tucker-Lewis Index (TLI)                            0.895
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3737.745   -3737.745
##   Scaling correction factor                                  1.093
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -3695.092   -3695.092
##   Scaling correction factor                                  1.043
##       for the MLR correction
##
##   Akaike (AIC)                                7535.490    7535.490
##   Bayesian (BIC)                              7646.703    7646.703
##   Sample-size adjusted Bayesian (BIC)         7551.560    7551.560
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.092       0.093
##   90 Percent confidence interval - lower         0.071       0.073
##   90 Percent confidence interval - upper         0.114       0.115
##   P-value RMSEA <= 0.05                          0.001       0.001
##
##   Robust RMSEA                                               0.092
##   90 Percent confidence interval - lower                     0.072
##   90 Percent confidence interval - upper                     0.114
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.060       0.060
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1                0.900    0.100    8.973    0.000    0.900    0.772
##     x2                0.498    0.088    5.681    0.000    0.498    0.424
```

```
##     x3                0.656    0.080    8.151    0.000    0.656    0.581
##   textual =~
##     x4                0.990    0.061   16.150    0.000    0.990    0.852
##     x5                1.102    0.055   20.146    0.000    1.102    0.855
##     x6                0.917    0.058   15.767    0.000    0.917    0.838
##   speed =~
##     x7                0.619    0.086    7.193    0.000    0.619    0.570
##     x8                0.731    0.093    7.875    0.000    0.731    0.723
##     x9                0.670    0.099    6.761    0.000    0.670    0.665
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.459    0.073    6.258    0.000    0.459    0.459
##     speed             0.471    0.119    3.954    0.000    0.471    0.471
##   textual ~~
##     speed             0.283    0.085    3.311    0.001    0.283    0.283
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1      (int1)    4.936    0.067   73.473    0.000    4.936    4.235
##    .x2      (int2)    6.088    0.068   89.855    0.000    6.088    5.179
##    .x3      (int3)    2.250    0.065   34.579    0.000    2.250    1.993
##    .x4      (int4)    3.061    0.067   45.694    0.000    3.061    2.634
##    .x5      (int5)    4.341    0.074   58.452    0.000    4.341    3.369
##    .x6      (int6)    2.186    0.063   34.667    0.000    2.186    1.998
##    .x7      (int7)    4.186    0.063   66.766    0.000    4.186    3.848
##    .x8      (int8)    5.527    0.058   94.854    0.000    5.527    5.467
##    .x9      (int9)    5.374    0.058   92.546    0.000    5.374    5.334
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1                0.549    0.156    3.509    0.000    0.549    0.404
##    .x2                1.134    0.112   10.135    0.000    1.134    0.821
##    .x3                0.844    0.100    8.419    0.000    0.844    0.662
##    .x4                0.371    0.050    7.382    0.000    0.371    0.275
##    .x5                0.446    0.057    7.870    0.000    0.446    0.269
##    .x6                0.356    0.047    7.658    0.000    0.356    0.298
##    .x7                0.799    0.097    8.222    0.000    0.799    0.676
##    .x8                0.488    0.120    4.080    0.000    0.488    0.477
```

```
##    .x9                0.566    0.119    4.768    0.000    0.566    0.558
##
## R-Square:
##                    Estimate
##     x1                0.596
##     x2                0.179
##     x3                0.338
##     x4                0.725
##     x5                0.731
##     x6                0.702
##     x7                0.324
##     x8                0.523
##     x9                0.442
```

5.9.4 Estimates of model fit

```
fitMeasures(cfaModelFit, fit.measures = c("chisq", "df", "pvalue",
                                          "chisq.scaled", "df.scaled", "pvalue.scaled"
                                          "chisq.scaling.factor",
                                          "rmsea", "cfi", "tli", "srmr",
                                          "rmsea.robust", "cfi.robust", "tli.robust"))
```

```
##                chisq                   df               pvalue
##               85.306               24.000                0.000
##         chisq.scaled            df.scaled        pvalue.scaled
##               87.132               24.000                0.000
## chisq.scaling.factor                rmsea                  cfi
##                0.979                0.092                0.931
##                  tli                 srmr         rmsea.robust
##                0.896                0.060                0.092
##           cfi.robust           tli.robust
##                0.930                0.895
```

5.9.5 Residuals

```
residuals(cfaModelFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
```

```
##    x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.030  0.000
## x3 -0.008  0.094  0.000
## x4  0.071 -0.012 -0.068  0.000
## x5 -0.009 -0.027 -0.151  0.005  0.000
## x6  0.060  0.030 -0.026 -0.009  0.003  0.000
## x7 -0.140 -0.189 -0.084  0.037 -0.036 -0.014  0.000
## x8 -0.039 -0.052 -0.012 -0.067 -0.036 -0.022  0.075  0.000
## x9  0.149  0.073  0.147  0.048  0.067  0.056 -0.038 -0.032  0.000
##
## $mean
## x1 x2 x3 x4 x5 x6 x7 x8 x9
##  0  0  0  0  0  0  0  0  0
```

5.9.6 Modification indices

```
modificationindices(cfaModelFit)
```

```
##        lhs op rhs     mi    epc sepc.lv sepc.all sepc.nox
## 37  visual =~  x4  1.211  0.069   0.069    0.059    0.059
## 38  visual =~  x5  7.441 -0.189  -0.189   -0.147   -0.147
## 39  visual =~  x6  2.843  0.100   0.100    0.092    0.092
## 40  visual =~  x7 18.631 -0.380  -0.380   -0.349   -0.349
## 41  visual =~  x8  4.295 -0.189  -0.189   -0.187   -0.187
## 42  visual =~  x9 36.411  0.519   0.519    0.515    0.515
## 43 textual =~  x1  8.903  0.347   0.347    0.297    0.297
## 44 textual =~  x2  0.017 -0.011  -0.011   -0.010   -0.010
## 45 textual =~  x3  9.151 -0.269  -0.269   -0.238   -0.238
## 46 textual =~  x7  0.098 -0.021  -0.021   -0.019   -0.019
## 47 textual =~  x8  3.359 -0.120  -0.120   -0.118   -0.118
## 48 textual =~  x9  4.796  0.137   0.137    0.136    0.136
## 49   speed =~  x1  0.014  0.015   0.015    0.013    0.013
## 50   speed =~  x2  1.580 -0.123  -0.123   -0.105   -0.105
## 51   speed =~  x3  0.716  0.084   0.084    0.075    0.075
## 52   speed =~  x4  0.003 -0.003  -0.003   -0.003   -0.003
## 53   speed =~  x5  0.201 -0.027  -0.027   -0.021   -0.021
## 54   speed =~  x6  0.273  0.027   0.027    0.025    0.025
## 55      x1 ~~  x2  3.606 -0.184  -0.184   -0.233   -0.233
## 56      x1 ~~  x3  0.935 -0.139  -0.139   -0.203   -0.203
## 57      x1 ~~  x4  3.554  0.078   0.078    0.173    0.173
## 58      x1 ~~  x5  0.522 -0.033  -0.033   -0.067   -0.067
## 59      x1 ~~  x6  0.048  0.009   0.009    0.020    0.020
```

```
## 60      x1 ~~  x7  5.420 -0.129  -0.129   -0.195   -0.195
## 61      x1 ~~  x8  0.634 -0.041  -0.041   -0.079   -0.079
## 62      x1 ~~  x9  7.335  0.138   0.138    0.247    0.247
## 63      x2 ~~  x3  8.532  0.218   0.218    0.223    0.223
## 64      x2 ~~  x4  0.534 -0.034  -0.034   -0.052   -0.052
## 65      x2 ~~  x5  0.023 -0.008  -0.008   -0.011   -0.011
## 66      x2 ~~  x6  0.785  0.039   0.039    0.062    0.062
## 67      x2 ~~  x7  8.918 -0.183  -0.183   -0.192   -0.192
## 68      x2 ~~  x8  0.054 -0.012  -0.012   -0.017   -0.017
## 69      x2 ~~  x9  1.895  0.075   0.075    0.094    0.094
## 70      x3 ~~  x4  0.142 -0.016  -0.016   -0.028   -0.028
## 71      x3 ~~  x5  7.858 -0.130  -0.130   -0.212   -0.212
## 72      x3 ~~  x6  1.855  0.055   0.055    0.100    0.100
## 73      x3 ~~  x7  0.638 -0.044  -0.044   -0.054   -0.054
## 74      x3 ~~  x8  0.059 -0.012  -0.012   -0.019   -0.019
## 75      x3 ~~  x9  4.126  0.102   0.102    0.147    0.147
## 76      x4 ~~  x5  2.534  0.186   0.186    0.457    0.457
## 77      x4 ~~  x6  6.221 -0.235  -0.235   -0.646   -0.646
## 78      x4 ~~  x7  5.920  0.098   0.098    0.180    0.180
## 79      x4 ~~  x8  3.805 -0.069  -0.069   -0.162   -0.162
## 80      x4 ~~  x9  0.196 -0.016  -0.016   -0.035   -0.035
## 81      x5 ~~  x6  0.916  0.101   0.101    0.253    0.253
## 82      x5 ~~  x7  1.233 -0.049  -0.049   -0.083   -0.083
## 83      x5 ~~  x8  0.347  0.023   0.023    0.049    0.049
## 84      x5 ~~  x9  0.999  0.040   0.040    0.079    0.079
## 85      x6 ~~  x7  0.259 -0.020  -0.020   -0.037   -0.037
## 86      x6 ~~  x8  0.275  0.018   0.018    0.043    0.043
## 87      x6 ~~  x9  0.097 -0.011  -0.011   -0.024   -0.024
## 88      x7 ~~  x8 34.145  0.536   0.536    0.859    0.859
## 89      x7 ~~  x9  5.183 -0.187  -0.187   -0.278   -0.278
## 90      x8 ~~  x9 14.946 -0.423  -0.423   -0.805   -0.805
```

5.9.7 Factor scores

```
as_tibble(predict(cfaModelFit))
```

```
## # A tibble: 301 x 3
##    visual      textual     speed
##    <lvn.mtrx>  <lvn.mtrx>  <lvn.mtrx>
##  1 -0.90891121 -0.13897749  0.09928950
##  2  0.05504485 -1.02327078  1.00971777
##  3 -0.84635388 -1.89178504 -1.35691384
##  4  0.46613185  0.01867839 -0.43801231
```

```
##  5 -0.46231162 -0.12352306  0.31370056
##  6  0.02585126 -1.34366645  1.14428315
##  7 -1.09487033  0.66446259  0.07055384
##  8  0.11876511 -0.10622255 -0.85928111
##  9 -0.28230909  0.40854205  0.32344842
## 10 -1.17184874  0.09388286 -0.91475661
## # ... with 291 more rows
```

5.9.8 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModelFit)
```

```
##           visual   textual     speed
## alpha  0.6261171 0.8827069 0.6884550
## omega  0.6253181 0.8851754 0.6877603
## omega2 0.6253181 0.8851754 0.6877603
## omega3 0.6120054 0.8850614 0.6858424
## avevar 0.3705590 0.7210165 0.4244886
```

5.9.9 Path Diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModelFit,
         what = "Std.all",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 0.8)
```

5.9.10 Modify model based on modification indices

5.9.10.1 Specify the model

Figure 5.6: Confirmatory factor analysis model

```
cfaModel2_syntax <- '
 #Factor loadings
 textual =~ x4 + x5 + x6
 visual  =~ x1 + x2 + x3 + x9
 speed   =~ x7 + x8 + x9
'

cfaModel2_fullSyntax <- '
 #Factor loadings (free the factor loading of the first indicator)
 textual =~ NA*x4 + x5 + x6
 visual  =~ NA*x1 + x2 + x3 + x9
 speed   =~ NA*x7 + x8 + x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one
 visual ~~ 1*visual
 textual ~~ 1*textual
 speed ~~ 1*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Free intercepts of manifest variables
 x1 ~ int1*1
 x2 ~ int2*1
 x3 ~ int3*1
 x4 ~ int4*1
 x5 ~ int5*1
```

```
 x6 ~ int6*1
 x7 ~ int7*1
 x8 ~ int8*1
 x9 ~ int9*1
'
```

5.9.10.2 Fit the model

```
cfaModel2Fit <- cfa(cfaModel2_syntax,
                    data = HolzingerSwineford1939,
                    missing = "ML",
                    estimator = "MLR",
                    std.lv = TRUE)

cfaModel2Fit_full <- lavaan(cfaModel2_fullSyntax,
                            data = HolzingerSwineford1939,
                            missing = "ML",
                            estimator = "MLR")
```

5.9.10.3 Display summary output

```
summary(cfaModel2Fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 39 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        31
##
##   Number of observations                           301
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                52.382      51.725
##   Degrees of freedom                                23          23
##   P-value (Chi-square)                           0.000       0.001
##   Scaling correction factor                                  1.013
```

```
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               918.852     880.082
##   Degrees of freedom                                36          36
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.044
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.967       0.966
##   Tucker-Lewis Index (TLI)                       0.948       0.947
##
##   Robust Comparative Fit Index (CFI)                         0.967
##   Robust Tucker-Lewis Index (TLI)                            0.948
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3721.283   -3721.283
##   Scaling correction factor                                  1.065
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -3695.092   -3695.092
##   Scaling correction factor                                  1.043
##       for the MLR correction
##
##   Akaike (AIC)                                7504.566    7504.566
##   Bayesian (BIC)                              7619.487    7619.487
##   Sample-size adjusted Bayesian (BIC)         7521.172    7521.172
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.065       0.064
##   90 Percent confidence interval - lower         0.042       0.041
##   90 Percent confidence interval - upper         0.089       0.088
##   P-value RMSEA <= 0.05                          0.133       0.143
##
##   Robust RMSEA                                               0.065
##   90 Percent confidence interval - lower                     0.041
##   90 Percent confidence interval - upper                     0.089
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.041       0.041
##
## Parameter Estimates:
```

```
##
##   Standard errors                              Sandwich
##   Information bread                            Observed
##   Observed information based on                 Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   textual =~
##     x4                0.989    0.061   16.150    0.000    0.989    0.851
##     x5                1.103    0.055   20.196    0.000    1.103    0.856
##     x6                0.916    0.058   15.728    0.000    0.916    0.838
##   visual =~
##     x1                0.885    0.091    9.673    0.000    0.885    0.759
##     x2                0.511    0.082    6.233    0.000    0.511    0.435
##     x3                0.667    0.072    9.214    0.000    0.667    0.590
##     x9                0.387    0.066    5.895    0.000    0.387    0.384
##   speed =~
##     x7                0.666    0.075    8.874    0.000    0.666    0.612
##     x8                0.804    0.080    9.997    0.000    0.804    0.795
##     x9                0.450    0.065    6.901    0.000    0.450    0.447
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   textual ~~
##     visual            0.453    0.074    6.091    0.000    0.453    0.453
##     speed             0.206    0.076    2.731    0.006    0.206    0.206
##   visual ~~
##     speed             0.301    0.084    3.567    0.000    0.301    0.301
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x4                3.061    0.067   45.694    0.000    3.061    2.634
##    .x5                4.341    0.074   58.452    0.000    4.341    3.369
##    .x6                2.186    0.063   34.667    0.000    2.186    1.998
##    .x1                4.936    0.067   73.473    0.000    4.936    4.235
##    .x2                6.088    0.068   89.855    0.000    6.088    5.179
##    .x3                2.250    0.065   34.579    0.000    2.250    1.993
##    .x9                5.374    0.058   92.546    0.000    5.374    5.334
##    .x7                4.186    0.063   66.766    0.000    4.186    3.848
##    .x8                5.527    0.058   94.854    0.000    5.527    5.467
##     textual           0.000                               0.000    0.000
##     visual            0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
```

```
##    .x4                0.373    0.050    7.403    0.000    0.373    0.276
##    .x5                0.444    0.057    7.827    0.000    0.444    0.267
##    .x6                0.357    0.046    7.674    0.000    0.357    0.298
##    .x1                0.576    0.135    4.256    0.000    0.576    0.424
##    .x2                1.120    0.109   10.296    0.000    1.120    0.811
##    .x3                0.830    0.087    9.561    0.000    0.830    0.651
##    .x9                0.558    0.066    8.482    0.000    0.558    0.550
##    .x7                0.740    0.089    8.266    0.000    0.740    0.625
##    .x8                0.375    0.105    3.583    0.000    0.375    0.367
##     textual           1.000                               1.000    1.000
##     visual            1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     x4                0.724
##     x5                0.733
##     x6                0.702
##     x1                0.576
##     x2                0.189
##     x3                0.349
##     x9                0.450
##     x7                0.375
##     x8                0.633
```

```
summary(cfaModel2Fit_full,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 39 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        31
##
##   Number of observations                           301
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                52.382      51.725
##   Degrees of freedom                                23          23
##   P-value (Chi-square)                           0.000       0.001
##   Scaling correction factor                                  1.013
```

```
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                                918.852     880.082
##   Degrees of freedom                                 36          36
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   1.044
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.967       0.966
##   Tucker-Lewis Index (TLI)                        0.948       0.947
##
##   Robust Comparative Fit Index (CFI)                          0.967
##   Robust Tucker-Lewis Index (TLI)                             0.948
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3721.283   -3721.283
##   Scaling correction factor                                   1.065
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -3695.092   -3695.092
##   Scaling correction factor                                   1.043
##       for the MLR correction
##
##   Akaike (AIC)                                7504.566    7504.566
##   Bayesian (BIC)                              7619.487    7619.487
##   Sample-size adjusted Bayesian (BIC)         7521.172    7521.172
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                           0.065       0.064
##   90 Percent confidence interval - lower          0.042       0.041
##   90 Percent confidence interval - upper          0.089       0.088
##   P-value RMSEA <= 0.05                           0.133       0.143
##
##   Robust RMSEA                                                0.065
##   90 Percent confidence interval - lower                      0.041
##   90 Percent confidence interval - upper                      0.089
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                            0.041       0.041
##
## Parameter Estimates:
```

```
##
##   Standard errors                              Sandwich
##   Information bread                            Observed
##   Observed information based on                 Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   textual =~
##     x4                0.989    0.061   16.150    0.000    0.989    0.851
##     x5                1.103    0.055   20.196    0.000    1.103    0.856
##     x6                0.916    0.058   15.728    0.000    0.916    0.838
##   visual =~
##     x1                0.885    0.091    9.673    0.000    0.885    0.759
##     x2                0.511    0.082    6.233    0.000    0.511    0.435
##     x3                0.667    0.072    9.214    0.000    0.667    0.590
##     x9                0.387    0.066    5.895    0.000    0.387    0.384
##   speed =~
##     x7                0.666    0.075    8.874    0.000    0.666    0.612
##     x8                0.804    0.080    9.997    0.000    0.804    0.795
##     x9                0.450    0.065    6.901    0.000    0.450    0.447
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   textual ~~
##     visual            0.453    0.074    6.091    0.000    0.453    0.453
##   visual ~~
##     speed             0.301    0.084    3.567    0.000    0.301    0.301
##   textual ~~
##     speed             0.206    0.076    2.731    0.006    0.206    0.206
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1      (int1)    4.936    0.067   73.473    0.000    4.936    4.235
##    .x2      (int2)    6.088    0.068   89.855    0.000    6.088    5.179
##    .x3      (int3)    2.250    0.065   34.579    0.000    2.250    1.993
##    .x4      (int4)    3.061    0.067   45.694    0.000    3.061    2.634
##    .x5      (int5)    4.341    0.074   58.452    0.000    4.341    3.369
##    .x6      (int6)    2.186    0.063   34.667    0.000    2.186    1.998
##    .x7      (int7)    4.186    0.063   66.766    0.000    4.186    3.848
##    .x8      (int8)    5.527    0.058   94.854    0.000    5.527    5.467
##    .x9      (int9)    5.374    0.058   92.546    0.000    5.374    5.334
##
## Variances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1                0.576    0.135    4.256    0.000    0.576    0.424
##    .x2                1.120    0.109   10.296    0.000    1.120    0.811
##    .x3                0.830    0.087    9.561    0.000    0.830    0.651
##    .x4                0.373    0.050    7.403    0.000    0.373    0.276
##    .x5                0.444    0.057    7.827    0.000    0.444    0.267
##    .x6                0.357    0.046    7.674    0.000    0.357    0.298
##    .x7                0.740    0.089    8.266    0.000    0.740    0.625
##    .x8                0.375    0.105    3.583    0.000    0.375    0.367
##    .x9                0.558    0.066    8.482    0.000    0.558    0.550
##
## R-Square:
##                    Estimate
##     x1                0.576
##     x2                0.189
##     x3                0.349
##     x4                0.724
##     x5                0.733
##     x6                0.702
##     x7                0.375
##     x8                0.633
##     x9                0.450
```

5.9.10.4 Estimates of model fit

```
fitMeasures(cfaModel2Fit, fit.measures = c("chisq", "df", "pvalue",
                                           "chisq.scaled", "df.scaled", "pvalue.scaled"
                                           "chisq.scaling.factor",
                                           "rmsea", "cfi", "tli", "srmr",
                                           "rmsea.robust", "cfi.robust", "tli.robust"))
```

```
##                chisq                   df               pvalue
##               52.382               23.000                0.000
##         chisq.scaled            df.scaled        pvalue.scaled
##               51.725               23.000                0.001
## chisq.scaling.factor                rmsea                  cfi
##                1.013                0.065                0.967
##                  tli                 srmr         rmsea.robust
##                0.948                0.041                0.065
##           cfi.robust           tli.robust
##                0.967                0.948
```

5.9.10.5 Residuals

```
residuals(cfaModel2Fit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##    x4     x5     x6     x1     x2     x3     x9     x7     x8
## x4  0.000
## x5  0.005  0.000
## x6 -0.008  0.003  0.000
## x1  0.080 -0.001  0.069  0.000
## x2 -0.015 -0.029  0.028 -0.033  0.000
## x3 -0.069 -0.152 -0.026 -0.008  0.083  0.000
## x9 -0.018  0.000 -0.009 -0.003 -0.019  0.023  0.000
## x7  0.066 -0.006  0.015 -0.073 -0.156 -0.037 -0.003  0.000
## x8 -0.033 -0.002  0.012  0.042 -0.012  0.045  0.002  0.000  0.000
##
## $mean
## x4 x5 x6 x1 x2 x3 x9 x7 x8
##  0  0  0  0  0  0  0  0  0
```

5.9.10.6 Path diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModel2Fit,
         what = "Std.all",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 0.8)
```

5.9.10.7 Compare model fit using chi-square difference test for nested models

The modified model with the cross-loading and the original model are considered "nested" models. The original model is nested within the modified model because the modified model includes all of the terms of the original model along with additional terms. Model fit of nested models can be compared with a chi-square difference test.

Figure 5.7: Modified confirmatory factor analysis model

```
anova(cfaModelFit, cfaModel2Fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##              Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## cfaModel2Fit 23 7504.6 7619.5 52.382
## cfaModelFit  24 7535.5 7646.7 85.305     160.61       1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with the cross-loading fits significantly better (i.e., has a significantly smaller chi-square value) than the model without the cross-loading.

5.10 Structural Equation Model (SEM)

The structural equation models models were fit in the `lavaan` package [Rosseel et al., 2020]. The examples were adapted from the `lavaan` documentation: http://lavaan.ugent.be/tutorial/sem.html

5.10.1 Specify the model

```
semModel_syntax <- '
 #Measurement model factor loadings
 ind60 =~ x1 + x2 + x3
 dem60 =~ y1 + y2 + y3 + y4
 dem65 =~ y5 + y6 + y7 + y8

 #Regression paths
 dem60 ~ ind60
 dem65 ~ ind60 + dem60

 #Covariances among residual variances (correlated errors)
 y1 ~~ y5
 y2 ~~ y4 + y6
 y3 ~~ y7
 y4 ~~ y8
```

```
 y6 ~~ y8
'

semModel_fullSyntax <- '
 #Measurement model factor loadings (free the factor loading of the first indicator)
 ind60 =~ NA*x1 + x2 + x3
 dem60 =~ NA*y1 + y2 + y3 + y4
 dem65 =~ NA*y5 + y6 + y7 + y8

 #Regression paths
 dem60 ~ ind60
 dem65 ~ ind60 + dem60

 #Covariances among residual variances (correlated errors)
 y1 ~~ y5
 y2 ~~ y4 + y6
 y3 ~~ y7
 y4 ~~ y8
 y6 ~~ y8

 #Fix latent means to zero
 ind60 ~ 0
 dem60 ~ 0
 dem65 ~ 0

 #Fix latent variances to one
 ind60 ~~ 1*ind60
 dem60 ~~ 1*dem60
 dem65 ~~ 1*dem65

 #Estimate covariances among latent variables (not necessary because the latent variabl

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 y1 ~~ y1
 y2 ~~ y2
 y3 ~~ y3
 y4 ~~ y4
 y5 ~~ y5
 y6 ~~ y6
 y7 ~~ y7
 y8 ~~ y8
```

```
 #Free intercepts of manifest variables
 x1 ~ intx1*1
 x2 ~ intx2*1
 x3 ~ intx3*1
 y1 ~ inty1*1
 y2 ~ inty2*1
 y3 ~ inty3*1
 y4 ~ inty4*1
 y5 ~ inty5*1
 y6 ~ inty6*1
 y7 ~ inty7*1
 y8 ~ inty8*1
'
```

5.10.1.1 Model Syntax in Table Form:

```
lavaanify(semModel_syntax)
```

```
##    id   lhs op   rhs user block group free ustart exo label plabel
## 1   1 ind60 =~    x1    1     1     1    1     NA   0         .p1.
## 2   2 ind60 =~    x2    1     1     1    2     NA   0         .p2.
## 3   3 ind60 =~    x3    1     1     1    3     NA   0         .p3.
## 4   4 dem60 =~    y1    1     1     1    4     NA   0         .p4.
## 5   5 dem60 =~    y2    1     1     1    5     NA   0         .p5.
## 6   6 dem60 =~    y3    1     1     1    6     NA   0         .p6.
## 7   7 dem60 =~    y4    1     1     1    7     NA   0         .p7.
## 8   8 dem65 =~    y5    1     1     1    8     NA   0         .p8.
## 9   9 dem65 =~    y6    1     1     1    9     NA   0         .p9.
## 10 10 dem65 =~    y7    1     1     1   10     NA   0        .p10.
## 11 11 dem65 =~    y8    1     1     1   11     NA   0        .p11.
## 12 12 dem60  ~ ind60    1     1     1   12     NA   0        .p12.
## 13 13 dem65  ~ ind60    1     1     1   13     NA   0        .p13.
## 14 14 dem65  ~ dem60    1     1     1   14     NA   0        .p14.
## 15 15    y1 ~~    y5    1     1     1   15     NA   0        .p15.
## 16 16    y2 ~~    y4    1     1     1   16     NA   0        .p16.
## 17 17    y2 ~~    y6    1     1     1   17     NA   0        .p17.
## 18 18    y3 ~~    y7    1     1     1   18     NA   0        .p18.
## 19 19    y4 ~~    y8    1     1     1   19     NA   0        .p19.
## 20 20    y6 ~~    y8    1     1     1   20     NA   0        .p20.
## 21 21    x1 ~~    x1    0     1     1    0      0   0        .p21.
## 22 22    x2 ~~    x2    0     1     1    0      0   0        .p22.
## 23 23    x3 ~~    x3    0     1     1    0      0   0        .p23.
## 24 24    y1 ~~    y1    0     1     1    0      0   0        .p24.
```

```
## 25 25    y2 ~~    y2    0     1     1    0      0   0        .p25.
## 26 26    y3 ~~    y3    0     1     1    0      0   0        .p26.
## 27 27    y4 ~~    y4    0     1     1    0      0   0        .p27.
## 28 28    y5 ~~    y5    0     1     1    0      0   0        .p28.
## 29 29    y6 ~~    y6    0     1     1    0      0   0        .p29.
## 30 30    y7 ~~    y7    0     1     1    0      0   0        .p30.
## 31 31    y8 ~~    y8    0     1     1    0      0   0        .p31.
## 32 32 ind60 ~~ ind60    0     1     1    0      0   0        .p32.
## 33 33 dem60 ~~ dem60    0     1     1    0      0   0        .p33.
## 34 34 dem65 ~~ dem65    0     1     1    0      0   0        .p34.
```

```
lavaanify(semModel_fullSyntax)
```

```
##    id   lhs op   rhs user block group free ustart exo label plabel
## 1   1 ind60 =~    x1    1     1     1    1     NA   0         .p1.
## 2   2 ind60 =~    x2    1     1     1    2     NA   0         .p2.
## 3   3 ind60 =~    x3    1     1     1    3     NA   0         .p3.
## 4   4 dem60 =~    y1    1     1     1    4     NA   0         .p4.
## 5   5 dem60 =~    y2    1     1     1    5     NA   0         .p5.
## 6   6 dem60 =~    y3    1     1     1    6     NA   0         .p6.
## 7   7 dem60 =~    y4    1     1     1    7     NA   0         .p7.
## 8   8 dem65 =~    y5    1     1     1    8     NA   0         .p8.
## 9   9 dem65 =~    y6    1     1     1    9     NA   0         .p9.
## 10 10 dem65 =~    y7    1     1     1   10     NA   0        .p10.
## 11 11 dem65 =~    y8    1     1     1   11     NA   0        .p11.
## 12 12 dem60  ~ ind60    1     1     1   12     NA   0        .p12.
## 13 13 dem65  ~ ind60    1     1     1   13     NA   0        .p13.
## 14 14 dem65  ~ dem60    1     1     1   14     NA   0        .p14.
## 15 15    y1 ~~    y5    1     1     1   15     NA   0        .p15.
## 16 16    y2 ~~    y4    1     1     1   16     NA   0        .p16.
## 17 17    y2 ~~    y6    1     1     1   17     NA   0        .p17.
## 18 18    y3 ~~    y7    1     1     1   18     NA   0        .p18.
## 19 19    y4 ~~    y8    1     1     1   19     NA   0        .p19.
## 20 20    y6 ~~    y8    1     1     1   20     NA   0        .p20.
## 21 21 ind60 ~1         1     1     1    0      0   0        .p21.
## 22 22 dem60 ~1         1     1     1    0      0   0        .p22.
## 23 23 dem65 ~1         1     1     1    0      0   0        .p23.
## 24 24 ind60 ~~ ind60    1     1     1    0      1   0        .p24.
## 25 25 dem60 ~~ dem60    1     1     1    0      1   0        .p25.
## 26 26 dem65 ~~ dem65    1     1     1    0      1   0        .p26.
## 27 27    x1 ~~    x1    1     1     1   21     NA   0        .p27.
## 28 28    x2 ~~    x2    1     1     1   22     NA   0        .p28.
## 29 29    x3 ~~    x3    1     1     1   23     NA   0        .p29.
## 30 30    y1 ~~    y1    1     1     1   24     NA   0        .p30.
## 31 31    y2 ~~    y2    1     1     1   25     NA   0        .p31.
```

```
## 32 32    y3 ~~    y3    1     1     1   26     NA   0        .p32.
## 33 33    y4 ~~    y4    1     1     1   27     NA   0        .p33.
## 34 34    y5 ~~    y5    1     1     1   28     NA   0        .p34.
## 35 35    y6 ~~    y6    1     1     1   29     NA   0        .p35.
## 36 36    y7 ~~    y7    1     1     1   30     NA   0        .p36.
## 37 37    y8 ~~    y8    1     1     1   31     NA   0        .p37.
## 38 38    x1 ~1          1     1     1   32     NA   0 intx1  .p38.
## 39 39    x2 ~1          1     1     1   33     NA   0 intx2  .p39.
## 40 40    x3 ~1          1     1     1   34     NA   0 intx3  .p40.
## 41 41    y1 ~1          1     1     1   35     NA   0 inty1  .p41.
## 42 42    y2 ~1          1     1     1   36     NA   0 inty2  .p42.
## 43 43    y3 ~1          1     1     1   37     NA   0 inty3  .p43.
## 44 44    y4 ~1          1     1     1   38     NA   0 inty4  .p44.
## 45 45    y5 ~1          1     1     1   39     NA   0 inty5  .p45.
## 46 46    y6 ~1          1     1     1   40     NA   0 inty6  .p46.
## 47 47    y7 ~1          1     1     1   41     NA   0 inty7  .p47.
## 48 48    y8 ~1          1     1     1   42     NA   0 inty8  .p48.
```

5.10.2 Fit the model

```
semModelFit <- sem(semModel_syntax,
                   data = PoliticalDemocracy,
                   missing = "ML",
                   estimator = "MLR",
                   std.lv = TRUE)

semModelFit_full <- lavaan(semModel_fullSyntax,
                           data = PoliticalDemocracy,
                           missing = "ML",
                           estimator = "MLR")
```

5.10.3 Display summary output

```
summary(semModelFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 84 iterations
##
##   Estimator                                         ML
```

```
##   Optimization method                           NLMINB
##   Number of model parameters                        42
##
##   Number of observations                            75
##   Number of missing patterns                        33
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                46.027      51.005
##   Degrees of freedom                                 35          35
##   P-value (Chi-square)                            0.101       0.039
##   Scaling correction factor                                   0.902
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               632.855     609.162
##   Degrees of freedom                                 55          55
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   1.039
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.981       0.971
##   Tucker-Lewis Index (TLI)                        0.970       0.955
##
##   Robust Comparative Fit Index (CFI)                          0.975
##   Robust Tucker-Lewis Index (TLI)                             0.961
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1412.514   -1412.514
##   Scaling correction factor                                   1.027
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -1389.501   -1389.501
##   Scaling correction factor                                   0.970
##       for the MLR correction
##
##   Akaike (AIC)                                2909.029    2909.029
##   Bayesian (BIC)                              3006.363    3006.363
##   Sample-size adjusted Bayesian (BIC)         2873.990    2873.990
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                           0.065       0.078
##   90 Percent confidence interval - lower          0.000       0.010
```

```
##   90 Percent confidence interval - upper         0.112       0.124
##   P-value RMSEA <= 0.05                          0.303       0.175
##
##   Robust RMSEA                                               0.074
##   90 Percent confidence interval - lower                     0.017
##   90 Percent confidence interval - upper                     0.116
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.049       0.049
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   ind60 =~
##     x1                0.656    0.055   11.878    0.000    0.656    0.918
##     x2                1.460    0.110   13.212    0.000    1.460    0.976
##     x3                1.181    0.106   11.181    0.000    1.181    0.859
##   dem60 =~
##     y1                2.073    0.270    7.667    0.000    2.295    0.856
##     y2                2.552    0.313    8.155    0.000    2.825    0.734
##     y3                2.224    0.317    7.023    0.000    2.461    0.740
##     y4                2.538    0.255    9.962    0.000    2.809    0.843
##   dem65 =~
##     y5                0.554    0.234    2.368    0.018    2.044    0.787
##     y6                0.685    0.294    2.330    0.020    2.528    0.767
##     y7                0.723    0.326    2.215    0.027    2.668    0.825
##     y8                0.737    0.332    2.223    0.026    2.722    0.836
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   dem60 ~
##     ind60             0.474    0.157    3.022    0.003    0.429    0.429
##   dem65 ~
##     ind60             0.798    0.493    1.620    0.105    0.216    0.216
##     dem60             2.835    1.329    2.133    0.033    0.850    0.850
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .y1 ~~
##    .y5                0.727    0.568    1.279    0.201    0.727    0.327
```

```
##  .y2 ~~
##    .y4                1.382    0.833    1.658    0.097    1.382    0.295
##    .y6                1.444    0.905    1.596    0.110    1.444    0.261
##  .y3 ~~
##    .y7                1.240    0.672    1.845    0.065    1.240    0.304
##  .y4 ~~
##    .y8                0.132    0.565    0.233    0.815    0.132    0.041
##  .y6 ~~
##    .y8                0.884    0.853    1.037    0.300    0.884    0.234
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                5.067    0.084   59.970    0.000    5.067    7.091
##    .x2                4.799    0.174   27.594    0.000    4.799    3.208
##    .x3                3.581    0.161   22.200    0.000    3.581    2.606
##    .y1                5.516    0.312   17.651    0.000    5.516    2.057
##    .y2                4.285    0.458    9.360    0.000    4.285    1.114
##    .y3                6.584    0.391   16.830    0.000    6.584    1.981
##    .y4                4.420    0.394   11.232    0.000    4.420    1.327
##    .y5                5.015    0.310   16.162    0.000    5.015    1.931
##    .y6                2.937    0.394    7.462    0.000    2.937    0.891
##    .y7                6.333    0.384   16.499    0.000    6.333    1.959
##    .y8                4.042    0.383   10.550    0.000    4.042    1.241
##     ind60             0.000                               0.000    0.000
##    .dem60             0.000                               0.000    0.000
##    .dem65             0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                0.080    0.021    3.844    0.000    0.080    0.157
##    .x2                0.107    0.083    1.286    0.198    0.107    0.048
##    .x3                0.494    0.093    5.291    0.000    0.494    0.262
##    .y1                1.921    0.610    3.151    0.002    1.921    0.267
##    .y2                6.821    1.402    4.867    0.000    6.821    0.461
##    .y3                4.992    1.217    4.102    0.000    4.992    0.452
##    .y4                3.209    0.918    3.494    0.000    3.209    0.289
##    .y5                2.568    0.676    3.796    0.000    2.568    0.381
##    .y6                4.476    0.845    5.297    0.000    4.476    0.412
##    .y7                3.335    0.690    4.833    0.000    3.335    0.319
##    .y8                3.201    1.104    2.899    0.004    3.201    0.302
##     ind60             1.000                               1.000    1.000
##    .dem60             1.000                               0.816    0.816
##    .dem65             1.000                               0.073    0.073
##
## R-Square:
##                    Estimate
```

```
##     x1                0.843
##     x2                0.952
##     x3                0.738
##     y1                0.733
##     y2                0.539
##     y3                0.548
##     y4                0.711
##     y5                0.619
##     y6                0.588
##     y7                0.681
##     y8                0.698
##     dem60             0.184
##     dem65             0.927
```

```
summary(semModelFit_full,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 84 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        42
##
##   Number of observations                            75
##   Number of missing patterns                        33
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                46.027      51.005
##   Degrees of freedom                                35          35
##   P-value (Chi-square)                           0.101       0.039
##   Scaling correction factor                                  0.902
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               632.855     609.162
##   Degrees of freedom                                55          55
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.039
##
## User Model versus Baseline Model:
##
```

```
##   Comparative Fit Index (CFI)                    0.981       0.971
##   Tucker-Lewis Index (TLI)                       0.970       0.955
##
##   Robust Comparative Fit Index (CFI)                         0.975
##   Robust Tucker-Lewis Index (TLI)                            0.961
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1412.514   -1412.514
##   Scaling correction factor                                  1.027
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -1389.501   -1389.501
##   Scaling correction factor                                  0.970
##       for the MLR correction
##
##   Akaike (AIC)                                2909.029    2909.029
##   Bayesian (BIC)                              3006.363    3006.363
##   Sample-size adjusted Bayesian (BIC)         2873.990    2873.990
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.065       0.078
##   90 Percent confidence interval - lower         0.000       0.010
##   90 Percent confidence interval - upper         0.112       0.124
##   P-value RMSEA <= 0.05                          0.303       0.175
##
##   Robust RMSEA                                               0.074
##   90 Percent confidence interval - lower                     0.017
##   90 Percent confidence interval - upper                     0.116
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.049       0.049
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   ind60 =~
##     x1                0.656    0.055   11.878    0.000    0.656    0.918
##     x2                1.460    0.110   13.212    0.000    1.460    0.976
##     x3                1.181    0.106   11.181    0.000    1.181    0.859
```

```
##   dem60 =~
##     y1                2.073    0.270    7.667    0.000    2.295    0.856
##     y2                2.552    0.313    8.155    0.000    2.825    0.734
##     y3                2.224    0.317    7.023    0.000    2.461    0.740
##     y4                2.538    0.255    9.962    0.000    2.809    0.843
##   dem65 =~
##     y5                0.554    0.234    2.368    0.018    2.044    0.787
##     y6                0.685    0.294    2.330    0.020    2.528    0.767
##     y7                0.723    0.326    2.215    0.027    2.668    0.825
##     y8                0.737    0.332    2.223    0.026    2.722    0.836
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   dem60 ~
##     ind60             0.474    0.157    3.022    0.003    0.429    0.429
##   dem65 ~
##     ind60             0.798    0.493    1.620    0.105    0.216    0.216
##     dem60             2.835    1.329    2.133    0.033    0.850    0.850
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .y1 ~~
##    .y5                0.727    0.568    1.279    0.201    0.727    0.327
##  .y2 ~~
##    .y4                1.382    0.833    1.658    0.097    1.382    0.295
##    .y6                1.444    0.905    1.596    0.110    1.444    0.261
##  .y3 ~~
##    .y7                1.240    0.672    1.845    0.065    1.240    0.304
##  .y4 ~~
##    .y8                0.132    0.565    0.233    0.815    0.132    0.041
##  .y6 ~~
##    .y8                0.884    0.853    1.037    0.300    0.884    0.234
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     ind60             0.000                               0.000    0.000
##    .dem60             0.000                               0.000    0.000
##    .dem65             0.000                               0.000    0.000
##    .x1     (intx1)    5.067    0.084   59.970    0.000    5.067    7.091
##    .x2     (intx2)    4.799    0.174   27.594    0.000    4.799    3.208
##    .x3     (intx3)    3.581    0.161   22.200    0.000    3.581    2.606
##    .y1     (inty1)    5.516    0.312   17.651    0.000    5.516    2.057
##    .y2     (inty2)    4.285    0.458    9.360    0.000    4.285    1.114
##    .y3     (inty3)    6.584    0.391   16.830    0.000    6.584    1.981
##    .y4      (int4)    4.420    0.394   11.232    0.000    4.420    1.327
##    .y5      (int5)    5.015    0.310   16.162    0.000    5.015    1.931
```

```
##    .y6      (int6)    2.937    0.394    7.462    0.000    2.937    0.891
##    .y7      (int7)    6.333    0.384   16.499    0.000    6.333    1.959
##    .y8      (int8)    4.042    0.383   10.550    0.000    4.042    1.241
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     ind60             1.000                               1.000    1.000
##    .dem60             1.000                               0.816    0.816
##    .dem65             1.000                               0.073    0.073
##    .x1                0.080    0.021    3.844    0.000    0.080    0.157
##    .x2                0.107    0.083    1.286    0.198    0.107    0.048
##    .x3                0.494    0.093    5.291    0.000    0.494    0.262
##    .y1                1.921    0.610    3.151    0.002    1.921    0.267
##    .y2                6.821    1.402    4.867    0.000    6.821    0.461
##    .y3                4.992    1.217    4.102    0.000    4.992    0.452
##    .y4                3.209    0.918    3.494    0.000    3.209    0.289
##    .y5                2.568    0.676    3.796    0.000    2.568    0.381
##    .y6                4.476    0.845    5.297    0.000    4.476    0.412
##    .y7                3.335    0.690    4.833    0.000    3.335    0.319
##    .y8                3.201    1.104    2.899    0.004    3.201    0.302
##
## R-Square:
##                    Estimate
##     dem60             0.184
##     dem65             0.927
##     x1                0.843
##     x2                0.952
##     x3                0.738
##     y1                0.733
##     y2                0.539
##     y3                0.548
##     y4                0.711
##     y5                0.619
##     y6                0.588
##     y7                0.681
##     y8                0.698
```

5.10.4 Estimates of model fit

```
fitMeasures(semModelFit, fit.measures = c("chisq", "df", "pvalue",
                                          "chisq.scaled", "df.scaled", "pvalue.scaled"
                                          "chisq.scaling.factor",
                                          "rmsea", "cfi", "tli", "srmr",
                                          "rmsea.robust", "cfi.robust", "tli.robust"))
```

```
##                chisq                 df             pvalue
##               46.027             35.000              0.101
##         chisq.scaled          df.scaled      pvalue.scaled
##               51.005             35.000              0.039
## chisq.scaling.factor              rmsea                cfi
##                0.902              0.065              0.981
##                  tli               srmr       rmsea.robust
##                0.970              0.049              0.074
##           cfi.robust         tli.robust
##                0.975              0.961
```

5.10.5 Residuals

```
residuals(semModelFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##    x1     x2     x3     y1     y2     y3     y4     y5     y6     y7     y8
## x1  0.000
## x2  0.001  0.000
## x3  0.006 -0.002  0.000
## y1 -0.011 -0.079 -0.123  0.000
## y2 -0.060 -0.032 -0.054 -0.005  0.000
## y3  0.087  0.023  0.000  0.077 -0.098  0.000
## y4  0.146  0.099  0.064 -0.014  0.012  0.018  0.000
## y5  0.147  0.087  0.041 -0.034 -0.005  0.023 -0.034  0.000
## y6 -0.028 -0.049 -0.031  0.032  0.039 -0.128  0.052 -0.013  0.000
## y7 -0.052 -0.038 -0.045  0.016 -0.001 -0.009 -0.011 -0.004 -0.037  0.000
## y8  0.005 -0.035 -0.056 -0.028  0.064 -0.074  0.028 -0.062  0.006  0.034  0.000
##
## $mean
##     x1     x2     x3     y1     y2     y3     y4     y5     y6     y7     y8
##  0.011  0.006 -0.001 -0.001 -0.009  0.020  0.003  0.016  0.002  0.001  0.009
```

5.10.6 Modification indices

```
modificationindices(semModelFit)
```

```
##       lhs op rhs    mi    epc sepc.lv sepc.all sepc.nox
```

```
## 49  ind60 =~  y1 4.609 -0.540  -0.540   -0.201   -0.201
## 50  ind60 =~  y2 1.396 -0.412  -0.412   -0.107   -0.107
## 51  ind60 =~  y3 0.315  0.189   0.189    0.057    0.057
## 52  ind60 =~  y4 6.324  0.708   0.708    0.213    0.213
## 53  ind60 =~  y5 4.010  0.586   0.586    0.226    0.226
## 54  ind60 =~  y6 0.118 -0.116  -0.116   -0.035   -0.035
## 55  ind60 =~  y7 0.871 -0.323  -0.323   -0.100   -0.100
## 56  ind60 =~  y8 0.258 -0.167  -0.167   -0.051   -0.051
## 57  dem60 =~  x1 0.791  0.038   0.042    0.059    0.059
## 58  dem60 =~  x2 0.078 -0.024  -0.026   -0.017   -0.017
## 59  dem60 =~  x3 0.540 -0.073  -0.081   -0.059   -0.059
## 60  dem60 =~  y5 1.779 -1.349  -1.493   -0.575   -0.575
## 61  dem60 =~  y6 0.471  0.760   0.841    0.255    0.255
## 62  dem60 =~  y7 0.061  0.289   0.320    0.099    0.099
## 63  dem60 =~  y8 0.051  0.247   0.274    0.084    0.084
## 64  dem65 =~  x1 0.521  0.011   0.039    0.054    0.054
## 65  dem65 =~  x2 0.035 -0.005  -0.020   -0.013   -0.013
## 66  dem65 =~  x3 0.424 -0.022  -0.080   -0.058   -0.058
## 67  dem65 =~  y1 3.481 -0.598  -2.209   -0.824   -0.824
## 68  dem65 =~  y2 0.037  0.077   0.284    0.074    0.074
## 69  dem65 =~  y3 0.136 -0.144  -0.532   -0.160   -0.160
## 70  dem65 =~  y4 2.845  0.559   2.063    0.619    0.619
## 71     x1 ~~  x2 0.112 -0.039  -0.039   -0.421   -0.421
## 72     x1 ~~  x3 0.008  0.006   0.006    0.029    0.029
## 73     x1 ~~  y1 0.331  0.036   0.036    0.092    0.092
## 74     x1 ~~  y2 4.854 -0.215  -0.215   -0.291   -0.291
## 75     x1 ~~  y3 2.042  0.130   0.130    0.207    0.207
## 76     x1 ~~  y4 2.176  0.111   0.111    0.220    0.220
## 77     x1 ~~  y5 1.140  0.072   0.072    0.159    0.159
## 78     x1 ~~  y6 0.013  0.009   0.009    0.016    0.016
## 79     x1 ~~  y7 3.615 -0.148  -0.148   -0.288   -0.288
## 80     x1 ~~  y8 0.013 -0.009  -0.009   -0.017   -0.017
## 81     x2 ~~  x3 0.028  0.028   0.028    0.120    0.120
## 82     x2 ~~  y1 0.655 -0.093  -0.093   -0.206   -0.206
## 83     x2 ~~  y2 1.142  0.192   0.192    0.225    0.225
## 84     x2 ~~  y3 0.265 -0.087  -0.087   -0.119   -0.119
## 85     x2 ~~  y4 0.082 -0.040  -0.040   -0.068   -0.068
## 86     x2 ~~  y5 0.058  0.030   0.030    0.058    0.058
## 87     x2 ~~  y6 0.354 -0.090  -0.090   -0.130   -0.130
## 88     x2 ~~  y7 0.494  0.102   0.102    0.171    0.171
## 89     x2 ~~  y8 0.037  0.028   0.028    0.048    0.048
## 90     x3 ~~  y1 0.781 -0.131  -0.131   -0.134   -0.134
## 91     x3 ~~  y2 0.013 -0.026  -0.026   -0.014   -0.014
## 92     x3 ~~  y3 0.128 -0.077  -0.077   -0.049   -0.049
## 93     x3 ~~  y4 0.168  0.074   0.074    0.058    0.058
## 94     x3 ~~  y5 0.143 -0.061  -0.061   -0.054   -0.054
```

```
## 95     x3 ~~  y6 0.428  0.125   0.125    0.084    0.084
## 96     x3 ~~  y7 0.446  0.123   0.123    0.096    0.096
## 97     x3 ~~  y8 0.302 -0.102  -0.102   -0.081   -0.081
## 98     y1 ~~  y2 0.202 -0.247  -0.247   -0.068   -0.068
## 99     y1 ~~  y3 2.744  0.837   0.837    0.270    0.270
## 100    y1 ~~  y4 0.579 -0.370  -0.370   -0.149   -0.149
## 101    y1 ~~  y6 1.772  0.574   0.574    0.196    0.196
## 102    y1 ~~  y7 0.001 -0.010  -0.010   -0.004   -0.004
## 103    y1 ~~  y8 0.398 -0.272  -0.272   -0.110   -0.110
## 104    y2 ~~  y3 3.539 -1.371  -1.371   -0.235   -0.235
## 105    y2 ~~  y5 0.245  0.262   0.262    0.063    0.063
## 106    y2 ~~  y7 0.704  0.524   0.524    0.110    0.110
## 107    y2 ~~  y8 3.213  1.393   1.393    0.298    0.298
## 108    y3 ~~  y4 1.666  0.777   0.777    0.194    0.194
## 109    y3 ~~  y5 0.297  0.278   0.278    0.078    0.078
## 110    y3 ~~  y6 2.017 -0.850  -0.850   -0.180   -0.180
## 111    y3 ~~  y8 0.968 -0.576  -0.576   -0.144   -0.144
## 112    y4 ~~  y5 0.602 -0.343  -0.343   -0.120   -0.120
## 113    y4 ~~  y6 0.463  0.443   0.443    0.117    0.117
## 114    y4 ~~  y7 1.608 -0.654  -0.654   -0.200   -0.200
## 115    y5 ~~  y6 0.040 -0.092  -0.092   -0.027   -0.027
## 116    y5 ~~  y7 0.038  0.088   0.088    0.030    0.030
## 117    y5 ~~  y8 1.133 -0.482  -0.482   -0.168   -0.168
## 118    y6 ~~  y7 0.289 -0.298  -0.298   -0.077   -0.077
## 119    y7 ~~  y8 2.225  0.827   0.827    0.253    0.253
```

5.10.7 Factor scores

```
as_tibble(predict(semModelFit))
```

```
## # A tibble: 75 x 3
##    ind60      dem60      dem65
##    <lvn.mtrx> <lvn.mtrx> <lvn.mtrx>
##  1 -0.8127201 -1.2457028 -3.9854353
##  2  0.2208433 -1.1641583 -2.0389666
##  3  1.0365097  1.3055927  5.0564656
##  4  1.8622609  1.4105198  5.2200074
##  5  1.2953660  1.3618011  4.4174362
##  6  0.2956297  0.2503439  0.1867156
##  7  0.1202354  0.3443112  0.5999217
##  8  0.1811830  0.4073711  1.8972017
##  9  0.3760624 -0.7503033 -1.3881267
## 10  0.4585226  1.7123007  5.6129274
## # ... with 65 more rows
```

5.10.8 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(semModelFit)
```

```
##            ind60     dem60     dem65
## alpha  0.8994675 0.8697475 0.8777902
## omega  0.9410346 0.8456300 0.8660230
## omega2 0.9410346 0.8456300 0.8660230
## omega3 0.9384378 0.8444352 0.8829178
## avevar 0.8531255 0.6161341 0.6488162
```

5.10.9 Path diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(semModelFit,
         what = "Std.all",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 0.7)
```

5.11 Benefits of SEM

There are many benefits of fitting a model in SEM (or in other latent variable approaches). First, unlike classical test theory, SEM can allow correlated errors. With SEM, you do not need to make as restrictive assumptions as in classical test theory. Second, unlike multiple regression, SEM can handle multiple dependent variables simultaneously. Third, SEM uses all available information (data) using a technique called full information maximum likelihood (FIML), even if participants have missing scores on some variables. By contrast, multiple regression and many other statistical analyses use listwise deletion, in which they discard participants if they have a missing score on any of the model variables. Fourth, as described in the next section, SEM can be used to account for different forms of measurement error (e.g., method bias). Accounting for measurement error allows disattenuating associations with other constructs. All of

Figure 5.8: Structural equation model

these benefits allow SEM to generate purer estimation of constructs, more accurate estimates of people's levels on constructs, and more accurate estimates of associations between constructs.

5.11.1 Accounting for Method Bias

SEM/CFA can be used to account for method biases and other forms of measurement error. You can use indicators that reflect different method biases, so that the method biases are discarded as unique errors, and are not combined in the "common variance" of the latent construct. SEM/CFA can be used to fit a multi-trait multi-method matrix to account for method variance. I provide an example of fitting a multi-trait multi-method matrix in CFA in Sections **??** and 12.5.12. But, estimation of a multi-trait multi-method matrix can be challenging without making additional constraints/assumptions.

A more practical utility of SEM is that allows one to obtain "purer" estimates of latent constructs (and people's standing on them) by discarding measurement error, and you do not have to assume all errors are uncorrelated!

5.12 Power Analysis using Monte Carlo Simulation

Power analysis for latent variable modeling approaches like SEM is more complicated than it is for other statistical analyses, such as correlation, multiple regression, t-tests, analysis of variance, etc. Statistical power for these more straightforward analytical approaches can be estimated in G*Power [Faul et al., 2009]: https://www.psychologie.hhu.de/arbeitsgruppen/allgemeine-psychologie-und-arbeitspsychologie/gpower.html

I provide an example of how to conduct power analysis of an SEM model to determine the sample size needed to detect a hypothesized effect of a given effect size. To perform the power analysis, I use Monte Carlo simulation [Hancock and French, 2013, Muthén and Muthén, 2002]. It is named after the Casino de Monte-Carlo in Monaco because Monte Carlo simulations involve random samples (random chance), as might be found in casino gambling. Monte Carlo simulations can be used to determine the sample size needed to detect a target parameter of a given effect size, using the following four steps [Wang and Rhemtulla, in press]:

1. Specify the sample size, a hypothesized true population SEM model, and all of its parameter values (e.g., factor loadings, intercepts, residuals, means, variances, covariances, regression paths, sample size);
2. Generate a large number (e.g., 1,000) of random samples based on the hypothesized model and population values specified;

3. Fit a SEM model to each of the generated samples, and for each one, record whether the target parameter is significantly different from zero;
4. Calculate power as the proportion of simulated samples that produce a statistically significant estimate of the target parameter.

These four steps can be repeated with different sample sizes to identify the sample size that is needed to have a particular level of power (e.g., .80).

Power analysis of structural equation models using Monte Carlo simulation was estimated using the `simsem` package [Pornprasertmanit et al., 2020]. These examples were adapted from the `simsem` documentation:

https://github.com/simsem/simsem/wiki/Vignette

https://github.com/simsem/simsem/wiki/Example-18:-Simulation-with-Varying-Sample-Size

https://github.com/simsem/simsem/wiki/Example-19:-Simulation-with-Varying-Sample-Size-and-Percent-Missing

https://github.com/simsem/simsem/blob/master/SupportingDocs/Examples/Version05/ex18/ex18.R

https://github.com/simsem/simsem/blob/master/SupportingDocs/Examples/Version05/ex19/ex19.R

5.12.1 Specify population model

The population model was used to generate the simulated data. It is important to specify each parameter value in the population model based on theory and/or prior empirical research, especially meta-analysis of the target population (when possible).

```
populationModel <- '
 #Specify measurement model factor loadings
 ind60 =~ .7*x1 + .7*x2 + .7*x3
 dem60 =~ .7*y1 + .7*y2 + .7*y3 + .7*y4
 dem65 =~ .7*y5 + .7*y6 + .7*y7 + .7*y8

 #Specify regression coefficients
 dem60 ~ .4*ind60
 dem65 ~ .25*ind60 + .85*dem60

 #Fix latent means to zero
 ind60 ~ 0
 dem60 ~ 0
 dem65 ~ 0
```

```
#Fix latent variances to one
ind60 ~~ 1*ind60
dem60 ~~ 1*dem60
dem65 ~~ 1*dem65

#Specify covariances among latent variables (not necessary because the latent variable

#Specify residual variances of manifest variables
x1 ~~ (1-.7^2)*x1
x2 ~~ (1-.7^2)*x2
x3 ~~ (1-.7^2)*x3
y1 ~~ (1-.7^2)*y1
y2 ~~ (1-.7^2)*y2
y3 ~~ (1-.7^2)*y3
y4 ~~ (1-.7^2)*y4
y5 ~~ (1-.7^2)*y5
y6 ~~ (1-.7^2)*y6
y7 ~~ (1-.7^2)*y7
y8 ~~ (1-.7^2)*y8

#Specify intercepts of manifest variables
x1 ~ 0*1
x2 ~ 0*1
x3 ~ 0*1
y1 ~ 0*1
y2 ~ 0*1
y3 ~ 0*1
y4 ~ 0*1
y5 ~ 0*1
y6 ~ 0*1
y7 ~ 0*1
y8 ~ 0*1
'
```

5.12.1.1 Show the model's fixed and default values

```
populationModel_fit <- lavaan(populationModel, do.fit = FALSE)
```

```
summary(populationModel_fit,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 did not run (perhaps do.fit = FALSE)?
```

```
## ** WARNING ** Estimates below are simply the starting values
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         0
##
##   Number of observations                             0
##
##
## Parameter Estimates:
##
##
## Latent Variables:
##                    Estimate   Std.lv  Std.all
##   ind60 =~
##     x1                0.700    0.700    0.700
##     x2                0.700    0.700    0.700
##     x3                0.700    0.700    0.700
##   dem60 =~
##     y1                0.700    0.754    0.726
##     y2                0.700    0.754    0.726
##     y3                0.700    0.754    0.726
##     y4                0.700    0.754    0.726
##   dem65 =~
##     y5                0.700    1.007    0.816
##     y6                0.700    1.007    0.816
##     y7                0.700    1.007    0.816
##     y8                0.700    1.007    0.816
##
## Regressions:
##                    Estimate   Std.lv  Std.all
##   dem60 ~
##     ind60             0.400    0.371    0.371
##   dem65 ~
##     ind60             0.250    0.174    0.174
##     dem60             0.850    0.636    0.636
##
## Intercepts:
##                    Estimate   Std.lv  Std.all
##     ind60             0.000    0.000    0.000
##    .dem60             0.000    0.000    0.000
##    .dem65             0.000    0.000    0.000
##    .x1                0.000    0.000    0.000
##    .x2                0.000    0.000    0.000
##    .x3                0.000    0.000    0.000
##    .y1                0.000    0.000    0.000
```

```
##    .y2                0.000    0.000    0.000
##    .y3                0.000    0.000    0.000
##    .y4                0.000    0.000    0.000
##    .y5                0.000    0.000    0.000
##    .y6                0.000    0.000    0.000
##    .y7                0.000    0.000    0.000
##    .y8                0.000    0.000    0.000
##
## Variances:
##                    Estimate   Std.lv  Std.all
##     ind60             1.000    1.000    1.000
##    .dem60             1.000    0.862    0.862
##    .dem65             1.000    0.483    0.483
##    .x1                0.510    0.510    0.510
##    .x2                0.510    0.510    0.510
##    .x3                0.510    0.510    0.510
##    .y1                0.510    0.510    0.473
##    .y2                0.510    0.510    0.473
##    .y3                0.510    0.510    0.473
##    .y4                0.510    0.510    0.473
##    .y5                0.510    0.510    0.335
##    .y6                0.510    0.510    0.335
##    .y7                0.510    0.510    0.335
##    .y8                0.510    0.510    0.335
##
## R-Square:
##                    Estimate
##     dem60             0.138
##     dem65             0.517
##     x1                0.490
##     x2                0.490
##     x3                0.490
##     y1                0.527
##     y2                0.527
##     y3                0.527
##     y4                0.527
##     y5                0.665
##     y6                0.665
##     y7                0.665
##     y8                0.665
```

5.12.1.2 Model-implied covariance and correlation matrix

```
fitted(populationModel_fit)
```

5.12.1.2.1 Model-implied covariance matrix:

```
## $cov
##    x1    x2    x3    y1    y2    y3    y4    y5    y6    y7    y8
## x1 1.000
## x2 0.490 1.000
## x3 0.490 0.490 1.000
## y1 0.196 0.196 0.196 1.078
## y2 0.196 0.196 0.196 0.568 1.078
## y3 0.196 0.196 0.196 0.568 0.568 1.078
## y4 0.196 0.196 0.196 0.568 0.568 0.568 1.078
## y5 0.289 0.289 0.289 0.532 0.532 0.532 0.532 1.525
## y6 0.289 0.289 0.289 0.532 0.532 0.532 0.532 1.015 1.525
## y7 0.289 0.289 0.289 0.532 0.532 0.532 0.532 1.015 1.015 1.525
## y8 0.289 0.289 0.289 0.532 0.532 0.532 0.532 1.015 1.015 1.015 1.525
##
## $mean
## x1 x2 x3 y1 y2 y3 y4 y5 y6 y7 y8
##  0  0  0  0  0  0  0  0  0  0  0
```

```
cov2cor(fitted(populationModel_fit)$cov)
```

5.12.1.2.2 Model-implied correlation matrix

```
##    x1    x2    x3    y1    y2    y3    y4    y5    y6    y7    y8
## x1 1.000
## x2 0.490 1.000
## x3 0.490 0.490 1.000
## y1 0.189 0.189 0.189 1.000
## y2 0.189 0.189 0.189 0.527 1.000
## y3 0.189 0.189 0.189 0.527 0.527 1.000
## y4 0.189 0.189 0.189 0.527 0.527 0.527 1.000
## y5 0.234 0.234 0.234 0.415 0.415 0.415 0.415 1.000
## y6 0.234 0.234 0.234 0.415 0.415 0.415 0.415 0.665 1.000
## y7 0.234 0.234 0.234 0.415 0.415 0.415 0.415 0.665 0.665 1.000
## y8 0.234 0.234 0.234 0.415 0.415 0.415 0.415 0.665 0.665 0.665 1.000
```

5.12.2 Specify analysis model

```
analysisModel_syntax <- '
 #Measurement model factor loadings (free the factor loading of the first indicator)
 ind60 =~ NA*x1 + x2 + x3
 dem60 =~ NA*y1 + y2 + y3 + y4
 dem65 =~ NA*y5 + y6 + y7 + y8

 #Regression paths
 dem60 ~ ind60
 dem65 ~ ind60 + dem60

 #Fix latent means to zero
 ind60 ~ 0
 dem60 ~ 0
 dem65 ~ 0

 #Fix latent variances to one
 ind60 ~~ 1*ind60
 dem60 ~~ 1*dem60
 dem65 ~~ 1*dem65

 #Estimate covariances among latent variables (not necessary because the latent variabl

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 y1 ~~ y1
 y2 ~~ y2
 y3 ~~ y3
 y4 ~~ y4
 y5 ~~ y5
 y6 ~~ y6
 y7 ~~ y7
 y8 ~~ y8

 #Free intercepts of manifest variables
 x1 ~ intx1*1
 x2 ~ intx2*1
 x3 ~ intx3*1
 y1 ~ inty1*1
 y2 ~ inty2*1
 y3 ~ inty3*1
 y4 ~ inty4*1
```

```
 y5 ~ inty5*1
 y6 ~ inty6*1
 y7 ~ inty7*1
 y8 ~ inty8*1
'
```

5.12.3 Specify distribution of data

Specifying the expected distributions of the data variables is an optional step, but it can help give you more realistic estimates of your likely power, especially when the data are non-normally distributed. First, identify the order in which the indicator variables appear in the model, so you can know the order to specify skewness and kurtosis (which I specify in the next section):

```
names(fitted(populationModel_fit)$mean)
```

```
##  [1] "x1" "x2" "x3" "y1" "y2" "y3" "y4" "y5" "y6" "y7" "y8"
```

Specify the skewness and kurtosis of the data variables. In this example, I set the variables $x1$–$x3$ (the first three variables) to have skewness of 1.3 and kurtosis of 1.8, and I set the variables $y1$–$y8$ (the next eight variables) to have a skewness of 2 and a kurtosis of 4.

```
numberOfIndicators <- length(fitted(populationModel_fit)$mean)
```

```
indicatorDistributions <- bindDist(p = numberOfIndicators,
                                   skewness = c(rep(1.3, 3), rep(2, 8)),
                                   kurtosis = c(rep(1.8, 3), rep(4, 8)))
```

5.12.4 Specify extent and type of missing data

5.12.4.1 Specify missingness

Specifying the extent and pattern of missingness is an optional step, but it can help give you more realistic estimates of your likely power, especially when there is extensive missing data and/or the data are not missing completely at random (MCAR). For an example of specifying the extent and pattern of missingness in the context of a Monte Carlo power analysis, see Beaujean [2014]. In this example, I set 10% of values to be missing for variables $x1$, $x2$, and $x3$. I set 15% of values to be missing for variables $y1$–$y8$. I assumed the missingness mechanism to be MCAR. To set missingness to be missing at random (MAR), add a covariate in the missingness formula [see Beaujean, 2014]. I set the model to use

full information maximum likelihood (FIML) estimation to handle missingness. If you set m to a value greater than zero, it will use multiple imputation instead of FIML.

```
percentMissingByVariable <- '
  x1 ~ p(0.10)
  x2 ~ p(0.10)
  x3 ~ p(0.10)
  y1 ~ p(0.15)
  y2 ~ p(0.15)
  y3 ~ p(0.15)
  y4 ~ p(0.15)
  y5 ~ p(0.15)
  y6 ~ p(0.15)
  y7 ~ p(0.15)
  y8 ~ p(0.15)
'
```

```
missingnessModel <- miss(logit = percentMissingByVariable, m = 0)
```

5.12.4.2 Plot of extent of missing data specified

```
plotLogitMiss(percentMissingByVariable)
```

5.12.5 Specify sample sizes and repetitions

Specify the sample sizes to evaluate in the Monte Carlo simulation and the number of repetitions per sample size.

```
sampleSizes <- 50:1000
repetitionsPerSampleSize <- 5
```

5.12.6 Monte Carlo simulation to generate data from the population parameter values

Conduct Monte Carlo simulation with 5 repetitions per sample size (Ns = 50–1000). The `multicore` backend was used for parallel processing. Parallel processing distributes a larger computation task across multiple computing processes or cores, and runs them simultaneously (in parallel) to speed up execution time. If you choose to do processing in serial rather than parallel (by setting

Figure 5.9: Percent missingness specified for each variable

`multicore = FALSE`), you will need to run a special `set.seed()` command, prior to running the `sim()` command, to get reproducible results with those obtained from parallel processing: `set.seed(seedNumber, "L'Ecuyer-CMRG")`, where `seedNumber` is the value used for the seed. **Warning**: this code takes a while to run based on 951 different sample sizes ($1000-50+1$) and 5 repetitions per sample size, for a total of 4755 iterations ($[1000-50+1]$ sample sizes $\times$ 5 repetitions per sample size = 4755 iterations). You can reduce the number of sample sizes and/or repetitions per sample size to be faster.

```
output <- simsem::sim(n = rep(sampleSizes, each = repetitionsPerSampleSize),
                      model = analysisModel_syntax,
                      generate = populationModel,
                      miss = missingnessModel,
                      indDist = indicatorDistributions,
                      lavaanfun = "lavaan",
                      missing = "ML",
                      estimator = "MLR",
                      std.lv = TRUE,
                      seed = 52242,
                      multicore = TRUE)
```

```
## Progress tracker is not available when 'multicore' is TRUE.
```

Return the seed to the normal seed behavior so that future calls to `set.seed()` use the default settings:

```
RNGkind("default", "default", "default")
```

5.12.7 Summary of population model

```
summaryPopulation(output)
```

```
##                   ind60=~x1 ind60=~x2 ind60=~x3 dem60=~y1 dem60=~y2 dem60=~y3
## Population Value 0.7       0.7       0.7       0.7       0.7       0.7
##                   dem60=~y4 dem65=~y5 dem65=~y6 dem65=~y7 dem65=~y8 dem60~ind60
## Population Value 0.7       0.7       0.7       0.7       0.7       0.4
##                   dem65~ind60 dem65~dem60 ind60~1 dem60~1 dem65~1 ind60~~ind60
## Population Value 0.25        0.85        0       0       0       1
##                   dem60~~dem60 dem65~~dem65 x1~~x1 x2~~x2 x3~~x3 y1~~y1 y2~~y2
## Population Value 1            1            0.51   0.51   0.51   0.51   0.51
##                   y3~~y3 y4~~y4 y5~~y5 y6~~y6 y7~~y7 y8~~y8 x1~1 x2~1 x3~1 y1~1
## Population Value 0.51   0.51   0.51   0.51   0.51   0.51   0    0    0    0
##                   y2~1 y3~1 y4~1 y5~1 y6~1 y7~1 y8~1
## Population Value 0    0    0    0    0    0    0
```

5.12.8 Summary of simulated data

```
summary(output)
```

```
## RESULT OBJECT
## Model Type
## [1] "lavaan"
## ========= Fit Indices Cutoffs ============
##      N chisq.scaled       aic       bic rmsea.scaled cfi.scaled tli.scaled
## 1   50       78.388  1432.886  1537.280        0.095          1      1.021
## 2  288       72.618  7627.217  7753.099        0.074          1      1.017
## 3  525       66.872 13795.522 13942.802        0.053          1      1.013
## 4  762       61.126 19963.827 20132.505        0.032          1      1.009
## 5 1000       55.356 26158.159 26348.324        0.011          1      1.005
##    srmr
## 1 0.083
```

```
## 2 0.067
## 3 0.051
## 4 0.035
## 5 0.019
## ========= Parameter Estimates and Standard Errors ============
##             Estimate Average Estimate SD Average SE Power (Not equal 0) Std Est
## ind60=~x1              0.700       0.078      0.071                0.998   0.700
## ind60=~x2              0.701       0.078      0.070                0.999   0.701
## ind60=~x3              0.701       0.080      0.071                0.998   0.701
## dem60=~y1              0.695       0.087      0.076                0.999   0.707
## dem60=~y2              0.695       0.087      0.077                0.998   0.707
## dem60=~y3              0.693       0.086      0.077                0.998   0.705
## dem60=~y4              0.695       0.087      0.076                0.999   0.707
## dem65=~y5              0.689       0.081      0.072                0.999   0.794
## dem65=~y6              0.688       0.081      0.072                0.999   0.794
## dem65=~y7              0.688       0.083      0.071                0.999   0.793
## dem65=~y8              0.689       0.082      0.071                0.999   0.795
## dem60~ind60            0.405       0.107      0.092                0.955   0.371
## dem65~ind60            0.255       0.126      0.107                0.744   0.174
## dem65~dem60            0.867       0.176      0.144                0.988   0.635
## x1~~x1                 0.504       0.086      0.077                0.986   0.507
## x2~~x2                 0.503       0.085      0.077                0.987   0.505
## x3~~x3                 0.503       0.089      0.077                0.986   0.505
## y1~~y1                 0.559       0.099      0.084                0.995   0.497
## y2~~y2                 0.560       0.095      0.085                0.995   0.498
## y3~~y3                 0.560       0.098      0.085                0.997   0.499
## y4~~y4                 0.559       0.096      0.085                0.995   0.498
## y5~~y5                 0.584       0.106      0.091                0.996   0.368
## y6~~y6                 0.584       0.107      0.091                0.996   0.368
## y7~~y7                 0.586       0.107      0.090                0.996   0.369
## y8~~y8                 0.581       0.109      0.090                0.994   0.367
## intx1                  0.001       0.058      0.053                0.050  -0.001
## intx2                  0.001       0.059      0.054                0.049  -0.001
## intx3                  0.001       0.057      0.053                0.052  -0.001
## inty1                  0.001       0.063      0.058                0.059  -0.002
## inty2                  0.001       0.063      0.058                0.050  -0.002
## inty3                  0.000       0.063      0.058                0.055  -0.004
## inty4                  0.000       0.064      0.058                0.059  -0.003
## inty5                  0.001       0.075      0.068                0.058  -0.002
## inty6                  0.000       0.076      0.068                0.056  -0.003
## inty7                  0.001       0.076      0.068                0.055  -0.002
## inty8                  0.000       0.074      0.068                0.059  -0.003
##             Std Est SD Std Ave SE r_coef.n r_se.n
## ind60=~x1        0.058      0.052   -0.004 -0.738
## ind60=~x2        0.058      0.052   -0.007 -0.757
## ind60=~x3        0.059      0.052   -0.006 -0.739
```

```
## dem60=~y1        0.055      0.048    0.061 -0.746
## dem60=~y2        0.055      0.048    0.071 -0.750
## dem60=~y3        0.056      0.049    0.065 -0.683
## dem60=~y4        0.055      0.048    0.058 -0.755
## dem65=~y5        0.042      0.035    0.113 -0.748
## dem65=~y6        0.040      0.035    0.099 -0.739
## dem65=~y7        0.040      0.035    0.102 -0.753
## dem65=~y8        0.041      0.035    0.105 -0.750
## dem60~ind60      0.080      0.072   -0.051 -0.701
## dem65~ind60      0.080      0.070   -0.032 -0.690
## dem65~dem60      0.074      0.064   -0.073 -0.638
## x1~~x1           0.081      0.072    0.074 -0.657
## x2~~x2           0.080      0.072    0.084 -0.681
## x3~~x3           0.083      0.072    0.084 -0.665
## y1~~y1           0.077      0.067    0.088 -0.645
## y2~~y2           0.075      0.067    0.046 -0.657
## y3~~y3           0.078      0.068    0.070 -0.506
## y4~~y4           0.075      0.067    0.064 -0.646
## y5~~y5           0.065      0.055    0.043 -0.666
## y6~~y6           0.062      0.056    0.055 -0.630
## y7~~y7           0.063      0.056    0.053 -0.659
## y8~~y8           0.064      0.055    0.094 -0.597
## intx1            0.060      0.054   -0.010 -0.846
## intx2            0.060      0.054   -0.002 -0.846
## intx3            0.059      0.054   -0.007 -0.844
## inty1            0.062      0.055    0.000 -0.840
## inty2            0.062      0.055   -0.009 -0.832
## inty3            0.061      0.055    0.004 -0.837
## inty4            0.061      0.055   -0.016 -0.837
## inty5            0.061      0.054   -0.011 -0.842
## inty6            0.062      0.054    0.008 -0.843
## inty7            0.062      0.054   -0.001 -0.839
## inty8            0.060      0.054    0.024 -0.843
## ========= Correlation between Fit Indices ============
##              chisq.scaled    aic    bic rmsea.scaled cfi.scaled tli.scaled
## chisq.scaled        1.000 -0.320 -0.320        0.865     -0.725     -0.772
## aic                -0.320  1.000  1.000       -0.536      0.467      0.437
## bic                -0.320  1.000  1.000       -0.536      0.467      0.438
## rmsea.scaled        0.865 -0.536 -0.536        1.000     -0.915     -0.929
## cfi.scaled         -0.725  0.467  0.467       -0.915      1.000      0.989
## tli.scaled         -0.772  0.437  0.438       -0.929      0.989      1.000
## srmr                0.593 -0.796 -0.796        0.829     -0.796     -0.780
## n                  -0.319  0.999  0.999       -0.536      0.467      0.437
##                 srmr      n
## chisq.scaled  0.593 -0.319
## aic          -0.796  0.999
```

```
## bic          -0.796  0.999
## rmsea.scaled  0.829 -0.536
## cfi.scaled   -0.796  0.467
## tli.scaled   -0.780  0.437
## srmr          1.000 -0.796
## n            -0.796  1.000
## ================== Replications =====================
## Number of replications = 4755
## Number of converged replications = 4741
## Number of nonconverged replications:
##    1. Nonconvergent Results = 2
##    2. Nonconvergent results from multiple imputation = 0
##    3. At least one SE were negative or NA = 0
##    4. Nonpositive-definite latent or observed (residual) covariance matrix
##       (e.g., Heywood case or linear dependency) = 12
## NOTE: The sample size is varying.
## NOTE: The data generation model is not the same as the analysis  model. See the summary of the
```

5.12.9 Summary of parameters

```
summaryParam(output, alpha = .05, detail = TRUE)
```

```
##              Estimate Average Estimate SD Average SE Power (Not equal 0)
## ind60=~x1       6.995046e-01  0.07831765 0.07051019          0.99831685
## ind60=~x2       7.010985e-01  0.07781324 0.07039089          0.99894803
## ind60=~x3       7.008020e-01  0.08026090 0.07065728          0.99831685
## dem60=~y1       6.946542e-01  0.08708161 0.07621555          0.99894803
## dem60=~y2       6.952848e-01  0.08719297 0.07662702          0.99810646
## dem60=~y3       6.927036e-01  0.08647365 0.07665526          0.99831685
## dem60=~y4       6.945686e-01  0.08680501 0.07622177          0.99852725
## dem65=~y5       6.894388e-01  0.08110963 0.07164416          0.99894803
## dem65=~y6       6.881543e-01  0.08139422 0.07152037          0.99894803
## dem65=~y7       6.884663e-01  0.08302549 0.07134471          0.99936882
## dem65=~y8       6.887302e-01  0.08166614 0.07132251          0.99852725
## dem60~ind60     4.053393e-01  0.10655382 0.09249689          0.95497580
## dem65~ind60     2.549976e-01  0.12604578 0.10730848          0.74353040
## dem65~dem60     8.671177e-01  0.17582970 0.14408669          0.98779718
## x1~~x1          5.043740e-01  0.08618636 0.07725451          0.98632443
## x2~~x2          5.028879e-01  0.08529269 0.07706992          0.98716600
## x3~~x3          5.027216e-01  0.08877345 0.07734806          0.98569325
## y1~~y1          5.587050e-01  0.09947124 0.08414601          0.99495056
## y2~~y2          5.600039e-01  0.09527433 0.08499985          0.99537134
## y3~~y3          5.602746e-01  0.09780868 0.08510295          0.99705449
```

```
## y4~~y4          5.591282e-01  0.09600897 0.08450029          0.99516095
## y5~~y5          5.842213e-01  0.10560732 0.09052838          0.99579213
## y6~~y6          5.837376e-01  0.10691459 0.09077895          0.99621292
## y7~~y7          5.858242e-01  0.10714940 0.09042423          0.99558174
## y8~~y8          5.808555e-01  0.10916596 0.08982873          0.99389859
## intx1           1.118970e-03  0.05837549 0.05349289          0.05028403
## intx2           7.708628e-04  0.05901364 0.05350565          0.04944246
## intx3           7.593853e-04  0.05745666 0.05348113          0.05196718
## inty1           1.153517e-03  0.06296734 0.05761141          0.05933095
## inty2           1.182127e-03  0.06343658 0.05778552          0.05028403
## inty3          -3.745732e-04  0.06252014 0.05759813          0.05470229
## inty4           4.971755e-04  0.06363476 0.05766412          0.05891016
## inty5           1.359816e-03  0.07473422 0.06787058          0.05764780
## inty6          -8.887317e-05  0.07585397 0.06778113          0.05638544
## inty7           1.087680e-03  0.07566286 0.06784423          0.05491269
## inty8          -2.287142e-04  0.07352998 0.06769919          0.05891016
##                  Std Est Std Est SD Std Ave SE      r_coef.n     r_se.n
## ind60=~x1    0.699788355 0.05796979 0.05176328 -0.0042113427 -0.7382679
## ind60=~x2    0.701141366 0.05790258 0.05167122 -0.0065663380 -0.7566669
## ind60=~x3    0.700942543 0.05886666 0.05181490 -0.0055761551 -0.7387659
## dem60=~y1    0.707018149 0.05478515 0.04807984  0.0607198608 -0.7461817
## dem60=~y2    0.706759328 0.05468020 0.04835186  0.0711570174 -0.7502419
## dem60=~y3    0.705497010 0.05567377 0.04865337  0.0648567288 -0.6829081
## dem60=~y4    0.706678866 0.05485288 0.04806475  0.0581146512 -0.7547719
## dem65=~y5    0.794092969 0.04166018 0.03527713  0.1128538239 -0.7477465
## dem65=~y6    0.793845176 0.03953565 0.03538540  0.0990125055 -0.7389075
## dem65=~y7    0.793429467 0.04036991 0.03532975  0.1017185517 -0.7533534
## dem65=~y8    0.794807868 0.04103815 0.03510403  0.1046098326 -0.7496666
## dem60~ind60  0.371187383 0.08013233 0.07170508 -0.0506714341 -0.7006954
## dem65~ind60  0.174237735 0.08021256 0.07047968 -0.0318963880 -0.6897060
## dem65~dem60  0.635392323 0.07429025 0.06372148 -0.0726902304 -0.6384646
## x1~~x1       0.506936468 0.08125511 0.07210708  0.0736197399 -0.6571250
## x2~~x2       0.505048781 0.08048378 0.07215000  0.0843496805 -0.6811898
## x3~~x3       0.505214998 0.08290343 0.07232567  0.0843158989 -0.6645077
## y1~~y1       0.497124556 0.07668553 0.06724601  0.0876441250 -0.6454158
## y2~~y2       0.497501956 0.07503660 0.06743666  0.0458745760 -0.6572096
## y3~~y3       0.499175052 0.07801820 0.06828680  0.0700029027 -0.5061744
## y4~~y4       0.497596775 0.07547646 0.06707996  0.0635434661 -0.6458635
## y5~~y5       0.367681151 0.06487939 0.05546578  0.0431121762 -0.6660183
## y6~~y6       0.368247099 0.06202388 0.05570378  0.0547620515 -0.6301391
## y7~~y7       0.368840294 0.06266518 0.05554067  0.0526663472 -0.6592553
## y8~~y8       0.366596677 0.06407565 0.05531794  0.0942312526 -0.5969828
## intx1       -0.001077139 0.05975028 0.05387350 -0.0095567855 -0.8460925
## intx2       -0.001449168 0.05979254 0.05389945 -0.0015056093 -0.8455671
## intx3       -0.001479491 0.05872492 0.05382462 -0.0074360403 -0.8436604
## inty1       -0.002220506 0.06172054 0.05483658 -0.0003624511 -0.8400702
```

```
## inty2       -0.002185112 0.06178022 0.05482292 -0.0092574922 -0.8316537
## inty3       -0.003579023 0.06081369 0.05485187  0.0037957856 -0.8370780
## inty4       -0.002800249 0.06146574 0.05479569 -0.0159764857 -0.8370183
## inty5       -0.002139593 0.06073543 0.05420695 -0.0112168547 -0.8419107
## inty6       -0.003339910 0.06178721 0.05431693  0.0081455820 -0.8431144
## inty7       -0.002490286 0.06215410 0.05421040 -0.0006114834 -0.8390720
## inty8       -0.003403516 0.06044950 0.05429265  0.0244666015 -0.8425943
```

5.12.10 Time to completion

```
summaryTime(output)
```

```
## ============ Wall Time ============
##   1. Error Checking and setting up data-generation and analysis template:        0.000
##   2. Set combinations of n, pmMCAR, and pmMAR:                                   0.000
##   3. Setting up simulation conditions for each replication:                       0.130
##   4. Total time elapsed running all replications:                              1795.830
##   5. Combining outputs from different replications:                               0.150
##
## ============ Average Time in Each Replication ============
##   1. Data Generation:                                  0.057
##   2. Impose Missing Values:                            0.015
##   3. User-defined Data-Transformation Function:        0.000
##   4. Main Data Analysis:                               2.694
##   5. Extracting Outputs:                               0.205
##
## ============ Summary ============
##   Start Time:                    2021-05-31 13:58:01
##   End Time:                      2021-05-31 14:27:57
##   Wall (Actual) Time:              1796.110
##   System (Processors) Time:       14122.630
##   Units:                         seconds
```

5.12.11 Cutoffs of fit indices

5.12.11.1 Plot of cutoffs of fit indices

At $\alpha = .05$

```
plotCutoff(output, alpha = .05)
```

Figure 5.10: Plot of cutoffs of fit indices from Monte Carlo power analysis

5.12.11.2 Cutoffs of fit indices at particular sample size

At $N = 200$, $\alpha = .05$

```
getCutoff(output, alpha = .05, nVal = 200)
```

```
##   chisq.scaled      aic      bic rmsea.scaled cfi.scaled tli.scaled       srmr
## 1     74.75107 5336.876 5454.813     0.081699          1   1.018547 0.07298416
```

5.12.12 Statistical Power

5.12.12.1 Plot of power to detect various parameters as a function of sample size

At $\alpha = .05$; dashed horizontal line represents power $(1 - \beta)$ of .8

```
par(mfrow = c(1,2))

plotPower(output,
```

```
          powerParam = "dem65~dem60",
          alpha = .05)

abline(h = 0.8, lwd = 2, lty = 2)

plotPower(output,
          powerParam = "dem65~ind60",
          alpha = .05)

abline(h = 0.8, lwd = 2, lty = 2)
```

Figure 5.11: Plot of power to detect various parameters as a function of sample size, from Monte Carlo power analysis

5.12.12.2 Sample size needed to detect a given parameter

At power $(1 - \beta) = .80$, $\alpha = .05$

```
powerEstimates <- getPower(output, alpha = .05)
findPower(powerEstimates, iv = "N", power = .80)
```

```
##   ind60=~x1   ind60=~x2   ind60=~x3   dem60=~y1   dem60=~y2   dem60=~y3
##          50          50          50          50          50          50
##   dem60=~y4   dem65=~y5   dem65=~y6   dem65=~y7   dem65=~y8 dem60~ind60
##          50          50          50          50          50         133
## dem65~ind60 dem65~dem60      x1~~x1      x2~~x2      x3~~x3      y1~~y1
##         486          73          74          73          75          56
##      y2~~y2      y3~~y3      y4~~y4      y5~~y5      y6~~y6      y7~~y7
##          52          50          57          56          50          50
##      y8~~y8       intx1       intx2       intx3       inty1       inty2
##          57          NA          NA          NA          NA          NA
##       inty3       inty4       inty5       inty6       inty7       inty8
##          NA          NA          NA          NA          NA          NA
```

5.12.12.3 Power to detect each parameter at a given sample size

At $N = 200$, $\alpha = .05$

```
getPower(output, alpha = .05, nVal = 200)
```

```
##      iv.N ind60=~x1 ind60=~x2 ind60=~x3 dem60=~y1 dem60=~y2 dem60=~y3 dem60=~y4
## [1,]  200 0.9999973 0.9999971 0.9999658         1         1 0.9999997 0.9999999
##      dem65=~y5 dem65=~y6 dem65=~y7 dem65=~y8 dem60~ind60 dem65~ind60
## [1,] 0.9999968 0.9999971 0.9999944 0.9999982   0.9492042   0.4263835
##      dem65~dem60   x1~~x1    x2~~x2    x3~~x3    y1~~y1    y2~~y2    y3~~y3
## [1,]   0.9997952 0.996662 0.9985341 0.9957747 0.9999801 0.9999269 0.9998382
##         y4~~y4    y5~~y5    y6~~y6    y7~~y7    y8~~y8      intx1      intx2
## [1,] 0.9999922 0.9999984 0.9998922 0.9998522 0.9998712 0.05234645 0.06083614
##           intx3      inty1      inty2      inty3      inty4      inty5
## [1,] 0.05105651 0.05562025 0.05364667 0.06271751 0.06154374 0.06058061
##           inty6      inty7      inty8
## [1,] 0.05469918 0.05543663 0.05253452
```

5.13 Generalizability Theory

There are also structural equation modeling approaches for performing generalizability theory analyses. The reader is referred to examples by Vispoel et al. [2018].

5.14 Conclusion

Structural equation modeling is an advanced modeling approach that allows estimating latent variables as the common variance from multiple measures.

SEM holds promise to account for measurement error and method biases, which allows get more accurate estimates of constructs, people's standing on constructs (i.e., individual differences), and associations between constructs.

5.15 Exercises

5.15.1 Questions

Note: several of the following questions use data from the Children of the National Longitudinal Survey of Youth Survey (CNLSY). The CNLSY is a publicly available longitudinal dataset provided by the Bureau of Labor Statistics (https://www.bls.gov/nls/nlsy79-children.htm#topical-guide). The CNLSY data file for these exercises is located on the course's page of the Open Science Framework (https://osf.io/3pwza). Children's behavior problems were rated in 1988 (time 1: T1) and then again in 1990 (time 2: T2) on the Behavior Problems Index (BPI). Below are the items corresponding to the Antisocial subscale of the BPI: 1) cheats or tells lies 2) bullies or is cruel/mean to others 3) does not seem to feel sorry after misbehaving 4) breaks things deliberately 5) is disobedient at school 6) has trouble getting along with teachers 7) has sudden changes in mood or feeling

1. Fit a confirmatory factor analysis model to the seven items of the Antisocial subscale of the Behavior Problems Index at T1. Set the first indicator to be the referent indicator (to set the scale of the latent factor) by setting its loading to one. Allow the factor loadings of the other indicators to be freely estimated. Set the mean (intercept) of the latent factor to be zero. Do not allow the residuals to be correlated. Use full information maximum likelihood (FIML) to account for missing data. Use robust standard errors to account for non-normally distributed data.

    a. This is an over-simplification, but for now let us assume a model fits "well" if CFI $\geq$ .95, RMSEA $<$ .08, and SRMR $<$ .08 [Schreiber et al., 2006]. Did the model fit well? What does this indicate?
    b. Examine the modification indices. Which modification would result in the greatest improvement in model fit? Why do you think this modification would improve model fit?

2. Fit the modified confirmatory factor analysis model to make the suggested revision you identified in `1b`.

    a. Provide a figure of the model with standardized coefficients.
    b. The modified model and the original model are considered "nested" models. The original model is nested within the modified model because the modified model includes all of the terms of the original

model along with additional terms. Model fit of nested models can be directly compared with a chi-square difference test. Did the modified model fit better than the original model?

c. Did the modified model fit well? What does this indicate? Which item is most strongly with the latent factor? Which item is most weakly associated with the latent factor?
d. What is the estimate of internal consistency reliability of the items, based on coefficient omega?

3. Fit a confirmatory factor analysis model to the seven items of the Antisocial subscale of the Behavior Problems Index at both T1 and T2 simultaneously in the same model. Allow the items at T1 to load onto a different factor than the items at T2 (i.e., a two-factor model—one antisocial at each time point). Set the scale of the latent factors by standardizing the latent factors—set their means to one and their variances to zero. This allows you to freely estimate the factor loadings of all items (instead of setting a reference indicator). Estimate the covariance between the two latent factors. Treat exogenous covariates as random variables (whose means, variances, and covariances are estimated) by specifying `fixed.x = FALSE`. Use full information maximum likelihood (FIML) to account for missing data. Use robust standard errors to account for non-normally distributed data. Apply the same modification you noted in 1b above to each factor.

   a. Because the two latent factors are standardized, the "covariance" path between the two latent factors represents a correlation. What is the correlation between the latent factors? What is the correlation between the sum scores (`bpi_antisocialT1Sum`, `bpi_antisocialT2Sum`)? Which is greater and why?
   b. Change the covariance path to a regression path from the latent factor at T1 predicting the latent factor at T2. Also include the sum score of anxious/depressed symptoms at T1 (`bpi_anxiousDepressedSum`) as a predictor of antisocial behavior at T2. Do anxious/depressed symptoms at T1 predict antisocial behavior at T2 controlling for prior levels of antisocial behavior at T1? Interpret the findings.

4. You plan to conduct a study that would examine whether stress and harsh parenting predict children's antisocial behavior. Your hypothesis is that stress and harsh parenting both lead to children's antisocial behavior. You would like to apply for a grant to test these hypotheses, but you first want to know what sample size you would need to have adequate power to detect the hypothesized effects. Because you read this book, you remember that measurement error attenuates the associations you would observe, which would make it less likely that you would be able to detect the true effect (if there truly is an effect). As a result, you plan to assess each construct with multiple measurement methods/measures. You plan to model each

construct with a latent variable in a structural equation modeling framework to account for measurement error and disattenuate the associations, which will make the associations more closely approximate the true effect and will make it more likely that you will detect the effect if it exists. You plan to assess stress with three methods (self-report, friend report, cortisol), harsh parenting with four methods (parents' self-report, spousal report, child report, observation), and children's antisocial behavior with four methods (parent report, teacher report, child report, observation).

You conduct a power analysis with the following assumptions that you made based on theory and prior empirical research:

- The factor loading for each measure on its latent variable is .75
- Stress influences children's antisocial behavior with a regression coefficient of .20
- Harsh parenting influences children's antisocial behavior with a regression coefficient of .45
- Stress influences harsh parenting with a regression coefficient of .4

Set the intercepts of the indicators to zero. Set the scale of the latent factors by standardizing the latent factors—set their means to one and their variances to zero. Don't estimate correlated errors. You expect each measure to show 10% missingness, and for missingness to be completely at random (MCAR). Use full information maximum likelihood (FIML) to handle missing values. As is common with measures in clinical psychology, you expect each measure to be positively skewed with a skewness of 2.5 and leptokurtic with a kurtosis of 5. Use robust standard errors to account for non-normally distributed data. Using a seed of 52242, an alpha level of .05, and 5 repetitions per sample size, in the `simsem` package:

a. What sample size would you need to have adequate power to detect the effect of stress on antisocial behavior and the effect of harsh parenting on antisocial behavior?
b. Due to financial and time constraints of the grant, you are only able to collect a sample size of 150. What power would you have to detect the effect of stress on antisocial behavior and the effect of harsh parenting on antisocial behavior?
c. Your study finds that neither stress nor harsh parenting predicts antisocial behavior. How would you interpret each of these findings?
d. Re-run the power analysis with normally distributed values (skewness = 0, kurtosis = 0). With a sample size of 150, did power to detect the hypothesized effects increase or decrease? What does this indicate?
e. Re-run the power analysis using multiple imputation instead of FIML (and normally distributed values). Use five imputations. With a sample size of 150, did power to detect the hypothesized effects increase or decrease? What does this indicate?

5.15.2 Answers

1. a. The model did not fit well according to CFI (0.88) and RMSEA (0.09). SRMR (0.05) was acceptable. The poor model fit indicates that it is unlikely that the causal process described by the hypothesized model gave rise to the observed data.
   b. The modification that would result in the greatest model fit according to the modification indices is to allow indicators 5 and 6 to be correlated. These indicators reflect "disobedience at school" and "trouble getting along with teachers," respectively. It is likely that allowing these two residuals to correlate would improve model fit because they both assess children's behavior in the school context, and so they would continue to be associated with each other even after accounting for variance from the latent factor.

2. a.

2. b. The modified model ($\chi^2[df = 13] = 33.68$) fit significantly better than the original model ($\chi^2[df = 14] = 391.38$) according to a chi-square difference test ($\Delta\chi^2[df = 1] = 833.50, p < .001$).
   c. The model fit well according to CFI (0.9923114), RMSEA (0.02416), and SRMR (0.0133991). This indicates that there is evidence that one factor may do a good job of explaining the covariance among the indicators, especially when allowing the residuals of items 5 and 6 to correlate. The item that shows the strongest association with the

latent factor is item 2 ("bullies or is cruel/mean to others": standardized factor loading = .63). The item that shows the weakest association with the latent factor is item 7 ("sudden changes in mood or feeling": standardized factor loading = .40). Thus, meanness seems more core to the construct of antisocial behavior compared to sudden mood changes.

d. The estimate of internal consistency reliability of items, based on coefficient omega (ω), is .68.

3. a. The correlation between the latent factor at T1 and T2 is $\phi = .76$. The correlation between the sum score at T1 and T2 is $r = .50$. This indicates that the correlation of individual differences across time (rank-order stability) is stronger for the latent factor than for the sum scores. This is likely because the latent factors account for measurement error whereas the sum scores do not, and associations are attenuated due to measurement error. Thus, the association of the latent factor at T1 and T2 likely more accurately reflects the "true" cross-time association of the construct (compared to the association of the sum scores at T1 and T2).

   b. Yes, anxious/depressed symptoms significantly predicted antisocial behavior at T2 while controlling for prior levels of antisocial behavior ($B = 0.11, \beta = .11, SE = 0.02, p < .001$). That is, anxious/depressed symptoms predicted relative (rank-order) changes in antisocial behavior from T1 to T2. This suggests that anxiety/depression may be a pathway to antisocial behavior for some children. Because the data come from an observational design, however, we cannot infer causality. For instance, the association could owe to the opposite direction of effect (antisocial behavior could lead to anxiety/depression) or to a third variable (e.g., victimization could lead to both antisocial behavior and anxiety/depression; i.e., antisocial behavior and anxiety/depression could share a common cause).

4. a. You would need a sample size of 667 to detect the effect of stress on children's antisocial behavior. You would need a sample size of 140 to detect the effect of harsh parenting on children's antisocial behavior.

   b. You would have a power of .26 to detect the effect of stress on children's antisocial behavior. You would have a power of .84 to detect the effect of harsh parenting on children's antisocial behavior.

   c. Because your study was well-powered to detect the effect of harsh parenting (power = .84) and you found no statistically significant association between harsh parenting and children's antisocial behavior, it suggests that harsh parenting did not influence antisocial behavior in this sample (at least not with a large enough effect size to be practically significant). Because your study was under-powered to detect the effect of stress (power = .26) and you found no statistically significant association between stress and children's antisocial behavior, we do not know whether you did not detect an association

because (a) stress did not influence antisocial behavior in this sample (i.e., your hypotheses were incorrect), or (b) there was an effect of stress (i.e., your hypotheses were correct), but your sample size was too small and/or your measurements were too unreliable to detect the effect given the effect size.

d. Power to detect the hypothesized effects increased when the data were normally distributed (compared to when the data were non-normally distributed). You would have a power of .38 to detect the effect of stress on children's antisocial behavior. You would have a power of .99 to detect the effect of harsh parenting on children's antisocial behavior. This indicates that statistical power tends to be lower when data are non-normally distributed (compared to when data are normally distributed).

e. Power to detect the hypothesized effects decreased when using multiple imputation compared to FIML. You would have a power of .33 to detect the effect of stress on children's antisocial behavior. You would have a power of .98 to detect the effect of harsh parenting on children's antisocial behavior. This is consistent with prior findings that statistical power with multiple imputation is lower than with FIML unless the number of imputations is large [Graham et al., 2007].

Chapter 6

Item Response Theory

In the chapter on reliability, we introduced classical test theory. Classical test theory is a measurement theory of how test scores relate to a construct. Classical test theory provides a way to estimate the relation between the measure (or item) and the construct. For instance, with a classical test theory approach, to estimate the relation between an item and the construct, you would compute an item–total correlation. An item–total correlation is the correlation of an item with the total score on the measure (e.g., sum score). The item–total correlation approximates the relation between the item and the construct. However, it is a crude estimate. And there are many other ways to characterize the relation between an item and a construct. On such way is with item response theory (IRT).

6.1 Overview of IRT

Unlike classical test theory, which is a measurement theory that of how test scores relate to a construct, IRT is a measurement theory that describes how an *item* is related to a construct. For instance, given a particular person's level on the construct, what is their chance of answering "TRUE" on a particular item?

In IRT, we estimate a person's construct score (i.e., level on the construct) based on their item responses. The construct is estimated as a latent factor that represents the common variance among all items as in structural equation modeling or confirmatory factor analysis. The person's level on the construct is called theta (θ). When dealing with performance-based tests, theta is sometimes called "ability".

6.1.1 Item Characteristic Curve

We IRT, we can plot an *item characteristic curve* (ICC). The ICC is a plot of the model-derived probability of a symptom being present (or a correct response) as a function of a person's standing on a latent continuum. For instance, we can create empirical ICCs that can take any shape (see Figure 6.1).

In a model-implied ICC, we fit a logistic (sigmoid) curve to each item's probability of a symptoms being present as a function of a person's level on the latent construct. The model-implied ICCs for the same 10 items are depicted in Figure 6.2.

An ICC provides more information than an item–total correlation. Visually, we can see the utility of various items by looking at the items' ICC plots. For instance, consider what might be a useless item for diagnostic purposes. For a particular item, among those with a low total score (level on the construct), 90% respond with "TRUE" to the item, whereas among everyone else, 100% response with "TRUE" (see Figure 6.3). This item has a ceiling effect and provides only a little information about who would be considered above clinical threshold for a disorder. So, the item is not very clinically useful.

Now, consider a different item. For those with a low level on the construct, 0% respond with "TRUE", so it has a floor effect and tells us nothing about the lower end of the construct. But, for those with a higher level on the construct, 30% respond true (see Figure 6.4). So, the item tells us something about the higher end of the distribution. Thus, an ICC allows us to immediately tell the utility of items.

6.1.2 Parameters

We can estimate up to four parameters in an IRT model and can glean up to four key pieces of information from an item's ICC:

1. difficulty (severity)
2. discrimination
3. guessing
4. inattention/careless errors

6.1.2.1 Difficulty (Severity)

The item's *difficulty* parameter is the item's location on the latent construct. It is quantified by the intercept, i.e., the location on the x-axis of the inflection point of the ICC. In a 1- or 2-parameter model, the inflection point is where 50% of the sample endorses the item (or gets the item correct), that is, the point on the x-axis where the ICC crosses .5. Some items are more useful at the

Figure 6.1: Empirical item characteristic curves of the probability of endorsement of a given item as a function of the person's sum score.

Figure 6.2: Item characteristic curves of the probability of endorsement of a given item as a function of the person's level on the latent construct.

Figure 6.3: Item characteristic curve of an item with a ceiling effect that is not diagnostically useful.

Figure 6.4: Item characteristic curve of an item a floor effect that is diagnostically useful.

higher levels of the construct, whereas other items are more useful at the lower levels of the construct. See Figure 6.1.2.1 for an example of an item with a low difficulty and an item with a high difficulty.

\begin{figure}
\caption{Item characteristic curves of an item with low difficulty versus high difficulty. The dashed horizontal line indicates a probability of item endorsement of .50. The dashed vertical line is the item difficulty, i.e., the person's level on the construct (the location on the x-axis) at the inflection point of the item characteristic curve. In a two-parameter logistic model, the inflection point corresponds to the probability of item endorsement is 50%. Thus, in a two-parameter logistic model, the difficulty of an item is the person's level on the construct where the probability of endorsing the item is 50%.} \end{figure}

When dealing with a measure of clinical symptoms (e.g., depression), the difficulty parameter is sometimes called severity, because symptoms that are endorsed less frequently tend to be more severe [e.g., suicidal behavior; Krueger et al. [2004]]. One way of thinking about the severity parameter of an item is: "How severe does your psychopathology have to be for half of people to endorse the symptom?"

When dealing with a measure of performance, aptitude, or intelligence, the parameter would be more likely to be called difficulty: "How high does your ability have to be for half of people to pass the item?" An item with a low difficulty would be considered easy, because even people with a low ability tend to pass the item. An item with a high difficulty would be considered difficult, because only people with a high ability tend to pass the item.

6.1.2.2 Discrimination

The item's *discrimination* parameter is how well the item can distinguish between those who ware higher versus lower on the construct, that is, how strongly the item is correlated with the construct (i.e., the latent factor) . It is quantified by the slope of the ICC, i.e., the steepness of the line at its steepest point. The slope reflects the inverse of how much range of construct levels it would take to flip 50/50 whether a person is likely to pass or fail an item.

Some items have ICCs that go up fast (have a steep slope). These items provide a fine distinction between people with lower versus higher levels on the construct and therefore have high discrimination. Some items go up gradually (less steep slope), so it provides less precision and information, and has a low discrimination. See Figure 6.5 for an example of an item with a low discrimination and an item with a high discrimination.

Figure 6.5: Item characteristic curves of an item with low discrimination versus high discrimination. The discrimination of an item is the slope of the line at its inflection point.

6.1.2.3 Guessing

The item's *guessing* parameter is reflected by the lower asymptote of the ICC. If the item has a lower asymptote above zero, it suggests that the probability

of getting the item correct (or endorsing the item) never reaches zero, for any level of the construct. On an educational test, this could correspond to the person's likelihood of being able to answer the item correctly by chance just by guessing. For example, for a 4-option multiple choice test, a respondent would be expected to get a given item correct 25% of the time just by guessing. See Figure 6.1.2.3 for an example of an item from a true/false exam and Figure 6.1.2.3 for an example of an item from a 4-option multiple choice exam.

\begin{figure}
\caption{Item characteristic curve of an item from a true/false exam, where test takers get the item correct at least 50% of the time.} \end{figure}

\begin{figure}
\caption{Item characteristic curve of an item from a 4-option multiple choice exam, where test takers get the item correct at least 25% of the time.}
\end{figure}

6.1.2.4 Inattention/Careless Errors

The item's *inattention* (or *careless error*) parameter is the reflected by the upper asymptote of the ICC. If the item has an upper asymptote below one, it suggests that the probability of getting the item correct (or endorsing the item) never reaches one, for any level on the construct. See Figure 6.6 for an example of an item whose probability of endorsement (or getting it correct) next exceeds .85.

6.1.3 Models

IRT models can be fit that estimate one or more of these four item parameters.

6.1.3.1 1-Parameter (Rasch)

A 1-parameter logistic IRT model, also called a Rasch model, estimates the item difficult parameter, and holds everything else fixed across items (see Figure 6.7). One-parameter models are common and are the easiest to fit. However, they make fairly strict assumptions. They assume that items have the same discrimination.

Figure 6.6: Item characteristic curve of an item where the probability of getting an item correct never exceeds .85.

Figure 6.7: One-parameter logistic model in item response theory.

A 1-parameter logistic model is only valid if there are not crossing of lines in empirical ICCs (see Figure 6.8).

6.1.3.2 2-Parameter

A 2-parameter logistic IRT model estimates item difficulty and discrimination, and it holds the asymptotes fixed across items (see Figure 6.9). Two-parameter models are also common.

6.1.3.3 3-Parameter

A 3-parameter logistic IRT model estimates item difficulty, discrimination, and guessing (lower asymptote), and it holds the upper asymptote fixed across items (see Figure 6.10). This model would provide information about where an item drops out. Three-parameter models are less common to estimate because it adds considerable computational complexity and requires a large sample size, and the guessing parameter is often not as important as difficulty and discrimination. Nevertheless, 3-parameter models are sometimes estimated in the education literature to account for getting items correct by random guessing.

Figure 6.8: Empirical item characteristic curves of the probability of endorsement of a given item as a function of the person's sum score. The empirical item characteristic curves of these items do not cross each other.

Figure 6.9: Two-parameter logistic model in item response theory.

6.1.3.4 4-Parameter

A 4-parameter logistic IRT model estimates item difficulty, discrimination, guessing, and careless errors (see Figure 6.11). The fourth parameter adds considerable computational complexity and is rare to estimate.

6.1.3.5 Graded Response Model

Graded response models and *generalized partial credit models* can be estimated with one, two, three, or four parameters. However, they use polytomous data, as described in the section below.

6.1.4 Type of Data

IRT model are most commonly estimated with binary or dichotomous data. For example, the measures have questions or items that can be considered collapsed into two groups (e.g., true/false, correct/incorrect, endorsed/not endorsed). IRT models can also be estimated with polytomous data (e.g., likert scale), which adds computational complexity. IRT models with polytomous data can be fit with a graded response model or generalized partial credit model.

Figure 6.10: Three-parameter logistic model in item response theory.

For example, see Figure 6.12 for an example of an item boundary characteristic curve for an item from a 5-level likert scale (based on a cumulative distribution). If an item has k response categories, it has $k-1$ thresholds. For example, an item with 5-level likert scale (1 = strongly disagree; 2 = disagree; 3 = neither agree nor disagree; 4 = agree; 5 = strongly agree) has 4 thresholds: one from 1–2, one from 2–3, one from 3–4, and one from 4–5. The item boundary characteristic curve is the probability that a person selects a response category higher than k of a polytomous item. As depicted, one likert scale item does equivalent work as 4 binary items. See Figure 6.13 for the same 5-level likert scale item plotted with an item response category characteristic curve (based on a static, non-cumulative distribution).

IRT does not handle continuous data well, with some exceptions [Chen et al., 2019] such as in a Bayesian framework [Bürkner, 2019]. If you want to use continuous data, you might consider moving to a factor analysis framework.

6.1.5 Sample Size

Sample size requirement depend on the complexity of the model. A 1-parameter model often requires ~100 participants. A 2-parameter model often requires ~1,000 participants. A 3-parameter model often requires ~10,000 participants.

Figure 6.11: Four-parameter logistic model in item response theory.

6.1.6 Reliability

IRT conceptualizes reliability in a different way than classical test theory does. Both IRT and classical test theory conceptualize reliability as involving the *precision* of a measure's scores. In classical test theory, (im)precision—as operationalized by the standard error of measurement—is estimated with a single index across the whole range of the construct. That is, in classical test theory, the same standard error of measurement applies to all scores in the population [Embretson, 1996]. However, IRT estimates how much measurement precision (information) or imprecision (standard error of measurement) each item, and the test as a whole, have at different construct levels. This allows IRT to conceptualize reliability in such a way that precision/reliability can *differ* at different construct levels, unlike in classical test theory [Embretson, 1996]. Thus, IRT does not have one index of reliability; rather, its estimate of reliability differs at different levels on the construct.

Based on an item's difficulty and discrimination, we can calculate how much information each item provides. In IRT, *information* is how much measurement precision or consistency an item (or the measure) provides. In other words, information is the degree to which an item (or measure) reduces the standard error of measurement, that is, how much it reduces uncertainty of a person's level on the construct. As a reminder (from Equation (3.3.2)), the standard

Figure 6.12: Item boundary characteristic curves from two-parameter graded response model in item response theory.

Figure 6.13: Item response category characteristic curves from two-parameter graded response model in item response theory.

error of measurement is calculated as:

$$\text{standard error of measurement (SEM)} = \sigma_x \sqrt{1 - r_{xx}}$$

where σ_x = standard deviation of observed scores on the item x and r_{xx} = reliability of the item x. The standard error of measurement is used to generate confidence intervals for people's scores. In IRT, the standard error of measurement (at a given construct level) can be calculated as the inverse of the square root of the amount of test information at that construct level:

$$\text{SEM}(\theta) = \frac{1}{\sqrt{\text{information}(\theta)}}$$

The standard error of measurement tends to be higher (i.e., reliability/information tends to be lower) at the extreme levels of the construct where there are fewer items.

Consider some hypothetical items depicted with ICCs in Figure 6.1.6.

\begin{figure}
\caption{Item characteristic curves from two-parameter logistic model in item response theory. The dashed horizontal line indicates a probability of item endorsement of .50. The dashed vertical line is the item difficulty, i.e., the person's level on the construct (the location on the x-axis) at the inflection point of the item characteristic curve. In a two-parameter logistic model, the inflection point corresponds to the probability of item endorsement is 50%.

Thus, in a two-parameter logistic model, the difficulty of an item is the person's level on the construct where the probability of endorsing the item is 50%.} \end{figure}

We can present the ICC in terms of an item information function (see Figure 6.14). The information peak is located at the severity of the item. The higher the discrimination, the higher the information peak.

Figure 6.14: Item information from two-parameter logistic model in item response theory. The dashed vertical line is the item difficulty, which is located at the peak of the item information curve.

We can aggregate information across items to determine how much information the measure as a whole provides. This is called the test information function (see Figure 6.15). Note that we get more information from likert/multiple response items compared to binary/dichotomous items. Having 10 items with a 5-level response scale yields as much information as 40 dichotomous items.

Based on test information, we can calculate the standard error of measurement (see Figure 6.16). Notice how the degree of (un)reliability differs at different construct levels.

Figure 6.15: Test information curve from two-parameter logistic model in item response theory.

Figure 6.16: Test standard error of measurement from two-parameter logistic model in item response theory.

6.1.7 Efficient Assessment

One of the benefits of IRT is for item selection to develop brief assessments. For instance, you could use two items to tell you where the person is on the construct: low, middle, or high (see Figure 6.17). If the responses to the two items do not meet expectations, for instance, the person passes the difficult item but fails the easy item, we would keep assessing additional items to determine their level on the construct. If two items perform similarly, that is, they have the same difficulty and discrimination, they are redundant, and we can sacrifice one of them. This leads to greater efficiency and better measurement in terms of reliability and validity. For more information on designing and evaluating short forms compared to their full-scale counterparts, see Smith et al. [2000].

Figure 6.17: Item characteristic curves from two-parameter logistic model in item response theory.

IRT forms the basis of adaptive testing, which is discussed in Chapter 19. As discussed earlier, briefer measures can increase reliability and validity of measurement if the items are tailored to the ability level of the participant. The idea of adaptive testing is that, instead of having a standard scale for all participants, the items adapt to each person. An example of a measure that has used computerized adaptive testing is the Graduate Record Examination (GRE).

With adaptive testing, it is important to develop a comprehensive item bank that spans the difficulty range of interest. The starting construct level is the

50th percentile. If the respondennt gets the first item correct, it moves moves to the next item that would provide the most information, based on a split of the remaining sample (e.g., 75th percentile). And so on… The goal of adaptive testing is to find the construct level where the respondent keep getting items right and wrong 50% of the time. It is a promising approach that saves time because it tailors which items are administered to which person (based on their construct level) to get the most reliable estimate in the shortest time possible. However, it assumes that if you get a more difficult item correct, that you would have gotten easier items correct, which might not be true in all contexts.

Although most uses of IRT have been in cognitive and educational testing, IRT may also benefit other domains of assessment including clinical assessment [Gibbons et al., 2016, Reise and Waller, 2009, Thomas, 2019].

6.1.7.1 A Good Measure

According to IRT a good measure should:

1. fit your goals of the assessment, in terms of the range of interest regarding levels on the construct
2. have good items that yield lots of information
3. have a good set of items that densely cover the construct within the range of interest, without redundancy

First, a good measure should fit your goals of the assessment, in terms of the "range of interest" or the "target range" of levels on the construct. For instance, if your goal is to perform diagnosis, you would only care about the high end of the construct—there is no use discriminating between "nothing", "almost nothing", and "a little bit." For secondary prevention, i.e., early identification of risk to prevent something from getting worse, you would be interested in finding people with elevated risk—e.g., you would need to know who is 1 or more standard deviations above the mean, but you would not need to discriminate beyond that. For assessing individual differences, you would want items that discriminate across the full range, including at the lower end. The items' difficulty should span the range of interest.

Second, a good measure should have good items that yield lots of information. For example, the items should have strong discrimination, that is, the items are strongly related to the construct. The items should have sufficient variability in responses. This can be achieved by having likert/multiple response items, as opposed to binary items.

Third, a good measure should have a good set of items that densely cover the construct within the range of interest, without redundancy. The items should not have the same difficulty, or they would be considered redundant. The items' difficulty should densely cover the construct within the range of interest. For

instance, if the construct range of interest is 1–2 standard deviations above the mean, the items should have difficulty that densely cover this range (e.g., 1.0, 1.05, 1.10, 1.15, 1.20, 1.25, 1.30, ..., 2.0).

With items that (1) span the range of interest, (2) have high discrimination and information, and (3) densely cover the range of interest without redundancy, the measure should have a high information in the range of interest. This would allow it to efficiently and accurately assess the construct for the intended purpose.

6.2 Getting Started

6.2.1 Load Libraries

```
library("mirt")
library("tidyverse")
library("here")
library("tinytex")
```

6.2.2 Load Data

```
mydataIRT <- expand.table(LSAT7)
```

6.3 Comparison of Scoring Approaches

A measure that is a raw symptom count (i.e., a count of how many symptoms a person endorses) is low in precision and has a high standard error of measurement. Some diagnostic measures provide an ordinal response scale for each symptom. For example, the Structured Clinical Interview of Mental Disorders (SCID) provides a response scale from 0–2, where 0 = the symptom is absent, 1 = the symptom is sub-threshold, and 2 = the symptom is present. If your measure was a raw symptom sum, as opposed to a count of how many symptoms were present, the measure would be slightly more precise and have a somewhat smaller standard error of measurement.

A weighted symptom sum is the classical test theory analog of IRT. In classical test theory, proportion correct (or endorsed) would correspond to item difficulty and the item–total correlation (i.e., a point-biserial correlation) would correspond to item discrimination. If we were to compute a weighted sum of each item according to its strenght of association with the construct (i.e., the

item–total correlation), this measure would be somewhat more precise than the raw symptom sum, but it is not a latent variable method.

In IRT analysis, the weight for each item influences the estimate of a person's level on the construct. IRT down-weights the poorly discriminating items and up-weights the strongly discriminating items. This leads to greater precision and a lower standard error of measurement than non-latent scoring approaches.

According to Embretson [1996], many perspectives have changed because of IRT. First, according to classical test theory, longer tests are more reliable than shorter tests, as described in Section 3.5.5.3 in the chapter on reliability. However, according to IRT, shorter tests (i.e., tests with fewer items) can be more reliable than longer tests. Item selection using IRT can lead to briefer assessments that have greater reliability than longer scales. For example, adaptive tests that tailor the difficulty of the items to the ability level of the participant.

Second, in classical test theory, a score's meaning is tied to its location in a distribution (i.e., the norm-referenced standard). In IRT, however, the people and items are calibrated on a common scale. Based on a child's IRT-estimated ability level (i.e., level on the construct), we can have a better sense of what the child knows and does not know, because it indicates the difficulty level at which they would tend to get items correct 50% of the time; the person would likely fail items with a higher difficulty from this level, whereas the person would likely pass items with a lower difficulty compared to this level. Consider Binet's distribution of ability that arranges the items from easiest to most difficult. Based on the item difficulty and content of the items and the child's performance, we can have a better indication that a child can perform items successfully in a particular range (e.g., count to 10) but might not be able to perform more difficult items (e.g., tie their shoes). From an intervention perspective, this would allow working in the "window of opportunity" or the zone of proximal development. Thus, IRT can provide more meaningful understanding of a person's ability compared to traditional classical test theory such as the child being at the "63rd percentile" for a child of their age, which lacks conceptual meaning.

According to Cooper et al. [2009], our current diagnostic system relies heavily on how many symptoms a person endorses as an index of severity, but this assumes that all that all symptom endorsements have the same overall weight (severity). Using IRT, we can determine the relative severity of each item (symptom)—and it is clear that some symptoms indicate more severity than others. From this analysis, a respondent can endorse fewer, more severe items, and have overall more severe psychopathology than an individual who endorses more, less severe items. Basically, not all items are equally severe—know your items!

6.4 Relation between IRT, SEM, and CFA

IRT, like structural equation modeling (SEM) or confirmatory factor analysis (CFA), is an approach to latent variable modeling. IRT is essentially a form of CFA with categorical, binary, or ordinal data. As we discussed in Chapter 5, SEM is CFA with regression paths that specify hypothesized causal relations between the latent variables (the structural component of the model).

6.5 Rasch Model (1-Parameter Logistic)

A one-parameter logistic (1PL) item response theory (IRT) model is a model fit to dichotomous data, which estimates a different difficulty (b) parameter for each item. Discrimination (a) is not estimated (i.e., it is fixed at the same value—one—across items). Rasch models were fit using the `mirt` package [Chalmers, 2020a].

6.5.1 Fit Model

```
raschModel <- mirt(data = mydataIRT,
                   model = 1,
                   itemtype = "Rasch",
                   technical = list(removeEmptyRows = TRUE))
```

6.5.2 Model Summary

```
summary(raschModel)
```

```
##           F1    h2
## Item.1 0.512 0.262
## Item.2 0.512 0.262
## Item.3 0.512 0.262
## Item.4 0.512 0.262
## Item.5 0.512 0.262
##
## SS loadings:  1.311
## Proportion Var:  0.262
##
## Factor correlations:
##
```

```
##    F1
## F1  1
```

```
coef(raschModel, simplify = TRUE, IRTpars = TRUE)
```

```
## $items
##        a      b g u
## Item.1 1 -1.868 0 1
## Item.2 1 -0.791 0 1
## Item.3 1 -1.461 0 1
## Item.4 1 -0.521 0 1
## Item.5 1 -1.993 0 1
##
## $means
## F1
##  0
##
## $cov
##       F1
## F1 1.022
```

6.5.3 Plots

6.5.3.1 Test Curves

6.5.3.1.1 Test Characteristic Curve Plot of the expected total score as a function of a person's level on the latent construct (theta; θ).

```
plot(raschModel, type = "score")
```

6.5.3.1.2 Test Information Function Plot of test information as a function of a person's level on the latent construct (theta; θ).

```
plot(raschModel, type = "info")
```

6.5.3.1.3 Test Reliability Plot of test reliability as a function of a person's level on the latent construct (theta; θ).

```
plot(raschModel, type = "rxx")
```

Figure 6.18: Test characteristic curve from Rasch item response theory model

6.5.3.1.4 Test Standard Error of Measurement Plot of test standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

```
plot(raschModel, type = "SE")
```

6.5.3.1.5 Test Information Function and Test Standard Error of Measurement Plot of test information and standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

```
plot(raschModel, type = "infoSE")
```

6.5.3.2 Item Curves

6.5.3.2.1 Item Characteristic Curves Plot of probability of item endorsement (or getting the item correct) as a function of a person's level on the latent construct (theta; θ).

Figure 6.19: Test information function from Rasch item response theory model

```
plot(raschModel, type = "itemscore", facet_items = FALSE)
```

```
plot(raschModel, type = "itemscore", facet_items = TRUE)
```

6.5.3.2.2 Item Information Functions Plot of item information as a function of a person's level on the latent construct (theta; θ).

```
plot(raschModel, type = "infotrace", facet_items = FALSE)
```

```
plot(raschModel, type = "infotrace", facet_items = TRUE)
```

6.6 Two-Parameter Logistic Model

A two-parameter logistic (2PL) IRT model is a model fit to dichotomous data, which estimates a different difficulty (b) and discrimination (a) parameter for each item. 2PL models were fit using the `mirt` package [Chalmers, 2020a].

Figure 6.20: Test reliability from Rasch item response theory model

6.6.1 Fit Model

```
twoPLModel <- mirt(data = mydataIRT,
                   model = 1,
                   itemtype = "2PL",
                   technical = list(removeEmptyRows = TRUE))
```

6.6.2 Model Summary

```
summary(twoPLModel)
```

```
##           F1    h2
## Item.1 0.502 0.252
## Item.2 0.536 0.287
## Item.3 0.708 0.501
## Item.4 0.410 0.168
## Item.5 0.397 0.157
```

Figure 6.21: Test standard error of measurement from Rasch item response theory model

Figure 6.22: Test information function and standard error of measurement from Rasch item response theory model

Figure 6.23: Item characteristic curves from Rasch item response theory model

```
##
## SS loadings:  1.366
## Proportion Var:  0.273
##
## Factor correlations:
##
##    F1
## F1  1
```

```
coef(twoPLModel, simplify = TRUE, IRTpars = TRUE)
```

```
## $items
##            a      b g u
## Item.1 0.988 -1.879 0 1
## Item.2 1.081 -0.748 0 1
## Item.3 1.706 -1.058 0 1
## Item.4 0.765 -0.635 0 1
## Item.5 0.736 -2.520 0 1
##
## $means
## F1
```

Figure 6.24: Item characteristic curves from Rasch item response theory model

```
##  0
##
## $cov
##    F1
## F1  1
```

6.6.3 Plots

6.6.3.1 Test Curves

6.6.3.1.1 Test Characteristic Curve Plot of the expected total score as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPLModel, type = "score")
```

6.6.3.1.2 Test Information Function Plot of test information as a function of a person's level on the latent construct (theta; θ).

Figure 6.25: Item information functions from Rasch item response theory model

```
plot(twoPLModel, type = "info")
```

6.6.3.1.3 Test Reliability Plot of test reliability as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPLModel, type = "rxx")
```

6.6.3.1.4 Test Standard Error of Measurement Plot of test standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPLModel, type = "SE")
```

6.6.3.1.5 Test Information Function and Standard Errors Plot of test information and standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

Figure 6.26: Item information functions from Rasch item response theory model

```
plot(twoPLModel, type = "infoSE")
```

6.6.3.2 Item Curves

6.6.3.2.1 Item Characteristic Curves Plot of probability of item endorsement (or getting the item correct) as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPLModel, type = "itemscore", facet_items = FALSE)
```

```
plot(twoPLModel, type = "itemscore", facet_items = TRUE)
```

6.6.3.2.2 Item Information Functions Plot of item information as a function of a person's level on the latent construct (theta; θ).

Figure 6.27: Test characteristic curve from two-parameter logistic item response theory model

Figure 6.28: Test information function from two-parameter logistic item response theory model

Figure 6.29: Test reliability from two-parameter logistic item response theory model

Figure 6.30: Test standard error of measurement from two-parameter logistic item response theory model

Figure 6.31: Test information function and standard error of measurement from two-parameter logistic item response theory model

Figure 6.32: Item characteristic curves from two-parameter logistic item response theory model

Figure 6.33: Item characteristic curves from two-parameter logistic item response theory model

```
plot(twoPLModel, type = "infotrace", facet_items = FALSE)
```

Figure 6.34: Item information functions from two-parameter logistic item response theory model

```
plot(twoPLModel, type = "infotrace", facet_items = TRUE)
```

6.7 Two-Parameter Multi-Dimensional Logistic Model

A 2PL multi-dimensional IRT model is a model that allows multiple dimensions (latent factors) and is fit to dichotomous data, which estimates a different difficulty (b) and discrimination (a) parameter for each item. Multi-dimensional IRT models were fit using the `mirt` package [Chalmers, 2020a]. In this example, I estimate a 2PL multi-dimensional IRT model by estimating two factors.

Figure 6.35: Item information functions from two-parameter logistic item response theory model

6.7.1 Fit Model

```
twoPL2FactorModel <- mirt(data = mydataIRT,
                          model = 2,
                          itemtype = "2PL",
                          technical = list(removeEmptyRows = TRUE))
```

6.7.2 Model Summary

```
summary(twoPL2FactorModel)
```

```
##
## Rotation:  oblimin
##
## Rotated factor loadings:
##
##             F1      F2    h2
## Item.1 -0.7942  0.0111 0.623
## Item.2 -0.0803 -0.4631 0.255
## Item.3  0.0129 -0.8628 0.734
## Item.4 -0.2794 -0.1925 0.165
## Item.5 -0.2930 -0.1771 0.165
##
## Rotated SS loadings:  0.801 1.027
##
## Factor correlations:
##
##       F1    F2
## F1 1.000 0.463
## F2 0.463 1.000
```

```
coef(twoPL2FactorModel, simplify = TRUE)
```

```
## $items
##            a1     a2     d g u
## Item.1 -2.006  0.870 2.647 0 1
## Item.2 -0.849 -0.522 0.788 0 1
## Item.3 -2.152 -1.836 2.482 0 1
## Item.4 -0.756 -0.028 0.485 0 1
## Item.5 -0.757  0.000 1.864 0 1
##
```

```
## $means
## F1 F2
##  0  0
##
## $cov
##    F1 F2
## F1  1  0
## F2  0  1
```

6.7.3 Compare model fit

The modified model with two factors and the original one-factor model are considered "nested" models. The original model is nested within the modified model because the modified model includes all of the terms of the original model along with additional terms. Model fit of nested models can be compared with a chi-square difference test.

```
anova(twoPLModel, twoPL2FactorModel)
```

```
##
## Model 1: mirt(data = mydataIRT, model = 1, itemtype = "2PL", technical = list(remove
## Model 2: mirt(data = mydataIRT, model = 2, itemtype = "2PL", technical = list(remove

##        AIC     AICc    SABIC       HQ      BIC    logLik     X2  df     p
## 1 5337.610 5337.833 5354.927 5356.263 5386.688 -2658.805    NaN NaN   NaN
## 2 5335.039 5335.466 5359.283 5361.154 5403.748 -2653.520 10.571   4 0.032
```

Using a chi-square difference test to compare two nested models, the two-factor model fits significantly better than the one-factor model.

6.7.4 Plots

6.7.4.1 Test Curves

6.7.4.1.1 Test Characteristic Curve Plot of the expected total score as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPL2FactorModel, type = "score")
```

6.7.4.1.2 Test Information Function Plot of test information as a function of a person's level on the latent construct (theta; θ).

Figure 6.36: Test characteristic curve from two-parameter multi-dimensional item response theory model

```
plot(twoPL2FactorModel, type = "info")
```

Figure 6.37: Test information function from two-parameter multi-dimensional item response theory model

6.7.4.1.3 Test Standard Error of Measurement Plot of test standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPL2FactorModel, type = "SE")
```

6.7.4.2 Item Curves

6.7.4.2.1 Item Characteristic Curves Plot of probability of item endorsement (or getting the item correct) as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPL2FactorModel, type = "itemscore", facet_items = FALSE)
```

Figure 6.38: Test standard error of measurement from two-parameter multi-dimensional item response theory model

Figure 6.39: Item characteristic curves from two-parameter multi-dimensional item response theory model

```
plot(twoPL2FactorModel, type = "itemscore", facet_items = TRUE)
```

Figure 6.40: Item characteristic curves from two-parameter multi-dimensional item response theory model

6.7.4.2.2 Item Information Functions Plot of item information as a function of a person's level on the latent construct (theta; θ).

```
plot(twoPL2FactorModel, type = "infotrace", facet_items = FALSE)
```

```
plot(twoPL2FactorModel, type = "infotrace", facet_items = TRUE)
```

6.8 Three-Parameter Logistic Model

A three-parameter logistic (2PL) IRT model is a model fit to dichotomous data, which estimates a different difficulty (b), discrimination (a), and guessing parameter for each item. 3PL models were fit using the `mirt` package [Chalmers, 2020a].

Figure 6.41: Item information functions from two-parameter multi-dimensional item response theory model

Figure 6.42: Item information functions from two-parameter multi-dimensional item response theory model

6.8.1 Fit Model

```
threePLModel <- mirt(data = mydataIRT,
                     model = 1,
                     itemtype = "3PL",
                     technical = list(removeEmptyRows = TRUE))
```

6.8.2 Model Summary

```
summary(threePLModel)
```

```
##           F1    h2
## Item.1 0.509 0.259
## Item.2 0.750 0.562
## Item.3 0.700 0.489
## Item.4 0.397 0.158
## Item.5 0.411 0.169
##
## SS loadings:  1.637
## Proportion Var:  0.327
##
## Factor correlations:
##
##    F1
## F1  1
```

```
coef(threePLModel, simplify = TRUE, IRTpars = TRUE)
```

```
## $items
##            a      b     g u
## Item.1 1.007 -1.853 0.000 1
## Item.2 1.928 -0.049 0.295 1
## Item.3 1.667 -1.068 0.000 1
## Item.4 0.736 -0.655 0.000 1
## Item.5 0.767 -2.436 0.000 1
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
```

6.8.3 Plots

6.8.3.1 Test Curves

6.8.3.1.1 Test Characteristic Curve Plot of the expected total score as a function of a person's level on the latent construct (theta; θ).

```
plot(threePLModel, type = "score")
```

Figure 6.43: Test characteristic curve from three-parameter logistic item response theory model

6.8.3.1.2 Test Information Function Plot of test information as a function of a person's level on the latent construct (theta; θ).

```
plot(threePLModel, type = "info")
```

6.8.3.1.3 Test Reliability Plot of test reliability as a function of a person's level on the latent construct (theta; θ).

Figure 6.44: Test information function from three-parameter logistic item response theory model

```
plot(threePLModel, type = "rxx")
```

Figure 6.45: Test reliability from three-parameter logistic item response theory model

6.8.3.1.4 Test Standard Error of Measurement Plot of test standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

```
plot(threePLModel, type = "SE")
```

6.8.3.1.5 Test Information Function and Standard Errors Plot of test information and standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

```
plot(threePLModel, type = "infoSE")
```

Figure 6.46: Test standard error of measurement from three-parameter logistic item response theory model

Figure 6.47: Test information function and standard error of measurement from three-parameter logistic item response theory model

6.8.3.2 Item Curves

6.8.3.2.1 Item Characteristic Curves Plot of probability of item endorsement (or getting the item correct) as a function of a person's level on the latent construct (theta; θ).

```
plot(threePLModel, type = "itemscore", facet_items = FALSE)
```

Figure 6.48: Item characteristic curves from three-parameter logistic item response theory model

```
plot(threePLModel, type = "itemscore", facet_items = TRUE)
```

6.8.3.2.2 Item Information Functions Plot of item information as a function of a person's level on the latent construct (theta; θ).

```
plot(threePLModel, type = "infotrace", facet_items = FALSE)
```

Figure 6.49: Item characteristic curves from three-parameter logistic item response theory model

Figure 6.50: Item information functions from three-parameter logistic item response theory model

```
plot(threePLModel, type = "infotrace", facet_items = TRUE)
```

Figure 6.51: Item information functions from three-parameter logistic item response theory model

6.9 Four-Parameter Logistic Model

A four-parameter logistic (2PL) IRT model is a model fit to dichotomous data, which estimates a different difficulty (b), discrimination (a), guessing, and careless errors parameter for each item. 4PL models were fit using the `mirt` package [Chalmers, 2020a].

6.9.1 Fit Model

```
fourPLModel <- mirt(data = mydataIRT,
                    model = 1,
                    itemtype = "4PL",
                    technical = list(removeEmptyRows = TRUE,
                                     NCYCLES = 2000))
```

6.9.2 Model Summary

```
summary(fourPLModel)
```

```
##           F1    h2
## Item.1 0.833 0.694
## Item.2 0.981 0.963
## Item.3 0.760 0.577
## Item.4 0.876 0.768
## Item.5 0.649 0.422
##
## SS loadings:  3.424
## Proportion Var:  0.685
##
## Factor correlations:
##
##    F1
## F1  1
```

```
coef(fourPLModel, simplify = TRUE, IRTpars = TRUE)
```

```
## $items
##            a      b     g     u
## Item.1 2.565 -1.619 0.000 0.911
## Item.2 8.667  0.096 0.370 0.993
## Item.3 1.988 -0.651 0.269 0.999
## Item.4 3.095 -1.224 0.000 0.708
## Item.5 1.453 -2.165 0.002 0.920
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
```

6.9.3 Plots

6.9.3.1 Test Curves

6.9.3.1.1 Test Characteristic Curve Plot of the expected total score as a function of a person's level on the latent construct (theta; θ).

```
plot(fourPLModel, type = "score")
```

Figure 6.52: Test characteristic curve from four-parameter logistic item response theory model

6.9.3.1.2 Test Information Function Plot of test information as a function of a person's level on the latent construct (theta; θ).

```
plot(fourPLModel, type = "info")
```

6.9.3.1.3 Test Reliability Plot of test reliability as a function of a person's level on the latent construct (theta; θ).

```
plot(fourPLModel, type = "rxx")
```

6.9.3.1.4 Test Standard Error of Measurement Plot of test standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

Figure 6.53: Test information function from four-parameter logistic item response theory model

Figure 6.54: Test reliability from four-parameter logistic item response theory model

```
plot(fourPLModel, type = "SE")
```

Figure 6.55: Test standard error of measurement from four-parameter logistic item response theory model

6.9.3.1.5 Test Information Function and Standard Errors Plot of test information and standard error of measurement (SEM) as a function of a person's level on the latent construct (theta; θ).

```
plot(fourPLModel, type = "infoSE")
```

6.9.3.2 Item Curves

6.9.3.2.1 Item Characteristic Curves Plot of probability of item endorsement (or getting the item correct) as a function of a person's level on the latent construct (theta; θ).

```
plot(fourPLModel, type = "itemscore", facet_items = FALSE)
```

Figure 6.56: Test information function and standard error of measurement from four-parameter logistic item response theory model

Figure 6.57: Item characteristic curves from four-parameter logistic item response theory model

```
plot(fourPLModel, type = "itemscore", facet_items = TRUE)
```

Figure 6.58: Item characteristic curves from four-parameter logistic item response theory model

6.9.3.2.2 Item Information Functions Plot of item information as a function of a person's level on the latent construct (theta; θ).

```
plot(fourPLModel, type = "infotrace", facet_items = FALSE)
```

```
plot(fourPLModel, type = "infotrace", facet_items = TRUE)
```

6.10 Graded Response Model

A graded response model (GRM) is a 2PL IRT model fit to polytomous data (in this case, 1–4), which estimates a different difficulty (b) and discrimination (a) parameter for each item. It estimates four parameters for each item: difficulty [for each of three threshold transitions: 1–2 (b_1), 2–3 (b_2), and 3–4 (b_3)] and discrimination (a). GRM models were fit using the `mirt` package [Chalmers, 2020a].

Figure 6.59: Item information functions from four-parameter logistic item response theory model

Figure 6.60: Item information functions from four-parameter logistic item response theory model

6.10.1 Fit Model

```
gradedResponseModel <- mirt(data = Science,
                            model = 1,
                            itemtype = "graded",
                            technical = list(removeEmptyRows = TRUE))
```

6.10.2 Model Summary

```
summary(gradedResponseModel)
```

```
##            F1    h2
## Comfort 0.522 0.273
## Work    0.584 0.342
## Future  0.803 0.645
## Benefit 0.541 0.293
##
## SS loadings:  1.552
## Proportion Var:  0.388
##
## Factor correlations:
##
##    F1
## F1  1
```

```
coef(gradedResponseModel, simplify = TRUE, IRTpars = TRUE)
```

```
## $items
##             a     b1     b2    b3
## Comfort 1.042 -4.669 -2.534 1.407
## Work    1.226 -2.385 -0.735 1.849
## Future  2.293 -2.282 -0.965 0.856
## Benefit 1.095 -3.058 -0.906 1.542
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
```

6.10.3 Plots

6.10.3.1 Test Curves

6.10.3.1.1 Test Characteristic Curve Plot of the expected total score as a function of a person's level on the latent construct (theta; θ).

```
plot(gradedResponseModel, type = "score")
```

Figure 6.61: Test characteristic curve from graded response model

6.10.3.1.2 Test Information Function Plot of test information as a function of a person's level on the latent construct (theta; θ).

```
plot(gradedResponseModel, type = "info")
```

6.10.3.1.3 Test Reliability Plot of test reliability as a function of a person's level on the latent construct (theta; θ).

Figure 6.62: Test information function from graded response model

```
plot(gradedResponseModel, type = "rxx")
```

6.10.3.1.4 Test Standard Error of Measurement Plot of test standard error of measurement as a function of a person's level on the latent construct (theta; θ).

```
plot(gradedResponseModel, type = "SE")
```

6.10.3.1.5 Test Information Function and Standard Errors Plot of test information and standard error of measurement as a function of a person's level on the latent construct (theta; θ).

```
plot(gradedResponseModel, type = "infoSE")
```

Figure 6.63: Test reliability from graded response model

6.10.3.2 Item Curves

6.10.3.2.1 Item Characteristic Curves Plot of expected item score as a function of a person's level on the latent construct (theta; θ).

```
plot(gradedResponseModel, type = "itemscore", facet_items = FALSE)
```

```
plot(gradedResponseModel, type = "itemscore", facet_items = TRUE)
```

6.10.3.2.2 Item Information Functions Plot of item information as a function of a person's level on the latent construct (theta; θ).

```
plot(gradedResponseModel, type = "infotrace", facet_items = FALSE)
```

```
plot(gradedResponseModel, type = "infotrace", facet_items = TRUE)
```

Figure 6.64: Test standard error of measurement from graded response model

6.10.3.2.3 Item Response Category Characteristic Curves Plot of probability of item threshold endorsement as a function of a person's level on the latent construct (theta; θ).

```
plot(gradedResponseModel, type = "trace")
```

6.10.3.2.4 Item Boundary Characteristic Curves (aka Item Operation Characteristic Curves) This plot of item boundary characteristic curves was adapted from an example by Aiden Loe: https://aidenloe.github.io/irtplots.html

```
modelCoefficients <- coef(gradedResponseModel, IRTpars = TRUE, simplify = TRUE)$items

twopl <- function(a, b, theta){ # 2pl function
  1 / (1 + exp(-(a * (theta - b))))}

theta <- seq(from = -6, to = 6, by = .1)

difficultyThresholds <- grep("b", dimnames(modelCoefficients)[[2]], value = TRUE)
```

Figure 6.65: Test information function and standard error of measurement from graded response model

Figure 6.66: Item characteristic curves from graded response model

```
numberDifficultyThresholds <- length(difficultyThresholds)
items <- dimnames(modelCoefficients)[[1]]
numberOfItems <- length(items)

lst <- lapply(1:numberOfItems, function(x) data.frame(matrix(ncol = numberDifficultyThr

for(i in 1:numberOfItems){
  for(j in 1:numberDifficultyThresholds){
    lst[[i]][,1] <- theta
    lst[[i]][,j + 1] <- twopl(a = modelCoefficients[i,1], b = modelCoefficients[i,j + 1
  }
}

names(lst) <- items
dat <- bind_rows(lst, .id = "item")
longer_data <- pivot_longer(dat, cols = all_of(difficultyThresholds))

ggplot(longer_data, aes(theta, value, group = interaction(item, name), color = item)) +
  geom_line() +
  ylab("Probability of Endorsing an Item Response Category that is Higher than the Bou
```

Figure 6.67: Item characteristic curves from graded response model

```
theme_bw()
```

6.11 Conclusion

Item response theory is a measurement theory and advanced modeling approach that allows estimating latent variables as the common variance from multiple items, and how the items relate to the construct (latent variable). IRT holds promise to enable the development of briefer assessments, including short forms and adaptive assessments, that have strong reliability and validity. If you are interested in learning more about IRT, I highly recommend the book by Embretson and Reise [2000].

6.12 Exercises

Figure 6.68: Item information functions from graded response model

6.12.1 Questions

Note: several of the following questions use data from the Children of the National Longitudinal Survey of Youth Survey (CNLSY). The CNLSY is a publicly available longitudinal dataset provided by the Bureau of Labor Statistics (https://www.bls.gov/nls/nlsy79-children.htm#topical-guide). The CNLSY data file for these exercises is located on the course's page of the Open Science Framework (https://osf.io/3pwza). Children's behavior problems were rated in 1988 (time 1: T1) and then again in 1990 (time 2: T2) on the Behavior Problems Index (BPI). Below are the items corresponding to the Antisocial subscale of the BPI: 1) cheats or tells lies 2) bullies or is cruel/mean to others 3) does not seem to feel sorry after misbehaving 4) breaks things deliberately 5) is disobedient at school 6) has trouble getting along with teachers 7) has sudden changes in mood or feeling

1. Fit a one-parameter (Rasch) model to the seven items of the Antisocial subscale of the BPI at T1. This will estimate the difficulty for each item threshold (one threshold from 0 to 1, and one threshold from 1 to 2), while constraining the discrimination for each item to be the same.

    a. Which item has the lowest difficulty (i.e., severity) in terms of endorsing a score of one (i.e., "sometimes true") as opposed to zero

Figure 6.69: Item information functions from graded response model

(i.e., "not true")? Which item has the highest difficulty in terms of endorsing a score of 2 (i.e., "often true")? What do these estimates of item difficulty indicate?

2. Fit a graded response model to the seven items of the Antisocial subscale of the BPI at T1. This will estimate the difficulty for each item threshold (one threshold from 0 to 1, and one threshold from 1 to 2), while allowing each item to have a different discrimination.

   a. Provide a figure of the item characteristic curves.
   b. Provide a figure of the item boundary characteristic curves.
   c. Which item has the lowest discrimination? Which item has the highest discrimination? What do these estimates of item discrimination indicate?
   d. Provide a figure of the item information functions.
   e. Examining the item information functions, which item provides the most information at upper construct levels (2–4 standard deviations above the mean)? Which item provides the most information at lower construct levels (2–4 standard deviations below the mean)?
   f. Provide a figure of the test information function.
   g. Examining the test information function, where (at what construct levels) does the measure do the best job of assessing? Based on

Figure 6.70: Item response category characteristic curves from graded response model

Figure 6.71: Item boundary category characteristic curves from graded response model

its information function, describe what purposes the test would be better- or worse-suited for.

3. Fit a multi-dimensional graded response model to the seven items of the Antisocial subscale of the BPI at T1, by estimating two latent factors.

   a. Which items loaded onto Factor 1? Which items loaded onto Factor 2? Provide a possible explanation as two why some of the items "broke off" (from Factor 1) and loaded onto a separate factor (Factor 2).
   b. The one-factor graded response model (in #2) and the two-factor graded response model are considered "nested" models. The one-factor model is nested within the two-factor model because the two-factor model includes all of the terms of the one-factor model along with additional terms. Model fit of nested models can be directly compared with a chi-square difference test. Did the two-factor model fit better than the one-factor model?

6.12.2 Answers

1. a. Item 7 ("sudden changes in mood or feeling") has the lowest difficulty in terms of endorsing a score of one ($b_1 = -0.95$). Item 5 ("disobedient at school") has the highest difficulty in terms of endorsing a score of two ($b_2 = 3.55$). The difficulty parameter indicates the construct-level at the inflection point of the item characteristic curve. In a one- or two-parameter model, the inflection point occurs where 50% of respondents endorse the item. Thus, in this model, the difficulty parameter indicates the construct-level at which 50% of respondents endorse the item. It takes a very high level of antisocial behavior for a child to be endorsed as being often disobedient at school, whereas it does not take a high construct-level for a child to be endorsed as sometimes showing sudden changes in mood.

2. a. Below is a figure of item characteristic curves:

2. b. Below is a figure of item boundary characteristic curves:

2. c. Item 7 ("sudden changes in mood or feeling") has the lowest discrimination ($a = 0.89$). Item 6 ("has trouble getting along with teachers") has the highest discrimination ($a = 2.06$). The discrimination parameter represents the steepness of the slope of the item characteristic curve. It indicates how strongly endorsing an item discriminates (differentiates) between lower versus higher construct levels. In other words, it indicates how strongly the item is associated

Figure 6.72: Exercise 1a: Item characteristic curves

with the construct. Item 7 shows the weakest association with the construct, whereas item 6 shows the strongest association with the construct. That suggests that "trouble getting along with teachers" is more core to the construct of antisocial behavior than "sudden changes in mood."

d. Below is a figure of item information functions:

2. e. Item 6 ("has trouble getting along with teachers") provides the most information at upper construct levels (2–4 standard deviations above the mean). Item 7 ("has trouble getting along with teachers") provides the most information at lower construct levels (2–4 standard deviations below the mean). Item 1 ("cheats or tells lies") provides the most information at somewhat low construct levels (0–2 standard deviations below the mean).

f.

2. g. The measure does the best job of assessing (i.e., provides the most information) at construct levels from 1–3 standard deviations above the mean. Because the measure provides the most information at upper construct levels and provides little information at lower construct

Figure 6.73: Exercise 2b: Item boundary characteristic curves

levels, the measure would be best used for assessing clinical vs. subclinical levels of antisocial behavior rather than assessing individual differences in antisocial behavior across a community sample.

3. a. Items 1, 2, 3, 4, and 7 loaded onto Factor 1. Items 5 and 6 loaded onto Factor 2. Items 5 ("disobedient at school") and 6 ("trouble getting along with teachers") both deal with school-related antisocial behavior. Thus, the items assessing school-related antisocial behavior may share variance owing to the shared context of the behavior (school).

   b. Yes, the two-factor model fit significantly better than the one-factor model according to a chi-square difference test ($\Delta\chi^2[df = 6.00] = 273.85, p < .001$). Thus, antisocial behavior may not be a monolithic construct, but may depend on the context in which the behavior occurs.

Figure 6.74: Exercise 2c: Item information functions

Figure 6.75: Exercise 2e: Test information function

Chapter 7

Prediction

> "It is very difficult to predict—especially the future."
> — Neils Bohr

[**Garb and Wood, 2019 citation**]

7.1 Overview of Prediction

In psychology, we are often interested in predicting behavior. Behavior is complex. The same behavior can occur for different reasons. Behavior is probabilistically influenced by many processes, including processes internal to the person in addition to external processes. Moreover, people's behavior occurs in the context of a dynamic system with nonlinear, probabilistic, and cascading influences that change across time. The ever-changing system makes behavior challenging to predict. And, similar to chaos theory, one small change in the system can lead to large differences later on.

Predictions can come in different types. Some predictions involve categorical data, whereas other predictions involve continuous data. When dealing with categorical data, we can evaluate predictions a 2x2 table known as a confusion matrix (see Figure 7.4), or with logistic regression models. When dealing with continuous data, we can evaluate predictions using multiple regression or similar variants such as structural equation modeling and mixed models.

Let's consider a prediction example, assuming the following probabilities:

- The probability of contracting HIV is .3%
- The probability of a positive test for HIV is 1%
- The probability of a positive test if you have HIV is 95%

What is the probability of HIV if you have a positive test?

As we will see, the probability is: $\frac{95\% \times .3\%}{1\%} = 28.5\%$. So based on the above probabilities, if you have a positive test, the probability that you HIV is 28.5%. Most people tend to vastly over-estimate the likelihood that the person has HIV in this example. Why? Because they do not pay enough attention to the base rate.

7.1.1 Base Rates

A *base rate* is the probability of an event. It is important to distinguish between different types of probabilities: marginal probabilities, joint probabilities, and conditional probabilities. Base rates are marginal probabilities. A *marginal probability* is the probability of an event irrespective of the outcome of another variable. For instance, we can consider the following marginal probabilities:

$P(C_i)$ is the probability (i.e., base rate) of a classification, C, independent of other things. In our example above, $P(C_i)$ is the base rate (i.e., prevalence) of HIV in the population: $P(\text{HIV}) = .3\%$. $P(R_i)$ is the probability (base rate) of a response, R, independent of other things. In the example above, $P(R_i)$ is the base rate of a positive test for HIV: $P(\text{positive test}) = 1\%$. The base rate of a positive test is known as the positivity rate.

A *joint probability* is the probability of two (or more) events occurring simultaneously. For instance, the probability of events A and B both occurring together is $P(A, B)$. A joint probability can be calculated using the marginal probability of each event:

$$P(A, B) = P(A) \cdot P(B)$$

A *conditional probability* is the probability of one event occurring given the occurrence of another event. Conditional probabilities are written as: $P(A|B)$. This is read as the probability that event A occurs given that event B occurred. For instance, we can consider the following conditional probabilities:

$P(C|R)$ is the probability of a classification, C, given a response, R. In other words, $P(C|R)$ is the probability of having HIV given a positive test: $P(\text{HIV}|\text{positive test})$. $P(R|C)$ is the probability of a response, R, given a classification, C. In the example above, $P(R|C)$ is the probability of having a positive test given that a person has HIV: $P(\text{positive test}|\text{HIV}) = 95\%$.

A conditional probability can be calculated using the joint probability and marginal probability (base rate):

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

Conversely (and rearranging the terms), a joint probability can also be calculated using the conditional probability and marginal probability:

$$P(A, B) = P(A|B) \cdot P(B)$$

A conditional probability is not the same thing as its reverse (or inverse) conditional probability. Unless the base rate of the two events (C and R) are the same, $P(C|R) \neq P(R|C)$. However, people frequently make the mistake of thinking that two inverse conditional probabilities are the same. This mistake is know as the "confusion of the inverse", or the "inverse fallacy", or the "conditional probability fallacy". The confusion of inverse probabilities is the logical error of representative thinking that leads people to assume that the probability of C given R is the same as the probability of R given C, even though this is not true. As a few examples to demonstrate the logical fallacy, if 93% of breast cancers occur in high-risk women, this does not mean that 93% of high-risk women will eventually get breast cancer. As an another example, if 77% of car accidents take place within 15 miles of a driver's home, this does not mean that you will get in an accident 77% of times you drive within 15 miles of your home.

7.1.1.1 Bayes' Theorem

An alternative way of calculating a conditional probability is using the inverse conditional probability (instead of the joint probability). This is known as Bayes' theorem. Bayes' theorem can help us calculate a conditional probability of some classification, C, given some response, R, if we know the inverse conditional probability and the base rate (marginal probability) of each. Bayes' theorem is as follows:

$$P(C|R) = \frac{P(R|C) \cdot P(C_i)}{P(R_i)}$$

$$\text{posterior probability} = \frac{\text{likelihood} \times \text{prior probability}}{\text{evidence}}$$

Or, equivalently (rearranging the terms):

$$\frac{P(C|R)}{P(R|C)} = \frac{P(C_i)}{P(R_i)}$$

Or, equivalently (rearranging the terms):

$$\frac{P(C|R)}{P(C_i)} = \frac{P(R|C)}{P(R_i)}$$

In the HIV example above, we can calculate the probability of HIV given a positive test using three terms: the probability of a positive test given HIV

(i.e., the sensitivity of the test), the base rate of HIV, and the base rate of a positive test for HIV. The probability of HIV given a positive test is:

$$
\begin{aligned}
P(C|R) &= \frac{P(R|C) \cdot P(C_i)}{P(R_i)} \\
P(\text{HIV}|\text{positive test}) &= \frac{P(\text{positive test}|\text{HIV}) \cdot P(\text{HIV})}{P(\text{positive test})} \\
&= \frac{\text{sensitivity of test} \times \text{base rate of HIV}}{\text{base rate of positive test}} \\
&= \frac{95\% \times .3\%}{1\%} = \frac{.95 \times .003}{.01} \\
&= 28.5\%
\end{aligned}
$$

```
pAgivenB <- function(pBgivenA, pA, pB){
  value <- pBgivenA * pA / pB

  value
}

pAgivenB(pBgivenA = .95, pA = .003, pB = .01)
```

```
## [1] 0.285
```

Thus, assuming the probabilities in the example above, the probability of having HIV if a person has a positive test is 28.5%. Given a positive test, chances are higher than not that the person does not have HIV.

Bayes' theorem can be depicted visually [Ballesteros-Pérez et al., 2018]. If we have 100,000 people in our population, we would be able to fill out a 2-by-2 confusion matrix, as depicted in Figure 7.1.

		Truth		Total Predicted	Selection Ratio
		HIV	No HIV		
Prediction	HIV	TP = 285 (.00285)	FP = 715 (.00715)	1,000	SR = .01
	No HIV	FN = 15 (.00015)	TN = 98,985 (.98985)	99,000	1 - SR = .99
Total Actual		300	99,700	*N* = 100,000	
Base Rate		BR = .003	1 - BR = .997		

Figure 7.1: Confusion matrix: 2x2 prediction matrix

We know that .3% of the population contracts HIV, so 300 people in the population of 100,000 would contract HIV. Therefore, we put 300 in the marginal sum of those with HIV ($.003 \times 100,000 = 300$), i.e., the base rate of HIV. That means 99,700 people do not contract HIV ($100,000 - 300 = 99,700$. We know

that 1% of the population tests positive for HIV, so we put 1,000 in the marginal sum of those who test positive $.01 \times 100,000 = 1,000$, i.e., the base rate of a positive test (the selection ratio). That means 99,000 people test negative for HIV ($100,000 - 1,000 = 99,000$). We also know that 95% of those who have HIV test positive for HIV. Three hundred people have HIV, so 95% of them (i.e., 285 people; $.95 \times 300 = 285$) tested positive for HIV (true positives). Because we know that 300 people have HIV and that 285 of those with HIV tested positive, that means that 15 people with HIV tested negative ($300 - 15 = 285$; false negatives). We know that 1,000 people tested positive for HIV, and 285 with HIV tested positive, so that means that 715 people without HIV tested positive ($1,000 - 285 = 715$; false positives). We know that 99,000 people tested negative for HIV, and 15 with HIV tested negative, so that means that 98,985 people without HIV tested negative ($99,000 - 15 = 98,985$; true negatives). So, to answer the question of what is the probability of having HIV if you have a positive test, we divide the number of people with HIV who had a positive test (285) by the total number of people who had a positive test (1000), which leads to a probability of 28.5%.

This can be depicted visually in Figures 7.2 and 7.3. Please note that the areas in the figure are not drawn to scale. Otherwise, some regions would be too small to include text.

Now let's see what happens if the person tests positive a second time. We would revise our "prior probability" for HIV from the general prevalence in the population (0.3%) to be the "posterior probability" of HIV given a first positive test (28.5%). This is known as *Bayesian updating*. We would also update the "evidence" to be the marginal probability of getting a second positive test.

If we do not know a marginal probability (i.e., base rate) of an event (e.g., getting a second positive test), we can calculate a marginal probability with the *law of total probability* using conditional probabilities and the marginal probability of another event (e.g., having HIV). According to the law of total probability, the probability of getting a positive test is the probability that a person with HIV gets a positive test (i.e., sensitivity) times the base rate of HIV plus the probability that a person without HIV gets a positive test (i.e., false positive rate) times the base rate of not having HIV:

$$
\begin{aligned}
P(\text{not } C_i) &= 1 - P(C_i) \\
P(R_i) &= P(R|C) \cdot P(C_i) + P(R|\text{not } C) \cdot P(\text{not } C_i) \\
1\% &= 95\% \times .3\% + P(R|\text{not } C) \times 99.7\%
\end{aligned}
$$

In this case, we know the marginal probability ($P(R_i)$), and we can use that to solve for the unknown conditional probability that reflects the false positive rate ($P(R|\text{not } C)$):

Figure 7.2: Bayes' theorem (and confusion matrix) depicted visually, where the marginal probability is the base rate (BR). The four boxes represent the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Note: boxes are not drawn to scale. Otherwise, some regions would be too small to include text.

Figure 7.3: Bayes' theorem (and confusion matrix) depicted visually, where the marginal probability is the selection ratio (SR). The four boxes represent the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Note: boxes are not drawn to scale. Otherwise, some regions would be too small to include text.

$$
\begin{aligned}
P(R_i) &= P(R|C) \cdot P(C_i) + P(R|\text{not } C) \cdot P(\text{not } C_i) \\
P(R_i) - [P(R|\text{not } C) \cdot P(\text{not } C_i)] &= P(R|C) \cdot P(C_i) && \text{Move } P(R|\text{not } C) \text{ to the left side} \\
-[P(R|\text{not } C) \cdot P(\text{not } C_i)] &= P(R|C) \cdot P(C_i) - P(R_i) && \text{Move } P(R_i) \text{ to the right side} \\
P(R|\text{not } C) \cdot P(\text{not } C_i) &= P(R_i) - [P(R|C) \cdot P(C_i)] && \text{Multiply by } -1 \\
P(R|\text{not } C) &= \frac{P(R_i) - [P(R|C) \cdot P(C_i)]}{P(\text{not } C_i)} && \text{Divide by } P(R|\text{not } C) \\
&= \frac{1\% - [95\% \times .3\%]}{99.7\%} = \frac{.01 - [.95 \times .003]}{.997} \\
&= .7171515\%
\end{aligned}
$$

```
pBgivenNotA <- function(pB, pBgivenA, pA){
  value <- (pB - (pBgivenA * pA)) / (1 - pA)

  value
}

pBgivenNotA(pB = .01, pBgivenA = .95, pA = .003)
```

```
## [1] 0.007171515
```

With this conditional probability ($P(R|\text{not } C)$), the updated marginal probability of having HIV ($P(C_i)$), and the updated marginal probability of not having HIV ($P(\text{not } C_i)$), we can now calculate an updated estimate of the marginal probability of getting a second positive test. The probability of getting a second positive test is the probability that a person with HIV gets a second positive test (i.e., sensitivity) times the updated probability of HIV plus the probability that a person without HIV gets a second positive test (i.e., false positive rate) times the updated probability of not having HIV:

$$
\begin{aligned}
P(R_i) &= P(R|C) \cdot P(C_i) + P(R|\text{not } C) \cdot P(\text{not } C_i) \\
&= 95\% \times 28.5\% + .7171515\% \times 71.5\% = .95 \times .285 + .007171515 \times .715 \\
&= 27.58776\%
\end{aligned}
$$

We then substitute the updated marginal probability of HIV ($P(C_i)$) and the updated marginal probability of getting a second positive test ($P(R_i)$) into Bayes' theorem to get the probability that the person has HIV if they have a second positive test (assuming the errors of each test are independent, i.e., uncorrelated):

$$
\begin{aligned}
P(C|R) &= \frac{P(R|C) \cdot P(C_i)}{P(R_i)} \\
P(\text{HIV}|\text{a second positive test}) &= \frac{P(\text{a second positive test}|\text{HIV}) \cdot P(\text{HIV})}{P(\text{a second positive test})} \\
&= \frac{\text{sensitivity of test} \times \text{updated base rate of HIV}}{\text{updated base rate of positive test}} \\
&= \frac{95\% \times 28.5\%}{27.58776\%} \\
&= 98.14\%
\end{aligned}
$$

Thus, a second positive test greatly increases the posterior probability that the person has HIV from 28.5% to over 98%.

As seen in the rearranged formula in Equation (7.1.1.1), the ratio of the conditional probabilities is equal to the ratio of the base rates. Thus, it is important to consider base rates. People have a strong tendency to ignore (or give insufficient weight to) base rates when making predictions. The failure to consider the base rate when making predictions when given specific information about a case, is a cognitive bias known as the base-rate fallacy or as base rate neglect. For example, people tend to say that the probability of a rare event is more likely than it actually is given specific information.

As seen in the rearranged formula in Equation (7.1.1.1), the inverse conditional probabilities ($P(C|R)$ and $P(R|C)$) are not equal unless the base rates of C and R are the same. If the base rates are are not equal, we are making at least some prediction errors. If $P(C_i) > P(R_i)$, our predictions must include some false negatives. If $P(R_i) > P(C_i)$, our predictions must include some false positives.

Using the law of total probability, we can substitute the calculation of the marginal probability ($P(R_i)$) into Bayes' theorem to get an alternative formulation of Bayes' theorem:

$$
\begin{aligned}
P(C|R) &= \frac{P(R|C) \cdot P(C_i)}{P(R_i)} \\
&= \frac{P(R|C) \cdot P(C_i)}{P(R|C) \cdot P(C_i) + P(R|\text{not } C) \cdot P(\text{not } C_i)} \\
&= \frac{P(R|C) \cdot P(C_i)}{P(R|C) \cdot P(C_i) + P(R|\text{not } C) \cdot [1 - P(C_i)]}
\end{aligned}
$$

Instead of using marginal probability (base rate) of R, as in the original formulation of Bayes' theorem, it uses the conditional probability, $P(R|\text{not } C)$. Thus, it uses three terms: two conditional probabilities—$P(R|C)$ and $P(R|\text{not } C)$—and one marginal probability, $P(C_i)$. This alternative formulation of Bayes'

theorem can be used to calculate positive predictive value, based on sensitivity, specificity, and the base rate, as presented in Equation (7.5.20).

Let us see how the alternative formulation of Bayes' theorem applies to the HIV example above. We can calculate the probability of HIV given a positive test using three terms: the probability that a person with HIV gets a positive test (i.e., sensitivity), the probability that a person without HIV gets a positive test (i.e., false positive rate), and the base rate of HIV. Using the $P(R|\text{not } C)$ calculated in Equation (7.1.1.1), the probability of HIV given a single positive test is:

$$
\begin{aligned}
P(C|R) &= \frac{P(R|C) \cdot P(C_i)}{P(R|C) \cdot P(C_i) + P(R|\text{not } C) \cdot [1 - P(C_i)]} \\
&= \frac{\text{sensitivity of test} \times \text{base rate of HIV}}{\text{sensitivity of test} \times \text{base rate of HIV} + \text{false positive rate of test} \times (1 - \text{base rate of HIV})} \\
&= \frac{95\% \times .3\%}{95\% \times .3\% + .7171515\% \times (1 - .3\%)} = \frac{.95 \times .003}{.95 \times .003 + .007171515 \times (1 - .003)} \\
&= 28.5\%
\end{aligned}
$$

```
pAgivenBalternative <- function(pBgivenA, pA, pBgivenNotA){
  value <- (pBgivenA * pA) /((pBgivenA * pA) + (pBgivenNotA * (1 - pA)))

  value
}

pAgivenBalternative(pBgivenA = .95, pA = .003, pBgivenNotA = .007171515)
```

```
## [1] 0.285
```

```
pAgivenBalternative(pBgivenA = .95, pA = .003, pBgivenNotA = pBgivenNotA(pB = .01, pBg
```

```
## [1] 0.285
```

To calculate the probability of HIV given a second positive test, we update our priors because the person has now tested positive for HIV. We update the prior probability of HIV ($P(C_i)$) based on the posterior probability of HIV after a positive test ($P(C|R)$) that we calculated above. We can calculate the probability of HIV given a second positive test using three terms: the probability that a person with HIV gets a positive test (i.e., sensitivity; which stays the same), the probability that a person without HIV gets a positive test (i.e., false positive rate; which stays the same), and the updated marginal probability of HIV. The probability of HIV given a second positive test is:

$$
\begin{aligned}
P(C|R) &= \frac{P(R|C) \cdot P(C_i)}{P(R|C) \cdot P(C_i) + P(R|\text{not } C) \cdot [1 - P(C_i)]} \\
&= \frac{\text{sensitivity of test} \times \text{updated base rate of HIV}}{\text{sensitivity of test} \times \text{updated base rate of HIV} + \text{false positive rate of test} \times (1 - \text{updated base rate of HIV})} \\
&= \frac{95\% \times 28.5\%}{95\% \times 28.5\% + .7171515\% \times (1 - 28.5\%)} = \frac{.95 \times .285}{.95 \times .285 + .007171515 \times (1 - .285)} \\
&= 98.14\%
\end{aligned}
$$

```
pAgivenBalternative(pBgivenA = .95, pA = .285, pBgivenNotA = .007171515)
```

```
## [1] 0.9814134
```

```
pAgivenBalternative(pBgivenA = .95, pA = .285, pBgivenNotA = pBgivenNotA(pB = .01, pBgivenA = .95
```

```
## [1] 0.9814134
```

If we want to compare the relative probability of two outcomes, we can use the odds form of Bayes' theorem:

$$
\begin{aligned}
P(C|R) &= \frac{P(R|C) \cdot P(C_i)}{P(R_i)} \\
P(\text{not } C|R) &= \frac{P(R|\text{not } C) \cdot P(\text{not } C_i)}{P(R_i)} \\
\frac{P(C|R)}{P(\text{not } C|R)} &= \frac{\frac{P(R|C) \cdot P(C_i)}{P(R_i)}}{\frac{P(R|\text{not } C) \cdot P(\text{not } C_i)}{P(R_i)}} \\
&= \frac{P(R|C) \cdot P(C_i)}{P(R|\text{not } C) \cdot P(\text{not } C_i)} \\
&= \frac{P(C_i)}{P(\text{not } C_i)} \times \frac{P(R|C)}{P(R|\text{not } C)} \\
\text{posterior odds} &= \text{prior odds} \times \text{likelihood ratio}
\end{aligned}
$$

In sum, the marginal probability, including the prior probability or base rate, should be weighed heavily in predictions unless there is sufficient data to indicate otherwise, i.e., to update the posterior probability based on new evidence. Bayes' theorem provides a powerful tool to anchor predictions to the base rate unless sufficient evidence changes the posterior probability (by updating the evidence and prior probability).

7.1.2 Prediction Accuracy

7.1.2.1 Decision Outcomes

To consider how we can evaluate the accuracy of predictions, consider an example adapted from Meehl and Rosen [1955]. The military conducts a test of its prospective members to screen out applicants who would likely fail basic training. To evaluate the accuracy of our predictions using the test, we can examine a confusion matrix. A confusion matrix is a matrix that presents the predicted outcome on one dimension and the actual outcome (truth) on the other dimension. If the predictions and outcomes are dichotomous, the confusion matrix is a 2x2 matrix with two rows and two columns that represent four possible predicted-actual combinations (decision outcomes): true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

When discussing the four decision outcomes, "true" means an accurate judgment, whereas "false" means an inaccurate judgment. "Positive" means that the judgment was that the person has the characteristic of interest, whereas "negative" means that the judgment was that the person does not have the characteristic of interest. A *true positive* is a correct judgment (or prediction) where the judgment was that the person has (or will have) the characteristic of interest, and, in truth, they actually have (or will have) the characteristic. A *true negative* is a correct judgment (or prediction) where the judgment was that the person does not have (or will not have) the characteristic of interest, and, in truth, they actually do not have (or will not have) the characteristic. A *false positive* is an incorrect judgment (or prediction) where the judgment was that the person has (or will have) the characteristic of interest, and, in truth, they actually do not have (or will not have) the characteristic. A *false negative* is an incorrect judgment (or prediction) where the judgment was that the person does not have (or will not have) the characteristic of interest, and, in truth, they actually do have (or will have) the characteristic.

An example of a confusion matrix is in Figure 7.4.

		Actual Adjustment	
		Poor	Good
Decision (Prediction)	Reject	TP = 86	FP = 422
	Retain	FN = 14	TN = 1,478

Figure 7.4: Confusion matrix: 2x2 prediction matrix

With the information in the confusion matrix, we can calculate the marginal sums, and the proportion of people in each cell (in parentheses), as depicted in Figure 7.5.

That is, we can sum across the rows and columns to identify how many people actually showed poor adjustment ($n = 100$) versus good adjustment ($n = 1,900$), and how many people were selected to reject ($n = 508$) versus retain

		Actual Adjustment		Total Predicted
		Poor	Good	
Decision (Prediction)	Reject	TP = 86 (.043)	FP = 422 (.211)	508
	Retain	FN = 14 (.007)	TN = 1,478 (.739)	1,492
Total Actual		100	1,900	*N* = 2,000

Figure 7.5: Confusion matrix: 2x2 prediction matrix with marginal sums

($n = 1,492$). If we sum the column of predicted marginal sums ($508 + 1,492$) or the row of actual marginal sums ($100 + 1,900$), we get the total number of people ($N = 2,000$).

Based on the marginal sums, we can compute the marginal probabilities, as depicted in Figure 7.6.

		Actual Adjustment		Total Predicted	Selection Ratio
		Poor	Good		
Decision (Prediction)	Reject	TP = 86 (.043)	FP = 422 (.211)	508	SR = 0.254
	Retain	FN = 14 (.007)	TN = 1,478 (.739)	1,492	1 – SR = 0.746
Total Actual		100	1,900	*N* = 2,000	
Base Rate		BR = 0.05	1 – BR = 0.95		

Figure 7.6: Confusion matrix: 2x2 prediction matrix with marginal sums and marginal probabilities

The marginal probability of the person having the characteristic of interest (i.e., showing poor adjustment) is called the *base rate* (BR). That is, the base rate is the proportion of people that have the characteristic. It is calculated by dividing the number of people with poor adjustment ($n = 100$) by the total number of people ($N = 2,000$): $BR = \frac{FN+TP}{N}$. Here, the base rate reflects the prevalence of poor adjustment. In this case, the base rate is .05, so there is a 5% chance that an applicant will be poorly adjusted. The marginal probability of good adjustment is equal to 1 minus the base rate.

The marginal probability of predicting that a person has the characteristic (i.e., rejecting a person) is called the *selection ratio* (SR). The selection ratio is the proportion of people who will be selected (in this case, rejected rather than retained); i.e., the proportion of people who are identified as having the characteristic. The selection ratio is calculated by dividing the number of people selected to reject ($n = 508$) by the total number of people ($N = 2,000$): $SR = \frac{TP+FP}{N}$. In this case, the selection ratio is .25, so 25% of people are rejected. The marginal probability of not selecting someone to reject (i.e., the marginal probability of retaining) is equal to 1 minus the selection ratio.

The selection ratio might be something that the test dictates according to its

cutoff score. Or, the selection ratio might be imposed by external factors that place limits on how many people you can assign a positive test value. For instance, when deciding whether to treat a client, the selection ratio may depend on how many therapists are available and how many cases can be treated.

7.1.2.2 Percent Accuracy

Based on the confusion matrix, we can calculate the prediction accuracy based on the percent accuracy of the predictions. The percent accuracy is the number of correct predictions divided by the total number of predictions, and multiplied by 100. In the context of a confusion matrix, this is calculated as: $100\% \times \frac{\text{TP+TN}}{n}$. In this case, our percent accuracy was 78%—that is, 78% of our predictions were accurate, and 22% of our predictions were inaccurate.

7.1.2.3 Percent Accuracy by Chance

78% sounds pretty accurate. And it is much higher than 50%, so we are doing a pretty good job, right? Well, it is important to compare our accuracy to what accuracy we would expect to get by chance alone, if predictions were made by a random process rather than using a test's scores. Our selection ratio was 25.4%. How accurate would we be if we randomly selected 25.4% of people to reject? To determine what accuracy we could get by chance alone given the selection ratio and the base rate, we can calculate the chance probability of true positives and the chance probability of true negatives. The probability of a given cell in the confusion matrix is a joint probability—the probability of two events occurring simultaneously. To calculate a joint probability, we multiply the probability of each event.

So, to get the chance expectancies of true positives, we would multiply the respective marginal probabilities:

$$\begin{aligned} P(TP) &= P(\text{Poor adjustment}) \times P(\text{Reject}) \\ &= BR \times SR \\ &= .05 \times .254 \\ &= .0127 \end{aligned}$$

To get the chance expectancies of true negatives, we would multiply the respective marginal probabilities:

$$\begin{aligned} P(TN) &= P(\text{Good adjustment}) \times P(\text{Retain}) \\ &= (1 - BR) \times (1 - SR) \\ &= .95 \times .746 \\ &= .7087 \end{aligned}$$

To get the percent accuracy by chance, we sum the chance expectancies for the correct predictions (TP and TN): .0127 + .7087 = .7214. Thus, the percent accuracy you can get by chance alone is 72%. This is because most of our predictions are to retain people, and the base rate of poor adjustment is quite low (.05). Our measure with 78% accuracy provides only a 6% increment in correct predictions. Thus, you cannot judge how good your judgment or prediction is until you know how you would do by random chance.

The chance expectancies for each cell of the confusion matrix are in Figure 7.7.

		Actual Adjustment		Selection Ratio
		Poor	Good	
Decision (Prediction)	Reject	.0127	.2413	SR = 0.254
	Retain	.0373	.7087	1 – SR = 0.746
Base Rate		BR = 0.05	1 – BR = 0.95	

Figure 7.7: Chance expectancies in 2x2 prediction matrix

7.1.2.4 Predicting from the Base Rate

Now, let us consider how well you would do if you were to predict from the base rate. Predicting from the base rate is also called "betting from the base rate", and it involves setting the selection ratio by taking advantage of the base rate so that you go with the most likely outcome in every prediction. Because the base rate is quite low (.05), we could predict from the base rate by selecting no one to reject (i.e., setting the selection ratio at zero). our percent accuracy by chance if we predict from the base rate would be calculated by multiplying the marginal probabilities, as we did above, but with a new selection ratio:

$$\begin{aligned} P(TP) &= P(\text{Poor adjustment}) \times P(\text{Reject}) \\ &= BR \times SR \\ &= .05 \times 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} P(TN) &= P(\text{Good adjustment}) \times P(\text{Retain}) \\ &= (1 - BR) \times (1 - SR) \\ &= .95 \times 1 \\ &= .95 \end{aligned}$$

We sum the chance expectancies for the correct predictions (TP and TN): $0 + .95 = .95$. Thus, our percent accuracy by predicting from the base rate is 95%. This is damning to our measure because it is a much higher accruacy than the

accuracy of our measure. That is, we can be much more accurate than our measure simply by predicting from the base rate and selecting no one to reject.

Going with the most likely outcome in every prediction (predicting from the base rate) can be highly accurate as noted by Meehl and Rosen [1955], especially when the base rate is very low or very high. This should serve as an important reminder that we need to compare the accuracy of our measures to the accuracy by (1) random chance and (2) predicting from the base rate. There are several important implications of the impact of base rates on prediction accuracy. One implication is that using the same test in different settings with different base rates will markedly change the accuracy of the test. Oftentimes, using a test will actually *decrease* the predictive accuracy when the base rate deviates greatly from .50. But, percent accuracy is not everything. Percent accuracy treats different kinds of errors as if they are equally important. However, the value we place on different kinds of errors may be different, as described next.

7.1.2.5 Different Kinds of Errors Have Different Costs

Some errors have a high cost and some errors have a low cost. Among the four decision outcomes, there were two types of errors: false positives and false negatives. The extent to which false positives and false negatives are costly depends on the prediction problem. So, even though you can often be most accurate by going with the base rate, it may be advantageous to using a screening instrument despite lower overall accuracy because of the huge difference in costs of false positives versus false negatives in some cases.

Consider the example of a screening instrument for HIV. False positives would be cases where we said that someone is at high risk of HIV when they are not, whereas false negatives are cases where we said that someone is not at high risk when they actually are. The costs of false positives include a shortage of blood, some follow-up testing, and potentially some anxiety, but that is about it. The costs of false negatives may be people getting HIV. In this case, the costs of false negatives greatly outweigh the costs of false positives, so we use a screening instrument to try to identify the cases at high risk for HIV because of the important consequences of failing to do so, even though using the screening instrument will lower our overall accuracy level.

Another example is when the Central Intelligence Agency (CIA) used a screen for protective typists during wartime to try to detect spies. False positives would be cases where the CIA believes that a person is a spy when they are not, and the CIA does not hire them. False negatives would be cases where the CIA believes that a person is not a spy when they actually are, and the CIA hires them. In this case, a false positive would be fine, but a false negative would be really bad.

How you the costs of different errors depends considerably on the domain and context. Possible costs of false positives to society include: unnecessary and

costly treatment with side effects and sending an innocent person to jail (despite our presumption of innocence in the United States criminal justice system that a person is innocent until proven guilty). Possible costs of false negatives to society include: Setting a guilty person free, failing to detect a bomb or tumor, and preventing someone from getting treatment who needs it.

The differential costs of different errors also depends on how much flexibility you have in the selection ratio in being able to set a stringent versus loose selection ratio. Consider if there is a high cost of getting rid of people during the selection process. For example, if you must hire 100 poeple and only 100 people apply for the position, you cannot lose people, so you need to hire even high-risk people. However, if you do not need to hire many people, then you can hire more conservatively.

Any time the selection ratio differs from the base rate, you will make errors. For example, if you reject 25% of applicants, and the base rate of poor adjustment is 5%, then you are marking errors of over-rejecting (false positives). By contrast, if you reject 1% of applicants and the base rate of poor adjustment is 5%, then you are making errors of under-rejecting or over-accepting (false negatives).

A low base rate makes it harder to make predictions, and tends to lead to less accurate predictions. For instance, it is very challenging to predict low base rate behaviors, including suicide [Kessler et al., 2020]. The difficulty in predicting events with a low base rate is apparent with the true score formula from classical test theory: $X = T + e$. As described in Equation (3.3), reliability is the ratio of true score variance to observed score variance. As true score variance increases, reliability increases. If the base rate is .05, then the max variance of the true scores is .05. The lower true score variance makes the measure less reliable and hard to make accurate predictions.

7.1.2.6 Sensitivity, Specificity, PPV, and NPV

As described earlier, percent accuracy is not the only important aspect of accuracy. Percent accuracy can be misleading because it is highly influenced by base rates. You can have a high percent accuracy by predicting from the base rate and saying that no one has the condition (if the base rate is low) or that everyone has the condition (if the base rate is high). Thus, it is also important to consider other aspects of accuracy, including sensitivity (SN), specificity (SP), positive predictive value (PPV), and negative predictive value (NPV). We want our predictions to be sensitive to be able to detect the chararacteristic but also to be specific so that we classify only the characteristic as having the characteristic.

Let us return to the confusion matrix in Figure 7.8. If we know the frequency of each of the four predicted-actual combinations of the confusion matrix (TP, TN, FP, FN), we can calculate sensitivity, specificity, PPV, and NPV.

		Actual Adjustment		Total Predicted
		Poor	Good	
Decision (Prediction)	Reject	TP = 86	FP = 422	508
	Retain	FN = 14	TN = 1,478	1,492
Total Actual		100	1,900	*N* = 2,000

Figure 7.8: Confusion matrix: 2x2 prediction matrix

Sensitivity is the proportion of those with the characteristic (TP + FN) that we identified with our measure (TP): $\frac{\text{TP}}{\text{TP+FN}} = \frac{86}{86+14} = .86$. Specificity is the proportion of those who do not have the characteristic (TN + FP) that we correctly classify as not having the characteristic (TN): $\frac{\text{TN}}{\text{TN+FP}} = \frac{1,478}{1,478+422} = .78$. PPV is the proportion of those who we classify as having the characteristic (TP + FP) who actually have the characteristic (TP): $\frac{\text{TP}}{\text{TP+FP}} = \frac{86}{86+422} = .17$. NPV is the proportion of those we classify as not having the characteristic (TN+ FN) who actually do not have the characteristic (TN): $\frac{\text{TN}}{\text{TN+FN}} = \frac{1,478}{1,478+14} = .99$.

Sensitivity, specificity, PPV, and NPV are proportions, and their values therefore range from 0 to 1, where higher values reflect greater accuracy. With sensitivity, specificity, PPV, and NPV, we have a good snapshot of how accurate the measure is at a given cutoff. In our case, our measure is good at finding who to reject (high sensitivity), but it is rejecting too many people who do not need to be rejected (lower PPV due to many FPs). Most people who we classify as having the characteristic do not actually have the characteristic. However, the fact that we are over-rejecting could be okay depending on our goals, for instance, if we do not care about over-dropping (i.e., the PPV being low).

7.1.2.6.1 Some Accuracy Estimates Depend on the Cutoff Sensitivity, specificity, PPV, and NPV differ based on the cutoff (i.e., threshold) for classification. Consider the following example. Aliens visit the earth and they develop a test to determine whether a berry is edible or inedible.

Figure 7.9 depicts the distributions of scores by berry type. Note how there are clearly two distinct distributions. However, the distributions overlap to some degree. Thus, any cutoff will have at least some inaccurate classifications.

Figure 7.10 depicts the distributions of scores by berry type with a cutoff. The red line indicates the cutoff—the level above which berries are classified by the test as inedible. There are errors on each side of the cutoff. Below the cutoff, there are some false negatives (blue): inedible berries that are inaccurately classified as edible. Above the cutoff, there are some false positives (green): edible berries that are inaccurately classified as inedible. Costs of false negatives could include sickness or death from eating the inedible berries. Costs of false positives could include taking longer to find food, finding insufficient food, and starvation.

Figure 7.9: Distribution of test scores by berry type

Based on our assessment goals, we might use a different selection ratio by changing the cutoff. Figure 7.11 depicts the distributions of scores by berry type when we raise the cutoff. There are now more false negatives (blue) and fewer false positives (green). If we raise the cutoff (to be more conservative), the number of false negatives increases and the number of false positives decreases. Consequently, as the cutoff increases, sensitivity and NPV decrease (because we have more false negatives), whereas specificity and PPV increase (because we have fewer false positives). A higher cutoff could be optimal if the costs of false positives are considered greater than the costs of false negatives. For instance, if the aliens cannot risk eating the inedible berries because the berries are fatal and there are sufficient edible berries that can be found to feed the alien colony.

Figure 7.12 depicts the distributions of scores by berry type when we lower the cutoff. There are now fewer false negatives (blue) and more false positives (green). If we lower the cutoff (to be more liberal), the number of false negatives decreases and the number of false positives increases. Consequently, as the cutoff decreases, sensitivity and NPV increase (because we have fewer false negatives), whereas specificity and PPV decrease (because we have more false positives). A lower cutoff could be optimal if the costs of false negatives are considered greater than the costs of false positives. For instance, if the aliens cannot risk missing edible berries because they are in short supply relative to the size of alien colony, and eating the inedible berries would, at worst, lead to minor,

Figure 7.10: Classifications based on a cutoff. Note that some true negatives and true positives are hidden behind the false positives and false negatives.

Figure 7.11: Classifications based on raising the cutoff. Note that some true negatives and true positives are hidden behind the false positives and false negatives.

temporary discomfort.

Figure 7.12: Classifications based on lowering the cutoff. Note that some true negatives and true positives are hidden behind the false positives and false negatives.

In sum, sensitivity and specificity differ based on the cutoff for classification. if we raise the cutoff, specificity and PPV increase (due to fewer false positives), whereas and sensitivity and NPV decrease (due to more false negatives). If we lower the cutoff, sensitivity and NPV increase (due to fewer false negatives), whereas specificity and PPV decrease (due to more false positives). Thus, the optimal cutoff depends on how costly each type of error is: false negatives and false positives. If false negatives are more costly than false positives, we would set a low cutoff. If false positives are more costly than false negatives, we would set a high cutoff.

7.1.2.7 Signal Detection Theory

Signal detection theory (SDT) is a probability-based theory for the detection of a given stimulus (signal) from a stimulus set that includes non-target stimuli (noise). SDT was developed research on radar (*RA*dio *D*etection *A*nd *R*anging) and sonar (*SO*und *N*avigation *A*nd *R*anging) in World War II based on research on sensory-perception research. The military wanted to be able to determine

which objects on radar/sonar were enemy aircraft/submarines, and which were noise (e.g., different object in the environment or even just the weather itself). SDT allowed determining how many errors operators made (how accurate they were) and decomposing errors into different kinds of errors. SDT distinguishes between sensitivity and bias. In SDT, *sensitivity* (or discriminability) is how well an assessment distinguishes between a target stimulus and non-target stimuli (i.e., how well the assessment detects the target stimulus amid non-target stimuli). *Bias* is the extent to which the probability of a selection decision from the assessment is higher or lower than the true rate of the target stimulus.

Some radar/sonar operators were not as sensitive to the differences between signal and noise, due to factors such as age, ability to distinguish gradations of a signal, etc. People who showed low sensitivity (i.e., who were not as successful at distinguishing between signal and noise) were screened out because the military perceived sensitivity as a skill that was not easily taught. By contrast, other operators could distinguish signal from noise, but their threshold was too low or high—they could take in information, but their decisions tended to be wrong due to systematic bias or poor calibration. That is, they systematically over-rejected or under-rejected stimuli. Over-rejecting leads to many false negatives (i.e., saying that a stimulus is safe when it is not). Under-rejecting leads to many false positives (i.e., saying that a stimulus is harmful when it is not). A person who showed good sensitivity but systematic bias was considered more teachable than a person who showed low sensitivity. Thus, radar and sonar operators were selected based on their sensitivity to distinguish signal from noise, and then were trained to improve the calibration so they reduce their systematic bias and do not systematically over- or under-reject.

Although SDT was originally developed for use in World War II, it now plays an important role in many areas of science and medicine. A medical application of SDT is tumor detection in radiology. SDT also plays an important role in psychology, especially cognitive psychology. For instance, research on social perception of sexual interest has shown that men tend to show lack of sensitivity to differences in women's affect—i.e., they have relative difficulties discriminating between friendliness and sexual interest [*CITE TERESA'S WORK*]. Men also tend to show systematic bias (poor calibration) such that they tend to over-estimate women's sexual interest in them—i.e., men tend to have too low of a threshold for determining that a women is showing sexual interest in them.

SDT metrics of sensitivity include d', A (or A'), and the area under the receiver operating characteristic (ROC) curve. SDT metrics of bias include β, c, and b.

7.1.2.7.1 Receiver Operating Characteristic (ROC) Curve The x-axis of the ROC curve is the false alarm rate or false positive rate (1−specificity). The y-axis is the hit rate or true positive rate (sensitivity). We can trace the ROC curve as the combination between sensitivity and specificity at every possible cutoff. At a cutoff of zero (top right of ROC curve), we calculate sensitivity (1.0) and specificity (0) and plot it. At a cutoff of zero, the assessment tell us

to make an action for every stimulus (i.e., it is the most liberal). We then gradually increase the cutoff, and plot sensitivity and specificity. As the cutoff increases, sensitivity decreases and specificity increases. We end at the highest possible cutoff, where the sensitivity is 0 and the specificity is 1.0 (i.e., we never make an action; i.e., it is the most conservative). Each point on the ROC curve corresponds to a pair of hit and false alarm rates (sensitivity and specificity) resulting from a specific cutoff value. Then, we can draw a lines or a curve that connects the points.

Figure 7.13 depicts an empirical ROC plot where lines are drawn to connect the hit and false alarm rates.

Figure 7.13: Empirical receiver operating characteristic curve. AUC = Area under the receiver operating characteristic curve.

Figure 7.14 depicts an ROC curve where a smoothed and fitted curve is drawn

to connect the hit and false alarm rates.

Figure 7.14: Smooth receiver operating characteristic curve. AUC = Area under the receiver operating characteristic curve.

7.1.2.7.1.1 Area Under the ROC Curve ROC methods can be used to compare and compute the discriminative power of measurement devices free from the influence of selection ratios, base rates, and costs and benefits. A ROC analysis yields a quantitative index of how well an index predicts a signal of interest or can discriminate between different signals. ROC analysis can help tell us how often our assessment would be correct. If we randomly pick two observations, and we were right once and wrong once, we were 50% accurate. But this would be a useless measure because it reflects chance responding.

The geometrical area under the ROC curve reflects the discriminative accuracy

of the measure. The index is called the area Under the ROC Curve (AUC). AUC quantifies the discriminative power of an assessment. AUC is the probability that a randomly selected target and a randomly selected non-target is ranked correctly by the assessment method. AUC values range from 0.0 to 1.0, where chance accuracy is 0.5 as indicated by diagonal line in the ROC curve. That is, a measure can be useful to the extent that its ROC curve is above the diagonal line (i.e., its discriminative accuracy is above chance).

Figure 7.15: Area under the receiver operating characteristic curve (AUC).

As an example, given an AUC of .75, this says that the overall score of an individual who has the characteristic in question will be higher 75% of the time than the overall score of an individual who does not have the characteristic. In lay terms, if we were to randomly pick one good and one bad outcome, what is the probability that we will classify correctly based on our instrument? AUC

Figure 7.16: Receiver operating characteristic curves (ROCs) for various levels of area under the ROC curve (AUC) for various measures.

is better than percent accuracy, because you can have high percent accuracy just by going with the base rate. AUC tells us how much better than chance a measure is at discriminating outcomes. AUC is useful as a measure of general discriminative accuracy, and tells us how accurate a measure is at all possible cutoffs Knowing the accuracy of a measure at all possible cutoffs can be helpful for selecting the optimal cutoff, given the goals of the assessment. In reality, however, we may not be interested in all cutoffs because not all errors are equal in their costs.

If we lower the base rate, we would need larger sample to get enough people to classify into each group. SDT/ROC methods are traditionally about dichotomous decisions (yes/no), not graded judgments. SDT/ROC methods can get messy with ordinal data that are more graded because you would have a a curve for each ordinal grouping.

7.2 Getting Started

7.2.1 Load Libraries

```
library("pROC")
library("rms")
library("ResourceSelection")
library("PredictABEL")
library("uroc")
library("rms")
library("gridExtra")
library("grid")
library("ggpubr")
library("msir")
library("viridis")
library("tidyverse")
library("here")
library("tinytex")
library("rmarkdown")
```

7.2.2 Prepare Data

7.2.2.1 Load Data

```
data(aSAH)
mydataSDT <- aSAH
```

7.2.2.2 Calculations

```
mydataSDT$testScore <- mydataSDT$s100b
mydataSDT$predictedProbability <- (mydataSDT$s100b - min(mydataSDT$s100b, na.rm = TRUE)) / (max(m
mydataSDT$continuousOutcome <- mydataSDT$testScore + rnorm(nrow(mydataSDT), mean = 0.20, sd = 0.2
mydataSDT$disorder <- NA
mydataSDT$disorder[mydataSDT$outcome == "Good"] <- 0
mydataSDT$disorder[mydataSDT$outcome == "Poor"] <- 1
```

7.2.2.3 Add Missing Data

Adding missing data to dataframes helps make examples more realistic to real-life data, and helps you get in the habit of programming to account for missing data.

```
mydataSDT$testScore[c(5,10)] <- NA
mydataSDT$disorder[c(10,15)] <- NA
```

7.3 Receiver Operating Characteristic (ROC) Curve

The receiver operating characteristic (ROC) curve shows the combination of hit rate (sensitivity) and false alarm rate (1−specificity) at every possible cutoff. It depicts that, as the cutoff increases (i.e., becomes more conservative), sensitivity decreases and specificity increases. It also depicts that, as the cutoff decreases (i.e., becomes more liberal), sensitivity increases and specificity decreases.

Receiver operating characteristic (ROC) curves were generated using the `pROC` package [Robin et al., 2021].

7.3.1 Empirical ROC Curve

```
rocCurve <- roc(data = mydataSDT, response = disorder, predictor = testScore, smooth = FALSE)
```

```
plot(rocCurve, legacy.axes = TRUE, print.auc = TRUE)
```

Figure 7.17: Empirical receiver operating characteristic curve. AUC = Area under the receiver operating characteristic curve.

7.3.2 Smooth ROC Curve

```
rocCurveSmooth <- roc(data = mydataSDT, response = disorder, predictor = testScore, smooth = TRUE
```

```
plot(rocCurveSmooth, legacy.axes = TRUE, print.auc = TRUE)
```

Figure 7.18: Smooth receiver operating characteristic curve. AUC = Area under the receiver operating characteristic curve.

7.3.3 Youden's J Statistic

The threshold at the Youden's J statistic is the threshold where the test has the maximum combination (i.e., sum) of sensitivity and specificity: $\max(\text{sensitivity} + \text{specificity} - 1)$

```
youdenJ <- coords(rocCurve, x = "best", best.method = "youden")
youdenJthreshold <- youdenJ$threshold
youdenJspecificity <- youdenJ$specificity
youdenJsensitivity <- youdenJ$sensitivity

youdenJ
```

```
##   threshold specificity sensitivity
## 1     0.205         0.8        0.65
```

For this test, the Youden's J Statistic is at a threshold of 0.205, where sensitivity is 0.65 and specificity is 0.8.

7.3.4 The point closest to the top-left part of the ROC curve with perfect sensitivity and specificity

The point closest to the top-left part of the ROC plot with perfect sensitivity and specificity: $\min[(1 - \text{sensitivity})^2 + (1 - \text{specificity})^2]$

```
closestTopLeft <- coords(rocCurve, x = "best", best.method = "closest.topleft")
closestTopLeftthreshold <- closestTopLeft$threshold
closestTopLeftspecificity <- closestTopLeft$specificity
closestTopLeftsensitivity <- closestTopLeft$sensitivity

closestTopLeft
```

```
##   threshold specificity sensitivity
## 1     0.205         0.8        0.65
```

For this test, the combination of sensitivity and specificity is closest to the top left of the ROC plot at a threshold of 0.205, where sensitivity is 0.65 and specificity is 0.8.

7.4 Prediction Accuracy Across Cutoffs

There are two primary dimensions of accuracy: (1) discrimination (e.g., sensitivity, specificity, area under the ROC curve) and (2) calibration. Some general

indexes of accuracy combine discrimination and calibration, as described in Section 7.4.1. This section (7.4) describes indexes of accuracy that span all possible cutoffs. That is, each index of accuracy described in this section provides a single numerical index of accuracy that aggregates the accuracy across all possible cutoffs. Aspects of accuracy at a particular cutoff are described in Section 7.5.

7.4.1 General Prediction Accuracy

There are many metrics of general prediction accuracy. When thinking about which metric(s) may be best for a given problem, it is important to consider the purpose of the assessment.

7.4.1.1 Scale-Dependent Accuracy Estimates

The estimates of prediction accuracy described in this section are scale-dependent. These accuracy estimates depend on the unit of measurement and therefore cannot be compared across measures with different scales or across data sets.

7.4.1.1.1 Mean Error Here, "error" (e) is the difference between the predicted and observed value. Mean error (also known as bias, see Section 3.5.1.2.2) is the mean difference between the predicted and observed values across individuals (i), that is, the mean of the errors across individuals (e_i). Values closer to zero reflect greater accuracy. If mean error is above zero, it indicates that predicted values are, on average, greater than observed values (i.e., over-estimating errors). If mean error is below zero, it indicates that predicted values are, on average, less than observed values (i.e., under-estimating errors). If both over-estimating and under-estimating errors are present, however, they can cancel each other out. As a result, even with a mean error of zero, there can still be considerable error present. Thus, although mean error can be helpful for examining whether predictions systematically under- or over-estimate the actual scores, other forms of accuracy are necessary to examine to examine the *extent* of error.

$$\begin{aligned}\text{mean error} &= \frac{\sum_{i=1}^{n}(\text{predicted}_i - \text{observed}_i)}{n} \\ &= \text{mean}(e_i)\end{aligned}$$

```
meanError <- function(predicted, actual){
  value <- mean(predicted - actual, na.rm = TRUE)
  return(value)
}
```

```
meanErrorValue <- meanError(mydataSDT$testScore, mydataSDT$disorder)
meanErrorValue
```

```
## [1] -0.1123636
```

7.4.1.1.2 Mean Absolute Error (MAE) Mean absolute error (MAE) is the mean of the absolute value of differences between the predicted and observed values across individuals, that is, the mean of the absolute value of errors. Smaller values (closer to zero) reflect greater accuracy.

$$\begin{aligned}\text{mean absolute error (MAE)} &= \frac{\sum_{i=1}^{n} |\text{predicted}_i - \text{observed}_i|}{n} \\ &= \text{mean}(|e_i|)\end{aligned}$$

```
meanAbsoluteError <- function(predicted, actual){
  value <- mean(abs(predicted - actual), na.rm = TRUE)
  return(value)
}

meanAbsoluteErrorValue <- meanAbsoluteError(mydataSDT$testScore, mydataSDT$disorder)
meanAbsoluteErrorValue
```

```
## [1] 0.3407273
```

7.4.1.1.3 Mean Squared Error (MSE) Smaller values (closer to zero) reflect greater accuracy.

$$\begin{aligned}\text{mean squared error (MSE)} &= \frac{\sum_{i=1}^{n} (\text{predicted}_i - \text{observed}_i)^2}{n} \\ &= \text{mean}(e_i^2)\end{aligned}$$

```
meanSquaredError = function(predicted, actual){
  value <- mean((predicted - actual)^2, na.rm = TRUE)
  return(value)
}

meanSquaredErrorValue <- meanSquaredError(mydataSDT$testScore, mydataSDT$disorder)
meanSquaredErrorValue
```

```
## [1] 0.2078273
```

7.4.1.1.4 Root Mean Squared Error (RMSE) Smaller values (closer to zero) reflect greater accuracy. RMSE penalizes larger errors more so than MAE does.

$$\begin{aligned}\text{root mean squared error (RMSE)} &= \sqrt{\frac{\sum_{i=1}^{n}(\text{predicted}_i - \text{observed}_i)^2}{n}} \\ &= \sqrt{\text{mean}(e_i^2)}\end{aligned}$$

```
rootMeanSquaredError = function(predicted, actual){
  value <- sqrt(mean((predicted - actual)^2, na.rm = TRUE))
  return(value)
}

rootMeanSquaredErrorValue <- rootMeanSquaredError(mydataSDT$testScore, mydataSDT$disorder)
rootMeanSquaredErrorValue
```

```
## [1] 0.4558808
```

7.4.1.2 Scale-Independent Accuracy Estimates

The estimates of prediction accuracy described in this section are intended to be scale-*independent* (unit-free) so the accuracy estimates can be compared across measures with different scales or across data sets [Hyndman and Athanasopoulos, 2018].

7.4.1.2.1 Mean Percentage Error (MPE) Values closer to zero reflect greater accuracy.

$$\text{percentage error } (p_i) = \frac{100\% \times (\text{observed}_i - \text{predicted}_i)}{\text{observed}_i}$$

$$\begin{aligned}\text{mean percentage error (MPE)} &= \frac{100\%}{n}\sum_{i=1}^{n}\frac{\text{observed}_i - \text{predicted}_i}{\text{observed}_i} \\ &= \text{mean(percentage error)} \\ &= \text{mean}(p_i)\end{aligned}$$

Note: mean percentage error is undefined when one or more of the observed values equals zero, due to division by zero. I provide the option in the function to drop undefined values so you can still generate an estimate of accuracy despite undefined values, but use this option at your own risk.

```
meanPercentageError = function(predicted, actual, dropUndefined = FALSE){
  percentageError <- 100 * (actual - predicted) / actual

  if(dropUndefined == TRUE){
    percentageError[!is.finite(percentageError)] <- NA
  }

  value <- mean(percentageError, na.rm = TRUE)
  return(value)
}

meanPercentageError(mydataSDT$testScore, mydataSDT$disorder)
```

```
## [1] -Inf
```

```
meanPercentageErrorValue <- meanPercentageError(mydataSDT$testScore, mydataSDT$disorder
meanPercentageErrorValue
```

```
## [1] 59.625
```

7.4.1.2.2 Mean Absolute Percentage Error (MAPE) Smaller Values (closer to zero) reflect greater accuracy.

$$\begin{aligned}\text{mean absolute percentage error (MAPE)} &= \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\text{observed}_i - \text{predicted}_i}{\text{observed}_i}\right| \\ &= \text{mean}(|\text{percentage error}|) \\ &= \text{mean}(|p_i|)\end{aligned}$$

Note: mean absolute percentage error is undefined when one or more of the observed values equals zero, due to division by zero. I provide the option in the function to drop undefined values so you can still generate an estimate of accuracy despite undefined values, but use this option at your own risk.

```
meanAbsolutePercentageError = function(predicted, actual, dropUndefined = FALSE){
  percentageError <- 100 * (actual - predicted) / actual

  if(dropUndefined == TRUE){
    percentageError[!is.finite(percentageError)] <- NA
  }

  value <- mean(abs(percentageError), na.rm = TRUE)
```

```
  return(value)
}

meanAbsolutePercentageError(mydataSDT$testScore, mydataSDT$disorder)
```

```
## [1] Inf
```

```
meanAbsolutePercentageErrorValue <- meanAbsolutePercentageError(mydataSDT$testScore, mydataSDT$di
meanAbsolutePercentageErrorValue
```

```
## [1] 64.975
```

7.4.1.2.3 Symmetric Mean Absolute Percentage Error (sMAPE)

Smaller values (closer to zero) reflect greater accuracy.

$$\text{symmetric mean absolute percentage error (sMAPE)} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|\text{predicted}_i - \text{observed}_i|}{|\text{predicted}_i| + |\text{observed}_i|}$$

Note: symmetric mean absolute percentage error is undefined when one or more of the individuals has a prediction–observed combination such that the sum of the absolute value of the predicted value and the absolute value of the observed value equals zero ($|\text{predicted}_i| + |\text{observed}_i|$), due to division by zero. I provide the option in the function to drop undefined values so you can still generate an estimate of accuracy despite undefined values, but use this option at your own risk.

```
symmetricMeanAbsolutePercentageError = function(predicted, actual, dropUndefined = FALSE){
  relativeError <- abs(predicted - actual)/(abs(predicted) + abs(actual))

  if(dropUndefined == TRUE){
    relativeError[!is.finite(relativeError)] <- NA
  }

  value <- 100 * mean(abs(relativeError), na.rm = TRUE)
  return(value)
}

symmetricMeanAbsolutePercentageErrorValue <- symmetricMeanAbsolutePercentageError(mydataSDT$testS
symmetricMeanAbsolutePercentageErrorValue
```

```
## [1] 82.45553
```

7.4.1.2.4 Mean Absolute Scaled Error (MASE) Mean absolute scaled error (MASE) is described by [Hyndman and Athanasopoulos, 2018]. Values closer to zero reflect greater accuracy.

The adapted formula for MASE with non-time series data is described here (https://stats.stackexchange.com/a/108963/20338), and is below:

$$\begin{aligned}\text{scaled error}(q_i) &= \frac{\text{observed}_i - \text{predicted}_i}{\text{scaling factor}} \\ &= \frac{\text{observed}_i - \text{predicted}_i}{\frac{1}{n}\sum_{i=1}^{n}|\text{observed}_i - \overline{\text{observed}}|}\end{aligned}$$

$$\begin{aligned}\text{mean absolute scaled error (MASE)} &= \frac{1}{n}\sum_{i=1}^{n}|q_i| \\ &= \text{mean}(|\text{scaled error}|) \\ &= \text{mean}(|q_i|)\end{aligned}$$

Note: mean absolute scaled error is undefined when the scaling factor is zero, due to division by zero. With non-time series data, the scaling factor is the average of the absolute value of individuals' observed scores minus the average observed score ($\frac{1}{n}\sum_{i=1}^{n}|\text{observed}_i - \overline{\text{observed}}|$).

```
meanAbsoluteScaledError <- function(predicted, actual){
  mydata <- data.frame(na.omit(cbind(predicted, actual)))

  errors <- mydata$actual - mydata$predicted
  scalingFactor <- mean(abs(mydata$actual - mean(mydata$actual)))
  scaledErrors <- errors/scalingFactor

  value <- mean(abs(scaledErrors))
  return(value)
}

meanAbsoluteScaledErrorValue <- meanAbsoluteScaledError(mydataSDT$testScore, mydataSDT$
meanAbsoluteScaledErrorValue
```

```
## [1] 0.7362143
```

7.4.1.2.5 Root Mean Squared Log of the Accuracy Ratio The squared log of the accuracy ratio is described by [Tofallis, 2015]. Squaring the log values keeps the values positive, such that smaller values (values closer to zero) reflect greater accuracy. Here, I take the mean of the squared log values, which keeps

the values positive, and I calculate the square root of the mean squared log values to put them back on the (pre-squared) log metric.

$$\text{accuracy ratio} = \frac{\text{predicted}_i}{\text{observed}_i}$$

$$\begin{aligned}\text{root mean squared log of the accuracy ratio} &= \sqrt{\sum_{i=1}^{n} log\left(\frac{\text{predicted}_i}{\text{observed}_i}\right)^2} \\ &= \sqrt{\text{mean}\left[log\left(\frac{\text{predicted}_i}{\text{observed}_i}\right)^2\right]} \\ &= \sqrt{\text{mean}[log(\text{accuracy ratio})^2]}\end{aligned}$$

There are other variations of prediction accuracy metrics that use the log of the accuracy ratio. One variation makes it similar to median symmetric percentage error [Morley et al., 2018].

Note: the log of the accuracy ratio is undefined when one or more predicted values or actual values equals zero. When predicted values are zero, there is a zero in the numerator of the accuracy ratio, which leads to $log(0)$, which is undefined. When actual values are zero, there is division by zero in the accuracy ratio. I provide the option in the function to drop undefined values so you can still generate an estimate of accuracy despite undefined values, but use this option at your own risk.

```
rootMeanSquaredLogAccuracyRatio <- function(predicted, actual, dropUndefined = FALSE){
  logAccuracyRatio <- log(predicted / actual)

  if(dropUndefined == TRUE){
    logAccuracyRatio[!is.finite(logAccuracyRatio)] <- NA
  }

  value <- sqrt(mean(logAccuracyRatio^2, na.rm = TRUE))

  return(value)
}

rootMeanSquaredLogAccuracyRatio(mydataSDT$testScore, mydataSDT$disorder)
```

```
## [1] Inf
```

```
rootMeanSquaredLogAccuracyRatioValue <- rootMeanSquaredLogAccuracyRatio(mydataSDT$testS
rootMeanSquaredLogAccuracyRatioValue
```

```
## [1] 1.606175
```

7.4.1.2.6 Coefficient of Determination (R^2) The coefficient of determination (R^2) is a general index of accuracy that combines both discrimination and calibration. It reflects the proportion of variance in the outcome (dependent) variable that is explained by the model predictions. Larger values reflect greater accuracy.

R^2:

```
rSquaredValue <- summary(lm(disorder ~ testScore, data = mydataSDT))$r.squared
rSquaredValue
```

```
## [1] 0.1778746
```

The predictor (`testScore`) explains 17.79% of the variance ($R^2 = 0.18$) in the outcome (`disorder` status).

7.4.1.2.6.1 Adjusted R^2 (R^2_{adj})

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

where p is the number of predictors in the model, and n is the sample size.

```
rSquaredAdjValue <- summary(lm(disorder ~ testScore, data = mydataSDT))$adj.r.squared
rSquaredAdjValue
```

```
## [1] 0.1702623
```

7.4.1.2.6.2 Predictive R^2 Predictive R^2 is described here: https://tomhopper.me/2014/05/16/can-we-do-better-than-r-squared/.

```
#predictive residual sum of squares (PRESS)
PRESS <- function(linear.model) {
  #calculate the predictive residuals
  pr <- residuals(linear.model)/(1-lm.influence(linear.model)$hat)
  #calculate the PRESS
  PRESS <- sum(pr^2)
```

```
  return(PRESS)
}

predictiveRSquared <- function(predicted, actual){
  #fit linear model
  linear.model <- lm(actual ~ predicted)

  #use anova() to get the sum of squares for the linear model
  lm.anova <- anova(linear.model)

  #calculate the total sum of squares
  tss <- sum(lm.anova$'Sum Sq')

  #calculate the predictive R^2
  value <- 1 - PRESS(linear.model)/(tss)

  return(value)
}

predictiveRSquaredValue <- predictiveRSquared(mydataSDT$testScore, mydataSDT$disorder)
predictiveRSquaredValue
```

```
## [1] 0.1190976
```

7.4.2 Discrimination

When dealing with a categorical outcome, discrimination is the ability to separate events from non-events. When dealing with a continuous outcome, discrimination is the strength of the association between the predictor and the outcome.

7.4.2.1 Area under the ROC curve (AUC)

The area under the ROC curve (AUC) is a general index of discrimination accuracy for a categorical outcome. It is also called the concordance (c) statistic. Larger values reflect greater discrimination accuracy. AUC was estimated using the `pROC` package [Robin et al., 2021]. Aspects of discrimination at a particular cutoff (e.g., sensitivity, specificity) are described in Section 7.5.

```
aucValue <- rocCurve$auc
aucValue
```

```
## Area under the curve: 0.7312
```

7.4.2.2 Coefficient of Predictive Ability (CPA)

The coefficient of predictive ability (CPA) is a generalization of AUC to handle non-binary outcomes including ordinal and continuous outcomes [Gneiting and Walz, 2020]. It was calculated using the `uroc` package [Gneiting and Walz, 2020].

```
cpaNoMissing <- na.omit(mydataSDT[,c("testScore","continuousOutcome")])
cpa(response = cpaNoMissing$continuousOutcome, predictor = cpaNoMissing$testScore)
```

```
## [1] 0.830537
```

7.4.2.3 Spearman's Rho (ρ) Rank Correlation

When the coefficient of predictive ability is applied to an ordinal or continuous outcome, it is linearly related to Spearman's rho (ρ) rank correlation [Gneiting and Walz, 2020]. Spearman's rho was estimated in the `rms` package [Harrell, Jr., 2021].

```
cor(x = mydataSDT$testScore, y = mydataSDT$continuousOutcome, use = "pairwise.complete
```

```
## [1] 0.6617247
```

```
orm(continuousOutcome ~ testScore, data = mydataSDT)$stats["rho"]
```

```
##       rho
## 0.6617247
```

7.4.2.4 Somers' D_{xy} Rank Correlation

Somers D_{xy} is an index of discrimination for an ordinal outcome [Harrell, 2015]. Somers' D_{xy} was estimated in the `rms` package [Harrell, Jr., 2021].

```
lrm(continuousOutcome ~ testScore, data = mydataSDT)$stats["Dxy"]
```

```
##       Dxy
## 0.4761671
```

```
rms::validate(lrm(continuousOutcome ~ testScore, data = mydataSDT, x = TRUE, y = TRUE)
```

```
## [1] 0.4761671
```

7.4.2.5 Kendall's Tau-a (τ_A) Rank Correlation

Kendall's tau-a (τ_A) is an index of the discrimination for an ordinal outcome [Harrell, 2015]. Kendall's tau-a was estimated in the `rms` package [Harrell, Jr., 2021].

```
cor(x = mydataSDT$testScore, y = mydataSDT$continuousOutcome, use = "pairwise.complete.obs", meth
```

```
## [1] 0.4831013
```

```
lrm(continuousOutcome ~ testScore, data = mydataSDT)$stats["Tau-a"]
```

```
##     Tau-a
## 0.4761671
```

7.4.2.6 c Index

The c index, also called the concordance probability, is a generalization of the area under the ROC curve that applies to a continuous outcome [Harrell, 2015]. The c index is the proportion of all pairs of predicted-actual values whose actual value can be ordered such that the pair with the higher predicted value is the one who had the higher actual value. The Brown–Hollander–Korwar nonparametric test of association was estimated in the `rms` package [Harrell, Jr., 2021].

```
lrm(continuousOutcome ~ testScore, data = mydataSDT)$stats["C"]
```

```
##         C
## 0.7380835
```

7.4.2.7 Effect Size (β) of Regression

The effect size of a predictor, i.e., the standardized regression coefficient called beta (β), is a general index of discrimination accuracy for a continuous outcome. We can obtain standardized regression coefficients by standardizing the predictors and outcome using the `scale()` function in R.

β:

```
regressionBeta <- lm(scale(continuousOutcome) ~ scale(testScore), data = mydataSDT)$coef["scale(t

regressionBeta
```

```
## scale(testScore)
##        0.8101931
```

7.4.3 Calibration

When dealing with a categorical outcome, calibration is the degree to which a probabilistic estimate of an event reflects the true underlying probability of the event. When dealing with a continuous outcome, calibration is the degree to which the predicted values are close in value to the outcome values. The importance of examining calibration (in addition to discrimination) is described by Lindhiem et al. [2020].

Calibration came to be considered a central aspect of weather forecast accuracy. For instance, on the days that the meteorologist says there is a 60% chance of rain, it should rain about 60% of the time. Through improvements in scientific understanding of weather systems, rain forecasts have become more accurate. For instance, rain forecasts from the National Weather Service are well calibrated [see Figure 7.20, Charba and Klein, 1980]. However, forecasts of rain may be exaggerated by local TV meteoreologists to boost ratings [Silver, 2012]. Interestingly, rain forecasts from The Weather Channel are somewhat miscalibrated under certain conditions. For instance, on days in which they forecast a 20% chance of rain, the actual chance of rain is around 5% [see Figure 7.19, Bickel and Kim, 2008]. However, this miscalibration is deliberate. People tend to be more angry when the meteorologist says it will not rain when it actually does (false negative) compared to when the meteorologist says it will rain when it actually does not (false positive). As Silver [2012] notes, "If it rains when it isn't supposed to, [people] curse the weatherman for ruining their picnic, whereas an unexpectedly sunny day is taken as a serendipitous bonus. It isn't good science, but as Dr. Rose at the Weather Channel acknowledged to [Mr. Silver]: 'If the forecast was objective, if it has zero bias in precipitation, we'd probably be in trouble.' " (p. 135).

Calibration is not just important for weather forecasts. It is also important for psychological assessment. Calibration can be examined in several ways, including Brier Scores (see Section 7.4.3.2), the Hosler-Lemeshow Test (see Section 7.4.3.3), Spiegelhalter's z (see Section 7.4.3.4), and the mean difference between predicted and observed values at different binned thresholds as depicted graphically with a calibration plot (see Figure 7.22).

7.4.3.1 Calibration Plot

Calibration plots can be helpful for identifying miscalibration. A calibration plot depicts the predicted probability of an event on the x-axis, and the actual (observed) probability of the event on the y-axis. The predictions are binned into a certain number of groups (commonly 10). The diagonal line reflects predictions that are perfectly calibrated. To the extent that predictions deviate from the diagonal line, the predictions are miscalibrated. There are four general patterns of miscalibration: overextremity, underextremity, overprediction, and underprediction (see Figure 7.21). Overextremity exists when the predicted

Figure 7.19: Calibration plot of same-day probability of precipitation (PoP) forecasts from The Weather Channel. Figure reprinted from @Bickel2008, Figure 2, p. 4872. Bickel, J. E., & Kim, S. D. (2008). Verification of the weather channel probability of precipitation forecasts. *Monthly Weather Review*, *136*(12), 4867–4881. doi: https://doi.org/10.1175/2008MWR2547.1 Copyright (c) American Meteorological Society. Used with permission.

Figure 7.20: Calibration plot of local probability of precipitation (PoP) forecasts for 87 stations from the United States National Weather Service. Numbers next to the plotted points are the sample sizes. Figure reprinted from @Charba1980, Figure 6, p. 1550. Charba, J. P., & Klein, W. H. (1980). Skill in precipitation forecasting in the National Weather Service. *Bulletin of the American Meteorological Society*, *61*(12), 1546-1555. https://doi.org/10.1175/1520-0477(1980)061<1546:SIPFIT>2.0.CO;2. Copyright (c) American Meteorological Society. Used with permission.

probabilites are too close to the extremes (zero or one). Underextremity exists when the predicted probabilities are too far away from the extremes. Overprediction exists when the predicted probabilities are consistently greater than the observed probabilities. Underprediction exists when the predicted probabilities are consistently less than the observed probabilities. For a more thorough description of these types of miscalibration, see Lindhiem et al. [2020].

Figure 7.21: Types of miscalibration

This calibration plot was generated using the `PredictABEL` package [Kundu et al., 2020].

```
colNumberOutcome <- which(names(mydataSDT) == "disorder")
myDataNoMissing <- na.omit(mydataSDT)
```

```
plotCalibration(data = na.omit(myDataNoMissing), cOutcome = colNumberOutcome, predRisk
```

Figure 7.22: Calibration plot 1

```
## $Table_HLtest
##                  total meanpred meanobs predicted observed
## [0.0000,0.0245)     19    0.013   0.211      0.25        4
## 0.0245               7    0.025   0.143      0.17        1
## 0.0294               8    0.029   0.250      0.24        2
## [0.0343,0.0441)     13    0.036   0.231      0.47        3
## [0.0441,0.0588)     10    0.051   0.300      0.51        3
## [0.0588,0.0735)     10    0.063   0.100      0.63        1
## [0.0735,0.1324)     10    0.099   0.500      0.99        5
```

```
## [0.1324,0.2059]    12    0.165   0.583      1.98        7
## [0.2059,0.2598]    10    0.222   0.300      2.22        3
## [0.2598,1.0000]    11    0.408   1.000      4.49       11
##
## $Chi_square
## [1] 152.274
##
## $df
## [1] 8
##
## $p_value
## [1] 0
```

This calibration plot (and ROC curve) was generated using the `rms` package [Harrell, Jr., 2021].

```
val.prob(mydataSDT$predictedProbability, mydataSDT$disorder)
```

```
##           Dxy       C (ROC)            R2             D      D:Chi-sq
##  4.769231e-01  7.384615e-01 -6.156954e-01 -3.797827e-01 -4.039632e+01
##           D:p             U      U:Chi-sq           U:p             Q
##            NA           Inf           Inf  0.000000e+00          -Inf
##         Brier     Intercept         Slope          Emax           E90
##  2.659086e-01  1.682881e+00  8.857501e-01  7.146778e-01  3.719585e-01
##          Eavg           S:z           S:p
##  2.618115e-01  1.052663e+01  6.512514e-26
```

This calibration plot was adapted from Darren Dahly's example here: https://darrendahly.github.io/post/homr/

```
g1 <- mutate(mydataSDT, bin = cut_number(predictedProbability, 10)) %>%
  # Bin prediction into 10ths
  group_by(bin) %>%
  mutate(n = length(na.omit(predictedProbability)), # Get ests and CIs
         bin_pred = mean(predictedProbability, na.rm = TRUE),
         bin_prob = mean(disorder, na.rm = TRUE),
         se = sd(disorder, na.rm = TRUE) / sqrt(n),
         ul = bin_prob + qnorm(.975) * se,
         ll = bin_prob - qnorm(.975) * se) %>%
  ungroup() %>%
  ggplot(aes(x = bin_pred, y = bin_prob, ymin = ll, ymax = ul)) +
  geom_pointrange(size = 0.5, color = "black") +
  scale_y_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.1)) +
  scale_x_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.1)) +
```

Figure 7.23: Calibration plot 2

```
  geom_abline() + # 45 degree line indicating perfect calibration
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed",
              color = "black", formula = y~-1 + x) +
  # straight line fit through estimates
  geom_smooth(aes(x = predictedProbability, y = disorder),
              color = "red", se = FALSE, method = "loess") +
  # loess fit through estimates
  xlab("") +
  ylab("Observed Probability") +
  theme_minimal()+
  xlab("Predicted Probability")

g2 <- ggplot(mydataSDT, aes(x = predictedProbability)) +
  geom_histogram(fill = "black", bins = 200) +
  scale_x_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.1)) +
  xlab("Histogram of Predicted Probability") +
  ylab("") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

g <- arrangeGrob(g1, g2, respect = TRUE, heights = c(1, 0.25), ncol = 1)
```

```
grid.arrange(g)
```

7.4.3.2 Brier Scores

Brier scores were calculated using the `rms` package [Harrell, Jr., 2021]. Smaller values reflect greater calibration accuracy.

```
brierValue <- val.prob(mydataSDT$predictedProbability, mydataSDT$disorder, pl = FALSE)["Brier"]
brierValue
```

```
##     Brier
## 0.2659086
```

7.4.3.3 Hosler-Lemeshow Test

The Hosler-Lemeshow Goodness of Fit (GOF) Test requires specifying how many groups/bins (g) to use to calculate quantiles. Smaller χ^2 values reflect greater calibration accuracy. The Hosler-Lemeshow GOF test was calculated using the `ResourceSelection` package [Lele et al., 2019]. The table of predicted versus expected frequencies was generated using the `PredictABEL` package [Kundu et al., 2020].

Figure 7.24: Calibration Plot 3

$g = 2$

```
gValue <- 2

hoslem.test(mydataSDT$disorder, mydataSDT$predictedProbability, g = gValue)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mydataSDT$disorder, mydataSDT$predictedProbability
## X-squared = 124.17, df = 0, p-value < 2.2e-16
```

```
plotCalibration(data = na.omit(myDataNoMissing), cOutcome = colNumberOutcome, predRisk = myDataNo
```

```
##                  total meanpred meanobs predicted observed
## [0.0000,0.0588]     57    0.029   0.228      1.64       13
## [0.0588,1.0000]     53    0.194   0.509     10.29       27
```

$g = 4$

```
gValue <- 4

hoslem.test(mydataSDT$disorder, mydataSDT$predictedProbability, g = gValue)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mydataSDT$disorder, mydataSDT$predictedProbability
## X-squared = 135.69, df = 2, p-value < 2.2e-16
```

```
plotCalibration(data = na.omit(myDataNoMissing), cOutcome = colNumberOutcome, predRisk = myDataNo
```

```
##                  total meanpred meanobs predicted observed
## [0.0000,0.0343]     34    0.019   0.206      0.66        7
## [0.0343,0.0588]     23    0.043   0.261      0.98        6
## [0.0588,0.1569]     26    0.095   0.346      2.48        9
## [0.1569,1.0000]     27    0.289   0.667      7.81       18
```

$g = 6$

```
gValue <- 6

hoslem.test(mydataSDT$disorder, mydataSDT$predictedProbability, g = gValue)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mydataSDT$disorder, mydataSDT$predictedProbability
## X-squared = 144.82, df = 4, p-value < 2.2e-16
```

```
plotCalibration(data = na.omit(myDataNoMissing), cOutcome = colNumberOutcome, predRisk
```

```
##                  total meanpred meanobs predicted observed
## [0.0000,0.0245]     19    0.013   0.211      0.25        4
## [0.0245,0.0392]     23    0.030   0.217      0.68        5
## [0.0392,0.0588]     15    0.047   0.267      0.71        4
## [0.0588,0.1127]     17    0.074   0.235      1.26        4
## [0.1127,0.2206]     19    0.167   0.474      3.18        9
## [0.2206,1.0000]     17    0.344   0.824      5.85       14
```

$g = 8$

```
gValue <- 8

hoslem.test(mydataSDT$disorder, mydataSDT$predictedProbability, g = gValue)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mydataSDT$disorder, mydataSDT$predictedProbability
## X-squared = 154.48, df = 6, p-value < 2.2e-16
```

```
plotCalibration(data = na.omit(myDataNoMissing), cOutcome = colNumberOutcome, predRisk
```

```
##                  total meanpred meanobs predicted observed
## [0.0000,0.0245]     19    0.013   0.211      0.25        4
## [0.0245,0.0343]     15    0.027   0.200      0.41        3
## 0.0343               8    0.034   0.250      0.27        2
## [0.0392,0.0588]     15    0.047   0.267      0.71        4
## [0.0588,0.0931]     13    0.066   0.077      0.86        1
## [0.0931,0.1569]     13    0.124   0.615      1.62        8
## [0.1569,0.2402]     15    0.206   0.400      3.09        6
## [0.2402,1.0000]     12    0.394   1.000      4.73       12
```

$g = 10$

```
gValue <- 10

hoslem.test(mydataSDT$disorder, mydataSDT$predictedProbability, g = gValue)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mydataSDT$disorder, mydataSDT$predictedProbability
## X-squared = 153.94, df = 8, p-value < 2.2e-16
```

```
plotCalibration(data = na.omit(myDataNoMissing), cOutcome = colNumberOutcome, predRisk = myDataNo
```

```
##                  total meanpred meanobs predicted observed
## [0.0000,0.0245)     19    0.013   0.211      0.25        4
## 0.0245               7    0.025   0.143      0.17        1
## 0.0294               8    0.029   0.250      0.24        2
## [0.0343,0.0441)     13    0.036   0.231      0.47        3
## [0.0441,0.0588)     10    0.051   0.300      0.51        3
## [0.0588,0.0735)     10    0.063   0.100      0.63        1
## [0.0735,0.1324)     10    0.099   0.500      0.99        5
## [0.1324,0.2059)     12    0.165   0.583      1.98        7
## [0.2059,0.2598)     10    0.222   0.300      2.22        3
## [0.2598,1.0000]     11    0.408   1.000      4.49       11
```

7.4.3.4 Spiegelhalter's z

Spiegelhalter's z was calculated using the `rms` package [Harrell, Jr., 2021]. Smaller z values (and larger associated p-values) reflect greater calibration accuracy.

```
val.prob(mydataSDT$predictedProbability, mydataSDT$disorder, pl = FALSE)["S:z"]; val.prob(mydataS
```

```
##      S:z
## 10.52663
```

```
##          S:p
## 6.512514e-26
```

7.4.3.5 Calibration for predicting a continuous outcome

When predicting a continuous outcome, calibration of the predicted values in relation to the outcome values can be examined in multiple ways including:

- in a calibration plot, the extent to which the intercept is near zero and the slope is near one
- in a calibration plot, the extent to which the 95% confidence interval of the observed value, across all values of the predicted values, includes the diagonal reference line with an intercept of zero and a slope of one.
- mean error
- mean absolute error
- mean squared error
- root mean squared error

With a plot of the predictions on the x-axis, and the outcomes on the y-axis (i.e., a calibration plot), calibration can be examined graphically as the extent to which the best-fit regression line has an intercept (alpha) close to zero and a slope (beta) close to one [Steyerberg and Vergouwe, 2014, Stevens and Poppe, 2020]. The intercept is also called "calibration-in-the-large", whereas "calibration-in-the-small" refers to the extent to which the predicted values match the observed values at a specific predicted value (e.g., when the weather forecaster says that there is a 10% chance of rain, does it actually rain 10% of the time?). For predictions to be well calibrated, the intercept should be close to zero and the slope should be close to one. If the slope is close to one but the intercept is not close to zero (or the intercept is close to zero but the slope is not close to one), the predictions would not be considered well calibrated. Likewise, if the intercept is close to zero but the slope is not closer to zero, the predictions would not be well calibrated. The 95% confidence interval of the observed value, across all values of the predicted values, should include the diagonal reference line whose intercept is zero and whose slope is one.

For instance, based on the intercept and slope of the calibration plot in Figure 7.25, the predictions are not well calibrated, despite having a slope near one, because the 95% confidence interval of the intercept does not include zero. The best-fit line is the yellow line. The intercept from the best-fit line is positive, as shown in the regression equation. This is a case of underprediction, where the predicted values are consistently less than the observed values. The confidence interval of the observed value (i.e., the purple band) is the interval within which we have 95% confidence that the true observed value would lie for a given predicted value, based on the model. The 95% prediction interval of the observed value (i.e., the dashed red lines) is the interval within which we would expect that 95% of future observations would lie for a given predicted value. The black diagonal line indicates the reference line with an intercept of zero and a slope of one. The predictions would be significantly miscalibrated at a given level of the predicted values if the 95% confidence interval of the observed value does

not include the reference line at that level of the predicted value. In this case, the 95% confidence interval of the observed value does not include the reference line (i.e., the actual observed value) at any level of the predicted values, so the predictions are miscalibrated at every level of the predicted values.

```
#95 prediction interval based on linear model
calibrationModel <- lm(continuousOutcome ~ testScore, data = mydataSDT)
calibrationModelPredictionInterval <- expand.grid(testScore = seq(from = min(mydataSDT$testScore,
calibrationModelPredictionInterval <- cbind(calibrationModelPredictionInterval, data.frame(predic

ggplot(mydataSDT, aes(x = testScore, y = continuousOutcome)) +
  geom_point() +
  geom_line(data = calibrationModelPredictionInterval, aes(y = lwr), color = "red", linetype = "d
  geom_line(data = calibrationModelPredictionInterval, aes(y = upr), color = "red", linetype = "d
  #geom_ribbon(data = calibrationModelPredictionInterval, aes(y = fit, ymin = lwr, ymax = upr), f
  geom_smooth(method = "lm", color = viridis(3)[3], fill = viridis(3)[1], alpha = 0.7) + #95% con
  geom_abline(slope = 1, intercept = 0) +
  xlim(0,2.4) +
  ylim(0,2.4) +
  xlab("Predicted Value") +
  ylab("Observed Value") +
  stat_cor(label.y = 2.2, aes(label = paste(..rr.label..))) +
  stat_regline_equation(label.y = 2.0) +
  theme_bw()
```

Gold-standard recommendations include examining the predicted values in relation to the observed values using a LOESS smoother [Austin and Steyerberg, 2014], such as in Figure 7.26. We can examine whether the LOESS-based 95% confidence interval of the observed value at every level of the predicted values includes the diagonal reference line (i.e., the actual observed value). In this case, the 95% confidence interval of the observed value does not include the reference line at any level of the predicted values, so the predictions are miscalibrated at every level of the predicted values.

```
#95 prediction interval based on LOESS model
calibrationLoessModel <- loess.sd(x = mydataSDT$testScore, y = mydataSDT$continuousOutcome, nsigm
calibrationLoessPredictionInterval <- data.frame(x = calibrationLoessModel$x, y = calibrationLoes

ggplot(mydataSDT, aes(x = testScore, y = continuousOutcome)) +
  geom_point() +
  geom_line(data = calibrationLoessPredictionInterval, aes(x = x, y = lower), color = "red", line
  geom_line(data = calibrationLoessPredictionInterval, aes(x = x, y = upper), color = "red", line
  #geom_ribbon(data = calibrationLoessPredictionInterval, aes(x = x, y = y, ymin = lower, ymax =
  geom_smooth(method = "loess", color = viridis(3)[3], fill = viridis(3)[1], alpha = 0.7) + #95%
  geom_abline(slope = 1, intercept = 0) +
```

Figure 7.25: Calibration Plot for Predictions of a Continuous Outcome. The black diagonal line indicates the reference line with an intercept of zero and a slope of one. The yellow line is the best-fit line. The purple band is the 95% confidence interval of the observed value. The dashed red lines are the 95% prediction interval of the observed value. The predictions are not well calibrated because the 95% confidence interval of the intercept does not include zero (even though the 95% confidence interval of the slope includes one). The intercept from the best-fit line is positive, as shown in the regression equation. This is a case of underprediction, where the predicted values are consistently less than the observed values. The 95% confidence interval of the observed value does not include the reference line (i.e., the actual observed value) at any level of the predicted values, so the predictions are miscalibrated at every level of the predicted values.

```
  xlim(0,2.4) +
  ylim(0,2.4) +
  xlab("Predicted Value") +
  ylab("Observed Value") +
  theme_bw()
```

7.5 Prediction Accuracy at a Given Cutoff

7.5.1 Set a Cutoff

Here, I set a cutoff at the Youden's J Statistic to calculate the accuracy statistics at that cutoff:

```
cutoff <- 0.205

mydataSDT$diagnosis <- NA
mydataSDT$diagnosis[mydataSDT$testScore < cutoff] <- 0
mydataSDT$diagnosis[mydataSDT$testScore >= cutoff] <- 1

mydataSDT$diagnosisFactor <- factor(mydataSDT$diagnosis, levels = c(1, 0), labels = c("Decision:
mydataSDT$disorderFactor <- factor(mydataSDT$disorder, levels = c(1, 0), labels = c("Truth: Disor
```

7.5.2 Confusion Matrix aka 2x2 Accuracy Table aka Cross-Tabulation aka Contingency Table

7.5.2.1 Number

```
table(mydataSDT$diagnosisFactor, mydataSDT$disorderFactor)
```

```
##
##                          Truth: Disorder Truth: No Disorder
##   Decision: Diagnosis                 26                 14
##   Decision: No Diagnosis              14                 56
```

7.5.2.2 Number with margins added

```
addmargins(table(mydataSDT$diagnosisFactor, mydataSDT$disorderFactor))
```

Figure 7.26: Calibration Plot for Predictions of a Continuous Outcome. The yellow line is the best-fit line based on a LOESS smoother. The purple band is the 95% confidence interval of the observed value. The dashed red lines are the 95% prediction interval of the observed value. The 95% confidence interval of the observed value does not include the reference line (i.e., the actual observed value) at lower levels of the predicted values, so the predictions are miscalibrated at lower levels of the predicted values.

```
##
##                          Truth: Disorder Truth: No Disorder Sum
##   Decision: Diagnosis                 26                 14  40
##   Decision: No Diagnosis              14                 56  70
##   Sum                                 40                 70 110
```

7.5.2.3 Proportions

```
prop.table(table(mydataSDT$diagnosisFactor, mydataSDT$disorderFactor))
```

```
##
##                          Truth: Disorder Truth: No Disorder
##   Decision: Diagnosis          0.2363636          0.1272727
##   Decision: No Diagnosis       0.1272727          0.5090909
```

7.5.2.4 Proportions with margins added

```
addmargins(prop.table(table(mydataSDT$diagnosisFactor, mydataSDT$disorderFactor)))
```

```
##
##                          Truth: Disorder Truth: No Disorder       Sum
##   Decision: Diagnosis          0.2363636          0.1272727 0.3636364
##   Decision: No Diagnosis       0.1272727          0.5090909 0.6363636
##   Sum                          0.3636364          0.6363636 1.0000000
```

7.5.3 True Positives (TP)

True positives (TPs) are also called valid positives (VPs) or hits. Higher values reflect greater accuracy.

```
TPvalue <- length(which(mydataSDT$diagnosis == 1 & mydataSDT$disorder == 1))
TPvalue
```

```
## [1] 26
```

7.5.4 True Negatives (TN)

True negatives (TNs) are also called valid negatives (VNs) or correct rejections. Higher values reflect greater accuracy.

```
TNvalue <- length(which(mydataSDT$diagnosis == 0 & mydataSDT$disorder == 0))
TNvalue
```

```
## [1] 56
```

7.5.5 False Positives (FP)

False positives (FPs) are also called false alarms (FAs). Lower values reflect greater accuracy.

```
FPvalue <- length(which(mydataSDT$diagnosis == 1 & mydataSDT$disorder == 0))
FPvalue
```

```
## [1] 14
```

7.5.6 False Negatives (FN)

False negatives (FNs) are also called misses. Lower values reflect greater accuracy.

```
FNvalue <- length(which(mydataSDT$diagnosis == 0 & mydataSDT$disorder == 1))
FNvalue
```

```
## [1] 14
```

7.5.7 Sample Size (N)

```
sampleSize <- function(TP, TN, FP, FN){
  value <- TP + TN + FP + FN

  return(value)
}

sampleSizeValue <- sampleSize(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
sampleSizeValue
```

```
## [1] 110
```

7.5.8 Selection Ratio (SR)

The selection ratio is the marginal probability of selection, independent of other things: $P(R_i)$. In clinical psychology, the selection ratio is how many people test positive for the disorder.

$$\begin{aligned} \text{SR} &= P(R_i) \\ &= \frac{\text{TP} + \text{FP}}{n} \end{aligned}$$

```
selectionRatio <- function(TP, TN, FP, FN){
  N <- TP + TN + FP + FN
  value <- (TP + FP)/N

  return(value)
}

selectionRatioValue <- selectionRatio(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
selectionRatioValue
```

```
## [1] 0.3636364
```

7.5.9 Base Rate (BR)

The base rate of a classification is its marginal probability, independent of other things: $P(C_i)$. In clinical psychology, the base rate of a disorder is its prevalence in the population. Without additional information, the base rate is used as the initial *pretest probability*.

$$\begin{aligned} \text{BR} &= P(C_i) \\ &= \frac{\text{TP} + \text{FN}}{n} \end{aligned}$$

```
baseRate <- function(TP, TN, FP, FN){
  N <- TP + TN + FP + FN
  value <- (TP + FN)/N

  return(value)
}

baseRateValue <- baseRate(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
baseRateValue
```

```
## [1] 0.3636364
```

7.5.10 Pretest Odds

The pretest odds of a classification can be calculated using the pretest probability (i.e., base rate). To convert a probability to odds, divide the probability by one minus that probability.

$$\text{pretest odds} = \frac{\text{pretest probability}}{1 - \text{pretest probability}}$$

```
pretestOdds <- function(TP, TN, FP, FN, pretestProb = NULL){
  if(!is.null(pretestProb)){
    pretestProbability <- pretestProb
  } else {
    N <- TP + TN + FP + FN
    pretestProbability <- (TP + FN)/N
  }

  value <- pretestProbability / (1 - pretestProbability)

  return(value)
}

pretestOddsValue <- pretestOdds(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue
pretestOddsValue
```

```
## [1] 0.5714286
```

```
pretestOdds(pretestProb = baseRate(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNval
```

```
## [1] 0.5714286
```

7.5.11 Percent Accuracy

Percent Accuracy is also called overall accuracy. Higher values reflect greater accuracy. Percent accuracy has several problems. First, it treats all errors (FP and FN) as equally important. However, in practice, it is rarely the case that FP and FN are equally important. Second, percent accuracy can be misleading because it is highly influenced by base rates. You can have a high percent accuracy by predicting from the base rate and saying that no one has the characteristic (if the base rate is low) or that everyone has the characteristic (if the base rate is high). Thus, it is also important to consider other aspects of accuracy.

$$\text{Percent Accuracy} = 100\% \times \frac{\text{TP} + \text{TN}}{n}$$

```
percentAccuracy <- function(TP, TN, FP, FN){
  N <- TP + TN + FP + FN
  value <- 100 * ((TP + TN)/N)

  return(value)
}

percentAccuracyValue <- percentAccuracy(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
percentAccuracyValue
```

```
## [1] 74.54545
```

7.5.12 Percent Accuracy by Chance

$$\begin{aligned}\text{Percent Accuracy by Chance} &= 100\% \times P(\text{TP}) + P(\text{TN}) \\ &= 100\% \times \{(\text{BR} \times \text{SR}) + [(1 - \text{BR}) \times (1 - \text{SR})]\}\end{aligned}$$

```
percentAccuracyByChance <- function(TP, TN, FP, FN){
  N <- TP + TN + FP + FN
  BR <- (TP + FN)/N
  SR <- (TP + FP)/N
  value <- 100 * ((BR * SR) + ((1 - BR) * (1 - SR)))

  return(value)
}

percentAccuracyByChanceValue <- percentAccuracyByChance(TP = TPvalue, TN = TNvalue, FP = FPvalue,
percentAccuracyByChanceValue
```

```
## [1] 53.71901
```

7.5.13 Percent Accuracy Predicting from the Base Rate

Predicting from the base rate is going with the most likely outcome in every prediction. It is also called "betting from the base rate." If the base rate is less than .50, it would involve predicting that the condition is absent for every case. If the base rate is .50 or above, it would involve predicting that the condition is present for every case. Predicting from the base rate is a special case of percent accuracy by chance when the selection ratio is set to either one (if the base rate $\geq .5$) or zero (if the base rate $< .5$).

```
percentAccuracyPredictingFromBaseRate <- function(TP, TN, FP, FN){
  N <- TP + TN + FP + FN
  BR <- (TP + FN)/N

  if(BR >= .5){
    SR <- 1
  } else if(BR < .5){
    SR <- 0
  }

  value <- 100 * ((BR * SR) + ((1 - BR) * (1 - SR)))

  return(value)
}

percentAccuracyPredictingFromBaseRateValue <- percentAccuracyPredictingFromBaseRate(TP
percentAccuracyPredictingFromBaseRateValue
```

```
## [1] 63.63636
```

7.5.14 Relative Improvement Over Chance (RIOC)

Relative improvement over chance (RIOC) is a prediction's improvement over chance as a proportion of the maximum possible improvement over chance, as described by [Farrington and Loeber, 1989]. Higher values reflect greater accuracy.

$$\text{relative improvement over chance (RIOC)} = \frac{\text{total correct} - \text{chance correct}}{\text{maximum correct} - \text{chance correct}}$$

```
relativeImprovementOverChance <- function(TP, TN, FP, FN){
  N <- TP + TN + FP + FN
  actualYes <- TP + FN
  predictedYes <- TP + FP
  value <- ((N * (TP + TN)) - (actualYes * predictedYes + (N - predictedYes) * (N - act

  return(value)
}

relativeImprovementOverChanceValue <- relativeImprovementOverChance(TP = TPvalue, TN =
relativeImprovementOverChanceValue
```

```
## [1] 0.45
```

7.5.15 Relative Improvement Over Predicting from the Base Rate

Relative improvement over chance (RIOC) is a prediction's improvement over chance as a proportion of the maximum possible improvement over chance, as described by [Farrington and Loeber, 1989]. Higher values reflect greater accuracy.

$$\text{relative improvement over predicting from base rate} = \frac{\text{total correct} - \text{correct by predicting from base rate}}{\text{maximum correct} - \text{correct by predicting from base rate}}$$

```
relativeImprovementOverPredictingFromBaseRate <- function(TP, TN, FP, FN){
  N <- TP + TN + FP + FN
  BR <- (TP + FN)/N

  if(BR >= .5){
    SR <- 1
  } else if(BR < .5){
    SR <- 0
  }

  actualYes <- TP + FN
  predictedYes <- SR * N
  value <- ((N * (TP + TN)) - (actualYes * predictedYes + (N - predictedYes) * (N - actualYes)))

  return(value)
}

relativeImprovementOverPredictingFromBaseRateValue <- relativeImprovementOverPredictingFromBaseRa
relativeImprovementOverPredictingFromBaseRateValue
```

```
## [1] 0.15
```

7.5.16 Sensitivity (SN)

Sensitivity (SN) is also called true positive rate (TPR), hit rate (HR), or recall. Sensitivity is the conditional probability of a positive test given that the person has the condition: $P(R|C)$. Higher values reflect greater accuracy. As described in Section 7.1.2.6.1 and depicted in Figures 7.27 and 7.28, as the cutoff increases (becomes more conservative), sensitivity decreases. As the cutoff decreases, sensitivity increases.

$$\begin{aligned}\text{sensitivity (SN)} &= P(R|C) \\ &= \frac{\text{TP}}{\text{TP} + \text{FN}} = 1 - \text{FNR}\end{aligned}$$

```
sensitivity <- function(TP, TN, FP, FN){
  value <- TP/(TP + FN)

  return(value)
}

sensitivityValue <- sensitivity(TP = TPvalue, FN = FNvalue)
sensitivityValue
```

```
## [1] 0.65
```

Below I compute sensitivity and specificity at every possible cutoff.

```
possibleCutoffs <- unique(na.omit(mydataSDT$testScore))
possibleCutoffs <- possibleCutoffs[order(possibleCutoffs)]
possibleCutoffs <- c(possibleCutoffs, max(possibleCutoffs, na.rm = TRUE) + 0.01)

specificity <- function(TP, TN, FP, FN){
  value <- TN/(TN + FP)

  return(value)
}

accuracyVariables <- c("cutoff", "TP", "TN", "FP", "FN")

accuracyStats <- data.frame(matrix(nrow = length(possibleCutoffs), ncol = length(accura
names(accuracyStats) <- accuracyVariables

for(i in 1:length(possibleCutoffs)){
  newCutoff <- possibleCutoffs[i]

  mydataSDT$diagnosis <- NA
  mydataSDT$diagnosis[mydataSDT$testScore < newCutoff] <- 0
  mydataSDT$diagnosis[mydataSDT$testScore >= newCutoff] <- 1

  accuracyStats[i, "cutoff"] <- newCutoff
  accuracyStats[i, "TP"] <- length(which(mydataSDT$diagnosis == 1 & mydataSDT$disorder
  accuracyStats[i, "TN"] <- length(which(mydataSDT$diagnosis == 0 & mydataSDT$disorder
  accuracyStats[i, "FP"] <- length(which(mydataSDT$diagnosis == 1 & mydataSDT$disorder
  accuracyStats[i, "FN"] <- length(which(mydataSDT$diagnosis == 0 & mydataSDT$disorder
}

accuracyStats$sensitivity <- accuracyStats$TPrate <- sensitivity(TP = accuracyStats$TP
accuracyStats$specificity <- accuracyStats$TNrate <- specificity(TP = accuracyStats$TP
```

```
sensitivitySpecificityData <- pivot_longer(accuracyStats, cols = all_of(c("sensitivity","specific
```

Figure 7.27: Sensitivity and specificity by cutoff

7.5.17 Specificity (SP)

Specificity is also called true negative rate (TNR) or selectivity. Specificity is the conditional probability of a negative test given that the person does not have the condition: $P(\text{not } R|\text{not } C)$. Higher values reflect greater accuracy. As described in Section 7.1.2.6.1 and depicted in Figures 7.27 and 7.28, as the cutoff increases (becomes more conservative), specificity increases. As the cutoff decreases, specificity decreases.

$$\begin{aligned}\text{specificity (SP)} &= P(\text{not } R|\text{not } C)\\ &= \frac{\text{TN}}{\text{TN}+\text{FP}} = 1 - \text{FPR}\end{aligned}$$

```
specificity <- function(TP, TN, FP, FN){
  value <- TN/(TN + FP)
```

```
  return(value)
}

specificityValue <- specificity(TN = TNvalue, FP = FPvalue)
specificityValue
```

```
## [1] 0.8
```

FIGURE 37.3. Sensitivity and Specificity as a function of K6 cutoff values.

Figure 7.28: Sensitivity and specificity as a function of the cutoff. Note that, unlike many measures, the cutoff for the measure depicted here becomes more liberal (not more conservative) as the cutoff increases. Figure reprinted from @Treat2012, Figure 37.3, p. 730. Treat, T. A., & Viken, R. J. (2012). Measuring test performance with signal detection theory techniques. In H. Cooper (Ed.), *Handbook of research methods in psychology: Foundations, planning, measures, and psychometrics* (Vol. 1, pp. 723–744). American Psychological Association. https://doi.org/10.1037/13619-038

7.5.18 False Negative Rate (FNR)

The false negative rate (FNR) is also called the miss rate. The false negative rate is the conditional probability of a negative test given that the person has the condition: $P(\text{not } R|C)$. Lower values reflect greater accuracy.

$$\begin{aligned}\text{false negative rate (FNR)} &= P(\text{not } R|C) \\ &= \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - \text{TPR}\end{aligned}$$

```
falseNegativeRate <- function(TP, TN, FP, FN){
  value <- FN/(FN + TP)

  return(value)
}

falseNegativeRateValue <- falseNegativeRate(TP = TPvalue, FN = FNvalue)
falseNegativeRateValue
```

```
## [1] 0.35
```

7.5.19 False Positive Rate (FPR)

The false positive rate (FPR) is also called the false alarm rate (FAR) or fall-out. The false positive rate is the conditional probability of a positive test given that the person does not have the condition: $P(R|\text{not } C)$. Lower values reflect greater accuracy.

$$\begin{aligned}\text{false positive rate (FPR)} &= P(R|\text{not } C) \\ &= \frac{\text{FP}}{\text{FP} + \text{TN}} = 1 - \text{TNR}\end{aligned}$$

```
falsePositiveRate <- function(TP, TN, FP, FN){
  value <- FP/(FP + TN)

  return(value)
}

falsePositiveRateValue <- falsePositiveRate(TN = TNvalue, FP = FPvalue)
falsePositiveRateValue
```

```
## [1] 0.2
```

7.5.20 Positive Predictive Value (PPV)

The positive predictive value (PPV) is also called the positive predictive power (PPP) or precision. Many people confuse sensitivity ($P(R|C)$) with its inverse conditional probability, PPV ($P(C|R)$). PPV is the conditional probability of having the condition given a positive test: $P(C|R)$. Higher values reflect greater accuracy. PPV can be low even when sensitivity is high because it depends not only on sensitivity, but also on specificity and the base rate. Because PPV depends on the base rate, PPV is not an intrinsic property of a measure. The same measure will have a different PPV in different contexts with different base rates. As described in Section 7.1.2.6.1 and depicted in Figures 7.29 and 7.31, as the base rate increases, PPV increases. As the base rate decreases, PPV decreases. PPV also differs as a function of the cutoff. As described in Section 7.1.2.6.1 and depicted in Figures 7.30 and 7.32, as the cutoff increases (becomes more conservative), PPV increases. As the cutoff decreases (becomes more liberal), PPV decreases.

$$\begin{aligned}\text{positive predictive value (PPV)} &= P(C|R) \\ &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ &= \frac{\text{sensitivity} \times \text{BR}}{\text{sensitivity} \times \text{BR} + [(1 - \text{specificity}) \times (1 - \text{BR})]}\end{aligned}$$

```
positivePredictiveValue <- function(TP, TN, FP, FN, BR = NULL, SN, SP){
  if(is.null(BR)){
    value <- TP/(TP + FP)
  } else{
    value <- (SN * BR)/(SN * BR + (1 - SP) * (1 - BR))
  }

  return(value)
}

positivePredictivevalueValue <- positivePredictiveValue(TP = TPvalue, FP = FPvalue)
positivePredictivevalueValue
```

```
## [1] 0.65
```

```
positivePredictiveValue(BR = baseRateValue, SN = sensitivityValue, SP = specificityValu
```

```
## [1] 0.65
```

Below I compute PPV and NPV at every possible base rate given the sensitivity and specificity at the current cutoff.

```
negativePredictiveValue <- function(TP, TN, FP, FN, BR = NULL, SN, SP){
  if(is.null(BR)){
    value <- TN/(TN + FN)
  } else{
    value <- (SP * (1 - BR))/(SP * (1 - BR) + (1 - SN) * BR)
  }

  return(value)
}

ppvNPVbaseRateData <- data.frame(BR = seq(from = 0, to = 1, by = .01),
                                 SN = sensitivityValue,
                                 SP = specificityValue)

ppvNPVbaseRateData$positivePredictiveValue <- positivePredictiveValue(BR = ppvNPVbaseRateData$BR,
ppvNPVbaseRateData$negativePredictiveValue <- negativePredictiveValue(BR = ppvNPVbaseRateData$BR,

ppvNPVbaseRateData_long <- pivot_longer(ppvNPVbaseRateData, cols = all_of(c("positivePredictiveVa
```

Figure 7.29: Positive predictive value and negative predictive value by cutoff

Below I compute PPV and NPV at every possible cutoff.

```
accuracyStats$positivePredictiveValue <- positivePredictiveValue(TP = accuracyStats$TP
accuracyStats$negativePredictiveValue <- negativePredictiveValue(TP = accuracyStats$TP

ppvNPVcutoffData <- pivot_longer(accuracyStats, cols = all_of(c("positivePredictiveValu
```

Figure 7.30: Positive predictive value and negative predictive value by cutoff

7.5.21 Negative Predictive Value (NPV)

The negative predictive value (NPV) is also called the negative predictive power (NPP). Many people confuse specificity ($P(\text{not } R|\text{not } C)$) with its inverse conditional probability, NPV ($P(\text{not } C|\text{not } R)$). NPV is the conditional probability of not having the condition given a negative test: $P(\text{not } C|\text{not } R)$. Higher values reflect greater accuracy. NPV can be low even when specificity is high because it depends not only on specificity, but also on sensitivity and the base rate. Because NPV depends on the base rate, NPV is not an intrinsic property of a measure. The same measure will have a different NPV in different contexts with different base rates. As described in Section 7.1.2.6.1 and depicted in Figures 7.29 and 7.31, as the base rate increases, NPV decreases. As the base rate decreases, NPV increases. NPV also differs as a function of the cutoff. As described in Section 7.1.2.6.1 and depicted in Figures 7.30 and 7.32, as the cutoff

increases (becomes more conservative), NPV decreases. As the cutoff decreases (becomes more liberal), NPV decreases.

$$\begin{aligned}\text{negative predictive value (NPV)} &= P(\text{not } C|\text{not } R)\\ &= \frac{\text{TN}}{\text{TN}+\text{FN}}\\ &= \frac{\text{specificity}\times(1-\text{BR})}{\text{specificity}\times(1-\text{BR})+[(1-\text{sensitivity})\times\text{BR})]}\end{aligned}$$

```
negativePredictiveValue <- function(TP, TN, FP, FN, BR = NULL, SN, SP){
  if(is.null(BR)){
    value <- TN/(TN + FN)
  } else{
    value <- (SP * (1 - BR))/(SP * (1 - BR) + (1 - SN) * BR)
  }

  return(value)
}

negativePredictiveValueValue <- negativePredictiveValue(TN = TNvalue, FN = FNvalue)
negativePredictiveValueValue
```

```
## [1] 0.8
```

```
negativePredictiveValue(BR = baseRateValue, SN = sensitivityValue, SP = specificityValue)
```

```
## [1] 0.8
```

7.5.22 False Discovery Rate (FDR)

Many people confuse the false positive rate ($P(R|\text{not } C)$) with its inverse conditional probability, the false discovery rate ($P(\text{not } C|R)$). The false discovery rate is the conditional probability of not having the condition given a positive test: $P(\text{not } C|R)$. Lower values reflect greater accuracy.

$$\begin{aligned}\text{false discovery rate (FDR)} &= P(\text{not } C|R)\\ &= \frac{\text{FP}}{\text{FP}+\text{TP}} = 1-\text{PPV}\end{aligned}$$

Figure 7.31: Positive predictive value and negative predictive value as a function of the base rate. Figure reprinted from @Treat2012, Figure 37.5, p. 730. Treat, T. A., & Viken, R. J. (2012). Measuring test performance with signal detection theory techniques. In H. Cooper (Ed.), *Handbook of research methods in psychology: Foundations, planning, measures, and psychometrics* (Vol. 1, pp. 723–744). American Psychological Association. https://doi.org/10.1037/13619-038

Figure 7.32: Positive predictive value and negative predictive value as a function of the cutoff. Note that, unlike many measures, the cutoff for the measure depicted here becomes more liberal (not more conservative) as the cutoff increases. Figure reprinted from @Treat2012, Figure 37.4, p. 730. Treat, T. A., & Viken, R. J. (2012). Measuring test performance with signal detection theory techniques. In H. Cooper (Ed.), *Handbook of research methods in psychology: Foundations, planning, measures, and psychometrics* (Vol. 1, pp. 723–744). American Psychological Association. https://doi.org/10.1037/13619-038

```
falseDiscoveryRate <- function(TP, TN, FP, FN){
  value <- FP/(FP + TP)

  return(value)
}

falseDiscoveryRateValue <- falseDiscoveryRate(TP = TPvalue, FP = FPvalue)
falseDiscoveryRateValue
```

```
## [1] 0.35
```

7.5.23 False Omission Rate (FOR)

Many people confuse the false negative rate ($P(\text{not } R|C)$) with its inverse conditional probability, the false omission rate ($P(C|\text{not } R)$). The false omission rate is the conditional probability of having the condition given a negative test: $P(C|\text{not } R)$. Lower values reflect greater accuracy.

$$\begin{aligned}\text{false omission rate (FOR)} &= P(C|\text{not } R)\\ &= \frac{\text{FN}}{\text{FN}+\text{TN}} = 1 - \text{NPV}\end{aligned}$$

```
falseOmissionRate <- function(TP, TN, FP, FN){
  value <- FN/(FN + TN)

  return(value)
}

falseOmissionRateValue <- falseOmissionRate(TN = TNvalue, FN = FNvalue)
falseOmissionRateValue
```

```
## [1] 0.2
```

7.5.24 Youden's J Statistic

Youden's J statistic is also called Youden's Index or informedness. Youden's J statistic is the sum of sensitivity and specificity (and subtracting one). Higher values reflect greater accuracy.

$$\text{Youden's J statistic} = \text{sensitivity} + \text{specificity} - 1$$

```
youdenJ <- function(TP, TN, FP, FN){
  SN <- TP/(TP + FN)
  SP <- TN/(TN + FP)
  value <- SN + SP - 1

  return(value)
}

youdenJValue <- youdenJ(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
youdenJValue
```

```
## [1] 0.45
```

7.5.25 Balanced Accuracy

Balanced accuracy is the average of sensitivity and specificity. Higher values reflect greater accuracy.

$$\text{balanced accuracy} = \frac{\text{sensitivity} + \text{specificity}}{2}$$

```
balancedAccuracy <- function(TP, TN, FP, FN){
  SN <- TP/(TP + FN)
  SP <- TN/(TN + FP)
  value <- (SN + SP) / 2

  return(value)
}

balancedAccuracyValue <- balancedAccuracy(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
balancedAccuracyValue
```

```
## [1] 0.725
```

7.5.26 F-Score

The F-score combines precision (positive predictive value) and recall (sensitivity), where β indicates how many times more important sensitivity is than the positive predictive value. If sensitivity and the positive predictive value are equally important, $\beta = 1$, and the F-score is called the F_1 score. Higher values reflect greater accuracy.

$$
\begin{aligned}
F_\beta &= (1+\beta^2) \cdot \frac{\text{positive predictive value} \cdot \text{sensitivity}}{(\beta^2 \cdot \text{positive predictive value}) + \text{sensitivity}} \\
&= \frac{(1+\beta^2) \cdot \text{TP}}{(1+\beta^2) \cdot \text{TP} + \beta^2 \cdot \text{FN} + \text{FP}}
\end{aligned}
$$

```
fScore <- function(TP, TN, FP, FN, beta = 1){
  value <- ((1 + beta^2) * TP) / ((1 + beta^2) * TP + beta^2 * FN + FP)

  return(value)
}

f1ScoreValue <- fScore(TP = TPvalue, FP = FPvalue, FN = FNvalue)
f1ScoreValue
```

```
## [1] 0.65
```

```
fScore(TP = TPvalue, FP = FPvalue, FN = FNvalue, beta = 2)
```

```
## [1] 0.65
```

```
fScore(TP = TPvalue, FP = FPvalue, FN = FNvalue, beta = 0.5)
```

```
## [1] 0.65
```

7.5.27 Matthews Correlation Coefficient (MCC)

The Matthews correlation coefficient (MCC) is also called the phi coefficient. It is a correlation coefficient between predicted and observed values from a binary classification. Higher values reflect greater accuracy.

$$
\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}
$$

```
mcc <- function(TP, TN, FP, FN){
  TP <- as.double(TP)
  TN <- as.double(TN)
  FP <- as.double(FP)
  FN <- as.double(FN)
  value <- ((TP * TN) - (FP * FN)) / sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN

  return(value)
```

```
}

mccValue <- mcc(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
mccValue
```

```
## [1] 0.45
```

7.5.28 Diagnostic Odds Ratio

The diagnostic odds ratio is the odds of a positive test in those with the condition relative to the odds of a positive test in those without the condition. Higher values reflect greater accuracy. If the predictor is bad, the diagnostic odds ratio could be less than one, and values can go up from there. If the diagnostic odds ratio is greater than 2, we take the odds ratio seriously because we are twice as likely to predict accurately than inaccurately. However, the diagnostic odds ratio ignores/hides base rates. When interpreting the diagnostic odds ratio, it is important to keep in mind the clinical significance, because otherwise it is not very meaningful. Consider a risk factor that has a diagnostic odds ratio of 3 for tuberculosis, i.e., it puts you at 3 times as likely to develop tuberculosis. The prevalence of tuberculosis is relatively low. Assuming the prevalence of tuberculosis is less than 1/10th of 1%, your risk of developing tuberculosis is still very low even if the risk factor (with a diagnostic odds ratio of 3) is present.

$$\text{diagnostic odds ratio} = \frac{\text{TP} \times \text{TN}}{\text{FP} \times \text{FN}}$$

```
diagnosticOddsRatio <- function(TP, TN, FP, FN){
  value <- (TP * TN) / (FP * FN)

  return(value)
}

diagnosticOddsRatioValue <- diagnosticOddsRatio(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FN
diagnosticOddsRatioValue
```

```
## [1] 7.428571
```

7.5.29 Positive Likelihood Ratio

A likelihood ratio is the ratio of two probabilities. It can be used to compare the likelihood of two possibilities. The positive likelihood ratio compares the true positive rate to the false positive rate. Higher values reflect greater accuracy because it indicates that a true positive is more likely than a false positive.

$$
\begin{aligned}
\text{positive likelihood ratio (LR+)} &= \frac{\text{TPR}}{\text{FPR}} \\
&= \frac{P(R|C)}{P(R|\text{not } C)} \\
&= \frac{P(R|C)}{1 - P(\text{not } R|\text{not } C)} \\
&= \frac{\text{sensitivity}}{1 - \text{specificity}}
\end{aligned}
$$

```
positiveLikelihoodRatio <- function(TP, TN, FP, FN){
  SN <- TP/(TP + FN)
  SP <- TN/(TN + FP)

  value <- SN/(1 - SP)

  return(value)
}

positiveLikelihoodRatioValue <- positiveLikelihoodRatio(TP = TPvalue, TN = TNvalue, FP
positiveLikelihoodRatioValue
```

```
## [1] 3.25
```

7.5.30 Negative Likelihood Ratio

The negative likelihood ratio compares the false negative rate to the true negative rate. Smaller values reflect greater accuracy because it indicates that a false negative is less likely than a true negative.

$$
\begin{aligned}
\text{negative likelihood ratio (LR}-) &= \frac{\text{FNR}}{\text{TNR}} \\
&= \frac{P(\text{not } R|C)}{P(\text{not } R|\text{not } C)} \\
&= \frac{1 - P(R|C)}{P(\text{not } R|\text{not } C)} \\
&= \frac{1 - \text{sensitivity}}{\text{specificity}}
\end{aligned}
$$

```
negativeLikelihoodRatio <- function(TP, TN, FP, FN){
  SN <- TP/(TP + FN)
  SP <- TN/(TN + FP)
```

```
  value <- (1 - SN)/SP

  return(value)
}

negativeLikelihoodRatioValue <- negativeLikelihoodRatio(TP = TPvalue, TN = TNvalue, FP = FPvalue,
negativeLikelihoodRatioValue
```

```
## [1] 0.4375
```

7.5.31 Posttest Odds

As presented in Equation (7.1.1.1), the posttest (or posterior) odds are equal to the pretest odds multiplied by the likelihood ratio. The posttest odds and posttest probability can be useful to calculate when the pretest probability is different from the base rate (or prevalence) of the classification. For instance, you might use a different pretest probability if a test result is already known and you want to know the updated posttest probability after conducting a second test.

$$\text{posttest odds} = \text{pretest odds} \times \text{likelihood ratio}$$

For calculating the posttest odds of a true positive compared to a false positive, we use the positive likelihood ratio below. We would use the negative likelihood ratio if we wanted to calculate the posttest odds of a false negative compared to a true negative.

```
posttestOdds <- function(TP, TN, FP, FN, pretestProb = NULL, SN = NULL, SP = NULL, likelihoodRati
  if(!is.null(pretestProb) & !is.null(SN) & !is.null(SP)){
    pretestProbability <- pretestProb
    pretestOdds <- pretestProbability / (1 - pretestProbability)

    likelihoodRatio <- SN/(1 - SP)
  } else if(!is.null(pretestProb) & !is.null(likelihoodRatio)){
    pretestProbability <- pretestProb
    pretestOdds <- pretestProbability / (1 - pretestProbability)

    likelihoodRatio <- likelihoodRatio
  } else {
    N <- TP + TN + FP + FN
    pretestProbability <- (TP + FN)/N
    pretestOdds <- pretestProbability / (1 - pretestProbability)
```

```
    SN <- TP/(TP + FN)
    SP <- TN/(TN + FP)
    likelihoodRatio <- SN/(1 - SP)
  }

  value <- pretestOdds * likelihoodRatio

  return(value)
}

posttestOddsValue <- posttestOdds(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalu
posttestOddsValue
```

```
## [1] 1.857143
```

```
posttestOdds(pretestProb = baseRate(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNva
             SN = sensitivity(TP = TPvalue, FN = FNvalue),
             SP = specificity(TN = TNvalue, FP = FPvalue))
```

```
## [1] 1.857143
```

```
posttestOdds(pretestProb = baseRate(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNva
             likelihoodRatio = positiveLikelihoodRatio(TP = TPvalue, TN = TNvalue, FP =
```

```
## [1] 1.857143
```

7.5.32 Posttest Probability

The posttest probability is the probability of having the disorder given a test result. When the base rate is used as the pretest probability, the posttest probability given a positive test is equal to positive predictive value. To convert odds to a probability, divide the odds by one plus the odds.

$$\text{posttest probability} = \frac{\text{posttest odds}}{1 + \text{posttest odds}}$$

```
posttestProbability <- function(TP, TN, FP, FN, pretestProb = NULL, SN = NULL, SP = NUL
  if(!is.null(pretestProb) & !is.null(SN) & !is.null(SP)){
    pretestProbability <- pretestProb
    pretestOdds <- pretestProbability / (1 - pretestProbability)

    likelihoodRatio <- SN/(1 - SP)
```

```
  }  else if(!is.null(pretestProb) & !is.null(likelihoodRatio)){
    pretestProbability <- pretestProb
    pretestOdds <- pretestProbability / (1 - pretestProbability)

    likelihoodRatio <- likelihoodRatio
  } else {
    N <- TP + TN + FP + FN
    pretestProbability <- (TP + FN)/N
    pretestOdds <- pretestProbability / (1 - pretestProbability)

    SN <- TP/(TP + FN)
    SP <- TN/(TN + FP)
    likelihoodRatio <- SN/(1 - SP)
  }

  posttestOdds <- pretestOdds * likelihoodRatio
  value <- posttestOdds / (1 + posttestOdds)

  return(value)
}

posttestProbabilityValue <- posttestProbability(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FN
posttestProbabilityValue
```

```
## [1] 0.65
```

```
posttestProbability(pretestProb = baseRate(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue
                    SN = sensitivity(TP = TPvalue, FN = FNvalue),
                    SP = specificity(TN = TNvalue, FP = FPvalue))
```

```
## [1] 0.65
```

```
posttestProbability(pretestProb = baseRate(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue
                    likelihoodRatio = positiveLikelihoodRatio(TP = TPvalue, TN = TNvalue, FP = FP
```

```
## [1] 0.65
```

Consider the following example: Assume the base rate of the condition is .03%. We have two tests. Test A has a sensitivity of .95 and a specificity of .80. Test B has a sensitivity of .70 and a specificity of .90. What is the probability of having the condition if a person has a positive test on Test A? Assuming the errors of the two tests are independent, what is the probability of having the condition if the person has a positive test on Test B after having a positive test on Test A?

```
probGivenTestA <- posttestProbability(pretestProb = .003, SN = .95, SP = .80)
probGivenTestAthenB <- posttestProbability(pretestProb = probGivenTestA, SN = .70, SP =

probGivenTestA
```

```
## [1] 0.01409147
```

```
probGivenTestAthenB
```

```
## [1] 0.09095054
```

The probability of having the condition if a person has a positive test on Test A is 1.4%. The probability of having the condition if the person has a positive test on Test B after having a positive test on Test A is 9.1%.

7.5.33 d' Sensitivity from Signal Detection Theory

d' is an index of sensitivity from signal detection theory, as described by [Stanislaw and Todorov, 1999]. Higher values reflect greater accuracy.

$$d' = z(\text{hit rate}) - z(\text{false alarm rate})$$

```
dPrimeSDT <- function(TP, TN, FP, FN){
  HR <- TP/(TP + FN)
  FAR <- FP/(FP + TN)
  value <- qnorm(HR) - qnorm(FAR)

  return(value)
}

dPrimeValue <- dPrimeSDT(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
dPrimeValue
```

```
## [1] 1.226942
```

7.5.34 A (Non-Parametric) Sensitivity from Signal Detection Theory

A is a non-parametric index of sensitivity from signal detection theory, as described by [Zhang and Mueller, 2005]. Higher values reflect greater accuracy.

https://sites.google.com/a/mtu.edu/whynotaprime/

$$A = \begin{cases} \frac{3}{4} + \frac{H-F}{4} - F(1-H) & \text{if } F \leq 0.5 \leq H; \\ \frac{3}{4} + \frac{H-F}{4} - \frac{F}{4H} & \text{if } F \leq H \leq 0.5; \\ \frac{3}{4} + \frac{H-F}{4} - \frac{1-H}{4(1-F)} & \text{if } 0.5 \leq F \leq H. \end{cases}$$

```
aSDT <- function(TP, TN, FP, FN){
  HR <- TP/(TP + FN)
  FAR <- FP/(FP + TN)

  if(FAR <= .5 & HR >= .5){
    value <- (3/4) + ((HR - FAR)/4) - (FAR * (1 - HR))
    } else if(FAR <= HR & HR <= .5){
      value <- (3/4) + ((HR - FAR)/4) - (FAR/(4 * HR))
      } else {
        value <- (3/4) + ((HR - FAR)/4) - ((1 - HR)/(4 * (1 - FAR)))
        }

  return(value)
}

Avalue <- aSDT(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
Avalue
```

```
## [1] 0.7925
```

7.5.35 β Bias from Signal Detection Theory

β is an index of bias from signal detection theory, as described by [Stanislaw and Todorov, 1999]. Smaller values reflect greater accuracy.

$$\beta =$$

```
betaSDT <- function(TP, TN, FP, FN){
  HR <- TP/(TP + FN)
  FAR <- FP/(FP + TN)
  value <- exp(qnorm(FAR)^2/2 - qnorm(HR)^2/2)

  return(value)
}

betaValue <- betaSDT(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
betaValue
```

```
## [1] 1.323034
```

7.5.36 c Bias from Signal Detection Theory

c is an index of bias from signal detection theory, as described by [Stanislaw and Todorov, 1999]. Smaller values reflect greater accuracy.

$$c =$$

```
cSDT <- function(TP, TN, FP, FN){
  HR <- TP/(TP + FN)
  FAR <- FP/(FP + TN)
  value <- -(qnorm(HR) + qnorm(FAR))/2

  return(value)
}

cValue <- cSDT(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
cValue
```

```
## [1] 0.2281504
```

7.5.37 b (Non-Parametric) Bias from Signal Detection Theory

b is a non-parametric index of bias from signal detection theory, as described by [Zhang and Mueller, 2005]. Smaller values reflect greater accuracy.

$$b = \begin{cases} \frac{5-4H}{1+4F} & \text{if } F \leq 0.5 \leq H; \\ \frac{H^2+H}{H^2+F} & \text{if } F \leq H \leq 0.5; \\ \frac{(1-F)^2+(1-H)}{(1-F)^2+(1-F)} & \text{if } 0.5 \leq F \leq H. \end{cases}$$

```
bSDT <- function(TP, TN, FP, FN){
  HR <- TP/(TP + FN)
  FAR <- FP/(FP + TN)

  if(FAR <= .5 & HR >= .5){
    value <-(5 - (4 * HR))/(1 + (4 * FAR))
    } else if(FAR <= HR & HR <= .5){
      value <- (HR^2 + HR)/(HR^2 + FAR)
      } else {
        value <- ((1 - FAR)^2 + (1 - HR))/((1 - FAR)^2 + (1 - FAR))
```

```
        }

  return(value)
}

bValue <- bSDT(TP = TPvalue, TN = TNvalue, FP = FPvalue, FN = FNvalue)
bValue
```

```
## [1] 1.333333
```

7.5.38 Mean Difference between Predicted versus Observed Values (Miscalibration)

The mean difference between predicted values versus expected values at a given cutoff is an index of miscalibration of predictions at that cutoff. It is called "calibration-in-the-small" (as opposed to calibration-in-the-large, which spans all cutoffs). Values closer to zero reflect greater accuracy. Values above zero indicate that the predicted values are, on average, greater than the observed values. Values below zero indicate that the observed values are, on average, greater than predicted values.

```
miscalibration <- function(predicted, actual, cutoff, bins = 10){
  data <- data.frame(na.omit(cbind(predicted, actual)))

  calibrationTable <- mutate(data, bin = cut_number(predicted, n = 10)) %>%
    group_by(bin) %>%
    summarise(n = length(predicted),
              meanPredicted = mean(predicted, na.rm = TRUE),
              meanObserved = mean(actual, na.rm = TRUE),
              .groups = "drop")

  calibrationTable$cutoffMin <- as.numeric(str_replace_all(str_split(calibrationTable$bin, patter
  calibrationTable$cutoffMax <- as.numeric(str_replace_all(str_split(calibrationTable$bin, patter

  calibrationTable$inRange <- with(calibrationTable, cutoff >= cutoffMin & cutoff <= cutoffMax)

  if(length(which(calibrationTable$inRange == TRUE)) > 0){
    nearestCutoff <- calibrationTable$bin[min(which(calibrationTable$inRange == TRUE))]
    calibrationAtNearestCutoff <- calibrationTable[which(calibrationTable$bin == nearestCutoff),]
    calibrationAtNearestCutoff <- as.data.frame(calibrationTable[max(which(calibrationTable$inRan

    meanPredicted <- calibrationAtNearestCutoff[, "meanPredicted"]
    meanObserved <- calibrationAtNearestCutoff[, "meanObserved"]
    differenceBetweenPredictedAndObserved <- meanPredicted - meanObserved
```

```
  } else{
    differenceBetweenPredictedAndObserved <- NA
  }

  return(differenceBetweenPredictedAndObserved)
}

miscalibration(predicted = mydataSDT$predictedProbability, actual = mydataSDT$disorder
```

```
## [1] -0.07843137
```

7.6 Optimal Cutoff Specification

7.6.1 Decision Theory

According to the decision theory approach to picking the optimal cutoff, the optimal cutoff depends on the value/importance placed on each of the four decision outcomes [(true positives, true negatives, false positives, false negatives); Treat and Viken [2012]]. Utility is the relative value placed on a specific decision-making outcome (i.e., user-perceived benefit or cost): utilities typically range between zero and one, where a value of zero represents the least desired outcome and a value of one indicates the most desired outcome. According to the decision theory approach, the optimal cutoff is the cutoff with the highest overall utility.

7.6.1.1 Overall utility of a specific cutoff value

The overall utility of a specific cutoff value is a utilities-weighted sum of the probabilities of the four decision-making outcomes (hits, misses, correct rejections, false alarms). That is, overall utility is the product of the probability of a particular outcome (e.g., $\text{BR} \times \text{TP}$) and the utility of that outcome (e.g., how much we value TPs relative to other outcomes). Higher values reflect greater utility, so you would pick the cutoff with the highest overall utility.

$$\begin{aligned} U_{\text{overall}} = & (\text{BR})(\text{HR})(U_{\text{H}}) \\ & + (\text{BR})(1 - \text{HR})(U_{\text{M}}) \\ & + (1 - \text{BR})(\text{FAR})(U_{\text{FA}}) \\ & + (1 - \text{BR})(1 - \text{FAR})(U_{\text{CR}}) \end{aligned}$$

where BR = base rate, HR = hit rate (true positive rate), FAR = false alarm rate (false positive rate), U_{H} = utility of hits (true positives),

U_{M} = utility of misses (false negatives), U_{FA} = utility of false alarms (false positives), U_{CR} = utility of correct rejections (true negatives)

```
Uoverall <- function(BR, HR, FAR, UH, UM, UCR, UFA){
  (BR*HR*UH) + (BR*(1 - HR)*UM) + ((1 - BR)*FAR*UFA) + ((1 - BR)*(1 - FAR)*(UCR))
}
```

```
Uoverall(BR = baseRateValue, HR = sensitivityValue, FAR = falsePositiveRateValue, UH = 1, UM = 0,
```

```
## [1] 0.65
```

```
Uoverall(BR = baseRateValue, HR = sensitivityValue, FAR = falsePositiveRateValue, UH = 1, UM = 0,
```

```
## [1] 0.7454545
```

```
Uoverall(BR = baseRateValue, HR = sensitivityValue, FAR = falsePositiveRateValue, UH = 0.75, UM =
```

```
## [1] 0.7181818
```

7.6.1.2 Utility ratio

The utility ratio is the user-perceived relative importance of decisions about negative versus positive cases. If the utility ratio value is one, it indicates that identifying negative cases and positive cases are equally important. Values above one indicate greater relative importance of identifying negative cases than positive cases. Values below one indicate greater relative importance of identifying positive cases than negative cases. Values of one indicate that you are maximizing percent accuracy.

$$\text{Utility Ratio} = \frac{U_{\mathrm{CR}} - U_{\mathrm{FA}}}{U_{\mathrm{H}} - U_{\mathrm{M}}}$$

where U_{H} = utility of hits (true positives), U_{M} = utility of misses (false negatives), U_{FA} = utility of false alarms (false positives), U_{CR} = utility of correct rejections (true negatives)

```
utilityRatio <- function(UH, UM, UCR, UFA){
  (UCR - UFA) / (UH - UM)
}
```

```
utilityRatio(UH = 1, UM = 0, UCR = 0.75, UFA = 0.25)
```

```
## [1] 0.5
```

```
utilityRatio(UH = 1, UM = 0, UCR = 1, UFA = 0)
```

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

```
utilityRatio(UH = 0.75, UM = 0.25, UCR = 1, UFA = 0)
```

```
## [1] 2
```

Decision theory has key advantages, because it identifies the cutoff that would help you best achieve the goals/purpose of the assessment. However, it can be challenging to specify the relative costs of errors. If you cannot decide values for outcomes (relative importance between FP and FN), you can use information theory to identify the optimal cutoff.

7.6.2 Information Theory

When the user does not differentially weigh the value/importance of the four decision making outcomes (hits, misses, correct rejections, false alarms), the information theory approach can be useful for specifying the optimal cutoff. According to the information theory approach, the optimal cutoff is the cutoff that provides the greatest information gain [Treat and Viken, 2012].

7.6.2.1 Information gain

Information gain (I_{gain}) is the reduction of uncertainty about the true classification of a case that results from administering an assessment or prediction measure [Treat and Viken, 2012]. Greater values reflect greater reduction of uncertainty, so the optimal cutoff can be specified as the cutoff with the highest information gain.

7.6.2.1.1 Formula from Treat and Viken [2012]

$$
\begin{aligned}
I_{\text{gain}} = &(\text{BR})(\text{HR})\left[\log_2\left(\frac{\text{HR}}{G}\right)\right] \\
&+ (\text{BR})(1-\text{HR})\left[\log_2\left(\frac{1-\text{HR}}{1-G}\right)\right] \\
&+ (1-\text{BR})(\text{FAR})\left[\log_2\left(\frac{\text{FAR}}{G}\right)\right] \\
&+ (1-\text{BR})(1-FAR)\left[\log_2\left(\frac{1-\text{FAR}}{1-G}\right)\right]
\end{aligned}
$$

where BR = base rate, HR = hit rate (true positive rate), FAR = false alarm rate (false positive rate), G = selection ratio = BR(HR) + $(1 - \text{BR})(\text{FAR})$, as reported in Somoza et al. [1989] [; see below].

```
Igain <- function(BR, HR, FAR){
  G <- BR*(HR) + (1 - BR)*(FAR)

  (BR*HR*log2(HR/G)) +
    (BR*(1 - HR)*(log2((1 - HR)/(1 - G)))) +
    ((1 - BR)*FAR*(log2(FAR/G))) +
    ((1 - BR)*(1 - FAR)*(log2((1 - FAR)/(1 - G))))
}
```

```
Igain(BR = baseRateValue, HR = sensitivityValue, FAR = falsePositiveRateValue)
```

```
## [1] 0.1465904
```

7.6.2.1.2 Alternative formula from Metz et al. [1973]

$$
\begin{aligned}
I_{\text{gain}} = {} & p(S|s) \cdot p(s) \cdot log_2 \left\{ \frac{p(S|s)}{p(S|s) \cdot p(s) + p(S|n)[1 - p(s)]} \right\} \\
& + p(S|n)[1 - p(s)] \times log_2 \left\{ \frac{p(S|n)}{p(S|s) \cdot p(s) + p(S|n)[1 - p(s)]} \right\} \\
& + [1 - p(S|s)] \cdot p(s) \times log_2 \left\{ \frac{1 - p(S|s)}{1 - p(S|s) \cdot p(s) - p(S|n)[1 - p(s)]} \right\} \\
& + [1 - p(S|n)][1 - p(s)] \times log_2 \left\{ \frac{1 - p(S|n)}{1 - p(S|s) \cdot p(s) - p(S|n)[1 - p(s)]} \right\}
\end{aligned}
$$

where $p(S|s) = $ sensitivity (hit rate or true positive rate); i.e., the conditional probability of deciding a signal is present (S) when the signal is in fact present(s), $p(S|n) = $ false positive rate (false alarm rate); i.e., the conditional probability of deciding a signal is present (S) when the signal is in fact absent (n), $p(s) = $ base rate, i.e., the probability that the signal is in fact present (s).

```
Igain2 <- function(BR, HR, FAR){
  HR * BR * log2(HR / ((HR * BR) + (FAR * (1 - BR)))) +
    FAR * (1 - BR) * log2(FAR / ((HR * BR) + (FAR * (1 - BR)))) +
    (1 - HR) * BR * log2((1 - HR) / (1 - (HR * BR) - (FAR * (1 - BR)))) +
    (1 - FAR) * (1 - BR) * log2((1 - FAR) / (1 - (HR * BR) - (FAR * (1 - BR))))
}
```

```
Igain2(BR = baseRateValue, HR = sensitivityValue, FAR = falsePositiveRateValue)
```

```
## [1] 0.1465904
```

7.6.2.1.3 Alternative formula from Somoza et al. [1989]

$$
\begin{aligned}
I_{\text{gain}} = & [(\text{TPR})(\text{Pr})] \times \log_2(\text{TPR}/G) \\
& + [(\text{FPR})(1-\text{Pr})] \times \log_2(\text{FPR}/G) \\
& + [(1-\text{TPR})(\text{Pr})] \times \log_2\left(\frac{1-\text{TPR}}{1-G}\right) \\
& + [(1-FPR)(1-\text{Pr})] \times \log_2\left(\frac{1-\text{FPR}}{1-G}\right)
\end{aligned}
$$

where TP = true positive rate (hit rate), Pr = prevalence (base rate), FP = false positive rate (false alarm rate), $G = \text{Pr}(\text{TP}) + (1-\text{Pr})(\text{FP})$ = selection ratio

```
Igain3 <- function(BR, HR, FAR){
  G <- BR*(HR) + (1 - BR)*(FAR)

  ((HR)*(BR))*log2((HR/G)) +
    ((FAR)*(1-BR))*log2((FAR/G)) +
    ((1-HR)*(BR))*log2((1-HR)/(1-G)) +
    ((1-FAR)*(1-BR))*log2((1-FAR)/(1-G))
}
```

```
Igain3(BR = baseRateValue, HR = sensitivityValue, FAR = falsePositiveRateValue)
```

```
## [1] 0.1465904
```

7.6.2.1.4 Examples

Case A from Exhibit 37.2 [Treat and Viken, 2012]:

```
Igain(HR = (911/1899), FAR = (509/4757), BR = (1899/6656))
```

```
## [1] 0.112081
```

Case B from Exhibit 37.2 [Treat and Viken, 2012]:

```
Igain(HR = (1597/3328), FAR = (356/3328), BR = (3328/6656))
```

```
## [1] 0.1283265
```

Case C from Exhibit 37.2 [Treat and Viken, 2012]:

```
Igain(HR = (2040/3328), FAR = (654/3328), BR = (3328/6656))
```

```
## [1] 0.1347846
```

Case B from Exhibit 37.3 [Treat and Viken, 2012]:

```
Igain(HR = (1164/1899), FAR = (935/4757), BR = (1899/6656))
```

```
## [1] 0.1135549
```

7.6.2.1.5 Effect of Base Rate Information gain depends on the base rate, as depicted in Figures 7.33 and 7.34. The maximum reduction of uncertainty (i.e., greatest information) occurs when the base rate is 0.5. A base rate tells us little a priori about the condition if the probability of the condition is 50/50, so the measure can provide more benefit. If the base rate is 0.3 or 0.7, we can do better than going with the with the base rate. If the base rate is 0.9 or 0.1, it is difficult for our measure to do better than going with the base rate. If the base rate is 0.05 of 0.95 (or more extreme), it is likely that our measure will do almost nothing in terms of information gain.

7.7 Accuracy at Every Possible Cutoff

7.7.1 Specify utility of each outcome

```
utilityHits <- 1
utilityMisses <- 0
utilityCorrectRejections <- 0.75
utilityFalseAlarms <- 0.25
```

7.7.2 Calculate Accuracy

```
possibleCutoffs <- unique(na.omit(mydataSDT$testScore))
possibleCutoffs <- possibleCutoffs[order(possibleCutoffs)]
possibleCutoffs <- c(possibleCutoffs, max(possibleCutoffs, na.rm = TRUE) + 0.01)
```

Figure 7.33: Information gain as a function of the base rate. BR = base rate. Figure reprinted from @Treat2012, Figure 37.11, p. 741. Treat, T. A., & Viken, R. J. (2012). Measuring test performance with signal detection theory techniques. In H. Cooper (Ed.), *Handbook of research methods in psychology: Foundations, planning, measures, and psychometrics* (Vol. 1, pp. 723–744). American Psychological Association. https://doi.org/10.1037/13619-038

Figure 7.34: Effect of the Base Rate on Information Gain

```
accuracyVariables <- c("cutoff", "TP", "TN", "FP", "FN", "differenceBetweenPredictedAndObserved")

accuracyStats <- data.frame(matrix(nrow = length(possibleCutoffs), ncol = length(accuracyVariable
names(accuracyStats) <- accuracyVariables

for(i in 1:length(possibleCutoffs)){
  cutoff <- possibleCutoffs[i]

  mydataSDT$diagnosis <- NA
  mydataSDT$diagnosis[mydataSDT$testScore < cutoff] <- 0
  mydataSDT$diagnosis[mydataSDT$testScore >= cutoff] <- 1

  accuracyStats[i, "cutoff"] <- cutoff
  accuracyStats[i, "TP"] <- length(which(mydataSDT$diagnosis == 1 & mydataSDT$disorder == 1))
  accuracyStats[i, "TN"] <- length(which(mydataSDT$diagnosis == 0 & mydataSDT$disorder == 0))
  accuracyStats[i, "FP"] <- length(which(mydataSDT$diagnosis == 1 & mydataSDT$disorder == 0))
  accuracyStats[i, "FN"] <- length(which(mydataSDT$diagnosis == 0 & mydataSDT$disorder == 1))

  accuracyStats[i, "differenceBetweenPredictedAndObserved"] <- miscalibration(predicted = mydataS
}

accuracyStats$N <- sampleSize(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = accuracyStats$FP
```

```
accuracyStats$selectionRatio <- selectionRatio(TP = accuracyStats$TP, TN = accuracyStat
accuracyStats$baseRate <- baseRate(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = a

accuracyStats$percentAccuracy <- percentAccuracy(TP = accuracyStats$TP, TN = accuracyS
accuracyStats$percentAccuracyByChance <- percentAccuracyByChance(TP = accuracyStats$TP
accuracyStats$relativeImprovementOverChance <- relativeImprovementOverChance(TP = accu
accuracyStats$relativeImprovementOverPredictingFromBaseRate <- relativeImprovementOver

accuracyStats$sensitivity <- accuracyStats$TPrate <- sensitivity(TP = accuracyStats$TP
accuracyStats$specificity <- accuracyStats$TNrate <- specificity(TP = accuracyStats$TP
accuracyStats$FNrate <- falseNegativeRate(TP = accuracyStats$TP, TN = accuracyStats$TN
accuracyStats$FPrate <- falsePositiveRate(TP = accuracyStats$TP, TN = accuracyStats$TN

accuracyStats$youdenJ <- youdenJ(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = acc

accuracyStats$positivePredictiveValue <- positivePredictiveValue(TP = accuracyStats$TP
accuracyStats$negativePredictiveValue <- negativePredictiveValue(TP = accuracyStats$TP
accuracyStats$falseDiscoveryRate <- falseDiscoveryRate(TP = accuracyStats$TP, TN = accu
accuracyStats$falseOmissionRate <- falseOmissionRate(TP = accuracyStats$TP, TN = accura

accuracyStats$balancedAccuracy <- balancedAccuracy(TP = accuracyStats$TP, TN = accurac
accuracyStats$f1Score <- fScore(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = accu
accuracyStats$mcc <- mcc(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = accuracySta

accuracyStats$diagnosticOddsRatio <- diagnosticOddsRatio(TP = accuracyStats$TP, TN = ac
accuracyStats$positiveLikelihoodRatio <- positiveLikelihoodRatio(TP = accuracyStats$TP
accuracyStats$negativeLikelihoodRatio <- negativeLikelihoodRatio(TP = accuracyStats$TP

accuracyStats$dPrimeSDT <- dPrimeSDT(TP = accuracyStats$TP, TN = accuracyStats$TN, FP =
accuracyStats$betaSDT <- betaSDT(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = acc
accuracyStats$cSDT <- cSDT(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = accuracyS
accuracyStats$ASDT <- aSDT(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = accuracyS
accuracyStats$bSDT <- bSDT(TP = accuracyStats$TP, TN = accuracyStats$TN, FP = accuracyS

accuracyStats$overallUtility <- Uoverall(BR = accuracyStats$baseRate, HR = accuracyStat
accuracyStats$utilityRatio <- utilityRatio(UH = utilityHits, UM = utilityMisses, UCR =
accuracyStats$informationGain <- Igain(BR = accuracyStats$baseRate, HR = accuracyStatsS

#Replace NaN and INF values with NA
is.nan.data.frame <- function(x)
  do.call(cbind, lapply(x, is.nan))

accuracyStats[is.nan.data.frame(accuracyStats)] <- NA
accuracyStats <- do.call(data.frame, lapply(accuracyStats, function(x) replace(x, is.in
```

7.7.3 All Accuracy Statistics

```
#Need to add separate check for epub -> kable()
paged_table(accuracyStats)
```

```
kable(accuracyStats,
      booktabs = TRUE)
```

cutoff	TP	TN	FP	FN	differenceBetweenPredictedAndObserved	N	selectionRatio	bas
0.03	40	0	70	0	-0.1531579	110	1.0000000	0.36
0.04	39	0	70	1	-0.1531579	110	0.9909091	0.36
0.05	39	4	66	1	-0.1531579	110	0.9545455	0.36
0.06	39	7	63	1	-0.1531579	110	0.9272727	0.36
0.07	39	9	61	1	-0.0628571	110	0.9090909	0.36
0.08	36	15	55	4	-0.1600000	110	0.8272727	0.36
0.09	35	21	49	5	-0.1269231	110	0.7636364	0.36
0.10	33	27	43	7	-0.1269231	110	0.6909091	0.36
0.11	31	33	37	9	-0.1660000	110	0.6181818	0.36
0.12	30	37	33	10	-0.1660000	110	0.5727273	0.36
0.13	29	37	33	11	-0.1660000	110	0.5636364	0.36
0.14	28	40	30	12	0.0580000	110	0.5272727	0.36
0.15	27	44	26	13	0.0580000	110	0.4818182	0.36
0.16	27	48	22	13	0.0580000	110	0.4454545	0.36
0.17	26	51	19	14	-0.2690000	110	0.4090909	0.36
0.18	26	53	17	14	-0.2690000	110	0.3909091	0.36
0.19	26	54	16	14	-0.2690000	110	0.3818182	0.36
0.22	26	56	14	14	-0.2690000	110	0.3636364	0.36
0.23	25	56	14	15	-0.2690000	110	0.3545455	0.36
0.24	24	56	14	16	-0.2690000	110	0.3454545	0.36
0.25	24	57	13	16	-0.2690000	110	0.3363636	0.36
0.26	23	57	13	17	-0.2690000	110	0.3272727	0.36
0.27	22	57	13	18	-0.2690000	110	0.3181818	0.36
0.28	21	57	13	19	-0.2690000	110	0.3090909	0.36
0.30	21	58	12	19	-0.2175000	110	0.3000000	0.36
0.32	20	58	12	20	-0.2175000	110	0.2909091	0.36
0.33	19	59	11	21	-0.2175000	110	0.2727273	0.36
0.34	18	60	10	22	-0.2175000	110	0.2545455	0.36
0.35	18	61	9	22	-0.2175000	110	0.2454545	0.36
0.38	17	61	9	23	-0.2175000	110	0.2363636	0.36
0.41	17	62	8	23	-0.2175000	110	0.2272727	0.36
0.43	16	62	8	24	-0.2175000	110	0.2181818	0.36
0.44	16	63	7	24	0.1820000	110	0.2090909	0.36
0.45	14	63	7	26	0.1820000	110	0.1909091	0.36
0.46	14	64	6	26	0.1820000	110	0.1818182	0.36
0.47	14	65	5	26	0.1820000	110	0.1727273	0.36
0.48	14	67	3	26	0.1820000	110	0.1545455	0.36
0.49	13	68	2	27	0.1820000	110	0.1363636	0.36
0.50	12	68	2	28	0.1820000	110	0.1272727	0.36
0.52	12	70	0	28	0.1820000	110	0.1090909	0.36
0.56	11	70	0	29	-0.1381818	110	0.1000000	0.36
0.58	10	70	0	30	-0.1381818	110	0.0909091	0.36
0.70	9	70	0	31	-0.1381818	110	0.0818182	0.36
0.71	8	70	0	32	-0.1381818	110	0.0727273	0.36
0.74	6	70	0	34	-0.1381818	110	0.0545455	0.36
0.77	5	70	0	35	-0.1381818	110	0.0454545	0.36
0.82	4	70	0	36	-0.1381818	110	0.0363636	0.36
0.86	3	70	0	37	-0.1381818	110	0.0272727	0.36
0.96	2	70	0	38	-0.1381818	110	0.0181818	0.36
2.07	1	70	0	39	-0.1381818	110	0.0090909	0.36

```
kable(accuracyStats,
      booktabs = TRUE)
```

7.7.4 Youden's J Statistic

7.7.4.1 Threshold

Threshold at maximum combination of sensitivity and specificity:

max(sensitivity + specificity)

```
youdenIndex <- coords(roc(data = mydataSDT, response = disorder, predictor = testScore, smooth =

youdenIndex
```

```
## [1] 0.205
```

7.7.4.2 Accuracy statistics at cutoff of Youden's J Statistic

```
accuracyStats[head(which(accuracyStats$cutoff >= youdenIndex), 1),]
```

```
##    cutoff TP TN FP FN differenceBetweenPredictedAndObserved   N selectionRatio
## 18   0.22 26 56 14 14                               -0.269 110      0.3636364
##     baseRate percentAccuracy percentAccuracyByChance
## 18 0.3636364        74.54545                53.71901
##    relativeImprovementOverChance relativeImprovementOverPredictingFromBaseRate
## 18                          0.45                                          0.15
##    TPrate sensitivity TNrate specificity FNrate FPrate youdenJ
## 18   0.65        0.65    0.8         0.8   0.35    0.2    0.45
##    positivePredictiveValue negativePredictiveValue falseDiscoveryRate
## 18                    0.65                     0.8               0.35
##    falseOmissionRate balancedAccuracy f1Score  mcc diagnosticOddsRatio
## 18               0.2            0.725    0.65 0.45            7.428571
##    positiveLikelihoodRatio negativeLikelihoodRatio dPrimeSDT  betaSDT      cSDT
## 18                    3.25                  0.4375  1.226942 1.323034 0.2281504
##        ASDT   bSDT overallUtility utilityRatio informationGain
## 18 0.753125 0.6875           0.65          0.5       0.1465904
```

```
accuracyStats[which(accuracyStats$youdenJ == max(accuracyStats$youdenJ, na.rm = TRUE)),]
```

```
##    cutoff TP TN FP FN differenceBetweenPredictedAndObserved   N selectionRatio
## 18   0.22 26 56 14 14                                -0.269 110      0.3636364
##     baseRate percentAccuracy percentAccuracyByChance
## 18 0.3636364        74.54545                53.71901
##    relativeImprovementOverChance relativeImprovementOverPredictingFromBaseRate
## 18                          0.45                                          0.15
##    TPrate sensitivity TNrate specificity FNrate FPrate youdenJ
## 18   0.65        0.65    0.8         0.8   0.35    0.2    0.45
##    positivePredictiveValue negativePredictiveValue falseDiscoveryRate
## 18                    0.65                     0.8               0.35
##    falseOmissionRate balancedAccuracy f1Score  mcc diagnosticOddsRatio
## 18               0.2            0.725    0.65 0.45            7.428571
##    positiveLikelihoodRatio negativeLikelihoodRatio dPrimeSDT  betaSDT      cSDT
## 18                    3.25                  0.4375  1.226942 1.323034 0.2281504
##        ASDT   bSDT overallUtility utilityRatio informationGain
## 18 0.753125 0.6875           0.65          0.5       0.1465904
```

7.7.5 Closest to the Top Left of the ROC Curve

7.7.5.1 Threshold

Threshold where the ROC plot is closest to the Top Left:

```
closestToTheTopLeft <- coords(roc(data = mydataSDT, response = disorder, predictor = te
```

7.7.5.2 Accuracy stats at cutoff where the ROC plot is closest to the Top Left

```
accuracyStats[head(which(accuracyStats$cutoff >= closestToTheTopLeft), 1),]
```

```
##    cutoff TP TN FP FN differenceBetweenPredictedAndObserved   N selectionRatio
## 18   0.22 26 56 14 14                                -0.269 110      0.3636364
##     baseRate percentAccuracy percentAccuracyByChance
## 18 0.3636364        74.54545                53.71901
##    relativeImprovementOverChance relativeImprovementOverPredictingFromBaseRate
## 18                          0.45                                          0.15
##    TPrate sensitivity TNrate specificity FNrate FPrate youdenJ
## 18   0.65        0.65    0.8         0.8   0.35    0.2    0.45
##    positivePredictiveValue negativePredictiveValue falseDiscoveryRate
## 18                    0.65                     0.8               0.35
##    falseOmissionRate balancedAccuracy f1Score  mcc diagnosticOddsRatio
## 18               0.2            0.725    0.65 0.45            7.428571
```

```
##    positiveLikelihoodRatio negativeLikelihoodRatio dPrimeSDT  betaSDT      cSDT
## 18                    3.25                  0.4375  1.226942 1.323034 0.2281504
##        ASDT   bSDT overallUtility utilityRatio informationGain
## 18 0.753125 0.6875           0.65          0.5       0.1465904
```

7.7.6 Cutoff that optimizes each of the following criteria:

7.7.6.1 Percent Accuracy

```
accuracyStats$cutoff[which(accuracyStats$percentAccuracy == max(accuracyStats$percentAccuracy, na
```

```
## [1] 0.22 0.52
```

7.7.6.2 Percent Accuracy by Chance

```
accuracyStats$cutoff[which(accuracyStats$percentAccuracyByChance == max(accuracyStats$percentAccu
```

```
## [1] 2.08
```

7.7.6.3 Relative Improvement Over Chance (ROIC)

```
accuracyStats$cutoff[which(accuracyStats$relativeImprovementOverChance == max(accuracyStats$relat
```

```
## [1] 0.07
```

7.7.7 Relative Improvement Over Predicting from the Base Rate

```
accuracyStats$cutoff[which(accuracyStats$relativeImprovementOverPredictingFromBaseRate == max(acc
```

```
## [1] 0.22 0.52
```

7.7.7.1 Sensitivity

```
accuracyStats$cutoff[which(accuracyStats$sensitivity == max(accuracyStats$sensitivity,
```

```
## [1] 0.03
```

7.7.7.2 Specificity

```
accuracyStats$cutoff[which(accuracyStats$specificity == max(accuracyStats$specificity,
```

```
##  [1] 0.52 0.56 0.58 0.70 0.71 0.74 0.77 0.82 0.86 0.96 2.07 2.08
```

7.7.7.3 Positive Predictive Value

```
accuracyStats$cutoff[which(accuracyStats$positivePredictiveValue == max(accuracyStats$p
```

```
##  [1] 0.52 0.56 0.58 0.70 0.71 0.74 0.77 0.82 0.86 0.96 2.07
```

7.7.7.4 Negative Predictive Value

```
accuracyStats$cutoff[which(accuracyStats$negativePredictiveValue == max(accuracyStats$n
```

```
## [1] 0.07
```

7.7.7.5 Youden's J Statistic

```
accuracyStats$cutoff[which(accuracyStats$youdenJ == max(accuracyStats$youdenJ, na.rm =
```

```
## [1] 0.22
```

7.7.7.6 Balanced Accuracy

```
accuracyStats$cutoff[which(accuracyStats$balancedAccuracy == max(accuracyStats$balanced
```

```
## [1] 0.22
```

7.7.7.7 F1 Score

```
accuracyStats$cutoff[which(accuracyStats$f1Score == max(accuracyStats$f1Score, na.rm = TRUE))]
```

```
## [1] 0.22
```

7.7.7.8 Matthews Correlation Coefficient

```
accuracyStats$cutoff[which(accuracyStats$mcc == max(accuracyStats$mcc, na.rm = TRUE))]
```

```
## [1] 0.52
```

7.7.7.9 Diagnostic Odds Ratio

```
accuracyStats$cutoff[which(accuracyStats$diagnosticOddsRatio == max(accuracyStats$diagnosticOddsR
```

```
## [1] 0.49
```

7.7.7.10 Positive Likelihood Ratio

```
accuracyStats$cutoff[which(accuracyStats$positiveLikelihoodRatio == max(accuracyStats$positiveLik
```

```
## [1] 0.49
```

7.7.7.11 Negative Likelihood Ratio

```
accuracyStats$cutoff[which(accuracyStats$negativeLikelihoodRatio == min(accuracyStats$negativeLik
```

```
## [1] 0.07
```

7.7.7.12 d' Sensitivity

```
accuracyStats$cutoff[which(accuracyStats$dPrimeSDT == max(accuracyStats$dPrimeSDT, na.r
```

```
## [1] 0.49
```

7.7.7.13 A (Non-Parametric) Sensitivity

```
accuracyStats$cutoff[which(accuracyStats$ASDT == max(accuracyStats$ASDT, na.rm = TRUE))
```

```
## [1] 0.22
```

7.7.7.14 β Bias

```
accuracyStats$cutoff[which(abs(accuracyStats$betaSDT) == min(abs(accuracyStats$betaSDT)
```

```
## [1] 0.07
```

7.7.7.15 c Bias

```
accuracyStats$cutoff[which(abs(accuracyStats$cSDT) == min(abs(accuracyStats$cSDT), na.r
```

```
## [1] 0.16
```

7.7.7.16 b (Non-Parametric) Bias

```
accuracyStats$cutoff[which(abs(accuracyStats$bSDT) == min(abs(accuracyStats$bSDT), na.r
```

```
## [1] 0.07
```

7.7.7.17 Mean difference between predicted and observed values (Miscalibration)

```
accuracyStats$cutoff[which(abs(accuracyStats$differenceBetweenPredictedAndObserved) ==
```

```
## [1] 0.14 0.15 0.16
```

7.7.7.18 Overall Utility

```
accuracyStats$cutoff[which(accuracyStats$overallUtility == max(accuracyStats$overallUtility, na.r
```

```
## [1] 0.22
```

7.7.7.19 Information Gain

```
accuracyStats$cutoff[which(accuracyStats$informationGain == max(accuracyStats$informationGain, na
```

```
## [1] 0.22
```

7.8 Regression for Prediction of Continuous Outcomes

When predicting a continuous outcome regression is particularly relevant (or multiple regression, when dealing with multiple predictors). Regression takes the following form:

$$y = b_0 + b_1 \cdot x_1 + e$$

where y is the outcome, b_0 is the intercept, b_1 is the slope of the association between the predictor (x_1) and outcome, and e is the error term.

7.8.1 Pseudo-Prediction

Consider the following example where you have one predictor and one outcome, as shown in Table 7.1.

Using the data, the best fitting regression model is: $y = 3.98 + 3.59 \cdot x_1$. In this example, the R^2 is 0.98. The equation is not a perfect prediction, but with a single predictor, it captures the majority of the variance in the outcome.

Now consider the following example where you add a second predictor to the data above, as shown in Table 7.2.

With the second predictor, the best fitting regression model is: $y = 0.00 + 4.00 \cdot x_1 + 1.00 \cdot x_2$. In this example, the R^2 is 1.00. The equation with the second predictor provides a perfect prediction of the outcome.

Table 7.1: Example data of predictor (x1) and outcome (y) used for regression model.

y	x1
7	1
13	2
29	7
10	2

Table 7.2: Example data of predictors (x1 and x2) and outcome (y) used for regression model.

y	x1	x2
7	1	3
13	2	5
29	7	1
10	2	2

Providing perfect prediction with the right set of predictors is the dream of multiple regression. So, in psychology, we often add predictors to incrementally improve prediction. Knowing how much variance would be accounted for by random chance follows Equation (7.8.1):

$$E(R^2) = \frac{K}{n-1}$$

where $E(R^2)$ is the expected value of R^2 (the proportion of variance explained), K is the number of predictors, and n is the sample size. The formula demonstrates that the more predictors in the regression model, the more variance will be accounted for by chance. With many predictors and a small sample, you can account for a large share of the variance merely by chance—this would be an example of pseudo-prediction.

As an example, consider that we have 13 predictors to predict children's behavior problems for 43 children. Assume that, with 13 predictors, we explain 38% of the variance ($R^2 = .38; r = .62$). Explaining more than 20–30% of the variance can be a big deal in psychology. We explained a lot of the variance in the outcome, but it is important to consider how much variance could have been explained by random chance: $E(R^2) = \frac{K}{n-1} = \frac{13}{43-1} = .31$. We expect to explain 31% of the variance, by chance, in the outcome. So, 82% of the variance explained was likely spurious. As the sample size increases, the spuriousness decreases. Adjusted R^2 accounts for the number of predictors in the model, based on how

Table 7.3: Example data of predictors (x1 and x2) and outcome (y) used for regression model.

y	x1	x2
9	2.0	4
11	3.0	6
17	4.0	8
3	1.0	2
21	5.0	10
13	3.5	7

much would be expected to be accounted for by chance. But, adjusted R^2 also has its problems.

7.8.2 Multi-Collinearity

Multi-collinearity occurs when two or more predictors in a regression model are highly correlated. The problem is that it makes it challenging to estimate the regression coefficients accurately.

Consider the following example where you have two predictors and one outcome, as shown in Table 7.3.

The second measure is not very good—it is exactly twice the value of the first measure. This means that there are different prediction equation possibilities that are equally good:

$$\begin{aligned} 2x_2 &= y \\ 0x_1 + 2x_2 &= y \\ 4x_1 &= y \\ 4x_1 + 0x_2 &= y \\ 2x_1 + 1x_2 &= y \\ 5x_1 - 0.5x_2 &= y \\ ... &= y \end{aligned}$$

Then, what are the regression coefficients? We do not know, and we could come up with arbitrary estimates with enormous standard error around each estimate. Any predictors that have a correlation above $\sim r = .30$ with each other could have an impact on the confidence interval of the regression coefficient. As the correlation among the predictors increase, the chance of getting an arbitrary answer increases. So, it is important to examine a correlation matrix of the predictors before putting them in the same regression model. You can

also examine indices such as variance inflation factor (VIF). To address multicollinearity, you can drop the a redundant predictor or you can also use principal component analysis or factor analysis of the predictors to reduce the predictors down to a smaller number of meaningful predictors. For a meaningful answer in a regression framework that is precise and confident, you need a low level of intercorrelation among predictors.

7.9 Ways to Improve Prediction Accuracy

On the whole, experts' predictions are inaccurate. Experts' predictions from many different domains tend to be inaccurate, including political scientists [Tetlock, 2017], physicians [Koehler et al., 2002], clinical psychologists [Oskamp, 1965], stock market traders and corporate financial officers [Skala, 2008], seismologists' predictions of earthquakes [Hough, 2016], economists' predictions about the economy [Makridakis et al., 2009], lawyers [Koehler et al., 2002], and business managers [Russo and Schoemaker, 1992]. The most common pattern of experts' predictions is that they show overextremity, that is, their predictions have probabilty judgments that tend to be too extreme, as described in Section 7.4.3. Overextremity of experts' predictions likely reflects overconfidence. The degree of confidence of a person's predictions is often not a good indicator of the accurcay of their predictions [and confidence and prediction accuracy are sometimes inversely associated; Silver [2012]]. Cognitive biases including the anchoring bias [Tversky and Kahneman, 1974], the confirmation bias [Hoch, 1985, Koriat et al., 1980], and base rate neglect [Eddy, 1982, Koehler et al., 2002] could contribute to overconfidence of predictions. Poorly calibrated predictions are especially likely when the base rate is very low (e.g., suicide), as is often the case in clinical psychology, or when the base rate is very high [Koehler et al., 2002].

Nevertheless, there are some domains that have shown greater predictive accuracy, from which we may learn what practices may lead to greater accuracy. For instance, experts have shown stronger predictive accuracy in weather forecasting [Murphy and Winkler, 1984], horse race betting [Johnson and Bruce, 2001], and playing the card game of bridge [Keren, 1987], but see Koehler et al. [2002] for exceptions.

Here are some potential ways to improve the accuracy (and honesty) of predictions and judgments:

- Provide appropriate anchoring of your predictions to the base rate of the phenomenon you are predicting. To the extent that the base rate of the event you are predicting is low, more extreme evidence should be necessary to consistently and accurately predict that the event will occur. Applying Bayes' theorem and Bayesian approaches can help you appropriately weigh base rates and evidence.

- Include multiple predictors, ideally from different measures and measurement methods. Include the predictors with the strongest validity based on theory of the causal process and based on criterion-related validity.
- When possible, aggregate multiple perspectives of predictions, especially predictions made independently (from different people/methods/etc.). The "wisdom of the crowd" is often more accurate than individual's predictions, including predictions by so-called "experts" [Silver, 2012].
- A goal of prediction is to capture as much signal as possible and as little noise (error) as possible [Silver, 2012]. Parsimony (i.e., not having too many predictors) can help reduce the amount of error variance captured by the prediction model. However, to accurately model complex systems like human behavior, the brain, etc., complex models may be necessary. However, strong theory of the causal processes and dynamics may be necessary to develop accurate complex models.
- Although incorporating theory can be helpful, provide more weight to empiricism than to theory (until our theories and measures are stronger). Ideally, we would use theory to design a model that mirrors the causal system, with accurate measures of each process in the system, so we could make accurate predictions. However, as described in Section 4.3.1.3.3, our psychological theories of the causal processes that influence outcomes are not yet very strong. Until we have stronger theories that specify the causal process for a given outcome, and until we have accurate measures of those causal processes, actuarial approaches are likely to be most accurate, as discussed in Chapter 8. At the same time, keep in mind that measures in psychology, and their resulting data, are often noisy. As a result, theoretically (conceptually) informed empirical approaches may lead to more accuracy than empiricism alone.
- Use an empirically validated and cross-validated statistical algorithm to combine information from the predictors in a formalized way. Give each predictor appropriate weight in the statistical algorithm, according to its strength of association with the outcome. Use measures with strong reliability and validity for assessing these processes to be used in the algorithm. Cross-validation will help reduce the likelihood that your model is fitting to noise and will maximize the likelihood that the model predicts accurately when applied to new data (i.e., the model's predictions accurately generalize).
- When presenting your predictions, acknowledge what you do not know.
- Express your predictions in terms of probabilistic estimates and present the uncertainty in your predictions with confidence intervals [even though bolder, more extreme predictions tend to receive stronger television ratings; Silver [2012]].
- Qualify your predictions by identifying and noting counter-examples that would not be well fit by your prediction model, such as extreme cases, edge cases, and "broken leg" cases.
- Provide clear, consistent, and timely feedback on the outcomes of the predictions to the people making the predictions [Bolger and Önkal Atay,

2004].

- Be self-critical about your predictions. Update your judgments based on their accuracy, rather than trying to confirm your beliefs [Atanasov et al., 2020].
- In addition to considering the accuracy of the prediction, consider the quality of the prediction *process*, especially when random chance is involved to a degree (such as in poker) [Silver, 2012].
- Work to identify and mitigate potential blindspots; be aware of cognitive biases, such as confirmation bias and base rate neglect.

7.10 Conclusion

7.11 Exercises

7.11.1 Questions

Note: several of the following questions use data from the survivorship of the Titanic accident. The data are publicly available (http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets). The Titanic data file for these exercises is located on the course's page of the Open Science Framework (https://osf.io/3pwza). Every record in the data set represents a passenger—including the passenger's age, sex, passenger class, number of siblings/spouses aboard (`sibsp`), number of parents/children aboard (`parch`) and, whether the passenger survived the accident. I used these variables to create a prediction model (based on a logistic regression model using Leave-10-out cross-validation) for whether the passenger survived the accident. The model's prediction for the passenger's likelihood of survival are in the variable called "prediction".

1. What are the two main types of prediction accuracy? Define each. How can you quantify each? What is an example of an index that combines both main types of prediction accuracy?
2. Provide the following indexes of overall prediction accuracy for the prediction model in predicting whether a passenger survived the Titanic:

    a. Mean error (bias)
    b. Mean absolute error (MAE)
    c. Mean squared error (MSE)
    d. Root mean squared error (RMSE)
    e. Mean percentage error (MPE)
    f. Mean absolute percentage error (MAPE)
    g. Symmetric mean absolute percentage error (sMAPE)
    h. Mean absolute scaled error (MASE)
    i. Root mean squared log of the accuracy ratio

   j. Coefficient of determination (R^2)
   k. Adjusted R^2 (R^2_{adj})
   l. Predictive R^2

3. Based on the mean error for the prediction model you found in `2a`, what does this indicate?
4. Create two receiver operating characteristic (ROC) curves for the prediction model in predicting whether a passenger survived the Titanic: one empirical ROC curve and one smooth ROC curve.
5. What is the area under the ROC curve (AUC) for the prediction model in predicting whether a passenger survived the Titanic? What does this indicate?
6. Create a calibration plot. Are the predictions well-calibrated? Provide empirical evidence and support your inferences with interpretation of the calibration plot. If the predictions are miscalibrated, describe the type of miscalibration present.
7. You predict that a passenger survived the Titanic if their predicted probability of survival is .50 or greater. Create a confusion matrix (2x2 matrix of prediction accuracy) for Titanic survival using this threshold. Make sure to include the marginal cells. Label each cell. Enter the number and proportion in each cell.
8. Using the 2x2 prediction matrix, identify or calculate the following:

   a. Selection ratio
   b. Base rate
   c. Percent accuracy
   d. Percent accuracy by chance
   e. Percent accuracy predicting from the base rate
   f. Relative improvement over chance (ROIC)
   g. Relative improvement over predicting from the base rate
   h. Sensitivity (true positive rate [TPR])
   i. Specificity (true negative rate [TNR])
   j. False negative rate (FNR)
   k. False positive rate (FPR)
   l. Positive predictive value (PPV)
   m. Negative predictive value (NPV)
   n. False discovery rate (FDR)
   o. False omission rate (FOR)
   p. Diagnostic odds ratio
   q. Youden's J statistic
   r. Balanced accuracy
   s. F_1 score
   t. Matthews correlation coefficient (MCC)
   u. Positive likelihood ratio
   v. Negative likelihood ratio
   w. d' sensitivity
   x. A (non-parametric) sensitivity

y. β bias
z. c bias aa. b (non-parametric) bias
ab. Miscalibration (mean difference between predicted and observed values; based on 10 groups)
ac. Information gain

9. In terms of percent accuracy, how much more accurate are the predictions compared to predicting from the base rate? What would happen to sensitivity and specificity if you raise the selection ratio? What would happen if you lower the selection ratio?
10. For your assessment goals, it is 4 times more important to identify survivors than to identify non-survivors. Consistent with these assessment goals, you specify the following utility for each of the four outcomes: hits: 1; misses: 0; correct rejections: 0.25; false alarms: 0. What is the utility ratio? What is the overall utility (U_{overall}) of the current cutoff? What cutoff has the highest overall utility?
11. Find the optimal cutoff threshold that optimizes each of the following criteria:

    a. Youden's J statistic
    b. Closest to the top left of the ROC curve
    c. Percent accuracy
    d. Percent accuracy by chance
    e. Relative improvement over chance (ROIC)
    f. Relative improvement over predicting from the base rate
    g. Sensitivity (true positive rate [TPR])
    h. Specificity (true negative rate [TNR])
    i. Positive predictive value (PPV)
    j. Negative predictive value (NPV)
    k. Diagnostic odds ratio
    l. Balanced accuracy
    m. F_1 score
    n. Matthews correlation coefficient (MCC)
    o. Positive likelihood ratio
    p. Negative likelihood ratio
    q. d' sensitivity
    r. A (non-parametric) sensitivity
    s. β bias
    t. c bias
    u. b (non-parametric) bias
    v. Miscalibration (mean difference between predicted and observed values; based on 10 groups)
    w. Overall utility
    x. Information gain

12. A company develops a test that seeks to determine whether someone has been previously infected with a novel coronavirus (COVID-75) based on

the presence of antibodies in their blood. You take the test and your test result is negative (i.e., the test says that you have not been infected). Your friend takes the test and their test result is positive for coronavirus (i.e., the test says that your friend has been infected). Assume the prevalence of the coronavirus is 5%, the sensitivity of the test is .90, and the specificity of the test is .95.

a. What is the probability that you actually had the coronavirus?
b. What is the probability that your friend actually had the coronavirus?
c. Your friend thinks that, given their positive test, that they have the antibodies (and thus may have immunity). What is the problem with your friend's logic?
d. What logical fallacy is your friend demonstrating?
e. Why is it challenging to interpret a positive test in this situation?

13. You just took a screening test for a genetic disease. Your test result is positive (i.e., the tests says that you have the disease). Assume the probability of having the disease is 0.5%, the probability of a positive test is 2%, and the probability of a positive test if you have the disease is 99%. What is the probability that you have the genetic disease?
14. You just took a screening test (Test A) for a virus. Your test result is positive. Assume the base rate of the virus is .05%. Test A has a sensitivity of .95 and a specificity of .90.

    a. What is your probability of having the virus after testing positive on Test A?
    b. After getting the results back from Test A, the physician wants greater confidence regarding whether you have the virus given its low base rate, so the physician orders a second test, Test B. You test positive on Test B. Test B has a sensitivity of .80 and a specificity of .95. Assuming the errors of the Tests A and B are independent, what is your updated probability of having the virus?

7.11.2 Answers

1. The two main types of prediction accuracy are discrimination and calibration. Discrimination refers to the ability of a prediction model to separate/differentiate data into classes (e.g., survived versus died). Calibration for a prediction model refers to how well the predicted probability of an event (e.g., survival) matches the true probability of an event. You can quantify discrimination with AUC; you can quantify various aspects of discrimination with sensitivity, specificity, positive predictive value, and negative predictive value). You can quantify degree of (poor) calibration with Spiegelhalter's z, though other metrics also exist (e.g., Brier scores and the Hosmer-Lemeshow goodness-of-fit statistic). R^2 is an overall index of accuracy that combines both discrimination and calibration.

2. a. Mean error (bias): 0.0006
   b. Mean absolute error (MAE): 0.30
   c. Mean squared error (MSE): 0.15
   d. Root mean squared error (RMSE): 0.39
   e. Mean percentage error (MPE): undefined, but when dropping undefined values: 36.74%
   f. Mean absolute percentage error (MAPE): undefined, but when dropping undefined values: 36.74%
   g. Symmetric mean absolute percentage error (sMAPE): 70.16%
   h. Mean absolute scaled error (MASE): 0.62
   i. Root mean squared log of the accuracy ratio: undefined, but when dropping undefined values: 0.96
   j. Coefficient of determination (R^2): .37
   k. Adjusted R^2 (R^2_{adj}): .37
   l. Predictive R^2: .37

3. The small mean error/bias (0.0006) indicates that predictions did not consistently under- or over-estimate the actual values to a considerable degree.

4. Empirical ROC curve:

Smooth ROC curve:

5. The AUC is .84. AUC indicates the probability that a randomly selected case has a higher test result than a randomly selected control. Thus, the probability is 84% that a randomly selected passenger who survived had a higher predicted probability of survival (based on the prediction model) than a randomly selected passenger who died. The AUC of .84 indicates that the prediction model was moderately accurate in terms of discrimination.
6. Calibration plot:

6. In general, the predictions are well-calibrated. There is not significant miscalibration according to Spiegehalter's z $(0.31, p = .76)$. This is verified graphically in the calibration plot, in which the predicted probabilities fall mostly near the actual/observed probabilities. However, based on the calibration plot, there does appear to be some miscalibration. When the predicted probability of survival was ~60%, the actual probability of survival was lower (~40%). This pattern of miscalibration is known as overprediction, as depicted in Figure 1 of Lindhiem et al. [2020].

7. The 2x2 prediction matrix is below:

Figure 7.35: Exercise 4: Empirical receiver operating characteristic curve

Figure 7.36: Exercise 4: Smooth receiver operating characteristic curve

Figure 7.37: Exercise 5: Calibration plot of predicted probability versus observed probability

		Actual Outcome		Total Predicted	Selection Ratio
		Survived	Died		
Prediction	Survived	TP = 299 (.286)	FP = 108 (.103)	407	SR = 0.39
	Died	FN = 128 (.122)	TN = 511 (.489)	639	1 – SR = 0.61
Total Actual		427	619	*N* = 1,046	
Base Rate		BR = 0.41	1 – BR = 0.59		

Figure 7.38: Exercise 6: 2x2 prediction matrix

8. a. Selection ratio: .39

   b. Base rate: .41

   c. Percent accuracy: 77%

   d. Percent accuracy by chance: 52%

   e. Percent accuracy predicting from the base rate: 59%

   f. Relative improvement over chance (ROIC): .51

   g. Relative improvement over predicting from the base rate: .22

   h. Sensitivity (true positive rate [TPR]): .70

   i. Specificity (true negative rate [TNR]): .83

   j. False negative rate (FNR): .30

   k. False positive rate (FPR): .17

   l. Positive predictive value (PPV): .73

   m. Negative predictive value (NPV): .80

   n. False discovery rate (FDR): .27

   o. False omission rate (FOR): .20

   p. Diagnostic odds ratio: 11.05

   q. Youden's J statistic: .53

   r. Balanced accuracy: .76

   s. F_1 score: 0.72

   t. Matthews correlation coefficient (MCC): .53

   u. Positive likelihood ratio: 4.01

   v. Negative likelihood ratio: 0.36

   w. d' sensitivity: 1.46

x. A (non-parametric) sensitivity: 0.83

y. β bias: 1.35

z. c bias: 0.21
aa. b (non-parametric) bias: 1.30
ab. Miscalibration (mean difference between predicted and observed values): 0.16
ac. Information gain: 0.21

9. Predicting from the base rate would have a percent accuracy of 59%. So, the predictions increase the percent accuracy by 18%. If you raise the selection ratio (i.e., predict more people survived) sensitivity will increase whereas specificity will decrease. If you lower the selection ratio, specificity will increase and sensitivity will decrease.

10. The utility ratio is 0.25. The overall utility (U_{overall}) of the cutoff is 0.41. The cutoff with the highest overall utility is 0.266.

11. The cutoff that optimizes each of the following criteria:

    a. Youden's J statistic: 0.556

    b. Closest to the top left of the ROC curve: 0.391

    c. Percent accuracy: 0.618

    d. Percent accuracy by chance: 1 (i.e., predicting from the base rate—that nobody will survive)

    e. Relative improvement over chance (ROIC): .017, .023, .027, .031, .032, .035, .035, .035, .038, .039

    f. Relative improvement over predicting from the base rate: .618

    g. Sensitivity (true positive rate [TPR]): 0 (i.e., predicting that everyone will survive will minimize false negatives)

    h. Specificity: 1 (i.e., predicting that nobody will survive will minimize false positives)

    i. Positive predictive value (PPP): 0.875

    j. Negative predictive value (NPV): 0.017, 0.023, 0.027, 0.031, 0.032, 0.035, 0.035, 0.035, 0.038, 0.039

k. Diagnostic odds ratio: 0.875

l. Balanced accuracy: 0.556

m. F_1 score: 0.391

n. Matthews correlation coefficient (MCC): 0.618

o. Positive likelihood ratio: 0.875

p. Negative likelihood ratio: 0.017, 0.023, 0.027, 0.031, 0.032, 0.035, 0.035, 0.035, 0.038, 0.039

q. Miscalibration (mean difference between predicted and observed values): 0.017–0.085

r. d' sensitivity: 0.833

s. A (non-parametric) sensitivity 0.316

t. β bias 0.017, 0.023, 0.027, 0.031, 0.032, 0.035, 0.035, 0.035, 0.038, 0.039, 0.979

u. c bias 0.387

v. b (non-parametric) bias 0.017

w. Overall utility: 0.266

x. Information gain: 0.631

12. a. Based on negative predictive value (i.e., the probability of no disease given a negative test), the probability that you actually had the coronavirus is less than 1 in 100 (.006).
    b. Based on positive predictive value (i.e., the probability of disease given a positive test), the probability that your friend actually had the coronavirus is less than 50% (.49).
    c. The problem with your friend's logic is that your friend is assuming they have the antibodies when chances are more likely that they do not.
    d. Your friend is likely demonstrating the fallacy of base rate neglect.
    e. A positive test on a screening device is hard to interpret in this situation because of the low base rate. "Even with a very accurate test, the fewer people in a population who have a condition, the more likely it is that an individual's positive result is wrong." For more info, see here: https://www.scientificamerican.com/article/

coronavirus-antibody-tests-have-a-mathematical-pitfall/

13. According to Bayes' theorem, the probability that you have the disease is 24.75%. For more info, see here: https://www.scientificamerican.com/article/what-is-bayess-theorem-an/

14. a. According to Bayes' theorem, the probability that you have the virus after testing positive on Test A is 4.6%.
    b. According to Bayes' theorem, the updated probability that you have the virus after testing positive on both Tests A and B is 43.3%.

Chapter 8

Clinical Judgment vs. Algorithmic Prediction

8.1 Approaches to Prediction

8.1.1 Human/Clinical Judgment

Using the clinical judgment method of prediction, all gathered information is collected and formulated into a diagnosis or prediction in the clinician's mind. The clinician selects, measures, and combines risk factors and produces risk estimate solely according to clinical experience and judgment

8.1.2 Actuarial/Statistical Method

In the actuarial or statistical method of prediction, information is gathered and combined systematically in an evidence-based statistical prediction formula, and established cutoffs are used. The method is based on equations and data, so both are needed.

An example of a statistical method of prediction is the Violence Risk Appraisal Guide [**CITATION**]. The Violence Risk Appraisal Guide is used in an attempt to predict violence and is used for parole decisions. For instance, the equation might be something like:

$$\text{Violence risk} = \beta\cdot\text{conduct disorder}+\beta\cdot\text{substance use}+\beta\cdot\text{suspended from school}+\beta\cdot\text{childhood aggression}+...$$

Then, based on their score and the established cutoffs, a person is given a "low risk," "medium risk," or "high risk" designation.

8.1.3 Combining Human Judgment and Statistical Algorithms

There are numerous ways in which humans and statistical algorithms could be involved.

[On one extreme… On the other extreme… In the middle…]

Qualitative information can be scored or coded to be quantified so that it can be included in statistical formulas.

"garbage in, garbage out"

The challenge is what to do when the human and the algorithm disagree. Hypothetically, humans reviewing and adjusting the results from the statistical algorithm could lead to more accurate predictions. However, human input also could lead to the possibility or exacerbation of biased predictions.

8.2 Errors in Clinical Judgment

Clinical judgment is naturally subject to errors. Here are a few:

Meehl [1957] acknowledged that there may be some situations where it is glaringly obvious that the statistical formula would be incorrect because it fails to account for an important factor. He called these special cases "broken leg" cases. The example goes like this:

> "If a sociologist were predicting whether Professor X would go to the movies on a certain night, he might have an equation involving age, academic specialty, and introversion score. The equation might yield a probability of .90 that Professor X goes to the movie tonight. But if the family doctor announced that Professor X had just broken his leg, no sensible sociologist would stick with the equation. Why didn't the factor of 'broken leg' appear in the formula? Because broken legs are very rare, and in the sociologist's entire sample of 500 criterion cases plus 250 cross-validating cases, he did not come upon a single instance of it. He uses the broken leg datum confidently, because 'broken leg' is a subclass of a larger class we may crudely denote as 'relatively immobilizing illness or injury,' and movie-attending is a subclass of a larger class of"actions requiring moderate mobility." [Meehl, 1957]

However, people too often think that cases where they disagree with the statistical algorithm are broken leg cases. People too often change the result of the statistical algorithm and are more likely to be wrong than right in doing so.

8.3 Humans versus Computers

8.3.1 Advantages of Computers

Here are some advantages of computers over humans:

- Computers can process lots of information simultaneously. So can humans. But computers can to an even greater degree.
- Computers are faster at making calculations.
- Computations by computers are error free (as long as the computations are programmed correctly)
- Computers' judgments will not be biased by fatigue or emotional responses.
- Computers' judgments will tend not to be biased in the way that humans' cognitive biases are, such as with anchoring bias, representativeness bias, confirmation bias, or recency bias. Computers are less likely to be overconfident in their judgments.
- Computers can more accurately weight the set of predictors based on large data sets. Humans tend to give too much weight to singular predictors.

8.3.2 Advantages of Humans

Computers are bad at some things too. There may be situations in which a clinical judgment would do better than an actuarial judgment. Here are some advantages of humans over computers (as of now):

- Humans can be better at identifying patterns in data (but also can mistakenly identify patterns where there are none)
- Humans can be flexible and take a different approach if a given approach is not working
- Humans are better at tasks requiring creativity and imagination, such as developing theories that explain phenomena
- Humans have the ability to reason, which is especially important when dealing with complex, abstract, or open-ended problems, or problems that have not been faced before (or for which we have insufficient data).
- Humans are better able to learn
- Humans are better at holistic, gestalt processing; this includes facial and linguistic processing
- [complex theories]

Figure 8.1

Figure 8.1: Depiction of the psychoanalytic tradition.

8.3.3 Comparison of Evidence

8.4 Accuracy of Different Statistical Models

8.5 Why Clinical Judgment is More Widely Used Than Statistical Formulas

8.6 Conclusion

Chapter 9

General Issues in Clinical Assessment

9.1 Terminology

Prevalence is the proportion of the population that has the condition a given point in time, whereas *incidence* is the rate of occurrence of new cases. Incidence indicates the risk of developing or contracting the condition, whereas prevalence indicates how widespread the condition is. *Point prevalence* is the proportion of the population that has the condition at a single point in time. *Lifetime prevalence* is the proportion of the population that will ever have a condition at any point in their lifetime.

The *positivity rate* is the proportion of tests that are positive. The *case fatality rate* is the proportion of people with a condition who die as a result of that condition.

9.2 The Three S's: Signs, Symptoms, and Syndromes

Signs are observable features (or manifestations) of a disorder [Lilienfeld et al., 2015]. Therefore, signs can be perceived by a clinician. By contrast, *symptoms* are unobservable manifestations of a disorder that can only be perceived by the client [Lilienfeld et al., 2015]. The term "sign" has a connotation of being somewhat more objective than symptom because a symptom is based on the client's perceptions, but at least in clinical psychology (and also in many medical fields), both signs and symptoms have considerable elements of subjectivity. A

syndrome is a collection of signs and symptoms that co-occur and may reflect a particular disorder.

9.3 Conclusion

Chapter 10

Evidence-Based Assessment

[**Ursenbach et al, 2019 citation**]

As presented in Equation (7.1.1.1), the posttest (or posterior) odds are equal to the pretest odds multiplied by the likelihood ratio. Bayes' theorem is discussed in Section 7.1.1.1. Using this formula, and converting odds to probabilities, we can use a Fagan probability nomogram to determine the posttest probability following a test result. The calculation of posttest probability is described in Section 7.5.32. A *probability nomogram* is a way of visually applying Bayes' theorem to determine the posttest probability of having a condition based on the pretest (or prior) probability and likelihood ratio. To use a probability nomogram, connect the dots from the starting probability (left line) with the likelihood ratio (middle line) to see the updated probability. The updated (posttest) probability is where the connecting line crosses the third, right line.

\begin{figure}

{

}

\caption{Probability Nomogram. Figure retrieved from https://upload.wikimedia.org/wikipedia/commons/thumb/6/66/Fagan_nomogram.svg/945px-Fagan_nomogram.svg.png} \end{figure}

For instance, if the starting probability is .5% and the likelihood ratio is 10

(e.g., sensitivity = .90, specificity = .91:
likelihood ratio $= \frac{\text{sensitivity}}{1-\text{specificity}} = \frac{.9}{1-.91} = 10$), the updated probability is less than 5%. The formula and function for computing posttest probability are provided in Section 7.5.32.

```
posttestProbability(pretestProb = .005, likelihoodRatio = 10)
```

```
## [1] 0.04784689
```

\begin{figure}

{

}

\caption{Probability Nomogram Example. Figure adapted from https://upload.wikimedia.org/wikipedia/commons/thumb/6/66/Fagan_nomogram.svg/945px-Fagan_nomogram.svg.png} \end{figure}

10.1 Conclusion

Chapter 11

Ethical Issues in Assessment

Ethics are relevant to all of our domains as psychologists, including (but not limited to) research, teaching, assessment, intervention, and supervision.

11.1 Conclusion

Chapter 12

Factor Analysis and Principal Component Analysis

[**Wright et al, 2019 citation in psych assessment**; **Bensch et al, 2019 citation** ;**Tackett et al, 2019 citation on MTMM**; **Simms et al. 2019 citation**; **Clark & Watson, 2019 citation**; **Sellbom & Tellegen, 2019**; **Wright & Zimmerman, 2019 citation**; **Burchett et al, 2019 citation**; **Chandler et al, 2020 citation**]

[**Costa et al, 2019 citation**]

[**Markon, 2019 article on bifactor models**]

12.1 Getting Started

12.1.1 Load Libraries

```
library("lavaan")
library("psych")
library("corrplot")
library("nFactors")
library("semPlot")
library("lavaan")
library("semTools")
library("dagitty")
```

```
library("tidyverse")
library("here")
library("tinytex")
library("knitr")
library("rmarkdown")
```

12.1.2 Load Data

The MTMM data file comes from W. Joel Schneider:
http://my.ilstu.edu/~wjschne/442/MTMM.csv

```
MTMM <- read_csv(here("Data", "MTMM.csv")) #read_csv("../Data/MTMM.csv")
```

12.1.3 Prepare Data

12.1.3.1 Add Missing Data

Adding missing data to dataframes helps make examples more realistic to real-life data, and helps you get in the habit of programming to account for missing data. For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
set.seed(52242)

varNames <- names(HolzingerSwineford1939)
dimensionsDf <- dim(HolzingerSwineford1939)
unlistedDf <- unlist(HolzingerSwineford1939)
unlistedDf[sample(1:length(unlistedDf), size = .15 * length(unlistedDf))] <- NA
HolzingerSwineford1939 <- as.data.frame(matrix(unlistedDf, ncol = dimensionsDf[2]))
names(HolzingerSwineford1939) <- varNames
```

12.2 Descriptive Statistics and Correlations

12.2.1 Descriptive Statistics

```
paged_table(psych::describe(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
```

```
kable(psych::describe(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")]),
      booktabs = TRUE)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	
x1	1	254	4.960630	1.164728	5.000000	4.996732	1.2355000	0.6666667	8.500000	7.833333	-0.3
x2	2	253	6.102767	1.177801	6.000000	6.029557	1.1119500	2.2500000	9.250000	7.000000	0.4
x3	3	259	2.263996	1.133360	2.125000	2.218900	1.2972750	0.2500000	4.500000	4.250000	0.3
x4	4	256	3.085938	1.146176	3.000000	3.042071	0.9884000	0.3333333	6.333333	6.000000	0.3
x5	5	243	4.260288	1.290019	4.500000	4.315385	1.4826000	1.0000000	6.750000	5.750000	-0.3
x6	6	259	2.145063	1.098601	2.000000	2.041695	1.0590000	0.1428571	6.142857	6.000000	0.9
x7	7	255	4.160614	1.065841	4.086957	4.137009	1.0313740	1.3043478	7.260870	5.956522	0.2
x8	8	258	5.480426	0.986827	5.450000	5.457692	0.8895600	3.0500000	9.100000	6.050000	0.3
x9	9	259	5.359181	1.004227	5.416667	5.350877	0.9472167	2.7777778	9.250000	6.472222	0.2

```
kable(psych::describe(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")]),
      booktabs = TRUE)
```

12.2.2 Correlations

Pairs panel plots were generated using the `psych` package [Revelle, 2020]. Correlation plots were generated using the `corrplot` package [Wei and Simko, 2017].

```
cor(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], use = "pairwise.com
```

```
##            x1          x2          x3         x4          x5        x6
## x1 1.00000000  0.24337355 0.437409290 0.36927561 0.303841799 0.3977263
## x2 0.24337355  1.00000000 0.320197143 0.17969150 0.087371474 0.1710913
## x3 0.43740929  0.32019714 1.000000000 0.16253905 0.005721348 0.1962540
## x4 0.36927561  0.17969150 0.162539051 1.00000000 0.719821135 0.7214570
## x5 0.30384180  0.08737147 0.005721348 0.71982114 1.000000000 0.7430381
## x6 0.39772632  0.17109127 0.196253983 0.72145701 0.743038088 1.0000000
## x7 0.07871661 -0.08715308 0.108872951 0.14986045 0.049425208 0.1369739
## x8 0.20426155  0.04856542 0.145302413 0.05582004 0.143301626 0.1537855
## x9 0.37198638  0.18263566 0.352981541 0.19175258 0.157930238 0.2615588
##             x7         x8        x9
## x1  0.07871661 0.20426155 0.3719864
## x2 -0.08715308 0.04856542 0.1826357
## x3  0.10887295 0.14530241 0.3529815
## x4  0.14986045 0.05582004 0.1917526
## x5  0.04942521 0.14330163 0.1579302
## x6  0.13697390 0.15378546 0.2615588
```

```
## x7  1.00000000 0.48196044 0.3561427
## x8  0.48196044 1.00000000 0.4327187
## x9  0.35614266 0.43271865 1.0000000
```

```
pairs.panels(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")])
```

Figure 12.1: Pairs panel plot

```
corrplot(cor(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")],
```

12.3 Factor Analysis

12.4 Exploratory Factor Analysis (EFA)

12.4.1 Determine number of factors

Determine the number of factors to retain using the Scree plot and Very Simple Structure plot.

Figure 12.2: Correlation plot

12.4.1.1 Scree Plot

Scree plots were generated using the `psych` [Revelle, 2020] and `nFactors` [Raiche and Magis, 2020] packages.

```
fa.parallel(x = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], fm = "m
```

```
## Parallel analysis suggests that the number of factors =  3  and the number of components =  NA
```

```
plot(nScree(x = cor(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], use
```

12.4.1.2 Very Simple Structure (VSS) Plot

Very simple structure (VSS) plots were generated using the `psych` package [Revelle, 2020].

Figure 12.3: Scree plots in exploratory factor analysis

```
vss(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate =
```

12.4.1.2.1 Orthogonal (Varimax) rotation

```
##
## Very Simple Structure
## Call: vss(x = HolzingerSwineford1939[, c("x1", "x2", "x3", "x4", "x5",
##     "x6", "x7", "x8", "x9")], rotate = "varimax", fm = "ml")
## VSS complexity 1 achieves a maximimum of 0.7  with  3  factors
## VSS complexity 2 achieves a maximimum of 0.85  with  4  factors
##
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
## BIC achieves a minimum of  -33.84  with  3  factors
## Sample Size adjusted BIC achieves a minimum of  -12.54  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.55 0.00 0.083  27 3.3e+02 1.3e-53     7.3 0.55 0.193 174.7 260.3     1.0
## 2 0.65 0.76 0.066  19 1.4e+02 6.0e-21     3.9 0.76 0.147  34.1  94.3     1.2
```

Figure 12.4: Scree plots in exploratory factor analysis

```
## 3 0.70 0.84 0.072  12 3.5e+01 5.3e-04     2.2 0.86 0.079 -33.8   4.2     1.3
## 4 0.67 0.85 0.119   6 2.7e+00 8.5e-01     1.8 0.89 0.000 -31.6 -12.5     1.4
## 5 0.67 0.82 0.209   1 2.0e-01 6.6e-01     1.6 0.90 0.000  -5.5  -2.3     1.6
## 6 0.64 0.79 0.367  -3 1.1e-02      NA     1.6 0.90    NA    NA    NA     1.7
## 7 0.63 0.79 0.500  -6 4.3e-11      NA     1.3 0.92    NA    NA    NA     1.8
## 8 0.46 0.73 1.000  -8 0.0e+00      NA     1.3 0.92    NA    NA    NA     1.9
##    eChisq    SRMR eCRMS  eBIC
## 1 5.7e+02 1.6e-01 0.186 411.1
## 2 1.6e+02 8.5e-02 0.117  49.4
## 3 1.5e+01 2.6e-02 0.045 -53.6
## 4 1.5e+00 8.2e-03 0.020 -32.8
## 5 1.0e-01 2.2e-03 0.013  -5.6
## 6 9.4e-03 6.6e-04    NA    NA
## 7 3.8e-11 4.2e-08    NA    NA
## 8 5.0e-14 1.5e-09    NA    NA
```

```
nfactors(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate = "vari
```

```
##
## Number of factors
```

Figure 12.5: Very simple structure plot with orthogonal rotation in exploratory factor analysis

Figure 12.6: Very simple structure plot with orthogonal rotation in exploratory factor analysis

```
## Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
##     n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
## VSS complexity 1 achieves a maximimum of 0.7  with  3  factors
## VSS complexity 2 achieves a maximimum of 0.85  with  4  factors
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
## Empirical BIC achieves a minimum of  -53.62  with  3  factors
## Sample Size adjusted BIC achieves a minimum of  -12.54  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.55 0.00 0.083  27 3.3e+02 1.3e-53     7.3 0.55 0.193 174.7 260.3     1.0
## 2 0.65 0.76 0.066  19 1.4e+02 6.0e-21     3.9 0.76 0.147  34.1  94.3     1.2
## 3 0.70 0.84 0.072  12 3.5e+01 5.3e-04     2.2 0.86 0.079 -33.8   4.2     1.3
## 4 0.67 0.85 0.119   6 2.7e+00 8.5e-01     1.8 0.89 0.000 -31.6 -12.5     1.4
## 5 0.67 0.82 0.209   1 2.0e-01 6.6e-01     1.6 0.90 0.000  -5.5  -2.3     1.6
## 6 0.64 0.79 0.367  -3 1.1e-02      NA     1.6 0.90    NA    NA    NA     1.7
## 7 0.63 0.79 0.500  -6 4.3e-11      NA     1.3 0.92    NA    NA    NA     1.8
## 8 0.46 0.73 1.000  -8 0.0e+00      NA     1.3 0.92    NA    NA    NA     1.9
## 9 0.67 0.81    NA  -9 4.5e+01      NA     2.5 0.84    NA    NA    NA     1.3
##    eChisq    SRMR eCRMS  eBIC
## 1 5.7e+02 1.6e-01 0.186 411.1
## 2 1.6e+02 8.5e-02 0.117  49.4
## 3 1.5e+01 2.6e-02 0.045 -53.6
## 4 1.5e+00 8.2e-03 0.020 -32.8
## 5 1.0e-01 2.2e-03 0.013  -5.6
## 6 9.4e-03 6.6e-04    NA    NA
## 7 3.8e-11 4.2e-08    NA    NA
## 8 5.0e-14 1.5e-09    NA    NA
## 9 2.2e+01 3.2e-02    NA    NA
```

```
vss(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate =
```

12.4.1.2.2 Oblique (Oblimin) rotation

```
##
## Very Simple Structure
## Call: vss(x = HolzingerSwineford1939[, c("x1", "x2", "x3", "x4", "x5",
##     "x6", "x7", "x8", "x9")], rotate = "oblimin", fm = "ml")
## VSS complexity 1 achieves a maximimum of 0.69  with  3  factors
## VSS complexity 2 achieves a maximimum of 0.78  with  3  factors
##
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
```

Figure 12.7: Very simple structure plot with oblique rotation in exploratory factor analysis

```
## BIC achieves a minimum of  -33.84  with  3  factors
## Sample Size adjusted BIC achieves a minimum of  -12.54  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.55 0.00 0.083  27 3.3e+02 1.3e-53     7.3 0.55 0.193 174.7 260.3     1.0
## 2 0.65 0.71 0.066  19 1.4e+02 6.0e-21     4.7 0.71 0.147  34.1  94.3     1.1
## 3 0.69 0.78 0.072  12 3.5e+01 5.3e-04     3.3 0.80 0.079 -33.8   4.2     1.2
## 4 0.67 0.76 0.119   6 2.7e+00 8.5e-01     3.2 0.80 0.000 -31.6 -12.5     1.3
## 5 0.46 0.60 0.209   1 2.0e-01 6.6e-01     5.0 0.69 0.000  -5.5  -2.3     1.9
## 6 0.45 0.53 0.367  -3 1.1e-02      NA     6.8 0.58    NA    NA    NA     1.7
## 7 0.45 0.50 0.500  -6 4.4e-11      NA     7.4 0.54    NA    NA    NA     1.6
## 8 0.41 0.48 1.000  -8 0.0e+00      NA     7.5 0.54    NA    NA    NA     1.6
##    eChisq    SRMR eCRMS  eBIC
## 1 5.7e+02 1.6e-01 0.186 411.1
## 2 1.6e+02 8.5e-02 0.117  49.4
## 3 1.5e+01 2.6e-02 0.045 -53.6
## 4 1.5e+00 8.2e-03 0.020 -32.8
## 5 1.0e-01 2.2e-03 0.013  -5.6
## 6 9.4e-03 6.6e-04    NA    NA
## 7 3.8e-11 4.2e-08    NA    NA
## 8 5.0e-14 1.5e-09    NA    NA
```

```
nfactors(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rota
```

```
##
## Number of factors
## Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
##     n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
## VSS complexity 1 achieves a maximimum of 0.69  with  3  factors
## VSS complexity 2 achieves a maximimum of 0.78  with  3  factors
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
## Empirical BIC achieves a minimum of  -53.62  with  3  factors
## Sample Size adjusted BIC achieves a minimum of  -12.54  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.55 0.00 0.083  27 3.3e+02 1.3e-53     7.3 0.55 0.193 174.7 260.3     1.0
## 2 0.65 0.71 0.066  19 1.4e+02 6.0e-21     4.7 0.71 0.147  34.1  94.3     1.1
## 3 0.69 0.78 0.072  12 3.5e+01 5.3e-04     3.3 0.80 0.079 -33.8   4.2     1.2
## 4 0.67 0.76 0.119   6 2.7e+00 8.5e-01     3.2 0.80 0.000 -31.6 -12.5     1.3
## 5 0.46 0.60 0.209   1 2.0e-01 6.6e-01     5.0 0.69 0.000  -5.5  -2.3     1.9
## 6 0.45 0.53 0.367  -3 1.1e-02      NA     6.8 0.58    NA    NA    NA     1.7
## 7 0.45 0.50 0.500  -6 4.4e-11      NA     7.4 0.54    NA    NA    NA     1.6
## 8 0.41 0.48 1.000  -8 0.0e+00      NA     7.5 0.54    NA    NA    NA     1.6
```

Figure 12.8: Very simple structure plot with oblique rotation in exploratory factor analysis

```
## 9 0.66 0.75    NA  -9 4.5e+01      NA     3.9 0.76    NA    NA    NA     1.2
##    eChisq    SRMR eCRMS  eBIC
## 1 5.7e+02 1.6e-01 0.186 411.1
## 2 1.6e+02 8.5e-02 0.117  49.4
## 3 1.5e+01 2.6e-02 0.045 -53.6
## 4 1.5e+00 8.2e-03 0.020 -32.8
## 5 1.0e-01 2.2e-03 0.013  -5.6
## 6 9.4e-03 6.6e-04    NA    NA
## 7 3.8e-11 4.2e-08    NA    NA
## 8 5.0e-14 1.5e-09    NA    NA
## 9 2.2e+01 3.2e-02    NA    NA
```

```
vss(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate =
```

Figure 12.9: Very simple structure plot with no rotation in exploratory factor analysis

12.4.1.2.3 No rotation

```
##
```

```
## Very Simple Structure
## Call: vss(x = HolzingerSwineford1939[, c("x1", "x2", "x3", "x4", "x5",
##     "x6", "x7", "x8", "x9")], rotate = "none", fm = "ml")
## Although the VSS complexity 1 shows  7  factors, it is probably more reasonable to think about
## VSS complexity 2 achieves a maximimum of 0.81  with  4  factors
##
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
## BIC achieves a minimum of  -33.84  with  3  factors
## Sample Size adjusted BIC achieves a minimum of  -12.54  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.55 0.00 0.083  27 3.3e+02 1.3e-53     7.3 0.55 0.193 174.7 260.3     1.0
## 2 0.64 0.76 0.066  19 1.4e+02 6.0e-21     3.9 0.76 0.147  34.1  94.3     1.4
## 3 0.64 0.78 0.072  12 3.5e+01 5.3e-04     2.2 0.86 0.079 -33.8   4.2     1.8
## 4 0.64 0.81 0.119   6 2.7e+00 8.5e-01     1.8 0.89 0.000 -31.6 -12.5     1.7
## 5 0.55 0.77 0.209   1 2.0e-01 6.6e-01     1.6 0.90 0.000  -5.5  -2.3     2.3
## 6 0.64 0.80 0.367  -3 1.1e-02      NA     1.6 0.90    NA    NA    NA     2.1
## 7 0.70 0.80 0.500  -6 4.4e-11      NA     1.3 0.92    NA    NA    NA     2.2
## 8 0.67 0.78 1.000  -8 0.0e+00      NA     1.3 0.92    NA    NA    NA     2.2
##    eChisq    SRMR eCRMS  eBIC
## 1 5.7e+02 1.6e-01 0.186 411.1
## 2 1.6e+02 8.5e-02 0.117  49.4
## 3 1.5e+01 2.6e-02 0.045 -53.6
## 4 1.5e+00 8.2e-03 0.020 -32.8
## 5 1.0e-01 2.2e-03 0.013  -5.6
## 6 9.4e-03 6.6e-04    NA    NA
## 7 3.8e-11 4.2e-08    NA    NA
## 8 5.0e-14 1.5e-09    NA    NA
```

```
nfactors(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate = "none
```

```
##
## Number of factors
## Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
##     n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
## VSS complexity 1 achieves a maximimum of Although the vss.max shows  7  factors, it is probabl
## VSS complexity 2 achieves a maximimum of 0.81  with  4  factors
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
## Empirical BIC achieves a minimum of  -53.62  with  3  factors
## Sample Size adjusted BIC achieves a minimum of  -12.54  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.55 0.00 0.083  27 3.3e+02 1.3e-53     7.3 0.55 0.193 174.7 260.3     1.0
```

Figure 12.10: Very simple structure plot with no rotation in exploratory factor analysis

```
## 2 0.64 0.76 0.066  19 1.4e+02 6.0e-21     3.9 0.76 0.147  34.1  94.3     1.4
## 3 0.64 0.78 0.072  12 3.5e+01 5.3e-04     2.2 0.86 0.079 -33.8   4.2     1.8
## 4 0.64 0.81 0.119   6 2.7e+00 8.5e-01     1.8 0.89 0.000 -31.6 -12.5     1.7
## 5 0.55 0.77 0.209   1 2.0e-01 6.6e-01     1.6 0.90 0.000  -5.5  -2.3     2.3
## 6 0.64 0.80 0.367  -3 1.1e-02      NA     1.6 0.90    NA    NA    NA     2.1
## 7 0.70 0.80 0.500  -6 4.4e-11      NA     1.3 0.92    NA    NA    NA     2.2
## 8 0.67 0.78 1.000  -8 0.0e+00      NA     1.3 0.92    NA    NA    NA     2.2
## 9 0.63 0.75    NA  -9 4.5e+01      NA     2.5 0.84    NA    NA    NA     1.8
##    eChisq    SRMR eCRMS  eBIC
## 1 5.7e+02 1.6e-01 0.186 411.1
## 2 1.6e+02 8.5e-02 0.117  49.4
## 3 1.5e+01 2.6e-02 0.045 -53.6
## 4 1.5e+00 8.2e-03 0.020 -32.8
## 5 1.0e-01 2.2e-03 0.013  -5.6
## 6 9.4e-03 6.6e-04    NA    NA
## 7 3.8e-11 4.2e-08    NA    NA
## 8 5.0e-14 1.5e-09    NA    NA
## 9 2.2e+01 3.2e-02    NA    NA
```

12.4.2 Run factor analysis

Exploratory factor analysis (EFA) models were fit using the `psych` package [Revelle, 2020].

12.4.2.1 Orthogonal (Varimax) rotation

Fit a different model with each number of possible factors:

```
efa1factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa2factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa3factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa4factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa5factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa6factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa7factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa8factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
efa9factorOrthogonal <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
```

12.4.2.2 Oblique (Oblimin) rotation

Fit a different model with each number of possible factors:

```
efa1factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa2factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa3factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa4factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa5factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa6factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa7factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa8factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
efa9factorOblique <- fa(r = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x
```

12.4.3 Factor scores

12.4.3.1 Orthogonal (Varimax) rotation

```
fa3Orthogonal <- efa3factorOrthogonal$scores
```

12.4.3.2 Oblique (Oblimin) rotation

```
fa3Oblique <- efa3factorOblique$scores
```

12.4.4 Plots

Pairs panel plots were generated using the `psych` package [Revelle, 2020]. Correlation plots were generated using the `corrplot` package [Wei and Simko, 2017].

12.4.4.1 Orthogonal (Varimax) rotation

```
factor.plot(efa3factorOrthogonal, cut = 0.5)
```

```
fa.diagram(efa3factorOrthogonal, digits = 2)
```

```
pairs.panels(fa3Orthogonal)
```

Figure 12.11: Factor plot with orthogonal rotation in exploratory factor analysis

```
corrplot(cor(fa3Orthogonal, use = "pairwise.complete.obs"))
```

12.4.4.2 Oblique (Oblimin) rotation

```
factor.plot(efa3factorOblique, cut = 0.5)
```

```
fa.diagram(efa3factorOblique, digits = 2)
```

```
pairs.panels(fa3Oblique)
```

```
corrplot(cor(fa3Oblique, use = "pairwise.complete.obs"))
```

Figure 12.12: Factor diagram with orthogonal rotation in exploratory factor analysis

Figure 12.13: Pairs panel plot with orthogonal rotation in exploratory factor analysis

Figure 12.14: Correlation plot with orthogonal rotation in exploratory factor analysis

Figure 12.15: Factor plot with oblique rotation in exploratory factor analysis

12.5 Confirmatory Factor Analysis (CFA)

I introduced confirmatory factor analysis (CFA) models in Section 5.3.3 in the chapter on structural equation models.

The CFA models were fit in the `lavaan` package [Rosseel et al., 2020]. The examples were adapted from the `lavaan` documentation: http://lavaan.ugent.be/tutorial/cfa.html

12.5.1 Specify the model

```
cfaModel_syntax <- '
 #Factor loadings
 visual  =~ x1 + x2 + x3
 textual =~ x4 + x5 + x6
 speed   =~ x7 + x8 + x9
'

cfaModel_fullSyntax <- '
```

Figure 12.16: Factor diagram with oblique rotation in exploratory factor analysis

Figure 12.17: Pairs panel plot with oblique rotation in exploratory factor analysis

Figure 12.18: Correlation plot with oblique rotation in exploratory factor analysis

```
 #Factor loadings (free the factor loading of the first indicator)
 visual  =~ NA*x1 + x2 + x3
 textual =~ NA*x4 + x5 + x6
 speed   =~ NA*x7 + x8 + x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one
 visual ~~ 1*visual
 textual ~~ 1*textual
 speed ~~ 1*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Free intercepts of manifest variables
 x1 ~ int1*1
 x2 ~ int2*1
 x3 ~ int3*1
 x4 ~ int4*1
 x5 ~ int5*1
 x6 ~ int6*1
 x7 ~ int7*1
 x8 ~ int8*1
 x9 ~ int9*1
'
```

12.5.1.1 Model syntax in table form:

```
lavaanify(cfaModel_syntax)
```

```
##    id     lhs op     rhs user block group free ustart exo label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0         .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0         .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0         .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0         .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0         .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0         .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0         .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0         .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0         .p9.
## 10 10      x1 ~~      x1    0     1     1    0      0   0        .p10.
## 11 11      x2 ~~      x2    0     1     1    0      0   0        .p11.
## 12 12      x3 ~~      x3    0     1     1    0      0   0        .p12.
## 13 13      x4 ~~      x4    0     1     1    0      0   0        .p13.
## 14 14      x5 ~~      x5    0     1     1    0      0   0        .p14.
## 15 15      x6 ~~      x6    0     1     1    0      0   0        .p15.
## 16 16      x7 ~~      x7    0     1     1    0      0   0        .p16.
## 17 17      x8 ~~      x8    0     1     1    0      0   0        .p17.
## 18 18      x9 ~~      x9    0     1     1    0      0   0        .p18.
## 19 19  visual ~~  visual    0     1     1    0      0   0        .p19.
## 20 20 textual ~~ textual    0     1     1    0      0   0        .p20.
## 21 21   speed ~~   speed    0     1     1    0      0   0        .p21.
```

```
lavaanify(cfaModel_fullSyntax)
```

```
##    id     lhs op     rhs user block group free ustart exo label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0         .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0         .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0         .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0         .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0         .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0         .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0         .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0         .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0         .p9.
## 10 10  visual ~1            1     1     1    0      0   0        .p10.
## 11 11 textual ~1            1     1     1    0      0   0        .p11.
## 12 12   speed ~1            1     1     1    0      0   0        .p12.
## 13 13  visual ~~  visual    1     1     1    0      1   0        .p13.
```

```
## 14 14 textual ~~ textual    1     1     1    0      1   0        .p14.
## 15 15   speed ~~   speed    1     1     1    0      1   0        .p15.
## 16 16  visual ~~ textual    1     1     1   10     NA   0        .p16.
## 17 17  visual ~~   speed    1     1     1   11     NA   0        .p17.
## 18 18 textual ~~   speed    1     1     1   12     NA   0        .p18.
## 19 19      x1 ~~      x1    1     1     1   13     NA   0        .p19.
## 20 20      x2 ~~      x2    1     1     1   14     NA   0        .p20.
## 21 21      x3 ~~      x3    1     1     1   15     NA   0        .p21.
## 22 22      x4 ~~      x4    1     1     1   16     NA   0        .p22.
## 23 23      x5 ~~      x5    1     1     1   17     NA   0        .p23.
## 24 24      x6 ~~      x6    1     1     1   18     NA   0        .p24.
## 25 25      x7 ~~      x7    1     1     1   19     NA   0        .p25.
## 26 26      x8 ~~      x8    1     1     1   20     NA   0        .p26.
## 27 27      x9 ~~      x9    1     1     1   21     NA   0        .p27.
## 28 28      x1 ~1            1     1     1   22     NA   0  int1  .p28.
## 29 29      x2 ~1            1     1     1   23     NA   0  int2  .p29.
## 30 30      x3 ~1            1     1     1   24     NA   0  int3  .p30.
## 31 31      x4 ~1            1     1     1   25     NA   0  int4  .p31.
## 32 32      x5 ~1            1     1     1   26     NA   0  int5  .p32.
## 33 33      x6 ~1            1     1     1   27     NA   0  int6  .p33.
## 34 34      x7 ~1            1     1     1   28     NA   0  int7  .p34.
## 35 35      x8 ~1            1     1     1   29     NA   0  int8  .p35.
## 36 36      x9 ~1            1     1     1   30     NA   0  int9  .p36.
```

12.5.2 Fit the model

```
cfaModelFit <- cfa(cfaModel_syntax,
                   data = HolzingerSwineford1939,
                   missing = "ML",
                   estimator = "MLR",
                   std.lv = TRUE)

cfaModelFit_full <- lavaan(cfaModel_fullSyntax,
                           data = HolzingerSwineford1939,
                           missing = "ML",
                           estimator = "MLR")
```

12.5.3 Display summary output

```
summary(cfaModelFit,
        fit.measures = TRUE,
```

```
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 40 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        30
##
##   Number of observations                           301
##   Number of missing patterns                        84
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                67.800      71.335
##   Degrees of freedom                                24          24
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.950
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               710.186     697.261
##   Degrees of freedom                                36          36
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.019
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.935       0.928
##   Tucker-Lewis Index (TLI)                       0.903       0.893
##
##   Robust Comparative Fit Index (CFI)                         0.933
##   Robust Tucker-Lewis Index (TLI)                            0.900
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3179.834   -3179.834
##   Scaling correction factor                                  1.082
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -3145.934   -3145.934
##   Scaling correction factor                                  1.023
##       for the MLR correction
##
##   Akaike (AIC)                                6419.669    6419.669
```

```
##   Bayesian (BIC)                              6530.882    6530.882
##   Sample-size adjusted Bayesian (BIC)         6435.739    6435.739
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.078       0.081
##   90 Percent confidence interval - lower         0.056       0.059
##   90 Percent confidence interval - upper         0.100       0.104
##   P-value RMSEA <= 0.05                          0.018       0.011
##
##   Robust RMSEA                                               0.079
##   90 Percent confidence interval - lower                     0.058
##   90 Percent confidence interval - upper                     0.100
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.061       0.061
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1                0.935    0.120    7.826    0.000    0.935    0.805
##     x2                0.428    0.104    4.130    0.000    0.428    0.364
##     x3                0.640    0.100    6.417    0.000    0.640    0.565
##   textual =~
##     x4                0.974    0.063   15.405    0.000    0.974    0.844
##     x5                1.152    0.065   17.781    0.000    1.152    0.877
##     x6                0.922    0.064   14.304    0.000    0.922    0.852
##   speed =~
##     x7                0.597    0.097    6.182    0.000    0.597    0.564
##     x8                0.709    0.108    6.566    0.000    0.709    0.713
##     x9                0.672    0.112    5.991    0.000    0.672    0.673
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.480    0.075    6.429    0.000    0.480    0.480
##     speed             0.466    0.131    3.561    0.000    0.466    0.466
##   textual ~~
##     speed             0.295    0.083    3.547    0.000    0.295    0.295
```

```
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                4.954    0.071   70.050    0.000    4.954    4.264
##    .x2                6.098    0.073   83.031    0.000    6.098    5.181
##    .x3                2.278    0.070   32.768    0.000    2.278    2.010
##    .x4                3.082    0.069   44.737    0.000    3.082    2.671
##    .x5                4.290    0.078   54.751    0.000    4.290    3.267
##    .x6                2.172    0.065   33.339    0.000    2.172    2.007
##    .x7                4.165    0.065   63.803    0.000    4.165    3.934
##    .x8                5.512    0.061   90.143    0.000    5.512    5.538
##    .x9                5.368    0.061   87.886    0.000    5.368    5.371
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                0.475    0.206    2.307    0.021    0.475    0.352
##    .x2                1.202    0.131    9.185    0.000    1.202    0.868
##    .x3                0.874    0.119    7.379    0.000    0.874    0.681
##    .x4                0.383    0.053    7.197    0.000    0.383    0.287
##    .x5                0.399    0.070    5.702    0.000    0.399    0.231
##    .x6                0.321    0.044    7.227    0.000    0.321    0.275
##    .x7                0.764    0.108    7.066    0.000    0.764    0.682
##    .x8                0.487    0.132    3.704    0.000    0.487    0.492
##    .x9                0.547    0.135    4.052    0.000    0.547    0.547
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.648
##     x2                0.132
##     x3                0.319
##     x4                0.713
##     x5                0.769
##     x6                0.725
##     x7                0.318
##     x8                0.508
##     x9                0.453
```

```
summary(cfaModelFit_full,
        fit.measures = TRUE,
```

```
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 40 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        30
##
##   Number of observations                           301
##   Number of missing patterns                        84
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                 67.800      71.335
##   Degrees of freedom                                 24          24
##   P-value (Chi-square)                            0.000       0.000
##   Scaling correction factor                                   0.950
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                                710.186     697.261
##   Degrees of freedom                                 36          36
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   1.019
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.935       0.928
##   Tucker-Lewis Index (TLI)                        0.903       0.893
##
##   Robust Comparative Fit Index (CFI)                          0.933
##   Robust Tucker-Lewis Index (TLI)                             0.900
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3179.834   -3179.834
##   Scaling correction factor                                   1.082
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -3145.934   -3145.934
##   Scaling correction factor                                   1.023
##       for the MLR correction
##
##   Akaike (AIC)                                6419.669    6419.669
```

```
##   Bayesian (BIC)                              6530.882    6530.882
##   Sample-size adjusted Bayesian (BIC)         6435.739    6435.739
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.078       0.081
##   90 Percent confidence interval - lower         0.056       0.059
##   90 Percent confidence interval - upper         0.100       0.104
##   P-value RMSEA <= 0.05                          0.018       0.011
##
##   Robust RMSEA                                               0.079
##   90 Percent confidence interval - lower                     0.058
##   90 Percent confidence interval - upper                     0.100
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.061       0.061
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1                0.935    0.120    7.826    0.000    0.935    0.805
##     x2                0.428    0.104    4.130    0.000    0.428    0.364
##     x3                0.640    0.100    6.417    0.000    0.640    0.565
##   textual =~
##     x4                0.974    0.063   15.405    0.000    0.974    0.844
##     x5                1.152    0.065   17.781    0.000    1.152    0.877
##     x6                0.922    0.064   14.304    0.000    0.922    0.852
##   speed =~
##     x7                0.597    0.097    6.182    0.000    0.597    0.564
##     x8                0.709    0.108    6.566    0.000    0.709    0.713
##     x9                0.672    0.112    5.991    0.000    0.672    0.673
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.480    0.075    6.429    0.000    0.480    0.480
##     speed             0.466    0.131    3.561    0.000    0.466    0.466
##   textual ~~
##     speed             0.295    0.083    3.547    0.000    0.295    0.295
```

```
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1      (int1)    4.954    0.071   70.050    0.000    4.954    4.264
##    .x2      (int2)    6.098    0.073   83.031    0.000    6.098    5.181
##    .x3      (int3)    2.278    0.070   32.768    0.000    2.278    2.010
##    .x4      (int4)    3.082    0.069   44.737    0.000    3.082    2.671
##    .x5      (int5)    4.290    0.078   54.751    0.000    4.290    3.267
##    .x6      (int6)    2.172    0.065   33.339    0.000    2.172    2.007
##    .x7      (int7)    4.165    0.065   63.803    0.000    4.165    3.934
##    .x8      (int8)    5.512    0.061   90.143    0.000    5.512    5.538
##    .x9      (int9)    5.368    0.061   87.886    0.000    5.368    5.371
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1                0.475    0.206    2.307    0.021    0.475    0.352
##    .x2                1.202    0.131    9.185    0.000    1.202    0.868
##    .x3                0.874    0.119    7.379    0.000    0.874    0.681
##    .x4                0.383    0.053    7.197    0.000    0.383    0.287
##    .x5                0.399    0.070    5.702    0.000    0.399    0.231
##    .x6                0.321    0.044    7.227    0.000    0.321    0.275
##    .x7                0.764    0.108    7.066    0.000    0.764    0.682
##    .x8                0.487    0.132    3.704    0.000    0.487    0.492
##    .x9                0.547    0.135    4.052    0.000    0.547    0.547
##
## R-Square:
##                    Estimate
##     x1                0.648
##     x2                0.132
##     x3                0.319
##     x4                0.713
##     x5                0.769
##     x6                0.725
##     x7                0.318
##     x8                0.508
##     x9                0.453
```

12.5.4 Estimates of model fit

```
fitMeasures(cfaModelFit, fit.measures = c("chisq", "df", "pvalue",
                                          "chisq.scaled", "df.scaled", "pvalue.scaled"
                                          "chisq.scaling.factor",
                                          "rmsea", "cfi", "tli", "srmr",
                                          "rmsea.robust", "cfi.robust", "tli.robust"))
```

```
##                chisq                   df               pvalue
##               67.800               24.000                0.000
##         chisq.scaled            df.scaled        pvalue.scaled
##               71.335               24.000                0.000
## chisq.scaling.factor                rmsea                  cfi
##                0.950                0.078                0.935
##                  tli                 srmr         rmsea.robust
##                0.903                0.061                0.079
##           cfi.robust           tli.robust
##                0.933                0.900
```

12.5.5 Residuals

```
residuals(cfaModelFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##    x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.037  0.000
## x3  0.004  0.137  0.000
## x4  0.060  0.026 -0.070  0.000
## x5  0.003 -0.046 -0.186  0.005  0.000
## x6  0.061  0.029 -0.046 -0.013  0.001  0.000
## x7 -0.132 -0.175 -0.035  0.014 -0.037 -0.032  0.000
## x8 -0.051 -0.069 -0.008 -0.077 -0.035 -0.007  0.085  0.000
## x9  0.109  0.073  0.160  0.033  0.038  0.062 -0.037 -0.027  0.000
##
## $mean
##     x1     x2     x3     x4     x5     x6     x7     x8     x9
## -0.005 -0.002 -0.003  0.001  0.007 -0.003  0.004  0.006  0.007
```

12.5.6 Modification indices

```
modificationindices(cfaModelFit)
```

```
##        lhs op rhs     mi    epc sepc.lv sepc.all sepc.nox
## 37  visual =~  x4  1.074  0.075   0.075    0.065    0.065
## 38  visual =~  x5  5.819 -0.199  -0.199   -0.152   -0.152
## 39  visual =~  x6  2.060  0.096   0.096    0.089    0.089
## 40  visual =~  x7 11.420 -0.329  -0.329   -0.311   -0.311
## 41  visual =~  x8  4.212 -0.212  -0.212   -0.213   -0.213
## 42  visual =~  x9 26.377  0.509   0.509    0.509    0.509
## 43 textual =~  x1 13.064  0.561   0.561    0.483    0.483
## 44 textual =~  x2  0.006 -0.008  -0.008   -0.007   -0.007
## 45 textual =~  x3 13.168 -0.393  -0.393   -0.347   -0.347
## 46 textual =~  x7  0.326 -0.042  -0.042   -0.039   -0.039
## 47 textual =~  x8  1.348 -0.085  -0.085   -0.086   -0.086
## 48 textual =~  x9  2.846  0.120   0.120    0.120    0.120
## 49   speed =~  x1  0.660 -0.130  -0.130   -0.112   -0.112
## 50   speed =~  x2  1.049 -0.115  -0.115   -0.098   -0.098
## 51   speed =~  x3  2.438  0.180   0.180    0.159    0.159
## 52   speed =~  x4  0.127 -0.023  -0.023   -0.020   -0.020
## 53   speed =~  x5  0.390 -0.045  -0.045   -0.034   -0.034
## 54   speed =~  x6  0.973  0.058   0.058    0.053    0.053
## 55      x1 ~~  x2  4.414 -0.232  -0.232   -0.307   -0.307
## 56      x1 ~~  x3  0.530 -0.137  -0.137   -0.212   -0.212
## 57      x1 ~~  x4  0.961  0.049   0.049    0.115    0.115
## 58      x1 ~~  x5  0.191  0.025   0.025    0.057    0.057
## 59      x1 ~~  x6  0.023  0.007   0.007    0.018    0.018
## 60      x1 ~~  x7  5.726 -0.155  -0.155   -0.258   -0.258
## 61      x1 ~~  x8  0.412 -0.040  -0.040   -0.083   -0.083
## 62      x1 ~~  x9  3.341  0.112   0.112    0.220    0.220
## 63      x2 ~~  x3  9.541  0.263   0.263    0.256    0.256
## 64      x2 ~~  x4  0.415  0.037   0.037    0.054    0.054
## 65      x2 ~~  x5  1.806 -0.085  -0.085   -0.123   -0.123
## 66      x2 ~~  x6  0.575  0.040   0.040    0.064    0.064
## 67      x2 ~~  x7  5.964 -0.178  -0.178   -0.186   -0.186
## 68      x2 ~~  x8  0.606 -0.050  -0.050   -0.066   -0.066
## 69      x2 ~~  x9  2.729  0.109   0.109    0.135    0.135
## 70      x3 ~~  x4  0.017  0.007   0.007    0.012    0.012
## 71      x3 ~~  x5 11.654 -0.196  -0.196   -0.331   -0.331
## 72      x3 ~~  x6  1.104  0.050   0.050    0.094    0.094
## 73      x3 ~~  x7  0.001  0.002   0.002    0.002    0.002
## 74      x3 ~~  x8  0.365 -0.036  -0.036   -0.055   -0.055
## 75      x3 ~~  x9  6.341  0.150   0.150    0.217    0.217
```

```
## 76      x4 ~~  x5  2.160  0.184   0.184    0.472    0.472
## 77      x4 ~~  x6  5.080 -0.222  -0.222   -0.634   -0.634
## 78      x4 ~~  x7  2.824  0.079   0.079    0.147    0.147
## 79      x4 ~~  x8  3.451 -0.079  -0.079   -0.184   -0.184
## 80      x4 ~~  x9  0.122 -0.015  -0.015   -0.033   -0.033
## 81      x5 ~~  x6  0.702  0.101   0.101    0.283    0.283
## 82      x5 ~~  x7  0.089 -0.016  -0.016   -0.029   -0.029
## 83      x5 ~~  x8  0.452  0.032   0.032    0.073    0.073
## 84      x5 ~~  x9  0.020 -0.007  -0.007   -0.015   -0.015
## 85      x6 ~~  x7  0.766 -0.038  -0.038   -0.077   -0.077
## 86      x6 ~~  x8  0.506  0.028   0.028    0.070    0.070
## 87      x6 ~~  x9  0.292  0.021   0.021    0.051    0.051
## 88      x7 ~~  x8 23.762  0.484   0.484    0.793    0.793
## 89      x7 ~~  x9  4.332 -0.193  -0.193   -0.299   -0.299
## 90      x8 ~~  x9  9.281 -0.375  -0.375   -0.726   -0.726
```

12.5.7 Factor scores

```
cfaFactorScores <- predict(cfaModelFit)
as_tibble(cfaFactorScores)
```

```
## # A tibble: 301 x 3
##    visual      textual     speed
##    <lvn.mtrx>  <lvn.mtrx>  <lvn.mtrx>
##  1 -0.86041164 -0.76758274  0.02244972
##  2  0.08526570 -0.99336413  1.06163579
##  3 -0.80315251 -1.71609058 -1.34554816
##  4  0.44536524  0.05121926 -0.34863763
##  5 -0.40449664 -0.09646545  0.33845087
##  6  0.02209274 -1.38926309  1.01154133
##  7 -1.18113628  0.68973492  0.06454226
##  8  0.15562492 -0.12097395 -0.87263727
##  9 -0.43774316  0.03022805 -0.43416919
## 10 -1.17665390  0.13889284 -0.90973870
## # ... with 291 more rows
```

12.5.7.1 Compare CFA factor scores to EFA factor scores

```
cor.test(x = cfaFactorScores[,"visual"], y = fa3Orthogonal[,"ML3"])
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  cfaFactorScores[, "visual"] and fa3Orthogonal[, "ML3"]
## t = 10.741, df = 73, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6756601 0.8572963
## sample estimates:
##       cor
## 0.7826095
```

```
cor.test(x = cfaFactorScores[,"textual"], y = fa3Orthogonal[,"ML1"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  cfaFactorScores[, "textual"] and fa3Orthogonal[, "ML1"]
## t = 46.782, df = 73, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9742968 0.9897173
## sample estimates:
##       cor
## 0.9837285
```

```
cor.test(x = cfaFactorScores[,"speed"], y = fa3Orthogonal[,"ML2"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  cfaFactorScores[, "speed"] and fa3Orthogonal[, "ML2"]
## t = 25.743, df = 73, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9203899 0.9676213
## sample estimates:
##       cor
## 0.9490926
```

12.5.8 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModelFit)
```

```
##           visual   textual     speed
## alpha  0.6187715 0.8890587 0.6902590
## omega  0.6114187 0.8938830 0.6853969
## omega2 0.6114187 0.8938830 0.6853969
## omega3 0.5862713 0.9001357 0.6801936
## avevar 0.3652270 0.7391895 0.4219080
```

12.5.9 Path diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModelFit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.3)
```

12.5.10 Class Examples

```
standardDeviations <- rep(1, 6)
sampleSize <- 300
```

12.5.10.1 Model 1

```
correlationMatrixModel1 <- matrix(.6, nrow = 6, ncol = 6)
diag(correlationMatrixModel1) <- 1
rownames(correlationMatrixModel1) <- colnames(correlationMatrixModel1) <- paste("V", 1

covarianceMatrixModel1 <- psych::cor2cov(correlationMatrixModel1, sigma = standardDevia
covarianceMatrixModel1
```

12.5.10.1.1 Create the covariance matrix

Figure 12.19: Confirmatory factor analysis model diagram

```
##     V1  V2  V3  V4  V5  V6
## V1 1.0 0.6 0.6 0.6 0.6 0.6
## V2 0.6 1.0 0.6 0.6 0.6 0.6
## V3 0.6 0.6 1.0 0.6 0.6 0.6
## V4 0.6 0.6 0.6 1.0 0.6 0.6
## V5 0.6 0.6 0.6 0.6 1.0 0.6
## V6 0.6 0.6 0.6 0.6 0.6 1.0
```

```
cfaModel1 <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3 + V4 + V5 + V6
'
```

12.5.10.1.2 Specify the model

```
lavaanify(cfaModel1)
```

12.5.10.1.3 Model syntax in table form:

```
##    id lhs op rhs user block group free ustart exo label plabel
## 1   1  f1 =~  V1    1     1     1    1     NA   0         .p1.
## 2   2  f1 =~  V2    1     1     1    2     NA   0         .p2.
## 3   3  f1 =~  V3    1     1     1    3     NA   0         .p3.
## 4   4  f1 =~  V4    1     1     1    4     NA   0         .p4.
## 5   5  f1 =~  V5    1     1     1    5     NA   0         .p5.
## 6   6  f1 =~  V6    1     1     1    6     NA   0         .p6.
## 7   7  V1 ~~  V1    0     1     1    0      0   0         .p7.
## 8   8  V2 ~~  V2    0     1     1    0      0   0         .p8.
## 9   9  V3 ~~  V3    0     1     1    0      0   0         .p9.
## 10 10  V4 ~~  V4    0     1     1    0      0   0        .p10.
## 11 11  V5 ~~  V5    0     1     1    0      0   0        .p11.
## 12 12  V6 ~~  V6    0     1     1    0      0   0        .p12.
## 13 13  f1 ~~  f1    0     1     1    0      0   0        .p13.
```

```
cfaModel1Fit <- cfa(cfaModel1,
                    sample.cov = covarianceMatrixModel1,
```

```
                    sample.nobs = sampleSize,
                    estimator = "ML",
                    std.lv = TRUE)
```

12.5.10.1.4 Fit the model

```
summary(cfaModel1Fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

12.5.10.1.5 Display summary output

```
## lavaan 0.6-8 ended normally after 10 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        12
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           1.000
##
## Model Test Baseline Model:
##
##   Test statistic                               958.548
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.016
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2071.810
##   Loglikelihood unrestricted model (H1)      -2071.810
```

```
##
##   Akaike (AIC)                                4167.621
##   Bayesian (BIC)                              4212.066
##   Sample-size adjusted Bayesian (BIC)         4174.009
##
## 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 RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## 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
##   f1 =~
##     V1                0.773    0.050   15.390    0.000    0.773    0.775
##     V2                0.773    0.050   15.390    0.000    0.773    0.775
##     V3                0.773    0.050   15.390    0.000    0.773    0.775
##     V4                0.773    0.050   15.390    0.000    0.773    0.775
##     V5                0.773    0.050   15.390    0.000    0.773    0.775
##     V6                0.773    0.050   15.390    0.000    0.773    0.775
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.399    0.039   10.171    0.000    0.399    0.400
##    .V2                0.399    0.039   10.171    0.000    0.399    0.400
##    .V3                0.399    0.039   10.171    0.000    0.399    0.400
##    .V4                0.399    0.039   10.171    0.000    0.399    0.400
##    .V5                0.399    0.039   10.171    0.000    0.399    0.400
##    .V6                0.399    0.039   10.171    0.000    0.399    0.400
##     f1                1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     V1                0.600
##     V2                0.600
```

```
##     V3                 0.600
##     V4                 0.600
##     V5                 0.600
##     V6                 0.600
```

```
fitMeasures(cfaModel1Fit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "tli", "srmr"
```

12.5.10.1.6 Estimates of model fit

```
##  chisq     df pvalue  rmsea    cfi    tli   srmr
##  0.000  9.000  1.000  0.000  1.000  1.016  0.000
```

```
residuals(cfaModel1Fit, type = "cor")
```

12.5.10.1.7 Residuals

```
## $type
## [1] "cor.bollen"
##
## $cov
##    V1 V2 V3 V4 V5 V6
## V1 0
## V2 0  0
## V3 0  0  0
## V4 0  0  0  0
## V5 0  0  0  0  0
## V6 0  0  0  0  0  0
```

```
modificationindices(cfaModel1Fit)
```

12.5.10.1.8 Modification indices

```
##    lhs op rhs mi epc sepc.lv sepc.all sepc.nox
## 14  V1 ~~  V2  0   0       0        0        0
## 15  V1 ~~  V3  0   0       0        0        0
## 16  V1 ~~  V4  0   0       0        0        0
```

```
## 17  V1 ~~  V5  0   0       0        0        0
## 18  V1 ~~  V6  0   0       0        0        0
## 19  V2 ~~  V3  0   0       0        0        0
## 20  V2 ~~  V4  0   0       0        0        0
## 21  V2 ~~  V5  0   0       0        0        0
## 22  V2 ~~  V6  0   0       0        0        0
## 23  V3 ~~  V4  0   0       0        0        0
## 24  V3 ~~  V5  0   0       0        0        0
## 25  V3 ~~  V6  0   0       0        0        0
## 26  V4 ~~  V5  0   0       0        0        0
## 27  V4 ~~  V6  0   0       0        0        0
## 28  V5 ~~  V6  0   0       0        0        0
```

12.5.10.1.9 Reliability Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModel1Fit)
```

```
##         f1
## alpha  0.9
## omega  0.9
## omega2 0.9
## omega3 0.9
## avevar 0.6
```

12.5.10.1.10 Path diagram Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModel1Fit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.5)
```

12.5.10.2 Model 2

```
correlationMatrixModel2 <- matrix(0, nrow = 6, ncol = 6)
diag(correlationMatrixModel2) <- 1
```

Figure 12.20: Confirmatory factor analysis model 1 diagram

```
rownames(correlationMatrixModel2) <- colnames(correlationMatrixModel2) <- paste("V", 1

covarianceMatrixModel2 <- psych::cor2cov(correlationMatrixModel2, sigma = standardDevia
covarianceMatrixModel2
```

12.5.10.2.1 Create the covariance matrix

```
##    V1 V2 V3 V4 V5 V6
## V1  1  0  0  0  0  0
## V2  0  1  0  0  0  0
## V3  0  0  1  0  0  0
## V4  0  0  0  1  0  0
## V5  0  0  0  0  1  0
## V6  0  0  0  0  0  1
```

```
cfaModel2 <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3 + V4 + V5 + V6
'
```

12.5.10.2.2 Specify the model

```
lavaanify(cfaModel2)
```

12.5.10.2.3 Model syntax in table form:

```
##    id lhs op rhs user block group free ustart exo label plabel
## 1   1  f1 =~  V1    1     1     1    1     NA   0         .p1.
## 2   2  f1 =~  V2    1     1     1    2     NA   0         .p2.
## 3   3  f1 =~  V3    1     1     1    3     NA   0         .p3.
## 4   4  f1 =~  V4    1     1     1    4     NA   0         .p4.
## 5   5  f1 =~  V5    1     1     1    5     NA   0         .p5.
## 6   6  f1 =~  V6    1     1     1    6     NA   0         .p6.
## 7   7  V1 ~~  V1    0     1     1    0      0   0         .p7.
## 8   8  V2 ~~  V2    0     1     1    0      0   0         .p8.
## 9   9  V3 ~~  V3    0     1     1    0      0   0         .p9.
## 10 10  V4 ~~  V4    0     1     1    0      0   0        .p10.
## 11 11  V5 ~~  V5    0     1     1    0      0   0        .p11.
## 12 12  V6 ~~  V6    0     1     1    0      0   0        .p12.
## 13 13  f1 ~~  f1    0     1     1    0      0   0        .p13.
```

```
cfaModel2Fit <- cfa(cfaModel2,
                    sample.cov = covarianceMatrixModel2,
                    sample.nobs = sampleSize,
                    estimator = "ML",
                    std.lv = TRUE)
```

12.5.10.2.4 Fit the model

```
summary(cfaModel2Fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

12.5.10.2.5 Display summary output

```
## lavaan 0.6-8 ended normally after 23 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        12
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           1.000
##
## Model Test Baseline Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                15
##   P-value                                        1.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       0.000
##
```

```
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2551.084
##   Loglikelihood unrestricted model (H1)      -2551.084
##
##   Akaike (AIC)                                5126.169
##   Bayesian (BIC)                              5170.614
##   Sample-size adjusted Bayesian (BIC)         5132.557
##
## 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 RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## 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
##   f1 =~
##     V1                0.000  227.294    0.000    1.000    0.000    0.000
##     V2                0.000  227.294    0.000    1.000    0.000    0.000
##     V3                0.000  227.294    0.000    1.000    0.000    0.000
##     V4                0.000  227.294    0.000    1.000    0.000    0.000
##     V5                0.000  227.294    0.000    1.000    0.000    0.000
##     V6                0.000  227.294    0.000    1.000    0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.997    0.098   10.171    0.000    0.997    1.000
##    .V2                0.997    0.098   10.171    0.000    0.997    1.000
##    .V3                0.997    0.098   10.171    0.000    0.997    1.000
##    .V4                0.997    0.098   10.171    0.000    0.997    1.000
##    .V5                0.997    0.098   10.171    0.000    0.997    1.000
##    .V6                0.997    0.098   10.171    0.000    0.997    1.000
##     f1                1.000                               1.000    1.000
##
```

```
## R-Square:
##                    Estimate
##     V1                0.000
##     V2                0.000
##     V3                0.000
##     V4                0.000
##     V5                0.000
##     V6                0.000
```

```
fitMeasures(cfaModel2Fit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "tli", "srmr"
```

12.5.10.2.6 Estimates of model fit

```
##  chisq     df pvalue  rmsea    cfi    tli   srmr
##      0      9      1      0      1      0      0
```

```
residuals(cfaModel2Fit, type = "cor")
```

12.5.10.2.7 Residuals

```
## $type
## [1] "cor.bollen"
##
## $cov
##    V1 V2 V3 V4 V5 V6
## V1 0
## V2 0  0
## V3 0  0  0
## V4 0  0  0  0
## V5 0  0  0  0  0
## V6 0  0  0  0  0  0
```

```
modificationindices(cfaModel1Fit)
```

12.5.10.2.8 Modification indices

```
##    lhs op rhs mi epc sepc.lv sepc.all sepc.nox
## 14  V1 ~~  V2  0   0       0        0        0
## 15  V1 ~~  V3  0   0       0        0        0
## 16  V1 ~~  V4  0   0       0        0        0
## 17  V1 ~~  V5  0   0       0        0        0
## 18  V1 ~~  V6  0   0       0        0        0
## 19  V2 ~~  V3  0   0       0        0        0
## 20  V2 ~~  V4  0   0       0        0        0
## 21  V2 ~~  V5  0   0       0        0        0
## 22  V2 ~~  V6  0   0       0        0        0
## 23  V3 ~~  V4  0   0       0        0        0
## 24  V3 ~~  V5  0   0       0        0        0
## 25  V3 ~~  V6  0   0       0        0        0
## 26  V4 ~~  V5  0   0       0        0        0
## 27  V4 ~~  V6  0   0       0        0        0
## 28  V5 ~~  V6  0   0       0        0        0
```

12.5.10.2.9 Reliability Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModel2Fit)
```

```
##                  f1
## alpha  0.000000e+00
## omega  8.681366e-08
## omega2 8.681366e-08
## omega3 8.681367e-08
## avevar 1.446894e-08
```

12.5.10.2.10 Path diagram Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModel2Fit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.5)
```

12.5.10.3 Model 3

Figure 12.21: Confirmatory factor analysis model 2 diagram

```
correlationMatrixModel3 <- matrix(c(1,.6,.6,0,0,0,
                                    .6,1,.6,0,0,0,
                                    0,0,1,0,0,0,
                                    0,0,0,1,.6,.6,
                                    0,0,0,.6,1,.6,
                                    0,0,0,.6,.6,1), nrow = 6, ncol = 6)
rownames(correlationMatrixModel3) <- colnames(correlationMatrixModel3) <- paste("V", 1

covarianceMatrixModel3 <- psych::cor2cov(correlationMatrixModel3, sigma = standardDevia
covarianceMatrixModel3
```

12.5.10.3.1 Create the covariance matrix

```
##     V1  V2 V3  V4  V5  V6
## V1 1.0 0.6  0 0.0 0.0 0.0
## V2 0.6 1.0  0 0.0 0.0 0.0
## V3 0.6 0.6  1 0.0 0.0 0.0
## V4 0.0 0.0  0 1.0 0.6 0.6
## V5 0.0 0.0  0 0.6 1.0 0.6
## V6 0.0 0.0  0 0.6 0.6 1.0
```

```
cfaModel3A <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3 + V4 + V5 + V6
'

cfaModel3B <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3
 f2 =~ V4 + V5 + V6
'
```

12.5.10.3.2 Specify the model

```
lavaanify(cfaModel3A)
```

12.5.10.3.3 Model syntax in table form:

```
##    id lhs op rhs user block group free ustart exo label plabel
## 1   1  f1 =~  V1    1     1     1    1     NA   0         .p1.
## 2   2  f1 =~  V2    1     1     1    2     NA   0         .p2.
## 3   3  f1 =~  V3    1     1     1    3     NA   0         .p3.
## 4   4  f1 =~  V4    1     1     1    4     NA   0         .p4.
## 5   5  f1 =~  V5    1     1     1    5     NA   0         .p5.
## 6   6  f1 =~  V6    1     1     1    6     NA   0         .p6.
## 7   7  V1 ~~  V1    0     1     1    0      0   0         .p7.
## 8   8  V2 ~~  V2    0     1     1    0      0   0         .p8.
## 9   9  V3 ~~  V3    0     1     1    0      0   0         .p9.
## 10 10  V4 ~~  V4    0     1     1    0      0   0        .p10.
## 11 11  V5 ~~  V5    0     1     1    0      0   0        .p11.
## 12 12  V6 ~~  V6    0     1     1    0      0   0        .p12.
## 13 13  f1 ~~  f1    0     1     1    0      0   0        .p13.
```

```
lavaanify(cfaModel3B)
```

```
##    id lhs op rhs user block group free ustart exo label plabel
## 1   1  f1 =~  V1    1     1     1    1     NA   0         .p1.
## 2   2  f1 =~  V2    1     1     1    2     NA   0         .p2.
## 3   3  f1 =~  V3    1     1     1    3     NA   0         .p3.
## 4   4  f2 =~  V4    1     1     1    4     NA   0         .p4.
## 5   5  f2 =~  V5    1     1     1    5     NA   0         .p5.
## 6   6  f2 =~  V6    1     1     1    6     NA   0         .p6.
## 7   7  V1 ~~  V1    0     1     1    0      0   0         .p7.
## 8   8  V2 ~~  V2    0     1     1    0      0   0         .p8.
## 9   9  V3 ~~  V3    0     1     1    0      0   0         .p9.
## 10 10  V4 ~~  V4    0     1     1    0      0   0        .p10.
## 11 11  V5 ~~  V5    0     1     1    0      0   0        .p11.
## 12 12  V6 ~~  V6    0     1     1    0      0   0        .p12.
## 13 13  f1 ~~  f1    0     1     1    0      0   0        .p13.
## 14 14  f2 ~~  f2    0     1     1    0      0   0        .p14.
```

```
cfaModel3AFit <- cfa(cfaModel3A,
                     sample.cov = covarianceMatrixModel3,
                     sample.nobs = sampleSize,
                     estimator = "ML",
                     std.lv = TRUE)

cfaModel3BFit <- cfa(cfaModel3B,
                     sample.cov = covarianceMatrixModel3,
                     sample.nobs = sampleSize,
```

```
                      estimator = "ML",
                      std.lv = TRUE)
```

12.5.10.3.4 Fit the model

```
summary(cfaModel3AFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

12.5.10.3.5 Display summary output

```
## lavaan 0.6-8 ended normally after 11 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        12
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                               313.237
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                               626.474
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.502
##   Tucker-Lewis Index (TLI)                       0.171
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2394.466
##   Loglikelihood unrestricted model (H1)      -2237.847
##
```

```
##   Akaike (AIC)                                4812.931
##   Bayesian (BIC)                              4857.377
##   Sample-size adjusted Bayesian (BIC)         4819.320
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.336
##   90 Percent confidence interval - lower         0.304
##   90 Percent confidence interval - upper         0.368
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.227
##
## 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
##   f1 =~
##     V1                0.773    0.055   14.142    0.000    0.773    0.775
##     V2                0.773    0.055   14.142    0.000    0.773    0.775
##     V3                0.773    0.055   14.142    0.000    0.773    0.775
##     V4                0.000    0.064    0.000    1.000    0.000    0.000
##     V5                0.000    0.064    0.000    1.000    0.000    0.000
##     V6                0.000    0.064    0.000    1.000    0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.399    0.051    7.746    0.000    0.399    0.400
##    .V2                0.399    0.051    7.746    0.000    0.399    0.400
##    .V3                0.399    0.051    7.746    0.000    0.399    0.400
##    .V4                0.997    0.081   12.247    0.000    0.997    1.000
##    .V5                0.997    0.081   12.247    0.000    0.997    1.000
##    .V6                0.997    0.081   12.247    0.000    0.997    1.000
##     f1                1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     V1                0.600
##     V2                0.600
##     V3                0.600
```

```
##     V4                0.000
##     V5                0.000
##     V6                0.000
```

```
summary(cfaModel3BFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 8 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           1.000
##
## Model Test Baseline Model:
##
##   Test statistic                               626.474
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.025
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2237.847
##   Loglikelihood unrestricted model (H1)      -2237.847
##
##   Akaike (AIC)                                4501.694
##   Bayesian (BIC)                              4549.843
##   Sample-size adjusted Bayesian (BIC)         4508.615
##
## 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 RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## 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
##   f1 =~
##     V1                0.773    0.055   14.142    0.000    0.773    0.775
##     V2                0.773    0.055   14.142    0.000    0.773    0.775
##     V3                0.773    0.055   14.142    0.000    0.773    0.775
##   f2 =~
##     V4                0.773    0.055   14.142    0.000    0.773    0.775
##     V5                0.773    0.055   14.142    0.000    0.773    0.775
##     V6                0.773    0.055   14.142    0.000    0.773    0.775
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   f1 ~~
##     f2                0.000    0.071    0.000    1.000    0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.399    0.051    7.746    0.000    0.399    0.400
##    .V2                0.399    0.051    7.746    0.000    0.399    0.400
##    .V3                0.399    0.051    7.746    0.000    0.399    0.400
##    .V4                0.399    0.051    7.746    0.000    0.399    0.400
##    .V5                0.399    0.051    7.746    0.000    0.399    0.400
##    .V6                0.399    0.051    7.746    0.000    0.399    0.400
##     f1                1.000                                1.000    1.000
##     f2                1.000                                1.000    1.000
##
## R-Square:
##                    Estimate
##     V1                0.600
##     V2                0.600
```

```
##     V3                0.600
##     V4                0.600
##     V5                0.600
##     V6                0.600
```

```
fitMeasures(cfaModel3AFit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "
```

12.5.10.3.6 Estimates of model fit

```
##   chisq      df  pvalue   rmsea     cfi     tli    srmr
## 313.237   9.000   0.000   0.336   0.502   0.171   0.227
```

```
fitMeasures(cfaModel3BFit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "
```

```
##  chisq     df pvalue  rmsea    cfi    tli   srmr
##  0.000  8.000  1.000  0.000  1.000  1.025  0.000
```

12.5.10.3.7 Compare model fit

Test of whether two-factor model fits better than one-factor model. Significant chi-square difference test indicates that two-factor model fits significantly better than one-factor model.

```
anova(cfaModel3BFit, cfaModel3AFit)
```

```
## Chi-Squared Difference Test
##
##               Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## cfaModel3BFit  8 4501.7 4549.8   0.00
## cfaModel3AFit  9 4812.9 4857.4 313.24     313.24       1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residuals(cfaModel3AFit, type = "cor")
```

12.5.10.3.8 Residuals

```
## $type
## [1] "cor.bollen"
```

```
##
## $cov
##    V1  V2  V3  V4  V5  V6
## V1 0.0
## V2 0.0 0.0
## V3 0.0 0.0 0.0
## V4 0.0 0.0 0.0 0.0
## V5 0.0 0.0 0.0 0.6 0.0
## V6 0.0 0.0 0.0 0.6 0.6 0.0
```

```
residuals(cfaModel3BFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##    V1 V2 V3 V4 V5 V6
## V1 0
## V2 0  0
## V3 0  0  0
## V4 0  0  0  0
## V5 0  0  0  0  0
## V6 0  0  0  0  0  0
```

```
modificationindices(cfaModel3AFit)
```

12.5.10.3.9 Modification indices

```
##    lhs op rhs  mi   epc sepc.lv sepc.all sepc.nox
## 16  V1 ~~  V4   0 0.000   0.000      0.0      0.0
## 17  V1 ~~  V5   0 0.000   0.000      0.0      0.0
## 18  V1 ~~  V6   0 0.000   0.000      0.0      0.0
## 20  V2 ~~  V4   0 0.000   0.000      0.0      0.0
## 21  V2 ~~  V5   0 0.000   0.000      0.0      0.0
## 22  V2 ~~  V6   0 0.000   0.000      0.0      0.0
## 23  V3 ~~  V4   0 0.000   0.000      0.0      0.0
## 24  V3 ~~  V5   0 0.000   0.000      0.0      0.0
## 25  V3 ~~  V6   0 0.000   0.000      0.0      0.0
## 26  V4 ~~  V5 108 0.598   0.598      0.6      0.6
## 27  V4 ~~  V6 108 0.598   0.598      0.6      0.6
## 28  V5 ~~  V6 108 0.598   0.598      0.6      0.6
```

```
modificationindices(cfaModel3BFit)
```

```
##    lhs op rhs mi epc sepc.lv sepc.all sepc.nox
## 16  f1 =~  V4  0   0       0        0        0
## 17  f1 =~  V5  0   0       0        0        0
## 18  f1 =~  V6  0   0       0        0        0
## 19  f2 =~  V1  0   0       0        0        0
## 20  f2 =~  V2  0   0       0        0        0
## 21  f2 =~  V3  0   0       0        0        0
## 24  V1 ~~  V4  0   0       0        0        0
## 25  V1 ~~  V5  0   0       0        0        0
## 26  V1 ~~  V6  0   0       0        0        0
## 28  V2 ~~  V4  0   0       0        0        0
## 29  V2 ~~  V5  0   0       0        0        0
## 30  V2 ~~  V6  0   0       0        0        0
## 31  V3 ~~  V4  0   0       0        0        0
## 32  V3 ~~  V5  0   0       0        0        0
## 33  V3 ~~  V6  0   0       0        0        0
```

12.5.10.3.10 Reliability Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModel3AFit)
```

```
##               f1
## alpha  0.8181818
## omega  0.8181818
## omega2 0.8181818
## omega3 0.8181818
## avevar 0.6000000
```

```
semTools::reliability(cfaModel3BFit)
```

```
##               f1        f2
## alpha  0.8181818 0.8181818
## omega  0.8181818 0.8181818
## omega2 0.8181818 0.8181818
## omega3 0.8181818 0.8181818
## avevar 0.6000000 0.6000000
```

12.5.10.3.11 Path diagram Below are path diagrams of the models generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModel3AFit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.5)
```

```
semPaths(cfaModel3BFit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.5)
```

12.5.10.4 Model 4

```
correlationMatrixModel4 <- matrix(c(1,.6,.6,.3,.3,.3,
                                    .6,1,.6,.3,.3,.3,
                                    .3,.3,1,.3,.3,.3,
                                    .3,.3,.3,1,.6,.6,
                                    .3,.3,.3,.6,1,.6,
                                    .3,.3,.3,.6,.6,1), nrow = 6, ncol = 6)
rownames(correlationMatrixModel4) <- colnames(correlationMatrixModel4) <- paste("V", 1:6, sep = '

covarianceMatrixModel4 <- psych::cor2cov(correlationMatrixModel4, sigma = standardDeviations)
covarianceMatrixModel4
```

12.5.10.4.1 Create the covariance matrix

```
##     V1  V2  V3  V4  V5  V6
## V1 1.0 0.6 0.3 0.3 0.3 0.3
## V2 0.6 1.0 0.3 0.3 0.3 0.3
## V3 0.6 0.6 1.0 0.3 0.3 0.3
## V4 0.3 0.3 0.3 1.0 0.6 0.6
## V5 0.3 0.3 0.3 0.6 1.0 0.6
## V6 0.3 0.3 0.3 0.6 0.6 1.0
```

Figure 12.22: Confirmatory factor analysis model 3A diagram

Figure 12.23: Confirmatory factor analysis model 3B diagram

```
cfaModel4A <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3
 f2 =~ V4 + V5 + V6
'

cfaModel4B <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3
 f2 =~ V4 + V5 + V6

 #Regression path
 f2 ~ f1
'

cfaModel4C <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3
 f2 =~ V4 + V5 + V6

 #Higher-order factor
 A1 = ~ fixloading*f1 + fixloading*f2
'

cfaModel4D <- '
 #Factor loadings
 f1 =~ V1 + V2 + V3 + V4 + V5 + V6

 #Correlated residuals/errors
 V1 ~~ fixcor*V2
 V2 ~~ fixcor*V3
 V1 ~~ fixcor*V3
 V4 ~~ fixcor*V5
 V5 ~~ fixcor*V6
 V4 ~~ fixcor*V6
'
```

12.5.10.4.2 Specify the model

```
lavaanify(cfaModel4A)
```

12.5.10.4.3 Model syntax in table form:

```
##    id lhs op rhs user block group free ustart exo label plabel
## 1   1  f1 =~  V1    1     1     1    1     NA   0         .p1.
## 2   2  f1 =~  V2    1     1     1    2     NA   0         .p2.
## 3   3  f1 =~  V3    1     1     1    3     NA   0         .p3.
## 4   4  f2 =~  V4    1     1     1    4     NA   0         .p4.
## 5   5  f2 =~  V5    1     1     1    5     NA   0         .p5.
## 6   6  f2 =~  V6    1     1     1    6     NA   0         .p6.
## 7   7  V1 ~~  V1    0     1     1    0      0   0         .p7.
## 8   8  V2 ~~  V2    0     1     1    0      0   0         .p8.
## 9   9  V3 ~~  V3    0     1     1    0      0   0         .p9.
## 10 10  V4 ~~  V4    0     1     1    0      0   0        .p10.
## 11 11  V5 ~~  V5    0     1     1    0      0   0        .p11.
## 12 12  V6 ~~  V6    0     1     1    0      0   0        .p12.
## 13 13  f1 ~~  f1    0     1     1    0      0   0        .p13.
## 14 14  f2 ~~  f2    0     1     1    0      0   0        .p14.
```

```
lavaanify(cfaModel4B)
```

```
##    id lhs op rhs user block group free ustart exo label plabel
## 1   1  f1 =~  V1    1     1     1    1     NA   0         .p1.
## 2   2  f1 =~  V2    1     1     1    2     NA   0         .p2.
## 3   3  f1 =~  V3    1     1     1    3     NA   0         .p3.
## 4   4  f2 =~  V4    1     1     1    4     NA   0         .p4.
## 5   5  f2 =~  V5    1     1     1    5     NA   0         .p5.
## 6   6  f2 =~  V6    1     1     1    6     NA   0         .p6.
## 7   7  f2  ~  f1    1     1     1    7     NA   0         .p7.
## 8   8  V1 ~~  V1    0     1     1    0      0   0         .p8.
## 9   9  V2 ~~  V2    0     1     1    0      0   0         .p9.
## 10 10  V3 ~~  V3    0     1     1    0      0   0        .p10.
## 11 11  V4 ~~  V4    0     1     1    0      0   0        .p11.
## 12 12  V5 ~~  V5    0     1     1    0      0   0        .p12.
## 13 13  V6 ~~  V6    0     1     1    0      0   0        .p13.
## 14 14  f1 ~~  f1    0     1     1    0      0   0        .p14.
## 15 15  f2 ~~  f2    0     1     1    0      0   0        .p15.
```

```
lavaanify(cfaModel4C)
```

```
##    id  lhs op  rhs user block group free ustart exo      label plabel
## 1   1   f1 =~   V1    1     1     1    1     NA   0               .p1.
## 2   2   f1 =~   V2    1     1     1    2     NA   0               .p2.
## 3   3   f1 =~   V3    1     1     1    3     NA   0               .p3.
## 4   4   f2 =~   V4    1     1     1    4     NA   0               .p4.
```

```
## 5   5   f2 =~   V5    1     1     1    5     NA   0              .p5.
## 6   6   f2 =~   V6    1     1     1    6     NA   0              .p6.
## 7   7   A1 =~   f1    1     1     1    7     NA   0 fixloading   .p7.
## 8   8   A1 =~   f2    1     1     1    8     NA   0 fixloading   .p8.
## 9   9   V1 ~~   V1    0     1     1    0      0   0              .p9.
## 10 10   V2 ~~   V2    0     1     1    0      0   0             .p10.
## 11 11   V3 ~~   V3    0     1     1    0      0   0             .p11.
## 12 12   V4 ~~   V4    0     1     1    0      0   0             .p12.
## 13 13   V5 ~~   V5    0     1     1    0      0   0             .p13.
## 14 14   V6 ~~   V6    0     1     1    0      0   0             .p14.
## 15 15   f1 ~~   f1    0     1     1    0      0   0             .p15.
## 16 16   f2 ~~   f2    0     1     1    0      0   0             .p16.
## 17 17   A1 ~~   A1    0     1     1    0      0   0             .p17.
## 18 18 .p7. == .p8.    2     0     0    0     NA   0
```

```
lavaanify(cfaModel4D)
```

```
##    id  lhs op   rhs user block group free ustart exo  label plabel
## 1   1   f1 =~    V1    1     1     1    1     NA   0          .p1.
## 2   2   f1 =~    V2    1     1     1    2     NA   0          .p2.
## 3   3   f1 =~    V3    1     1     1    3     NA   0          .p3.
## 4   4   f1 =~    V4    1     1     1    4     NA   0          .p4.
## 5   5   f1 =~    V5    1     1     1    5     NA   0          .p5.
## 6   6   f1 =~    V6    1     1     1    6     NA   0          .p6.
## 7   7   V1 ~~    V2    1     1     1    7     NA   0 fixcor   .p7.
## 8   8   V2 ~~    V3    1     1     1    8     NA   0 fixcor   .p8.
## 9   9   V1 ~~    V3    1     1     1    9     NA   0 fixcor   .p9.
## 10 10   V4 ~~    V5    1     1     1   10     NA   0 fixcor  .p10.
## 11 11   V5 ~~    V6    1     1     1   11     NA   0 fixcor  .p11.
## 12 12   V4 ~~    V6    1     1     1   12     NA   0 fixcor  .p12.
## 13 13   V1 ~~    V1    0     1     1    0      0   0         .p13.
## 14 14   V2 ~~    V2    0     1     1    0      0   0         .p14.
## 15 15   V3 ~~    V3    0     1     1    0      0   0         .p15.
## 16 16   V4 ~~    V4    0     1     1    0      0   0         .p16.
## 17 17   V5 ~~    V5    0     1     1    0      0   0         .p17.
## 18 18   V6 ~~    V6    0     1     1    0      0   0         .p18.
## 19 19   f1 ~~    f1    0     1     1    0      0   0         .p19.
## 20 20 .p7. ==  .p8.    2     0     0    0     NA   0
## 21 21 .p7. ==  .p9.    2     0     0    0     NA   0
## 22 22 .p7. == .p10.    2     0     0    0     NA   0
## 23 23 .p7. == .p11.    2     0     0    0     NA   0
## 24 24 .p7. == .p12.    2     0     0    0     NA   0
```

```
cfaModel4AFit <- cfa(cfaModel4A,
                     sample.cov = covarianceMatrixModel4,
                     sample.nobs = sampleSize,
                     estimator = "ML",
                     std.lv = TRUE)

cfaModel4BFit <- cfa(cfaModel4B,
                     sample.cov = covarianceMatrixModel4,
                     sample.nobs = sampleSize,
                     estimator = "ML",
                     std.lv = TRUE)

cfaModel4CFit <- cfa(cfaModel4C,
                     sample.cov = covarianceMatrixModel4,
                     sample.nobs = sampleSize,
                     estimator = "ML",
                     std.lv = TRUE)

cfaModel4DFit <- cfa(cfaModel4D,
                     sample.cov = covarianceMatrixModel4,
                     sample.nobs = sampleSize,
                     estimator = "ML",
                     std.lv = TRUE)
```

12.5.10.4.4 Fit the model

```
summary(cfaModel4AFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

12.5.10.4.5 Display summary output

```
## lavaan 0.6-8 ended normally after 9 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                           300
```

```
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           1.000
##
## Model Test Baseline Model:
##
##   Test statistic                               681.419
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.023
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2210.375
##   Loglikelihood unrestricted model (H1)      -2210.375
##
##   Akaike (AIC)                                4446.750
##   Bayesian (BIC)                              4494.899
##   Sample-size adjusted Bayesian (BIC)         4453.671
##
## 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 RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## 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
```

```
##   f1 =~
##     V1                0.773    0.054   14.351    0.000    0.773    0.775
##     V2                0.773    0.054   14.351    0.000    0.773    0.775
##     V3                0.773    0.054   14.351    0.000    0.773    0.775
##   f2 =~
##     V4                0.773    0.054   14.351    0.000    0.773    0.775
##     V5                0.773    0.054   14.351    0.000    0.773    0.775
##     V6                0.773    0.054   14.351    0.000    0.773    0.775
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   f1 ~~
##     f2                0.500    0.057    8.822    0.000    0.500    0.500
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.399    0.049    8.067    0.000    0.399    0.400
##    .V2                0.399    0.049    8.067    0.000    0.399    0.400
##    .V3                0.399    0.049    8.067    0.000    0.399    0.400
##    .V4                0.399    0.049    8.067    0.000    0.399    0.400
##    .V5                0.399    0.049    8.067    0.000    0.399    0.400
##    .V6                0.399    0.049    8.067    0.000    0.399    0.400
##     f1                1.000                               1.000    1.000
##     f2                1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     V1                0.600
##     V2                0.600
##     V3                0.600
##     V4                0.600
##     V5                0.600
##     V6                0.600
```

```
summary(cfaModel4BFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 11 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
```

```
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           1.000
##
## Model Test Baseline Model:
##
##   Test statistic                               681.419
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.023
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2210.375
##   Loglikelihood unrestricted model (H1)      -2210.375
##
##   Akaike (AIC)                                4446.750
##   Bayesian (BIC)                              4494.899
##   Sample-size adjusted Bayesian (BIC)         4453.671
##
## 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 RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## 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
##   f1 =~
##     V1                0.773    0.054   14.351    0.000    0.773    0.775
##     V2                0.773    0.054   14.351    0.000    0.773    0.775
##     V3                0.773    0.054   14.351    0.000    0.773    0.775
##   f2 =~
##     V4                0.670    0.050   13.445    0.000    0.773    0.775
##     V5                0.670    0.050   13.445    0.000    0.773    0.775
##     V6                0.670    0.050   13.445    0.000    0.773    0.775
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   f2 ~
##     f1                0.577    0.087    6.616    0.000    0.500    0.500
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.399    0.049    8.067    0.000    0.399    0.400
##    .V2                0.399    0.049    8.067    0.000    0.399    0.400
##    .V3                0.399    0.049    8.067    0.000    0.399    0.400
##    .V4                0.399    0.049    8.067    0.000    0.399    0.400
##    .V5                0.399    0.049    8.067    0.000    0.399    0.400
##    .V6                0.399    0.049    8.067    0.000    0.399    0.400
##     f1                1.000                               1.000    1.000
##    .f2                1.000                               0.750    0.750
##
## R-Square:
##                    Estimate
##     V1                0.600
##     V2                0.600
##     V3                0.600
##     V4                0.600
##     V5                0.600
##     V6                0.600
##     f2                0.250
```

```r
summary(cfaModel4CFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 15 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
```

```
##   Number of model parameters                        14
##   Number of equality constraints                     1
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           1.000
##
## Model Test Baseline Model:
##
##   Test statistic                               681.419
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.023
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2210.375
##   Loglikelihood unrestricted model (H1)      -2210.375
##
##   Akaike (AIC)                                4446.750
##   Bayesian (BIC)                              4494.899
##   Sample-size adjusted Bayesian (BIC)         4453.671
##
## 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 RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## 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
##   f1 =~
##     V1                0.547    0.046   12.003    0.000    0.773    0.775
##     V2                0.547    0.046   12.003    0.000    0.773    0.775
##     V3                0.547    0.046   12.003    0.000    0.773    0.775
##   f2 =~
##     V4                0.547    0.046   12.003    0.000    0.773    0.775
##     V5                0.547    0.046   12.003    0.000    0.773    0.775
##     V6                0.547    0.046   12.003    0.000    0.773    0.775
##   A1 =~
##     f1      (fxld)    1.000    0.113    8.822    0.000    0.707    0.707
##     f2      (fxld)    1.000    0.113    8.822    0.000    0.707    0.707
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.399    0.049    8.067    0.000    0.399    0.400
##    .V2                0.399    0.049    8.067    0.000    0.399    0.400
##    .V3                0.399    0.049    8.067    0.000    0.399    0.400
##    .V4                0.399    0.049    8.067    0.000    0.399    0.400
##    .V5                0.399    0.049    8.067    0.000    0.399    0.400
##    .V6                0.399    0.049    8.067    0.000    0.399    0.400
##    .f1                1.000                               0.500    0.500
##    .f2                1.000                               0.500    0.500
##     A1                1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     V1                0.600
##     V2                0.600
##     V3                0.600
##     V4                0.600
##     V5                0.600
##     V6                0.600
##     f1                0.500
##     f2                0.500
```

```r
summary(cfaModel4DFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 17 iterations
```

```
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        18
##   Number of equality constraints                     5
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           1.000
##
## Model Test Baseline Model:
##
##   Test statistic                               681.419
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.023
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2210.375
##   Loglikelihood unrestricted model (H1)      -2210.375
##
##   Akaike (AIC)                                4446.750
##   Bayesian (BIC)                              4494.899
##   Sample-size adjusted Bayesian (BIC)         4453.671
##
## 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 RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## 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
##   f1 =~
##     V1                0.547    0.068    7.994    0.000    0.547    0.548
##     V2                0.547    0.068    7.994    0.000    0.547    0.548
##     V3                0.547    0.068    7.994    0.000    0.547    0.548
##     V4                0.547    0.068    7.994    0.000    0.547    0.548
##     V5                0.547    0.068    7.994    0.000    0.547    0.548
##     V6                0.547    0.068    7.994    0.000    0.547    0.548
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .V1 ~~
##    .V2      (fxcr)    0.299    0.036    8.380    0.000    0.299    0.429
##  .V2 ~~
##    .V3      (fxcr)    0.299    0.036    8.380    0.000    0.299    0.429
##  .V1 ~~
##    .V3      (fxcr)    0.299    0.036    8.380    0.000    0.299    0.429
##  .V4 ~~
##    .V5      (fxcr)    0.299    0.036    8.380    0.000    0.299    0.429
##  .V5 ~~
##    .V6      (fxcr)    0.299    0.036    8.380    0.000    0.299    0.429
##  .V4 ~~
##    .V6      (fxcr)    0.299    0.036    8.380    0.000    0.299    0.429
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.698    0.057   12.244    0.000    0.698    0.700
##    .V2                0.698    0.057   12.244    0.000    0.698    0.700
##    .V3                0.698    0.057   12.244    0.000    0.698    0.700
##    .V4                0.698    0.057   12.244    0.000    0.698    0.700
##    .V5                0.698    0.057   12.244    0.000    0.698    0.700
##    .V6                0.698    0.057   12.244    0.000    0.698    0.700
##     f1                1.000                              1.000    1.000
##
## R-Square:
##                    Estimate
##     V1                0.300
##     V2                0.300
##     V3                0.300
##     V4                0.300
```

```
##     V5                0.300
##     V6                0.300
```

```
fitMeasures(cfaModel4AFit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "t
```

12.5.10.4.6 Estimates of model fit

```
##  chisq     df pvalue  rmsea    cfi    tli   srmr
##  0.000  8.000  1.000  0.000  1.000  1.023  0.000
```

```
fitMeasures(cfaModel4BFit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "t
```

```
##  chisq     df pvalue  rmsea    cfi    tli   srmr
##  0.000  8.000  1.000  0.000  1.000  1.023  0.000
```

```
fitMeasures(cfaModel4CFit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "t
```

```
##  chisq     df pvalue  rmsea    cfi    tli   srmr
##  0.000  8.000  1.000  0.000  1.000  1.023  0.000
```

```
fitMeasures(cfaModel4DFit, fit.measures = c("chisq", "df", "pvalue", "rmsea", "cfi", "t
```

```
##  chisq     df pvalue  rmsea    cfi    tli   srmr
##  0.000  8.000  1.000  0.000  1.000  1.023  0.000
```

```
residuals(cfaModel4AFit, type = "cor")
```

12.5.10.4.7 Residuals

```
## $type
## [1] "cor.bollen"
##
## $cov
##    V1 V2 V3 V4 V5 V6
## V1 0
## V2 0  0
## V3 0  0  0
## V4 0  0  0  0
## V5 0  0  0  0  0
## V6 0  0  0  0  0  0
```

```r
residuals(cfaModel4BFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##    V1 V2 V3 V4 V5 V6
## V1 0
## V2 0  0
## V3 0  0  0
## V4 0  0  0  0
## V5 0  0  0  0  0
## V6 0  0  0  0  0  0
```

```r
residuals(cfaModel4CFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##    V1 V2 V3 V4 V5 V6
## V1 0
## V2 0  0
## V3 0  0  0
## V4 0  0  0  0
## V5 0  0  0  0  0
## V6 0  0  0  0  0  0
```

```r
residuals(cfaModel4DFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##    V1 V2 V3 V4 V5 V6
## V1 0
## V2 0  0
## V3 0  0  0
## V4 0  0  0  0
## V5 0  0  0  0  0
## V6 0  0  0  0  0  0
```

```
modificationindices(cfaModel4AFit)
```

12.5.10.4.8 Modification indices

```
##    lhs op rhs mi epc sepc.lv sepc.all sepc.nox
## 16  f1 =~  V4  0   0       0        0        0
## 17  f1 =~  V5  0   0       0        0        0
## 18  f1 =~  V6  0   0       0        0        0
## 19  f2 =~  V1  0   0       0        0        0
## 20  f2 =~  V2  0   0       0        0        0
## 21  f2 =~  V3  0   0       0        0        0
## 22  V1 ~~  V2  0   0       0        0        0
## 23  V1 ~~  V3  0   0       0        0        0
## 24  V1 ~~  V4  0   0       0        0        0
## 25  V1 ~~  V5  0   0       0        0        0
## 26  V1 ~~  V6  0   0       0        0        0
## 27  V2 ~~  V3  0   0       0        0        0
## 28  V2 ~~  V4  0   0       0        0        0
## 29  V2 ~~  V5  0   0       0        0        0
## 30  V2 ~~  V6  0   0       0        0        0
## 31  V3 ~~  V4  0   0       0        0        0
## 32  V3 ~~  V5  0   0       0        0        0
## 33  V3 ~~  V6  0   0       0        0        0
## 34  V4 ~~  V5  0   0       0        0        0
## 35  V4 ~~  V6  0   0       0        0        0
## 36  V5 ~~  V6  0   0       0        0        0
```

```
modificationindices(cfaModel4BFit)
```

```
##    lhs op rhs mi epc sepc.lv sepc.all sepc.nox
## 16  f1 =~  V4  0   0       0        0        0
## 17  f1 =~  V5  0   0       0        0        0
## 18  f1 =~  V6  0   0       0        0        0
## 19  f2 =~  V1  0   0       0        0        0
## 20  f2 =~  V2  0   0       0        0        0
## 21  f2 =~  V3  0   0       0        0        0
## 22  V1 ~~  V2  0   0       0        0        0
## 23  V1 ~~  V3  0   0       0        0        0
## 24  V1 ~~  V4  0   0       0        0        0
## 25  V1 ~~  V5  0   0       0        0        0
## 26  V1 ~~  V6  0   0       0        0        0
## 27  V2 ~~  V3  0   0       0        0        0
## 28  V2 ~~  V4  0   0       0        0        0
```

```
## 29  V2 ~~  V5  0   0       0        0        0
## 30  V2 ~~  V6  0   0       0        0        0
## 31  V3 ~~  V4  0   0       0        0        0
## 32  V3 ~~  V5  0   0       0        0        0
## 33  V3 ~~  V6  0   0       0        0        0
## 34  V4 ~~  V5  0   0       0        0        0
## 35  V4 ~~  V6  0   0       0        0        0
## 36  V5 ~~  V6  0   0       0        0        0
```

```
modificationindices(cfaModel4CFit)
```

```
##    lhs op rhs mi epc sepc.lv sepc.all sepc.nox
## 19  f1 =~  V4  0   0       0        0        0
## 20  f1 =~  V5  0   0       0        0        0
## 21  f1 =~  V6  0   0       0        0        0
## 22  f2 =~  V1  0   0       0        0        0
## 23  f2 =~  V2  0   0       0        0        0
## 24  f2 =~  V3  0   0       0        0        0
## 25  A1 =~  V1  0   0       0        0        0
## 26  A1 =~  V2  0   0       0        0        0
## 27  A1 =~  V3  0   0       0        0        0
## 28  A1 =~  V4  0   0       0        0        0
## 29  A1 =~  V5  0   0       0        0        0
## 30  A1 =~  V6  0   0       0        0        0
## 31  V1 ~~  V2  0   0       0        0        0
## 32  V1 ~~  V3  0   0       0        0        0
## 33  V1 ~~  V4  0   0       0        0        0
## 34  V1 ~~  V5  0   0       0        0        0
## 35  V1 ~~  V6  0   0       0        0        0
## 36  V2 ~~  V3  0   0       0        0        0
## 37  V2 ~~  V4  0   0       0        0        0
## 38  V2 ~~  V5  0   0       0        0        0
## 39  V2 ~~  V6  0   0       0        0        0
## 40  V3 ~~  V4  0   0       0        0        0
## 41  V3 ~~  V5  0   0       0        0        0
## 42  V3 ~~  V6  0   0       0        0        0
## 43  V4 ~~  V5  0   0       0        0        0
## 44  V4 ~~  V6  0   0       0        0        0
## 45  V5 ~~  V6  0   0       0        0        0
```

```
modificationindices(cfaModel4DFit)
```

```
##    lhs op rhs mi epc sepc.lv sepc.all sepc.nox
## 25  V1 ~~  V4  0   0       0        0        0
```

```
## 26  V1 ~~  V5  0   0       0        0        0
## 27  V1 ~~  V6  0   0       0        0        0
## 28  V2 ~~  V4  0   0       0        0        0
## 29  V2 ~~  V5  0   0       0        0        0
## 30  V2 ~~  V6  0   0       0        0        0
## 31  V3 ~~  V4  0   0       0        0        0
## 32  V3 ~~  V5  0   0       0        0        0
## 33  V3 ~~  V6  0   0       0        0        0
```

12.5.10.4.9 Reliability Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModel4AFit)
```

```
##               f1        f2
## alpha  0.8181818 0.8181818
## omega  0.8181818 0.8181818
## omega2 0.8181818 0.8181818
## omega3 0.8181818 0.8181818
## avevar 0.6000000 0.6000000
```

```
semTools::reliability(cfaModel4BFit)
```

```
##               f1        f2
## alpha  0.8181818 0.8181818
## omega  0.8181818 0.8181818
## omega2 0.8181818 0.8181818
## omega3 0.8181818 0.8181818
## avevar 0.6000000 0.6000000
```

```
semTools::reliability(cfaModel4CFit)
```

```
##               f1        f2
## alpha  0.8181818 0.8181818
## omega  0.8181818 0.8181818
## omega2 0.8181818 0.8181818
## omega3 0.8181818 0.8181818
## avevar 0.6000000 0.6000000
```

```
semTools::reliabilityL2(cfaModel4CFit, "A1")
```

```
##        omegaL1        omegaL2 partialOmegaL1
##      0.5806452      0.6666667      0.8181818
```

```
semTools::reliability(cfaModel4DFit)
```

```
##               f1
## alpha  0.8129032
## omega  0.5806452
## omega2 0.5806452
## omega3 0.5806452
## avevar 0.3000000
```

12.5.10.4.10 Path diagram Below are path diagrams of the models generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModel4AFit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.3)
```

```
semPaths(cfaModel4BFit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.3)
```

```
semPaths(cfaModel4CFit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.3)
```

```
semPaths(cfaModel4DFit,
         what = "std",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         edge.label.cex = 1.3)
```

Figure 12.24: Confirmatory factor analysis model 4A diagram

Figure 12.25: Confirmatory factor analysis model 4B diagram

Figure 12.26: Confirmatory factor analysis model 4C diagram

Figure 12.27: Confirmatory factor analysis model 4D diagram

12.5.10.4.11 Equivalently fitting models Markov equivalent directed acyclic graphs (DAGs) were depicted using the `dagitty` package [Textor et al., 2021].

```
dagModel <- lavaanToGraph(cfaModel4AFit)

par(mfrow = c(4, 4))
```

```
lapply(equivalentDAGs(dagModel, n = 16), plot)
```

```
## [[1]]
## [[1]]$mar
## [1] 0 0 0 0
##
##
## [[2]]
## [[2]]$mar
## [1] 0 0 0 0
##
##
## [[3]]
## [[3]]$mar
## [1] 0 0 0 0
##
##
## [[4]]
## [[4]]$mar
## [1] 0 0 0 0
##
##
## [[5]]
## [[5]]$mar
## [1] 0 0 0 0
##
##
## [[6]]
## [[6]]$mar
## [1] 0 0 0 0
##
##
## [[7]]
## [[7]]$mar
## [1] 0 0 0 0
##
```

Figure 12.28: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.29: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.30: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.31: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.32: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.33: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.34: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.35: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.36: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.37: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.38: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.39: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.40: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.41: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.42: Equivalently fitting models to confirmatory factor analysis model 4

Figure 12.43: Equivalently fitting models to confirmatory factor analysis model 4

```
##
## [[8]]
## [[8]]$mar
## [1] 0 0 0 0
##
##
## [[9]]
## [[9]]$mar
## [1] 0 0 0 0
##
##
## [[10]]
## [[10]]$mar
## [1] 0 0 0 0
##
##
## [[11]]
## [[11]]$mar
## [1] 0 0 0 0
##
##
## [[12]]
## [[12]]$mar
## [1] 0 0 0 0
##
##
## [[13]]
## [[13]]$mar
## [1] 0 0 0 0
##
##
## [[14]]
## [[14]]$mar
## [1] 0 0 0 0
##
##
## [[15]]
## [[15]]$mar
## [1] 0 0 0 0
##
##
## [[16]]
## [[16]]$mar
## [1] 0 0 0 0
```

12.5.11 Bifactor Model

12.5.11.1 Specify the model

```
cfaModelBifactor_syntax <- '
  Verbal =~
    VO1 + VO2 + VO3 +
    VW1 + VW2 + VW3 +
    VM1 + VM2 + VM3
  Spatial =~
    SO1 + SO2 + SO3 +
    SW1 + SW2 + SW3 +
    SM1 + SM2 + SM3
  Quant =~
    QO1 + QO2 + QO3 +
    QW1 + QW2 + QW3 +
    QM1 + QM2 + QM3
  g =~
    VO1 + VO2 + VO3 +
    VW1 + VW2 + VW3 +
    VM1 + VM2 + VM3 +
    SO1 + SO2 + SO3 +
    SW1 + SW2 + SW3 +
    SM1 + SM2 + SM3 +
    QO1 + QO2 + QO3 +
    QW1 + QW2 + QW3 +
    QM1 + QM2 + QM3
'

cfaModelBifactor_fullSyntax <- '
  #Factor loadings (free the factor loading of the first indicator)
  Verbal =~
    NA*VO1 + VO2 + VO3 +
    VW1 + VW2 + VW3 +
    VM1 + VM2 + VM3
  Spatial =~
    NA*SO1 + SO2 + SO3 +
    SW1 + SW2 + SW3 +
    SM1 + SM2 + SM3
  Quant =~
    NA*QO1 + QO2 + QO3 +
    QW1 + QW2 + QW3 +
    QM1 + QM2 + QM3
  g =~
    NA*VO1 + VO2 + VO3 +
```

```
  VW1 + VW2 + VW3 +
  VM1 + VM2 + VM3 +
  SO1 + SO2 + SO3 +
  SW1 + SW2 + SW3 +
  SM1 + SM2 + SM3 +
  QO1 + QO2 + QO3 +
  QW1 + QW2 + QW3 +
  QM1 + QM2 + QM3

#Fix latent means to zero
Verbal ~ 0
Spatial ~ 0
Quant ~ 0
g ~ 0

#Fix latent variances to one
Verbal ~~ 1*Verbal
Spatial ~~ 1*Spatial
Quant ~~ 1*Quant
g ~~ 1*g

#Fix covariances among latent variables at zero
Verbal ~~ 0*Spatial
Verbal ~~ 0*Quant
Spatial ~~ 0*Quant
g ~~ 0*Verbal
g ~~ 0*Spatial
g ~~ 0*Quant

#Estimate residual variances of manifest variables
VO1 ~~ VO1
VO2 ~~ VO2
VO3 ~~ VO3
VW1 ~~ VW1
VW2 ~~ VW2
VW3 ~~ VW3
VM1 ~~ VM1
VM2 ~~ VM2
VM3 ~~ VM3
SO1 ~~ SO1
SO2 ~~ SO2
SO3 ~~ SO3
SW1 ~~ SW1
SW2 ~~ SW2
SW3 ~~ SW3
```

```
  SM1 ~~ SM1
  SM2 ~~ SM2
  SM3 ~~ SM3
  QO1 ~~ QO1
  QO2 ~~ QO2
  QO3 ~~ QO3
  QW1 ~~ QW1
  QW2 ~~ QW2
  QW3 ~~ QW3
  QM1 ~~ QM1
  QM2 ~~ QM2
  QM3 ~~ QM3

  #Free intercepts of manifest variables
  VO1 ~ intVO1*1
  VO2 ~ intVO2*1
  VO3 ~ intVO3*1
  VW1 ~ intVW1*1
  VW2 ~ intVW2*1
  VW3 ~ intVW3*1
  VM1 ~ intVM1*1
  VM2 ~ intVM2*1
  VM3 ~ intVM3*1
  SO1 ~ intSO1*1
  SO2 ~ intSO2*1
  SO3 ~ intSO3*1
  SW1 ~ intSW1*1
  SW2 ~ intSW2*1
  SW3 ~ intSW3*1
  SM1 ~ intSM1*1
  SM2 ~ intSM2*1
  SM3 ~ intSM3*1
  QO1 ~ intQO1*1
  QO2 ~ intQO2*1
  QO3 ~ intQO3*1
  QW1 ~ intQW1*1
  QW2 ~ intQW2*1
  QW3 ~ intQW3*1
  QM1 ~ intQM1*1
  QM2 ~ intQM2*1
  QM3 ~ intQM3*1
'
```

12.5.11.2 Model syntax in table form:

```
lavaanify(cfaModelBifactor_syntax)
```

```
##    id     lhs op     rhs user block group free ustart exo label plabel
## 1   1  Verbal =~     VO1    1     1     1    1     NA   0         .p1.
## 2   2  Verbal =~     VO2    1     1     1    2     NA   0         .p2.
## 3   3  Verbal =~     VO3    1     1     1    3     NA   0         .p3.
## 4   4  Verbal =~     VW1    1     1     1    4     NA   0         .p4.
## 5   5  Verbal =~     VW2    1     1     1    5     NA   0         .p5.
## 6   6  Verbal =~     VW3    1     1     1    6     NA   0         .p6.
## 7   7  Verbal =~     VM1    1     1     1    7     NA   0         .p7.
## 8   8  Verbal =~     VM2    1     1     1    8     NA   0         .p8.
## 9   9  Verbal =~     VM3    1     1     1    9     NA   0         .p9.
## 10 10 Spatial =~     SO1    1     1     1   10     NA   0        .p10.
## 11 11 Spatial =~     SO2    1     1     1   11     NA   0        .p11.
## 12 12 Spatial =~     SO3    1     1     1   12     NA   0        .p12.
## 13 13 Spatial =~     SW1    1     1     1   13     NA   0        .p13.
## 14 14 Spatial =~     SW2    1     1     1   14     NA   0        .p14.
## 15 15 Spatial =~     SW3    1     1     1   15     NA   0        .p15.
## 16 16 Spatial =~     SM1    1     1     1   16     NA   0        .p16.
## 17 17 Spatial =~     SM2    1     1     1   17     NA   0        .p17.
## 18 18 Spatial =~     SM3    1     1     1   18     NA   0        .p18.
## 19 19   Quant =~     QO1    1     1     1   19     NA   0        .p19.
## 20 20   Quant =~     QO2    1     1     1   20     NA   0        .p20.
## 21 21   Quant =~     QO3    1     1     1   21     NA   0        .p21.
## 22 22   Quant =~     QW1    1     1     1   22     NA   0        .p22.
## 23 23   Quant =~     QW2    1     1     1   23     NA   0        .p23.
## 24 24   Quant =~     QW3    1     1     1   24     NA   0        .p24.
## 25 25   Quant =~     QM1    1     1     1   25     NA   0        .p25.
## 26 26   Quant =~     QM2    1     1     1   26     NA   0        .p26.
## 27 27   Quant =~     QM3    1     1     1   27     NA   0        .p27.
## 28 28       g =~     VO1    1     1     1   28     NA   0        .p28.
## 29 29       g =~     VO2    1     1     1   29     NA   0        .p29.
## 30 30       g =~     VO3    1     1     1   30     NA   0        .p30.
## 31 31       g =~     VW1    1     1     1   31     NA   0        .p31.
## 32 32       g =~     VW2    1     1     1   32     NA   0        .p32.
## 33 33       g =~     VW3    1     1     1   33     NA   0        .p33.
## 34 34       g =~     VM1    1     1     1   34     NA   0        .p34.
## 35 35       g =~     VM2    1     1     1   35     NA   0        .p35.
## 36 36       g =~     VM3    1     1     1   36     NA   0        .p36.
## 37 37       g =~     SO1    1     1     1   37     NA   0        .p37.
## 38 38       g =~     SO2    1     1     1   38     NA   0        .p38.
## 39 39       g =~     SO3    1     1     1   39     NA   0        .p39.
```

```
## 40 40       g =~     SW1    1     1     1   40     NA   0        .p40.
## 41 41       g =~     SW2    1     1     1   41     NA   0        .p41.
## 42 42       g =~     SW3    1     1     1   42     NA   0        .p42.
## 43 43       g =~     SM1    1     1     1   43     NA   0        .p43.
## 44 44       g =~     SM2    1     1     1   44     NA   0        .p44.
## 45 45       g =~     SM3    1     1     1   45     NA   0        .p45.
## 46 46       g =~     QO1    1     1     1   46     NA   0        .p46.
## 47 47       g =~     QO2    1     1     1   47     NA   0        .p47.
## 48 48       g =~     QO3    1     1     1   48     NA   0        .p48.
## 49 49       g =~     QW1    1     1     1   49     NA   0        .p49.
## 50 50       g =~     QW2    1     1     1   50     NA   0        .p50.
## 51 51       g =~     QW3    1     1     1   51     NA   0        .p51.
## 52 52       g =~     QM1    1     1     1   52     NA   0        .p52.
## 53 53       g =~     QM2    1     1     1   53     NA   0        .p53.
## 54 54       g =~     QM3    1     1     1   54     NA   0        .p54.
## 55 55     VO1 ~~     VO1    0     1     1    0      0   0        .p55.
## 56 56     VO2 ~~     VO2    0     1     1    0      0   0        .p56.
## 57 57     VO3 ~~     VO3    0     1     1    0      0   0        .p57.
## 58 58     VW1 ~~     VW1    0     1     1    0      0   0        .p58.
## 59 59     VW2 ~~     VW2    0     1     1    0      0   0        .p59.
## 60 60     VW3 ~~     VW3    0     1     1    0      0   0        .p60.
## 61 61     VM1 ~~     VM1    0     1     1    0      0   0        .p61.
## 62 62     VM2 ~~     VM2    0     1     1    0      0   0        .p62.
## 63 63     VM3 ~~     VM3    0     1     1    0      0   0        .p63.
## 64 64     SO1 ~~     SO1    0     1     1    0      0   0        .p64.
## 65 65     SO2 ~~     SO2    0     1     1    0      0   0        .p65.
## 66 66     SO3 ~~     SO3    0     1     1    0      0   0        .p66.
## 67 67     SW1 ~~     SW1    0     1     1    0      0   0        .p67.
## 68 68     SW2 ~~     SW2    0     1     1    0      0   0        .p68.
## 69 69     SW3 ~~     SW3    0     1     1    0      0   0        .p69.
## 70 70     SM1 ~~     SM1    0     1     1    0      0   0        .p70.
## 71 71     SM2 ~~     SM2    0     1     1    0      0   0        .p71.
## 72 72     SM3 ~~     SM3    0     1     1    0      0   0        .p72.
## 73 73     QO1 ~~     QO1    0     1     1    0      0   0        .p73.
## 74 74     QO2 ~~     QO2    0     1     1    0      0   0        .p74.
## 75 75     QO3 ~~     QO3    0     1     1    0      0   0        .p75.
## 76 76     QW1 ~~     QW1    0     1     1    0      0   0        .p76.
## 77 77     QW2 ~~     QW2    0     1     1    0      0   0        .p77.
## 78 78     QW3 ~~     QW3    0     1     1    0      0   0        .p78.
## 79 79     QM1 ~~     QM1    0     1     1    0      0   0        .p79.
## 80 80     QM2 ~~     QM2    0     1     1    0      0   0        .p80.
## 81 81     QM3 ~~     QM3    0     1     1    0      0   0        .p81.
## 82 82  Verbal ~~  Verbal    0     1     1    0      0   0        .p82.
## 83 83 Spatial ~~ Spatial    0     1     1    0      0   0        .p83.
## 84 84   Quant ~~   Quant    0     1     1    0      0   0        .p84.
## 85 85       g ~~       g    0     1     1    0      0   0        .p85.
```

```
lavaanify(cfaModelBifactor_fullSyntax)
```

```
##      id     lhs op     rhs user block group free ustart exo  label plabel
## 1     1  Verbal =~     VO1    1     1     1    1     NA   0          .p1.
## 2     2  Verbal =~     VO2    1     1     1    2     NA   0          .p2.
## 3     3  Verbal =~     VO3    1     1     1    3     NA   0          .p3.
## 4     4  Verbal =~     VW1    1     1     1    4     NA   0          .p4.
## 5     5  Verbal =~     VW2    1     1     1    5     NA   0          .p5.
## 6     6  Verbal =~     VW3    1     1     1    6     NA   0          .p6.
## 7     7  Verbal =~     VM1    1     1     1    7     NA   0          .p7.
## 8     8  Verbal =~     VM2    1     1     1    8     NA   0          .p8.
## 9     9  Verbal =~     VM3    1     1     1    9     NA   0          .p9.
## 10   10 Spatial =~     SO1    1     1     1   10     NA   0         .p10.
## 11   11 Spatial =~     SO2    1     1     1   11     NA   0         .p11.
## 12   12 Spatial =~     SO3    1     1     1   12     NA   0         .p12.
## 13   13 Spatial =~     SW1    1     1     1   13     NA   0         .p13.
## 14   14 Spatial =~     SW2    1     1     1   14     NA   0         .p14.
## 15   15 Spatial =~     SW3    1     1     1   15     NA   0         .p15.
## 16   16 Spatial =~     SM1    1     1     1   16     NA   0         .p16.
## 17   17 Spatial =~     SM2    1     1     1   17     NA   0         .p17.
## 18   18 Spatial =~     SM3    1     1     1   18     NA   0         .p18.
## 19   19   Quant =~     QO1    1     1     1   19     NA   0         .p19.
## 20   20   Quant =~     QO2    1     1     1   20     NA   0         .p20.
## 21   21   Quant =~     QO3    1     1     1   21     NA   0         .p21.
## 22   22   Quant =~     QW1    1     1     1   22     NA   0         .p22.
## 23   23   Quant =~     QW2    1     1     1   23     NA   0         .p23.
## 24   24   Quant =~     QW3    1     1     1   24     NA   0         .p24.
## 25   25   Quant =~     QM1    1     1     1   25     NA   0         .p25.
## 26   26   Quant =~     QM2    1     1     1   26     NA   0         .p26.
## 27   27   Quant =~     QM3    1     1     1   27     NA   0         .p27.
## 28   28       g =~     VO1    1     1     1   28     NA   0         .p28.
## 29   29       g =~     VO2    1     1     1   29     NA   0         .p29.
## 30   30       g =~     VO3    1     1     1   30     NA   0         .p30.
## 31   31       g =~     VW1    1     1     1   31     NA   0         .p31.
## 32   32       g =~     VW2    1     1     1   32     NA   0         .p32.
## 33   33       g =~     VW3    1     1     1   33     NA   0         .p33.
## 34   34       g =~     VM1    1     1     1   34     NA   0         .p34.
## 35   35       g =~     VM2    1     1     1   35     NA   0         .p35.
## 36   36       g =~     VM3    1     1     1   36     NA   0         .p36.
## 37   37       g =~     SO1    1     1     1   37     NA   0         .p37.
## 38   38       g =~     SO2    1     1     1   38     NA   0         .p38.
## 39   39       g =~     SO3    1     1     1   39     NA   0         .p39.
## 40   40       g =~     SW1    1     1     1   40     NA   0         .p40.
## 41   41       g =~     SW2    1     1     1   41     NA   0         .p41.
## 42   42       g =~     SW3    1     1     1   42     NA   0         .p42.
```

```
## 43   43       g =~     SM1    1     1     1   43     NA   0          .p43.
## 44   44       g =~     SM2    1     1     1   44     NA   0          .p44.
## 45   45       g =~     SM3    1     1     1   45     NA   0          .p45.
## 46   46       g =~     QO1    1     1     1   46     NA   0          .p46.
## 47   47       g =~     QO2    1     1     1   47     NA   0          .p47.
## 48   48       g =~     QO3    1     1     1   48     NA   0          .p48.
## 49   49       g =~     QW1    1     1     1   49     NA   0          .p49.
## 50   50       g =~     QW2    1     1     1   50     NA   0          .p50.
## 51   51       g =~     QW3    1     1     1   51     NA   0          .p51.
## 52   52       g =~     QM1    1     1     1   52     NA   0          .p52.
## 53   53       g =~     QM2    1     1     1   53     NA   0          .p53.
## 54   54       g =~     QM3    1     1     1   54     NA   0          .p54.
## 55   55  Verbal ~1            1     1     1    0      0   0          .p55.
## 56   56 Spatial ~1            1     1     1    0      0   0          .p56.
## 57   57   Quant ~1            1     1     1    0      0   0          .p57.
## 58   58       g ~1            1     1     1    0      0   0          .p58.
## 59   59  Verbal ~~  Verbal    1     1     1    0      1   0          .p59.
## 60   60 Spatial ~~ Spatial    1     1     1    0      1   0          .p60.
## 61   61   Quant ~~   Quant    1     1     1    0      1   0          .p61.
## 62   62       g ~~       g    1     1     1    0      1   0          .p62.
## 63   63  Verbal ~~ Spatial    1     1     1    0      0   0          .p63.
## 64   64  Verbal ~~   Quant    1     1     1    0      0   0          .p64.
## 65   65 Spatial ~~   Quant    1     1     1    0      0   0          .p65.
## 66   66  Verbal ~~       g    1     1     1    0      0   0          .p66.
## 67   67 Spatial ~~       g    1     1     1    0      0   0          .p67.
## 68   68   Quant ~~       g    1     1     1    0      0   0          .p68.
## 69   69     VO1 ~~     VO1    1     1     1   55     NA   0          .p69.
## 70   70     VO2 ~~     VO2    1     1     1   56     NA   0          .p70.
## 71   71     VO3 ~~     VO3    1     1     1   57     NA   0          .p71.
## 72   72     VW1 ~~     VW1    1     1     1   58     NA   0          .p72.
## 73   73     VW2 ~~     VW2    1     1     1   59     NA   0          .p73.
## 74   74     VW3 ~~     VW3    1     1     1   60     NA   0          .p74.
## 75   75     VM1 ~~     VM1    1     1     1   61     NA   0          .p75.
## 76   76     VM2 ~~     VM2    1     1     1   62     NA   0          .p76.
## 77   77     VM3 ~~     VM3    1     1     1   63     NA   0          .p77.
## 78   78     SO1 ~~     SO1    1     1     1   64     NA   0          .p78.
## 79   79     SO2 ~~     SO2    1     1     1   65     NA   0          .p79.
## 80   80     SO3 ~~     SO3    1     1     1   66     NA   0          .p80.
## 81   81     SW1 ~~     SW1    1     1     1   67     NA   0          .p81.
## 82   82     SW2 ~~     SW2    1     1     1   68     NA   0          .p82.
## 83   83     SW3 ~~     SW3    1     1     1   69     NA   0          .p83.
## 84   84     SM1 ~~     SM1    1     1     1   70     NA   0          .p84.
## 85   85     SM2 ~~     SM2    1     1     1   71     NA   0          .p85.
## 86   86     SM3 ~~     SM3    1     1     1   72     NA   0          .p86.
## 87   87     QO1 ~~     QO1    1     1     1   73     NA   0          .p87.
## 88   88     QO2 ~~     QO2    1     1     1   74     NA   0          .p88.
```

```
## 89   89     QO3 ~~     QO3    1     1     1   75     NA   0         .p89.
## 90   90     QW1 ~~     QW1    1     1     1   76     NA   0         .p90.
## 91   91     QW2 ~~     QW2    1     1     1   77     NA   0         .p91.
## 92   92     QW3 ~~     QW3    1     1     1   78     NA   0         .p92.
## 93   93     QM1 ~~     QM1    1     1     1   79     NA   0         .p93.
## 94   94     QM2 ~~     QM2    1     1     1   80     NA   0         .p94.
## 95   95     QM3 ~~     QM3    1     1     1   81     NA   0         .p95.
## 96   96     VO1 ~1            1     1     1   82     NA   0 intVO1  .p96.
## 97   97     VO2 ~1            1     1     1   83     NA   0 intVO2  .p97.
## 98   98     VO3 ~1            1     1     1   84     NA   0 intVO3  .p98.
## 99   99     VW1 ~1            1     1     1   85     NA   0 intVW1  .p99.
## 100 100     VW2 ~1            1     1     1   86     NA   0 intVW2 .p100.
## 101 101     VW3 ~1            1     1     1   87     NA   0 intVW3 .p101.
## 102 102     VM1 ~1            1     1     1   88     NA   0 intVM1 .p102.
## 103 103     VM2 ~1            1     1     1   89     NA   0 intVM2 .p103.
## 104 104     VM3 ~1            1     1     1   90     NA   0 intVM3 .p104.
## 105 105     SO1 ~1            1     1     1   91     NA   0 intSO1 .p105.
## 106 106     SO2 ~1            1     1     1   92     NA   0 intSO2 .p106.
## 107 107     SO3 ~1            1     1     1   93     NA   0 intSO3 .p107.
## 108 108     SW1 ~1            1     1     1   94     NA   0 intSW1 .p108.
## 109 109     SW2 ~1            1     1     1   95     NA   0 intSW2 .p109.
## 110 110     SW3 ~1            1     1     1   96     NA   0 intSW3 .p110.
## 111 111     SM1 ~1            1     1     1   97     NA   0 intSM1 .p111.
## 112 112     SM2 ~1            1     1     1   98     NA   0 intSM2 .p112.
## 113 113     SM3 ~1            1     1     1   99     NA   0 intSM3 .p113.
## 114 114     QO1 ~1            1     1     1  100     NA   0 intQO1 .p114.
## 115 115     QO2 ~1            1     1     1  101     NA   0 intQO2 .p115.
## 116 116     QO3 ~1            1     1     1  102     NA   0 intQO3 .p116.
## 117 117     QW1 ~1            1     1     1  103     NA   0 intQW1 .p117.
## 118 118     QW2 ~1            1     1     1  104     NA   0 intQW2 .p118.
## 119 119     QW3 ~1            1     1     1  105     NA   0 intQW3 .p119.
## 120 120     QM1 ~1            1     1     1  106     NA   0 intQM1 .p120.
## 121 121     QM2 ~1            1     1     1  107     NA   0 intQM2 .p121.
## 122 122     QM3 ~1            1     1     1  108     NA   0 intQM3 .p122.
```

12.5.11.3 Fit the model

```
cfaModelBifactorFit <- cfa(cfaModelBifactor_syntax,
                           data = MTMM,
                           orthogonal = TRUE,
                           std.lv = TRUE,
                           missing = "ML",
                           estimator = "MLR")
```

```
cfaModelBifactorFit_full <- lavaan(cfaModelBifactor_fullSyntax,
                                   data = MTMM,
                                   missing = "ML",
                                   estimator = "MLR")
```

12.5.11.4 Display summary output

```
summary(cfaModelBifactorFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 25 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                       108
##
##   Number of observations                          1000
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                              1232.045    1251.910
##   Degrees of freedom                               297         297
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.984
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                             14923.912   14951.143
##   Degrees of freedom                               351         351
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  0.998
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.936       0.935
##   Tucker-Lewis Index (TLI)                       0.924       0.923
##
##   Robust Comparative Fit Index (CFI)                         0.936
##   Robust Tucker-Lewis Index (TLI)                            0.924
```

```
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -31830.739  -31830.739
##   Scaling correction factor                                  1.037
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)     -31214.716  -31214.716
##   Scaling correction factor                                  0.998
##       for the MLR correction
##
##   Akaike (AIC)                               63877.477   63877.477
##   Bayesian (BIC)                             64407.515   64407.515
##   Sample-size adjusted Bayesian (BIC)        64064.501   64064.501
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.056       0.057
##   90 Percent confidence interval - lower         0.053       0.053
##   90 Percent confidence interval - upper         0.059       0.060
##   P-value RMSEA <= 0.05                          0.001       0.000
##
##   Robust RMSEA                                               0.056
##   90 Percent confidence interval - lower                     0.053
##   90 Percent confidence interval - upper                     0.059
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.060       0.060
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Verbal =~
##     VO1               0.382    0.034   11.287    0.000    0.382    0.376
##     VO2               0.372    0.034   10.802    0.000    0.372    0.362
##     VO3               0.365    0.035   10.487    0.000    0.365    0.361
##     VW1               0.587    0.037   15.703    0.000    0.587    0.599
##     VW2               0.617    0.037   16.825    0.000    0.617    0.608
##     VW3               0.621    0.034   18.132    0.000    0.621    0.599
##     VM1               0.443    0.035   12.724    0.000    0.443    0.446
##     VM2               0.429    0.036   11.768    0.000    0.429    0.420
```

```
##     VM3              0.467    0.035   13.336    0.000    0.467    0.455
##   Spatial =~
##     SO1              0.483    0.035   13.880    0.000    0.483    0.476
##     SO2              0.470    0.032   14.475    0.000    0.470    0.470
##     SO3              0.447    0.034   13.136    0.000    0.447    0.440
##     SW1              0.604    0.032   18.631    0.000    0.604    0.610
##     SW2              0.612    0.035   17.264    0.000    0.612    0.603
##     SW3              0.691    0.034   20.154    0.000    0.691    0.680
##     SM1              0.508    0.033   15.413    0.000    0.508    0.500
##     SM2              0.477    0.033   14.459    0.000    0.477    0.473
##     SM3              0.480    0.034   14.278    0.000    0.480    0.476
##   Quant =~
##     QO1              0.280    0.058    4.845    0.000    0.280    0.284
##     QO2              0.267    0.064    4.169    0.000    0.267    0.267
##     QO3              0.268    0.064    4.195    0.000    0.268    0.251
##     QW1              0.558    0.044   12.766    0.000    0.558    0.561
##     QW2              0.549    0.044   12.568    0.000    0.549    0.543
##     QW3              0.614    0.044   14.104    0.000    0.614    0.594
##     QM1              0.376    0.051    7.324    0.000    0.376    0.372
##     QM2              0.313    0.054    5.800    0.000    0.313    0.306
##     QM3              0.357    0.053    6.739    0.000    0.357    0.346
##   g =~
##     VO1              0.611    0.034   17.753    0.000    0.611    0.600
##     VO2              0.665    0.033   20.259    0.000    0.665    0.646
##     VO3              0.626    0.033   19.082    0.000    0.626    0.618
##     VW1              0.429    0.042   10.238    0.000    0.429    0.438
##     VW2              0.477    0.042   11.247    0.000    0.477    0.470
##     VW3              0.454    0.042   10.866    0.000    0.454    0.438
##     VM1              0.552    0.036   15.439    0.000    0.552    0.556
##     VM2              0.590    0.036   16.569    0.000    0.590    0.577
##     VM3              0.584    0.036   16.364    0.000    0.584    0.569
##     SO1              0.519    0.035   14.625    0.000    0.519    0.512
##     SO2              0.569    0.033   17.384    0.000    0.569    0.569
##     SO3              0.551    0.037   15.010    0.000    0.551    0.543
##     SW1              0.421    0.041   10.342    0.000    0.421    0.425
##     SW2              0.449    0.044   10.195    0.000    0.449    0.443
##     SW3              0.372    0.044    8.491    0.000    0.372    0.367
##     SM1              0.494    0.037   13.393    0.000    0.494    0.486
##     SM2              0.507    0.035   14.579    0.000    0.507    0.502
##     SM3              0.534    0.036   14.711    0.000    0.534    0.530
##     QO1              0.664    0.037   17.838    0.000    0.664    0.673
##     QO2              0.677    0.038   17.580    0.000    0.677    0.677
##     QO3              0.750    0.039   19.314    0.000    0.750    0.702
##     QW1              0.477    0.048    9.873    0.000    0.477    0.480
##     QW2              0.530    0.048   10.984    0.000    0.530    0.525
##     QW3              0.501    0.052    9.596    0.000    0.501    0.485
```

```
##     QM1               0.637    0.039   16.253    0.000    0.637    0.631
##     QM2               0.663    0.039   17.074    0.000    0.663    0.649
##     QM3               0.640    0.041   15.472    0.000    0.640    0.621
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Verbal ~~
##     Spatial           0.000                               0.000    0.000
##     Quant             0.000                               0.000    0.000
##     g                 0.000                               0.000    0.000
##   Spatial ~~
##     Quant             0.000                               0.000    0.000
##     g                 0.000                               0.000    0.000
##   Quant ~~
##     g                 0.000                               0.000    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .VO1               0.040    0.032    1.255    0.210    0.040    0.040
##    .VO2               0.023    0.033    0.694    0.488    0.023    0.022
##    .VO3               0.023    0.032    0.711    0.477    0.023    0.022
##    .VW1              -0.003    0.031   -0.083    0.934   -0.003   -0.003
##    .VW2               0.016    0.032    0.496    0.620    0.016    0.016
##    .VW3               0.034    0.033    1.051    0.293    0.034    0.033
##    .VM1               0.064    0.031    2.035    0.042    0.064    0.064
##    .VM2               0.015    0.032    0.476    0.634    0.015    0.015
##    .VM3               0.061    0.033    1.880    0.060    0.061    0.059
##    .SO1               0.013    0.032    0.407    0.684    0.013    0.013
##    .SO2              -0.021    0.032   -0.652    0.514   -0.021   -0.021
##    .SO3               0.004    0.032    0.110    0.913    0.004    0.003
##    .SW1               0.002    0.031    0.070    0.944    0.002    0.002
##    .SW2               0.006    0.032    0.197    0.844    0.006    0.006
##    .SW3               0.013    0.032    0.404    0.686    0.013    0.013
##    .SM1               0.066    0.032    2.043    0.041    0.066    0.065
##    .SM2               0.008    0.032    0.241    0.810    0.008    0.008
##    .SM3               0.004    0.032    0.118    0.906    0.004    0.004
##    .QO1               0.046    0.031    1.474    0.141    0.046    0.047
##    .QO2               0.025    0.032    0.781    0.435    0.025    0.025
##    .QO3               0.023    0.034    0.691    0.489    0.023    0.022
##    .QW1               0.047    0.031    1.497    0.134    0.047    0.047
##    .QW2               0.044    0.032    1.391    0.164    0.044    0.044
##    .QW3               0.006    0.033    0.175    0.861    0.006    0.006
##    .QM1               0.068    0.032    2.141    0.032    0.068    0.068
##    .QM2               0.011    0.032    0.328    0.743    0.011    0.010
##    .QM3               0.045    0.033    1.385    0.166    0.045    0.044
##     Verbal            0.000                               0.000    0.000
```

```
##     Spatial           0.000                               0.000    0.000
##     Quant             0.000                               0.000    0.000
##     g                 0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .VO1               0.518    0.026   19.974    0.000    0.518    0.499
##    .VO2               0.478    0.024   19.954    0.000    0.478    0.452
##    .VO3               0.500    0.024   20.619    0.000    0.500    0.487
##    .VW1               0.431    0.026   16.818    0.000    0.431    0.449
##    .VW2               0.421    0.026   16.320    0.000    0.421    0.409
##    .VW3               0.483    0.026   18.773    0.000    0.483    0.449
##    .VM1               0.484    0.024   20.509    0.000    0.484    0.491
##    .VM2               0.512    0.025   20.269    0.000    0.512    0.490
##    .VM3               0.497    0.025   19.884    0.000    0.497    0.470
##    .SO1               0.526    0.025   21.407    0.000    0.526    0.512
##    .SO2               0.456    0.024   18.718    0.000    0.456    0.456
##    .SO3               0.528    0.027   19.409    0.000    0.528    0.512
##    .SW1               0.439    0.024   18.328    0.000    0.439    0.447
##    .SW2               0.453    0.026   17.698    0.000    0.453    0.440
##    .SW3               0.416    0.025   16.425    0.000    0.416    0.403
##    .SM1               0.530    0.027   19.602    0.000    0.530    0.514
##    .SM2               0.534    0.026   20.155    0.000    0.534    0.524
##    .SM3               0.500    0.024   20.898    0.000    0.500    0.493
##    .QO1               0.454    0.023   19.617    0.000    0.454    0.467
##    .QO2               0.471    0.024   19.520    0.000    0.471    0.471
##    .QO3               0.508    0.026   19.461    0.000    0.508    0.445
##    .QW1               0.448    0.026   17.169    0.000    0.448    0.454
##    .QW2               0.439    0.026   16.716    0.000    0.439    0.429
##    .QW3               0.439    0.028   15.616    0.000    0.439    0.411
##    .QM1               0.472    0.022   21.064    0.000    0.472    0.464
##    .QM2               0.507    0.025   20.073    0.000    0.507    0.485
##    .QM3               0.526    0.026   20.341    0.000    0.526    0.495
##     Verbal            1.000                               1.000    1.000
##     Spatial           1.000                               1.000    1.000
##     Quant             1.000                               1.000    1.000
##     g                 1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     VO1               0.501
##     VO2               0.548
##     VO3               0.513
##     VW1               0.551
##     VW2               0.591
##     VW3               0.551
```

```
##     VM1               0.509
##     VM2               0.510
##     VM3               0.530
##     SO1               0.488
##     SO2               0.544
##     SO3               0.488
##     SW1               0.553
##     SW2               0.560
##     SW3               0.597
##     SM1               0.486
##     SM2               0.476
##     SM3               0.507
##     QO1               0.533
##     QO2               0.529
##     QO3               0.555
##     QW1               0.546
##     QW2               0.571
##     QW3               0.589
##     QM1               0.536
##     QM2               0.515
##     QM3               0.505
```

```
summary(cfaModelBifactorFit_full,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 25 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                       108
##
##   Number of observations                          1000
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                              1232.045    1251.910
##   Degrees of freedom                               297         297
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.984
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
```

```
##
##   Test statistic                              14923.912   14951.143
##   Degrees of freedom                                351         351
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   0.998
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.936       0.935
##   Tucker-Lewis Index (TLI)                        0.924       0.923
##
##   Robust Comparative Fit Index (CFI)                          0.936
##   Robust Tucker-Lewis Index (TLI)                             0.924
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -31830.739  -31830.739
##   Scaling correction factor                                   1.037
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)     -31214.716  -31214.716
##   Scaling correction factor                                   0.998
##       for the MLR correction
##
##   Akaike (AIC)                               63877.477   63877.477
##   Bayesian (BIC)                             64407.515   64407.515
##   Sample-size adjusted Bayesian (BIC)        64064.501   64064.501
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                           0.056       0.057
##   90 Percent confidence interval - lower          0.053       0.053
##   90 Percent confidence interval - upper          0.059       0.060
##   P-value RMSEA <= 0.05                           0.001       0.000
##
##   Robust RMSEA                                                0.056
##   90 Percent confidence interval - lower                      0.053
##   90 Percent confidence interval - upper                      0.059
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                            0.060       0.060
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
```

```
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Verbal =~
##     VO1               0.382    0.034   11.287    0.000    0.382    0.376
##     VO2               0.372    0.034   10.802    0.000    0.372    0.362
##     VO3               0.365    0.035   10.487    0.000    0.365    0.361
##     VW1               0.587    0.037   15.703    0.000    0.587    0.599
##     VW2               0.617    0.037   16.825    0.000    0.617    0.608
##     VW3               0.621    0.034   18.132    0.000    0.621    0.599
##     VM1               0.443    0.035   12.724    0.000    0.443    0.446
##     VM2               0.429    0.036   11.768    0.000    0.429    0.420
##     VM3               0.467    0.035   13.336    0.000    0.467    0.455
##   Spatial =~
##     SO1               0.483    0.035   13.880    0.000    0.483    0.476
##     SO2               0.470    0.032   14.475    0.000    0.470    0.470
##     SO3               0.447    0.034   13.136    0.000    0.447    0.440
##     SW1               0.604    0.032   18.631    0.000    0.604    0.610
##     SW2               0.612    0.035   17.264    0.000    0.612    0.603
##     SW3               0.691    0.034   20.154    0.000    0.691    0.680
##     SM1               0.508    0.033   15.413    0.000    0.508    0.500
##     SM2               0.477    0.033   14.459    0.000    0.477    0.473
##     SM3               0.480    0.034   14.278    0.000    0.480    0.476
##   Quant =~
##     QO1               0.280    0.058    4.845    0.000    0.280    0.284
##     QO2               0.267    0.064    4.169    0.000    0.267    0.267
##     QO3               0.268    0.064    4.195    0.000    0.268    0.251
##     QW1               0.558    0.044   12.766    0.000    0.558    0.561
##     QW2               0.549    0.044   12.568    0.000    0.549    0.543
##     QW3               0.614    0.044   14.104    0.000    0.614    0.594
##     QM1               0.376    0.051    7.324    0.000    0.376    0.372
##     QM2               0.313    0.054    5.800    0.000    0.313    0.306
##     QM3               0.357    0.053    6.739    0.000    0.357    0.346
##   g =~
##     VO1               0.611    0.034   17.753    0.000    0.611    0.600
##     VO2               0.665    0.033   20.259    0.000    0.665    0.646
##     VO3               0.626    0.033   19.082    0.000    0.626    0.618
##     VW1               0.429    0.042   10.238    0.000    0.429    0.438
##     VW2               0.477    0.042   11.247    0.000    0.477    0.470
##     VW3               0.454    0.042   10.866    0.000    0.454    0.438
##     VM1               0.552    0.036   15.439    0.000    0.552    0.556
##     VM2               0.590    0.036   16.569    0.000    0.590    0.577
##     VM3               0.584    0.036   16.364    0.000    0.584    0.569
##     SO1               0.519    0.035   14.625    0.000    0.519    0.512
##     SO2               0.569    0.033   17.384    0.000    0.569    0.569
```

```
##     SO3               0.551    0.037   15.010    0.000    0.551    0.543
##     SW1               0.421    0.041   10.342    0.000    0.421    0.425
##     SW2               0.449    0.044   10.195    0.000    0.449    0.443
##     SW3               0.372    0.044    8.491    0.000    0.372    0.367
##     SM1               0.494    0.037   13.393    0.000    0.494    0.486
##     SM2               0.507    0.035   14.579    0.000    0.507    0.502
##     SM3               0.534    0.036   14.711    0.000    0.534    0.530
##     QO1               0.664    0.037   17.838    0.000    0.664    0.673
##     QO2               0.677    0.038   17.580    0.000    0.677    0.677
##     QO3               0.750    0.039   19.314    0.000    0.750    0.702
##     QW1               0.477    0.048    9.873    0.000    0.477    0.480
##     QW2               0.530    0.048   10.984    0.000    0.530    0.525
##     QW3               0.501    0.052    9.596    0.000    0.501    0.485
##     QM1               0.637    0.039   16.253    0.000    0.637    0.631
##     QM2               0.663    0.039   17.074    0.000    0.663    0.649
##     QM3               0.640    0.041   15.472    0.000    0.640    0.621
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Verbal ~~
##     Spatial           0.000                               0.000    0.000
##     Quant             0.000                               0.000    0.000
##   Spatial ~~
##     Quant             0.000                               0.000    0.000
##   Verbal ~~
##     g                 0.000                               0.000    0.000
##   Spatial ~~
##     g                 0.000                               0.000    0.000
##   Quant ~~
##     g                 0.000                               0.000    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Verbal            0.000                               0.000    0.000
##     Spatial           0.000                               0.000    0.000
##     Quant             0.000                               0.000    0.000
##     g                 0.000                               0.000    0.000
##    .VO1     (iVO1)    0.040    0.032    1.255    0.210    0.040    0.040
##    .VO2     (iVO2)    0.023    0.033    0.694    0.488    0.023    0.022
##    .VO3     (iVO3)    0.023    0.032    0.711    0.477    0.023    0.022
##    .VW1     (iVW1)   -0.003    0.031   -0.083    0.934   -0.003   -0.003
##    .VW2     (iVW2)    0.016    0.032    0.496    0.620    0.016    0.016
##    .VW3     (iVW3)    0.034    0.033    1.051    0.293    0.034    0.033
##    .VM1     (iVM1)    0.064    0.031    2.035    0.042    0.064    0.064
##    .VM2     (iVM2)    0.015    0.032    0.476    0.634    0.015    0.015
##    .VM3     (iVM3)    0.061    0.033    1.880    0.060    0.061    0.059
```

```
##    .SO1     (iSO1)    0.013    0.032    0.407    0.684    0.013    0.013
##    .SO2     (iSO2)   -0.021    0.032   -0.652    0.514   -0.021   -0.021
##    .SO3     (iSO3)    0.004    0.032    0.110    0.913    0.004    0.003
##    .SW1     (iSW1)    0.002    0.031    0.070    0.944    0.002    0.002
##    .SW2     (iSW2)    0.006    0.032    0.197    0.844    0.006    0.006
##    .SW3     (iSW3)    0.013    0.032    0.404    0.686    0.013    0.013
##    .SM1     (iSM1)    0.066    0.032    2.043    0.041    0.066    0.065
##    .SM2     (iSM2)    0.008    0.032    0.241    0.810    0.008    0.008
##    .SM3     (iSM3)    0.004    0.032    0.118    0.906    0.004    0.004
##    .QO1     (iQO1)    0.046    0.031    1.474    0.141    0.046    0.047
##    .QO2     (iQO2)    0.025    0.032    0.781    0.435    0.025    0.025
##    .QO3     (iQO3)    0.023    0.034    0.691    0.489    0.023    0.022
##    .QW1     (iQW1)    0.047    0.031    1.497    0.134    0.047    0.047
##    .QW2     (iQW2)    0.044    0.032    1.391    0.164    0.044    0.044
##    .QW3     (iQW3)    0.006    0.033    0.175    0.861    0.006    0.006
##    .QM1     (iQM1)    0.068    0.032    2.141    0.032    0.068    0.068
##    .QM2     (iQM2)    0.011    0.032    0.328    0.743    0.011    0.010
##    .QM3     (iQM3)    0.045    0.033    1.385    0.166    0.045    0.044
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Verbal            1.000                               1.000    1.000
##     Spatial           1.000                               1.000    1.000
##     Quant             1.000                               1.000    1.000
##     g                 1.000                               1.000    1.000
##    .VO1               0.518    0.026   19.974    0.000    0.518    0.499
##    .VO2               0.478    0.024   19.954    0.000    0.478    0.452
##    .VO3               0.500    0.024   20.619    0.000    0.500    0.487
##    .VW1               0.431    0.026   16.818    0.000    0.431    0.449
##    .VW2               0.421    0.026   16.320    0.000    0.421    0.409
##    .VW3               0.483    0.026   18.773    0.000    0.483    0.449
##    .VM1               0.484    0.024   20.509    0.000    0.484    0.491
##    .VM2               0.512    0.025   20.269    0.000    0.512    0.490
##    .VM3               0.497    0.025   19.884    0.000    0.497    0.470
##    .SO1               0.526    0.025   21.407    0.000    0.526    0.512
##    .SO2               0.456    0.024   18.718    0.000    0.456    0.456
##    .SO3               0.528    0.027   19.409    0.000    0.528    0.512
##    .SW1               0.439    0.024   18.328    0.000    0.439    0.447
##    .SW2               0.453    0.026   17.698    0.000    0.453    0.440
##    .SW3               0.416    0.025   16.425    0.000    0.416    0.403
##    .SM1               0.530    0.027   19.602    0.000    0.530    0.514
##    .SM2               0.534    0.026   20.155    0.000    0.534    0.524
##    .SM3               0.500    0.024   20.898    0.000    0.500    0.493
##    .QO1               0.454    0.023   19.617    0.000    0.454    0.467
##    .QO2               0.471    0.024   19.520    0.000    0.471    0.471
##    .QO3               0.508    0.026   19.461    0.000    0.508    0.445
```

```
##    .QW1               0.448    0.026   17.169    0.000    0.448    0.454
##    .QW2               0.439    0.026   16.716    0.000    0.439    0.429
##    .QW3               0.439    0.028   15.616    0.000    0.439    0.411
##    .QM1               0.472    0.022   21.064    0.000    0.472    0.464
##    .QM2               0.507    0.025   20.073    0.000    0.507    0.485
##    .QM3               0.526    0.026   20.341    0.000    0.526    0.495
##
## R-Square:
##                     Estimate
##     VO1                0.501
##     VO2                0.548
##     VO3                0.513
##     VW1                0.551
##     VW2                0.591
##     VW3                0.551
##     VM1                0.509
##     VM2                0.510
##     VM3                0.530
##     SO1                0.488
##     SO2                0.544
##     SO3                0.488
##     SW1                0.553
##     SW2                0.560
##     SW3                0.597
##     SM1                0.486
##     SM2                0.476
##     SM3                0.507
##     QO1                0.533
##     QO2                0.529
##     QO3                0.555
##     QW1                0.546
##     QW2                0.571
##     QW3                0.589
##     QM1                0.536
##     QM2                0.515
##     QM3                0.505
```

12.5.11.5 Estimates of model fit

```
fitMeasures(cfaModelBifactorFit, fit.measures = c("chisq", "df", "pvalue",
                                                  "chisq.scaled", "df.scaled", "pvalue
                                                  "chisq.scaling.factor",
                                                  "rmsea", "cfi", "tli", "srmr",
                                                  "rmsea.robust", "cfi.robust", "tli.r
```

```
##                chisq                  df               pvalue
##             1232.045             297.000                0.000
##         chisq.scaled           df.scaled        pvalue.scaled
##             1251.910             297.000                0.000
## chisq.scaling.factor               rmsea                  cfi
##                0.984               0.056                0.936
##                  tli                srmr         rmsea.robust
##                0.924               0.060                0.056
##           cfi.robust           tli.robust
##                0.936               0.924
```

12.5.11.6 Residuals

```
residuals(cfaModelBifactorFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##     VO1    VO2    VO3    VW1    VW2    VW3    VM1    VM2    VM3    SO1
## VO1  0.000
## VO2  0.022  0.000
## VO3  0.036  0.035  0.000
## VW1 -0.004 -0.015  0.005  0.000
## VW2 -0.026  0.000 -0.018  0.035  0.000
## VW3  0.008  0.000 -0.008  0.015  0.013  0.000
## VM1  0.000  0.008 -0.021 -0.026 -0.017 -0.003  0.000
## VM2  0.008  0.002  0.008 -0.024 -0.018 -0.026  0.031  0.000
## VM3 -0.018 -0.032 -0.012 -0.017  0.001 -0.014  0.048  0.053  0.000
## SO1  0.014  0.029  0.035 -0.017  0.019  0.045 -0.036 -0.005 -0.043  0.000
## SO2  0.001  0.021  0.025  0.031  0.062  0.034 -0.023 -0.038 -0.016  0.051
## SO3  0.034  0.018  0.038  0.010  0.046  0.047 -0.012  0.015 -0.015  0.068
## SW1  0.050  0.058  0.036  0.169  0.220  0.175  0.056  0.066  0.051 -0.026
## SW2  0.009  0.012  0.039  0.176  0.196  0.186  0.032  0.039  0.051 -0.037
## SW3  0.016  0.063  0.055  0.152  0.237  0.211  0.081  0.065  0.041 -0.001
## SM1 -0.049 -0.014  0.010  0.045  0.055  0.019  0.031  0.055  0.027 -0.012
## SM2 -0.019  0.004 -0.008  0.051  0.074  0.042  0.036  0.071  0.053 -0.010
## SM3 -0.014 -0.018 -0.013  0.035  0.069  0.074  0.073  0.066  0.042 -0.002
## QO1  0.026  0.034  0.008  0.045  0.036  0.041 -0.037 -0.032 -0.040 -0.006
## QO2  0.001  0.010  0.007  0.016  0.010  0.008 -0.056 -0.054 -0.016  0.011
## QO3  0.015 -0.015  0.017  0.020  0.021 -0.010 -0.016 -0.016 -0.009 -0.022
## QW1  0.024  0.023  0.030  0.215  0.208  0.179  0.060  0.071  0.083  0.007
## QW2  0.035  0.020  0.016  0.201  0.164  0.166  0.060  0.036  0.090  0.002
```

```
## QW3  0.024  0.033  0.031  0.218  0.214  0.165  0.069  0.047  0.081  0.022
## QM1 -0.035 -0.019 -0.033  0.058  0.053  0.039  0.064  0.044  0.069 -0.027
## QM2 -0.025 -0.029 -0.017  0.031  0.032  0.031  0.053  0.015  0.043 -0.028
## QM3 -0.027  0.002 -0.074  0.077  0.045  0.049  0.073  0.022  0.067 -0.003
##     SO2    SO3    SW1    SW2    SW3    SM1    SM2    SM3    QO1    QO2
## VO1
## VO2
## VO3
## VW1
## VW2
## VW3
## VM1
## VM2
## VM3
## SO1
## SO2  0.000
## SO3  0.061  0.000
## SW1 -0.023 -0.029  0.000
## SW2 -0.009 -0.034  0.038  0.000
## SW3 -0.017 -0.015  0.039  0.040  0.000
## SM1 -0.005 -0.011 -0.017 -0.008 -0.037  0.000
## SM2 -0.013 -0.005 -0.002 -0.034 -0.015  0.061  0.000
## SM3 -0.012  0.000 -0.028 -0.007 -0.027  0.071  0.045  0.000
## QO1  0.013 -0.018  0.050  0.033  0.014 -0.025 -0.007 -0.013  0.000
## QO2  0.004 -0.003  0.032  0.047  0.023 -0.015 -0.043 -0.050  0.032  0.000
## QO3 -0.001 -0.004 -0.025  0.003 -0.017 -0.070 -0.041 -0.031  0.030  0.069
## QW1  0.027  0.016  0.162  0.209  0.194  0.082  0.054  0.077 -0.023  0.000
## QW2  0.042  0.024  0.171  0.194  0.167  0.035  0.063  0.056  0.008 -0.009
## QW3  0.028  0.040  0.167  0.238  0.216  0.072  0.058  0.075  0.004 -0.017
## QM1 -0.002 -0.034  0.032  0.075  0.079  0.063  0.053  0.060 -0.012 -0.006
## QM2 -0.011 -0.023  0.016  0.062  0.017  0.055  0.017  0.033 -0.011 -0.009
## QM3 -0.015 -0.031  0.030  0.074  0.045  0.056  0.054  0.049 -0.009 -0.018
##     QO3    QW1    QW2    QW3    QM1    QM2    QM3
## VO1
## VO2
## VO3
## VW1
## VW2
## VW3
## VM1
## VM2
## VM3
## SO1
## SO2
## SO3
## SW1
```

```
## SW2
## SW3
## SM1
## SM2
## SM3
## QO1
## QO2
## QO3  0.000
## QW1  0.011  0.000
## QW2 -0.011  0.009  0.000
## QW3 -0.014  0.015  0.009  0.000
## QM1 -0.015 -0.004 -0.023  0.006  0.000
## QM2 -0.023 -0.020 -0.014 -0.006  0.048  0.000
## QM3 -0.013 -0.012  0.014 -0.022  0.022  0.056  0.000
##
## $mean
## VO1 VO2 VO3 VW1 VW2 VW3 VM1 VM2 VM3 SO1 SO2 SO3 SW1 SW2 SW3 SM1 SM2 SM3 QO1 QO2
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## QO3 QW1 QW2 QW3 QM1 QM2 QM3
##   0   0   0   0   0   0   0
```

12.5.11.7 Modification indices

```
modificationindices(cfaModelBifactorFit)
```

```
##         lhs op     rhs      mi    epc sepc.lv sepc.all sepc.nox
## 86   Verbal ~~ Spatial  69.105  0.424   0.424    0.424    0.424
## 87   Verbal ~~   Quant 151.050  0.783   0.783    0.783    0.783
## 89  Spatial ~~   Quant  96.722  0.564   0.564    0.564    0.564
## 123  Verbal =~     SO1  15.753 -0.118  -0.118   -0.116   -0.116
## 124  Verbal =~     SO2   9.606 -0.087  -0.087   -0.087   -0.087
## 125  Verbal =~     SO3   4.155 -0.061  -0.061   -0.060   -0.060
## 126  Verbal =~     SW1  31.873  0.157   0.157    0.158    0.158
## 127  Verbal =~     SW2  19.296  0.124   0.124    0.122    0.122
## 128  Verbal =~     SW3  37.843  0.174   0.174    0.171    0.171
## 129  Verbal =~     SM1   5.305 -0.069  -0.069   -0.068   -0.068
## 130  Verbal =~     SM2   0.444 -0.020  -0.020   -0.020   -0.020
## 131  Verbal =~     SM3   0.328 -0.017  -0.017   -0.017   -0.017
## 132  Verbal =~     QO1   4.397 -0.061  -0.061   -0.062   -0.062
## 133  Verbal =~     QO2  15.056 -0.116  -0.116   -0.116   -0.116
## 134  Verbal =~     QO3   9.109 -0.096  -0.096   -0.090   -0.090
## 135  Verbal =~     QW1  34.131  0.174   0.174    0.176    0.176
## 136  Verbal =~     QW2  21.101  0.134   0.134    0.133    0.133
## 137  Verbal =~     QW3  32.835  0.175   0.175    0.169    0.169
```

```
## 138  Verbal =~     QM1   1.086 -0.030  -0.030   -0.030   -0.030
## 139  Verbal =~     QM2   2.707 -0.050  -0.050   -0.049   -0.049
## 140  Verbal =~     QM3   0.410 -0.020  -0.020   -0.019   -0.019
## 141 Spatial =~     VO1   7.787 -0.079  -0.079   -0.078   -0.078
## 142 Spatial =~     VO2   1.763 -0.037  -0.037   -0.036   -0.036
## 143 Spatial =~     VO3   0.838 -0.026  -0.026   -0.025   -0.025
## 144 Spatial =~     VW1   5.147  0.060   0.060    0.062    0.062
## 145 Spatial =~     VW2  39.336  0.167   0.167    0.164    0.164
## 146 Spatial =~     VW3  18.330  0.121   0.121    0.116    0.116
## 147 Spatial =~     VM1   1.615 -0.035  -0.035   -0.035   -0.035
## 148 Spatial =~     VM2   0.122 -0.010  -0.010   -0.010   -0.010
## 149 Spatial =~     VM3   4.047 -0.056  -0.056   -0.054   -0.054
## 150 Spatial =~     QO1   4.400 -0.057  -0.057   -0.058   -0.058
## 151 Spatial =~     QO2   4.701 -0.060  -0.060   -0.060   -0.060
## 152 Spatial =~     QO3  22.589 -0.140  -0.140   -0.131   -0.131
## 153 Spatial =~     QW1  19.319  0.122   0.122    0.123    0.123
## 154 Spatial =~     QW2  14.118  0.103   0.103    0.102    0.102
## 155 Spatial =~     QW3  30.205  0.156   0.156    0.151    0.151
## 156 Spatial =~     QM1   0.053 -0.006  -0.006   -0.006   -0.006
## 157 Spatial =~     QM2   2.013 -0.040  -0.040   -0.039   -0.039
## 158 Spatial =~     QM3   0.289 -0.015  -0.015   -0.015   -0.015
## 159   Quant =~     VO1  13.580 -0.122  -0.122   -0.120   -0.120
## 160   Quant =~     VO2  12.556 -0.116  -0.116   -0.113   -0.113
## 161   Quant =~     VO3  17.088 -0.136  -0.136   -0.134   -0.134
## 162   Quant =~     VW1  60.552  0.241   0.241    0.246    0.246
## 163   Quant =~     VW2  42.281  0.201   0.201    0.198    0.198
## 164   Quant =~     VW3  17.732  0.138   0.138    0.133    0.133
## 165   Quant =~     VM1   0.761 -0.028  -0.028   -0.028   -0.028
## 166   Quant =~     VM2   5.326 -0.075  -0.075   -0.074   -0.074
## 167   Quant =~     VM3   0.250  0.016   0.016    0.016    0.016
## 168   Quant =~     SO1  27.434 -0.170  -0.170   -0.167   -0.167
## 169   Quant =~     SO2  13.480 -0.113  -0.113   -0.113   -0.113
## 170   Quant =~     SO3  18.743 -0.142  -0.142   -0.140   -0.140
## 171   Quant =~     SW1  19.203  0.133   0.133    0.134    0.134
## 172   Quant =~     SW2  82.977  0.279   0.279    0.276    0.276
## 173   Quant =~     SW3  37.801  0.190   0.190    0.187    0.187
## 174   Quant =~     SM1   0.933 -0.031  -0.031   -0.031   -0.031
## 175   Quant =~     SM2   1.583 -0.041  -0.041   -0.041   -0.041
## 176   Quant =~     SM3   0.331 -0.018  -0.018   -0.018   -0.018
## 177     VO1 ~~     VO2   2.641  0.029   0.029    0.057    0.057
## 178     VO1 ~~     VO3   6.367  0.045   0.045    0.088    0.088
## 179     VO1 ~~     VW1   0.080 -0.005  -0.005   -0.011   -0.011
## 180     VO1 ~~     VW2   4.827 -0.039  -0.039   -0.084   -0.084
## 181     VO1 ~~     VW3   0.380  0.012   0.012    0.023    0.023
## 182     VO1 ~~     VM1   0.001 -0.001  -0.001   -0.001   -0.001
## 183     VO1 ~~     VM2   0.309  0.010   0.010    0.019    0.019
```

```
## 184     VO1 ~~     VM3   1.655 -0.023  -0.023   -0.045   -0.045
## 185     VO1 ~~     SO1   5.503  0.042   0.042    0.081    0.081
## 186     VO1 ~~     SO2   1.149  0.018   0.018    0.037    0.037
## 187     VO1 ~~     SO3   8.303  0.052   0.052    0.099    0.099
## 188     VO1 ~~     SW1   0.452  0.011   0.011    0.024    0.024
## 189     VO1 ~~     SW2   3.283 -0.031  -0.031   -0.064   -0.064
## 190     VO1 ~~     SW3   4.132 -0.035  -0.035   -0.075   -0.075
## 191     VO1 ~~     SM1   7.080 -0.048  -0.048   -0.092   -0.092
## 192     VO1 ~~     SM2   1.685 -0.023  -0.023   -0.045   -0.045
## 193     VO1 ~~     SM3   1.254 -0.020  -0.020   -0.039   -0.039
## 194     VO1 ~~     QO1   6.209  0.043   0.043    0.088    0.088
## 195     VO1 ~~     QO2   2.156  0.026   0.026    0.052    0.052
## 196     VO1 ~~     QO3   4.362  0.038   0.038    0.075    0.075
## 197     VO1 ~~     QW1   1.951 -0.024  -0.024   -0.051   -0.051
## 198     VO1 ~~     QW2   0.001  0.000   0.000    0.001    0.001
## 199     VO1 ~~     QW3   1.510 -0.022  -0.022   -0.046   -0.046
## 200     VO1 ~~     QM1   6.254 -0.043  -0.043   -0.087   -0.087
## 201     VO1 ~~     QM2   1.993 -0.025  -0.025   -0.049   -0.049
## 202     VO1 ~~     QM3   4.037 -0.036  -0.036   -0.070   -0.070
## 203     VO2 ~~     VO3   7.090  0.046   0.046    0.094    0.094
## 204     VO2 ~~     VW1   1.620 -0.022  -0.022   -0.048   -0.048
## 205     VO2 ~~     VW2   0.000  0.000   0.000   -0.001   -0.001
## 206     VO2 ~~     VW3   0.000  0.000   0.000   -0.001   -0.001
## 207     VO2 ~~     VM1   0.377  0.010   0.010    0.022    0.022
## 208     VO2 ~~     VM2   0.022  0.003   0.003    0.005    0.005
## 209     VO2 ~~     VM3   6.058 -0.043  -0.043   -0.088   -0.088
## 210     VO2 ~~     SO1   6.889  0.046   0.046    0.092    0.092
## 211     VO2 ~~     SO2   2.942  0.028   0.028    0.061    0.061
## 212     VO2 ~~     SO3   1.120  0.019   0.019    0.037    0.037
## 213     VO2 ~~     SW1   0.046  0.004   0.004    0.008    0.008
## 214     VO2 ~~     SW2   8.173 -0.048  -0.048   -0.102   -0.102
## 215     VO2 ~~     SW3   0.252  0.008   0.008    0.019    0.019
## 216     VO2 ~~     SM1   1.673 -0.023  -0.023   -0.045   -0.045
## 217     VO2 ~~     SM2   0.401 -0.011  -0.011   -0.022   -0.022
## 218     VO2 ~~     SM3   5.449 -0.040  -0.040   -0.082   -0.082
## 219     VO2 ~~     QO1   9.953  0.053   0.053    0.113    0.113
## 220     VO2 ~~     QO2   4.257  0.035   0.035    0.074    0.074
## 221     VO2 ~~     QO3   0.100 -0.006  -0.006   -0.011   -0.011
## 222     VO2 ~~     QW1   2.666 -0.028  -0.028   -0.060   -0.060
## 223     VO2 ~~     QW2   1.568 -0.021  -0.021   -0.046   -0.046
## 224     VO2 ~~     QW3   0.428 -0.011  -0.011   -0.025   -0.025
## 225     VO2 ~~     QM1   2.481 -0.027  -0.027   -0.056   -0.056
## 226     VO2 ~~     QM2   3.545 -0.033  -0.033   -0.067   -0.067
## 227     VO2 ~~     QM3   0.010 -0.002  -0.002   -0.004   -0.004
## 228     VO3 ~~     VW1   0.136  0.006   0.006    0.014    0.014
## 229     VO3 ~~     VW2   2.458 -0.027  -0.027   -0.060   -0.060
```

```
## 230     VO3 ~~     VW3   0.399 -0.012  -0.012   -0.024   -0.024
## 231     VO3 ~~     VM1   2.223 -0.026  -0.026   -0.052   -0.052
## 232     VO3 ~~     VM2   0.352  0.011   0.011    0.021    0.021
## 233     VO3 ~~     VM3   0.780 -0.016  -0.016   -0.031   -0.031
## 234     VO3 ~~     SO1   7.570  0.049   0.049    0.095    0.095
## 235     VO3 ~~     SO2   2.975  0.029   0.029    0.060    0.060
## 236     VO3 ~~     SO3   5.041  0.040   0.040    0.078    0.078
## 237     VO3 ~~     SW1   2.945 -0.029  -0.029   -0.061   -0.061
## 238     VO3 ~~     SW2   0.875 -0.016  -0.016   -0.033   -0.033
## 239     VO3 ~~     SW3   0.245 -0.008  -0.008   -0.018   -0.018
## 240     VO3 ~~     SM1   0.038  0.003   0.003    0.007    0.007
## 241     VO3 ~~     SM2   2.721 -0.029  -0.029   -0.057   -0.057
## 242     VO3 ~~     SM3   4.313 -0.036  -0.036   -0.072   -0.072
## 243     VO3 ~~     QO1   1.809  0.023   0.023    0.048    0.048
## 244     VO3 ~~     QO2   4.029  0.035   0.035    0.071    0.071
## 245     VO3 ~~     QO3   6.002  0.044   0.044    0.088    0.088
## 246     VO3 ~~     QW1   0.189 -0.008  -0.008   -0.016   -0.016
## 247     VO3 ~~     QW2   0.998 -0.017  -0.017   -0.036   -0.036
## 248     VO3 ~~     QW3   0.022 -0.003  -0.003   -0.006   -0.006
## 249     VO3 ~~     QM1   4.965 -0.038  -0.038   -0.078   -0.078
## 250     VO3 ~~     QM2   0.446 -0.012  -0.012   -0.023   -0.023
## 251     VO3 ~~     QM3  26.455 -0.092  -0.092   -0.179   -0.179
## 252     VW1 ~~     VW2  13.570  0.070   0.070    0.165    0.165
## 253     VW1 ~~     VW3   2.057  0.029   0.029    0.063    0.063
## 254     VW1 ~~     VM1   4.536 -0.037  -0.037   -0.082   -0.082
## 255     VW1 ~~     VM2   3.618 -0.034  -0.034   -0.072   -0.072
## 256     VW1 ~~     VM3   2.092 -0.026  -0.026   -0.056   -0.056
## 257     VW1 ~~     SO1   8.635 -0.050  -0.050   -0.105   -0.105
## 258     VW1 ~~     SO2   0.029 -0.003  -0.003   -0.006   -0.006
## 259     VW1 ~~     SO3   4.388 -0.036  -0.036   -0.075   -0.075
## 260     VW1 ~~     SW1   6.949  0.042   0.042    0.097    0.097
## 261     VW1 ~~     SW2  19.116  0.071   0.071    0.160    0.160
## 262     VW1 ~~     SW3   0.328  0.009   0.009    0.022    0.022
## 263     VW1 ~~     SM1   0.016  0.002   0.002    0.004    0.004
## 264     VW1 ~~     SM2   0.190 -0.007  -0.007   -0.016   -0.016
## 265     VW1 ~~     SM3   3.309 -0.030  -0.030   -0.065   -0.065
## 266     VW1 ~~     QO1   0.001  0.000   0.000    0.001    0.001
## 267     VW1 ~~     QO2   0.409 -0.011  -0.011   -0.023   -0.023
## 268     VW1 ~~     QO3   0.590 -0.013  -0.013   -0.028   -0.028
## 269     VW1 ~~     QW1   8.699  0.048   0.048    0.110    0.110
## 270     VW1 ~~     QW2   9.824  0.051   0.051    0.117    0.117
## 271     VW1 ~~     QW3  10.317  0.054   0.054    0.123    0.123
## 272     VW1 ~~     QM1   1.627 -0.021  -0.021   -0.046   -0.046
## 273     VW1 ~~     QM2   2.790 -0.028  -0.028   -0.060   -0.060
## 274     VW1 ~~     QM3   0.166  0.007   0.007    0.015    0.015
## 275     VW2 ~~     VW3   1.843  0.027   0.027    0.061    0.061
```

```
## 276     VW2 ~~     VM1   2.276 -0.027  -0.027   -0.059   -0.059
## 277     VW2 ~~     VM2   2.464 -0.028  -0.028   -0.061   -0.061
## 278     VW2 ~~     VM3   0.002  0.001   0.001    0.002    0.002
## 279     VW2 ~~     SO1   6.627 -0.044  -0.044   -0.093   -0.093
## 280     VW2 ~~     SO2   0.000  0.000   0.000    0.000    0.000
## 281     VW2 ~~     SO3   2.219 -0.025  -0.025   -0.054   -0.054
## 282     VW2 ~~     SW1  14.987  0.062   0.062    0.144    0.144
## 283     VW2 ~~     SW2   9.435  0.050   0.050    0.114    0.114
## 284     VW2 ~~     SW3  21.456  0.074   0.074    0.178    0.178
## 285     VW2 ~~     SM1   2.186 -0.025  -0.025   -0.053   -0.053
## 286     VW2 ~~     SM2   0.843 -0.016  -0.016   -0.033   -0.033
## 287     VW2 ~~     SM3   2.335 -0.025  -0.025   -0.055   -0.055
## 288     VW2 ~~     QO1   0.026 -0.003  -0.003   -0.006   -0.006
## 289     VW2 ~~     QO2   0.409 -0.010  -0.010   -0.024   -0.024
## 290     VW2 ~~     QO3   0.083 -0.005  -0.005   -0.011   -0.011
## 291     VW2 ~~     QW1  11.961  0.057   0.057    0.131    0.131
## 292     VW2 ~~     QW2   1.088  0.017   0.017    0.039    0.039
## 293     VW2 ~~     QW3  17.590  0.070   0.070    0.162    0.162
## 294     VW2 ~~     QM1   1.110 -0.017  -0.017   -0.038   -0.038
## 295     VW2 ~~     QM2   1.423 -0.020  -0.020   -0.044   -0.044
## 296     VW2 ~~     QM3   2.530 -0.027  -0.027   -0.058   -0.058
## 297     VW3 ~~     VM1   0.052 -0.004  -0.004   -0.009   -0.009
## 298     VW3 ~~     VM2   4.397 -0.040  -0.040   -0.080   -0.080
## 299     VW3 ~~     VM3   1.471 -0.023  -0.023   -0.047   -0.047
## 300     VW3 ~~     SO1   0.192  0.008   0.008    0.016    0.016
## 301     VW3 ~~     SO2   1.476 -0.021  -0.021   -0.044   -0.044
## 302     VW3 ~~     SO3   0.204 -0.008  -0.008   -0.016   -0.016
## 303     VW3 ~~     SW1   2.144  0.025   0.025    0.054    0.054
## 304     VW3 ~~     SW2  12.289  0.060   0.060    0.129    0.129
## 305     VW3 ~~     SW3  14.659  0.065   0.065    0.146    0.146
## 306     VW3 ~~     SM1   9.512 -0.056  -0.056   -0.110   -0.110
## 307     VW3 ~~     SM2   5.059 -0.041  -0.041   -0.080   -0.080
## 308     VW3 ~~     SM3   0.004 -0.001  -0.001   -0.002   -0.002
## 309     VW3 ~~     QO1   0.723  0.015   0.015    0.031    0.031
## 310     VW3 ~~     QO2   0.036 -0.003  -0.003   -0.007   -0.007
## 311     VW3 ~~     QO3   5.086 -0.041  -0.041   -0.083   -0.083
## 312     VW3 ~~     QW1   5.313  0.040   0.040    0.086    0.086
## 313     VW3 ~~     QW2   6.403  0.043   0.043    0.094    0.094
## 314     VW3 ~~     QW3   1.256  0.020   0.020    0.043    0.043
## 315     VW3 ~~     QM1   1.576 -0.022  -0.022   -0.045   -0.045
## 316     VW3 ~~     QM2   0.245 -0.009  -0.009   -0.018   -0.018
## 317     VW3 ~~     QM3   0.147 -0.007  -0.007   -0.014   -0.014
## 318     VM1 ~~     VM2   5.054  0.040   0.040    0.079    0.079
## 319     VM1 ~~     VM3  12.868  0.063   0.063    0.128    0.128
## 320     VM1 ~~     SO1   3.998 -0.035  -0.035   -0.069   -0.069
## 321     VM1 ~~     SO2   3.110 -0.029  -0.029   -0.061   -0.061
```

```
## 322     VM1 ~~     SO3   1.484 -0.021  -0.021   -0.042   -0.042
## 323     VM1 ~~     SW1   1.287 -0.019  -0.019   -0.040   -0.040
## 324     VM1 ~~     SW2   4.747 -0.036  -0.036   -0.077   -0.077
## 325     VM1 ~~     SW3   0.358  0.010   0.010    0.022    0.022
## 326     VM1 ~~     SM1   2.771  0.029   0.029    0.057    0.057
## 327     VM1 ~~     SM2   1.473  0.021   0.021    0.042    0.042
## 328     VM1 ~~     SM3  14.261  0.064   0.064    0.131    0.131
## 329     VM1 ~~     QO1  10.118 -0.052  -0.052   -0.112   -0.112
## 330     VM1 ~~     QO2  12.610 -0.060  -0.060   -0.125   -0.125
## 331     VM1 ~~     QO3   1.225 -0.020  -0.020   -0.039   -0.039
## 332     VM1 ~~     QW1   2.430 -0.026  -0.026   -0.056   -0.056
## 333     VM1 ~~     QW2   0.581 -0.013  -0.013   -0.027   -0.027
## 334     VM1 ~~     QW3   0.617 -0.013  -0.013   -0.029   -0.029
## 335     VM1 ~~     QM1  10.661  0.054   0.054    0.114    0.114
## 336     VM1 ~~     QM2   9.163  0.052   0.052    0.106    0.106
## 337     VM1 ~~     QM3  13.946  0.065   0.065    0.130    0.130
## 338     VM2 ~~     VM3  15.251  0.070   0.070    0.139    0.139
## 339     VM2 ~~     SO1   0.140 -0.007  -0.007   -0.013   -0.013
## 340     VM2 ~~     SO2  11.616 -0.057  -0.057   -0.119   -0.119
## 341     VM2 ~~     SO3   0.036  0.003   0.003    0.007    0.007
## 342     VM2 ~~     SW1   0.598 -0.013  -0.013   -0.027   -0.027
## 343     VM2 ~~     SW2   4.568 -0.036  -0.036   -0.076   -0.076
## 344     VM2 ~~     SW3   1.460 -0.020  -0.020   -0.044   -0.044
## 345     VM2 ~~     SM1   8.045  0.051   0.051    0.098    0.098
## 346     VM2 ~~     SM2   9.776  0.056   0.056    0.108    0.108
## 347     VM2 ~~     SM3   8.251  0.050   0.050    0.099    0.099
## 348     VM2 ~~     QO1   5.314 -0.039  -0.039   -0.081   -0.081
## 349     VM2 ~~     QO2   8.823 -0.051  -0.051   -0.105   -0.105
## 350     VM2 ~~     QO3   0.382 -0.011  -0.011   -0.022   -0.022
## 351     VM2 ~~     QW1   0.449  0.012   0.012    0.024    0.024
## 352     VM2 ~~     QW2   1.853 -0.023  -0.023   -0.049   -0.049
## 353     VM2 ~~     QW3   1.340 -0.020  -0.020   -0.043   -0.043
## 354     VM2 ~~     QM1   6.227  0.043   0.043    0.087    0.087
## 355     VM2 ~~     QM2   0.739  0.015   0.015    0.030    0.030
## 356     VM2 ~~     QM3   0.575  0.014   0.014    0.026    0.026
## 357     VM3 ~~     SO1   4.532 -0.038  -0.038   -0.074   -0.074
## 358     VM3 ~~     SO2   0.822 -0.015  -0.015   -0.032   -0.032
## 359     VM3 ~~     SO3   1.265 -0.020  -0.020   -0.039   -0.039
## 360     VM3 ~~     SW1   1.088 -0.017  -0.017   -0.037   -0.037
## 361     VM3 ~~     SW2   0.132 -0.006  -0.006   -0.013   -0.013
## 362     VM3 ~~     SW3   4.782 -0.037  -0.037   -0.081   -0.081
## 363     VM3 ~~     SM1   3.282  0.032   0.032    0.063    0.063
## 364     VM3 ~~     SM2   7.598  0.049   0.049    0.095    0.095
## 365     VM3 ~~     SM3   3.949  0.034   0.034    0.069    0.069
## 366     VM3 ~~     QO1  15.293 -0.066  -0.066   -0.138   -0.138
## 367     VM3 ~~     QO2   1.096 -0.018  -0.018   -0.037   -0.037
```

```
## 368     VM3 ~~     QO3   1.009 -0.018  -0.018   -0.036   -0.036
## 369     VM3 ~~     QW1   0.659 -0.014  -0.014   -0.029   -0.029
## 370     VM3 ~~     QW2   0.306  0.009   0.009    0.020    0.020
## 371     VM3 ~~     QW3   1.086 -0.018  -0.018   -0.039   -0.039
## 372     VM3 ~~     QM1   9.361  0.052   0.052    0.107    0.107
## 373     VM3 ~~     QM2   3.636  0.034   0.034    0.067    0.067
## 374     VM3 ~~     QM3   7.693  0.049   0.049    0.097    0.097
## 375     SO1 ~~     SO2  14.207  0.065   0.065    0.134    0.134
## 376     SO1 ~~     SO3  21.620  0.086   0.086    0.162    0.162
## 377     SO1 ~~     SW1   3.964 -0.035  -0.035   -0.074   -0.074
## 378     SO1 ~~     SW2   8.269 -0.052  -0.052   -0.106   -0.106
## 379     SO1 ~~     SW3   0.011 -0.002  -0.002   -0.004   -0.004
## 380     SO1 ~~     SM1   0.734 -0.016  -0.016   -0.030   -0.030
## 381     SO1 ~~     SM2   0.469 -0.013  -0.013   -0.024   -0.024
## 382     SO1 ~~     SM3   0.028 -0.003  -0.003   -0.006   -0.006
## 383     SO1 ~~     QO1   0.664  0.014   0.014    0.028    0.028
## 384     SO1 ~~     QO2   4.392  0.036   0.036    0.073    0.073
## 385     SO1 ~~     QO3   1.323  0.021   0.021    0.041    0.041
## 386     SO1 ~~     QW1   3.662 -0.033  -0.033   -0.069   -0.069
## 387     SO1 ~~     QW2   3.829 -0.034  -0.034   -0.070   -0.070
## 388     SO1 ~~     QW3   1.743 -0.023  -0.023   -0.049   -0.049
## 389     SO1 ~~     QM1   3.131 -0.031  -0.031   -0.061   -0.061
## 390     SO1 ~~     QM2   1.226 -0.020  -0.020   -0.038   -0.038
## 391     SO1 ~~     QM3   0.022  0.003   0.003    0.005    0.005
## 392     SO2 ~~     SO3  20.158  0.078   0.078    0.159    0.159
## 393     SO2 ~~     SW1   3.484 -0.031  -0.031   -0.070   -0.070
## 394     SO2 ~~     SW2   0.619 -0.013  -0.013   -0.029   -0.029
## 395     SO2 ~~     SW3   2.354 -0.026  -0.026   -0.061   -0.061
## 396     SO2 ~~     SM1   0.156 -0.007  -0.007   -0.014   -0.014
## 397     SO2 ~~     SM2   0.924 -0.017  -0.017   -0.034   -0.034
## 398     SO2 ~~     SM3   0.805 -0.015  -0.015   -0.032   -0.032
## 399     SO2 ~~     QO1   2.905  0.027   0.027    0.060    0.060
## 400     SO2 ~~     QO2   1.190  0.018   0.018    0.039    0.039
## 401     SO2 ~~     QO3   5.155  0.039   0.039    0.082    0.082
## 402     SO2 ~~     QW1   2.799 -0.028  -0.028   -0.061   -0.061
## 403     SO2 ~~     QW2   0.003 -0.001  -0.001   -0.002   -0.002
## 404     SO2 ~~     QW3   5.340 -0.039  -0.039   -0.086   -0.086
## 405     SO2 ~~     QM1   0.579 -0.012  -0.012   -0.027   -0.027
## 406     SO2 ~~     QM2   0.301 -0.009  -0.009   -0.019   -0.019
## 407     SO2 ~~     QM3   2.178 -0.025  -0.025   -0.052   -0.052
## 408     SO3 ~~     SW1   4.949 -0.039  -0.039   -0.081   -0.081
## 409     SO3 ~~     SW2   7.015 -0.047  -0.047   -0.097   -0.097
## 410     SO3 ~~     SW3   1.648 -0.023  -0.023   -0.050   -0.050
## 411     SO3 ~~     SM1   0.538 -0.014  -0.014   -0.026   -0.026
## 412     SO3 ~~     SM2   0.096 -0.006  -0.006   -0.011   -0.011
## 413     SO3 ~~     SM3   0.000  0.000   0.000    0.000    0.000
```

```
## 414     SO3 ~~     QO1   0.104 -0.006  -0.006   -0.011   -0.011
## 415     SO3 ~~     QO2   0.681  0.014   0.014    0.029    0.029
## 416     SO3 ~~     QO3   4.325  0.038   0.038    0.074    0.074
## 417     SO3 ~~     QW1   1.964 -0.025  -0.025   -0.051   -0.051
## 418     SO3 ~~     QW2   0.180 -0.007  -0.007   -0.015   -0.015
## 419     SO3 ~~     QW3   0.002 -0.001  -0.001   -0.002   -0.002
## 420     SO3 ~~     QM1   6.237 -0.043  -0.043   -0.087   -0.087
## 421     SO3 ~~     QM2   0.967 -0.018  -0.018   -0.034   -0.034
## 422     SO3 ~~     QM3   4.004 -0.036  -0.036   -0.069   -0.069
## 423     SW1 ~~     SW2  11.691  0.061   0.061    0.136    0.136
## 424     SW1 ~~     SW3  15.546  0.073   0.073    0.171    0.171
## 425     SW1 ~~     SM1   1.716 -0.023  -0.023   -0.049   -0.049
## 426     SW1 ~~     SM2   0.016 -0.002  -0.002   -0.005   -0.005
## 427     SW1 ~~     SM3   4.984 -0.039  -0.039   -0.083   -0.083
## 428     SW1 ~~     QO1   3.339  0.029   0.029    0.066    0.066
## 429     SW1 ~~     QO2   0.589  0.013   0.013    0.028    0.028
## 430     SW1 ~~     QO3   2.720 -0.028  -0.028   -0.060   -0.060
## 431     SW1 ~~     QW1   3.813  0.032   0.032    0.072    0.072
## 432     SW1 ~~     QW2  12.781  0.058   0.058    0.132    0.132
## 433     SW1 ~~     QW3   2.173  0.025   0.025    0.056    0.056
## 434     SW1 ~~     QM1   4.818 -0.036  -0.036   -0.078   -0.078
## 435     SW1 ~~     QM2   2.379 -0.026  -0.026   -0.055   -0.055
## 436     SW1 ~~     QM3   3.037 -0.030  -0.030   -0.062   -0.062
## 437     SW2 ~~     SW3  16.785  0.077   0.077    0.177    0.177
## 438     SW2 ~~     SM1   0.416 -0.012  -0.012   -0.024   -0.024
## 439     SW2 ~~     SM2   6.818 -0.047  -0.047   -0.096   -0.096
## 440     SW2 ~~     SM3   0.338 -0.010  -0.010   -0.022   -0.022
## 441     SW2 ~~     QO1   0.986 -0.016  -0.016   -0.036   -0.036
## 442     SW2 ~~     QO2   0.276  0.009   0.009    0.019    0.019
## 443     SW2 ~~     QO3   0.712 -0.015  -0.015   -0.031   -0.031
## 444     SW2 ~~     QW1   7.855  0.047   0.047    0.104    0.104
## 445     SW2 ~~     QW2   6.386  0.041   0.041    0.093    0.093
## 446     SW2 ~~     QW3  18.242  0.072   0.072    0.162    0.162
## 447     SW2 ~~     QM1   0.684 -0.014  -0.014   -0.029   -0.029
## 448     SW2 ~~     QM2   0.141  0.006   0.006    0.013    0.013
## 449     SW2 ~~     QM3   0.025 -0.003  -0.003   -0.006   -0.006
## 450     SW3 ~~     SM1  10.381 -0.060  -0.060   -0.127   -0.127
## 451     SW3 ~~     SM2   1.719 -0.024  -0.024   -0.051   -0.051
## 452     SW3 ~~     SM3   5.789 -0.043  -0.043   -0.095   -0.095
## 453     SW3 ~~     QO1   2.430 -0.025  -0.025   -0.058   -0.058
## 454     SW3 ~~     QO2   0.165 -0.007  -0.007   -0.015   -0.015
## 455     SW3 ~~     QO3   1.774 -0.023  -0.023   -0.050   -0.050
## 456     SW3 ~~     QW1   9.216  0.050   0.050    0.116    0.116
## 457     SW3 ~~     QW2   2.937  0.028   0.028    0.066    0.066
## 458     SW3 ~~     QW3  16.349  0.068   0.068    0.159    0.159
## 459     SW3 ~~     QM1   0.406  0.010   0.010    0.024    0.024
```

```
## 460     SW3 ~~     QM2   4.910 -0.038  -0.038   -0.082   -0.082
## 461     SW3 ~~     QM3   1.878 -0.024  -0.024   -0.050   -0.050
## 462     SM1 ~~     SM2  16.889  0.077   0.077    0.144    0.144
## 463     SM1 ~~     SM3  24.736  0.090   0.090    0.176    0.176
## 464     SM1 ~~     QO1   3.648 -0.033  -0.033   -0.067   -0.067
## 465     SM1 ~~     QO2   1.247 -0.020  -0.020   -0.039   -0.039
## 466     SM1 ~~     QO3  13.166 -0.067  -0.067   -0.128   -0.128
## 467     SM1 ~~     QW1   0.294  0.010   0.010    0.020    0.020
## 468     SM1 ~~     QW2   6.230 -0.043  -0.043   -0.090   -0.090
## 469     SM1 ~~     QW3   0.800 -0.016  -0.016   -0.033   -0.033
## 470     SM1 ~~     QM1   7.587  0.048   0.048    0.096    0.096
## 471     SM1 ~~     QM2  10.134  0.057   0.057    0.110    0.110
## 472     SM1 ~~     QM3   6.342  0.046   0.046    0.087    0.087
## 473     SM2 ~~     SM3   9.466  0.056   0.056    0.108    0.108
## 474     SM2 ~~     QO1   0.272 -0.009  -0.009   -0.018   -0.018
## 475     SM2 ~~     QO2   8.457 -0.051  -0.051   -0.102   -0.102
## 476     SM2 ~~     QO3   2.199 -0.027  -0.027   -0.052   -0.052
## 477     SM2 ~~     QW1   1.031 -0.018  -0.018   -0.037   -0.037
## 478     SM2 ~~     QW2   0.016  0.002   0.002    0.005    0.005
## 479     SM2 ~~     QW3   1.796 -0.024  -0.024   -0.049   -0.049
## 480     SM2 ~~     QM1   5.003  0.039   0.039    0.078    0.078
## 481     SM2 ~~     QM2   0.482  0.013   0.013    0.024    0.024
## 482     SM2 ~~     QM3   6.484  0.047   0.047    0.088    0.088
## 483     SM3 ~~     QO1   1.512 -0.021  -0.021   -0.043   -0.043
## 484     SM3 ~~     QO2  14.577 -0.065  -0.065   -0.134   -0.134
## 485     SM3 ~~     QO3   0.858 -0.017  -0.017   -0.033   -0.033
## 486     SM3 ~~     QW1   0.024  0.003   0.003    0.006    0.006
## 487     SM3 ~~     QW2   0.835 -0.015  -0.015   -0.033   -0.033
## 488     SM3 ~~     QW3   0.408 -0.011  -0.011   -0.024   -0.024
## 489     SM3 ~~     QM1   6.268  0.042   0.042    0.087    0.087
## 490     SM3 ~~     QM2   2.552  0.028   0.028    0.056    0.056
## 491     SM3 ~~     QM3   3.831  0.035   0.035    0.068    0.068
## 492     QO1 ~~     QO2   6.015  0.041   0.041    0.088    0.088
## 493     QO1 ~~     QO3   5.612  0.041   0.041    0.086    0.086
## 494     QO1 ~~     QW1   3.648 -0.033  -0.033   -0.074   -0.074
## 495     QO1 ~~     QW2   0.439  0.011   0.011    0.025    0.025
## 496     QO1 ~~     QW3   0.144  0.007   0.007    0.015    0.015
## 497     QO1 ~~     QM1   0.855 -0.015  -0.015   -0.033   -0.033
## 498     QO1 ~~     QM2   0.622 -0.013  -0.013   -0.028   -0.028
## 499     QO1 ~~     QM3   0.409 -0.011  -0.011   -0.023   -0.023
## 500     QO2 ~~     QO3  30.351  0.098   0.098    0.201    0.201
## 501     QO2 ~~     QW1   0.000  0.000   0.000    0.000    0.000
## 502     QO2 ~~     QW2   0.566 -0.013  -0.013   -0.029   -0.029
## 503     QO2 ~~     QW3   2.310 -0.028  -0.028   -0.061   -0.061
## 504     QO2 ~~     QM1   0.226 -0.008  -0.008   -0.017   -0.017
## 505     QO2 ~~     QM2   0.489 -0.012  -0.012   -0.025   -0.025
```

```
## 506     QO2 ~~     QM3   1.663 -0.023  -0.023   -0.046   -0.046
## 507     QO3 ~~     QW1   0.919  0.018   0.018    0.037    0.037
## 508     QO3 ~~     QW2   0.916 -0.017  -0.017   -0.037   -0.037
## 509     QO3 ~~     QW3   1.840 -0.026  -0.026   -0.055   -0.055
## 510     QO3 ~~     QM1   1.479 -0.022  -0.022   -0.044   -0.044
## 511     QO3 ~~     QM2   3.326 -0.033  -0.033   -0.066   -0.066
## 512     QO3 ~~     QM3   1.028 -0.019  -0.019   -0.036   -0.036
## 513     QW1 ~~     QW2   1.014  0.022   0.022    0.050    0.050
## 514     QW1 ~~     QW3   3.715  0.048   0.048    0.109    0.109
## 515     QW1 ~~     QM1   0.142 -0.007  -0.007   -0.015   -0.015
## 516     QW1 ~~     QM2   2.831 -0.031  -0.031   -0.065   -0.065
## 517     QW1 ~~     QM3   1.041 -0.019  -0.019   -0.040   -0.040
## 518     QW2 ~~     QW3   1.404  0.028   0.028    0.065    0.065
## 519     QW2 ~~     QM1   4.102 -0.036  -0.036   -0.079   -0.079
## 520     QW2 ~~     QM2   1.409 -0.021  -0.021   -0.045   -0.045
## 521     QW2 ~~     QM3   1.319  0.021   0.021    0.044    0.044
## 522     QW3 ~~     QM1   0.332  0.011   0.011    0.024    0.024
## 523     QW3 ~~     QM2   0.335 -0.011  -0.011   -0.023   -0.023
## 524     QW3 ~~     QM3   4.147 -0.040  -0.040   -0.083   -0.083
## 525     QM1 ~~     QM2  12.624  0.062   0.062    0.126    0.126
## 526     QM1 ~~     QM3   2.646  0.029   0.029    0.058    0.058
## 527     QM2 ~~     QM3  16.253  0.073   0.073    0.142    0.142
```

12.5.11.8 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModelBifactorFit)
```

```
##           Verbal   Spatial     Quant         g
## alpha  0.9059841 0.9042265 0.9059678 0.9456580
## omega  0.8093910 0.8385836 0.7504999 0.9451057
## omega2 0.3859801 0.4881122 0.2684385 0.7694090
## omega3 0.3855064 0.4873407 0.2679810 0.7189177
## avevar        NA        NA        NA        NA
```

12.5.11.9 Path diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModelBifactorFit,
         what = "std",
         layout = "tree3",
         optimizeLatRes = FALSE,
         bifactor = c("g"),
         exoVar = FALSE,
         edge.label.cex = 1.3)
```

12.5.11.10 Equivalently fitting models

Markov equivalent directed acyclic graphs (DAGs) were depicted using the `dagitty` package [Textor et al., 2021].

```
dagModelBifactor <- lavaanToGraph(cfaModelBifactorFit)

par(mfrow = c(2, 2))
```

```
lapply(equivalentDAGs(dagModelBifactor, n = 4), plot)
```

```
## [[1]]
## [[1]]$mar
## [1] 0 0 0 0
##
##
## [[2]]
## [[2]]$mar
## [1] 0 0 0 0
##
##
## [[3]]
## [[3]]$mar
## [1] 0 0 0 0
##
##
## [[4]]
## [[4]]$mar
## [1] 0 0 0 0
```

12.5.12 Multi-Trait Multi-Method Model (MTMM)

Figure 12.44: Bifactor model

Figure 12.45: Equivalently fitting models to a bifactor model

Figure 12.46: Equivalently fitting models to a bifactor model

Figure 12.47: Equivalently fitting models to a bifactor model

Figure 12.48: Equivalently fitting models to a bifactor model

12.5.12.1 Specify the model

```
cfaModelMTMM_syntax <- '
  g =~ Verbal + Spatial + Quant
  Verbal =~
    VO1 + VO2 + VO3 +
    VW1 + VW2 + VW3 +
    VM1 + VM2 + VM3
  Spatial =~
    SO1 + SO2 + SO3 +
    SW1 + SW2 + SW3 +
    SM1 + SM2 + SM3
  Quant =~
    QO1 + QO2 + QO3 +
    QW1 + QW2 + QW3 +
    QM1 + QM2 + QM3
  Oral =~
    VO1 + VO2 + VO3 +
    SO1 + SO2 + SO3 +
    QO1 + QO2 + QO3
  Written =~
    VW1 + VW2 + VW3 +
    SW1 + SW2 + SW3 +
    QW1 + QW2 + QW3
  Manipulative =~
    VM1 + VM2 + VM3 +
    SM1 + SM2 + SM3 +
    QM1 + QM2 + QM3
'

cfaModelMTMM_fullSyntax <- '
  #Factor loadings (free the factor loading of the first indicator)
  g =~ NA*Verbal + Spatial + Quant
  Verbal =~
    NA*VO1 + VO2 + VO3 +
    VW1 + VW2 + VW3 +
    VM1 + VM2 + VM3
  Spatial =~
    NA*SO1 + SO2 + SO3 +
    SW1 + SW2 + SW3 +
    SM1 + SM2 + SM3
  Quant =~
    NA*QO1 + QO2 + QO3 +
    QW1 + QW2 + QW3 +
    QM1 + QM2 + QM3
```

```
Oral =~
  NA*VO1 + VO2 + VO3 +
  SO1 + SO2 + SO3 +
  QO1 + QO2 + QO3
Written =~
  NA*VW1 + VW2 + VW3 +
  SW1 + SW2 + SW3 +
  QW1 + QW2 + QW3
Manipulative =~
  NA*VM1 + VM2 + VM3 +
  SM1 + SM2 + SM3 +
  QM1 + QM2 + QM3

#Fix latent means to zero
Verbal ~ 0
Spatial ~ 0
Quant ~ 0
Oral ~ 0
Written ~ 0
Manipulative ~ 0
g ~ 0

#Fix latent variances to one
Verbal ~~ 1*Verbal
Spatial ~~ 1*Spatial
Quant ~~ 1*Quant
Oral ~~ 1*Oral
Written ~~ 1*Written
Manipulative ~~ 1*Manipulative
g ~~ 1*g

#Fix covariances among latent variables at zero
Verbal ~~ 0*Spatial
Verbal ~~ 0*Quant
Verbal ~~ 0*Oral
Verbal ~~ 0*Written
Verbal ~~ 0*Manipulative
Spatial ~~ 0*Quant
Spatial ~~ 0*Oral
Spatial ~~ 0*Written
Spatial ~~ 0*Manipulative
Quant ~~ 0*Oral
Quant ~~ 0*Written
Quant ~~ 0*Manipulative
Oral ~~ 0*Written
```

```
Oral ~~ 0*Manipulative
Written ~~ 0*Manipulative
g ~~ 0*Verbal
g ~~ 0*Spatial
g ~~ 0*Quant
g ~~ 0*Oral
g ~~ 0*Written
g ~~ 0*Manipulative

#Estimate residual variances of manifest variables
VO1 ~~ VO1
VO2 ~~ VO2
VO3 ~~ VO3
VW1 ~~ VW1
VW2 ~~ VW2
VW3 ~~ VW3
VM1 ~~ VM1
VM2 ~~ VM2
VM3 ~~ VM3
SO1 ~~ SO1
SO2 ~~ SO2
SO3 ~~ SO3
SW1 ~~ SW1
SW2 ~~ SW2
SW3 ~~ SW3
SM1 ~~ SM1
SM2 ~~ SM2
SM3 ~~ SM3
QO1 ~~ QO1
QO2 ~~ QO2
QO3 ~~ QO3
QW1 ~~ QW1
QW2 ~~ QW2
QW3 ~~ QW3
QM1 ~~ QM1
QM2 ~~ QM2
QM3 ~~ QM3

#Free intercepts of manifest variables
VO1 ~ intVO1*1
VO2 ~ intVO2*1
VO3 ~ intVO3*1
VW1 ~ intVW1*1
VW2 ~ intVW2*1
VW3 ~ intVW3*1
```

```
  VM1 ~ intVM1*1
  VM2 ~ intVM2*1
  VM3 ~ intVM3*1
  SO1 ~ intSO1*1
  SO2 ~ intSO2*1
  SO3 ~ intSO3*1
  SW1 ~ intSW1*1
  SW2 ~ intSW2*1
  SW3 ~ intSW3*1
  SM1 ~ intSM1*1
  SM2 ~ intSM2*1
  SM3 ~ intSM3*1
  QO1 ~ intQO1*1
  QO2 ~ intQO2*1
  QO3 ~ intQO3*1
  QW1 ~ intQW1*1
  QW2 ~ intQW2*1
  QW3 ~ intQW3*1
  QM1 ~ intQM1*1
  QM2 ~ intQM2*1
  QM3 ~ intQM3*1
'
```

12.5.12.2 Model syntax in table form:

```
lavaanify(cfaModelMTMM_syntax)
```

```
##    id          lhs op          rhs user block group free ustart exo label
## 1   1            g =~       Verbal    1     1     1    1     NA   0
## 2   2            g =~      Spatial    1     1     1    2     NA   0
## 3   3            g =~        Quant    1     1     1    3     NA   0
## 4   4       Verbal =~          VO1    1     1     1    4     NA   0
## 5   5       Verbal =~          VO2    1     1     1    5     NA   0
## 6   6       Verbal =~          VO3    1     1     1    6     NA   0
## 7   7       Verbal =~          VW1    1     1     1    7     NA   0
## 8   8       Verbal =~          VW2    1     1     1    8     NA   0
## 9   9       Verbal =~          VW3    1     1     1    9     NA   0
## 10 10       Verbal =~          VM1    1     1     1   10     NA   0
## 11 11       Verbal =~          VM2    1     1     1   11     NA   0
## 12 12       Verbal =~          VM3    1     1     1   12     NA   0
## 13 13      Spatial =~          SO1    1     1     1   13     NA   0
## 14 14      Spatial =~          SO2    1     1     1   14     NA   0
## 15 15      Spatial =~          SO3    1     1     1   15     NA   0
```

```
## 16 16      Spatial =~          SW1    1     1     1   16     NA   0
## 17 17      Spatial =~          SW2    1     1     1   17     NA   0
## 18 18      Spatial =~          SW3    1     1     1   18     NA   0
## 19 19      Spatial =~          SM1    1     1     1   19     NA   0
## 20 20      Spatial =~          SM2    1     1     1   20     NA   0
## 21 21      Spatial =~          SM3    1     1     1   21     NA   0
## 22 22        Quant =~          QO1    1     1     1   22     NA   0
## 23 23        Quant =~          QO2    1     1     1   23     NA   0
## 24 24        Quant =~          QO3    1     1     1   24     NA   0
## 25 25        Quant =~          QW1    1     1     1   25     NA   0
## 26 26        Quant =~          QW2    1     1     1   26     NA   0
## 27 27        Quant =~          QW3    1     1     1   27     NA   0
## 28 28        Quant =~          QM1    1     1     1   28     NA   0
## 29 29        Quant =~          QM2    1     1     1   29     NA   0
## 30 30        Quant =~          QM3    1     1     1   30     NA   0
## 31 31         Oral =~          VO1    1     1     1   31     NA   0
## 32 32         Oral =~          VO2    1     1     1   32     NA   0
## 33 33         Oral =~          VO3    1     1     1   33     NA   0
## 34 34         Oral =~          SO1    1     1     1   34     NA   0
## 35 35         Oral =~          SO2    1     1     1   35     NA   0
## 36 36         Oral =~          SO3    1     1     1   36     NA   0
## 37 37         Oral =~          QO1    1     1     1   37     NA   0
## 38 38         Oral =~          QO2    1     1     1   38     NA   0
## 39 39         Oral =~          QO3    1     1     1   39     NA   0
## 40 40      Written =~          VW1    1     1     1   40     NA   0
## 41 41      Written =~          VW2    1     1     1   41     NA   0
## 42 42      Written =~          VW3    1     1     1   42     NA   0
## 43 43      Written =~          SW1    1     1     1   43     NA   0
## 44 44      Written =~          SW2    1     1     1   44     NA   0
## 45 45      Written =~          SW3    1     1     1   45     NA   0
## 46 46      Written =~          QW1    1     1     1   46     NA   0
## 47 47      Written =~          QW2    1     1     1   47     NA   0
## 48 48      Written =~          QW3    1     1     1   48     NA   0
## 49 49 Manipulative =~          VM1    1     1     1   49     NA   0
## 50 50 Manipulative =~          VM2    1     1     1   50     NA   0
## 51 51 Manipulative =~          VM3    1     1     1   51     NA   0
## 52 52 Manipulative =~          SM1    1     1     1   52     NA   0
## 53 53 Manipulative =~          SM2    1     1     1   53     NA   0
## 54 54 Manipulative =~          SM3    1     1     1   54     NA   0
## 55 55 Manipulative =~          QM1    1     1     1   55     NA   0
## 56 56 Manipulative =~          QM2    1     1     1   56     NA   0
## 57 57 Manipulative =~          QM3    1     1     1   57     NA   0
## 58 58          VO1 ~~          VO1    0     1     1    0      0   0
## 59 59          VO2 ~~          VO2    0     1     1    0      0   0
## 60 60          VO3 ~~          VO3    0     1     1    0      0   0
## 61 61          VW1 ~~          VW1    0     1     1    0      0   0
```

```
## 62 62          VW2 ~~          VW2    0     1     1    0      0   0
## 63 63          VW3 ~~          VW3    0     1     1    0      0   0
## 64 64          VM1 ~~          VM1    0     1     1    0      0   0
## 65 65          VM2 ~~          VM2    0     1     1    0      0   0
## 66 66          VM3 ~~          VM3    0     1     1    0      0   0
## 67 67          SO1 ~~          SO1    0     1     1    0      0   0
## 68 68          SO2 ~~          SO2    0     1     1    0      0   0
## 69 69          SO3 ~~          SO3    0     1     1    0      0   0
## 70 70          SW1 ~~          SW1    0     1     1    0      0   0
## 71 71          SW2 ~~          SW2    0     1     1    0      0   0
## 72 72          SW3 ~~          SW3    0     1     1    0      0   0
## 73 73          SM1 ~~          SM1    0     1     1    0      0   0
## 74 74          SM2 ~~          SM2    0     1     1    0      0   0
## 75 75          SM3 ~~          SM3    0     1     1    0      0   0
## 76 76          QO1 ~~          QO1    0     1     1    0      0   0
## 77 77          QO2 ~~          QO2    0     1     1    0      0   0
## 78 78          QO3 ~~          QO3    0     1     1    0      0   0
## 79 79          QW1 ~~          QW1    0     1     1    0      0   0
## 80 80          QW2 ~~          QW2    0     1     1    0      0   0
## 81 81          QW3 ~~          QW3    0     1     1    0      0   0
## 82 82          QM1 ~~          QM1    0     1     1    0      0   0
## 83 83          QM2 ~~          QM2    0     1     1    0      0   0
## 84 84          QM3 ~~          QM3    0     1     1    0      0   0
## 85 85            g ~~            g    0     1     1    0      0   0
## 86 86       Verbal ~~       Verbal    0     1     1    0      0   0
## 87 87      Spatial ~~      Spatial    0     1     1    0      0   0
## 88 88        Quant ~~        Quant    0     1     1    0      0   0
## 89 89         Oral ~~         Oral    0     1     1    0      0   0
## 90 90      Written ~~      Written    0     1     1    0      0   0
## 91 91 Manipulative ~~ Manipulative    0     1     1    0      0   0
##    plabel
## 1    .p1.
## 2    .p2.
## 3    .p3.
## 4    .p4.
## 5    .p5.
## 6    .p6.
## 7    .p7.
## 8    .p8.
## 9    .p9.
## 10  .p10.
## 11  .p11.
## 12  .p12.
## 13  .p13.
## 14  .p14.
## 15  .p15.
```

```
## 16  .p16.
## 17  .p17.
## 18  .p18.
## 19  .p19.
## 20  .p20.
## 21  .p21.
## 22  .p22.
## 23  .p23.
## 24  .p24.
## 25  .p25.
## 26  .p26.
## 27  .p27.
## 28  .p28.
## 29  .p29.
## 30  .p30.
## 31  .p31.
## 32  .p32.
## 33  .p33.
## 34  .p34.
## 35  .p35.
## 36  .p36.
## 37  .p37.
## 38  .p38.
## 39  .p39.
## 40  .p40.
## 41  .p41.
## 42  .p42.
## 43  .p43.
## 44  .p44.
## 45  .p45.
## 46  .p46.
## 47  .p47.
## 48  .p48.
## 49  .p49.
## 50  .p50.
## 51  .p51.
## 52  .p52.
## 53  .p53.
## 54  .p54.
## 55  .p55.
## 56  .p56.
## 57  .p57.
## 58  .p58.
## 59  .p59.
## 60  .p60.
## 61  .p61.
```

```
## 62  .p62.
## 63  .p63.
## 64  .p64.
## 65  .p65.
## 66  .p66.
## 67  .p67.
## 68  .p68.
## 69  .p69.
## 70  .p70.
## 71  .p71.
## 72  .p72.
## 73  .p73.
## 74  .p74.
## 75  .p75.
## 76  .p76.
## 77  .p77.
## 78  .p78.
## 79  .p79.
## 80  .p80.
## 81  .p81.
## 82  .p82.
## 83  .p83.
## 84  .p84.
## 85  .p85.
## 86  .p86.
## 87  .p87.
## 88  .p88.
## 89  .p89.
## 90  .p90.
## 91  .p91.
```

```
lavaanify(cfaModelMTMM_fullSyntax)
```

```
##      id          lhs op          rhs user block group free ustart exo  label
## 1     1            g =~       Verbal    1     1     1    1     NA   0
## 2     2            g =~      Spatial    1     1     1    2     NA   0
## 3     3            g =~        Quant    1     1     1    3     NA   0
## 4     4       Verbal =~          VO1    1     1     1    4     NA   0
## 5     5       Verbal =~          VO2    1     1     1    5     NA   0
## 6     6       Verbal =~          VO3    1     1     1    6     NA   0
## 7     7       Verbal =~          VW1    1     1     1    7     NA   0
## 8     8       Verbal =~          VW2    1     1     1    8     NA   0
## 9     9       Verbal =~          VW3    1     1     1    9     NA   0
## 10   10       Verbal =~          VM1    1     1     1   10     NA   0
## 11   11       Verbal =~          VM2    1     1     1   11     NA   0
```

```
## 12   12       Verbal =~          VM3    1      1      1   12      NA   0
## 13   13      Spatial =~          SO1    1      1      1   13      NA   0
## 14   14      Spatial =~          SO2    1      1      1   14      NA   0
## 15   15      Spatial =~          SO3    1      1      1   15      NA   0
## 16   16      Spatial =~          SW1    1      1      1   16      NA   0
## 17   17      Spatial =~          SW2    1      1      1   17      NA   0
## 18   18      Spatial =~          SW3    1      1      1   18      NA   0
## 19   19      Spatial =~          SM1    1      1      1   19      NA   0
## 20   20      Spatial =~          SM2    1      1      1   20      NA   0
## 21   21      Spatial =~          SM3    1      1      1   21      NA   0
## 22   22        Quant =~          QO1    1      1      1   22      NA   0
## 23   23        Quant =~          QO2    1      1      1   23      NA   0
## 24   24        Quant =~          QO3    1      1      1   24      NA   0
## 25   25        Quant =~          QW1    1      1      1   25      NA   0
## 26   26        Quant =~          QW2    1      1      1   26      NA   0
## 27   27        Quant =~          QW3    1      1      1   27      NA   0
## 28   28        Quant =~          QM1    1      1      1   28      NA   0
## 29   29        Quant =~          QM2    1      1      1   29      NA   0
## 30   30        Quant =~          QM3    1      1      1   30      NA   0
## 31   31         Oral =~          VO1    1      1      1   31      NA   0
## 32   32         Oral =~          VO2    1      1      1   32      NA   0
## 33   33         Oral =~          VO3    1      1      1   33      NA   0
## 34   34         Oral =~          SO1    1      1      1   34      NA   0
## 35   35         Oral =~          SO2    1      1      1   35      NA   0
## 36   36         Oral =~          SO3    1      1      1   36      NA   0
## 37   37         Oral =~          QO1    1      1      1   37      NA   0
## 38   38         Oral =~          QO2    1      1      1   38      NA   0
## 39   39         Oral =~          QO3    1      1      1   39      NA   0
## 40   40      Written =~          VW1    1      1      1   40      NA   0
## 41   41      Written =~          VW2    1      1      1   41      NA   0
## 42   42      Written =~          VW3    1      1      1   42      NA   0
## 43   43      Written =~          SW1    1      1      1   43      NA   0
## 44   44      Written =~          SW2    1      1      1   44      NA   0
## 45   45      Written =~          SW3    1      1      1   45      NA   0
## 46   46      Written =~          QW1    1      1      1   46      NA   0
## 47   47      Written =~          QW2    1      1      1   47      NA   0
## 48   48      Written =~          QW3    1      1      1   48      NA   0
## 49   49 Manipulative =~          VM1    1      1      1   49      NA   0
## 50   50 Manipulative =~          VM2    1      1      1   50      NA   0
## 51   51 Manipulative =~          VM3    1      1      1   51      NA   0
## 52   52 Manipulative =~          SM1    1      1      1   52      NA   0
## 53   53 Manipulative =~          SM2    1      1      1   53      NA   0
## 54   54 Manipulative =~          SM3    1      1      1   54      NA   0
## 55   55 Manipulative =~          QM1    1      1      1   55      NA   0
## 56   56 Manipulative =~          QM2    1      1      1   56      NA   0
## 57   57 Manipulative =~          QM3    1      1      1   57      NA   0
```

```
## 58   58       Verbal ~1                1     1     1    0      0   0
## 59   59      Spatial ~1                1     1     1    0      0   0
## 60   60        Quant ~1                1     1     1    0      0   0
## 61   61         Oral ~1                1     1     1    0      0   0
## 62   62      Written ~1                1     1     1    0      0   0
## 63   63 Manipulative ~1                1     1     1    0      0   0
## 64   64            g ~1                1     1     1    0      0   0
## 65   65       Verbal ~~       Verbal   1     1     1    0      1   0
## 66   66      Spatial ~~      Spatial   1     1     1    0      1   0
## 67   67        Quant ~~        Quant   1     1     1    0      1   0
## 68   68         Oral ~~         Oral   1     1     1    0      1   0
## 69   69      Written ~~      Written   1     1     1    0      1   0
## 70   70 Manipulative ~~ Manipulative   1     1     1    0      1   0
## 71   71            g ~~            g   1     1     1    0      1   0
## 72   72       Verbal ~~      Spatial   1     1     1    0      0   0
## 73   73       Verbal ~~        Quant   1     1     1    0      0   0
## 74   74       Verbal ~~         Oral   1     1     1    0      0   0
## 75   75       Verbal ~~      Written   1     1     1    0      0   0
## 76   76       Verbal ~~ Manipulative   1     1     1    0      0   0
## 77   77      Spatial ~~        Quant   1     1     1    0      0   0
## 78   78      Spatial ~~         Oral   1     1     1    0      0   0
## 79   79      Spatial ~~      Written   1     1     1    0      0   0
## 80   80      Spatial ~~ Manipulative   1     1     1    0      0   0
## 81   81        Quant ~~         Oral   1     1     1    0      0   0
## 82   82        Quant ~~      Written   1     1     1    0      0   0
## 83   83        Quant ~~ Manipulative   1     1     1    0      0   0
## 84   84         Oral ~~      Written   1     1     1    0      0   0
## 85   85         Oral ~~ Manipulative   1     1     1    0      0   0
## 86   86      Written ~~ Manipulative   1     1     1    0      0   0
## 87   87            g ~~       Verbal   1     1     1    0      0   0
## 88   88            g ~~      Spatial   1     1     1    0      0   0
## 89   89            g ~~        Quant   1     1     1    0      0   0
## 90   90            g ~~         Oral   1     1     1    0      0   0
## 91   91            g ~~      Written   1     1     1    0      0   0
## 92   92            g ~~ Manipulative   1     1     1    0      0   0
## 93   93          VO1 ~~          VO1   1     1     1   58     NA   0
## 94   94          VO2 ~~          VO2   1     1     1   59     NA   0
## 95   95          VO3 ~~          VO3   1     1     1   60     NA   0
## 96   96          VW1 ~~          VW1   1     1     1   61     NA   0
## 97   97          VW2 ~~          VW2   1     1     1   62     NA   0
## 98   98          VW3 ~~          VW3   1     1     1   63     NA   0
## 99   99          VM1 ~~          VM1   1     1     1   64     NA   0
## 100 100          VM2 ~~          VM2   1     1     1   65     NA   0
## 101 101          VM3 ~~          VM3   1     1     1   66     NA   0
## 102 102          SO1 ~~          SO1   1     1     1   67     NA   0
## 103 103          SO2 ~~          SO2   1     1     1   68     NA   0
```

```
## 104 104          SO3 ~~          SO3    1     1     1   69     NA   0
## 105 105          SW1 ~~          SW1    1     1     1   70     NA   0
## 106 106          SW2 ~~          SW2    1     1     1   71     NA   0
## 107 107          SW3 ~~          SW3    1     1     1   72     NA   0
## 108 108          SM1 ~~          SM1    1     1     1   73     NA   0
## 109 109          SM2 ~~          SM2    1     1     1   74     NA   0
## 110 110          SM3 ~~          SM3    1     1     1   75     NA   0
## 111 111          QO1 ~~          QO1    1     1     1   76     NA   0
## 112 112          QO2 ~~          QO2    1     1     1   77     NA   0
## 113 113          QO3 ~~          QO3    1     1     1   78     NA   0
## 114 114          QW1 ~~          QW1    1     1     1   79     NA   0
## 115 115          QW2 ~~          QW2    1     1     1   80     NA   0
## 116 116          QW3 ~~          QW3    1     1     1   81     NA   0
## 117 117          QM1 ~~          QM1    1     1     1   82     NA   0
## 118 118          QM2 ~~          QM2    1     1     1   83     NA   0
## 119 119          QM3 ~~          QM3    1     1     1   84     NA   0
## 120 120          VO1 ~1                 1     1     1   85     NA   0 intVO1
## 121 121          VO2 ~1                 1     1     1   86     NA   0 intVO2
## 122 122          VO3 ~1                 1     1     1   87     NA   0 intVO3
## 123 123          VW1 ~1                 1     1     1   88     NA   0 intVW1
## 124 124          VW2 ~1                 1     1     1   89     NA   0 intVW2
## 125 125          VW3 ~1                 1     1     1   90     NA   0 intVW3
## 126 126          VM1 ~1                 1     1     1   91     NA   0 intVM1
## 127 127          VM2 ~1                 1     1     1   92     NA   0 intVM2
## 128 128          VM3 ~1                 1     1     1   93     NA   0 intVM3
## 129 129          SO1 ~1                 1     1     1   94     NA   0 intSO1
## 130 130          SO2 ~1                 1     1     1   95     NA   0 intSO2
## 131 131          SO3 ~1                 1     1     1   96     NA   0 intSO3
## 132 132          SW1 ~1                 1     1     1   97     NA   0 intSW1
## 133 133          SW2 ~1                 1     1     1   98     NA   0 intSW2
## 134 134          SW3 ~1                 1     1     1   99     NA   0 intSW3
## 135 135          SM1 ~1                 1     1     1  100     NA   0 intSM1
## 136 136          SM2 ~1                 1     1     1  101     NA   0 intSM2
## 137 137          SM3 ~1                 1     1     1  102     NA   0 intSM3
## 138 138          QO1 ~1                 1     1     1  103     NA   0 intQO1
## 139 139          QO2 ~1                 1     1     1  104     NA   0 intQO2
## 140 140          QO3 ~1                 1     1     1  105     NA   0 intQO3
## 141 141          QW1 ~1                 1     1     1  106     NA   0 intQW1
## 142 142          QW2 ~1                 1     1     1  107     NA   0 intQW2
## 143 143          QW3 ~1                 1     1     1  108     NA   0 intQW3
## 144 144          QM1 ~1                 1     1     1  109     NA   0 intQM1
## 145 145          QM2 ~1                 1     1     1  110     NA   0 intQM2
## 146 146          QM3 ~1                 1     1     1  111     NA   0 intQM3
##     plabel
## 1     .p1.
## 2     .p2.
```

```
## 3     .p3.
## 4     .p4.
## 5     .p5.
## 6     .p6.
## 7     .p7.
## 8     .p8.
## 9     .p9.
## 10   .p10.
## 11   .p11.
## 12   .p12.
## 13   .p13.
## 14   .p14.
## 15   .p15.
## 16   .p16.
## 17   .p17.
## 18   .p18.
## 19   .p19.
## 20   .p20.
## 21   .p21.
## 22   .p22.
## 23   .p23.
## 24   .p24.
## 25   .p25.
## 26   .p26.
## 27   .p27.
## 28   .p28.
## 29   .p29.
## 30   .p30.
## 31   .p31.
## 32   .p32.
## 33   .p33.
## 34   .p34.
## 35   .p35.
## 36   .p36.
## 37   .p37.
## 38   .p38.
## 39   .p39.
## 40   .p40.
## 41   .p41.
## 42   .p42.
## 43   .p43.
## 44   .p44.
## 45   .p45.
## 46   .p46.
## 47   .p47.
## 48   .p48.
```

```
## 49   .p49.
## 50   .p50.
## 51   .p51.
## 52   .p52.
## 53   .p53.
## 54   .p54.
## 55   .p55.
## 56   .p56.
## 57   .p57.
## 58   .p58.
## 59   .p59.
## 60   .p60.
## 61   .p61.
## 62   .p62.
## 63   .p63.
## 64   .p64.
## 65   .p65.
## 66   .p66.
## 67   .p67.
## 68   .p68.
## 69   .p69.
## 70   .p70.
## 71   .p71.
## 72   .p72.
## 73   .p73.
## 74   .p74.
## 75   .p75.
## 76   .p76.
## 77   .p77.
## 78   .p78.
## 79   .p79.
## 80   .p80.
## 81   .p81.
## 82   .p82.
## 83   .p83.
## 84   .p84.
## 85   .p85.
## 86   .p86.
## 87   .p87.
## 88   .p88.
## 89   .p89.
## 90   .p90.
## 91   .p91.
## 92   .p92.
## 93   .p93.
## 94   .p94.
```

```
## 95   .p95.
## 96   .p96.
## 97   .p97.
## 98   .p98.
## 99   .p99.
## 100 .p100.
## 101 .p101.
## 102 .p102.
## 103 .p103.
## 104 .p104.
## 105 .p105.
## 106 .p106.
## 107 .p107.
## 108 .p108.
## 109 .p109.
## 110 .p110.
## 111 .p111.
## 112 .p112.
## 113 .p113.
## 114 .p114.
## 115 .p115.
## 116 .p116.
## 117 .p117.
## 118 .p118.
## 119 .p119.
## 120 .p120.
## 121 .p121.
## 122 .p122.
## 123 .p123.
## 124 .p124.
## 125 .p125.
## 126 .p126.
## 127 .p127.
## 128 .p128.
## 129 .p129.
## 130 .p130.
## 131 .p131.
## 132 .p132.
## 133 .p133.
## 134 .p134.
## 135 .p135.
## 136 .p136.
## 137 .p137.
## 138 .p138.
## 139 .p139.
## 140 .p140.
```

```
## 141 .p141.
## 142 .p142.
## 143 .p143.
## 144 .p144.
## 145 .p145.
## 146 .p146.
```

12.5.12.3 Fit the model

```
cfaModelMTMMFit <- cfa(cfaModelMTMM_syntax,
                       data = MTMM,
                       orthogonal = TRUE,
                       std.lv = TRUE,
                       missing = "ML",
                       estimator = "MLR")

cfaModelMTMMFit_full <- lavaan(cfaModelMTMM_fullSyntax,
                               data = MTMM,
                               missing = "ML",
                               estimator = "MLR")
```

12.5.12.4 Display summary output

```
summary(cfaModelMTMMFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 54 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                       111
##
##   Number of observations                          1000
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               321.321     324.370
##   Degrees of freedom                               294         294
```

```
##   P-value (Chi-square)                           0.131       0.108
##   Scaling correction factor                                  0.991
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                             14923.912   14951.143
##   Degrees of freedom                               351         351
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  0.998
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.998       0.998
##   Tucker-Lewis Index (TLI)                       0.998       0.998
##
##   Robust Comparative Fit Index (CFI)                         0.998
##   Robust Tucker-Lewis Index (TLI)                            0.998
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)            -31375.377  -31375.377
##   Scaling correction factor                                  1.018
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)    -31214.716  -31214.716
##   Scaling correction factor                                  0.998
##       for the MLR correction
##
##   Akaike (AIC)                              62972.753   62972.753
##   Bayesian (BIC)                            63517.514   63517.514
##   Sample-size adjusted Bayesian (BIC)       63164.972   63164.972
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.010       0.010
##   90 Percent confidence interval - lower         0.000       0.000
##   90 Percent confidence interval - upper         0.016       0.016
##   P-value RMSEA <= 0.05                          1.000       1.000
##
##   Robust RMSEA                                               0.010
##   90 Percent confidence interval - lower                     0.000
##   90 Percent confidence interval - upper                     0.016
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.020       0.020
```

```
##
## Parameter Estimates:
##
##   Standard errors                              Sandwich
##   Information bread                            Observed
##   Observed information based on                 Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   g =~
##     Verbal            1.355    0.098   13.877    0.000    0.805    0.805
##     Spatial           1.036    0.070   14.703    0.000    0.720    0.720
##     Quant             1.699    0.158   10.763    0.000    0.862    0.862
##   Verbal =~
##     VO1               0.418    0.024   17.718    0.000    0.704    0.692
##     VO2               0.440    0.024   18.609    0.000    0.741    0.721
##     VO3               0.416    0.023   17.952    0.000    0.701    0.693
##     VW1               0.404    0.023   17.224    0.000    0.680    0.691
##     VW2               0.437    0.024   18.383    0.000    0.736    0.721
##     VW3               0.428    0.025   17.395    0.000    0.721    0.693
##     VM1               0.421    0.024   17.322    0.000    0.708    0.715
##     VM2               0.435    0.024   17.927    0.000    0.732    0.717
##     VM3               0.445    0.025   17.729    0.000    0.749    0.729
##   Spatial =~
##     SO1               0.486    0.025   19.575    0.000    0.700    0.692
##     SO2               0.505    0.025   20.548    0.000    0.728    0.729
##     SO3               0.485    0.024   20.335    0.000    0.699    0.689
##     SW1               0.486    0.023   21.322    0.000    0.701    0.704
##     SW2               0.503    0.023   22.268    0.000    0.725    0.712
##     SW3               0.500    0.024   20.503    0.000    0.721    0.706
##     SM1               0.491    0.023   21.309    0.000    0.707    0.696
##     SM2               0.484    0.024   19.901    0.000    0.697    0.690
##     SM3               0.497    0.024   20.970    0.000    0.716    0.711
##   Quant =~
##     QO1               0.353    0.027   13.168    0.000    0.695    0.706
##     QO2               0.352    0.027   13.087    0.000    0.694    0.696
##     QO3               0.383    0.029   13.308    0.000    0.755    0.708
##     QW1               0.346    0.027   13.028    0.000    0.681    0.680
##     QW2               0.369    0.027   13.564    0.000    0.728    0.715
##     QW3               0.369    0.027   13.587    0.000    0.728    0.698
##     QM1               0.375    0.028   13.377    0.000    0.739    0.734
##     QM2               0.370    0.028   13.220    0.000    0.730    0.716
##     QM3               0.373    0.028   13.404    0.000    0.736    0.715
##   Oral =~
##     VO1               0.255    0.038    6.804    0.000    0.255    0.251
##     VO2               0.250    0.039    6.447    0.000    0.250    0.243
```

```
##     VO3               0.289    0.038    7.607    0.000    0.289    0.286
##     SO1               0.252    0.042    5.984    0.000    0.252    0.249
##     SO2               0.239    0.036    6.563    0.000    0.239    0.240
##     SO3               0.254    0.042    5.986    0.000    0.254    0.251
##     QO1               0.247    0.042    5.914    0.000    0.247    0.251
##     QO2               0.316    0.039    8.089    0.000    0.316    0.316
##     QO3               0.344    0.042    8.193    0.000    0.344    0.322
##   Written =~
##     VW1               0.310    0.035    8.888    0.000    0.310    0.315
##     VW2               0.343    0.033   10.542    0.000    0.343    0.336
##     VW3               0.292    0.034    8.640    0.000    0.292    0.280
##     SW1               0.290    0.035    8.395    0.000    0.290    0.291
##     SW2               0.344    0.033   10.581    0.000    0.344    0.338
##     SW3               0.355    0.035   10.156    0.000    0.355    0.348
##     QW1               0.323    0.035    9.140    0.000    0.323    0.323
##     QW2               0.270    0.035    7.608    0.000    0.270    0.265
##     QW3               0.358    0.033   10.708    0.000    0.358    0.344
##   Manipulative =~
##     VM1               0.217    0.056    3.894    0.000    0.217    0.219
##     VM2               0.202    0.052    3.868    0.000    0.202    0.197
##     VM3               0.203    0.051    4.015    0.000    0.203    0.198
##     SM1               0.251    0.054    4.641    0.000    0.251    0.247
##     SM2               0.203    0.055    3.704    0.000    0.203    0.201
##     SM3               0.249    0.052    4.829    0.000    0.249    0.247
##     QM1               0.205    0.046    4.432    0.000    0.205    0.204
##     QM2               0.194    0.055    3.501    0.000    0.194    0.190
##     QM3               0.194    0.055    3.497    0.000    0.194    0.188
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   g ~~
##     Oral              0.000                               0.000    0.000
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##   Oral ~~
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##   Written ~~
##     Manipulative      0.000                               0.000    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .VO1               0.040    0.032    1.255    0.210    0.040    0.040
##    .VO2               0.023    0.033    0.694    0.488    0.023    0.022
##    .VO3               0.023    0.032    0.711    0.477    0.023    0.023
##    .VW1              -0.003    0.031   -0.083    0.934   -0.003   -0.003
```

```
##    .VW2               0.016    0.032    0.496    0.620    0.016    0.016
##    .VW3               0.034    0.033    1.051    0.293    0.034    0.033
##    .VM1               0.064    0.031    2.035    0.042    0.064    0.064
##    .VM2               0.015    0.032    0.476    0.634    0.015    0.015
##    .VM3               0.061    0.033    1.880    0.060    0.061    0.060
##    .SO1               0.013    0.032    0.407    0.684    0.013    0.013
##    .SO2              -0.021    0.032   -0.652    0.514   -0.021   -0.021
##    .SO3               0.004    0.032    0.110    0.913    0.004    0.003
##    .SW1               0.002    0.031    0.070    0.944    0.002    0.002
##    .SW2               0.006    0.032    0.197    0.844    0.006    0.006
##    .SW3               0.013    0.032    0.404    0.686    0.013    0.013
##    .SM1               0.066    0.032    2.043    0.041    0.066    0.065
##    .SM2               0.008    0.032    0.241    0.810    0.008    0.008
##    .SM3               0.004    0.032    0.118    0.906    0.004    0.004
##    .QO1               0.046    0.031    1.474    0.141    0.046    0.047
##    .QO2               0.025    0.032    0.781    0.435    0.025    0.025
##    .QO3               0.023    0.034    0.691    0.489    0.023    0.022
##    .QW1               0.047    0.031    1.497    0.134    0.047    0.047
##    .QW2               0.044    0.032    1.391    0.164    0.044    0.044
##    .QW3               0.006    0.033    0.175    0.861    0.006    0.005
##    .QM1               0.068    0.032    2.141    0.032    0.068    0.068
##    .QM2               0.011    0.032    0.328    0.743    0.011    0.010
##    .QM3               0.045    0.033    1.385    0.166    0.045    0.044
##     g                 0.000                                0.000    0.000
##    .Verbal            0.000                                0.000    0.000
##    .Spatial           0.000                                0.000    0.000
##    .Quant             0.000                                0.000    0.000
##     Oral              0.000                                0.000    0.000
##     Written           0.000                                0.000    0.000
##     Manipulative      0.000                                0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .VO1               0.475    0.025   19.276    0.000    0.475    0.459
##    .VO2               0.445    0.024   18.342    0.000    0.445    0.421
##    .VO3               0.448    0.025   18.226    0.000    0.448    0.438
##    .VW1               0.410    0.023   17.861    0.000    0.410    0.424
##    .VW2               0.381    0.023   16.577    0.000    0.381    0.366
##    .VW3               0.479    0.024   19.719    0.000    0.479    0.441
##    .VM1               0.434    0.026   16.826    0.000    0.434    0.441
##    .VM2               0.466    0.025   18.940    0.000    0.466    0.447
##    .VM3               0.452    0.025   17.797    0.000    0.452    0.429
##    .SO1               0.470    0.025   19.147    0.000    0.470    0.459
##    .SO2               0.410    0.023   17.455    0.000    0.410    0.411
##    .SO3               0.476    0.027   17.777    0.000    0.476    0.463
##    .SW1               0.415    0.022   18.569    0.000    0.415    0.419
```

```
##    .SW2              0.394    0.022   17.759    0.000    0.394    0.380
##    .SW3              0.396    0.022   18.040    0.000    0.396    0.380
##    .SM1              0.468    0.028   16.949    0.000    0.468    0.454
##    .SM2              0.492    0.026   18.839    0.000    0.492    0.483
##    .SM3              0.440    0.025   17.328    0.000    0.440    0.433
##    .QO1              0.426    0.022   19.094    0.000    0.426    0.439
##    .QO2              0.415    0.024   17.327    0.000    0.415    0.416
##    .QO3              0.450    0.027   16.924    0.000    0.450    0.395
##    .QW1              0.434    0.023   18.785    0.000    0.434    0.433
##    .QW2              0.433    0.024   17.756    0.000    0.433    0.418
##    .QW3              0.428    0.023   18.785    0.000    0.428    0.394
##    .QM1              0.426    0.024   17.891    0.000    0.426    0.420
##    .QM2              0.469    0.024   19.297    0.000    0.469    0.451
##    .QM3              0.480    0.025   19.426    0.000    0.480    0.453
##     g                1.000                                1.000    1.000
##    .Verbal           1.000                                0.353    0.353
##    .Spatial          1.000                                0.482    0.482
##    .Quant            1.000                                0.257    0.257
##     Oral             1.000                                1.000    1.000
##     Written          1.000                                1.000    1.000
##     Manipulative     1.000                                1.000    1.000
##
## R-Square:
##                    Estimate
##     VO1               0.541
##     VO2               0.579
##     VO3               0.562
##     VW1               0.576
##     VW2               0.634
##     VW3               0.559
##     VM1               0.559
##     VM2               0.553
##     VM3               0.571
##     SO1               0.541
##     SO2               0.589
##     SO3               0.537
##     SW1               0.581
##     SW2               0.620
##     SW3               0.620
##     SM1               0.546
##     SM2               0.517
##     SM3               0.567
##     QO1               0.561
##     QO2               0.584
##     QO3               0.605
##     QW1               0.567
```

```
##     QW2               0.582
##     QW3               0.606
##     QM1               0.580
##     QM2               0.549
##     QM3               0.547
##     Verbal            0.647
##     Spatial           0.518
##     Quant             0.743
```

```
summary(cfaModelMTMMFit_full,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 54 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                       111
##
##   Number of observations                          1000
##   Number of missing patterns                         1
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               321.321     324.370
##   Degrees of freedom                               294         294
##   P-value (Chi-square)                           0.131       0.108
##   Scaling correction factor                                  0.991
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                             14923.912   14951.143
##   Degrees of freedom                               351         351
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  0.998
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.998       0.998
##   Tucker-Lewis Index (TLI)                       0.998       0.998
##
##   Robust Comparative Fit Index (CFI)                         0.998
##   Robust Tucker-Lewis Index (TLI)                            0.998
```

```
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -31375.377  -31375.377
##   Scaling correction factor                                  1.018
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)     -31214.716  -31214.716
##   Scaling correction factor                                  0.998
##       for the MLR correction
##
##   Akaike (AIC)                               62972.753   62972.753
##   Bayesian (BIC)                             63517.514   63517.514
##   Sample-size adjusted Bayesian (BIC)        63164.972   63164.972
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.010       0.010
##   90 Percent confidence interval - lower         0.000       0.000
##   90 Percent confidence interval - upper         0.016       0.016
##   P-value RMSEA <= 0.05                          1.000       1.000
##
##   Robust RMSEA                                               0.010
##   90 Percent confidence interval - lower                     0.000
##   90 Percent confidence interval - upper                     0.016
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.020       0.020
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   g =~
##     Verbal            1.355    0.098   13.877    0.000    0.805    0.805
##     Spatial           1.036    0.070   14.703    0.000    0.720    0.720
##     Quant             1.699    0.158   10.763    0.000    0.862    0.862
##   Verbal =~
##     VO1               0.418    0.024   17.718    0.000    0.704    0.692
##     VO2               0.440    0.024   18.609    0.000    0.741    0.721
##     VO3               0.416    0.023   17.952    0.000    0.701    0.693
##     VW1               0.404    0.023   17.224    0.000    0.680    0.691
```

```
##     VW2              0.437    0.024   18.383    0.000    0.736    0.721
##     VW3              0.428    0.025   17.395    0.000    0.721    0.693
##     VM1              0.421    0.024   17.322    0.000    0.708    0.715
##     VM2              0.435    0.024   17.927    0.000    0.732    0.717
##     VM3              0.445    0.025   17.729    0.000    0.749    0.729
##   Spatial =~
##     SO1              0.486    0.025   19.575    0.000    0.700    0.692
##     SO2              0.505    0.025   20.548    0.000    0.728    0.729
##     SO3              0.485    0.024   20.335    0.000    0.699    0.689
##     SW1              0.486    0.023   21.322    0.000    0.701    0.704
##     SW2              0.503    0.023   22.268    0.000    0.725    0.712
##     SW3              0.500    0.024   20.503    0.000    0.721    0.706
##     SM1              0.491    0.023   21.309    0.000    0.707    0.696
##     SM2              0.484    0.024   19.901    0.000    0.697    0.690
##     SM3              0.497    0.024   20.970    0.000    0.716    0.711
##   Quant =~
##     QO1              0.353    0.027   13.168    0.000    0.695    0.706
##     QO2              0.352    0.027   13.087    0.000    0.694    0.696
##     QO3              0.383    0.029   13.308    0.000    0.755    0.708
##     QW1              0.346    0.027   13.028    0.000    0.681    0.680
##     QW2              0.369    0.027   13.564    0.000    0.728    0.715
##     QW3              0.369    0.027   13.587    0.000    0.728    0.698
##     QM1              0.375    0.028   13.377    0.000    0.739    0.734
##     QM2              0.370    0.028   13.220    0.000    0.730    0.716
##     QM3              0.373    0.028   13.404    0.000    0.736    0.715
##   Oral =~
##     VO1              0.255    0.038    6.804    0.000    0.255    0.251
##     VO2              0.250    0.039    6.447    0.000    0.250    0.243
##     VO3              0.289    0.038    7.607    0.000    0.289    0.286
##     SO1              0.252    0.042    5.984    0.000    0.252    0.249
##     SO2              0.239    0.036    6.563    0.000    0.239    0.240
##     SO3              0.254    0.042    5.986    0.000    0.254    0.251
##     QO1              0.247    0.042    5.914    0.000    0.247    0.251
##     QO2              0.316    0.039    8.089    0.000    0.316    0.316
##     QO3              0.344    0.042    8.193    0.000    0.344    0.322
##   Written =~
##     VW1              0.310    0.035    8.888    0.000    0.310    0.315
##     VW2              0.343    0.033   10.542    0.000    0.343    0.336
##     VW3              0.292    0.034    8.640    0.000    0.292    0.280
##     SW1              0.290    0.035    8.395    0.000    0.290    0.291
##     SW2              0.344    0.033   10.581    0.000    0.344    0.338
##     SW3              0.355    0.035   10.156    0.000    0.355    0.348
##     QW1              0.323    0.035    9.140    0.000    0.323    0.323
##     QW2              0.270    0.035    7.608    0.000    0.270    0.265
##     QW3              0.358    0.033   10.708    0.000    0.358    0.344
##   Manipulative =~
```

```
##     VM1               0.217    0.056    3.894    0.000    0.217    0.219
##     VM2               0.202    0.052    3.868    0.000    0.202    0.197
##     VM3               0.203    0.051    4.015    0.000    0.203    0.198
##     SM1               0.251    0.054    4.641    0.000    0.251    0.247
##     SM2               0.203    0.055    3.704    0.000    0.203    0.201
##     SM3               0.249    0.052    4.829    0.000    0.249    0.247
##     QM1               0.205    0.046    4.432    0.000    0.205    0.204
##     QM2               0.194    0.055    3.501    0.000    0.194    0.190
##     QM3               0.194    0.055    3.497    0.000    0.194    0.188
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .Verbal ~~
##    .Spatial           0.000                               0.000    0.000
##    .Quant             0.000                               0.000    0.000
##     Oral              0.000                               0.000    0.000
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##  .Spatial ~~
##    .Quant             0.000                               0.000    0.000
##     Oral              0.000                               0.000    0.000
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##  .Quant ~~
##     Oral              0.000                               0.000    0.000
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##   Oral ~~
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##   Written ~~
##     Manipulative      0.000                               0.000    0.000
##   g ~~
##    .Verbal            0.000                               0.000    0.000
##    .Spatial           0.000                               0.000    0.000
##    .Quant             0.000                               0.000    0.000
##     Oral              0.000                               0.000    0.000
##     Written           0.000                               0.000    0.000
##     Manipulative      0.000                               0.000    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Verbal            0.000                               0.000    0.000
##    .Spatial           0.000                               0.000    0.000
##    .Quant             0.000                               0.000    0.000
##     Oral              0.000                               0.000    0.000
```

```
##     Written           0.000                                0.000    0.000
##     Manpltv           0.000                                0.000    0.000
##     g                 0.000                                0.000    0.000
##    .VO1     (iVO1)    0.040    0.032    1.255    0.210    0.040    0.040
##    .VO2     (iVO2)    0.023    0.033    0.694    0.488    0.023    0.022
##    .VO3     (iVO3)    0.023    0.032    0.711    0.477    0.023    0.023
##    .VW1     (iVW1)   -0.003    0.031   -0.083    0.934   -0.003   -0.003
##    .VW2     (iVW2)    0.016    0.032    0.496    0.620    0.016    0.016
##    .VW3     (iVW3)    0.034    0.033    1.051    0.293    0.034    0.033
##    .VM1     (iVM1)    0.064    0.031    2.035    0.042    0.064    0.064
##    .VM2     (iVM2)    0.015    0.032    0.476    0.634    0.015    0.015
##    .VM3     (iVM3)    0.061    0.033    1.880    0.060    0.061    0.060
##    .SO1     (iSO1)    0.013    0.032    0.407    0.684    0.013    0.013
##    .SO2     (iSO2)   -0.021    0.032   -0.652    0.514   -0.021   -0.021
##    .SO3     (iSO3)    0.004    0.032    0.110    0.913    0.004    0.003
##    .SW1     (iSW1)    0.002    0.031    0.070    0.944    0.002    0.002
##    .SW2     (iSW2)    0.006    0.032    0.197    0.844    0.006    0.006
##    .SW3     (iSW3)    0.013    0.032    0.404    0.686    0.013    0.013
##    .SM1     (iSM1)    0.066    0.032    2.043    0.041    0.066    0.065
##    .SM2     (iSM2)    0.008    0.032    0.241    0.810    0.008    0.008
##    .SM3     (iSM3)    0.004    0.032    0.118    0.906    0.004    0.004
##    .QO1     (iQO1)    0.046    0.031    1.474    0.141    0.046    0.047
##    .QO2     (iQO2)    0.025    0.032    0.781    0.435    0.025    0.025
##    .QO3     (iQO3)    0.023    0.034    0.691    0.489    0.023    0.022
##    .QW1     (iQW1)    0.047    0.031    1.497    0.134    0.047    0.047
##    .QW2     (iQW2)    0.044    0.032    1.391    0.164    0.044    0.044
##    .QW3     (iQW3)    0.006    0.033    0.175    0.861    0.006    0.005
##    .QM1     (iQM1)    0.068    0.032    2.141    0.032    0.068    0.068
##    .QM2     (iQM2)    0.011    0.032    0.328    0.743    0.011    0.010
##    .QM3     (iQM3)    0.045    0.033    1.385    0.166    0.045    0.044
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Verbal            1.000                                0.353    0.353
##    .Spatial           1.000                                0.482    0.482
##    .Quant             1.000                                0.257    0.257
##     Oral              1.000                                1.000    1.000
##     Written           1.000                                1.000    1.000
##     Manipulative      1.000                                1.000    1.000
##     g                 1.000                                1.000    1.000
##    .VO1               0.475    0.025   19.276    0.000    0.475    0.459
##    .VO2               0.445    0.024   18.342    0.000    0.445    0.421
##    .VO3               0.448    0.025   18.226    0.000    0.448    0.438
##    .VW1               0.410    0.023   17.861    0.000    0.410    0.424
##    .VW2               0.381    0.023   16.577    0.000    0.381    0.366
##    .VW3               0.479    0.024   19.719    0.000    0.479    0.441
```

```
##    .VM1                0.434    0.026   16.826    0.000    0.434    0.441
##    .VM2                0.466    0.025   18.940    0.000    0.466    0.447
##    .VM3                0.452    0.025   17.797    0.000    0.452    0.429
##    .SO1                0.470    0.025   19.147    0.000    0.470    0.459
##    .SO2                0.410    0.023   17.455    0.000    0.410    0.411
##    .SO3                0.476    0.027   17.777    0.000    0.476    0.463
##    .SW1                0.415    0.022   18.569    0.000    0.415    0.419
##    .SW2                0.394    0.022   17.759    0.000    0.394    0.380
##    .SW3                0.396    0.022   18.040    0.000    0.396    0.380
##    .SM1                0.468    0.028   16.949    0.000    0.468    0.454
##    .SM2                0.492    0.026   18.839    0.000    0.492    0.483
##    .SM3                0.440    0.025   17.328    0.000    0.440    0.433
##    .QO1                0.426    0.022   19.094    0.000    0.426    0.439
##    .QO2                0.415    0.024   17.327    0.000    0.415    0.416
##    .QO3                0.450    0.027   16.924    0.000    0.450    0.395
##    .QW1                0.434    0.023   18.785    0.000    0.434    0.433
##    .QW2                0.433    0.024   17.756    0.000    0.433    0.418
##    .QW3                0.428    0.023   18.785    0.000    0.428    0.394
##    .QM1                0.426    0.024   17.891    0.000    0.426    0.420
##    .QM2                0.469    0.024   19.297    0.000    0.469    0.451
##    .QM3                0.480    0.025   19.426    0.000    0.480    0.453
##
## R-Square:
##                    Estimate
##     Verbal            0.647
##     Spatial           0.518
##     Quant             0.743
##     VO1               0.541
##     VO2               0.579
##     VO3               0.562
##     VW1               0.576
##     VW2               0.634
##     VW3               0.559
##     VM1               0.559
##     VM2               0.553
##     VM3               0.571
##     SO1               0.541
##     SO2               0.589
##     SO3               0.537
##     SW1               0.581
##     SW2               0.620
##     SW3               0.620
##     SM1               0.546
##     SM2               0.517
##     SM3               0.567
##     QO1               0.561
```

```
##     QO2               0.584
##     QO3               0.605
##     QW1               0.567
##     QW2               0.582
##     QW3               0.606
##     QM1               0.580
##     QM2               0.549
##     QM3               0.547
```

12.5.12.5 Estimates of model fit

```
fitMeasures(cfaModelMTMMFit, fit.measures = c("chisq", "df", "pvalue",
                                              "chisq.scaled", "df.scaled", "pvalue.scaled",
                                              "chisq.scaling.factor",
                                              "rmsea", "cfi", "tli", "srmr",
                                              "rmsea.robust", "cfi.robust", "tli.robust"))
```

```
##                chisq                   df               pvalue
##              321.321              294.000                0.131
##         chisq.scaled            df.scaled        pvalue.scaled
##              324.370              294.000                0.108
## chisq.scaling.factor                rmsea                  cfi
##                0.991                0.010                0.998
##                  tli                 srmr         rmsea.robust
##                0.998                0.020                0.010
##           cfi.robust           tli.robust
##                0.998                0.998
```

12.5.12.6 Residuals

```
residuals(cfaModelMTMMFit, type = "cor")
```

```
## $type
## [1] "cor.bollen"
##
## $cov
##     VO1    VO2    VO3    VW1    VW2    VW3    VM1    VM2    VM3    SO1
## VO1  0.000
## VO2 -0.015  0.000
## VO3 -0.009 -0.004  0.000
## VW1  0.006 -0.013  0.013  0.000
```

```
## VW2 -0.015  0.003 -0.008  0.001  0.000
## VW3  0.016  0.000 -0.001 -0.001 -0.011  0.000
## VM1  0.006  0.014 -0.011 -0.009  0.000  0.013  0.000
## VM2  0.016  0.010  0.020 -0.014 -0.008 -0.018 -0.016  0.000
## VM3 -0.010 -0.026 -0.002  0.001  0.018  0.002  0.003  0.010  0.000
## SO1 -0.019  0.010  0.003 -0.069 -0.030 -0.009 -0.038  0.003 -0.044  0.000
## SO2 -0.010  0.026  0.016 -0.011  0.025 -0.010 -0.008 -0.012 -0.001  0.002
## SO3  0.020  0.020  0.026 -0.028  0.014  0.008  0.005  0.043  0.003  0.016
## SW1  0.022  0.038  0.016 -0.019  0.027 -0.003  0.001  0.019 -0.004 -0.005
## SW2 -0.011  0.001  0.027 -0.021 -0.006  0.000 -0.016 -0.001  0.002 -0.016
## SW3 -0.047  0.005 -0.001 -0.080 -0.002 -0.009 -0.007 -0.016 -0.049  0.021
## SM1 -0.036  0.009  0.031 -0.021 -0.007 -0.048 -0.041 -0.002 -0.040 -0.008
## SM2  0.006  0.041  0.025 -0.005  0.022 -0.015 -0.014  0.034  0.007 -0.006
## SM3  0.018  0.028  0.030 -0.018  0.021  0.020  0.019  0.028 -0.006  0.003
## QO1  0.028  0.055  0.014  0.001 -0.001 -0.004 -0.013  0.006 -0.014 -0.027
## QO2 -0.006  0.022  0.001 -0.021 -0.020 -0.030 -0.024 -0.009  0.017 -0.020
## QO3  0.015  0.006  0.019 -0.011 -0.004 -0.043  0.023  0.038  0.032 -0.046
## QW1 -0.014 -0.007  0.001 -0.002 -0.015 -0.028 -0.010  0.011  0.012 -0.039
## QW2  0.006  0.002 -0.003  0.004 -0.036 -0.022 -0.002 -0.016  0.026 -0.037
## QW3 -0.019 -0.003 -0.004 -0.012 -0.022 -0.055 -0.007 -0.020  0.003 -0.030
## QM1 -0.008  0.022  0.004 -0.017 -0.017 -0.037  0.007  0.003  0.016 -0.020
## QM2  0.021  0.033  0.041 -0.027 -0.022 -0.029  0.017 -0.004  0.012 -0.004
## QM3  0.002  0.045 -0.033  0.007 -0.021 -0.022  0.023 -0.012  0.021  0.008
##     SO2    SO3    SW1    SW2    SW3    SM1    SM2    SM3    QO1    QO2
## VO1
## VO2
## VO3
## VW1
## VW2
## VW3
## VM1
## VM2
## VM3
## SO1
## SO2  0.000
## SO3  0.014  0.000
## SW1 -0.008 -0.015  0.000
## SW2  0.007 -0.019 -0.005  0.000
## SW3 -0.003 -0.003  0.011 -0.007  0.000
## SM1 -0.002 -0.007  0.004  0.013 -0.011  0.000
## SM2 -0.009  0.001  0.014 -0.018  0.003  0.011  0.000
## SM3 -0.006  0.007 -0.014  0.008 -0.011  0.010 -0.005  0.000
## QO1  0.016 -0.017  0.027  0.019 -0.049 -0.002  0.029  0.032  0.000
## QO2 -0.002 -0.012  0.016  0.040 -0.033  0.013  0.000  0.001 -0.007  0.000
## QO3  0.001 -0.007 -0.036  0.001 -0.070 -0.035  0.008  0.029 -0.007  0.017
## QW1 -0.007 -0.014 -0.025  0.013 -0.041  0.022  0.004  0.031 -0.020  0.002
```

```
## QW2  0.017  0.004  0.004  0.021 -0.046 -0.019  0.020  0.019  0.010 -0.006
## QW3 -0.012  0.005 -0.032  0.029 -0.032  0.006  0.003  0.024  0.006 -0.016
## QM1  0.025 -0.005 -0.020  0.030 -0.012  0.002  0.014  0.020  0.000  0.009
## QM2  0.034  0.023 -0.021  0.033 -0.058  0.014 -0.002  0.014  0.007  0.013
## QM3  0.014  0.000 -0.019  0.033 -0.040  0.003  0.022  0.015  0.003 -0.002
##     QO3    QW1    QW2    QW3    QM1    QM2    QM3
## VO1
## VO2
## VO3
## VW1
## VW2
## VW3
## VM1
## VM2
## VM3
## SO1
## SO2
## SO3
## SW1
## SW2
## SW3
## SM1
## SM2
## SM3
## QO1
## QO2
## QO3  0.000
## QW1  0.008  0.000
## QW2 -0.012 -0.006  0.000
## QW3 -0.019 -0.004 -0.004  0.000
## QM1  0.001  0.008 -0.015  0.021  0.000
## QM2  0.002 -0.024 -0.020 -0.010  0.007  0.000
## QM3  0.003 -0.006  0.016 -0.015 -0.021  0.017  0.000
##
## $mean
## VO1 VO2 VO3 VW1 VW2 VW3 VM1 VM2 VM3 SO1 SO2 SO3 SW1 SW2 SW3 SM1 SM2 SM3 QO1 QO2
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## QO3 QW1 QW2 QW3 QM1 QM2 QM3
##   0   0   0   0   0   0   0
```

12.5.12.7 Modification indices

```
modificationindices(cfaModelMTMMFit)
```

```
##              lhs op          rhs     mi    epc sepc.lv sepc.all sepc.nox
## 92             g ~~         Oral  0.486  0.065   0.065    0.065    0.065
## 93             g ~~      Written  5.120 -0.174  -0.174   -0.174   -0.174
## 94             g ~~ Manipulative  1.673  0.145   0.145    0.145    0.145
## 95          Oral ~~      Written  2.508 -0.224  -0.224   -0.224   -0.224
## 96          Oral ~~ Manipulative  2.372  0.331   0.331    0.331    0.331
## 97       Written ~~ Manipulative  0.047  0.041   0.041    0.041    0.041
## 132            g =~          VO1  0.003 -0.003  -0.003   -0.003   -0.003
## 133            g =~          VO2  5.242  0.133   0.133    0.130    0.130
## 134            g =~          VO3  0.821  0.053   0.053    0.053    0.053
## 135            g =~          VW1  1.195 -0.061  -0.061   -0.062   -0.062
## 136            g =~          VW2  0.149 -0.021  -0.021   -0.021   -0.021
## 137            g =~          VW3  3.638 -0.113  -0.113   -0.109   -0.109
## 138            g =~          VM1  0.146 -0.022  -0.022   -0.022   -0.022
## 139            g =~          VM2  0.036  0.011   0.011    0.011    0.011
## 140            g =~          VM3  0.130  0.021   0.021    0.021    0.021
## 141            g =~          SO1  5.127 -0.102  -0.102   -0.101   -0.101
## 142            g =~          SO2  0.492  0.030   0.030    0.030    0.030
## 143            g =~          SO3  0.186  0.020   0.020    0.019    0.019
## 144            g =~          SW1  0.113  0.014   0.014    0.014    0.014
## 145            g =~          SW2  4.881  0.094   0.094    0.092    0.092
## 146            g =~          SW3  9.409 -0.131  -0.131   -0.129   -0.129
## 147            g =~          SM1  1.047 -0.047  -0.047   -0.046   -0.046
## 148            g =~          SM2  0.880  0.043   0.043    0.043    0.043
## 149            g =~          SM3  2.756  0.074   0.074    0.074    0.074
## 150            g =~          QO1  0.578  0.059   0.059    0.060    0.060
## 151            g =~          QO2  0.320 -0.044  -0.044   -0.044   -0.044
## 152            g =~          QO3  0.012 -0.009  -0.009   -0.009   -0.009
## 153            g =~          QW1  0.147 -0.030  -0.030   -0.030   -0.030
## 154            g =~          QW2  0.007 -0.007  -0.007   -0.007   -0.007
## 155            g =~          QW3  0.978 -0.079  -0.079   -0.076   -0.076
## 156            g =~          QM1  0.004  0.005   0.005    0.005    0.005
## 157            g =~          QM2  0.639  0.065   0.065    0.064    0.064
## 158            g =~          QM3  0.258  0.042   0.042    0.041    0.041
## 159       Verbal =~          SO1  3.250 -0.034  -0.058   -0.057   -0.057
## 160       Verbal =~          SO2  0.092  0.005   0.009    0.009    0.009
## 161       Verbal =~          SO3  1.128  0.020   0.034    0.034    0.034
## 162       Verbal =~          SW1  1.474  0.022   0.037    0.037    0.037
## 163       Verbal =~          SW2  0.122  0.006   0.011    0.010    0.010
## 164       Verbal =~          SW3  3.446 -0.034  -0.056   -0.055   -0.055
## 165       Verbal =~          SM1  2.347 -0.030  -0.050   -0.049   -0.049
## 166       Verbal =~          SM2  0.882  0.018   0.031    0.030    0.030
## 167       Verbal =~          SM3  1.747  0.025   0.042    0.042    0.042
## 168       Verbal =~          QO1  0.698  0.018   0.030    0.030    0.030
## 169       Verbal =~          QO2  0.570 -0.016  -0.027   -0.027   -0.027
## 170       Verbal =~          QO3  1.012  0.023   0.038    0.036    0.036
```

```
## 171       Verbal =~          QW1  0.083 -0.006  -0.010   -0.010   -0.010
## 172       Verbal =~          QW2  0.038 -0.004  -0.007   -0.007   -0.007
## 173       Verbal =~          QW3  1.533 -0.027  -0.046   -0.044   -0.044
## 174       Verbal =~          QM1  0.075 -0.006  -0.010   -0.010   -0.010
## 175       Verbal =~          QM2  0.392  0.014   0.023    0.023    0.023
## 176       Verbal =~          QM3  0.078  0.006   0.011    0.010    0.010
## 177      Spatial =~          VO1  0.300 -0.012  -0.018   -0.017   -0.017
## 178      Spatial =~          VO2  3.219  0.039   0.057    0.055    0.055
## 179      Spatial =~          VO3  2.245  0.033   0.048    0.047    0.047
## 180      Spatial =~          VW1  5.330 -0.049  -0.070   -0.071   -0.071
## 181      Spatial =~          VW2  0.983  0.021   0.030    0.029    0.029
## 182      Spatial =~          VW3  0.150 -0.009  -0.013   -0.012   -0.012
## 183      Spatial =~          VM1  0.668 -0.018  -0.026   -0.026   -0.026
## 184      Spatial =~          VM2  0.776  0.020   0.028    0.028    0.028
## 185      Spatial =~          VM3  1.486 -0.027  -0.039   -0.038   -0.038
## 186      Spatial =~          QO1  0.079  0.006   0.009    0.009    0.009
## 187      Spatial =~          QO2  0.001 -0.001  -0.001   -0.001   -0.001
## 188      Spatial =~          QO3  2.522 -0.037  -0.054   -0.051   -0.051
## 189      Spatial =~          QW1  0.105 -0.007  -0.011   -0.011   -0.011
## 190      Spatial =~          QW2  0.012  0.002   0.004    0.003    0.003
## 191      Spatial =~          QW3  0.022 -0.003  -0.005   -0.005   -0.005
## 192      Spatial =~          QM1  0.247  0.011   0.016    0.016    0.016
## 193      Spatial =~          QM2  0.411  0.015   0.022    0.021    0.021
## 194      Spatial =~          QM3  0.323  0.013   0.019    0.019    0.019
## 195        Quant =~          VO1  0.049  0.004   0.008    0.008    0.008
## 196        Quant =~          VO2  4.030  0.038   0.074    0.072    0.072
## 197        Quant =~          VO3  0.146  0.007   0.014    0.014    0.014
## 198        Quant =~          VW1  0.042 -0.004  -0.007   -0.007   -0.007
## 199        Quant =~          VW2  1.019 -0.018  -0.036   -0.035   -0.035
## 200        Quant =~          VW3  5.061 -0.043  -0.085   -0.081   -0.081
## 201        Quant =~          VM1  0.004 -0.001  -0.002   -0.002   -0.002
## 202        Quant =~          VM2  0.046 -0.004  -0.008   -0.008   -0.008
## 203        Quant =~          VM3  1.200  0.021   0.041    0.040    0.040
## 204        Quant =~          SO1  4.575 -0.037  -0.072   -0.071   -0.071
## 205        Quant =~          SO2  0.714  0.014   0.027    0.027    0.027
## 206        Quant =~          SO3  0.012 -0.002  -0.004   -0.004   -0.004
## 207        Quant =~          SW1  0.130 -0.006  -0.011   -0.011   -0.011
## 208        Quant =~          SW2  9.555  0.050   0.098    0.096    0.096
## 209        Quant =~          SW3 10.961 -0.054  -0.106   -0.104   -0.104
## 210        Quant =~          SM1  0.206 -0.008  -0.015   -0.015   -0.015
## 211        Quant =~          SM2  0.561  0.013   0.026    0.025    0.025
## 212        Quant =~          SM3  2.455  0.026   0.052    0.052    0.052
## 213         Oral =~          VW1  0.178 -0.016  -0.016   -0.016   -0.016
## 214         Oral =~          VW2  0.029  0.006   0.006    0.006    0.006
## 215         Oral =~          VW3  0.166  0.016   0.016    0.016    0.016
## 216         Oral =~          VM1  1.000 -0.042  -0.042   -0.043   -0.043
```

```
## 217          Oral =~          VM2  3.318  0.080   0.080    0.078    0.078
## 218          Oral =~          VM3  1.317 -0.050  -0.050   -0.049   -0.049
## 219          Oral =~          SW1  1.248  0.042   0.042    0.043    0.043
## 220          Oral =~          SW2  0.005 -0.003  -0.003   -0.003   -0.003
## 221          Oral =~          SW3  0.247 -0.019  -0.019   -0.018   -0.018
## 222          Oral =~          SM1  0.280 -0.023  -0.023   -0.023   -0.023
## 223          Oral =~          SM2  0.001  0.001   0.001    0.001    0.001
## 224          Oral =~          SM3  0.163  0.017   0.017    0.017    0.017
## 225          Oral =~          QW1  0.638 -0.031  -0.031   -0.031   -0.031
## 226          Oral =~          QW2  0.024 -0.006  -0.006   -0.006   -0.006
## 227          Oral =~          QW3  0.724 -0.033  -0.033   -0.032   -0.032
## 228          Oral =~          QM1  0.000  0.000   0.000    0.000    0.000
## 229          Oral =~          QM2  6.792  0.116   0.116    0.114    0.114
## 230          Oral =~          QM3  0.112 -0.015  -0.015   -0.015   -0.015
## 231       Written =~          VO1  0.080 -0.010  -0.010   -0.010   -0.010
## 232       Written =~          VO2  0.352 -0.021  -0.021   -0.021   -0.021
## 233       Written =~          VO3  0.347  0.021   0.021    0.021    0.021
## 234       Written =~          VM1  0.202  0.016   0.016    0.017    0.017
## 235       Written =~          VM2  1.040 -0.039  -0.039   -0.038   -0.038
## 236       Written =~          VM3  1.157  0.041   0.041    0.039    0.039
## 237       Written =~          SO1  0.815 -0.033  -0.033   -0.032   -0.032
## 238       Written =~          SO2  0.115  0.012   0.012    0.012    0.012
## 239       Written =~          SO3  0.357 -0.022  -0.022   -0.021   -0.021
## 240       Written =~          SM1  0.455  0.026   0.026    0.025    0.025
## 241       Written =~          SM2  0.010  0.004   0.004    0.004    0.004
## 242       Written =~          SM3  0.298  0.020   0.020    0.020    0.020
## 243       Written =~          QO1  0.639  0.028   0.028    0.028    0.028
## 244       Written =~          QO2  0.605  0.027   0.027    0.027    0.027
## 245       Written =~          QO3  2.379 -0.056  -0.056   -0.052   -0.052
## 246       Written =~          QM1  0.452  0.025   0.025    0.025    0.025
## 247       Written =~          QM2  6.268 -0.095  -0.095   -0.093   -0.093
## 248       Written =~          QM3  0.080 -0.011  -0.011   -0.011   -0.011
## 249 Manipulative =~          VO1  0.095 -0.017  -0.017   -0.016   -0.016
## 250 Manipulative =~          VO2  1.082  0.055   0.055    0.054    0.054
## 251 Manipulative =~          VO3  0.317  0.030   0.030    0.029    0.029
## 252 Manipulative =~          VW1  0.125 -0.017  -0.017   -0.017   -0.017
## 253 Manipulative =~          VW2  0.002  0.002   0.002    0.002    0.002
## 254 Manipulative =~          VW3  1.780 -0.069  -0.069   -0.066   -0.066
## 255 Manipulative =~          SO1  0.053 -0.012  -0.012   -0.012   -0.012
## 256 Manipulative =~          SO2  0.218 -0.024  -0.024   -0.025   -0.025
## 257 Manipulative =~          SO3  0.313  0.030   0.030    0.030    0.030
## 258 Manipulative =~          SW1  1.148 -0.053  -0.053   -0.053   -0.053
## 259 Manipulative =~          SW2  1.086  0.051   0.051    0.050    0.050
## 260 Manipulative =~          SW3  0.533 -0.036  -0.036   -0.035   -0.035
## 261 Manipulative =~          QO1  1.388 -0.060  -0.060   -0.061   -0.061
## 262 Manipulative =~          QO2  0.555 -0.038  -0.038   -0.038   -0.038
```

```
## 263 Manipulative =~          QO3  3.235  0.096   0.096    0.090    0.090
## 264 Manipulative =~          QW1  2.734  0.081   0.081    0.081    0.081
## 265 Manipulative =~          QW2  0.215 -0.023  -0.023   -0.023   -0.023
## 266 Manipulative =~          QW3  0.828  0.045   0.045    0.043    0.043
## 267          VO1 ~~          VO2  1.907 -0.025  -0.025   -0.055   -0.055
## 268          VO1 ~~          VO3  0.797 -0.017  -0.017   -0.036   -0.036
## 269          VO1 ~~          VW1  0.968  0.016   0.016    0.037    0.037
## 270          VO1 ~~          VW2  1.050 -0.017  -0.017   -0.039   -0.039
## 271          VO1 ~~          VW3  2.297  0.027   0.027    0.056    0.056
## 272          VO1 ~~          VM1  0.478  0.012   0.012    0.026    0.026
## 273          VO1 ~~          VM2  0.775  0.016   0.016    0.033    0.033
## 274          VO1 ~~          VM3  0.495 -0.012  -0.012   -0.027   -0.027
## 275          VO1 ~~          SO1  0.056  0.004   0.004    0.009    0.009
## 276          VO1 ~~          SO2  0.595 -0.013  -0.013   -0.029   -0.029
## 277          VO1 ~~          SO3  1.237  0.020   0.020    0.042    0.042
## 278          VO1 ~~          SW1  2.001  0.023   0.023    0.052    0.052
## 279          VO1 ~~          SW2  0.090 -0.005  -0.005   -0.011   -0.011
## 280          VO1 ~~          SW3  2.754 -0.027  -0.027   -0.062   -0.062
## 281          VO1 ~~          SM1  1.816 -0.023  -0.023   -0.050   -0.050
## 282          VO1 ~~          SM2  0.000  0.000   0.000   -0.001   -0.001
## 283          VO1 ~~          SM3  0.452  0.011   0.011    0.025    0.025
## 284          VO1 ~~          QO1  2.281  0.025   0.025    0.056    0.056
## 285          VO1 ~~          QO2  0.065 -0.004  -0.004   -0.010   -0.010
## 286          VO1 ~~          QO3  0.142  0.007   0.007    0.015    0.015
## 287          VO1 ~~          QW1  0.447 -0.011  -0.011   -0.024   -0.024
## 288          VO1 ~~          QW2  0.516  0.012   0.012    0.026    0.026
## 289          VO1 ~~          QW3  0.263 -0.009  -0.009   -0.019   -0.019
## 290          VO1 ~~          QM1  0.622 -0.013  -0.013   -0.029   -0.029
## 291          VO1 ~~          QM2  0.207  0.008   0.008    0.017    0.017
## 292          VO1 ~~          QM3  0.037 -0.003  -0.003   -0.007   -0.007
## 293          VO2 ~~          VO3  0.191 -0.008  -0.008   -0.018   -0.018
## 294          VO2 ~~          VW1  0.382 -0.010  -0.010   -0.023   -0.023
## 295          VO2 ~~          VW2  0.680  0.013   0.013    0.032    0.032
## 296          VO2 ~~          VW3  0.081  0.005   0.005    0.011    0.011
## 297          VO2 ~~          VM1  1.336  0.019   0.019    0.044    0.044
## 298          VO2 ~~          VM2  0.056  0.004   0.004    0.009    0.009
## 299          VO2 ~~          VM3  5.515 -0.040  -0.040   -0.090   -0.090
## 300          VO2 ~~          SO1  0.754  0.015   0.015    0.033    0.033
## 301          VO2 ~~          SO2  0.100  0.005   0.005    0.012    0.012
## 302          VO2 ~~          SO3  0.678 -0.014  -0.014   -0.031   -0.031
## 303          VO2 ~~          SW1  0.883  0.015   0.015    0.035    0.035
## 304          VO2 ~~          SW2  1.727 -0.021  -0.021   -0.049   -0.049
## 305          VO2 ~~          SW3  1.299  0.018   0.018    0.043    0.043
## 306          VO2 ~~          SM1  0.107  0.006   0.006    0.012    0.012
## 307          VO2 ~~          SM2  0.546  0.013   0.013    0.027    0.027
## 308          VO2 ~~          SM3  0.598 -0.013  -0.013   -0.029   -0.029
```

```
## 309          VO2 ~~          QO1  6.884  0.043   0.043    0.099    0.099
## 310          VO2 ~~          QO2  1.087  0.018   0.018    0.041    0.041
## 311          VO2 ~~          QO3  3.934 -0.036  -0.036   -0.080   -0.080
## 312          VO2 ~~          QW1  1.357 -0.019  -0.019   -0.043   -0.043
## 313          VO2 ~~          QW2  0.718 -0.014  -0.014   -0.031   -0.031
## 314          VO2 ~~          QW3  0.032 -0.003  -0.003   -0.007   -0.007
## 315          VO2 ~~          QM1  0.095  0.005   0.005    0.012    0.012
## 316          VO2 ~~          QM2  0.005  0.001   0.001    0.003    0.003
## 317          VO2 ~~          QM3  4.065  0.034   0.034    0.075    0.075
## 318          VO3 ~~          VW1  1.719  0.021   0.021    0.049    0.049
## 319          VO3 ~~          VW2  0.616 -0.013  -0.013   -0.030   -0.030
## 320          VO3 ~~          VW3  0.026 -0.003  -0.003   -0.006   -0.006
## 321          VO3 ~~          VM1  0.831 -0.015  -0.015   -0.035   -0.035
## 322          VO3 ~~          VM2  1.142  0.018   0.018    0.040    0.040
## 323          VO3 ~~          VM3  0.004 -0.001  -0.001   -0.002   -0.002
## 324          VO3 ~~          SO1  0.156  0.007   0.007    0.015    0.015
## 325          VO3 ~~          SO2  0.143 -0.006  -0.006   -0.015   -0.015
## 326          VO3 ~~          SO3  0.006 -0.001  -0.001   -0.003   -0.003
## 327          VO3 ~~          SW1  1.293 -0.018  -0.018   -0.042   -0.042
## 328          VO3 ~~          SW2  0.814  0.014   0.014    0.034    0.034
## 329          VO3 ~~          SW3  0.056  0.004   0.004    0.009    0.009
## 330          VO3 ~~          SM1  5.546  0.040   0.040    0.087    0.087
## 331          VO3 ~~          SM2  0.046 -0.004  -0.004   -0.008   -0.008
## 332          VO3 ~~          SM3  0.083 -0.005  -0.005   -0.011   -0.011
## 333          VO3 ~~          QO1  0.032  0.003   0.003    0.007    0.007
## 334          VO3 ~~          QO2  0.055  0.004   0.004    0.010    0.010
## 335          VO3 ~~          QO3  0.440  0.012   0.012    0.027    0.027
## 336          VO3 ~~          QW1  0.042  0.003   0.003    0.008    0.008
## 337          VO3 ~~          QW2  0.169 -0.007  -0.007   -0.015   -0.015
## 338          VO3 ~~          QW3  0.191  0.007   0.007    0.016    0.016
## 339          VO3 ~~          QM1  0.001  0.000   0.000    0.001    0.001
## 340          VO3 ~~          QM2  3.319  0.031   0.031    0.067    0.067
## 341          VO3 ~~          QM3 11.327 -0.058  -0.058   -0.124   -0.124
## 342          VW1 ~~          VW2  0.330  0.010   0.010    0.025    0.025
## 343          VW1 ~~          VW3  0.060  0.004   0.004    0.010    0.010
## 344          VW1 ~~          VM1  0.459 -0.011  -0.011   -0.026   -0.026
## 345          VW1 ~~          VM2  0.406 -0.010  -0.010   -0.024   -0.024
## 346          VW1 ~~          VM3  0.004 -0.001  -0.001   -0.002   -0.002
## 347          VW1 ~~          SO1  2.964 -0.028  -0.028   -0.063   -0.063
## 348          VW1 ~~          SO2  1.281  0.017   0.017    0.042    0.042
## 349          VW1 ~~          SO3  0.591 -0.012  -0.012   -0.028   -0.028
## 350          VW1 ~~          SW1  0.037 -0.003  -0.003   -0.007   -0.007
## 351          VW1 ~~          SW2  0.222  0.007   0.007    0.019    0.019
## 352          VW1 ~~          SW3 11.435 -0.054  -0.054   -0.134   -0.134
## 353          VW1 ~~          SM1  1.951  0.023   0.023    0.052    0.052
## 354          VW1 ~~          SM2  0.563  0.012   0.012    0.027    0.027
```

```
## 355          VW1 ~~          SM3  0.379 -0.010  -0.010   -0.023   -0.023
## 356          VW1 ~~          QO1  0.006  0.001   0.001    0.003    0.003
## 357          VW1 ~~          QO2  0.496 -0.011  -0.011   -0.026   -0.026
## 358          VW1 ~~          QO3  0.081 -0.005  -0.005   -0.011   -0.011
## 359          VW1 ~~          QW1  0.723  0.014   0.014    0.033    0.033
## 360          VW1 ~~          QW2  1.722  0.021   0.021    0.049    0.049
## 361          VW1 ~~          QW3  0.386  0.010   0.010    0.025    0.025
## 362          VW1 ~~          QM1  0.577 -0.012  -0.012   -0.028   -0.028
## 363          VW1 ~~          QM2  1.464 -0.019  -0.019   -0.044   -0.044
## 364          VW1 ~~          QM3  1.348  0.019   0.019    0.043    0.043
## 365          VW2 ~~          VW3  0.487 -0.012  -0.012   -0.029   -0.029
## 366          VW2 ~~          VM1  0.000  0.000   0.000    0.000    0.000
## 367          VW2 ~~          VM2  0.049 -0.004  -0.004   -0.009   -0.009
## 368          VW2 ~~          VM3  2.425  0.025   0.025    0.061    0.061
## 369          VW2 ~~          SO1  1.222 -0.018  -0.018   -0.042   -0.042
## 370          VW2 ~~          SO2  2.123  0.022   0.022    0.056    0.056
## 371          VW2 ~~          SO3  0.001 -0.001  -0.001   -0.001   -0.001
## 372          VW2 ~~          SW1  1.433  0.019   0.019    0.047    0.047
## 373          VW2 ~~          SW2  1.799 -0.021  -0.021   -0.055   -0.055
## 374          VW2 ~~          SW3  1.760  0.021   0.021    0.055    0.055
## 375          VW2 ~~          SM1  0.014 -0.002  -0.002   -0.004   -0.004
## 376          VW2 ~~          SM2  0.174  0.007   0.007    0.016    0.016
## 377          VW2 ~~          SM3  0.021 -0.002  -0.002   -0.006   -0.006
## 378          VW2 ~~          QO1  0.064  0.004   0.004    0.009    0.009
## 379          VW2 ~~          QO2  0.080 -0.004  -0.004   -0.011   -0.011
## 380          VW2 ~~          QO3  0.743  0.014   0.014    0.033    0.033
## 381          VW2 ~~          QW1  0.049  0.004   0.004    0.009    0.009
## 382          VW2 ~~          QW2  3.604 -0.030  -0.030   -0.074   -0.074
## 383          VW2 ~~          QW3  0.087  0.005   0.005    0.012    0.012
## 384          VW2 ~~          QM1  0.094 -0.005  -0.005   -0.012   -0.012
## 385          VW2 ~~          QM2  0.067 -0.004  -0.004   -0.010   -0.010
## 386          VW2 ~~          QM3  0.626 -0.013  -0.013   -0.030   -0.030
## 387          VW3 ~~          VM1  1.755  0.023   0.023    0.050    0.050
## 388          VW3 ~~          VM2  0.778 -0.016  -0.016   -0.033   -0.033
## 389          VW3 ~~          VM3  0.037  0.003   0.003    0.007    0.007
## 390          VW3 ~~          SO1  2.091  0.025   0.025    0.053    0.053
## 391          VW3 ~~          SO2  0.431 -0.011  -0.011   -0.024   -0.024
## 392          VW3 ~~          SO3  0.166  0.007   0.007    0.015    0.015
## 393          VW3 ~~          SW1  0.252 -0.008  -0.008   -0.019   -0.019
## 394          VW3 ~~          SW2  0.330  0.010   0.010    0.022    0.022
## 395          VW3 ~~          SW3  2.497  0.027   0.027    0.061    0.061
## 396          VW3 ~~          SM1  4.907 -0.038  -0.038   -0.081   -0.081
## 397          VW3 ~~          SM2  1.840 -0.024  -0.024   -0.049   -0.049
## 398          VW3 ~~          SM3  1.395  0.020   0.020    0.044    0.044
## 399          VW3 ~~          QO1  0.742  0.014   0.014    0.031    0.031
## 400          VW3 ~~          QO2  0.070 -0.004  -0.004   -0.010   -0.010
```

```
## 401          VW3 ~~          QO3  3.769 -0.033  -0.033   -0.072   -0.072
## 402          VW3 ~~          QW1  0.001  0.000   0.000    0.001    0.001
## 403          VW3 ~~          QW2  0.243  0.008   0.008    0.018    0.018
## 404          VW3 ~~          QW3  2.987 -0.030  -0.030   -0.067   -0.067
## 405          VW3 ~~          QM1  0.568 -0.013  -0.013   -0.028   -0.028
## 406          VW3 ~~          QM2  0.021  0.003   0.003    0.005    0.005
## 407          VW3 ~~          QM3  0.079  0.005   0.005    0.010    0.010
## 408          VM1 ~~          VM2  2.322 -0.028  -0.028   -0.062   -0.062
## 409          VM1 ~~          VM3  0.037  0.003   0.003    0.008    0.008
## 410          VM1 ~~          SO1  0.558 -0.012  -0.012   -0.028   -0.028
## 411          VM1 ~~          SO2  0.000  0.000   0.000   -0.001   -0.001
## 412          VM1 ~~          SO3  0.058  0.004   0.004    0.009    0.009
## 413          VM1 ~~          SW1  0.002  0.001   0.001    0.002    0.002
## 414          VM1 ~~          SW2  0.903 -0.015  -0.015   -0.036   -0.036
## 415          VM1 ~~          SW3  3.798  0.030   0.030    0.073    0.073
## 416          VM1 ~~          SM1  2.771 -0.030  -0.030   -0.067   -0.067
## 417          VM1 ~~          SM2  2.005 -0.025  -0.025   -0.054   -0.054
## 418          VM1 ~~          SM3  1.574  0.022   0.022    0.051    0.051
## 419          VM1 ~~          QO1  2.190 -0.023  -0.023   -0.054   -0.054
## 420          VM1 ~~          QO2  2.030 -0.023  -0.023   -0.053   -0.053
## 421          VM1 ~~          QO3  1.348  0.019   0.019    0.044    0.044
## 422          VM1 ~~          QW1  0.977 -0.016  -0.016   -0.036   -0.036
## 423          VM1 ~~          QW2  0.011 -0.002  -0.002   -0.004   -0.004
## 424          VM1 ~~          QW3  0.001  0.000   0.000    0.001    0.001
## 425          VM1 ~~          QM1  0.462  0.011   0.011    0.027    0.027
## 426          VM1 ~~          QM2  0.761  0.015   0.015    0.034    0.034
## 427          VM1 ~~          QM3  2.708  0.029   0.029    0.063    0.063
## 428          VM2 ~~          VM3  0.757  0.016   0.016    0.035    0.035
## 429          VM2 ~~          SO1  0.462  0.012   0.012    0.025    0.025
## 430          VM2 ~~          SO2  6.304 -0.041  -0.041   -0.094   -0.094
## 431          VM2 ~~          SO3  1.817  0.023   0.023    0.049    0.049
## 432          VM2 ~~          SW1  0.315  0.009   0.009    0.021    0.021
## 433          VM2 ~~          SW2  0.364 -0.010  -0.010   -0.023   -0.023
## 434          VM2 ~~          SW3  0.017  0.002   0.002    0.005    0.005
## 435          VM2 ~~          SM1  0.122  0.006   0.006    0.014    0.014
## 436          VM2 ~~          SM2  1.439  0.022   0.022    0.045    0.045
## 437          VM2 ~~          SM3  0.229  0.009   0.009    0.019    0.019
## 438          VM2 ~~          QO1  0.198 -0.007  -0.007   -0.016   -0.016
## 439          VM2 ~~          QO2  0.539 -0.012  -0.012   -0.027   -0.027
## 440          VM2 ~~          QO3  3.140  0.030   0.030    0.066    0.066
## 441          VM2 ~~          QW1  1.869  0.023   0.023    0.050    0.050
## 442          VM2 ~~          QW2  0.584 -0.013  -0.013   -0.028   -0.028
## 443          VM2 ~~          QW3  0.213 -0.008  -0.008   -0.017   -0.017
## 444          VM2 ~~          QM1  0.235  0.008   0.008    0.019    0.019
## 445          VM2 ~~          QM2  0.650 -0.014  -0.014   -0.030   -0.030
## 446          VM2 ~~          QM3  1.506 -0.022  -0.022   -0.046   -0.046
```

```
## 447          VM3 ~~          SO1  0.881 -0.016  -0.016   -0.035   -0.035
## 448          VM3 ~~          SO2  0.976  0.016   0.016    0.037    0.037
## 449          VM3 ~~          SO3  0.126  0.006   0.006    0.013    0.013
## 450          VM3 ~~          SW1  0.079 -0.005  -0.005   -0.010   -0.010
## 451          VM3 ~~          SW2  0.636  0.013   0.013    0.030    0.030
## 452          VM3 ~~          SW3  2.936 -0.027  -0.027   -0.065   -0.065
## 453          VM3 ~~          SM1  1.463 -0.022  -0.022   -0.048   -0.048
## 454          VM3 ~~          SM2  0.381  0.011   0.011    0.023    0.023
## 455          VM3 ~~          SM3  0.667 -0.015  -0.015   -0.033   -0.033
## 456          VM3 ~~          QO1  5.929 -0.039  -0.039   -0.090   -0.090
## 457          VM3 ~~          QO2  2.012  0.023   0.023    0.053    0.053
## 458          VM3 ~~          QO3  1.254  0.019   0.019    0.042    0.042
## 459          VM3 ~~          QW1  0.002 -0.001  -0.001   -0.002   -0.002
## 460          VM3 ~~          QW2  1.860  0.022   0.022    0.050    0.050
## 461          VM3 ~~          QW3  0.010 -0.002  -0.002   -0.004   -0.004
## 462          VM3 ~~          QM1  0.360  0.010   0.010    0.023    0.023
## 463          VM3 ~~          QM2  0.163 -0.007  -0.007   -0.015   -0.015
## 464          VM3 ~~          QM3  0.346  0.010   0.010    0.022    0.022
## 465          SO1 ~~          SO2  0.001  0.001   0.001    0.001    0.001
## 466          SO1 ~~          SO3  1.620  0.024   0.024    0.050    0.050
## 467          SO1 ~~          SW1  0.071 -0.004  -0.004   -0.010   -0.010
## 468          SO1 ~~          SW2  0.748 -0.014  -0.014   -0.033   -0.033
## 469          SO1 ~~          SW3  7.146  0.044   0.044    0.102    0.102
## 470          SO1 ~~          SM1  0.385 -0.011  -0.011   -0.023   -0.023
## 471          SO1 ~~          SM2  0.285 -0.010  -0.010   -0.020   -0.020
## 472          SO1 ~~          SM3  0.009  0.002   0.002    0.004    0.004
## 473          SO1 ~~          QO1  0.827 -0.015  -0.015   -0.034   -0.034
## 474          SO1 ~~          QO2  0.008 -0.002  -0.002   -0.004   -0.004
## 475          SO1 ~~          QO3  1.802 -0.025  -0.025   -0.054   -0.054
## 476          SO1 ~~          QW1  0.397 -0.010  -0.010   -0.023   -0.023
## 477          SO1 ~~          QW2  0.946 -0.016  -0.016   -0.036   -0.036
## 478          SO1 ~~          QW3  0.003  0.001   0.001    0.002    0.002
## 479          SO1 ~~          QM1  0.062 -0.004  -0.004   -0.009   -0.009
## 480          SO1 ~~          QM2  0.123  0.006   0.006    0.013    0.013
## 481          SO1 ~~          QM3  3.670  0.033   0.033    0.070    0.070
## 482          SO2 ~~          SO3  1.452  0.021   0.021    0.049    0.049
## 483          SO2 ~~          SW1  0.831 -0.014  -0.014   -0.035   -0.035
## 484          SO2 ~~          SW2  0.639  0.013   0.013    0.031    0.031
## 485          SO2 ~~          SW3  0.009 -0.001  -0.001   -0.004   -0.004
## 486          SO2 ~~          SM1  0.011 -0.002  -0.002   -0.004   -0.004
## 487          SO2 ~~          SM2  0.649 -0.014  -0.014   -0.031   -0.031
## 488          SO2 ~~          SM3  0.425 -0.011  -0.011   -0.025   -0.025
## 489          SO2 ~~          QO1  0.133  0.006   0.006    0.014    0.014
## 490          SO2 ~~          QO2  1.111 -0.017  -0.017   -0.042   -0.042
## 491          SO2 ~~          QO3  0.274  0.009   0.009    0.021    0.021
## 492          SO2 ~~          QW1  0.504 -0.011  -0.011   -0.026   -0.026
```

```
## 493          SO2 ~~          QW2  0.607  0.012   0.012    0.029    0.029
## 494          SO2 ~~          QW3  2.039 -0.023  -0.023   -0.054   -0.054
## 495          SO2 ~~          QM1  1.283  0.018   0.018    0.043    0.043
## 496          SO2 ~~          QM2  1.457  0.020   0.020    0.045    0.045
## 497          SO2 ~~          QM3  0.024  0.003   0.003    0.006    0.006
## 498          SO3 ~~          SW1  1.412 -0.020  -0.020   -0.045   -0.045
## 499          SO3 ~~          SW2  1.564 -0.021  -0.021   -0.048   -0.048
## 500          SO3 ~~          SW3  0.092  0.005   0.005    0.012    0.012
## 501          SO3 ~~          SM1  0.442 -0.012  -0.012   -0.025   -0.025
## 502          SO3 ~~          SM2  0.019 -0.003  -0.003   -0.005   -0.005
## 503          SO3 ~~          SM3  0.087  0.005   0.005    0.011    0.011
## 504          SO3 ~~          QO1  3.037 -0.029  -0.029   -0.065   -0.065
## 505          SO3 ~~          QO2  1.342 -0.020  -0.020   -0.046   -0.046
## 506          SO3 ~~          QO3  0.305  0.010   0.010    0.022    0.022
## 507          SO3 ~~          QW1  0.044 -0.003  -0.003   -0.008   -0.008
## 508          SO3 ~~          QW2  0.391  0.010   0.010    0.023    0.023
## 509          SO3 ~~          QW3  1.515  0.021   0.021    0.046    0.046
## 510          SO3 ~~          QM1  0.786 -0.015  -0.015   -0.033   -0.033
## 511          SO3 ~~          QM2  0.648  0.014   0.014    0.030    0.030
## 512          SO3 ~~          QM3  0.141 -0.007  -0.007   -0.014   -0.014
## 513          SW1 ~~          SW2  0.031 -0.003  -0.003   -0.007   -0.007
## 514          SW1 ~~          SW3  2.218  0.025   0.025    0.062    0.062
## 515          SW1 ~~          SM1  0.244  0.008   0.008    0.019    0.019
## 516          SW1 ~~          SM2  2.070  0.024   0.024    0.054    0.054
## 517          SW1 ~~          SM3  1.037 -0.017  -0.017   -0.039   -0.039
## 518          SW1 ~~          QO1  5.484  0.036   0.036    0.086    0.086
## 519          SW1 ~~          QO2  2.717  0.025   0.025    0.061    0.061
## 520          SW1 ~~          QO3  1.127 -0.017  -0.017   -0.040   -0.040
## 521          SW1 ~~          QW1  1.031 -0.016  -0.016   -0.039   -0.039
## 522          SW1 ~~          QW2  1.392  0.019   0.019    0.044    0.044
## 523          SW1 ~~          QW3  3.535 -0.031  -0.031   -0.073   -0.073
## 524          SW1 ~~          QM1  1.693 -0.020  -0.020   -0.048   -0.048
## 525          SW1 ~~          QM2  0.273 -0.008  -0.008   -0.019   -0.019
## 526          SW1 ~~          QM3  0.974 -0.016  -0.016   -0.036   -0.036
## 527          SW2 ~~          SW3  0.180 -0.007  -0.007   -0.019   -0.019
## 528          SW2 ~~          SM1  0.854  0.015   0.015    0.036    0.036
## 529          SW2 ~~          SM2  2.478 -0.026  -0.026   -0.060   -0.060
## 530          SW2 ~~          SM3  0.344  0.010   0.010    0.023    0.023
## 531          SW2 ~~          QO1  0.344 -0.009  -0.009   -0.022   -0.022
## 532          SW2 ~~          QO2  2.797  0.026   0.026    0.063    0.063
## 533          SW2 ~~          QO3  0.014  0.002   0.002    0.005    0.005
## 534          SW2 ~~          QW1  0.045 -0.003  -0.003   -0.008   -0.008
## 535          SW2 ~~          QW2  0.004  0.001   0.001    0.003    0.003
## 536          SW2 ~~          QW3  1.299  0.019   0.019    0.046    0.046
## 537          SW2 ~~          QM1  0.022  0.002   0.002    0.006    0.006
## 538          SW2 ~~          QM2  2.190  0.024   0.024    0.055    0.055
```

```
## 539          SW2 ~~          QM3  0.492  0.011   0.011    0.026    0.026
## 540          SW3 ~~          SM1  0.830 -0.015  -0.015   -0.035   -0.035
## 541          SW3 ~~          SM2  0.273  0.009   0.009    0.020    0.020
## 542          SW3 ~~          SM3  0.904 -0.015  -0.015   -0.037   -0.037
## 543          SW3 ~~          QO1  2.659 -0.025  -0.025   -0.061   -0.061
## 544          SW3 ~~          QO2  0.000  0.000   0.000    0.000    0.000
## 545          SW3 ~~          QO3  1.595 -0.020  -0.020   -0.048   -0.048
## 546          SW3 ~~          QW1  0.033 -0.003  -0.003   -0.007   -0.007
## 547          SW3 ~~          QW2  0.970 -0.016  -0.016   -0.038   -0.038
## 548          SW3 ~~          QW3  0.365  0.010   0.010    0.025    0.025
## 549          SW3 ~~          QM1  3.554  0.029   0.029    0.072    0.072
## 550          SW3 ~~          QM2  2.167 -0.024  -0.024   -0.055   -0.055
## 551          SW3 ~~          QM3  0.362 -0.010  -0.010   -0.023   -0.023
## 552          SM1 ~~          SM2  0.862  0.018   0.018    0.038    0.038
## 553          SM1 ~~          SM3  0.967  0.020   0.020    0.045    0.045
## 554          SM1 ~~          QO1  0.016 -0.002  -0.002   -0.005   -0.005
## 555          SM1 ~~          QO2  1.847  0.022   0.022    0.051    0.051
## 556          SM1 ~~          QO3  4.344 -0.036  -0.036   -0.079   -0.079
## 557          SM1 ~~          QW1  3.334  0.030   0.030    0.067    0.067
## 558          SM1 ~~          QW2  2.272 -0.025  -0.025   -0.056   -0.056
## 559          SM1 ~~          QW3  0.284  0.009   0.009    0.020    0.020
## 560          SM1 ~~          QM1  0.001 -0.001  -0.001   -0.001   -0.001
## 561          SM1 ~~          QM2  0.753  0.016   0.016    0.034    0.034
## 562          SM1 ~~          QM3  0.005 -0.001  -0.001   -0.003   -0.003
## 563          SM2 ~~          SM3  0.222 -0.009  -0.009   -0.020   -0.020
## 564          SM2 ~~          QO1  1.753  0.022   0.022    0.048    0.048
## 565          SM2 ~~          QO2  0.787 -0.015  -0.015   -0.033   -0.033
## 566          SM2 ~~          QO3  0.335  0.010   0.010    0.022    0.022
## 567          SM2 ~~          QW1  0.094 -0.005  -0.005   -0.011   -0.011
## 568          SM2 ~~          QW2  0.800  0.015   0.015    0.033    0.033
## 569          SM2 ~~          QW3  0.217 -0.008  -0.008   -0.017   -0.017
## 570          SM2 ~~          QM1  0.000  0.000   0.000    0.000    0.000
## 571          SM2 ~~          QM2  1.930 -0.025  -0.025   -0.052   -0.052
## 572          SM2 ~~          QM3  0.325  0.010   0.010    0.021    0.021
## 573          SM3 ~~          QO1  1.007  0.016   0.016    0.037    0.037
## 574          SM3 ~~          QO2  2.433 -0.025  -0.025   -0.059   -0.059
## 575          SM3 ~~          QO3  2.917  0.029   0.029    0.065    0.065
## 576          SM3 ~~          QW1  1.377  0.019   0.019    0.044    0.044
## 577          SM3 ~~          QW2  0.005  0.001   0.001    0.003    0.003
## 578          SM3 ~~          QW3  0.271  0.009   0.009    0.020    0.020
## 579          SM3 ~~          QM1  0.052 -0.004  -0.004   -0.009   -0.009
## 580          SM3 ~~          QM2  0.703 -0.015  -0.015   -0.033   -0.033
## 581          SM3 ~~          QM3  0.403 -0.012  -0.012   -0.025   -0.025
## 582          QO1 ~~          QO2  0.695 -0.015  -0.015   -0.035   -0.035
## 583          QO1 ~~          QO3  0.790 -0.016  -0.016   -0.038   -0.038
## 584          QO1 ~~          QW1  2.043 -0.023  -0.023   -0.053   -0.053
```

```
## 585          QO1 ~~          QW2  0.958  0.016   0.016    0.037    0.037
## 586          QO1 ~~          QW3  0.790  0.014   0.014    0.034    0.034
## 587          QO1 ~~          QM1  0.000  0.000   0.000    0.000    0.000
## 588          QO1 ~~          QM2  0.002  0.001   0.001    0.002    0.002
## 589          QO1 ~~          QM3  0.076  0.005   0.005    0.010    0.010
## 590          QO2 ~~          QO3  3.003  0.035   0.035    0.080    0.080
## 591          QO2 ~~          QW1  0.276  0.008   0.008    0.020    0.020
## 592          QO2 ~~          QW2  0.213 -0.007  -0.007   -0.018   -0.018
## 593          QO2 ~~          QW3  1.115 -0.017  -0.017   -0.041   -0.041
## 594          QO2 ~~          QM1  0.577  0.012   0.012    0.029    0.029
## 595          QO2 ~~          QM2  0.094  0.005   0.005    0.012    0.012
## 596          QO2 ~~          QM3  0.054 -0.004  -0.004   -0.009   -0.009
## 597          QO3 ~~          QW1  3.715  0.032   0.032    0.073    0.073
## 598          QO3 ~~          QW2  0.126 -0.006  -0.006   -0.014   -0.014
## 599          QO3 ~~          QW3  0.228 -0.008  -0.008   -0.019   -0.019
## 600          QO3 ~~          QM1  0.150 -0.007  -0.007   -0.015   -0.015
## 601          QO3 ~~          QM2  1.192 -0.019  -0.019   -0.042   -0.042
## 602          QO3 ~~          QM3  0.005 -0.001  -0.001   -0.003   -0.003
## 603          QW1 ~~          QW2  0.001  0.001   0.001    0.001    0.001
## 604          QW1 ~~          QW3  0.058  0.004   0.004    0.010    0.010
## 605          QW1 ~~          QM1  0.305  0.009   0.009    0.021    0.021
## 606          QW1 ~~          QM2  1.369 -0.020  -0.020   -0.044   -0.044
## 607          QW1 ~~          QM3  0.147 -0.007  -0.007   -0.014   -0.014
## 608          QW2 ~~          QW3  0.061  0.004   0.004    0.010    0.010
## 609          QW2 ~~          QM1  1.572 -0.021  -0.021   -0.048   -0.048
## 610          QW2 ~~          QM2  0.566 -0.013  -0.013   -0.028   -0.028
## 611          QW2 ~~          QM3  3.016  0.030   0.030    0.066    0.066
## 612          QW3 ~~          QM1  3.580  0.031   0.031    0.073    0.073
## 613          QW3 ~~          QM2  0.053  0.004   0.004    0.009    0.009
## 614          QW3 ~~          QM3  1.082 -0.018  -0.018   -0.040   -0.040
## 615          QM1 ~~          QM2  0.170  0.007   0.007    0.017    0.017
## 616          QM1 ~~          QM3  4.481 -0.039  -0.039   -0.086   -0.086
## 617          QM2 ~~          QM3  1.763  0.025   0.025    0.052    0.052
## 623       Verbal ~~         Oral  0.017 -0.009  -0.009   -0.009   -0.009
## 624       Verbal ~~      Written  0.027  0.011   0.011    0.011    0.011
## 625       Verbal ~~ Manipulative  0.068 -0.023  -0.023   -0.023   -0.023
## 627      Spatial ~~         Oral  0.142  0.024   0.024    0.024    0.024
## 628      Spatial ~~      Written  0.014  0.007   0.007    0.007    0.007
## 629      Spatial ~~ Manipulative  0.540 -0.061  -0.061   -0.061   -0.061
## 630        Quant ~~         Oral  0.037  0.016   0.016    0.016    0.016
## 631        Quant ~~      Written  2.287 -0.111  -0.111   -0.111   -0.111
## 632        Quant ~~ Manipulative  2.131  0.147   0.147    0.147    0.147
```

12.5.12.8 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(cfaModelMTMMFit)
```

```
##           Verbal   Spatial     Quant
## alpha  0.9059841 0.9042265 0.9059678
## omega  0.9130134 0.9116608 0.9139800
## omega2 0.8765047 0.8722275 0.8745058
## omega3 0.8795909 0.8749995 0.8791473
## avevar        NA        NA        NA
```

```
semTools::reliabilityL2(cfaModelMTMMFit, "g")
```

```
##        omegaL1        omegaL2 partialOmegaL1
##      0.8055888      0.8479337      0.9521648
```

12.5.12.9 Path diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(cfaModelMTMMFit,
         what = "std",
         layout = "tree3",
         optimizeLatRes = FALSE,
         bifactor = c("Verbal","Spatial","Quant","g"),
         exoVar = FALSE,
         edge.label.cex = 1.3)
```

12.5.12.10 Equivalently fitting models

Markov equivalent directed acyclic graphs (DAGs) were depicted using the `dagitty` package [Textor et al., 2021].

```
dagModelMTMM <- lavaanToGraph(cfaModelMTMMFit)

par(mfrow = c(2, 2))
```

Figure 12.49: Multi-trait multi-method model in confirmatory factor analysis

```
lapply(equivalentDAGs(dagModelMTMM, n = 4), plot)
```

```
## [[1]]
## [[1]]$mar
## [1] 0 0 0 0
##
##
## [[2]]
## [[2]]$mar
## [1] 0 0 0 0
##
##
## [[3]]
## [[3]]$mar
## [1] 0 0 0 0
##
##
## [[4]]
## [[4]]$mar
## [1] 0 0 0 0
```

12.6 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a technique for data reduction. PCA seeks to identify the principal components of the data, so that data can be reduced to that smaller set of components. Although PCA is sometimes referred to as factor analysis, PCA is not factor analysis [Lilienfeld et al., 2015]. PCA uses the total variance of all variables whereas factor analysis uses the common variance among the variables.

12.6.1 Determine number of components

Determine the number of components to retain using the Scree plot and Very Simple Structure (VSS) plot.

12.6.1.1 Scree Plot

Scree plots were generated using the **psych** [Revelle, 2020] and **nFactors** [Raiche and Magis, 2020] packages.

Figure 12.50: Equivalently fitting models to a multi-trait multi-method model in confirmatory factor analysis

Figure 12.51: Equivalently fitting models to a multi-trait multi-method model in confirmatory factor analysis

Figure 12.52: Equivalently fitting models to a multi-trait multi-method model in confirmatory factor analysis

Figure 12.53: Equivalently fitting models to a multi-trait multi-method model in confirmatory factor analysis

```
fa.parallel(x = HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")
```

Figure 12.54: Scree plots in principal component analysis

```
## Parallel analysis suggests that the number of factors =  NA  and the number of compo
```

```
plot(nScree(x = cor(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8",
```

12.6.1.2 Very Simple Structure (VSS) Plot

Very simple structure (VSS) plots were generated using the `psych` package [Revelle, 2020].

```
vss(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate =
```

12.6.1.2.1 Orthogonal (Varimax) rotation

Figure 12.55: Scree plots in principal component analysis

```
##
## Very Simple Structure
## Call: vss(x = HolzingerSwineford1939[, c("x1", "x2", "x3", "x4", "x5",
##     "x6", "x7", "x8", "x9")], rotate = "varimax", fm = "pc")
## VSS complexity 1 achieves a maximimum of 0.76  with  4  factors
## VSS complexity 2 achieves a maximimum of 0.91  with  4  factors
##
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
## BIC achieves a minimum of  Inf  with    factors
## Sample Size adjusted BIC achieves a minimum of  Inf  with    factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq prob sqresid  fit RMSEA BIC SABIC complex eChisq
## 1 0.62 0.00 0.083   0    NA   NA   6.205 0.62    NA  NA    NA      NA     NA
## 2 0.70 0.80 0.066   0    NA   NA   3.323 0.80    NA  NA    NA      NA     NA
## 3 0.75 0.88 0.072   0    NA   NA   1.484 0.91    NA  NA    NA      NA     NA
## 4 0.76 0.91 0.119   0    NA   NA   0.946 0.94    NA  NA    NA      NA     NA
## 5 0.73 0.90 0.209   0    NA   NA   0.586 0.96    NA  NA    NA      NA     NA
## 6 0.74 0.88 0.367   0    NA   NA   0.320 0.98    NA  NA    NA      NA     NA
## 7 0.74 0.86 0.500   0    NA   NA   0.114 0.99    NA  NA    NA      NA     NA
## 8 0.74 0.86 1.000   0    NA   NA   0.046 1.00    NA  NA    NA      NA     NA
```

Figure 12.56: Very simple structure plot in principal component analysis

```
##   SRMR eCRMS eBIC
## 1   NA    NA   NA
## 2   NA    NA   NA
## 3   NA    NA   NA
## 4   NA    NA   NA
## 5   NA    NA   NA
## 6   NA    NA   NA
## 7   NA    NA   NA
## 8   NA    NA   NA
```

```
vss(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate =
```

12.6.1.2.2 Oblique (Oblimin) rotation

```
##
## Very Simple Structure
## Call: vss(x = HolzingerSwineford1939[, c("x1", "x2", "x3", "x4", "x5",
##     "x6", "x7", "x8", "x9")], rotate = "oblimin", fm = "pc")
```

Figure 12.57: Very simple structure plot in principal component analysis

```
## Although the VSS complexity 1 shows  7  factors, it is probably more reasonable to think about
## VSS complexity 2 achieves a maximimum of 0.99  with  8  factors
##
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
## BIC achieves a minimum of  Inf  with    factors
## Sample Size adjusted BIC achieves a minimum of  Inf  with    factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq prob sqresid  fit RMSEA BIC SABIC complex eChisq
## 1 0.62 0.00 0.083   0    NA   NA   6.205 0.62    NA  NA    NA      NA     NA
## 2 0.74 0.80 0.066   0    NA   NA   3.323 0.80    NA  NA    NA      NA     NA
## 3 0.82 0.90 0.072   0    NA   NA   1.484 0.91    NA  NA    NA      NA     NA
## 4 0.87 0.94 0.119   0    NA   NA   0.946 0.94    NA  NA    NA      NA     NA
## 5 0.85 0.89 0.209   0    NA   NA   0.586 0.96    NA  NA    NA      NA     NA
## 6 0.90 0.94 0.367   0    NA   NA   0.320 0.98    NA  NA    NA      NA     NA
## 7 0.98 0.98 0.500   0    NA   NA   0.114 0.99    NA  NA    NA      NA     NA
## 8 0.93 0.99 1.000   0    NA   NA   0.046 1.00    NA  NA    NA      NA     NA
##   SRMR eCRMS eBIC
## 1   NA    NA   NA
## 2   NA    NA   NA
## 3   NA    NA   NA
```

```
## 4    NA     NA    NA
## 5    NA     NA    NA
## 6    NA     NA    NA
## 7    NA     NA    NA
## 8    NA     NA    NA
```

```
vss(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x8","x9")], rotate =
```

Figure 12.58: Very simple structure plot in principal component analysis

12.6.1.2.3 No rotation

```
##
## Very Simple Structure
## Call: vss(x = HolzingerSwineford1939[, c("x1", "x2", "x3", "x4", "x5",
##     "x6", "x7", "x8", "x9")], rotate = "none", fm = "pc")
## VSS complexity 1 achieves a maximimum of 0.77  with  4  factors
## VSS complexity 2 achieves a maximimum of 0.92  with  5  factors
##
## The Velicer MAP achieves a minimum of 0.07  with  2  factors
```

```
## BIC achieves a minimum of  Inf  with    factors
## Sample Size adjusted BIC achieves a minimum of  Inf  with    factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq prob sqresid  fit RMSEA BIC SABIC complex eChisq
## 1 0.62 0.00 0.083   0    NA   NA   6.205 0.62    NA  NA    NA      NA     NA
## 2 0.70 0.80 0.066   0    NA   NA   3.323 0.80    NA  NA    NA      NA     NA
## 3 0.75 0.88 0.072   0    NA   NA   1.484 0.91    NA  NA    NA      NA     NA
## 4 0.77 0.91 0.119   0    NA   NA   0.946 0.94    NA  NA    NA      NA     NA
## 5 0.73 0.92 0.209   0    NA   NA   0.586 0.96    NA  NA    NA      NA     NA
## 6 0.74 0.88 0.367   0    NA   NA   0.320 0.98    NA  NA    NA      NA     NA
## 7 0.74 0.86 0.500   0    NA   NA   0.114 0.99    NA  NA    NA      NA     NA
## 8 0.74 0.86 1.000   0    NA   NA   0.046 1.00    NA  NA    NA      NA     NA
##   SRMR eCRMS eBIC
## 1   NA    NA   NA
## 2   NA    NA   NA
## 3   NA    NA   NA
## 4   NA    NA   NA
## 5   NA    NA   NA
## 6   NA    NA   NA
## 7   NA    NA   NA
## 8   NA    NA   NA
```

12.6.2 Run the Principal Component Analysis

Principal component analysis (PCA) models were fit using the `psych` package [Revelle, 2020].

12.6.2.1 Orthogonal (Varimax) rotation

```
pca1factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca2factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca3factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca4factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca5factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca6factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca7factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca8factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
pca9factorOrthogonal <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6","x7","x
```

12.6.2.2 Oblique (Oblimin) rotation

```
pca1factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca2factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca3factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca4factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca5factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca6factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca7factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca8factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
pca9factorOblique <- principal(HolzingerSwineford1939[,c("x1","x2","x3","x4","x5","x6"
```

12.6.3 Component scores

12.6.3.1 Orthogonal (Varimax) rotation

```
pca3Orthogonal <- pca3factorOrthogonal$scores
```

12.6.3.2 Oblique (Oblimin) rotation

```
pca3Oblique <- pca3factorOblique$scores
```

12.6.4 Plots

Pairs panel plots were generated using the `psych` package [Revelle, 2020]. Correlation plots were generated using the `corrplot` package [Wei and Simko, 2017].

12.6.4.1 Orthogonal (Varimax) rotation

```
pairs.panels(pca3Orthogonal)
```

```
corrplot(cor(pca3Orthogonal, use = "pairwise.complete.obs"))
```

Figure 12.59: Pairs panel plot using orthogonal rotation in principal component analysis

Figure 12.60: Correlation plot using orthogonal rotation in principal component analysis

12.6.4.2 Oblique (Oblimin) rotation

```
pairs.panels(pca3Oblique)
```

Figure 12.61: Pairs panel plot using oblique rotation in principal component analysis

```
corrplot(cor(pca3Oblique, use = "pairwise.complete.obs"))
```

12.7 Conclusion

12.8 Exercises

12.8.1 Questions

Note: several of the following questions use data from the Children of the National Longitudinal Survey of Youth Survey (CNLSY). The CNLSY is a publicly available longitudinal dataset provided by the Bureau of Labor

Figure 12.62: Correlation plot using oblique rotation in principal component analysis

Statistics (https://www.bls.gov/nls/nlsy79-children.htm#topical-guide). The CNLSY data file for these exercises is located on the course's page of the Open Science Framework (https://osf.io/3pwza). Children's behavior problems were rated in 1988 (time 1: T1) and then again in 1990 (time 2: T2) on the Behavior Problems Index (BPI). Below are the items corresponding to the Antisocial subscale of the BPI: 1) cheats or tells lies 2) bullies or is cruel/mean to others 3) does not seem to feel sorry after misbehaving 4) breaks things deliberately 5) is disobedient at school 6) has trouble getting along with teachers 7) has sudden changes in mood or feeling

1. Fit an exploratory factor analysis (EFA) model using maximum likelihood to the seven items of the Antisocial subscale of the Behavior Problems Index at T1.

    a. Create a scree plot using the raw data (not the correlation or covariance matrix).
    b. How many factors should be retained according to the Kaiser–Guttman rule? How many factors should be retained according to a Parallel Test? Based on the available information, how many factors would you retain?
    c. Extract two factors with an oblique (oblimin) rotation. Which items load most strongly on Factor 1? Which items load most strongly on Factor 2? Why do you think these items load onto Factor 2? Which item has the highest cross-loading (on both Factors 1 and 2)?
    d. What is the association between Factors 1 and 2 when using an oblique rotation? What does this indicate about whether you should use an orthogonal or oblique rotation?

2. [For exercises about confirmatory factor analysis, see the exercises on "Structural Equation Modeling".] – **INCLUDE CROSS-REFERENCE**
3. Fit a principal component analysis (PCA) model to the seven items of the Antisocial subscale of the Behavior Problems Index at T1. How many components should be kept?
4. Fit a one-factor confirmatory factor analysis (CFA) model to the seven items of the Antisocial subscale of the Behavior Problems Index at T1. Set the scale of the latent factor by standardizing the latent factor—set the mean of the factor to one and the variance of the factor to zero. Freely estimate the factor loadings of all indicators. Allow the residuals of indicators 5 and 6 to be correlated. Use full information maximum likelihood (FIML) to account for missing data. Use robust standard errors to account for non-normally distributed data.

    a. How strongly are the factor scores from the one-factor CFA model associated with the factors from a one-factor EFA model and a one-component PCA model?

5. If your model fits well, why is it important to consider alternative models? What are some possible alternative models?

12.8.2 Answers

1. a. [**INSERT FIGURE**]
   b. The Kaiser-Guttman rule states that all factors should be retained whose eigenvalues are greater than one. According to the Kaiser-Guttman rule, one factor should be retained in this case. According to the Parallel Test, factors should be retained that explain more variance than randomly generated data. According to the Parallel Test, two factors should be retained. However, it is not conceptually elegant to include factors that explain less than one variable's worth of variance (i.e., one eigenvalue), so in this case, it would probably make sense to retain one factor.
   c. Items 1 (loading = .52), 2 (loading = .66), 3 (loading = .48), 4 (loading = .52), and 7 (loading = .39) load most strongly on Factor 1. Items 5 ("disobedient at school"; loading = 1.00) and 6 ("trouble getting along with teachers"; loading = .42) load most strongly on Factor 2. It is likely that these two items load onto a separate factor from the other antisocial items because they reflect school-related antisocial behavior. Item 6 has the highest cross-loading—in addition to loading onto Factor 2 (.42), it has a non-trivial loading on Factor 1 (.22).
   d. The association between Factors 1 and 2 is .48. This indicates that the factors are moderately correlated and that using an orthogonal rotation would likely be improper. An oblique rotation would be important to capture the association between the two factors. However, this could come at the expense of having simple structure and, therefore, straightforward interpretation of the factors.

2. [See the exercises on "Structural Equation Modeling".] [**INSERT CROSS-REFERENCE**]

3. According to the Kaiser-Guttman rule and the Parallel Test, one component should be retained.

4. The factor scores from the CFA model are correlated r = .97 and r = .98 with the factor scores of the EFA model and the component scores of the PCA model, respectively. The factor scores of the EFA model are correlated r = .997 with the component scores of the PCA model.

5. Even if a given model fits well, there are many other models that fit just as well. Any given data matrix can produce an infinite number of factor models that accurately represent the data structure (indeterminacy). For any model, there always exist an infinite number of models that fit exactly

the same. There is no way to distinguish which factor model is correct from the data matrix. Although the fit of the structural model may be consistent with the hypothesized model, fit does not demonstrate that a given model is uniquely valid. Thus, it is important to consider alternative models that may better capture the causal processes, even if their fit is mathematically equivalent. For instance, you could consider models with multiple factors, correlated factors, correlated residuals, cross-loading indicators, regression paths, hierarchical (higher-order) factors, bifactors, etc. Remember, an important part of science is specifying and testing plausible alternative explanations! Below are some alternative, equivalently fitting models:

[**INSERT FIGURE**]

Part II

Part II

Chapter 13

Intellectual Assessment

[**Dombrowksi et al, 2019 citation**]

Figure 13.1

Figure 13.1: Depiction of Spearman's two factor theory of intelligence.

Figure 13.2

Figure 13.2: Depiction of Thurstone's theory of intelligence (with only two factors depicted for simplicity).

Figure 13.3

Figure 13.3: Depiction of Cattell's Gf-Gc theory of intelligence. Gf = fluid intelligence; Gc = crystallized intelligence

Figure 13.4

Figure 13.5

13.1 Conclusion

Figure 13.4: Depiction of the Cattell-Horn-Carroll hierarhical theory of intelligence.

Figure 13.5: Depiction of bifactor model of intelligence.

Chapter 14

Test Bias

14.1 Getting Started

14.1.1 Load Libraries

```
library("lavaan")
library("semTools")
library("semPlot")
library("mirt")
library("dmacs")
library("tidyverse")
library("here")
library("tinytex")
```

14.1.2 Prepare Data

14.1.2.1 Load Data

```
cnlsy <- read_csv(here("Data", "cnlsy.csv")) #read_csv("../Data/cnlsy.csv")
```

14.1.2.2 Simulate Data

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
sampleSize <- 4000

set.seed(52242)

mydataBias <- data.frame(ID = 1:sampleSize,
                      group = factor(c("male","female"), levels = c("male","female")),
                      unbiasedPredictor1 = NA, unbiasedPredictor2 = NA, unbiasedPredicto
                      unbiasedCriterion1 = NA, unbiasedCriterion2 = NA, unbiasedCriteri
                      predictor = rnorm(sampleSize, mean = 100, sd = 15),
                      criterion1 = NA, criterion2 = NA, criterion3 = NA, criterion4 = NA

mydataBias$unbiasedPredictor1 <- rnorm(sampleSize, mean = 100, sd = 15)
mydataBias$unbiasedPredictor2[which(mydataBias$group == "male")] <- rnorm(length(which
mydataBias$unbiasedPredictor2[which(mydataBias$group == "female")] <- rnorm(length(whic
mydataBias$unbiasedPredictor3[which(mydataBias$group == "male")] <- rnorm(length(which
mydataBias$unbiasedPredictor3[which(mydataBias$group == "female")] <- rnorm(length(whic

mydataBias$unbiasedCriterion1 <- 1 * mydataBias$unbiasedPredictor1 + rnorm(sampleSize,
mydataBias$unbiasedCriterion2 <- 1 * mydataBias$unbiasedPredictor2 + rnorm(sampleSize,
mydataBias$unbiasedCriterion3 <- 1 * mydataBias$unbiasedPredictor3 + rnorm(sampleSize,

mydataBias$criterion1[which(mydataBias$group == "male")] <- .7 * mydataBias$predictor[w
mydataBias$criterion1[which(mydataBias$group == "female")] <- .7 * mydataBias$predictor
mydataBias$criterion2[which(mydataBias$group == "male")] <- .7 * mydataBias$predictor[w
mydataBias$criterion2[which(mydataBias$group == "female")] <- .7 * mydataBias$predictor
mydataBias$criterion3[which(mydataBias$group == "male")] <- .7 * mydataBias$predictor[w
mydataBias$criterion3[which(mydataBias$group == "female")] <- .3 * mydataBias$predictor
mydataBias$criterion4[which(mydataBias$group == "male")] <- .7 * mydataBias$predictor[w
mydataBias$criterion4[which(mydataBias$group == "female")] <- .3 * mydataBias$predictor
mydataBias$criterion5[which(mydataBias$group == "male")] <- .7 * mydataBias$predictor[w
mydataBias$criterion5[which(mydataBias$group == "female")] <- .7 * mydataBias$predictor
```

14.1.2.3 Add Missing Data

Adding missing data to dataframes helps make examples more realistic to real-life data, and helps you get in the habit of programming to account for missing data.

```
varNames <- names(mydataBias)
dimensionsDf <- dim(mydataBias[,-c(1,2)])
unlistedDf <- unlist(mydataBias[,-c(1,2)])
unlistedDf[sample(1:length(unlistedDf), size = .01 * length(unlistedDf))] <- NA
mydataBias <- cbind(mydataBias[,c("ID","group")], as.data.frame(matrix(unlistedDf, nco
names(mydataBias) <- varNames
```

```
varNames <- names(HolzingerSwineford1939)
dimensionsDf <- dim(HolzingerSwineford1939[,paste("x", 1:9, sep = "")])
unlistedDf <- unlist(HolzingerSwineford1939[,paste("x", 1:9, sep = "")])
unlistedDf[sample(1:length(unlistedDf), size = .01 * length(unlistedDf))] <- NA
HolzingerSwineford1939 <- cbind(HolzingerSwineford1939[,1:6], as.data.frame(matrix(unlistedDf, no
names(HolzingerSwineford1939) <- varNames
```

14.2 Examples of Unbiased Tests (in terms of predictive bias)

14.2.1 Unbiased test where males and females have equal means on predictor and criterion

This is an example of an unbiased test where males and females have equal means on the predictor and criterion. The test is unbiased because there are no significant differences in the regression lines (of `predictor` predicting `criterion`) between males and females.

```
summary(lm(unbiasedCriterion1 ~ unbiasedPredictor1 + group + unbiasedPredictor1:group, data = myd
```

```
##
## Call:
## lm(formula = unbiasedCriterion1 ~ unbiasedPredictor1 + group +
##     unbiasedPredictor1:group, data = mydataBias)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -48.86 -10.25   0.21  10.37  63.24
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.90494    2.27656   0.398    0.691
## unbiasedPredictor1              0.99317    0.02256  44.026   <2e-16 ***
## groupfemale                    -2.73873    3.22835  -0.848    0.396
## unbiasedPredictor1:groupfemale  0.02849    0.03191   0.893    0.372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.12 on 3915 degrees of freedom
##   (81 observations deleted due to missingness)
## Multiple R-squared:  0.5047, Adjusted R-squared:  0.5043
## F-statistic:  1330 on 3 and 3915 DF,  p-value: < 2.2e-16
```

```
plot(unbiasedCriterion1 ~ unbiasedPredictor1, data = mydataBias, xlim = c(0, max(c(myda
points(mydataBias$unbiasedPredictor1[which(mydataBias$group == "male")], mydataBias$unb
points(mydataBias$unbiasedPredictor1[which(mydataBias$group == "female")], mydataBias$u
abline(lm(unbiasedCriterion1 ~ unbiasedPredictor1, data = mydataBias[which(mydataBias$g
abline(lm(unbiasedCriterion1 ~ unbiasedPredictor1, data = mydataBias[which(mydataBias$g
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue",'
```

Figure 14.1: Unbiased test where males and females have equal means on predictor and criterion

14.2.2 Unbiased test where females have higher means than males on predictor and criterion

This is an example of an unbiased test where females have higher means than males on the predictor and criterion. The test is unbiased because there are no differences in the regression lines (of `predictor` predicting `criterion`) between males and females.

```
summary(lm(unbiasedCriterion2 ~ unbiasedPredictor2 + group + unbiasedPredictor2:group,
```

```
##
```

```
## Call:
## lm(formula = unbiasedCriterion2 ~ unbiasedPredictor2 + group +
##     unbiasedPredictor2:group, data = mydataBias)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -63.694  -9.811  -0.125  10.176  52.205
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.54459    1.61129  -0.959    0.338
## unbiasedPredictor2             1.02166    0.02269  45.032   <2e-16 ***
## groupfemale                   -2.54872    3.33099  -0.765    0.444
## unbiasedPredictor2:groupfemale  0.01133    0.03179   0.357    0.721
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.11 on 3905 degrees of freedom
##   (91 observations deleted due to missingness)
## Multiple R-squared:  0.8369, Adjusted R-squared:  0.8368
## F-statistic:  6680 on 3 and 3905 DF,  p-value: < 2.2e-16
```

```
plot(unbiasedCriterion2 ~ unbiasedPredictor2, data = mydataBias, xlim = c(0, max(c(mydataBias$unb
points(mydataBias$unbiasedPredictor2[which(mydataBias$group == "male")], mydataBias$unbiasedCrite
points(mydataBias$unbiasedPredictor2[which(mydataBias$group == "female")], mydataBias$unbiasedCri
abline(lm(unbiasedCriterion2 ~ unbiasedPredictor2, data = mydataBias[which(mydataBias$group == "m
abline(lm(unbiasedCriterion2 ~ unbiasedPredictor2, data = mydataBias[which(mydataBias$group == "f
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue","red"))
```

14.2.3 Unbiased test where males have higher means than females on predictor and criterion

This is an example of an unbiased test where males have higher means than females on the predictor and criterion. The test is unbiased because there are are no differences in the regression lines (of `predictor` predicting `criterion`) between males and females.

```
summary(lm(unbiasedCriterion3 ~ unbiasedPredictor3 + group + unbiasedPredictor3:group, data = myd
```

```
##
## Call:
## lm(formula = unbiasedCriterion3 ~ unbiasedPredictor3 + group +
##     unbiasedPredictor3:group, data = mydataBias)
```

Figure 14.2: Unbiased test where females have higher means than males on predictor and criterion

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.705  -9.956   0.059  10.240  49.985
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.30794    3.06049   0.427    0.669
## unbiasedPredictor3              0.98860    0.02350  42.077   <2e-16 ***
## groupfemale                    -1.74798    3.47084  -0.504    0.615
## unbiasedPredictor3:groupfemale  0.03026    0.03276   0.924    0.356
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.23 on 3915 degrees of freedom
##   (81 observations deleted due to missingness)
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.8222
## F-statistic:  6040 on 3 and 3915 DF,  p-value: < 2.2e-16
```

```
plot(unbiasedCriterion3 ~ unbiasedPredictor3, data = mydataBias, xlim = c(0, max(c(mydataBias$unb
points(mydataBias$unbiasedPredictor3[which(mydataBias$group == "male")], mydataBias$unbiasedCrite
points(mydataBias$unbiasedPredictor3[which(mydataBias$group == "female")], mydataBias$unbiasedCri
abline(lm(unbiasedCriterion3 ~ unbiasedPredictor3, data = mydataBias[which(mydataBias$group == "m
abline(lm(unbiasedCriterion3 ~ unbiasedPredictor3, data = mydataBias[which(mydataBias$group == "f
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue","red"))
```

14.3 Predictive Bias: Different Regression Lines

14.3.1 Example of unbiased prediction (no differences in intercepts or slopes)

This is an example of an unbiased test males and females have equal means on the predictor and criterion. The test is unbiased because there are no differences in the regression lines (of `predictor` predicting `criterion1`) between males and females.

```
summary(lm(criterion1 ~ predictor + group + predictor:group, data = mydataBias))
```

```
##
## Call:
## lm(formula = criterion1 ~ predictor + group + predictor:group,
```

Figure 14.3: Unbiased test where males have higher means than females on predictor and criterion

```
##     data = mydataBias)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4655  -3.3475   0.0486   3.4017  17.8082
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -0.519383   0.755030  -0.688    0.492
## predictor              0.705713   0.007488  94.245   <2e-16 ***
## groupfemale            0.311050   1.072245   0.290    0.772
## predictor:groupfemale -0.003892   0.010611  -0.367    0.714
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.986 on 3922 degrees of freedom
##   (74 observations deleted due to missingness)
## Multiple R-squared:  0.8178, Adjusted R-squared:  0.8176
## F-statistic:  5867 on 3 and 3922 DF,  p-value: < 2.2e-16
```

```
plot(criterion1 ~ predictor, data = mydataBias, xlim = c(0, max(c(mydataBias$criterion1, mydataBi
points(mydataBias$predictor[which(mydataBias$group == "male")], mydataBias$criterion1[which(mydat
points(mydataBias$predictor[which(mydataBias$group == "female")], mydataBias$criterion1[which(myd
abline(lm(criterion1 ~ predictor, data = mydataBias[which(mydataBias$group == "male"),]), lty = 1
abline(lm(criterion1 ~ predictor, data = mydataBias[which(mydataBias$group == "female"),]), lty =
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue","red"))
```

14.3.2 Example of intercept bias

This is an example of a biased test due to intercept bias. There are differences in the intercepts of the regression lines (of `predictor` predicting `criterion2`) between males and females: males have a higher intercept than females. That is, the same score on the predictor results in higher predictions for males than females.

```
summary(lm(criterion2 ~ predictor + group + predictor:group, data = mydataBias))
```

```
##
## Call:
## lm(formula = criterion2 ~ predictor + group + predictor:group,
##     data = mydataBias)
##
## Residuals:
```

Figure 14.4: Example of unbiased prediction (no differences in intercepts or slopes between males and females)

```
##      Min       1Q   Median       3Q      Max
## -20.2463  -3.3082   0.0302   3.4864  16.6332
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              9.852293   0.772959  12.746   <2e-16 ***
## predictor                0.703160   0.007669  91.691   <2e-16 ***
## groupfemale            -10.821609   1.098104  -9.855   <2e-16 ***
## predictor:groupfemale   0.006631   0.010868   0.610    0.542
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.11 on 3917 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:  0.8396, Adjusted R-squared:  0.8395
## F-statistic:  6835 on 3 and 3917 DF,  p-value: < 2.2e-16
```

```
plot(criterion2 ~ predictor, data = mydataBias, xlim = c(0, max(c(mydataBias$criterion2, mydataBi
points(mydataBias$predictor[which(mydataBias$group == "male")], mydataBias$criterion2[which(mydat
points(mydataBias$predictor[which(mydataBias$group == "female")], mydataBias$criterion2[which(myd
abline(lm(criterion2 ~ predictor, data = mydataBias[which(mydataBias$group == "male"),]), lty = 1
abline(lm(criterion2 ~ predictor, data = mydataBias[which(mydataBias$group == "female"),]), lty =
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue","red"))
```

14.3.3 Example of slope bias

This is an example of a biased test due to slope bias. There are differences in the slopes of the regression lines (of `predictor` predicting `criterion3`) between males and females: males have a higher slope than females. That is, scores have stronger predictive validity for males than females.

```
summary(lm(criterion3 ~ predictor + group + predictor:group, data = mydataBias))
```

```
##
## Call:
## lm(formula = criterion3 ~ predictor + group + predictor:group,
##     data = mydataBias)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9788  -3.3382   0.0383   3.2798  17.7093
##
## Coefficients:
```

Figure 14.5: Example of intercept bias in prediction (different intercepts between males and females)

```
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.415276   0.746493   0.556    0.578
## predictor              0.695085   0.007406  93.851   <2e-16 ***
## groupfemale            0.563171   1.061512   0.531    0.596
## predictor:groupfemale -0.404930   0.010507 -38.540   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.942 on 3917 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:  0.9493, Adjusted R-squared:  0.9493
## F-statistic: 2.447e+04 on 3 and 3917 DF,  p-value: < 2.2e-16
```

```
plot(criterion3 ~ predictor, data = mydataBias, xlim = c(0, max(c(mydataBias$criterion3, mydataBi
points(mydataBias$predictor[which(mydataBias$group == "male")], mydataBias$criterion3[which(mydat
points(mydataBias$predictor[which(mydataBias$group == "female")], mydataBias$criterion3[which(myd
abline(lm(criterion3 ~ predictor, data = mydataBias[which(mydataBias$group == "male"),]), lty = 1
abline(lm(criterion3 ~ predictor, data = mydataBias[which(mydataBias$group == "female"),]), lty =
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue","red"))
```

Figure 14.6: Example of slope bias in prediction (different slopes between males and females)

14.3.4 Example of intercept and slope bias

This is an example of a biased test due to intercept and slope bias. There are differences in the intercepts and slopes of the regression lines (of `predictor` predicting `criterion4`) between males and females: males have a higher slope than females. That is, scores have stronger predictive validity for males than females. Females have higher a intercept than males. That is, at lower scores, the same score on the predictor results in higher predictions for females than males; at higher scores on the predictor, the same score results in higher predictions for males than females).

```
summary(lm(criterion4 ~ predictor + group + predictor:group, data = mydataBias))
```

```
##
## Call:
## lm(formula = criterion4 ~ predictor + group + predictor:group,
##     data = mydataBias)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.6940  -3.4091  -0.0356   3.4552  18.9310
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.554502   0.765605   0.724    0.469
## predictor              0.697093   0.007597  91.765   <2e-16 ***
## groupfemale           29.427567   1.087503  27.060   <2e-16 ***
## predictor:groupfemale -0.395012   0.010764 -36.699   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.073 on 3933 degrees of freedom
##   (63 observations deleted due to missingness)
## Multiple R-squared:  0.7763, Adjusted R-squared:  0.7761
## F-statistic:  4550 on 3 and 3933 DF,  p-value: < 2.2e-16
```

```
plot(criterion4 ~ predictor, data = mydataBias, xlim = c(0, max(c(mydataBias$criterion4
points(mydataBias$predictor[which(mydataBias$group == "male")], mydataBias$criterion4[w
points(mydataBias$predictor[which(mydataBias$group == "female")], mydataBias$criterion4
abline(lm(criterion4 ~ predictor, data = mydataBias[which(mydataBias$group == "male"),]
abline(lm(criterion4 ~ predictor, data = mydataBias[which(mydataBias$group == "female")
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue",
```

Figure 14.7: Example of intercept and slope bias in prediction (different intercepts and slopes between males and females)

14.3.5 Example of different measurement reliability/error across groups

In this example, there are differences in the measurement reliability/error on the criterion between males and females: males' scores have a lower reliability (higher measurement error) on the criterion than females' scores. That is, we are more confident about a female's level on the criterion given a particular score on the predictor than we are about a male's level on the criterion given a particular score on the predictor.

```
summary(lm(criterion5 ~ predictor + group + predictor:group, data = mydataBias))
```

```
##
## Call:
## lm(formula = criterion5 ~ predictor + group + predictor:group,
##     data = mydataBias)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -124.904   -6.515   -0.019    6.444  101.939
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.64878    3.20544  -0.514    0.607
## predictor              0.71055    0.03180  22.342   <2e-16 ***
## groupfemale            2.01698    4.56222   0.442    0.658
## predictor:groupfemale -0.01383    0.04516  -0.306    0.759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.2 on 3916 degrees of freedom
##   (80 observations deleted due to missingness)
## Multiple R-squared:  0.1991, Adjusted R-squared:  0.1985
## F-statistic: 324.5 on 3 and 3916 DF,  p-value: < 2.2e-16
```

```
plot(criterion5 ~ predictor, data = mydataBias, xlim = c(0, max(c(mydataBias$criterion5
points(mydataBias$predictor[which(mydataBias$group == "male")], mydataBias$criterion5[w
points(mydataBias$predictor[which(mydataBias$group == "female")], mydataBias$criterion5
abline(lm(criterion5 ~ predictor, data = mydataBias[which(mydataBias$group == "male"),]
abline(lm(criterion5 ~ predictor, data = mydataBias[which(mydataBias$group == "female")
legend("bottomright", c("Male","Female"), lty = c(1,2), pch = c(20,1), col = c("blue",
```

Figure 14.8: Example of different measurement reliability/error across groups

14.4 Differential Item Functioning (DIF)

Differential item functioning (DIF) indicates that one or more items functions differently across groups. That is, one or more of the item parameters for an item differs across groups. For instance, the severity or discrimination of an item could be higher in one group compared to another group. If an item functions differently across groups, it can lead to biased scores for particular groups. Thus, when observing group differences in level on the measure or group differences in the measure's association with other measures, it is unclear whether the observed group differences reflect true group differences or differences in the functioning of the measure across groups. [**GIVE EXAMPLE**]

When examining differential item functioning with many items across many groups, there can be many tests, which will make it likely that DIF will be detected, especially with a large sample. Some detected DIF may be artificial or trivial, but other DIF may be real and important. It is important to consider how you will proceed when detecting DIF. Considerations of effect size and theory can be important for evaluating the DIF and whether it is negligible or important to address. When detecting non-negligible DIF, one option is to remove the items that show DIF. Another option is to *resolve* items that show non-negligible DIF, by keeping the item, but allowing the item's parameters to differ across groups (i.e., essentially treating it as a different item for each group). Removing items reduces a measure's reliability and ability to detect individual differences, but resolving DIF has a very small effect on reliability and person separation [Hagquist and Andrich, 2017, Hagquist, 2019]. Where possible, it therefore can be beneficial to resolve non-negligible DIF instead of removing items. If an item is clearly invalid in one group but valid in another group, another option is to keep the item in one group, and to remove it in another group. Addressing items that show larger DIF can also reduce artificial DIF in other items [Hagquist and Andrich, 2017]. Thus, it can also be beneficial to review and address DIF sequentially by magnitude (from high to low).

Tests of DIF were conducted using the `mirt` package [Chalmers, 2020a].

14.4.1 Unconstrained Model

14.4.1.1 Fit Model

```
unconstrainedModel <- multipleGroup(data = cnlsy[,c("bpi_antisocialT1_1","bpi_antisocia
                                    model = 1,
                                    group = cnlsy$sex,
                                    SE = TRUE,
                                    technical = list(removeEmptyRows = TRUE))
```

14.4.1.2 Model Summary

Items that appear to differ:

- items 5 and 6 are more discriminating (a parameter) among females than males
- item 4 has higher difficulty/severity (b_1 and b_2 parameters) among males than females

```
summary(unconstrainedModel)
```

```
##
## ----------
## GROUP: female
##                        F1    h2
## bpi_antisocialT1_1 0.698 0.487
## bpi_antisocialT1_2 0.706 0.499
## bpi_antisocialT1_3 0.532 0.283
## bpi_antisocialT1_4 0.683 0.466
## bpi_antisocialT1_5 0.763 0.582
## bpi_antisocialT1_6 0.838 0.702
## bpi_antisocialT1_7 0.429 0.184
##
## SS loadings:  3.202
## Proportion Var:  0.457
##
## Factor correlations:
##
##    F1
## F1  1
##
## ----------
## GROUP: male
##                        F1    h2
## bpi_antisocialT1_1 0.663 0.440
## bpi_antisocialT1_2 0.736 0.542
## bpi_antisocialT1_3 0.531 0.282
## bpi_antisocialT1_4 0.713 0.508
## bpi_antisocialT1_5 0.691 0.477
## bpi_antisocialT1_6 0.702 0.492
## bpi_antisocialT1_7 0.507 0.257
##
## SS loadings:  2.999
## Proportion Var:  0.428
##
```

```
## Factor correlations:
##
##    F1
## F1  1
```

```
coef(unconstrainedModel, simplify = TRUE, IRTpars = TRUE)
```

```
## $female
## $items
##                        a     b1    b2
## bpi_antisocialT1_1 1.657  0.029 2.451
## bpi_antisocialT1_2 1.697  0.953 2.763
## bpi_antisocialT1_3 1.069  0.650 2.585
## bpi_antisocialT1_4 1.590  1.895 3.465
## bpi_antisocialT1_5 2.008  1.281 2.936
## bpi_antisocialT1_6 2.614  1.625 2.770
## bpi_antisocialT1_7 0.807 -1.184 2.583
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
##
##
## $male
## $items
##                        a     b1    b2
## bpi_antisocialT1_1 1.507 -0.238 2.505
## bpi_antisocialT1_2 1.853  0.582 2.455
## bpi_antisocialT1_3 1.067  0.384 2.480
## bpi_antisocialT1_4 1.729  1.165 2.765
## bpi_antisocialT1_5 1.625  0.635 2.757
## bpi_antisocialT1_6 1.677  1.494 2.873
## bpi_antisocialT1_7 1.002 -0.986 2.143
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
```

14.4.2 Constrained Model

Constrain item parameters to be equal across groups (to use as baseline model for identifying DIF).

14.4.2.1 Fit Model

```
constrainedModel <- multipleGroup(data = cnlsy[,c("bpi_antisocialT1_1","bpi_antisocialT1_2","bpi_
                                  model = 1,
                                  group = cnlsy$sex,
                                  invariance = c(c("bpi_antisocialT1_1","bpi_antisocialT1_2","bpi
                                  SE = TRUE,
                                  technical = list(removeEmptyRows = TRUE))
```

14.4.2.2 Model Summary

```
summary(constrainedModel)
```

```
##
## ----------
## GROUP: female
##                       F1    h2
## bpi_antisocialT1_1 0.659 0.434
## bpi_antisocialT1_2 0.709 0.503
## bpi_antisocialT1_3 0.514 0.264
## bpi_antisocialT1_4 0.704 0.495
## bpi_antisocialT1_5 0.737 0.543
## bpi_antisocialT1_6 0.768 0.590
## bpi_antisocialT1_7 0.441 0.195
##
## SS loadings:  3.024
## Proportion Var:  0.432
##
## Factor correlations:
##
##    F1
## F1  1
##
## ----------
## GROUP: male
##                       F1    h2
```

```
## bpi_antisocialT1_1 0.677 0.458
## bpi_antisocialT1_2 0.728 0.530
## bpi_antisocialT1_3 0.528 0.279
## bpi_antisocialT1_4 0.723 0.522
## bpi_antisocialT1_5 0.756 0.572
## bpi_antisocialT1_6 0.788 0.621
## bpi_antisocialT1_7 0.453 0.205
##
## SS loadings:  3.187
## Proportion Var:  0.455
##
## Factor correlations:
##
##    F1
## F1  1
```

```
coef(constrainedModel, simplify = TRUE, IRTpars = TRUE)
```

```
## $female
## $items
##                        a     b1    b2
## bpi_antisocialT1_1 1.492  0.089 2.791
## bpi_antisocialT1_2 1.713  0.978 2.883
## bpi_antisocialT1_3 1.020  0.733 2.836
## bpi_antisocialT1_4 1.686  1.687 3.265
## bpi_antisocialT1_5 1.854  1.138 3.014
## bpi_antisocialT1_6 2.040  1.779 3.034
## bpi_antisocialT1_7 0.837 -0.952 2.697
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
##
##
## $male
## $items
##                        a     b1    b2
## bpi_antisocialT1_1 1.492  0.089 2.791
## bpi_antisocialT1_2 1.713  0.978 2.883
## bpi_antisocialT1_3 1.020  0.733 2.836
## bpi_antisocialT1_4 1.686  1.687 3.265
```

```
## bpi_antisocialT1_5 1.854  1.138 3.014
## bpi_antisocialT1_6 2.040  1.779 3.034
## bpi_antisocialT1_7 0.837 -0.952 2.697
##
## $means
##   F1
## 0.39
##
## $cov
##       F1
## F1 1.054
```

14.4.3 Compare model fit of constrained model to unconstrained model

The constrained model and the unconstrained model are considered "nested" models. The constrained model is nested within the unconstrained model because the unconstrained model includes all of the terms of the constrained model along with additional terms. Model fit of nested models can be compared with a chi-square difference test.

```
anova(unconstrainedModel, constrainedModel)
```

```
##
## Model 1: multipleGroup(data = cnlsy[, c("bpi_antisocialT1_1", "bpi_antisocialT1_2",
##     "bpi_antisocialT1_3", "bpi_antisocialT1_4", "bpi_antisocialT1_5",
##     "bpi_antisocialT1_6", "bpi_antisocialT1_7")], model = 1,
##     group = cnlsy$sex, invariance = c(c("bpi_antisocialT1_1",
##         "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##         "bpi_antisocialT1_5", "bpi_antisocialT1_6", "bpi_antisocialT1_7"),
##         "free_means", "free_var"), SE = TRUE, technical = list(removeEmptyRows = TRUE))
## Model 2: multipleGroup(data = cnlsy[, c("bpi_antisocialT1_1", "bpi_antisocialT1_2",
##     "bpi_antisocialT1_3", "bpi_antisocialT1_4", "bpi_antisocialT1_5",
##     "bpi_antisocialT1_6", "bpi_antisocialT1_7")], model = 1,
##     group = cnlsy$sex, SE = TRUE, technical = list(removeEmptyRows = TRUE))
```

```
##        AIC     AICc    SABIC       HQ      BIC    logLik      X2  df   p
## 1 34331.88 34332.17 34403.08 34383.08 34476.17 -17142.94     NaN NaN NaN
## 2 34247.21 34248.14 34377.22 34340.71 34510.68 -17081.61 122.672  19   0
```

The constrained model fits significantly worse than the unconstrained model, which suggests that item parameters differ between males and females.

14.4.4 Identify DIF by iteratively removing constraints from fully constrained model

14.4.4.1 Items that differ in discrimination and/or severity

Items 1, 3, 4, 5, 6, and 7 showed DIF in discrimination and/or severity.

```
difDropItems <- DIF(constrainedModel,
                    c("a1","d1","d2"),
                    scheme = "drop",
                    simplify = TRUE)
difDropItems
```

```
##                        AIC    AICc   SABIC      HQ     BIC     X2 df     p
## bpi_antisocialT1_1 -11.898 -11.821  -2.612  -5.220   6.921 17.898  3 0.000
## bpi_antisocialT1_2   1.891   1.968  11.178   8.569  20.710  4.109  3 0.250
## bpi_antisocialT1_3  -1.774  -1.696   7.513   4.904  17.045  7.774  3 0.051
## bpi_antisocialT1_4 -14.222 -14.145  -4.936  -7.544   4.597 20.222  3 0.000
## bpi_antisocialT1_5 -40.265 -40.187 -30.978 -33.587 -21.445 46.265  3 0.000
## bpi_antisocialT1_6  -4.361  -4.284   4.925   2.317  14.458 10.361  3 0.016
## bpi_antisocialT1_7 -24.693 -24.615 -15.406 -18.015  -5.873 30.693  3 0.000
```

14.4.4.2 Items that differ in discrimination

Items 1, 4, 5, and 7 showed DIF in discrimination.

```
difDropItemsDiscrimination <- DIF(constrainedModel,
                                  c("a1"),
                                  scheme = "drop",
                                  simplify = TRUE)
difDropItemsDiscrimination
```

```
##                        AIC    AICc   SABIC      HQ    BIC     X2 df     p
## bpi_antisocialT1_1 -13.095 -13.071 -10.000 -10.869 -6.822 15.095  1 0.000
## bpi_antisocialT1_2  -1.403  -1.379   1.692   0.823  4.870  3.403  1 0.065
## bpi_antisocialT1_3  -3.544  -3.519  -0.448  -1.318  2.729  5.544  1 0.019
## bpi_antisocialT1_4 -12.043 -12.018  -8.947  -9.817 -5.770 14.043  1 0.000
## bpi_antisocialT1_5 -14.249 -14.224 -11.153 -12.023 -7.975 16.249  1 0.000
## bpi_antisocialT1_6   0.827   0.852   3.923   3.053  7.100  1.173  1 0.279
## bpi_antisocialT1_7   1.969   1.994   5.064   4.195  8.242  0.031  1 0.860
```

14.4.4.3 Items that differ in severity

Items 1, 3, 4, 5, and 7 showed DIF in severity/difficulty.

```
difDropItemsSeverity <- DIF(constrainedModel,
                            c("d1","d2"),
                            scheme = "drop",
                            simplify = TRUE)
difDropItemsSeverity
```

```
##                        AIC    AICc   SABIC      HQ     BIC     X2 df     p
## bpi_antisocialT1_1 -10.052 -10.001  -3.861  -5.600   2.494 14.052  2 0.001
## bpi_antisocialT1_2   0.108   0.159   6.300   4.560  12.655  3.892  2 0.143
## bpi_antisocialT1_3  -3.041  -2.990   3.151   1.411   9.506  7.041  2 0.030
## bpi_antisocialT1_4 -16.102 -16.052  -9.911 -11.650  -3.556 20.102  2 0.000
## bpi_antisocialT1_5 -40.896 -40.846 -34.705 -36.444 -28.350 44.896  2 0.000
## bpi_antisocialT1_6   3.628   3.679   9.820   8.080  16.175  0.372  2 0.830
## bpi_antisocialT1_7 -23.573 -23.523 -17.382 -19.121 -11.027 27.573  2 0.000
```

14.4.5 Identify DIF by iteratively adding constraints to unconstrained model

14.4.5.1 Items that differ in discrimination and/or severity

Items 1, 2, 3, 4, 5, and 6 showed DIF in discrimination and/or severity.

```
difAddItems <- DIF(unconstrainedModel,
                   c("a1","d1","d2"),
                   scheme = "add",
                   simplify = TRUE,
                   plotdif = TRUE)
```

```
difAddItems
```

```
##                         AIC     AICc    SABIC       HQ      BIC      X2 df
## bpi_antisocialT1_1  -14.560  -14.432   -5.273   -7.882    4.259  20.560  3
## bpi_antisocialT1_2  -27.694  -27.567  -18.408  -21.016   -8.875  33.694  3
## bpi_antisocialT1_3   -6.873   -6.745    2.414   -0.195   11.946  12.873  3
## bpi_antisocialT1_4  -75.770  -75.642  -66.483  -69.092  -56.950  81.770  3
## bpi_antisocialT1_5 -124.660 -124.532 -115.373 -117.982 -105.841 130.660  3
## bpi_antisocialT1_6  -38.409  -38.282  -29.123  -31.731  -19.590  44.409  3
## bpi_antisocialT1_7    1.962    2.090   11.249    8.640   20.781   4.038  3
##                        p
## bpi_antisocialT1_1 0.000
## bpi_antisocialT1_2 0.000
## bpi_antisocialT1_3 0.005
## bpi_antisocialT1_4 0.000
## bpi_antisocialT1_5 0.000
## bpi_antisocialT1_6 0.000
## bpi_antisocialT1_7 0.257
```

14.4.5.2 Items that differ in discrimination

Items 6 and 7 showed DIF in discrimination.

```
difAddItemsDiscrimination <- DIF(unconstrainedModel,
                                 c("a1"),
                                 scheme = "add",
                                 simplify = TRUE,
                                 plotdif = TRUE)
```

```
difAddItemsDiscrimination
```

```
##                       AIC   AICc  SABIC     HQ    BIC     X2 df     p
## bpi_antisocialT1_1  1.042  1.085  4.137  3.268  7.315  0.958  1 0.328
## bpi_antisocialT1_2  1.284  1.327  4.379  3.510  7.557  0.716  1 0.397
## bpi_antisocialT1_3  2.000  2.043  5.095  4.226  8.273  0.000  1 0.988
## bpi_antisocialT1_4  1.493  1.537  4.589  3.719  7.766  0.507  1 0.476
## bpi_antisocialT1_5 -1.232 -1.188  1.864  0.994  5.041  3.232  1 0.072
## bpi_antisocialT1_6 -8.464 -8.420 -5.368 -6.238 -2.191 10.464  1 0.001
## bpi_antisocialT1_7 -1.878 -1.835  1.217  0.348  4.395  3.878  1 0.049
```

14.4.5.3 Items that differ in severity

Items 1, 2, 3, 4, 5, and 6 showed DIF in severity/difficulty.

```
difAddItemsSeverity <- DIF(unconstrainedModel,
                           c("d1","d2"),
                           scheme = "add",
                           simplify = TRUE,
                           plotdif = TRUE)
```

```
difAddItemsSeverity
```

```
##                        AIC    AICc   SABIC      HQ     BIC     X2 df     p
## bpi_antisocialT1_1 -15.562 -15.476  -9.371 -11.110  -3.015 19.562  2 0.000
## bpi_antisocialT1_2 -19.151 -19.065 -12.960 -14.699  -6.605 23.151  2 0.000
## bpi_antisocialT1_3  -8.534  -8.448  -2.343  -4.082   4.012 12.534  2 0.002
## bpi_antisocialT1_4 -28.940 -28.854 -22.749 -24.488 -16.394 32.940  2 0.000
## bpi_antisocialT1_5 -92.485 -92.399 -86.294 -88.033 -79.939 96.485  2 0.000
## bpi_antisocialT1_6 -28.929 -28.843 -22.738 -24.477 -16.383 32.929  2 0.000
## bpi_antisocialT1_7   3.489   3.575   9.680   7.941  16.035  0.511  2 0.774
```

14.4.6 Compute effect size of DIF

[CITATION by Gonzalez and Pelham, in press, Assessment]

Effect size measures of DIF were computed based on expected scores [Meade, 2010] using the `mirt` package [Chalmers, 2020a].

14.4.6.1 Test-level DIF

Differential test functioning

```
empirical_ES(unconstrainedModel, DIF = FALSE)
```

```
##           Effect Size     Value
## 1                STDS 0.6052089
## 2                UTDS 0.6413932
## 3              UETSDS 0.6052089
## 4               ETSSD 0.3652185
## 5         Starks.DTFR 0.6085074
## 6               UDTFR 0.6487743
## 7              UETSDN 0.6102540
## 8 theta.of.max.test.D 1.2157431
## 9           Test.Dmax 1.1263941
```

```
empirical_ES(unconstrainedModel, DIF = FALSE, plot = TRUE)
```

Figure 14.9: Differential test functioning by sex

14.4.6.2 Item-level DIF

```
empirical_ES(unconstrainedModel, DIF = TRUE)
```

```
##         SIDS  UIDS  SIDN  UIDN   ESSD theta.of.max.D max.D mean.ES.foc
## item.1 0.083 0.084 0.076 0.077  0.275         -0.335 0.114       0.605
## item.2 0.090 0.090 0.093 0.093  0.317          1.053 0.180       0.346
## item.3 0.066 0.066 0.064 0.064  0.252          0.764 0.083       0.499
## item.4 0.110 0.110 0.119 0.119  0.567          2.550 0.396       0.208
## item.5 0.182 0.182 0.181 0.181  0.730          0.882 0.319       0.340
## item.6 0.074 0.075 0.074 0.076  0.397          1.096 0.175       0.153
## item.7 0.000 0.035 0.001 0.039 -0.002          3.221 0.134       0.831
##        mean.ES.ref
## item.1       0.522
## item.2       0.256
## item.3       0.433
## item.4       0.098
## item.5       0.157
## item.6       0.079
## item.7       0.832
```

```
empirical_ES(unconstrainedModel, DIF = TRUE, plot = TRUE)
```

14.4.7 Item plots

```
itemplot(unconstrainedModel, "bpi_antisocialT1_1", type = "trace")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_2", type = "trace")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_3", type = "trace")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_4", type = "trace")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_5", type = "trace")
```

Figure 14.10: Differential item functioning by sex

```
itemplot(unconstrainedModel, "bpi_antisocialT1_6", type = "trace")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_7", type = "trace")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_1", type = "info")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_2", type = "info")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_3", type = "info")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_4", type = "info")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_5", type = "info")
```

Figure 14.11: Item category boundary curves by sex

```
itemplot(unconstrainedModel, "bpi_antisocialT1_6", type = "info")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_7", type = "info")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_1", type = "score")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_2", type = "score")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_3", type = "score")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_4", type = "score")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_5", type = "score")
```

Figure 14.12: Item category boundary curves by sex

```
itemplot(unconstrainedModel, "bpi_antisocialT1_6", type = "score")
```

```
itemplot(unconstrainedModel, "bpi_antisocialT1_7", type = "score")
```

14.4.8 Addressing DIF

Based on the analyses above, item 5 shows the largest magnitude of DIF. Specifically, item 5 has a stronger discrimination parameter for women than men. So, we should handle item 5 first. If we deem the DIF for item 5 to be non-negligible, we have three primary options: 1) drop item 5 for both men and women, 2) drop item 5 for men but keep it for women, or 3) freely estimate parameters for item 5 across groups.

14.4.8.1 Drop item for both groups

The first option is to drop item 5 for both groups. You might do this if you want to use a measure that has only those items that function equivalently

Figure 14.13: Item category boundary curves by sex

across groups. However, this can lead to lower reliability and weaker ability to detect individual differences.

```
constrainedModelDropItem5 <- multipleGroup(data = cnlsy[,c("bpi_antisocialT1_1","bpi_an
                                    model = 1,
                                    group = cnlsy$sex,
                                    invariance = c(c("bpi_antisocialT1_1","bpi_antisocial
                                    SE = TRUE,
                                    technical = list(removeEmptyRows = TRUE))

coef(constrainedModelDropItem5, simplify = TRUE)
```

14.4.8.2 Drop item for one group but not another group

A second option is to drop item 5 for men but to keep it for women. You might do this if the item is invalid for men, but still valid for women. The coefficients show an item parameter for item 5 for men, but it was constrained to be equal to the parameter for women, and data were removed from the estimation of item 5 for men.

Figure 14.14: Item category boundary curves by sex

```
dropItem5ForMen <- cnlsy
dropItem5ForMen[which(dropItem5ForMen$sex == "male"), "bpi_antisocialT1_5"] <- NA

constrainedModelDropItem5ForMen <- multipleGroup(data = dropItem5ForMen[,c("bpi_antisocialT1_1",'
                                    model = 1,
                                    group = dropItem5ForMen$sex,
                                    invariance = c(c("bpi_antisocialT1_1","bpi_antisocialT1_2","bpi
                                    SE = TRUE,
                                    technical = list(removeEmptyRows = TRUE))

coef(constrainedModelDropItem5ForMen, simplify = TRUE)
```

14.4.8.3 Freely estimate item to have different parameters across groups

Alternatively, we can resolve DIF by allowing the item to have different parameters across both groups. You might do this if the item is valid for both men and women, but it has importantly different item parameters nonetheless.

Figure 14.15: Item category boundary curves by sex

```
constrainedModelResolveItem5 <- multipleGroup(data = cnlsy[,c("bpi_antisocialT1_1","bpi
                                    model = 1,
                                    group = cnlsy$sex,
                                    invariance = c(c("bpi_antisocialT1_1","bpi_antisocial
                                    SE = TRUE,
                                    technical = list(removeEmptyRows = TRUE))

coef(constrainedModelResolveItem5, simplify = TRUE)
```

14.4.8.4 Compare model fit

We can compare the model fit against the fully constrained model. In this case, all three of these approaches for handling DIF resulted in significant improvement in model fit.

```
anova(constrainedModelDropItem5, constrainedModel)
```

```
##
## Model 1: multipleGroup(data = cnlsy[, c("bpi_antisocialT1_1", "bpi_antisocialT1_2",
```

Figure 14.16: Item category boundary curves by sex

```
##     "bpi_antisocialT1_3", "bpi_antisocialT1_4", "bpi_antisocialT1_5",
##     "bpi_antisocialT1_6", "bpi_antisocialT1_7")], model = 1,
##     group = cnlsy$sex, invariance = c(c("bpi_antisocialT1_1",
##         "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##         "bpi_antisocialT1_5", "bpi_antisocialT1_6", "bpi_antisocialT1_7"),
##         "free_means", "free_var"), SE = TRUE, technical = list(removeEmptyRows = TRUE))
## Model 2: multipleGroup(data = cnlsy[, c("bpi_antisocialT1_1", "bpi_antisocialT1_2",
##     "bpi_antisocialT1_3", "bpi_antisocialT1_4", "bpi_antisocialT1_6",
##     "bpi_antisocialT1_7")], model = 1, group = cnlsy$sex, invariance = c(c("bpi_antisocialT1_1
##     "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##     "bpi_antisocialT1_6", "bpi_antisocialT1_7"), "free_means",
##     "free_var"), SE = TRUE, technical = list(removeEmptyRows = TRUE))
```

```
##        AIC     AICc    SABIC       HQ      BIC    logLik       X2   df   p
## 1 34331.88 34332.17 34403.08 34383.08 34476.17 -17142.94      NaN  NaN NaN
## 2 30777.96 30778.18 30839.87 30822.48 30903.42 -15368.98 3547.921 2913   0
```

```
anova(constrainedModelDropItem5ForMen, constrainedModel)
```

```
##
```

Figure 14.17: Item category boundary curves by sex

```
## Model 1: multipleGroup(data = cnlsy[, c("bpi_antisocialT1_1", "bpi_antisocialT1_2",
##     "bpi_antisocialT1_3", "bpi_antisocialT1_4", "bpi_antisocialT1_5",
##     "bpi_antisocialT1_6", "bpi_antisocialT1_7")], model = 1,
##     group = cnlsy$sex, invariance = c(c("bpi_antisocialT1_1",
##         "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##         "bpi_antisocialT1_5", "bpi_antisocialT1_6", "bpi_antisocialT1_7"),
##         "free_means", "free_var"), SE = TRUE, technical = list(removeEmptyRows = TRU
## Model 2: multipleGroup(data = dropItem5ForMen[, c("bpi_antisocialT1_1",
##     "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##     "bpi_antisocialT1_5", "bpi_antisocialT1_6", "bpi_antisocialT1_7")],
##     model = 1, group = dropItem5ForMen$sex, invariance = c(c("bpi_antisocialT1_1",
##         "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##         "bpi_antisocialT1_5", "bpi_antisocialT1_6", "bpi_antisocialT1_7"),
##         "free_means", "free_var"), SE = TRUE, technical = list(removeEmptyRows = TRU
```

```
##        AIC     AICc    SABIC       HQ      BIC    logLik       X2  df   p
## 1 34331.88 34332.17 34403.08 34383.08 34476.17 -17142.94      NaN NaN NaN
## 2 32179.16 32179.45 32250.36 32230.36 32323.44 -16066.58 2152.722   0   0
```

Figure 14.18: Item information functions by sex

```
anova(constrainedModelResolveItem5, constrainedModel)
```

```
##
## Model 1: multipleGroup(data = cnlsy[, c("bpi_antisocialT1_1", "bpi_antisocialT1_2",
##     "bpi_antisocialT1_3", "bpi_antisocialT1_4", "bpi_antisocialT1_5",
##     "bpi_antisocialT1_6", "bpi_antisocialT1_7")], model = 1,
##     group = cnlsy$sex, invariance = c(c("bpi_antisocialT1_1",
##         "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##         "bpi_antisocialT1_5", "bpi_antisocialT1_6", "bpi_antisocialT1_7"),
##         "free_means", "free_var"), SE = TRUE, technical = list(removeEmptyRows = TRUE))
## Model 2: multipleGroup(data = cnlsy[, c("bpi_antisocialT1_1", "bpi_antisocialT1_2",
##     "bpi_antisocialT1_3", "bpi_antisocialT1_4", "bpi_antisocialT1_5",
##     "bpi_antisocialT1_6", "bpi_antisocialT1_7")], model = 1,
##     group = cnlsy$sex, invariance = c(c("bpi_antisocialT1_1",
##         "bpi_antisocialT1_2", "bpi_antisocialT1_3", "bpi_antisocialT1_4",
##         "bpi_antisocialT1_6", "bpi_antisocialT1_7"), "free_means",
##         "free_var"), SE = TRUE, technical = list(removeEmptyRows = TRUE))

##        AIC     AICc    SABIC       HQ      BIC    logLik     X2  df   p
## 1 34331.88 34332.17 34403.08 34383.08 34476.17 -17142.94    NaN NaN NaN
## 2 34291.61 34291.97 34372.09 34349.49 34454.71 -17119.81 46.274   3   0
```

Figure 14.19: Item information functions by sex

14.4.8.5 Next steps

Whichever of these approaches we select, we would then identify and handle the remaining non-negligent DIF sequentially by magnitude. For instance, if we chose to resolve DIF in item 5 by allowing the item to have different parameters across both groups, we would then iteratively drop constraints to see which items continue to show non-negligble DIF.

```
difDropItemsNew <- DIF(constrainedModelResolveItem5,
                       c("a1","d1","d2"),
                       scheme = "drop",
                       simplify = TRUE)
difDropItemsNew
```

```
##                        AIC    AICc   SABIC      HQ    BIC     X2 df     p
## bpi_antisocialT1_1  -5.375  -5.289   3.911   1.303 13.444 11.375  3 0.010
## bpi_antisocialT1_2   3.904   3.991  13.191  10.583 22.724  2.096  3 0.553
## bpi_antisocialT1_3   1.503   1.590  10.789   8.181 20.322  4.497  3 0.213
## bpi_antisocialT1_4 -26.452 -26.365 -17.165 -19.774 -7.633 32.452  3 0.000
## bpi_antisocialT1_5  -0.001  -0.001  -0.001  -0.001 -0.001  0.001  0 0.000
## bpi_antisocialT1_6  -9.912  -9.826  -0.626  -3.234  8.907 15.912  3 0.001
```

Figure 14.20: Item information functions by sex

```
## bpi_antisocialT1_7 -16.552 -16.466  -7.266  -9.874  2.267 22.552  3 0.000
```

Of the remaining DIF, item 4 appears to have the largest DIF. If we deem item 4 to show non-negligible DIF, we would address it using one of the three approaches above and then see which items continue to show the largest DIF, address it if necessary, then identify remaining DIF, etc.

14.5 Measurement/Factorial Invariance

Before making comparisons across groups in terms of associations between constructs or their level on the construct, it is important to establish *measurement invariance* (also called *factorial invariance*) across the groups [Millsap, 2011]. Measurement invariance can help provide greater confidence that the measure functions equivalently across groups, that is, the items have the same strength of association with the latent factor, and the latent factor is on the same metric (i.e., the items have the same level when accounting for the latent factor). For instance, If you observe differences between groups in level of depression without establishing measurement invariance across the groups, you do not know whether the differences observed reflect true

Figure 14.21: Item information functions by sex

group-related differences in depression or differences in the functioning of the measure across the groups.

The tests of measurement invariance in confirmatory factor analysis (CFA) models were fit in the `lavaan` package [Rosseel et al., 2020]. The examples were adapted from `lavaan` documentation: http://lavaan.ugent.be/tutorial/groups.html. Procedures for testing measurement invariance are outlined in Putnick and Bornstein [2016]. In addition to nested chi-square difference tests (χ^2_{diff}), I also demonstrate permutation procedures for testing measurement invariance, as described by Jorgensen et al. [2018]. Also, you are encouraged to read about multiple-indicator, multiple-causes (MIMIC) models [Cheng et al., 2016, Wang et al., 2009] and moderated nonlinear factor analysis (MNLFA) that allows for testing measurement invariance across continuous moderators [Bauer et al., 2020, Curran et al., 2014, Gottfredson et al., 2019]. MNLFA is implemented in the `mnlfa` package [Robitzsch, 2019]. You can also generate syntax for conducting MNLFA in Mplus software [Muthén and Muthén, 2019] using the `aMNLFA` package [Cole et al., 2018].

Model fit of nested models for can be compared with a chi-square difference test, as a way of testing measurement invariance. In this approach to testing measurement invariance, first model fit is evaluated in a a configural

Figure 14.22: Item information functions by sex

invariance model, in which the same number of factors are specified in each group, and which indicators load on which factors are the same in each group. Then, successive constraints are made across groups, including constraining factor loadings, intercepts, and residuals across groups. The metric ("weak factorial") invariance model is similar to the configural invariance model, but it constrains the factor loadings to be the same across groups. The scalar ("strong factorial") invariance model keeps the constraints of the metric invariance model, but it also constrains the intercepts to be the same across groups. The residual ("strict factorial") invariance model keeps the constraints of the scalar invariance model, but it also constrains the residuals to be the same across groups. The fit of each constrained model is compared to the previous model without such constraints. That is, the fit of the metric invariance model is compared to the fit of the configural invariance model, the fit of the scalar invariance model is compared to the fit of the metric invariance model, and the fit of the residual invariance model is compared to the fit of the scalar invariance model.

The chi-square difference test is sensitive to sample size, and trivial differences in fit can be detected with large samples [Cheung and Rensvold, 2002]. As a result, researchers also recommend examining change in additional criteria, including CFI, RMSEA, and SRMR [Chen, 2007]. Cheung and Rensvold [2002] recommend a cutoff of $\Delta\text{CFI} \geq -.01$ for identifying measurement

Figure 14.23: Item information functions by sex

non-invariance. Chen [2007] recommends the following cutoffs for identifying measurement non-invariance:

- with a small sample size (total $N \leq 300$):
  - testing invariance of factor loadings:
    - ΔCFI $\geq -.005$
    - supplemented by ΔRMSEA $\geq .010$ or ΔSRMR $\geq .025$
  - testing invariance of intercepts or residuals:
    - ΔCFI $\geq -.005$
    - supplemented by ΔRMSEA $\geq .010$ or ΔSRMR $\geq .005$
- with an adequate sample size (total $N > 300$):
  - testing invariance of factor loadings:
    - ΔCFI $\geq -.010$
    - supplemented by ΔRMSEA $\geq .015$ or ΔSRMR $\geq .030$
  - testing invariance of intercepts or residuals:
    - ΔCFI $\geq -.005$
    - supplemented by ΔRMSEA $\geq .015$ or ΔSRMR $\geq .010$

Figure 14.24: Item information functions by sex

Little et al. [2007] suggested that researchers establish at least partial invariance of factor loadings (metric invariance) to compare covariances across groups, and at least partial invariance of intercepts (scalar invariance) to compare mean levels across groups. Partial invariance refers to invariance with some but not all indicators. So, to examine associations with other variables, invariance of at least some factor loadings would be preferable. To examine differences in level or growth, invariance of at least some intercepts would be preferable. Residual invariance is considered overly restrictive, and it is not generally expected that one establish residual invariance [Little, 2013].

Although measurement invariance is important to test, it is also worth noting that tests of measurement invariance and DIF rest on various fundamentally untestable assumptions related to scale setting [Raykov et al., in press]. For instance, to give the latent factor units, oftentimes researchers set one item's factor loading to be equal across time (i.e., the marker variable). Tesults of factorial invariance tests can depend highly on which item is used as the anchor item for setting the scale of the latent factor [Belzak and Bauer, in press]. And subsequent tests of measurement invariance then rest on the fundamentally untestable assumption that changes in the level of the latent factor precipitates the same amount of change in the item across groups. If this scaling is not proper, however, it could lead to improper conclusions. Researchers are, therefore, not conclusively able to establish measurement

Figure 14.25: Item information functions by sex

invariance.

Several possible approaches may help address this. First, it can be helpful to place greater focus on *degree* of measurement invariance and confidence (rather than presence versus absence of measurement invariance). To the extent that non-invariance is trivial in effect size, it provides the researcher with greater confidence that they can use the measure to assess a construct in a comparable way over time. A number of studies describe how to test the effect size of measurement invariance [e.g., Liu et al., 2017] or differential item functioning [e.g., Meade, 2010]. Second, there may be important robustness checks. I describe each in greater detail below. One important sensitivity analysis could be to see if measurement invariance holds when using different marker variables. This would provide greater evidence that their apparent measurement invariance was not specific to one marker variable (i.e., one particular set of assumptions). A second robustness check would be to use effects coding, in which the average of items' factor loadings is equal to 1, so the metric of the latent variable is on the metric of all of the items rather than just one item [Little et al., 2006]. A third robustness check would be to use regularization, which is an alternative method to select the anchor items and to identify differential item functioning based on a machine learning technique that applies penalization to remove parameters that have little impact on model fit [Bauer et al., 2020, Belzak and Bauer, in press].

Figure 14.26: Item information functions by sex

Another issue with measurement invariance is that the null hypothesis that adding constraints across groups will not worsen fit is likely always false, because there are often at least slight differences across groups in factor loadings, intercepts, or residuals that reflecting sampling variability. However, we are not interested in trivial differences across groups that reflect sampling variability. Instead, we are interested in the extent to which there are substantive, meaningful differences across the groups of large enough magnitude to be practically meaningful. Thus, it can also be helpful to consider whether the measures show *approximate measurement invariance* [Van De Schoot et al., 2015]. For details on how to test approximate measurement invariance, see Van De Schoot et al. [2013].

Like DIF, when testing measurement invariance with many items or measures across many groups, there can be many tests, which will make it likely that measurement non-invariance will be detected, especially with a large sample. Some detected measurement non-invariance may be artificial or trivial, but other measurement non-invariance may be real and important. It is important to consider how you will proceed when detecting measurement non-invariance. Considerations of effect size and theory can be important for evaluating the measurement non-invariance and whether it is negligible or important to address. When detecting measurement non-invariance in a given parameter (e.g., factor loadings, intercepts, or residuals), one can identify the specific

Figure 14.27: Item information functions by sex

items that show measurement non-invariance in one of two primary ways: 1) starting with a model that allows the given parameter to differ across groups, a researcher can iteratively add constraints to identify the item(s) for which measurement invariance fails, or 2) starting with a model that constrains the given parameter to be the same across groups, a researcher can iteratively remove constraints to identify the item(s) for which measurement invariance becomes established (and by process of elimination, the items for which measurement invariance does not become established).

When detecting non-negligible measurement non-invariance, one option is to remove the items that show measurement non-invariance. Another option is to *resolve* items that show non-negligible measurement non-invariance, by keeping the item, but allowing the item's parameters to differ across groups (i.e., essentially treating it as a different item for each group). Removing items reduces a measure's reliability and ability to detect individual differences, but resolving measurement non-invariance has a very small effect on reliability and person separation [Hagquist and Andrich, 2017, Hagquist, 2019]. Where possible, it therefore can be beneficial to resolve non-negligible measurement non-invariance instead of removing items. If an item is clearly invalid in one group but valid in another group, another option is to keep the item in one group, and to remove it in another group. Addressing items that show larger measurement non-invariance can also reduce artificial measurement

Figure 14.28: Item information functions by sex

non-invariance in other items [Hagquist and Andrich, 2017]. Thus, it can also be beneficial to review and address measurement non-invariance sequentially by magnitude (from high to low).

14.5.1 Specify Models

14.5.1.1 Null Model

Fix residual variances and intercepts of manifest variables to be equal across groups.

```
nullModel <- '
#Fix residual variances of manifest variables to be equal across groups
x1 ~~ c(psi1, psi1)*x1
x2 ~~ c(psi2, psi2)*x2
x3 ~~ c(psi3, psi3)*x3
x4 ~~ c(psi4, psi4)*x4
x5 ~~ c(psi5, psi5)*x5
x6 ~~ c(psi6, psi6)*x6
x7 ~~ c(psi7, psi7)*x7
```

Figure 14.29: Item information functions by sex

```
x8 ~~ c(psi8, psi8)*x8
x9 ~~ c(psi9, psi9)*x9

#Fix intercepts of manifest variables to be equal across groups
x1 ~ c(tau1, tau1)*1
x2 ~ c(tau2, tau2)*1
x3 ~ c(tau3, tau3)*1
x4 ~ c(tau4, tau4)*1
x5 ~ c(tau5, tau5)*1
x6 ~ c(tau6, tau6)*1
x7 ~ c(tau7, tau7)*1
x8 ~ c(tau8, tau8)*1
x9 ~ c(tau9, tau9)*1
'
```

14.5.1.2 CFA Model

```
cfaModel <- '
 #Factor loadings
```

Figure 14.30: Item information functions by sex

```
 visual  =~ x1 + x2 + x3
 textual =~ x4 + x5 + x6
 speed   =~ x7 + x8 + x9
'
```

14.5.1.3 Configural Invariance

Specify the same number of factors in each group, and which indicators load on which factors are the same in each group.

```
cfaModel_configuralInvariance <- '
 #Factor loadings (free the factor loading of the first indicator)
 visual  =~ NA*x1 + x2 + x3
 textual =~ NA*x4 + x5 + x6
 speed   =~ NA*x7 + x8 + x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0
```

Figure 14.31: Item information functions by sex

```
#Fix latent variances to one
visual ~~ 1*visual
textual ~~ 1*textual
speed ~~ 1*speed

#Estimate covariances among latent variables
visual ~~ textual
visual ~~ speed
textual ~~ speed

#Estimate residual variances of manifest variables
x1 ~~ x1
x2 ~~ x2
x3 ~~ x3
x4 ~~ x4
x5 ~~ x5
x6 ~~ x6
x7 ~~ x7
x8 ~~ x8
x9 ~~ x9
```

```
 #Free intercepts of manifest variables
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'
```

14.5.1.4 Metric ("Weak Factorial") Invariance

Specify invariance of factor loadings across groups.

```
cfaModel_metricInvariance <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
```

```
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Free intercepts of manifest variables
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'
```

14.5.1.5 Scalar ("Strong Factorial") Invariance

Specify invariance of factor loadings and intercepts across groups.

```
cfaModel_scalarInvariance <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero in group 1; free latent means in group 2
 visual ~ c(0, NA)*1
 textual ~ c(0, NA)*1
 speed ~ c(0, NA)*1

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
```

```
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ c(intx2, intx2)*1
 x3 ~ c(intx3, intx3)*1
 x4 ~ c(intx4, intx4)*1
 x5 ~ c(intx5, intx5)*1
 x6 ~ c(intx6, intx6)*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'
```

14.5.1.6 Residual ("Strict Factorial") Invariance

Specify invariance of factor loadings, intercepts, and residuals across groups.

```
cfaModel_residualInvariance <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ c(0, NA)*1
 textual ~ c(0, NA)*1
 speed ~ c(0, NA)*1

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed
```

```
 #Fix residual variances of manifest variables across groups
 x1 ~~ c(residx1, residx1)*x1
 x2 ~~ c(residx2, residx2)*x2
 x3 ~~ c(residx3, residx3)*x3
 x4 ~~ c(residx4, residx4)*x4
 x5 ~~ c(residx5, residx5)*x5
 x6 ~~ c(residx6, residx6)*x6
 x7 ~~ c(residx7, residx7)*x7
 x8 ~~ c(residx8, residx8)*x8
 x9 ~~ c(residx9, residx9)*x9

 #Fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ c(intx2, intx2)*1
 x3 ~ c(intx3, intx3)*1
 x4 ~ c(intx4, intx4)*1
 x5 ~ c(intx5, intx5)*1
 x6 ~ c(intx6, intx6)*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'
```

14.5.2 Specify the Fit Indices of Interest

```
myAFIs <- c("chisq", "chisq.scaled",
            "rmsea", "cfi", "tli", "srmr",
            "rmsea.robust", "cfi.robust", "tli.robust")
moreAFIs <- NULL # c("gammaHat","gammaHat.scaled")
```

14.5.3 Null Model

14.5.3.1 Fit the Model

```
nullModelFit <- lavaan(nullModel,
                       data = HolzingerSwineford1939,
                       group = "school",
                       missing = "ML",
                       estimator = "MLR")
```

14.5.3.2 Model Summary

```
summary(nullModelFit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 41 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        36
##   Number of equality constraints                    18
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                                Standard      Robust
##   Test Statistic                                703.984     726.181
##   Degrees of freedom                                 90          90
##   P-value (Chi-square)                            0.000       0.000
##   Scaling correction factor                                   0.969
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                           358.143     369.435
##     1                                           345.842     356.746
##
## Model Test Baseline Model:
##
##   Test statistic                                609.050     638.762
##   Degrees of freedom                                 72          72
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.000       0.000
##   Tucker-Lewis Index (TLI)                        0.085       0.102
##
```

```
##   Robust Comparative Fit Index (CFI)                          0.000
##   Robust Tucker-Lewis Index (TLI)                             0.087
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2895.953   -2895.953
##   Scaling correction factor                                   0.500
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -2543.961   -2543.961
##   Scaling correction factor                                   0.975
##       for the MLR correction
##
##   Akaike (AIC)                                5827.907    5827.907
##   Bayesian (BIC)                              5891.579    5891.579
##   Sample-size adjusted Bayesian (BIC)         5834.515    5834.515
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.232       0.236
##   90 Percent confidence interval - lower         0.216       0.220
##   90 Percent confidence interval - upper         0.248       0.252
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                                0.232
##   90 Percent confidence interval - lower                      0.217
##   90 Percent confidence interval - upper                      0.248
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.291       0.291
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     x1      (tau1)    4.970    0.082   60.613    0.000    4.970    4.193
##     x2      (tau2)    6.147    0.080   76.367    0.000    6.147    5.295
##     x3      (tau3)    2.244    0.075   29.742    0.000    2.244    2.028
##     x4      (tau4)    3.048    0.082   37.318    0.000    3.048    2.588
```

```
##     x5      (tau5)    4.227    0.092   45.994    0.000    4.227    3.220
##     x6      (tau6)    2.140    0.075   28.384    0.000    2.140    1.931
##     x7      (tau7)    4.171    0.072   57.902    0.000    4.171    3.949
##     x8      (tau8)    5.500    0.064   86.112    0.000    5.500    5.832
##     x9      (tau9)    5.369    0.067   79.747    0.000    5.369    5.490
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     x1      (psi1)    1.405    0.158    8.877    0.000    1.405    1.000
##     x2      (psi2)    1.347    0.129   10.415    0.000    1.347    1.000
##     x3      (psi3)    1.224    0.090   13.640    0.000    1.224    1.000
##     x4      (psi4)    1.388    0.139   10.001    0.000    1.388    1.000
##     x5      (psi5)    1.723    0.145   11.922    0.000    1.723    1.000
##     x6      (psi6)    1.228    0.141    8.698    0.000    1.228    1.000
##     x7      (psi7)    1.116    0.103   10.847    0.000    1.116    1.000
##     x8      (psi8)    0.889    0.085   10.508    0.000    0.889    1.000
##     x9      (psi9)    0.956    0.096   10.007    0.000    0.956    1.000
##
##
## Group 2 [1]:
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     x1      (tau1)    4.970    0.082   60.613    0.000    4.970    4.193
##     x2      (tau2)    6.147    0.080   76.367    0.000    6.147    5.295
##     x3      (tau3)    2.244    0.075   29.742    0.000    2.244    2.028
##     x4      (tau4)    3.048    0.082   37.318    0.000    3.048    2.588
##     x5      (tau5)    4.227    0.092   45.994    0.000    4.227    3.220
##     x6      (tau6)    2.140    0.075   28.384    0.000    2.140    1.931
##     x7      (tau7)    4.171    0.072   57.902    0.000    4.171    3.949
##     x8      (tau8)    5.500    0.064   86.112    0.000    5.500    5.832
##     x9      (tau9)    5.369    0.067   79.747    0.000    5.369    5.490
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     x1      (psi1)    1.405    0.158    8.877    0.000    1.405    1.000
##     x2      (psi2)    1.347    0.129   10.415    0.000    1.347    1.000
##     x3      (psi3)    1.224    0.090   13.640    0.000    1.224    1.000
##     x4      (psi4)    1.388    0.139   10.001    0.000    1.388    1.000
##     x5      (psi5)    1.723    0.145   11.922    0.000    1.723    1.000
##     x6      (psi6)    1.228    0.141    8.698    0.000    1.228    1.000
##     x7      (psi7)    1.116    0.103   10.847    0.000    1.116    1.000
##     x8      (psi8)    0.889    0.085   10.508    0.000    0.889    1.000
##     x9      (psi9)    0.956    0.096   10.007    0.000    0.956    1.000
```

14.5.4 Configural Invariance

Specify the same number of factors in each group, and which indicators load on which factors are the same in each group.

14.5.4.1 Model Syntax

```
configuralInvarianceModel <- measEq.syntax(configural.model = cfaModel,
                                           data = HolzingerSwineford1939,
                                           ID.fac = "std.lv",
                                           group = "school")
```

```
cat(as.character(configuralInvarianceModel))
```

```
## ## LOADINGS:
##
## visual =~ c(NA, NA)*x1 + c(lambda.1_1.g1, lambda.1_1.g2)*x1
## visual =~ c(NA, NA)*x2 + c(lambda.2_1.g1, lambda.2_1.g2)*x2
## visual =~ c(NA, NA)*x3 + c(lambda.3_1.g1, lambda.3_1.g2)*x3
## textual =~ c(NA, NA)*x4 + c(lambda.4_2.g1, lambda.4_2.g2)*x4
## textual =~ c(NA, NA)*x5 + c(lambda.5_2.g1, lambda.5_2.g2)*x5
## textual =~ c(NA, NA)*x6 + c(lambda.6_2.g1, lambda.6_2.g2)*x6
## speed =~ c(NA, NA)*x7 + c(lambda.7_3.g1, lambda.7_3.g2)*x7
## speed =~ c(NA, NA)*x8 + c(lambda.8_3.g1, lambda.8_3.g2)*x8
## speed =~ c(NA, NA)*x9 + c(lambda.9_3.g1, lambda.9_3.g2)*x9
##
## ## INTERCEPTS:
##
## x1 ~ c(NA, NA)*1 + c(nu.1.g1, nu.1.g2)*1
## x2 ~ c(NA, NA)*1 + c(nu.2.g1, nu.2.g2)*1
## x3 ~ c(NA, NA)*1 + c(nu.3.g1, nu.3.g2)*1
## x4 ~ c(NA, NA)*1 + c(nu.4.g1, nu.4.g2)*1
## x5 ~ c(NA, NA)*1 + c(nu.5.g1, nu.5.g2)*1
## x6 ~ c(NA, NA)*1 + c(nu.6.g1, nu.6.g2)*1
## x7 ~ c(NA, NA)*1 + c(nu.7.g1, nu.7.g2)*1
## x8 ~ c(NA, NA)*1 + c(nu.8.g1, nu.8.g2)*1
## x9 ~ c(NA, NA)*1 + c(nu.9.g1, nu.9.g2)*1
##
## ## UNIQUE-FACTOR VARIANCES:
##
## x1 ~~ c(NA, NA)*x1 + c(theta.1_1.g1, theta.1_1.g2)*x1
## x2 ~~ c(NA, NA)*x2 + c(theta.2_2.g1, theta.2_2.g2)*x2
## x3 ~~ c(NA, NA)*x3 + c(theta.3_3.g1, theta.3_3.g2)*x3
```

```
## x4 ~~ c(NA, NA)*x4 + c(theta.4_4.g1, theta.4_4.g2)*x4
## x5 ~~ c(NA, NA)*x5 + c(theta.5_5.g1, theta.5_5.g2)*x5
## x6 ~~ c(NA, NA)*x6 + c(theta.6_6.g1, theta.6_6.g2)*x6
## x7 ~~ c(NA, NA)*x7 + c(theta.7_7.g1, theta.7_7.g2)*x7
## x8 ~~ c(NA, NA)*x8 + c(theta.8_8.g1, theta.8_8.g2)*x8
## x9 ~~ c(NA, NA)*x9 + c(theta.9_9.g1, theta.9_9.g2)*x9
##
## ## LATENT MEANS/INTERCEPTS:
##
## visual ~ c(0, 0)*1 + c(alpha.1.g1, alpha.1.g2)*1
## textual ~ c(0, 0)*1 + c(alpha.2.g1, alpha.2.g2)*1
## speed ~ c(0, 0)*1 + c(alpha.3.g1, alpha.3.g2)*1
##
## ## COMMON-FACTOR VARIANCES:
##
## visual ~~ c(1, 1)*visual + c(psi.1_1.g1, psi.1_1.g2)*visual
## textual ~~ c(1, 1)*textual + c(psi.2_2.g1, psi.2_2.g2)*textual
## speed ~~ c(1, 1)*speed + c(psi.3_3.g1, psi.3_3.g2)*speed
##
## ## COMMON-FACTOR COVARIANCES:
##
## visual ~~ c(NA, NA)*textual + c(psi.2_1.g1, psi.2_1.g2)*textual
## visual ~~ c(NA, NA)*speed + c(psi.3_1.g1, psi.3_1.g2)*speed
## textual ~~ c(NA, NA)*speed + c(psi.3_2.g1, psi.3_2.g2)*speed
```

```
configuralInvarianceSyntax <- as.character(configuralInvarianceModel)
```

```
summary(configuralInvarianceModel)
```

14.5.4.1.1 Summary of Model Features

```
## This lavaan model syntax specifies a CFA with 9 manifest indicators of 3 common factor(s).
##
## To identify the location and scale of each common factor, the factor means and variances were
##
## Pattern matrix indicating num(eric), ord(ered), and lat(ent) indicators per factor:
##
##    visual textual speed
## x1 num
## x2 num
## x3 num
## x4        num
```

```
## x5        num
## x6        num
## x7                num
## x8                num
## x9                num
##
##
## This model hypothesizes only configural invariance.
```

```
lavaanify(cfaModel, ngroups = 2)
```

14.5.4.1.2 Model Syntax in Table Form

```
##    id     lhs op     rhs user block group free ustart exo label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0         .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0         .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0         .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0         .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0         .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0         .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0         .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0         .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0         .p9.
## 10 10      x1 ~~      x1    0     1     1    0      0   0        .p10.
## 11 11      x2 ~~      x2    0     1     1    0      0   0        .p11.
## 12 12      x3 ~~      x3    0     1     1    0      0   0        .p12.
## 13 13      x4 ~~      x4    0     1     1    0      0   0        .p13.
## 14 14      x5 ~~      x5    0     1     1    0      0   0        .p14.
## 15 15      x6 ~~      x6    0     1     1    0      0   0        .p15.
## 16 16      x7 ~~      x7    0     1     1    0      0   0        .p16.
## 17 17      x8 ~~      x8    0     1     1    0      0   0        .p17.
## 18 18      x9 ~~      x9    0     1     1    0      0   0        .p18.
## 19 19  visual ~~  visual    0     1     1    0      0   0        .p19.
## 20 20 textual ~~ textual    0     1     1    0      0   0        .p20.
## 21 21   speed ~~   speed    0     1     1    0      0   0        .p21.
## 22 22  visual =~      x1    1     2     2   10     NA   0        .p22.
## 23 23  visual =~      x2    1     2     2   11     NA   0        .p23.
## 24 24  visual =~      x3    1     2     2   12     NA   0        .p24.
## 25 25 textual =~      x4    1     2     2   13     NA   0        .p25.
## 26 26 textual =~      x5    1     2     2   14     NA   0        .p26.
## 27 27 textual =~      x6    1     2     2   15     NA   0        .p27.
## 28 28   speed =~      x7    1     2     2   16     NA   0        .p28.
```

```
## 29 29   speed =~      x8    1     2     2   17     NA   0        .p29.
## 30 30   speed =~      x9    1     2     2   18     NA   0        .p30.
## 31 31      x1 ~~      x1    0     2     2    0      0   0        .p31.
## 32 32      x2 ~~      x2    0     2     2    0      0   0        .p32.
## 33 33      x3 ~~      x3    0     2     2    0      0   0        .p33.
## 34 34      x4 ~~      x4    0     2     2    0      0   0        .p34.
## 35 35      x5 ~~      x5    0     2     2    0      0   0        .p35.
## 36 36      x6 ~~      x6    0     2     2    0      0   0        .p36.
## 37 37      x7 ~~      x7    0     2     2    0      0   0        .p37.
## 38 38      x8 ~~      x8    0     2     2    0      0   0        .p38.
## 39 39      x9 ~~      x9    0     2     2    0      0   0        .p39.
## 40 40  visual ~~  visual    0     2     2    0      0   0        .p40.
## 41 41 textual ~~ textual    0     2     2    0      0   0        .p41.
## 42 42   speed ~~   speed    0     2     2    0      0   0        .p42.
```

14.5.4.2 Fit the Model

```
configuralInvarianceModel_fit <- cfa(configuralInvarianceSyntax,
                                     data = HolzingerSwineford1939,
                                     group = "school",
                                     missing = "ML",
                                     estimator = "MLR")
```

```
configuralInvarianceModelFullSyntax_fit <- lavaan(cfaModel_configuralInvariance,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")
```

14.5.4.3 Model Summary

```
summary(configuralInvarianceModel_fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 61 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        60
```

```
##
##   Number of observations per group:
##     2                                                132
##     1                                                122
##   Number of missing patterns per group:
##     2                                                 51
##     1                                                 56
##
## Model Test User Model:
##                                                Standard      Robust
##   Test Statistic                                 73.311      82.543
##   Degrees of freedom                                 48          48
##   P-value (Chi-square)                            0.011       0.001
##   Scaling correction factor                                   0.888
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                            39.740      44.744
##     1                                            33.571      37.799
##
## Model Test Baseline Model:
##
##   Test statistic                                609.050     638.762
##   Degrees of freedom                                 72          72
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.953       0.939
##   Tucker-Lewis Index (TLI)                        0.929       0.909
##
##   Robust Comparative Fit Index (CFI)                          0.943
##   Robust Tucker-Lewis Index (TLI)                             0.915
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2580.617   -2580.617
##   Scaling correction factor                                   1.044
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -2543.961   -2543.961
##   Scaling correction factor                                   0.975
##       for the MLR correction
##
##   Akaike (AIC)                                5281.233    5281.233
##   Bayesian (BIC)                              5493.473    5493.473
##   Sample-size adjusted Bayesian (BIC)         5303.261    5303.261
```

```
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.064       0.075
##   90 Percent confidence interval - lower         0.032       0.045
##   90 Percent confidence interval - upper         0.093       0.104
##   P-value RMSEA <= 0.05                          0.204       0.081
##
##   Robust RMSEA                                               0.071
##   90 Percent confidence interval - lower                     0.044
##   90 Percent confidence interval - upper                     0.096
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.068       0.068
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    1.168    0.179    6.539    0.000    1.168    0.941
##     x2      (l.2_)    0.399    0.153    2.615    0.009    0.399    0.318
##     x3      (l.3_)    0.648    0.144    4.511    0.000    0.648    0.563
##   textual =~
##     x4      (l.4_)    1.001    0.096   10.460    0.000    1.001    0.838
##     x5      (l.5_)    1.190    0.111   10.755    0.000    1.190    0.890
##     x6      (l.6_)    0.818    0.104    7.852    0.000    0.818    0.843
##   speed =~
##     x7      (l.7_)    0.521    0.161    3.231    0.001    0.521    0.499
##     x8      (l.8_)    0.608    0.150    4.053    0.000    0.608    0.615
##     x9      (l.9_)    0.718    0.192    3.744    0.000    0.718    0.699
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.533    0.080    6.634    0.000    0.533    0.533
##     speed  (p.3_1)    0.445    0.183    2.430    0.015    0.445    0.445
##   textual ~~
```

```
##     speed  (p.3_2)    0.324    0.105    3.082    0.002    0.324    0.324
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (n.1.)    4.921    0.113   43.445    0.000    4.921    3.963
##    .x2      (n.2.)    6.025    0.120   50.409    0.000    6.025    4.805
##    .x3      (n.3.)    2.518    0.106   23.836    0.000    2.518    2.186
##    .x4      (n.4.)    2.837    0.108   26.165    0.000    2.837    2.375
##    .x5      (n.5.)    3.875    0.120   32.327    0.000    3.875    2.897
##    .x6      (n.6.)    1.875    0.089   21.141    0.000    1.875    1.933
##    .x7      (n.7.)    4.440    0.096   46.235    0.000    4.440    4.250
##    .x8      (n.8.)    5.581    0.092   60.781    0.000    5.581    5.643
##    .x9      (n.9.)    5.391    0.095   56.462    0.000    5.391    5.250
##     visual  (a.1.)    0.000                               0.000    0.000
##     textual (a.2.)    0.000                               0.000    0.000
##     speed   (a.3.)    0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.177    0.369    0.479    0.632    0.177    0.115
##    .x2      (t.2_)    1.413    0.230    6.139    0.000    1.413    0.899
##    .x3      (t.3_)    0.907    0.171    5.310    0.000    0.907    0.684
##    .x4      (t.4_)    0.425    0.086    4.933    0.000    0.425    0.298
##    .x5      (t.5_)    0.373    0.103    3.611    0.000    0.373    0.209
##    .x6      (t.6_)    0.272    0.059    4.646    0.000    0.272    0.289
##    .x7      (t.7_)    0.820    0.167    4.915    0.000    0.820    0.751
##    .x8      (t.8_)    0.608    0.149    4.084    0.000    0.608    0.622
##    .x9      (t.9_)    0.539    0.244    2.205    0.027    0.539    0.511
##     visual  (p.1_)    1.000                               1.000    1.000
##     textual (p.2_)    1.000                               1.000    1.000
##     speed   (p.3_)    1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.885
##     x2                0.101
##     x3                0.316
##     x4                0.702
##     x5                0.791
##     x6                0.711
##     x7                0.249
##     x8                0.378
##     x9                0.489
##
##
## Group 2 [1]:
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    0.590    0.198    2.976    0.003    0.590    0.531
##     x2      (l.2_)    0.445    0.150    2.962    0.003    0.445    0.431
##     x3      (l.3_)    0.542    0.170    3.185    0.001    0.542    0.543
##   textual =~
##     x4      (l.4_)    0.994    0.095   10.440    0.000    0.994    0.871
##     x5      (l.5_)    1.034    0.108    9.558    0.000    1.034    0.862
##     x6      (l.6_)    0.958    0.091   10.539    0.000    0.958    0.849
##   speed =~
##     x7      (l.7_)    0.669    0.108    6.213    0.000    0.669    0.684
##     x8      (l.8_)    0.727    0.105    6.895    0.000    0.727    0.811
##     x9      (l.9_)    0.548    0.111    4.929    0.000    0.548    0.593
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.666    0.165    4.035    0.000    0.666    0.666
##     speed  (p.3_1)    0.441    0.192    2.304    0.021    0.441    0.441
##   textual ~~
##     speed  (p.3_2)    0.252    0.147    1.722    0.085    0.252    0.252
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (n.1.)    5.013    0.108   46.296    0.000    5.013    4.507
##    .x2      (n.2.)    6.260    0.103   60.544    0.000    6.260    6.058
##    .x3      (n.3.)    1.991    0.098   20.314    0.000    1.991    1.995
##    .x4      (n.4.)    3.281    0.108   30.285    0.000    3.281    2.874
##    .x5      (n.5.)    4.682    0.116   40.278    0.000    4.682    3.905
##    .x6      (n.6.)    2.510    0.107   23.388    0.000    2.510    2.224
##    .x7      (n.7.)    3.863    0.095   40.611    0.000    3.863    3.948
##    .x8      (n.8.)    5.450    0.085   64.021    0.000    5.450    6.083
##    .x9      (n.9.)    5.342    0.091   58.903    0.000    5.342    5.781
##     visual  (a.1.)    0.000                                0.000    0.000
##     textual (a.2.)    0.000                                0.000    0.000
##     speed   (a.3.)    0.000                                0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.889    0.245    3.623    0.000    0.889    0.718
##    .x2      (t.2_)    0.870    0.161    5.390    0.000    0.870    0.815
##    .x3      (t.3_)    0.703    0.184    3.810    0.000    0.703    0.706
##    .x4      (t.4_)    0.315    0.079    4.018    0.000    0.315    0.242
##    .x5      (t.5_)    0.369    0.076    4.857    0.000    0.369    0.257
```

```
##    .x6      (t.6_)    0.355    0.086    4.153    0.000    0.355    0.279
##    .x7      (t.7_)    0.510    0.103    4.936    0.000    0.510    0.532
##    .x8      (t.8_)    0.275    0.113    2.441    0.015    0.275    0.342
##    .x9      (t.9_)    0.553    0.114    4.864    0.000    0.553    0.648
##     visual  (p.1_)    1.000                                1.000    1.000
##     textual (p.2_)    1.000                                1.000    1.000
##     speed   (p.3_)    1.000                                1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.282
##     x2                0.185
##     x3                0.294
##     x4                0.758
##     x5                0.743
##     x6                0.721
##     x7                0.468
##     x8                0.658
##     x9                0.352
```

```
summary(configuralInvarianceModelFullSyntax_fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 61 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        60
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                73.311      82.543
##   Degrees of freedom                                48          48
##   P-value (Chi-square)                           0.011       0.001
##   Scaling correction factor                                  0.888
##        Yuan-Bentler correction (Mplus variant)
```

```
##   Test statistic for each group:
##     2                                          39.740      44.744
##     1                                          33.571      37.799
##
## Model Test Baseline Model:
##
##   Test statistic                              609.050     638.762
##   Degrees of freedom                               72          72
##   P-value                                       0.000       0.000
##   Scaling correction factor                                 0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                   0.953       0.939
##   Tucker-Lewis Index (TLI)                      0.929       0.909
##
##   Robust Comparative Fit Index (CFI)                        0.943
##   Robust Tucker-Lewis Index (TLI)                           0.915
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -2580.617   -2580.617
##   Scaling correction factor                                 1.044
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)     -2543.961   -2543.961
##   Scaling correction factor                                 0.975
##       for the MLR correction
##
##   Akaike (AIC)                               5281.233    5281.233
##   Bayesian (BIC)                             5493.473    5493.473
##   Sample-size adjusted Bayesian (BIC)        5303.261    5303.261
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.064       0.075
##   90 Percent confidence interval - lower        0.032       0.045
##   90 Percent confidence interval - upper        0.093       0.104
##   P-value RMSEA <= 0.05                         0.204       0.081
##
##   Robust RMSEA                                              0.071
##   90 Percent confidence interval - lower                    0.044
##   90 Percent confidence interval - upper                    0.096
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.068       0.068
```

```
##
## Parameter Estimates:
##
##   Standard errors                              Sandwich
##   Information bread                            Observed
##   Observed information based on                 Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1                1.168    0.179    6.539    0.000    1.168    0.941
##     x2                0.399    0.153    2.615    0.009    0.399    0.318
##     x3                0.648    0.144    4.511    0.000    0.648    0.563
##   textual =~
##     x4                1.001    0.096   10.460    0.000    1.001    0.838
##     x5                1.190    0.111   10.755    0.000    1.190    0.890
##     x6                0.818    0.104    7.852    0.000    0.818    0.843
##   speed =~
##     x7                0.521    0.161    3.231    0.001    0.521    0.499
##     x8                0.608    0.150    4.053    0.000    0.608    0.615
##     x9                0.718    0.192    3.744    0.000    0.718    0.699
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.533    0.080    6.634    0.000    0.533    0.533
##     speed             0.445    0.183    2.430    0.015    0.445    0.445
##   textual ~~
##     speed             0.324    0.105    3.082    0.002    0.324    0.324
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1                4.921    0.113   43.445    0.000    4.921    3.963
##    .x2                6.025    0.120   50.409    0.000    6.025    4.805
##    .x3                2.518    0.106   23.836    0.000    2.518    2.186
##    .x4                2.837    0.108   26.165    0.000    2.837    2.375
##    .x5                3.875    0.120   32.327    0.000    3.875    2.897
##    .x6                1.875    0.089   21.141    0.000    1.875    1.933
##    .x7                4.440    0.096   46.235    0.000    4.440    4.250
##    .x8                5.581    0.092   60.781    0.000    5.581    5.643
```

```
##    .x9                5.391    0.095   56.462    0.000    5.391    5.250
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1                0.177    0.369    0.479    0.632    0.177    0.115
##    .x2                1.413    0.230    6.139    0.000    1.413    0.899
##    .x3                0.907    0.171    5.310    0.000    0.907    0.684
##    .x4                0.425    0.086    4.933    0.000    0.425    0.298
##    .x5                0.373    0.103    3.611    0.000    0.373    0.209
##    .x6                0.272    0.059    4.646    0.000    0.272    0.289
##    .x7                0.820    0.167    4.915    0.000    0.820    0.751
##    .x8                0.608    0.149    4.084    0.000    0.608    0.622
##    .x9                0.539    0.244    2.205    0.027    0.539    0.511
##
## R-Square:
##                    Estimate
##     x1                0.885
##     x2                0.101
##     x3                0.316
##     x4                0.702
##     x5                0.791
##     x6                0.711
##     x7                0.249
##     x8                0.378
##     x9                0.489
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1                0.590    0.198    2.976    0.003    0.590    0.531
##     x2                0.445    0.150    2.962    0.003    0.445    0.431
##     x3                0.542    0.170    3.185    0.001    0.542    0.543
##   textual =~
##     x4                0.994    0.095   10.440    0.000    0.994    0.871
##     x5                1.034    0.108    9.558    0.000    1.034    0.862
##     x6                0.958    0.091   10.539    0.000    0.958    0.849
##   speed =~
##     x7                0.669    0.108    6.213    0.000    0.669    0.684
##     x8                0.727    0.105    6.895    0.000    0.727    0.811
##     x9                0.548    0.111    4.929    0.000    0.548    0.593
```

```
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.666    0.165    4.035    0.000    0.666    0.666
##     speed             0.441    0.192    2.304    0.021    0.441    0.441
##   textual ~~
##     speed             0.252    0.147    1.722    0.085    0.252    0.252
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1                5.013    0.108   46.296    0.000    5.013    4.507
##    .x2                6.260    0.103   60.544    0.000    6.260    6.058
##    .x3                1.991    0.098   20.314    0.000    1.991    1.995
##    .x4                3.281    0.108   30.285    0.000    3.281    2.874
##    .x5                4.682    0.116   40.278    0.000    4.682    3.905
##    .x6                2.510    0.107   23.388    0.000    2.510    2.224
##    .x7                3.863    0.095   40.611    0.000    3.863    3.948
##    .x8                5.450    0.085   64.021    0.000    5.450    6.083
##    .x9                5.342    0.091   58.903    0.000    5.342    5.781
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1                0.889    0.245    3.623    0.000    0.889    0.718
##    .x2                0.870    0.161    5.390    0.000    0.870    0.815
##    .x3                0.703    0.184    3.810    0.000    0.703    0.706
##    .x4                0.315    0.079    4.018    0.000    0.315    0.242
##    .x5                0.369    0.076    4.857    0.000    0.369    0.257
##    .x6                0.355    0.086    4.153    0.000    0.355    0.279
##    .x7                0.510    0.103    4.936    0.000    0.510    0.532
##    .x8                0.275    0.113    2.441    0.015    0.275    0.342
##    .x9                0.553    0.114    4.864    0.000    0.553    0.648
##
## R-Square:
##                    Estimate
##     x1                0.282
##     x2                0.185
##     x3                0.294
##     x4                0.758
##     x5                0.743
```

```
##     x6                0.721
##     x7                0.468
##     x8                0.658
##     x9                0.352
```

14.5.4.4 Model Fit

You can specify the null model as the baseline model using: `baseline.model = nullModelFit`

```
fitMeasures(configuralInvarianceModel_fit)
```

```
##                          npar                         fmin
##                        60.000                        0.144
##                         chisq                           df
##                        73.311                       48.000
##                        pvalue                  chisq.scaled
##                         0.011                       82.543
##                     df.scaled                 pvalue.scaled
##                        48.000                        0.001
##          chisq.scaling.factor                baseline.chisq
##                         0.888                      609.050
##                   baseline.df               baseline.pvalue
##                        72.000                        0.000
##         baseline.chisq.scaled            baseline.df.scaled
##                       638.762                       72.000
##        baseline.pvalue.scaled baseline.chisq.scaling.factor
##                         0.000                        0.953
##                           cfi                          tli
##                         0.953                        0.929
##                          nnfi                          rfi
##                         0.929                        0.819
##                           nfi                         pnfi
##                         0.880                        0.586
##                           ifi                          rni
##                         0.955                        0.953
##                    cfi.scaled                    tli.scaled
##                         0.939                        0.909
##                    cfi.robust                    tli.robust
##                         0.943                        0.915
##                   nnfi.scaled                   nnfi.robust
##                         0.909                        0.915
##                    rfi.scaled                    nfi.scaled
##                         0.806                        0.871
```

```
##                    ifi.scaled                    rni.scaled
##                         0.942                         0.939
##                    rni.robust                          logl
##                         0.943                     -2580.617
##             unrestricted.logl                           aic
##                     -2543.961                      5281.233
##                           bic                        ntotal
##                      5493.473                       254.000
##                          bic2             scaling.factor.h1
##                      5303.261                         0.975
##             scaling.factor.h0                         rmsea
##                         1.044                         0.064
##                rmsea.ci.lower                rmsea.ci.upper
##                         0.032                         0.093
##                  rmsea.pvalue                  rmsea.scaled
##                         0.204                         0.075
##         rmsea.ci.lower.scaled         rmsea.ci.upper.scaled
##                         0.045                         0.104
##           rmsea.pvalue.scaled                  rmsea.robust
##                         0.081                         0.071
##         rmsea.ci.lower.robust         rmsea.ci.upper.robust
##                         0.044                         0.096
##           rmsea.pvalue.robust                           rmr
##                           NA                         0.082
##                    rmr_nomean                          srmr
##                         0.090                         0.068
##                  srmr_bentler           srmr_bentler_nomean
##                         0.068                         0.075
##                          crmr                   crmr_nomean
##                         0.075                         0.084
##                     srmr_mplus             srmr_mplus_nomean
##                         0.069                         0.075
##                          cn_05                         cn_01
##                       226.797                       256.288
##                           gfi                          agfi
##                         0.993                         0.985
##                          pgfi                           mfi
##                         0.441                         0.951
```

```
configuralInvarianceModelFitIndices <- fitMeasures(configuralInvarianceModel_fit)[c("c
```

14.5.4.5 Effect size of non-invariance

14.5.4.5.1 dMACS The effect size of measurement non-invariance, as described by Nye et al. [2019], was calculated using the `dmacs` package

[Dueber, 2019].

```
lavaan_dmacs(configuralInvarianceModel_fit)
```

```
## $DMACS
##       visual   textual     speed
## x1 0.4929182        NA        NA
## x2 0.2071013        NA        NA
## x3 0.4957645        NA        NA
## x4        NA 0.3818460        NA
## x5        NA 0.6665635        NA
## x6        NA 0.6082664        NA
## x7        NA        NA 0.5828214
## x8        NA        NA 0.1870107
## x9        NA        NA 0.1806999
##
## $ItemDeltaMean
##         visual   textual      speed
## x1  0.09258276        NA         NA
## x2  0.23472817        NA         NA
## x3 -0.52694912        NA         NA
## x4          NA 0.4439753         NA
## x5          NA 0.8063651         NA
## x6          NA 0.6342353         NA
## x7          NA        NA -0.5769773
## x8          NA        NA -0.1307670
## x9          NA        NA -0.0493774
##
## $MeanDiff
##     visual    textual      speed
## -0.1996382  1.8845757 -0.7571216
```

14.5.4.6 Permutation Test

Permutation procedures for testing measurement invariance are described in [Jorgensen et al., 2018].

```
numPermutations <- 1000
```

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed. You can specify the null model as the baseline model using: `baseline.model = nullModelFit`. **Warning**: this code takes a while to run based on 1000 iterations. You can reduce the number of iterations to be faster.

```
configuralInvarianceTest <- permuteMeasEq(nPermute = numPermutations,
                                          modelType = "mgcfa",
                                          con = configuralInvarianceModel_fit,
                                          uncon = NULL,
                                          AFIs = myAFIs,
                                          moreAFIs = moreAFIs,
                                          iseed = 52242)
```

```
configuralInvarianceTest
```

```
## Omnibus p value based on parametric chi-squared difference test:
##
## Chisq diff    Df diff Pr(>Chisq)
##     82.543     48.000      0.001
##
##
## Omnibus p values based on nonparametric permutation method:
##
##              AFI.Difference p.value
## chisq                73.311   0.844
## chisq.scaled         82.543   0.732
## rmsea                 0.064   0.844
## cfi                   0.953   0.836
## tli                   0.929   0.836
## srmr                  0.068   0.895
## rmsea.robust          0.071   0.790
## cfi.robust            0.943   0.784
## tli.robust            0.915   0.784
```

14.5.4.7 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(configuralInvarianceModel_fit)
```

```
## $`2`
##           visual   textual     speed
## alpha  0.6524461 0.8841445 0.6430042
## omega  0.6627992 0.8943290 0.6343158
## omega2 0.6627992 0.8943290 0.6343158
## omega3 0.6276424 0.8970070 0.6221817
```

```
## avevar 0.4377562 0.7426442 0.3703283
##
## $`1`
##           visual   textual     speed
## alpha  0.4648456 0.8955126 0.7327296
## omega  0.5025716 0.8955020 0.7385482
## omega2 0.5025716 0.8955020 0.7385482
## omega3 0.5196544 0.8883048 0.7371072
## avevar 0.2544172 0.7408845 0.4882357
```

14.5.4.8 Path Diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(configuralInvarianceModel_fit,
         what = "est",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         ask = FALSE,
         edge.label.cex = 1.2)
```

14.5.5 Metric ("Weak Factorial") Invariance Model

Specify invariance of factor loadings across groups.

14.5.5.1 Model Syntax

```
metricInvarianceModel <- measEq.syntax(configural.model = cfaModel,
                                       data = HolzingerSwineford1939,
                                       ID.fac = "std.lv",
                                       group = "school",
                                       group.equal = "loadings")
```

```
cat(as.character(metricInvarianceModel))
```

```
## ## LOADINGS:
##
## visual =~ c(NA, NA)*x1 + c(lambda.1_1, lambda.1_1)*x1
```

Figure 14.32: Configural invariance model in confirmatory factor analysis

Figure 14.33: Configural invariance model in confirmatory factor analysis

```
## visual =~ c(NA, NA)*x2 + c(lambda.2_1, lambda.2_1)*x2
## visual =~ c(NA, NA)*x3 + c(lambda.3_1, lambda.3_1)*x3
## textual =~ c(NA, NA)*x4 + c(lambda.4_2, lambda.4_2)*x4
## textual =~ c(NA, NA)*x5 + c(lambda.5_2, lambda.5_2)*x5
## textual =~ c(NA, NA)*x6 + c(lambda.6_2, lambda.6_2)*x6
## speed =~ c(NA, NA)*x7 + c(lambda.7_3, lambda.7_3)*x7
## speed =~ c(NA, NA)*x8 + c(lambda.8_3, lambda.8_3)*x8
## speed =~ c(NA, NA)*x9 + c(lambda.9_3, lambda.9_3)*x9
##
## ## INTERCEPTS:
##
## x1 ~ c(NA, NA)*1 + c(nu.1.g1, nu.1.g2)*1
## x2 ~ c(NA, NA)*1 + c(nu.2.g1, nu.2.g2)*1
## x3 ~ c(NA, NA)*1 + c(nu.3.g1, nu.3.g2)*1
## x4 ~ c(NA, NA)*1 + c(nu.4.g1, nu.4.g2)*1
## x5 ~ c(NA, NA)*1 + c(nu.5.g1, nu.5.g2)*1
## x6 ~ c(NA, NA)*1 + c(nu.6.g1, nu.6.g2)*1
## x7 ~ c(NA, NA)*1 + c(nu.7.g1, nu.7.g2)*1
## x8 ~ c(NA, NA)*1 + c(nu.8.g1, nu.8.g2)*1
## x9 ~ c(NA, NA)*1 + c(nu.9.g1, nu.9.g2)*1
##
## ## UNIQUE-FACTOR VARIANCES:
##
## x1 ~~ c(NA, NA)*x1 + c(theta.1_1.g1, theta.1_1.g2)*x1
## x2 ~~ c(NA, NA)*x2 + c(theta.2_2.g1, theta.2_2.g2)*x2
## x3 ~~ c(NA, NA)*x3 + c(theta.3_3.g1, theta.3_3.g2)*x3
## x4 ~~ c(NA, NA)*x4 + c(theta.4_4.g1, theta.4_4.g2)*x4
## x5 ~~ c(NA, NA)*x5 + c(theta.5_5.g1, theta.5_5.g2)*x5
## x6 ~~ c(NA, NA)*x6 + c(theta.6_6.g1, theta.6_6.g2)*x6
## x7 ~~ c(NA, NA)*x7 + c(theta.7_7.g1, theta.7_7.g2)*x7
## x8 ~~ c(NA, NA)*x8 + c(theta.8_8.g1, theta.8_8.g2)*x8
## x9 ~~ c(NA, NA)*x9 + c(theta.9_9.g1, theta.9_9.g2)*x9
##
## ## LATENT MEANS/INTERCEPTS:
##
## visual ~ c(0, 0)*1 + c(alpha.1.g1, alpha.1.g2)*1
## textual ~ c(0, 0)*1 + c(alpha.2.g1, alpha.2.g2)*1
## speed ~ c(0, 0)*1 + c(alpha.3.g1, alpha.3.g2)*1
##
## ## COMMON-FACTOR VARIANCES:
##
## visual ~~ c(1, NA)*visual + c(psi.1_1.g1, psi.1_1.g2)*visual
## textual ~~ c(1, NA)*textual + c(psi.2_2.g1, psi.2_2.g2)*textual
## speed ~~ c(1, NA)*speed + c(psi.3_3.g1, psi.3_3.g2)*speed
##
## ## COMMON-FACTOR COVARIANCES:
```

```
##
## visual ~~ c(NA, NA)*textual + c(psi.2_1.g1, psi.2_1.g2)*textual
## visual ~~ c(NA, NA)*speed + c(psi.3_1.g1, psi.3_1.g2)*speed
## textual ~~ c(NA, NA)*speed + c(psi.3_2.g1, psi.3_2.g2)*speed
```

```
metricInvarianceSyntax <- as.character(metricInvarianceModel)
```

```
summary(metricInvarianceModel)
```

14.5.5.1.1 Summary of Model Features

```
## This lavaan model syntax specifies a CFA with 9 manifest indicators of 3 common factor(s).
##
## To identify the location and scale of each common factor, the factor means and variances were
##
## Pattern matrix indicating num(eric), ord(ered), and lat(ent) indicators per factor:
##
##    visual textual speed
## x1 num
## x2 num
## x3 num
## x4        num
## x5        num
## x6        num
## x7                num
## x8                num
## x9                num
##
## The following types of parameter were constrained to equality across groups:
##
##  loadings
```

14.5.5.2 Model Syntax in Table Form

```
lavaanify(metricInvarianceModel, ngroups = 2)
```

```
##    id     lhs op     rhs user block group free ustart exo        label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0   lambda.1_1   .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0   lambda.2_1   .p2.
```

```
## 3   3  visual =~      x3    1     1     1    3     NA   0   lambda.3_1   .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0   lambda.4_2   .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0   lambda.5_2   .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0   lambda.6_2   .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0   lambda.7_3   .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0   lambda.8_3   .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0   lambda.9_3   .p9.
## 10 10      x1 ~1            1     1     1   10     NA   0      nu.1.g1  .p10.
## 11 11      x2 ~1            1     1     1   11     NA   0      nu.2.g1  .p11.
## 12 12      x3 ~1            1     1     1   12     NA   0      nu.3.g1  .p12.
## 13 13      x4 ~1            1     1     1   13     NA   0      nu.4.g1  .p13.
## 14 14      x5 ~1            1     1     1   14     NA   0      nu.5.g1  .p14.
## 15 15      x6 ~1            1     1     1   15     NA   0      nu.6.g1  .p15.
## 16 16      x7 ~1            1     1     1   16     NA   0      nu.7.g1  .p16.
## 17 17      x8 ~1            1     1     1   17     NA   0      nu.8.g1  .p17.
## 18 18      x9 ~1            1     1     1   18     NA   0      nu.9.g1  .p18.
## 19 19      x1 ~~      x1    1     1     1   19     NA   0 theta.1_1.g1  .p19.
## 20 20      x2 ~~      x2    1     1     1   20     NA   0 theta.2_2.g1  .p20.
## 21 21      x3 ~~      x3    1     1     1   21     NA   0 theta.3_3.g1  .p21.
## 22 22      x4 ~~      x4    1     1     1   22     NA   0 theta.4_4.g1  .p22.
## 23 23      x5 ~~      x5    1     1     1   23     NA   0 theta.5_5.g1  .p23.
## 24 24      x6 ~~      x6    1     1     1   24     NA   0 theta.6_6.g1  .p24.
## 25 25      x7 ~~      x7    1     1     1   25     NA   0 theta.7_7.g1  .p25.
## 26 26      x8 ~~      x8    1     1     1   26     NA   0 theta.8_8.g1  .p26.
## 27 27      x9 ~~      x9    1     1     1   27     NA   0 theta.9_9.g1  .p27.
## 28 28  visual ~1            1     1     1    0      0   0   alpha.1.g1  .p28.
## 29 29 textual ~1            1     1     1    0      0   0   alpha.2.g1  .p29.
## 30 30   speed ~1            1     1     1    0      0   0   alpha.3.g1  .p30.
## 31 31  visual ~~  visual    1     1     1    0      1   0   psi.1_1.g1  .p31.
## 32 32 textual ~~ textual    1     1     1    0      1   0   psi.2_2.g1  .p32.
## 33 33   speed ~~   speed    1     1     1    0      1   0   psi.3_3.g1  .p33.
## 34 34  visual ~~ textual    1     1     1   28     NA   0   psi.2_1.g1  .p34.
## 35 35  visual ~~   speed    1     1     1   29     NA   0   psi.3_1.g1  .p35.
## 36 36 textual ~~   speed    1     1     1   30     NA   0   psi.3_2.g1  .p36.
## 37 37  visual =~      x1    1     2     2   31     NA   0   lambda.1_1  .p37.
## 38 38  visual =~      x2    1     2     2   32     NA   0   lambda.2_1  .p38.
## 39 39  visual =~      x3    1     2     2   33     NA   0   lambda.3_1  .p39.
## 40 40 textual =~      x4    1     2     2   34     NA   0   lambda.4_2  .p40.
## 41 41 textual =~      x5    1     2     2   35     NA   0   lambda.5_2  .p41.
## 42 42 textual =~      x6    1     2     2   36     NA   0   lambda.6_2  .p42.
## 43 43   speed =~      x7    1     2     2   37     NA   0   lambda.7_3  .p43.
## 44 44   speed =~      x8    1     2     2   38     NA   0   lambda.8_3  .p44.
## 45 45   speed =~      x9    1     2     2   39     NA   0   lambda.9_3  .p45.
## 46 46      x1 ~1            1     2     2   40     NA   0      nu.1.g2  .p46.
## 47 47      x2 ~1            1     2     2   41     NA   0      nu.2.g2  .p47.
## 48 48      x3 ~1            1     2     2   42     NA   0      nu.3.g2  .p48.
```

```
## 49 49      x4 ~1            1     2     2   43     NA   0      nu.4.g2  .p49.
## 50 50      x5 ~1            1     2     2   44     NA   0      nu.5.g2  .p50.
## 51 51      x6 ~1            1     2     2   45     NA   0      nu.6.g2  .p51.
## 52 52      x7 ~1            1     2     2   46     NA   0      nu.7.g2  .p52.
## 53 53      x8 ~1            1     2     2   47     NA   0      nu.8.g2  .p53.
## 54 54      x9 ~1            1     2     2   48     NA   0      nu.9.g2  .p54.
## 55 55      x1 ~~      x1    1     2     2   49     NA   0 theta.1_1.g2  .p55.
## 56 56      x2 ~~      x2    1     2     2   50     NA   0 theta.2_2.g2  .p56.
## 57 57      x3 ~~      x3    1     2     2   51     NA   0 theta.3_3.g2  .p57.
## 58 58      x4 ~~      x4    1     2     2   52     NA   0 theta.4_4.g2  .p58.
## 59 59      x5 ~~      x5    1     2     2   53     NA   0 theta.5_5.g2  .p59.
## 60 60      x6 ~~      x6    1     2     2   54     NA   0 theta.6_6.g2  .p60.
## 61 61      x7 ~~      x7    1     2     2   55     NA   0 theta.7_7.g2  .p61.
## 62 62      x8 ~~      x8    1     2     2   56     NA   0 theta.8_8.g2  .p62.
## 63 63      x9 ~~      x9    1     2     2   57     NA   0 theta.9_9.g2  .p63.
## 64 64  visual ~1            1     2     2    0      0   0   alpha.1.g2  .p64.
## 65 65 textual ~1            1     2     2    0      0   0   alpha.2.g2  .p65.
## 66 66   speed ~1            1     2     2    0      0   0   alpha.3.g2  .p66.
## 67 67  visual ~~  visual    1     2     2   58     NA   0   psi.1_1.g2  .p67.
## 68 68 textual ~~ textual    1     2     2   59     NA   0   psi.2_2.g2  .p68.
## 69 69   speed ~~   speed    1     2     2   60     NA   0   psi.3_3.g2  .p69.
## 70 70  visual ~~ textual    1     2     2   61     NA   0   psi.2_1.g2  .p70.
## 71 71  visual ~~   speed    1     2     2   62     NA   0   psi.3_1.g2  .p71.
## 72 72 textual ~~   speed    1     2     2   63     NA   0   psi.3_2.g2  .p72.
## 73 73    .p1. ==   .p37.    2     0     0    0     NA   0
## 74 74    .p2. ==   .p38.    2     0     0    0     NA   0
## 75 75    .p3. ==   .p39.    2     0     0    0     NA   0
## 76 76    .p4. ==   .p40.    2     0     0    0     NA   0
## 77 77    .p5. ==   .p41.    2     0     0    0     NA   0
## 78 78    .p6. ==   .p42.    2     0     0    0     NA   0
## 79 79    .p7. ==   .p43.    2     0     0    0     NA   0
## 80 80    .p8. ==   .p44.    2     0     0    0     NA   0
## 81 81    .p9. ==   .p45.    2     0     0    0     NA   0
```

```
lavaanify(cfaModel_metricInvariance, ngroups = 2)
```

```
##    id     lhs op     rhs user block group free ustart exo    label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0 lambdax1   .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0 lambdax2   .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0 lambdax3   .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0 lambdax4   .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0 lambdax5   .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0 lambdax6   .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0 lambdax7   .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0 lambdax8   .p8.
```

```
## 9   9   speed =~      x9    1     1     1    9     NA   0 lambdax9   .p9.
## 10 10  visual ~1            1     1     1    0      0   0           .p10.
## 11 11 textual ~1            1     1     1    0      0   0           .p11.
## 12 12   speed ~1            1     1     1    0      0   0           .p12.
## 13 13  visual ~~  visual    1     1     1    0      1   0           .p13.
## 14 14 textual ~~ textual    1     1     1    0      1   0           .p14.
## 15 15   speed ~~   speed    1     1     1    0      1   0           .p15.
## 16 16  visual ~~ textual    1     1     1   10     NA   0           .p16.
## 17 17  visual ~~   speed    1     1     1   11     NA   0           .p17.
## 18 18 textual ~~   speed    1     1     1   12     NA   0           .p18.
## 19 19      x1 ~~      x1    1     1     1   13     NA   0           .p19.
## 20 20      x2 ~~      x2    1     1     1   14     NA   0           .p20.
## 21 21      x3 ~~      x3    1     1     1   15     NA   0           .p21.
## 22 22      x4 ~~      x4    1     1     1   16     NA   0           .p22.
## 23 23      x5 ~~      x5    1     1     1   17     NA   0           .p23.
## 24 24      x6 ~~      x6    1     1     1   18     NA   0           .p24.
## 25 25      x7 ~~      x7    1     1     1   19     NA   0           .p25.
## 26 26      x8 ~~      x8    1     1     1   20     NA   0           .p26.
## 27 27      x9 ~~      x9    1     1     1   21     NA   0           .p27.
## 28 28      x1 ~1            1     1     1   22     NA   0           .p28.
## 29 29      x2 ~1            1     1     1   23     NA   0           .p29.
## 30 30      x3 ~1            1     1     1   24     NA   0           .p30.
## 31 31      x4 ~1            1     1     1   25     NA   0           .p31.
## 32 32      x5 ~1            1     1     1   26     NA   0           .p32.
## 33 33      x6 ~1            1     1     1   27     NA   0           .p33.
## 34 34      x7 ~1            1     1     1   28     NA   0           .p34.
## 35 35      x8 ~1            1     1     1   29     NA   0           .p35.
## 36 36      x9 ~1            1     1     1   30     NA   0           .p36.
## 37 37  visual =~      x1    1     2     2   31     NA   0 lambdax1  .p37.
## 38 38  visual =~      x2    1     2     2   32     NA   0 lambdax2  .p38.
## 39 39  visual =~      x3    1     2     2   33     NA   0 lambdax3  .p39.
## 40 40 textual =~      x4    1     2     2   34     NA   0 lambdax4  .p40.
## 41 41 textual =~      x5    1     2     2   35     NA   0 lambdax5  .p41.
## 42 42 textual =~      x6    1     2     2   36     NA   0 lambdax6  .p42.
## 43 43   speed =~      x7    1     2     2   37     NA   0 lambdax7  .p43.
## 44 44   speed =~      x8    1     2     2   38     NA   0 lambdax8  .p44.
## 45 45   speed =~      x9    1     2     2   39     NA   0 lambdax9  .p45.
## 46 46  visual ~1            1     2     2    0      0   0           .p46.
## 47 47 textual ~1            1     2     2    0      0   0           .p47.
## 48 48   speed ~1            1     2     2    0      0   0           .p48.
## 49 49  visual ~~  visual    1     2     2   40     NA   0           .p49.
## 50 50 textual ~~ textual    1     2     2   41     NA   0           .p50.
## 51 51   speed ~~   speed    1     2     2   42     NA   0           .p51.
## 52 52  visual ~~ textual    1     2     2   43     NA   0           .p52.
## 53 53  visual ~~   speed    1     2     2   44     NA   0           .p53.
## 54 54 textual ~~   speed    1     2     2   45     NA   0           .p54.
```

```
## 55 55      x1 ~~      x1    1     2     2   46     NA   0          .p55.
## 56 56      x2 ~~      x2    1     2     2   47     NA   0          .p56.
## 57 57      x3 ~~      x3    1     2     2   48     NA   0          .p57.
## 58 58      x4 ~~      x4    1     2     2   49     NA   0          .p58.
## 59 59      x5 ~~      x5    1     2     2   50     NA   0          .p59.
## 60 60      x6 ~~      x6    1     2     2   51     NA   0          .p60.
## 61 61      x7 ~~      x7    1     2     2   52     NA   0          .p61.
## 62 62      x8 ~~      x8    1     2     2   53     NA   0          .p62.
## 63 63      x9 ~~      x9    1     2     2   54     NA   0          .p63.
## 64 64      x1 ~1            1     2     2   55     NA   0          .p64.
## 65 65      x2 ~1            1     2     2   56     NA   0          .p65.
## 66 66      x3 ~1            1     2     2   57     NA   0          .p66.
## 67 67      x4 ~1            1     2     2   58     NA   0          .p67.
## 68 68      x5 ~1            1     2     2   59     NA   0          .p68.
## 69 69      x6 ~1            1     2     2   60     NA   0          .p69.
## 70 70      x7 ~1            1     2     2   61     NA   0          .p70.
## 71 71      x8 ~1            1     2     2   62     NA   0          .p71.
## 72 72      x9 ~1            1     2     2   63     NA   0          .p72.
## 73 73    .p1. ==   .p37.    2     0     0    0     NA   0
## 74 74    .p2. ==   .p38.    2     0     0    0     NA   0
## 75 75    .p3. ==   .p39.    2     0     0    0     NA   0
## 76 76    .p4. ==   .p40.    2     0     0    0     NA   0
## 77 77    .p5. ==   .p41.    2     0     0    0     NA   0
## 78 78    .p6. ==   .p42.    2     0     0    0     NA   0
## 79 79    .p7. ==   .p43.    2     0     0    0     NA   0
## 80 80    .p8. ==   .p44.    2     0     0    0     NA   0
## 81 81    .p9. ==   .p45.    2     0     0    0     NA   0
```

14.5.5.3 Fit the Model

```
metricInvarianceModel_fit <- cfa(metricInvarianceSyntax,
                                 data = HolzingerSwineford1939,
                                 group = "school",
                                 missing = "ML",
                                 estimator = "MLR")
```

```
metricInvarianceModelFullSyntax_fit <- lavaan(cfaModel_metricInvariance,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")
```

14.5.5.4 Model Summary

```
summary(metricInvarianceModel_fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 81 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        63
##   Number of equality constraints                     9
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                                Standard      Robust
##   Test Statistic                                 81.456      86.907
##   Degrees of freedom                                 54          54
##   P-value (Chi-square)                            0.009       0.003
##   Scaling correction factor                                   0.937
##       Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                            43.580      46.496
##     1                                            37.876      40.411
##
## Model Test Baseline Model:
##
##   Test statistic                                609.050     638.762
##   Degrees of freedom                                 72          72
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.949       0.942
##   Tucker-Lewis Index (TLI)                        0.932       0.923
##
```

```
##   Robust Comparative Fit Index (CFI)                         0.943
##   Robust Tucker-Lewis Index (TLI)                            0.924
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2584.689   -2584.689
##   Scaling correction factor                                  0.867
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -2543.961   -2543.961
##   Scaling correction factor                                  0.975
##       for the MLR correction
##
##   Akaike (AIC)                                5277.379    5277.379
##   Bayesian (BIC)                              5468.395    5468.395
##   Sample-size adjusted Bayesian (BIC)         5297.203    5297.203
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.063       0.069
##   90 Percent confidence interval - lower         0.032       0.040
##   90 Percent confidence interval - upper         0.090       0.096
##   P-value RMSEA <= 0.05                          0.211       0.128
##
##   Robust RMSEA                                               0.067
##   90 Percent confidence interval - lower                     0.039
##   90 Percent confidence interval - upper                     0.092
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.075       0.075
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    1.093    0.151    7.242    0.000    1.093    0.888
##     x2      (l.2_)    0.483    0.121    4.008    0.000    0.483    0.381
##     x3      (l.3_)    0.710    0.102    6.970    0.000    0.710    0.609
```

```
##   textual =~
##     x4      (l.4_)    0.992    0.090   11.037    0.000    0.992    0.831
##     x5      (l.5_)    1.097    0.106   10.304    0.000    1.097    0.854
##     x6      (l.6_)    0.872    0.090    9.693    0.000    0.872    0.871
##   speed =~
##     x7      (l.7_)    0.602    0.085    7.084    0.000    0.602    0.569
##     x8      (l.8_)    0.671    0.090    7.490    0.000    0.671    0.674
##     x9      (l.9_)    0.587    0.108    5.441    0.000    0.587    0.585
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.545    0.083    6.536    0.000    0.545    0.545
##     speed  (p.3_1)    0.411    0.130    3.156    0.002    0.411    0.411
##   textual ~~
##     speed  (p.3_2)    0.347    0.097    3.560    0.000    0.347    0.347
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (n.1.)    4.922    0.113   43.524    0.000    4.922    3.999
##    .x2      (n.2.)    6.024    0.119   50.541    0.000    6.024    4.745
##    .x3      (n.3.)    2.525    0.106   23.818    0.000    2.525    2.165
##    .x4      (n.4.)    2.837    0.109   26.090    0.000    2.837    2.379
##    .x5      (n.5.)    3.871    0.119   32.489    0.000    3.871    3.012
##    .x6      (n.6.)    1.876    0.088   21.269    0.000    1.876    1.874
##    .x7      (n.7.)    4.442    0.096   46.284    0.000    4.442    4.199
##    .x8      (n.8.)    5.586    0.092   60.978    0.000    5.586    5.609
##    .x9      (n.9.)    5.394    0.096   56.397    0.000    5.394    5.374
##     visual  (a.1.)    0.000                                0.000    0.000
##     textual (a.2.)    0.000                                0.000    0.000
##     speed   (a.3.)    0.000                                0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.320    0.230    1.388    0.165    0.320    0.211
##    .x2      (t.2_)    1.378    0.219    6.299    0.000    1.378    0.855
##    .x3      (t.3_)    0.857    0.139    6.168    0.000    0.857    0.630
##    .x4      (t.4_)    0.439    0.086    5.087    0.000    0.439    0.309
##    .x5      (t.5_)    0.448    0.095    4.736    0.000    0.448    0.271
##    .x6      (t.6_)    0.241    0.063    3.857    0.000    0.241    0.241
##    .x7      (t.7_)    0.757    0.126    6.003    0.000    0.757    0.677
##    .x8      (t.8_)    0.542    0.101    5.360    0.000    0.542    0.546
##    .x9      (t.9_)    0.663    0.144    4.605    0.000    0.663    0.658
##     visual  (p.1_)    1.000                                1.000    1.000
##     textual (p.2_)    1.000                                1.000    1.000
##     speed   (p.3_)    1.000                                1.000    1.000
```

```
##
## R-Square:
##                    Estimate
##     x1                0.789
##     x2                0.145
##     x3                0.370
##     x4                0.691
##     x5                0.729
##     x6                0.759
##     x7                0.323
##     x8                0.454
##     x9                0.342
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    1.093    0.151    7.242    0.000    0.708    0.622
##     x2      (l.2_)    0.483    0.121    4.008    0.000    0.313    0.308
##     x3      (l.3_)    0.710    0.102    6.970    0.000    0.460    0.468
##   textual =~
##     x4      (l.4_)    0.992    0.090   11.037    0.000    0.998    0.869
##     x5      (l.5_)    1.097    0.106   10.304    0.000    1.105    0.890
##     x6      (l.6_)    0.872    0.090    9.693    0.000    0.878    0.812
##   speed =~
##     x7      (l.7_)    0.602    0.085    7.084    0.000    0.628    0.651
##     x8      (l.8_)    0.671    0.090    7.490    0.000    0.701    0.787
##     x9      (l.9_)    0.587    0.108    5.441    0.000    0.613    0.649
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.428    0.133    3.222    0.001    0.655    0.655
##     speed  (p.3_1)    0.343    0.152    2.264    0.024    0.507    0.507
##   textual ~~
##     speed  (p.3_2)    0.286    0.173    1.657    0.098    0.272    0.272
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (n.1.)    5.008    0.109   46.007    0.000    5.008    4.398
##    .x2      (n.2.)    6.263    0.104   60.369    0.000    6.263    6.150
##    .x3      (n.3.)    1.993    0.098   20.317    0.000    1.993    2.028
##    .x4      (n.4.)    3.280    0.108   30.417    0.000    3.280    2.857
##    .x5      (n.5.)    4.680    0.116   40.342    0.000    4.680    3.769
```

```
##    .x6      (n.6.)    2.505    0.107   23.426    0.000    2.505    2.317
##    .x7      (n.7.)    3.864    0.095   40.541    0.000    3.864    4.001
##    .x8      (n.8.)    5.452    0.085   63.893    0.000    5.452    6.121
##    .x9      (n.9.)    5.344    0.091   58.974    0.000    5.344    5.656
##     visual  (a.1.)    0.000                               0.000    0.000
##     textual (a.2.)    0.000                               0.000    0.000
##     speed   (a.3.)    0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.794    0.231    3.437    0.001    0.794    0.613
##    .x2      (t.2_)    0.939    0.139    6.776    0.000    0.939    0.905
##    .x3      (t.3_)    0.755    0.142    5.309    0.000    0.755    0.781
##    .x4      (t.4_)    0.322    0.079    4.059    0.000    0.322    0.244
##    .x5      (t.5_)    0.322    0.078    4.098    0.000    0.322    0.209
##    .x6      (t.6_)    0.397    0.082    4.839    0.000    0.397    0.340
##    .x7      (t.7_)    0.538    0.110    4.878    0.000    0.538    0.577
##    .x8      (t.8_)    0.302    0.110    2.740    0.006    0.302    0.380
##    .x9      (t.9_)    0.517    0.119    4.350    0.000    0.517    0.579
##     visual  (p.1_)    0.420    0.183    2.289    0.022    1.000    1.000
##     textual (p.2_)    1.013    0.225    4.513    0.000    1.000    1.000
##     speed   (p.3_)    1.091    0.332    3.288    0.001    1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.387
##     x2                0.095
##     x3                0.219
##     x4                0.756
##     x5                0.791
##     x6                0.660
##     x7                0.423
##     x8                0.620
##     x9                0.421
```

```
summary(metricInvarianceModelFullSyntax_fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 81 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        63
```

```
##   Number of equality constraints                     9
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                                Standard      Robust
##   Test Statistic                                 81.456      86.907
##   Degrees of freedom                                 54          54
##   P-value (Chi-square)                            0.009       0.003
##   Scaling correction factor                                   0.937
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                            43.580      46.496
##     1                                            37.876      40.411
##
## Model Test Baseline Model:
##
##   Test statistic                                609.050     638.762
##   Degrees of freedom                                 72          72
##   P-value                                         0.000       0.000
##   Scaling correction factor                                   0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.949       0.942
##   Tucker-Lewis Index (TLI)                        0.932       0.923
##
##   Robust Comparative Fit Index (CFI)                          0.943
##   Robust Tucker-Lewis Index (TLI)                             0.924
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2584.689   -2584.689
##   Scaling correction factor                                   0.867
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -2543.961   -2543.961
##   Scaling correction factor                                   0.975
##       for the MLR correction
##
##   Akaike (AIC)                                5277.379    5277.379
##   Bayesian (BIC)                              5468.395    5468.395
```

```
##   Sample-size adjusted Bayesian (BIC)         5297.203    5297.203
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.063       0.069
##   90 Percent confidence interval - lower         0.032       0.040
##   90 Percent confidence interval - upper         0.090       0.096
##   P-value RMSEA <= 0.05                          0.211       0.128
##
##   Robust RMSEA                                               0.067
##   90 Percent confidence interval - lower                     0.039
##   90 Percent confidence interval - upper                     0.092
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.075       0.075
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (lmb1)    1.093    0.151    7.242    0.000    1.093    0.888
##     x2      (lmb2)    0.483    0.121    4.008    0.000    0.483    0.381
##     x3      (lmb3)    0.710    0.102    6.970    0.000    0.710    0.609
##   textual =~
##     x4      (lmb4)    0.992    0.090   11.037    0.000    0.992    0.831
##     x5      (lmb5)    1.097    0.106   10.304    0.000    1.097    0.854
##     x6      (lmb6)    0.872    0.090    9.693    0.000    0.872    0.871
##   speed =~
##     x7      (lmb7)    0.602    0.085    7.084    0.000    0.602    0.569
##     x8      (lmb8)    0.671    0.090    7.490    0.000    0.671    0.674
##     x9      (lmb9)    0.587    0.108    5.441    0.000    0.587    0.585
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.545    0.083    6.536    0.000    0.545    0.545
##     speed             0.411    0.130    3.156    0.002    0.411    0.411
```

```
##   textual ~~
##     speed             0.347    0.097    3.560    0.000    0.347    0.347
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1                4.922    0.113   43.524    0.000    4.922    3.999
##    .x2                6.024    0.119   50.541    0.000    6.024    4.745
##    .x3                2.525    0.106   23.818    0.000    2.525    2.165
##    .x4                2.837    0.109   26.090    0.000    2.837    2.379
##    .x5                3.871    0.119   32.489    0.000    3.871    3.012
##    .x6                1.876    0.088   21.269    0.000    1.876    1.874
##    .x7                4.442    0.096   46.284    0.000    4.442    4.199
##    .x8                5.586    0.092   60.978    0.000    5.586    5.609
##    .x9                5.394    0.096   56.397    0.000    5.394    5.374
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1                0.320    0.230    1.388    0.165    0.320    0.211
##    .x2                1.378    0.219    6.299    0.000    1.378    0.855
##    .x3                0.857    0.139    6.168    0.000    0.857    0.630
##    .x4                0.439    0.086    5.087    0.000    0.439    0.309
##    .x5                0.448    0.095    4.736    0.000    0.448    0.271
##    .x6                0.241    0.063    3.857    0.000    0.241    0.241
##    .x7                0.757    0.126    6.003    0.000    0.757    0.677
##    .x8                0.542    0.101    5.360    0.000    0.542    0.546
##    .x9                0.663    0.144    4.605    0.000    0.663    0.658
##
## R-Square:
##                    Estimate
##     x1                0.789
##     x2                0.145
##     x3                0.370
##     x4                0.691
##     x5                0.729
##     x6                0.759
##     x7                0.323
##     x8                0.454
##     x9                0.342
##
##
```

```
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (lmb1)    1.093    0.151    7.242    0.000    0.708    0.622
##     x2      (lmb2)    0.483    0.121    4.008    0.000    0.313    0.308
##     x3      (lmb3)    0.710    0.102    6.970    0.000    0.460    0.468
##   textual =~
##     x4      (lmb4)    0.992    0.090   11.037    0.000    0.998    0.869
##     x5      (lmb5)    1.097    0.106   10.304    0.000    1.105    0.890
##     x6      (lmb6)    0.872    0.090    9.693    0.000    0.878    0.812
##   speed =~
##     x7      (lmb7)    0.602    0.085    7.084    0.000    0.628    0.651
##     x8      (lmb8)    0.671    0.090    7.490    0.000    0.701    0.787
##     x9      (lmb9)    0.587    0.108    5.441    0.000    0.613    0.649
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.428    0.133    3.222    0.001    0.655    0.655
##     speed             0.343    0.152    2.264    0.024    0.507    0.507
##   textual ~~
##     speed             0.286    0.173    1.657    0.098    0.272    0.272
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1                5.008    0.109   46.007    0.000    5.008    4.398
##    .x2                6.263    0.104   60.369    0.000    6.263    6.150
##    .x3                1.993    0.098   20.317    0.000    1.993    2.028
##    .x4                3.280    0.108   30.417    0.000    3.280    2.857
##    .x5                4.680    0.116   40.342    0.000    4.680    3.769
##    .x6                2.505    0.107   23.426    0.000    2.505    2.317
##    .x7                3.864    0.095   40.541    0.000    3.864    4.001
##    .x8                5.452    0.085   63.893    0.000    5.452    6.121
##    .x9                5.344    0.091   58.974    0.000    5.344    5.656
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.420    0.183    2.289    0.022    1.000    1.000
##     textual           1.013    0.225    4.513    0.000    1.000    1.000
##     speed             1.091    0.332    3.288    0.001    1.000    1.000
##    .x1                0.794    0.231    3.437    0.001    0.794    0.613
```

```
##    .x2                0.939    0.139    6.776    0.000    0.939    0.905
##    .x3                0.755    0.142    5.309    0.000    0.755    0.781
##    .x4                0.322    0.079    4.059    0.000    0.322    0.244
##    .x5                0.322    0.078    4.098    0.000    0.322    0.209
##    .x6                0.397    0.082    4.839    0.000    0.397    0.340
##    .x7                0.538    0.110    4.878    0.000    0.538    0.577
##    .x8                0.302    0.110    2.740    0.006    0.302    0.380
##    .x9                0.517    0.119    4.350    0.000    0.517    0.579
##
## R-Square:
##                    Estimate
##     x1                0.387
##     x2                0.095
##     x3                0.219
##     x4                0.756
##     x5                0.791
##     x6                0.660
##     x7                0.423
##     x8                0.620
##     x9                0.421
```

14.5.5.5 Model Fit

You can specify the null model as the baseline model using: `baseline.model = nullModelFit`

```
fitMeasures(metricInvarianceModel_fit)
```

```
##                          npar                          fmin
##                        54.000                         0.160
##                         chisq                            df
##                        81.456                        54.000
##                        pvalue                   chisq.scaled
##                         0.009                        86.907
##                     df.scaled                  pvalue.scaled
##                        54.000                         0.003
##          chisq.scaling.factor                 baseline.chisq
##                         0.937                        609.050
##                   baseline.df                baseline.pvalue
##                        72.000                         0.000
##         baseline.chisq.scaled             baseline.df.scaled
##                       638.762                        72.000
##        baseline.pvalue.scaled baseline.chisq.scaling.factor
##                         0.000                         0.953
##                           cfi                           tli
```

```
##                       0.949                       0.932
##                        nnfi                         rfi
##                       0.932                       0.822
##                         nfi                        pnfi
##                       0.866                       0.650
##                         ifi                         rni
##                       0.951                       0.949
##                  cfi.scaled                  tli.scaled
##                       0.942                       0.923
##                  cfi.robust                  tli.robust
##                       0.943                       0.924
##                 nnfi.scaled                 nnfi.robust
##                       0.923                       0.924
##                  rfi.scaled                  nfi.scaled
##                       0.819                       0.864
##                  ifi.scaled                  rni.scaled
##                       0.944                       0.942
##                  rni.robust                        logl
##                       0.943                   -2584.689
##           unrestricted.logl                         aic
##                   -2543.961                    5277.379
##                         bic                      ntotal
##                    5468.395                     254.000
##                        bic2           scaling.factor.h1
##                    5297.203                       0.975
##           scaling.factor.h0                       rmsea
##                       0.867                       0.063
##              rmsea.ci.lower              rmsea.ci.upper
##                       0.032                       0.090
##                 rmsea.pvalue                 rmsea.scaled
##                       0.211                       0.069
##       rmsea.ci.lower.scaled       rmsea.ci.upper.scaled
##                       0.040                       0.096
##         rmsea.pvalue.scaled                 rmsea.robust
##                       0.128                       0.067
##       rmsea.ci.lower.robust       rmsea.ci.upper.robust
##                       0.039                       0.092
##         rmsea.pvalue.robust                         rmr
##                         NA                       0.090
##                  rmr_nomean                        srmr
##                       0.099                       0.075
##                srmr_bentler          srmr_bentler_nomean
##                       0.075                       0.082
##                        crmr                 crmr_nomean
##                       0.085                       0.088
##                  srmr_mplus           srmr_mplus_nomean
```

```
##                          0.080                          0.080
##                          cn_05                          cn_01
##                        225.992                        253.793
##                            gfi                           agfi
##                          0.993                          0.987
##                           pgfi                            mfi
##                          0.497                          0.947
```

```
metricInvarianceModelFitIndices <- fitMeasures(metricInvarianceModel_fit)[c("cfi.robust", "rmsea.
```

14.5.5.6 Compare Model Fit

14.5.5.6.1 Nested Model (χ^2) Difference Test The configural invariance model and the metric ("weak factorial") invariance model are considered "nested" models. The metric invariance model is nested within the configural invariance model because the configural invariance model includes all of the terms of the metric invariance model along with additional terms. Model fit of nested models can be compared with a chi-square difference test, also known as a likelihood ratio test or deviance test. A significant chi-square difference test would indicate that the simplified model with additional constraints (the metric invariance model) is significantly worse fitting than the more complex model that has fewer constraints (the configural invariance model).

```
anova(configuralInvarianceModel_fit, metricInvarianceModel_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                              Df    AIC    BIC  Chisq Chisq diff Df diff
## configuralInvarianceModel_fit 48 5281.2 5493.5 73.311
## metricInvarianceModel_fit     54 5277.4 5468.4 81.456      6.123       6
##                              Pr(>Chisq)
## configuralInvarianceModel_fit
## metricInvarianceModel_fit         0.4096
```

The metric invariance model did not fit significantly worse than the configural invariance model, so metric invariance held. This provides evidence of measurement invariance of factor loadings across groups. Measurement invariance of factor loadings across groups provides support for examining

whether the groups show different associations of the factor with other constructs [Little et al., 2007].

```
round(metricInvarianceModelFitIndices - configuralInvarianceModelFitIndices, digits = 3
```

14.5.5.6.2 Compare Other Fit Criteria

```
##   cfi.robust rmsea.robust         srmr
##        0.000       -0.004        0.006
```

14.5.5.6.3 Permutation Test Permutation procedures for testing measurement invariance are described in [Jorgensen et al., 2018].

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed. You can specify the null model as the baseline model using: `baseline.model = nullModelFit`. **Warning**: this code takes a while to run based on 1000 iterations. You can reduce the number of iterations to be faster.

```
metricInvarianceTest <- permuteMeasEq(nPermute = numPermutations,
                                      modelType = "mgcfa",
                                      con = metricInvarianceModel_fit,
                                      uncon = configuralInvarianceModel_fit,
                                      AFIs = myAFIs,
                                      moreAFIs = moreAFIs,
                                      iseed = 52242)
```

```
metricInvarianceTest
```

```
## Omnibus p value based on parametric chi-squared difference test:
##
## Chisq diff    Df diff Pr(>Chisq)
##      6.123      6.000      0.410
##
##
## Omnibus p values based on nonparametric permutation method:
##
##              AFI.Difference p.value
## chisq                 8.145   0.369
## chisq.scaled          4.364   0.490
## rmsea                -0.001   0.316
## cfi                  -0.004   0.362
```

```
## tli                   0.003   0.310
## srmr                  0.006   0.370
## rmsea.robust         -0.004   0.412
## cfi.robust            0.000   0.454
## tli.robust            0.009   0.385
```

14.5.5.7 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(metricInvarianceModel_fit)
```

```
## $`2`
##           visual   textual     speed
## alpha  0.6524461 0.8841445 0.6430042
## omega  0.6717417 0.8859944 0.6378981
## omega2 0.6717417 0.8859944 0.6378981
## omega3 0.6686532 0.8686982 0.6304521
## avevar 0.4306932 0.7232192 0.3707856
##
## $`1`
##           visual   textual     speed
## alpha  0.4648456 0.8955126 0.7327296
## omega  0.4687754 0.8951470 0.7355618
## omega2 0.4687754 0.8951470 0.7355618
## omega3 0.4587684 0.8856268 0.7360761
## avevar 0.2459821 0.7416289 0.4819893
```

14.5.5.8 Path Diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(metricInvarianceModel_fit,
         what = "est",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         ask = FALSE,
         edge.label.cex = 1.2)
```

Figure 14.34: Metric invariance model in confirmatory factor analysis

Figure 14.35: Metric invariance model in confirmatory factor analysis

14.5.6 Scalar ("Strong Factorial") Invariance Model

Specify invariance of factor loadings and intercepts across groups.

14.5.6.1 Model Syntax

```
scalarInvarianceModel <- measEq.syntax(configural.model = cfaModel,
                                       data = HolzingerSwineford1939,
                                       ID.fac = "std.lv",
                                       group = "school",
                                       group.equal = c("loadings","intercepts"))
```

```
cat(as.character(scalarInvarianceModel))
```

```
## ## LOADINGS:
##
## visual =~ c(NA, NA)*x1 + c(lambda.1_1, lambda.1_1)*x1
## visual =~ c(NA, NA)*x2 + c(lambda.2_1, lambda.2_1)*x2
## visual =~ c(NA, NA)*x3 + c(lambda.3_1, lambda.3_1)*x3
## textual =~ c(NA, NA)*x4 + c(lambda.4_2, lambda.4_2)*x4
## textual =~ c(NA, NA)*x5 + c(lambda.5_2, lambda.5_2)*x5
## textual =~ c(NA, NA)*x6 + c(lambda.6_2, lambda.6_2)*x6
## speed =~ c(NA, NA)*x7 + c(lambda.7_3, lambda.7_3)*x7
## speed =~ c(NA, NA)*x8 + c(lambda.8_3, lambda.8_3)*x8
## speed =~ c(NA, NA)*x9 + c(lambda.9_3, lambda.9_3)*x9
##
## ## INTERCEPTS:
##
## x1 ~ c(NA, NA)*1 + c(nu.1, nu.1)*1
## x2 ~ c(NA, NA)*1 + c(nu.2, nu.2)*1
## x3 ~ c(NA, NA)*1 + c(nu.3, nu.3)*1
## x4 ~ c(NA, NA)*1 + c(nu.4, nu.4)*1
## x5 ~ c(NA, NA)*1 + c(nu.5, nu.5)*1
## x6 ~ c(NA, NA)*1 + c(nu.6, nu.6)*1
## x7 ~ c(NA, NA)*1 + c(nu.7, nu.7)*1
## x8 ~ c(NA, NA)*1 + c(nu.8, nu.8)*1
## x9 ~ c(NA, NA)*1 + c(nu.9, nu.9)*1
##
## ## UNIQUE-FACTOR VARIANCES:
##
## x1 ~~ c(NA, NA)*x1 + c(theta.1_1.g1, theta.1_1.g2)*x1
## x2 ~~ c(NA, NA)*x2 + c(theta.2_2.g1, theta.2_2.g2)*x2
## x3 ~~ c(NA, NA)*x3 + c(theta.3_3.g1, theta.3_3.g2)*x3
```

```
## x4 ~~ c(NA, NA)*x4 + c(theta.4_4.g1, theta.4_4.g2)*x4
## x5 ~~ c(NA, NA)*x5 + c(theta.5_5.g1, theta.5_5.g2)*x5
## x6 ~~ c(NA, NA)*x6 + c(theta.6_6.g1, theta.6_6.g2)*x6
## x7 ~~ c(NA, NA)*x7 + c(theta.7_7.g1, theta.7_7.g2)*x7
## x8 ~~ c(NA, NA)*x8 + c(theta.8_8.g1, theta.8_8.g2)*x8
## x9 ~~ c(NA, NA)*x9 + c(theta.9_9.g1, theta.9_9.g2)*x9
##
## ## LATENT MEANS/INTERCEPTS:
##
## visual ~ c(0, NA)*1 + c(alpha.1.g1, alpha.1.g2)*1
## textual ~ c(0, NA)*1 + c(alpha.2.g1, alpha.2.g2)*1
## speed ~ c(0, NA)*1 + c(alpha.3.g1, alpha.3.g2)*1
##
## ## COMMON-FACTOR VARIANCES:
##
## visual ~~ c(1, NA)*visual + c(psi.1_1.g1, psi.1_1.g2)*visual
## textual ~~ c(1, NA)*textual + c(psi.2_2.g1, psi.2_2.g2)*textual
## speed ~~ c(1, NA)*speed + c(psi.3_3.g1, psi.3_3.g2)*speed
##
## ## COMMON-FACTOR COVARIANCES:
##
## visual ~~ c(NA, NA)*textual + c(psi.2_1.g1, psi.2_1.g2)*textual
## visual ~~ c(NA, NA)*speed + c(psi.3_1.g1, psi.3_1.g2)*speed
## textual ~~ c(NA, NA)*speed + c(psi.3_2.g1, psi.3_2.g2)*speed
```

```
scalarInvarianceSyntax <- as.character(scalarInvarianceModel)
```

```
summary(scalarInvarianceModel)
```

14.5.6.1.1 Summary of Model Features

```
## This lavaan model syntax specifies a CFA with 9 manifest indicators of 3 common factor(s).
##
## To identify the location and scale of each common factor, the factor means and variances were
##
## Pattern matrix indicating num(eric), ord(ered), and lat(ent) indicators per factor:
##
##    visual textual speed
## x1 num
## x2 num
## x3 num
## x4        num
```

```
## x5        num
## x6        num
## x7                num
## x8                num
## x9                num
##
## The following types of parameter were constrained to equality across groups:
##
##  loadings
##  intercepts
```

14.5.6.2 Model Syntax in Table Form

```
lavaanify(scalarInvarianceModel, ngroups = 2)
```

```
##    id     lhs op     rhs user block group free ustart exo        label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0   lambda.1_1   .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0   lambda.2_1   .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0   lambda.3_1   .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0   lambda.4_2   .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0   lambda.5_2   .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0   lambda.6_2   .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0   lambda.7_3   .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0   lambda.8_3   .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0   lambda.9_3   .p9.
## 10 10      x1 ~1            1     1     1   10     NA   0         nu.1  .p10.
## 11 11      x2 ~1            1     1     1   11     NA   0         nu.2  .p11.
## 12 12      x3 ~1            1     1     1   12     NA   0         nu.3  .p12.
## 13 13      x4 ~1            1     1     1   13     NA   0         nu.4  .p13.
## 14 14      x5 ~1            1     1     1   14     NA   0         nu.5  .p14.
## 15 15      x6 ~1            1     1     1   15     NA   0         nu.6  .p15.
## 16 16      x7 ~1            1     1     1   16     NA   0         nu.7  .p16.
## 17 17      x8 ~1            1     1     1   17     NA   0         nu.8  .p17.
## 18 18      x9 ~1            1     1     1   18     NA   0         nu.9  .p18.
## 19 19      x1 ~~      x1    1     1     1   19     NA   0 theta.1_1.g1  .p19.
## 20 20      x2 ~~      x2    1     1     1   20     NA   0 theta.2_2.g1  .p20.
## 21 21      x3 ~~      x3    1     1     1   21     NA   0 theta.3_3.g1  .p21.
## 22 22      x4 ~~      x4    1     1     1   22     NA   0 theta.4_4.g1  .p22.
## 23 23      x5 ~~      x5    1     1     1   23     NA   0 theta.5_5.g1  .p23.
## 24 24      x6 ~~      x6    1     1     1   24     NA   0 theta.6_6.g1  .p24.
## 25 25      x7 ~~      x7    1     1     1   25     NA   0 theta.7_7.g1  .p25.
## 26 26      x8 ~~      x8    1     1     1   26     NA   0 theta.8_8.g1  .p26.
## 27 27      x9 ~~      x9    1     1     1   27     NA   0 theta.9_9.g1  .p27.
```

```
## 28 28  visual ~1            1     1     1    0      0   0   alpha.1.g1  .p28.
## 29 29 textual ~1            1     1     1    0      0   0   alpha.2.g1  .p29.
## 30 30   speed ~1            1     1     1    0      0   0   alpha.3.g1  .p30.
## 31 31  visual ~~  visual    1     1     1    0      1   0   psi.1_1.g1  .p31.
## 32 32 textual ~~ textual    1     1     1    0      1   0   psi.2_2.g1  .p32.
## 33 33   speed ~~   speed    1     1     1    0      1   0   psi.3_3.g1  .p33.
## 34 34  visual ~~ textual    1     1     1   28     NA   0   psi.2_1.g1  .p34.
## 35 35  visual ~~   speed    1     1     1   29     NA   0   psi.3_1.g1  .p35.
## 36 36 textual ~~   speed    1     1     1   30     NA   0   psi.3_2.g1  .p36.
## 37 37  visual =~      x1    1     2     2   31     NA   0   lambda.1_1  .p37.
## 38 38  visual =~      x2    1     2     2   32     NA   0   lambda.2_1  .p38.
## 39 39  visual =~      x3    1     2     2   33     NA   0   lambda.3_1  .p39.
## 40 40 textual =~      x4    1     2     2   34     NA   0   lambda.4_2  .p40.
## 41 41 textual =~      x5    1     2     2   35     NA   0   lambda.5_2  .p41.
## 42 42 textual =~      x6    1     2     2   36     NA   0   lambda.6_2  .p42.
## 43 43   speed =~      x7    1     2     2   37     NA   0   lambda.7_3  .p43.
## 44 44   speed =~      x8    1     2     2   38     NA   0   lambda.8_3  .p44.
## 45 45   speed =~      x9    1     2     2   39     NA   0   lambda.9_3  .p45.
## 46 46      x1 ~1            1     2     2   40     NA   0         nu.1  .p46.
## 47 47      x2 ~1            1     2     2   41     NA   0         nu.2  .p47.
## 48 48      x3 ~1            1     2     2   42     NA   0         nu.3  .p48.
## 49 49      x4 ~1            1     2     2   43     NA   0         nu.4  .p49.
## 50 50      x5 ~1            1     2     2   44     NA   0         nu.5  .p50.
## 51 51      x6 ~1            1     2     2   45     NA   0         nu.6  .p51.
## 52 52      x7 ~1            1     2     2   46     NA   0         nu.7  .p52.
## 53 53      x8 ~1            1     2     2   47     NA   0         nu.8  .p53.
## 54 54      x9 ~1            1     2     2   48     NA   0         nu.9  .p54.
## 55 55      x1 ~~      x1    1     2     2   49     NA   0 theta.1_1.g2  .p55.
## 56 56      x2 ~~      x2    1     2     2   50     NA   0 theta.2_2.g2  .p56.
## 57 57      x3 ~~      x3    1     2     2   51     NA   0 theta.3_3.g2  .p57.
## 58 58      x4 ~~      x4    1     2     2   52     NA   0 theta.4_4.g2  .p58.
## 59 59      x5 ~~      x5    1     2     2   53     NA   0 theta.5_5.g2  .p59.
## 60 60      x6 ~~      x6    1     2     2   54     NA   0 theta.6_6.g2  .p60.
## 61 61      x7 ~~      x7    1     2     2   55     NA   0 theta.7_7.g2  .p61.
## 62 62      x8 ~~      x8    1     2     2   56     NA   0 theta.8_8.g2  .p62.
## 63 63      x9 ~~      x9    1     2     2   57     NA   0 theta.9_9.g2  .p63.
## 64 64  visual ~1            1     2     2   58     NA   0   alpha.1.g2  .p64.
## 65 65 textual ~1            1     2     2   59     NA   0   alpha.2.g2  .p65.
## 66 66   speed ~1            1     2     2   60     NA   0   alpha.3.g2  .p66.
## 67 67  visual ~~  visual    1     2     2   61     NA   0   psi.1_1.g2  .p67.
## 68 68 textual ~~ textual    1     2     2   62     NA   0   psi.2_2.g2  .p68.
## 69 69   speed ~~   speed    1     2     2   63     NA   0   psi.3_3.g2  .p69.
## 70 70  visual ~~ textual    1     2     2   64     NA   0   psi.2_1.g2  .p70.
## 71 71  visual ~~   speed    1     2     2   65     NA   0   psi.3_1.g2  .p71.
## 72 72 textual ~~   speed    1     2     2   66     NA   0   psi.3_2.g2  .p72.
## 73 73    .p1. ==   .p37.    2     0     0    0     NA   0
```

```
## 74 74    .p2. ==   .p38.    2     0     0    0     NA   0
## 75 75    .p3. ==   .p39.    2     0     0    0     NA   0
## 76 76    .p4. ==   .p40.    2     0     0    0     NA   0
## 77 77    .p5. ==   .p41.    2     0     0    0     NA   0
## 78 78    .p6. ==   .p42.    2     0     0    0     NA   0
## 79 79    .p7. ==   .p43.    2     0     0    0     NA   0
## 80 80    .p8. ==   .p44.    2     0     0    0     NA   0
## 81 81    .p9. ==   .p45.    2     0     0    0     NA   0
## 82 82   .p10. ==   .p46.    2     0     0    0     NA   0
## 83 83   .p11. ==   .p47.    2     0     0    0     NA   0
## 84 84   .p12. ==   .p48.    2     0     0    0     NA   0
## 85 85   .p13. ==   .p49.    2     0     0    0     NA   0
## 86 86   .p14. ==   .p50.    2     0     0    0     NA   0
## 87 87   .p15. ==   .p51.    2     0     0    0     NA   0
## 88 88   .p16. ==   .p52.    2     0     0    0     NA   0
## 89 89   .p17. ==   .p53.    2     0     0    0     NA   0
## 90 90   .p18. ==   .p54.    2     0     0    0     NA   0
```

```r
lavaanify(cfaModel_scalarInvariance, ngroups = 2)
```

```
##    id     lhs op     rhs user block group free ustart exo    label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0 lambdax1   .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0 lambdax2   .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0 lambdax3   .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0 lambdax4   .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0 lambdax5   .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0 lambdax6   .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0 lambdax7   .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0 lambdax8   .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0 lambdax9   .p9.
## 10 10  visual ~1            1     1     1    0      0   0           .p10.
## 11 11 textual ~1            1     1     1    0      0   0           .p11.
## 12 12   speed ~1            1     1     1    0      0   0           .p12.
## 13 13  visual ~~  visual    1     1     1    0      1   0           .p13.
## 14 14 textual ~~ textual    1     1     1    0      1   0           .p14.
## 15 15   speed ~~   speed    1     1     1    0      1   0           .p15.
## 16 16  visual ~~ textual    1     1     1   10     NA   0           .p16.
## 17 17  visual ~~   speed    1     1     1   11     NA   0           .p17.
## 18 18 textual ~~   speed    1     1     1   12     NA   0           .p18.
## 19 19      x1 ~~      x1    1     1     1   13     NA   0           .p19.
## 20 20      x2 ~~      x2    1     1     1   14     NA   0           .p20.
## 21 21      x3 ~~      x3    1     1     1   15     NA   0           .p21.
## 22 22      x4 ~~      x4    1     1     1   16     NA   0           .p22.
## 23 23      x5 ~~      x5    1     1     1   17     NA   0           .p23.
## 24 24      x6 ~~      x6    1     1     1   18     NA   0           .p24.
```

```
## 25 25      x7 ~~      x7    1     1     1   19     NA   0           .p25.
## 26 26      x8 ~~      x8    1     1     1   20     NA   0           .p26.
## 27 27      x9 ~~      x9    1     1     1   21     NA   0           .p27.
## 28 28      x1 ~1            1     1     1   22     NA   0    intx1  .p28.
## 29 29      x2 ~1            1     1     1   23     NA   0    intx2  .p29.
## 30 30      x3 ~1            1     1     1   24     NA   0    intx3  .p30.
## 31 31      x4 ~1            1     1     1   25     NA   0    intx4  .p31.
## 32 32      x5 ~1            1     1     1   26     NA   0    intx5  .p32.
## 33 33      x6 ~1            1     1     1   27     NA   0    intx6  .p33.
## 34 34      x7 ~1            1     1     1   28     NA   0    intx7  .p34.
## 35 35      x8 ~1            1     1     1   29     NA   0    intx8  .p35.
## 36 36      x9 ~1            1     1     1   30     NA   0    intx9  .p36.
## 37 37  visual =~      x1    1     2     2   31     NA   0 lambdax1  .p37.
## 38 38  visual =~      x2    1     2     2   32     NA   0 lambdax2  .p38.
## 39 39  visual =~      x3    1     2     2   33     NA   0 lambdax3  .p39.
## 40 40 textual =~      x4    1     2     2   34     NA   0 lambdax4  .p40.
## 41 41 textual =~      x5    1     2     2   35     NA   0 lambdax5  .p41.
## 42 42 textual =~      x6    1     2     2   36     NA   0 lambdax6  .p42.
## 43 43   speed =~      x7    1     2     2   37     NA   0 lambdax7  .p43.
## 44 44   speed =~      x8    1     2     2   38     NA   0 lambdax8  .p44.
## 45 45   speed =~      x9    1     2     2   39     NA   0 lambdax9  .p45.
## 46 46  visual ~1            1     2     2   40     NA   0           .p46.
## 47 47 textual ~1            1     2     2   41     NA   0           .p47.
## 48 48   speed ~1            1     2     2   42     NA   0           .p48.
## 49 49  visual ~~  visual    1     2     2   43     NA   0           .p49.
## 50 50 textual ~~ textual    1     2     2   44     NA   0           .p50.
## 51 51   speed ~~   speed    1     2     2   45     NA   0           .p51.
## 52 52  visual ~~ textual    1     2     2   46     NA   0           .p52.
## 53 53  visual ~~   speed    1     2     2   47     NA   0           .p53.
## 54 54 textual ~~   speed    1     2     2   48     NA   0           .p54.
## 55 55      x1 ~~      x1    1     2     2   49     NA   0           .p55.
## 56 56      x2 ~~      x2    1     2     2   50     NA   0           .p56.
## 57 57      x3 ~~      x3    1     2     2   51     NA   0           .p57.
## 58 58      x4 ~~      x4    1     2     2   52     NA   0           .p58.
## 59 59      x5 ~~      x5    1     2     2   53     NA   0           .p59.
## 60 60      x6 ~~      x6    1     2     2   54     NA   0           .p60.
## 61 61      x7 ~~      x7    1     2     2   55     NA   0           .p61.
## 62 62      x8 ~~      x8    1     2     2   56     NA   0           .p62.
## 63 63      x9 ~~      x9    1     2     2   57     NA   0           .p63.
## 64 64      x1 ~1            1     2     2   58     NA   0    intx1  .p64.
## 65 65      x2 ~1            1     2     2   59     NA   0    intx2  .p65.
## 66 66      x3 ~1            1     2     2   60     NA   0    intx3  .p66.
## 67 67      x4 ~1            1     2     2   61     NA   0    intx4  .p67.
## 68 68      x5 ~1            1     2     2   62     NA   0    intx5  .p68.
## 69 69      x6 ~1            1     2     2   63     NA   0    intx6  .p69.
## 70 70      x7 ~1            1     2     2   64     NA   0    intx7  .p70.
```

```
## 71 71      x8 ~1            1     2     2   65     NA   0    intx8  .p71.
## 72 72      x9 ~1            1     2     2   66     NA   0    intx9  .p72.
## 73 73    .p1. ==   .p37.    2     0     0    0     NA   0
## 74 74    .p2. ==   .p38.    2     0     0    0     NA   0
## 75 75    .p3. ==   .p39.    2     0     0    0     NA   0
## 76 76    .p4. ==   .p40.    2     0     0    0     NA   0
## 77 77    .p5. ==   .p41.    2     0     0    0     NA   0
## 78 78    .p6. ==   .p42.    2     0     0    0     NA   0
## 79 79    .p7. ==   .p43.    2     0     0    0     NA   0
## 80 80    .p8. ==   .p44.    2     0     0    0     NA   0
## 81 81    .p9. ==   .p45.    2     0     0    0     NA   0
## 82 82   .p28. ==   .p64.    2     0     0    0     NA   0
## 83 83   .p29. ==   .p65.    2     0     0    0     NA   0
## 84 84   .p30. ==   .p66.    2     0     0    0     NA   0
## 85 85   .p31. ==   .p67.    2     0     0    0     NA   0
## 86 86   .p32. ==   .p68.    2     0     0    0     NA   0
## 87 87   .p33. ==   .p69.    2     0     0    0     NA   0
## 88 88   .p34. ==   .p70.    2     0     0    0     NA   0
## 89 89   .p35. ==   .p71.    2     0     0    0     NA   0
## 90 90   .p36. ==   .p72.    2     0     0    0     NA   0
```

14.5.6.3 Fit the Model

```
scalarInvarianceModel_fit <- cfa(scalarInvarianceSyntax,
                                 data = HolzingerSwineford1939,
                                 group = "school",
                                 missing = "ML",
                                 estimator = "MLR")
```

```
scalarInvarianceModelFullSyntax_fit <- lavaan(cfaModel_scalarInvariance,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")
```

14.5.6.4 Model Summary

```
summary(scalarInvarianceModel_fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 71 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        66
##   Number of equality constraints                    18
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               119.966     127.299
##   Degrees of freedom                                60          60
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.942
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                           63.397      67.272
##     1                                           56.569      60.027
##
## Model Test Baseline Model:
##
##   Test statistic                              609.050     638.762
##   Degrees of freedom                               72          72
##   P-value                                       0.000       0.000
##   Scaling correction factor                                 0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                   0.888       0.881
##   Tucker-Lewis Index (TLI)                      0.866       0.858
##
##   Robust Comparative Fit Index (CFI)                        0.883
##   Robust Tucker-Lewis Index (TLI)                           0.859
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2603.944   -2603.944
##   Scaling correction factor                                 0.738
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -2543.961   -2543.961
```

```
##   Scaling correction factor                                  0.975
##       for the MLR correction
##
##   Akaike (AIC)                                5303.889    5303.889
##   Bayesian (BIC)                              5473.681    5473.681
##   Sample-size adjusted Bayesian (BIC)         5321.511    5321.511
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.089       0.094
##   90 Percent confidence interval - lower         0.065       0.071
##   90 Percent confidence interval - upper         0.112       0.117
##   P-value RMSEA <= 0.05                          0.005       0.002
##
##   Robust RMSEA                                               0.091
##   90 Percent confidence interval - lower                     0.069
##   90 Percent confidence interval - upper                     0.113
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.090       0.090
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    1.120    0.173    6.468    0.000    1.120    0.908
##     x2      (l.2_)    0.464    0.114    4.083    0.000    0.464    0.363
##     x3      (l.3_)    0.671    0.109    6.179    0.000    0.671    0.567
##   textual =~
##     x4      (l.4_)    0.948    0.094   10.083    0.000    0.948    0.808
##     x5      (l.5_)    1.123    0.113    9.930    0.000    1.123    0.862
##     x6      (l.6_)    0.885    0.087   10.181    0.000    0.885    0.879
##   speed =~
##     x7      (l.7_)    0.645    0.094    6.897    0.000    0.645    0.594
##     x8      (l.8_)    0.641    0.090    7.092    0.000    0.641    0.648
##     x9      (l.9_)    0.547    0.100    5.489    0.000    0.547    0.546
##
```

```
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.543    0.081    6.709    0.000    0.543    0.543
##     speed  (p.3_1)    0.391    0.132    2.962    0.003    0.391    0.391
##   textual ~~
##     speed  (p.3_2)    0.351    0.100    3.517    0.000    0.351    0.351
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (nu.1)    4.964    0.123   40.499    0.000    4.964    4.024
##    .x2      (nu.2)    6.178    0.094   66.009    0.000    6.178    4.832
##    .x3      (nu.3)    2.277    0.103   22.077    0.000    2.277    1.924
##    .x4      (nu.4)    2.744    0.098   27.920    0.000    2.744    2.338
##    .x5      (nu.5)    3.918    0.118   33.149    0.000    3.918    3.008
##    .x6      (nu.6)    1.899    0.086   22.155    0.000    1.899    1.885
##    .x7      (nu.7)    4.270    0.100   42.508    0.000    4.270    3.932
##    .x8      (nu.8)    5.659    0.090   62.668    0.000    5.659    5.718
##    .x9      (nu.9)    5.484    0.085   64.454    0.000    5.484    5.480
##     visual  (a.1.)    0.000                                0.000    0.000
##     textual (a.2.)    0.000                                0.000    0.000
##     speed   (a.3.)    0.000                                0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.266    0.274    0.973    0.330    0.266    0.175
##    .x2      (t.2_)    1.420    0.211    6.743    0.000    1.420    0.869
##    .x3      (t.3_)    0.950    0.157    6.051    0.000    0.950    0.678
##    .x4      (t.4_)    0.479    0.096    4.969    0.000    0.479    0.347
##    .x5      (t.5_)    0.436    0.101    4.338    0.000    0.436    0.257
##    .x6      (t.6_)    0.231    0.065    3.569    0.000    0.231    0.228
##    .x7      (t.7_)    0.763    0.152    5.015    0.000    0.763    0.647
##    .x8      (t.8_)    0.569    0.100    5.708    0.000    0.569    0.581
##    .x9      (t.9_)    0.703    0.139    5.047    0.000    0.703    0.701
##     visual  (p.1_)    1.000                                1.000    1.000
##     textual (p.2_)    1.000                                1.000    1.000
##     speed   (p.3_)    1.000                                1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.825
##     x2                0.131
##     x3                0.322
##     x4                0.653
##     x5                0.743
##     x6                0.772
```

```
##     x7                0.353
##     x8                0.419
##     x9                0.299
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    1.120    0.173    6.468    0.000    0.694    0.605
##     x2      (l.2_)    0.464    0.114    4.083    0.000    0.287    0.281
##     x3      (l.3_)    0.671    0.109    6.179    0.000    0.416    0.410
##   textual =~
##     x4      (l.4_)    0.948    0.094   10.083    0.000    0.956    0.849
##     x5      (l.5_)    1.123    0.113    9.930    0.000    1.132    0.898
##     x6      (l.6_)    0.885    0.087   10.181    0.000    0.893    0.819
##   speed =~
##     x7      (l.7_)    0.645    0.094    6.897    0.000    0.682    0.682
##     x8      (l.8_)    0.641    0.090    7.092    0.000    0.678    0.767
##     x9      (l.9_)    0.547    0.100    5.489    0.000    0.578    0.620
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.440    0.139    3.157    0.002    0.705    0.705
##     speed  (p.3_1)    0.336    0.154    2.180    0.029    0.513    0.513
##   textual ~~
##     speed  (p.3_2)    0.296    0.173    1.710    0.087    0.277    0.277
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (nu.1)    4.964    0.123   40.499    0.000    4.964    4.327
##    .x2      (nu.2)    6.178    0.094   66.009    0.000    6.178    6.042
##    .x3      (nu.3)    2.277    0.103   22.077    0.000    2.277    2.248
##    .x4      (nu.4)    2.744    0.098   27.920    0.000    2.744    2.436
##    .x5      (nu.5)    3.918    0.118   33.149    0.000    3.918    3.108
##    .x6      (nu.6)    1.899    0.086   22.155    0.000    1.899    1.742
##    .x7      (nu.7)    4.270    0.100   42.508    0.000    4.270    4.269
##    .x8      (nu.8)    5.659    0.090   62.668    0.000    5.659    6.400
##    .x9      (nu.9)    5.484    0.085   64.454    0.000    5.484    5.883
##     visual  (a.1.)   -0.060    0.167   -0.356    0.722   -0.096   -0.096
##     textual (a.2.)    0.645    0.161    3.999    0.000    0.640    0.640
##     speed   (a.3.)   -0.401    0.195   -2.059    0.039   -0.379   -0.379
##
## Variances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.835    0.246    3.392    0.001    0.835    0.634
##    .x2      (t.2_)    0.963    0.145    6.637    0.000    0.963    0.921
##    .x3      (t.3_)    0.853    0.148    5.757    0.000    0.853    0.832
##    .x4      (t.4_)    0.355    0.079    4.491    0.000    0.355    0.280
##    .x5      (t.5_)    0.308    0.082    3.752    0.000    0.308    0.194
##    .x6      (t.6_)    0.391    0.084    4.642    0.000    0.391    0.329
##    .x7      (t.7_)    0.535    0.127    4.215    0.000    0.535    0.535
##    .x8      (t.8_)    0.322    0.100    3.238    0.001    0.322    0.412
##    .x9      (t.9_)    0.535    0.108    4.971    0.000    0.535    0.615
##     visual  (p.1_)    0.384    0.189    2.034    0.042    1.000    1.000
##     textual (p.2_)    1.016    0.235    4.331    0.000    1.000    1.000
##     speed   (p.3_)    1.118    0.343    3.260    0.001    1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.366
##     x2                0.079
##     x3                0.168
##     x4                0.720
##     x5                0.806
##     x6                0.671
##     x7                0.465
##     x8                0.588
##     x9                0.385
```

```
summary(scalarInvarianceModelFullSyntax_fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 66 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        66
##   Number of equality constraints                    18
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
```

```
## Model Test User Model:
##                                              Standard      Robust
##   Test Statistic                              119.966     127.299
##   Degrees of freedom                               60          60
##   P-value (Chi-square)                          0.000       0.000
##   Scaling correction factor                                 0.942
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                          63.397      67.272
##     1                                          56.569      60.027
##
## Model Test Baseline Model:
##
##   Test statistic                             609.050     638.762
##   Degrees of freedom                              72          72
##   P-value                                       0.000       0.000
##   Scaling correction factor                                 0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                   0.888       0.881
##   Tucker-Lewis Index (TLI)                      0.866       0.858
##
##   Robust Comparative Fit Index (CFI)                        0.883
##   Robust Tucker-Lewis Index (TLI)                           0.859
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -2603.944   -2603.944
##   Scaling correction factor                                 0.738
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)     -2543.961   -2543.961
##   Scaling correction factor                                 0.975
##       for the MLR correction
##
##   Akaike (AIC)                               5303.889    5303.889
##   Bayesian (BIC)                             5473.681    5473.681
##   Sample-size adjusted Bayesian (BIC)        5321.511    5321.511
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.089       0.094
##   90 Percent confidence interval - lower        0.065       0.071
##   90 Percent confidence interval - upper        0.112       0.117
##   P-value RMSEA <= 0.05                         0.005       0.002
##
```

```
##   Robust RMSEA                                                0.091
##   90 Percent confidence interval - lower                      0.069
##   90 Percent confidence interval - upper                      0.113
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                             0.090       0.090
##
## Parameter Estimates:
##
##   Standard errors                               Sandwich
##   Information bread                             Observed
##   Observed information based on                  Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (lmb1)    1.120    0.173    6.468    0.000    1.120    0.908
##     x2      (lmb2)    0.464    0.114    4.083    0.000    0.464    0.363
##     x3      (lmb3)    0.671    0.109    6.179    0.000    0.671    0.567
##   textual =~
##     x4      (lmb4)    0.948    0.094   10.083    0.000    0.948    0.808
##     x5      (lmb5)    1.123    0.113    9.930    0.000    1.123    0.862
##     x6      (lmb6)    0.885    0.087   10.181    0.000    0.885    0.879
##   speed =~
##     x7      (lmb7)    0.645    0.094    6.897    0.000    0.645    0.594
##     x8      (lmb8)    0.641    0.090    7.092    0.000    0.641    0.648
##     x9      (lmb9)    0.547    0.100    5.489    0.000    0.547    0.546
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.543    0.081    6.709    0.000    0.543    0.543
##     speed             0.391    0.132    2.962    0.003    0.391    0.391
##   textual ~~
##     speed             0.351    0.100    3.517    0.000    0.351    0.351
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1      (int1)    4.964    0.123   40.499    0.000    4.964    4.024
```

```
##    .x2      (int2)    6.178    0.094   66.009    0.000    6.178    4.832
##    .x3      (int3)    2.277    0.103   22.077    0.000    2.277    1.924
##    .x4      (int4)    2.744    0.098   27.920    0.000    2.744    2.338
##    .x5      (int5)    3.918    0.118   33.149    0.000    3.918    3.008
##    .x6      (int6)    1.899    0.086   22.155    0.000    1.899    1.885
##    .x7      (int7)    4.270    0.100   42.508    0.000    4.270    3.932
##    .x8      (int8)    5.659    0.090   62.668    0.000    5.659    5.718
##    .x9      (int9)    5.484    0.085   64.454    0.000    5.484    5.480
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1                0.266    0.274    0.973    0.330    0.266    0.175
##    .x2                1.420    0.211    6.743    0.000    1.420    0.869
##    .x3                0.950    0.157    6.051    0.000    0.950    0.678
##    .x4                0.479    0.096    4.969    0.000    0.479    0.347
##    .x5                0.436    0.101    4.338    0.000    0.436    0.257
##    .x6                0.231    0.065    3.569    0.000    0.231    0.228
##    .x7                0.763    0.152    5.015    0.000    0.763    0.647
##    .x8                0.569    0.100    5.708    0.000    0.569    0.581
##    .x9                0.703    0.139    5.047    0.000    0.703    0.701
##
## R-Square:
##                    Estimate
##     x1                0.825
##     x2                0.131
##     x3                0.322
##     x4                0.653
##     x5                0.743
##     x6                0.772
##     x7                0.353
##     x8                0.419
##     x9                0.299
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1       (lmb1)    1.120    0.173    6.468    0.000    0.694    0.605
##     x2       (lmb2)    0.464    0.114    4.083    0.000    0.287    0.281
##     x3       (lmb3)    0.671    0.109    6.179    0.000    0.416    0.410
##   textual =~
```

```
##     x4      (lmb4)    0.948    0.094   10.083    0.000    0.956    0.849
##     x5      (lmb5)    1.123    0.113    9.930    0.000    1.132    0.898
##     x6      (lmb6)    0.885    0.087   10.181    0.000    0.893    0.819
##   speed =~
##     x7      (lmb7)    0.645    0.094    6.897    0.000    0.682    0.682
##     x8      (lmb8)    0.641    0.090    7.092    0.000    0.678    0.767
##     x9      (lmb9)    0.547    0.100    5.489    0.000    0.578    0.620
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.440    0.139    3.157    0.002    0.705    0.705
##     speed             0.336    0.154    2.180    0.029    0.513    0.513
##   textual ~~
##     speed             0.296    0.173    1.710    0.087    0.277    0.277
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual           -0.060    0.167   -0.356    0.722   -0.096   -0.096
##     textual           0.645    0.161    3.999    0.000    0.640    0.640
##     speed            -0.401    0.195   -2.059    0.039   -0.379   -0.379
##    .x1      (int1)    4.964    0.123   40.499    0.000    4.964    4.327
##    .x2      (int2)    6.178    0.094   66.009    0.000    6.178    6.042
##    .x3      (int3)    2.277    0.103   22.077    0.000    2.277    2.248
##    .x4      (int4)    2.744    0.098   27.920    0.000    2.744    2.436
##    .x5      (int5)    3.918    0.118   33.149    0.000    3.918    3.108
##    .x6      (int6)    1.899    0.086   22.155    0.000    1.899    1.742
##    .x7      (int7)    4.270    0.100   42.508    0.000    4.270    4.269
##    .x8      (int8)    5.659    0.090   62.668    0.000    5.659    6.400
##    .x9      (int9)    5.484    0.085   64.454    0.000    5.484    5.883
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.384    0.189    2.034    0.042    1.000    1.000
##     textual           1.016    0.235    4.331    0.000    1.000    1.000
##     speed             1.118    0.343    3.260    0.001    1.000    1.000
##    .x1                0.835    0.246    3.392    0.001    0.835    0.634
##    .x2                0.963    0.145    6.637    0.000    0.963    0.921
##    .x3                0.853    0.148    5.757    0.000    0.853    0.832
##    .x4                0.355    0.079    4.491    0.000    0.355    0.280
##    .x5                0.308    0.082    3.752    0.000    0.308    0.194
##    .x6                0.391    0.084    4.642    0.000    0.391    0.329
##    .x7                0.535    0.127    4.215    0.000    0.535    0.535
##    .x8                0.322    0.100    3.238    0.001    0.322    0.412
##    .x9                0.535    0.108    4.971    0.000    0.535    0.615
##
```

```
## R-Square:
##                    Estimate
##     x1                0.366
##     x2                0.079
##     x3                0.168
##     x4                0.720
##     x5                0.806
##     x6                0.671
##     x7                0.465
##     x8                0.588
##     x9                0.385
```

14.5.6.5 Model Fit

You can specify the null model as the baseline model using: `baseline.model = nullModelFit`

```
fitMeasures(scalarInvarianceModel_fit)
```

```
##                          npar                          fmin
##                        48.000                         0.236
##                         chisq                            df
##                       119.966                        60.000
##                       pvalue                   chisq.scaled
##                        0.000                       127.299
##                     df.scaled                  pvalue.scaled
##                       60.000                         0.000
##          chisq.scaling.factor                 baseline.chisq
##                        0.942                        609.050
##                   baseline.df                baseline.pvalue
##                       72.000                         0.000
##         baseline.chisq.scaled             baseline.df.scaled
##                       638.762                        72.000
##        baseline.pvalue.scaled baseline.chisq.scaling.factor
##                        0.000                         0.953
##                          cfi                           tli
##                        0.888                         0.866
##                         nnfi                           rfi
##                        0.866                         0.764
##                          nfi                          pnfi
##                        0.803                         0.669
##                          ifi                           rni
##                        0.891                         0.888
##                    cfi.scaled                     tli.scaled
##                        0.881                         0.858
```

```
##                    cfi.robust                    tli.robust
##                         0.883                         0.859
##                  nnfi.scaled                   nnfi.robust
##                         0.858                         0.859
##                   rfi.scaled                    nfi.scaled
##                         0.761                         0.801
##                   ifi.scaled                    rni.scaled
##                         0.884                         0.881
##                   rni.robust                          logl
##                         0.883                     -2603.944
##             unrestricted.logl                           aic
##                      -2543.961                      5303.889
##                           bic                        ntotal
##                       5473.681                       254.000
##                          bic2             scaling.factor.h1
##                       5321.511                         0.975
##             scaling.factor.h0                         rmsea
##                         0.738                         0.089
##                rmsea.ci.lower                rmsea.ci.upper
##                         0.065                         0.112
##                  rmsea.pvalue                  rmsea.scaled
##                         0.005                         0.094
##         rmsea.ci.lower.scaled         rmsea.ci.upper.scaled
##                         0.071                         0.117
##           rmsea.pvalue.scaled                  rmsea.robust
##                         0.002                         0.091
##         rmsea.ci.lower.robust         rmsea.ci.upper.robust
##                         0.069                         0.113
##           rmsea.pvalue.robust                           rmr
##                           NA                         0.106
##                     rmr_nomean                          srmr
##                         0.102                         0.090
##                   srmr_bentler           srmr_bentler_nomean
##                         0.090                         0.084
##                           crmr                   crmr_nomean
##                         0.107                         0.090
##                     srmr_mplus             srmr_mplus_nomean
##                         0.100                         0.083
##                          cn_05                         cn_01
##                        168.437                       188.122
##                            gfi                          agfi
##                         0.991                         0.985
##                           pgfi                           mfi
##                         0.551                         0.889
```

```
scalarInvarianceModelFitIndices <- fitMeasures(scalarInvarianceModel_fit)[c("cfi.robust
```

14.5.6.6 Compare Model Fit

14.5.6.6.1 Nested Model (χ^2) Difference Test The metric invariance model and the scalar ("strong factorial") invariance model are considered "nested" models. The scalar invariance model is nested within the metric invariance model because the metric invariance model includes all of the terms of the metric invariance model along with additional terms. Model fit of nested models can be compared with a chi-square difference test, also known as a likelihood ratio test or deviance test. A significant chi-square difference test would indicate that the simplified model with additional constraints (the scalar invariance model) is significantly worse fitting than the more complex model that has fewer constraints (the metric invariance model).

```
anova(metricInvarianceModel_fit, scalarInvarianceModel_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                           Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit 54 5277.4 5468.4  81.456
## scalarInvarianceModel_fit 60 5303.9 5473.7 119.966     38.958       6
##                           Pr(>Chisq)
## metricInvarianceModel_fit
## scalarInvarianceModel_fit  7.294e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The scalar invariance model fit significantly worse than the configural invariance model, so scalar invariance did not hold. This provides evidence of measurement non-invariance of indicator intercepts across groups. Measurement non-invariance of indicator intercepts poses challenges to being able to meaningfully compare levels on the latent factor across groups [Little et al., 2007].

```
round(scalarInvarianceModelFitIndices - metricInvarianceModelFitIndices, digits = 3)
```

14.5.6.6.2 Compare Other Fit Criteria

```
##   cfi.robust rmsea.robust         srmr
##       -0.060        0.024        0.016
```

14.5.6.6.3 Permutation Test Permutation procedures for testing measurement invariance are described in [Jorgensen et al., 2018]. For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed. You can specify the null model as the baseline model using: `baseline.model = nullModelFit`

Warning: this code takes a while to run based on 1000 iterations. You can reduce the number of iterations to be faster.

```
scalarInvarianceTest <- permuteMeasEq(nPermute = numPermutations,
                                      modelType = "mgcfa",
                                      con = scalarInvarianceModel_fit,
                                      uncon = metricInvarianceModel_fit,
                                      AFIs = myAFIs,
                                      moreAFIs = moreAFIs,
                                      iseed = 52242)
```

```
scalarInvarianceTest
```

```
## Omnibus p value based on parametric chi-squared difference test:
##
## Chisq diff    Df diff Pr(>Chisq)
##     38.958      6.000      0.000
##
##
## Omnibus p values based on nonparametric permutation method:
##
##              AFI.Difference p.value
## chisq                38.510       0
## chisq.scaled         40.392       0
## rmsea                 0.025       0
## cfi                  -0.061       0
## tli                  -0.066       0
## srmr                  0.016       0
## rmsea.robust          0.024       0
## cfi.robust           -0.060       0
## tli.robust           -0.065       0
```

14.5.6.7 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(scalarInvarianceModel_fit)
```

```
## $`2`
##           visual   textual     speed
## alpha  0.6524461 0.8841445 0.6430042
## omega  0.6585476 0.8841007 0.6228032
## omega2 0.6585476 0.8841007 0.6228032
## omega3 0.6504539 0.8659784 0.6125281
## avevar 0.4214365 0.7198178 0.3562555
##
## $`1`
##           visual   textual     speed
## alpha  0.4648456 0.8955126 0.7327296
## omega  0.4238460 0.8939806 0.7295190
## omega2 0.4238460 0.8939806 0.7295190
## omega3 0.4075077 0.8856526 0.7325713
## avevar 0.2174290 0.7395770 0.4747880
```

14.5.6.8 Path Diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(scalarInvarianceModel_fit,
         what = "est",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         ask = FALSE,
         edge.label.cex = 1.2)
```

14.5.7 Residual ("Strict Factorial") Invariance Model

Specify invariance of factor loadings, intercepts, and residuals across groups.

Figure 14.36: Scalar invariance model in confirmatory factor analysis

Figure 14.37: Scalar invariance model in confirmatory factor analysis

14.5.7.1 Model Syntax

```
residualInvarianceModel <- measEq.syntax(configural.model = cfaModel,
                                         data = HolzingerSwineford1939,
                                         ID.fac = "std.lv",
                                         group = "school",
                                         group.equal = c("loadings","intercepts","residuals"))
```

```
cat(as.character(residualInvarianceModel))
```

```
## ## LOADINGS:
##
## visual =~ c(NA, NA)*x1 + c(lambda.1_1, lambda.1_1)*x1
## visual =~ c(NA, NA)*x2 + c(lambda.2_1, lambda.2_1)*x2
## visual =~ c(NA, NA)*x3 + c(lambda.3_1, lambda.3_1)*x3
## textual =~ c(NA, NA)*x4 + c(lambda.4_2, lambda.4_2)*x4
## textual =~ c(NA, NA)*x5 + c(lambda.5_2, lambda.5_2)*x5
## textual =~ c(NA, NA)*x6 + c(lambda.6_2, lambda.6_2)*x6
## speed =~ c(NA, NA)*x7 + c(lambda.7_3, lambda.7_3)*x7
## speed =~ c(NA, NA)*x8 + c(lambda.8_3, lambda.8_3)*x8
## speed =~ c(NA, NA)*x9 + c(lambda.9_3, lambda.9_3)*x9
##
## ## INTERCEPTS:
##
## x1 ~ c(NA, NA)*1 + c(nu.1, nu.1)*1
## x2 ~ c(NA, NA)*1 + c(nu.2, nu.2)*1
## x3 ~ c(NA, NA)*1 + c(nu.3, nu.3)*1
## x4 ~ c(NA, NA)*1 + c(nu.4, nu.4)*1
## x5 ~ c(NA, NA)*1 + c(nu.5, nu.5)*1
## x6 ~ c(NA, NA)*1 + c(nu.6, nu.6)*1
## x7 ~ c(NA, NA)*1 + c(nu.7, nu.7)*1
## x8 ~ c(NA, NA)*1 + c(nu.8, nu.8)*1
## x9 ~ c(NA, NA)*1 + c(nu.9, nu.9)*1
##
## ## UNIQUE-FACTOR VARIANCES:
##
## x1 ~~ c(NA, NA)*x1 + c(theta.1_1, theta.1_1)*x1
## x2 ~~ c(NA, NA)*x2 + c(theta.2_2, theta.2_2)*x2
## x3 ~~ c(NA, NA)*x3 + c(theta.3_3, theta.3_3)*x3
## x4 ~~ c(NA, NA)*x4 + c(theta.4_4, theta.4_4)*x4
## x5 ~~ c(NA, NA)*x5 + c(theta.5_5, theta.5_5)*x5
## x6 ~~ c(NA, NA)*x6 + c(theta.6_6, theta.6_6)*x6
## x7 ~~ c(NA, NA)*x7 + c(theta.7_7, theta.7_7)*x7
## x8 ~~ c(NA, NA)*x8 + c(theta.8_8, theta.8_8)*x8
```

```
## x9 ~~ c(NA, NA)*x9 + c(theta.9_9, theta.9_9)*x9
##
## ## LATENT MEANS/INTERCEPTS:
##
## visual ~ c(0, NA)*1 + c(alpha.1.g1, alpha.1.g2)*1
## textual ~ c(0, NA)*1 + c(alpha.2.g1, alpha.2.g2)*1
## speed ~ c(0, NA)*1 + c(alpha.3.g1, alpha.3.g2)*1
##
## ## COMMON-FACTOR VARIANCES:
##
## visual ~~ c(1, NA)*visual + c(psi.1_1.g1, psi.1_1.g2)*visual
## textual ~~ c(1, NA)*textual + c(psi.2_2.g1, psi.2_2.g2)*textual
## speed ~~ c(1, NA)*speed + c(psi.3_3.g1, psi.3_3.g2)*speed
##
## ## COMMON-FACTOR COVARIANCES:
##
## visual ~~ c(NA, NA)*textual + c(psi.2_1.g1, psi.2_1.g2)*textual
## visual ~~ c(NA, NA)*speed + c(psi.3_1.g1, psi.3_1.g2)*speed
## textual ~~ c(NA, NA)*speed + c(psi.3_2.g1, psi.3_2.g2)*speed
```

```
residualInvarianceSyntax <- as.character(residualInvarianceModel)
```

```
summary(residualInvarianceModel)
```

14.5.7.1.1 Summary of Model Features

```
## This lavaan model syntax specifies a CFA with 9 manifest indicators of 3 common fac
##
## To identify the location and scale of each common factor, the factor means and varia
##
## Pattern matrix indicating num(eric), ord(ered), and lat(ent) indicators per factor:
##
##    visual textual speed
## x1 num
## x2 num
## x3 num
## x4        num
## x5        num
## x6        num
## x7                num
## x8                num
## x9                num
```

```
##
## The following types of parameter were constrained to equality across groups:
##
##  loadings
##  intercepts
##  residuals
```

14.5.7.2 Model Syntax in Table Form

```
lavaanify(residualInvarianceModel, ngroups = 2)
```

```
##    id     lhs op     rhs user block group free ustart exo      label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0 lambda.1_1   .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0 lambda.2_1   .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0 lambda.3_1   .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0 lambda.4_2   .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0 lambda.5_2   .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0 lambda.6_2   .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0 lambda.7_3   .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0 lambda.8_3   .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0 lambda.9_3   .p9.
## 10 10      x1 ~1            1     1     1   10     NA   0       nu.1  .p10.
## 11 11      x2 ~1            1     1     1   11     NA   0       nu.2  .p11.
## 12 12      x3 ~1            1     1     1   12     NA   0       nu.3  .p12.
## 13 13      x4 ~1            1     1     1   13     NA   0       nu.4  .p13.
## 14 14      x5 ~1            1     1     1   14     NA   0       nu.5  .p14.
## 15 15      x6 ~1            1     1     1   15     NA   0       nu.6  .p15.
## 16 16      x7 ~1            1     1     1   16     NA   0       nu.7  .p16.
## 17 17      x8 ~1            1     1     1   17     NA   0       nu.8  .p17.
## 18 18      x9 ~1            1     1     1   18     NA   0       nu.9  .p18.
## 19 19      x1 ~~      x1    1     1     1   19     NA   0  theta.1_1  .p19.
## 20 20      x2 ~~      x2    1     1     1   20     NA   0  theta.2_2  .p20.
## 21 21      x3 ~~      x3    1     1     1   21     NA   0  theta.3_3  .p21.
## 22 22      x4 ~~      x4    1     1     1   22     NA   0  theta.4_4  .p22.
## 23 23      x5 ~~      x5    1     1     1   23     NA   0  theta.5_5  .p23.
## 24 24      x6 ~~      x6    1     1     1   24     NA   0  theta.6_6  .p24.
## 25 25      x7 ~~      x7    1     1     1   25     NA   0  theta.7_7  .p25.
## 26 26      x8 ~~      x8    1     1     1   26     NA   0  theta.8_8  .p26.
## 27 27      x9 ~~      x9    1     1     1   27     NA   0  theta.9_9  .p27.
## 28 28  visual ~1            1     1     1    0      0   0 alpha.1.g1  .p28.
## 29 29 textual ~1            1     1     1    0      0   0 alpha.2.g1  .p29.
## 30 30   speed ~1            1     1     1    0      0   0 alpha.3.g1  .p30.
## 31 31  visual ~~  visual    1     1     1    0      1   0 psi.1_1.g1  .p31.
```

```
## 32 32 textual ~~ textual    1     1     1    0      1   0 psi.2_2.g1  .p32.
## 33 33   speed ~~   speed    1     1     1    0      1   0 psi.3_3.g1  .p33.
## 34 34  visual ~~ textual    1     1     1   28     NA   0 psi.2_1.g1  .p34.
## 35 35  visual ~~   speed    1     1     1   29     NA   0 psi.3_1.g1  .p35.
## 36 36 textual ~~   speed    1     1     1   30     NA   0 psi.3_2.g1  .p36.
## 37 37  visual =~      x1    1     2     2   31     NA   0 lambda.1_1  .p37.
## 38 38  visual =~      x2    1     2     2   32     NA   0 lambda.2_1  .p38.
## 39 39  visual =~      x3    1     2     2   33     NA   0 lambda.3_1  .p39.
## 40 40 textual =~      x4    1     2     2   34     NA   0 lambda.4_2  .p40.
## 41 41 textual =~      x5    1     2     2   35     NA   0 lambda.5_2  .p41.
## 42 42 textual =~      x6    1     2     2   36     NA   0 lambda.6_2  .p42.
## 43 43   speed =~      x7    1     2     2   37     NA   0 lambda.7_3  .p43.
## 44 44   speed =~      x8    1     2     2   38     NA   0 lambda.8_3  .p44.
## 45 45   speed =~      x9    1     2     2   39     NA   0 lambda.9_3  .p45.
## 46 46      x1 ~1            1     2     2   40     NA   0       nu.1  .p46.
## 47 47      x2 ~1            1     2     2   41     NA   0       nu.2  .p47.
## 48 48      x3 ~1            1     2     2   42     NA   0       nu.3  .p48.
## 49 49      x4 ~1            1     2     2   43     NA   0       nu.4  .p49.
## 50 50      x5 ~1            1     2     2   44     NA   0       nu.5  .p50.
## 51 51      x6 ~1            1     2     2   45     NA   0       nu.6  .p51.
## 52 52      x7 ~1            1     2     2   46     NA   0       nu.7  .p52.
## 53 53      x8 ~1            1     2     2   47     NA   0       nu.8  .p53.
## 54 54      x9 ~1            1     2     2   48     NA   0       nu.9  .p54.
## 55 55      x1 ~~      x1    1     2     2   49     NA   0  theta.1_1  .p55.
## 56 56      x2 ~~      x2    1     2     2   50     NA   0  theta.2_2  .p56.
## 57 57      x3 ~~      x3    1     2     2   51     NA   0  theta.3_3  .p57.
## 58 58      x4 ~~      x4    1     2     2   52     NA   0  theta.4_4  .p58.
## 59 59      x5 ~~      x5    1     2     2   53     NA   0  theta.5_5  .p59.
## 60 60      x6 ~~      x6    1     2     2   54     NA   0  theta.6_6  .p60.
## 61 61      x7 ~~      x7    1     2     2   55     NA   0  theta.7_7  .p61.
## 62 62      x8 ~~      x8    1     2     2   56     NA   0  theta.8_8  .p62.
## 63 63      x9 ~~      x9    1     2     2   57     NA   0  theta.9_9  .p63.
## 64 64  visual ~1            1     2     2   58     NA   0 alpha.1.g2  .p64.
## 65 65 textual ~1            1     2     2   59     NA   0 alpha.2.g2  .p65.
## 66 66   speed ~1            1     2     2   60     NA   0 alpha.3.g2  .p66.
## 67 67  visual ~~  visual    1     2     2   61     NA   0 psi.1_1.g2  .p67.
## 68 68 textual ~~ textual    1     2     2   62     NA   0 psi.2_2.g2  .p68.
## 69 69   speed ~~   speed    1     2     2   63     NA   0 psi.3_3.g2  .p69.
## 70 70  visual ~~ textual    1     2     2   64     NA   0 psi.2_1.g2  .p70.
## 71 71  visual ~~   speed    1     2     2   65     NA   0 psi.3_1.g2  .p71.
## 72 72 textual ~~   speed    1     2     2   66     NA   0 psi.3_2.g2  .p72.
## 73 73    .p1. ==   .p37.    2     0     0    0     NA   0
## 74 74    .p2. ==   .p38.    2     0     0    0     NA   0
## 75 75    .p3. ==   .p39.    2     0     0    0     NA   0
## 76 76    .p4. ==   .p40.    2     0     0    0     NA   0
## 77 77    .p5. ==   .p41.    2     0     0    0     NA   0
```

```
## 78 78    .p6. ==   .p42.    2     0     0    0     NA   0
## 79 79    .p7. ==   .p43.    2     0     0    0     NA   0
## 80 80    .p8. ==   .p44.    2     0     0    0     NA   0
## 81 81    .p9. ==   .p45.    2     0     0    0     NA   0
## 82 82   .p10. ==   .p46.    2     0     0    0     NA   0
## 83 83   .p11. ==   .p47.    2     0     0    0     NA   0
## 84 84   .p12. ==   .p48.    2     0     0    0     NA   0
## 85 85   .p13. ==   .p49.    2     0     0    0     NA   0
## 86 86   .p14. ==   .p50.    2     0     0    0     NA   0
## 87 87   .p15. ==   .p51.    2     0     0    0     NA   0
## 88 88   .p16. ==   .p52.    2     0     0    0     NA   0
## 89 89   .p17. ==   .p53.    2     0     0    0     NA   0
## 90 90   .p18. ==   .p54.    2     0     0    0     NA   0
## 91 91   .p19. ==   .p55.    2     0     0    0     NA   0
## 92 92   .p20. ==   .p56.    2     0     0    0     NA   0
## 93 93   .p21. ==   .p57.    2     0     0    0     NA   0
## 94 94   .p22. ==   .p58.    2     0     0    0     NA   0
## 95 95   .p23. ==   .p59.    2     0     0    0     NA   0
## 96 96   .p24. ==   .p60.    2     0     0    0     NA   0
## 97 97   .p25. ==   .p61.    2     0     0    0     NA   0
## 98 98   .p26. ==   .p62.    2     0     0    0     NA   0
## 99 99   .p27. ==   .p63.    2     0     0    0     NA   0
```

```
lavaanify(cfaModel_residualInvariance, ngroups = 2)
```

```
##    id     lhs op     rhs user block group free ustart exo    label plabel
## 1   1  visual =~      x1    1     1     1    1     NA   0 lambdax1   .p1.
## 2   2  visual =~      x2    1     1     1    2     NA   0 lambdax2   .p2.
## 3   3  visual =~      x3    1     1     1    3     NA   0 lambdax3   .p3.
## 4   4 textual =~      x4    1     1     1    4     NA   0 lambdax4   .p4.
## 5   5 textual =~      x5    1     1     1    5     NA   0 lambdax5   .p5.
## 6   6 textual =~      x6    1     1     1    6     NA   0 lambdax6   .p6.
## 7   7   speed =~      x7    1     1     1    7     NA   0 lambdax7   .p7.
## 8   8   speed =~      x8    1     1     1    8     NA   0 lambdax8   .p8.
## 9   9   speed =~      x9    1     1     1    9     NA   0 lambdax9   .p9.
## 10 10  visual ~1            1     1     1    0      0   0           .p10.
## 11 11 textual ~1            1     1     1    0      0   0           .p11.
## 12 12   speed ~1            1     1     1    0      0   0           .p12.
## 13 13  visual ~~  visual    1     1     1    0      1   0           .p13.
## 14 14 textual ~~ textual    1     1     1    0      1   0           .p14.
## 15 15   speed ~~   speed    1     1     1    0      1   0           .p15.
## 16 16  visual ~~ textual    1     1     1   10     NA   0           .p16.
## 17 17  visual ~~   speed    1     1     1   11     NA   0           .p17.
## 18 18 textual ~~   speed    1     1     1   12     NA   0           .p18.
## 19 19      x1 ~~      x1    1     1     1   13     NA   0  residx1  .p19.
```

```
## 20 20      x2 ~~      x2    1     1     1   14     NA   0  residx2  .p20.
## 21 21      x3 ~~      x3    1     1     1   15     NA   0  residx3  .p21.
## 22 22      x4 ~~      x4    1     1     1   16     NA   0  residx4  .p22.
## 23 23      x5 ~~      x5    1     1     1   17     NA   0  residx5  .p23.
## 24 24      x6 ~~      x6    1     1     1   18     NA   0  residx6  .p24.
## 25 25      x7 ~~      x7    1     1     1   19     NA   0  residx7  .p25.
## 26 26      x8 ~~      x8    1     1     1   20     NA   0  residx8  .p26.
## 27 27      x9 ~~      x9    1     1     1   21     NA   0  residx9  .p27.
## 28 28      x1 ~1            1     1     1   22     NA   0    intx1  .p28.
## 29 29      x2 ~1            1     1     1   23     NA   0    intx2  .p29.
## 30 30      x3 ~1            1     1     1   24     NA   0    intx3  .p30.
## 31 31      x4 ~1            1     1     1   25     NA   0    intx4  .p31.
## 32 32      x5 ~1            1     1     1   26     NA   0    intx5  .p32.
## 33 33      x6 ~1            1     1     1   27     NA   0    intx6  .p33.
## 34 34      x7 ~1            1     1     1   28     NA   0    intx7  .p34.
## 35 35      x8 ~1            1     1     1   29     NA   0    intx8  .p35.
## 36 36      x9 ~1            1     1     1   30     NA   0    intx9  .p36.
## 37 37  visual =~      x1    1     2     2   31     NA   0 lambdax1  .p37.
## 38 38  visual =~      x2    1     2     2   32     NA   0 lambdax2  .p38.
## 39 39  visual =~      x3    1     2     2   33     NA   0 lambdax3  .p39.
## 40 40 textual =~      x4    1     2     2   34     NA   0 lambdax4  .p40.
## 41 41 textual =~      x5    1     2     2   35     NA   0 lambdax5  .p41.
## 42 42 textual =~      x6    1     2     2   36     NA   0 lambdax6  .p42.
## 43 43   speed =~      x7    1     2     2   37     NA   0 lambdax7  .p43.
## 44 44   speed =~      x8    1     2     2   38     NA   0 lambdax8  .p44.
## 45 45   speed =~      x9    1     2     2   39     NA   0 lambdax9  .p45.
## 46 46  visual ~1            1     2     2   40     NA   0           .p46.
## 47 47 textual ~1            1     2     2   41     NA   0           .p47.
## 48 48   speed ~1            1     2     2   42     NA   0           .p48.
## 49 49  visual ~~  visual    1     2     2   43     NA   0           .p49.
## 50 50 textual ~~ textual    1     2     2   44     NA   0           .p50.
## 51 51   speed ~~   speed    1     2     2   45     NA   0           .p51.
## 52 52  visual ~~ textual    1     2     2   46     NA   0           .p52.
## 53 53  visual ~~   speed    1     2     2   47     NA   0           .p53.
## 54 54 textual ~~   speed    1     2     2   48     NA   0           .p54.
## 55 55      x1 ~~      x1    1     2     2   49     NA   0  residx1  .p55.
## 56 56      x2 ~~      x2    1     2     2   50     NA   0  residx2  .p56.
## 57 57      x3 ~~      x3    1     2     2   51     NA   0  residx3  .p57.
## 58 58      x4 ~~      x4    1     2     2   52     NA   0  residx4  .p58.
## 59 59      x5 ~~      x5    1     2     2   53     NA   0  residx5  .p59.
## 60 60      x6 ~~      x6    1     2     2   54     NA   0  residx6  .p60.
## 61 61      x7 ~~      x7    1     2     2   55     NA   0  residx7  .p61.
## 62 62      x8 ~~      x8    1     2     2   56     NA   0  residx8  .p62.
## 63 63      x9 ~~      x9    1     2     2   57     NA   0  residx9  .p63.
## 64 64      x1 ~1            1     2     2   58     NA   0    intx1  .p64.
## 65 65      x2 ~1            1     2     2   59     NA   0    intx2  .p65.
```

```
## 66 66      x3 ~1            1     2     2   60     NA   0    intx3  .p66.
## 67 67      x4 ~1            1     2     2   61     NA   0    intx4  .p67.
## 68 68      x5 ~1            1     2     2   62     NA   0    intx5  .p68.
## 69 69      x6 ~1            1     2     2   63     NA   0    intx6  .p69.
## 70 70      x7 ~1            1     2     2   64     NA   0    intx7  .p70.
## 71 71      x8 ~1            1     2     2   65     NA   0    intx8  .p71.
## 72 72      x9 ~1            1     2     2   66     NA   0    intx9  .p72.
## 73 73    .p1. ==   .p37.    2     0     0    0     NA   0
## 74 74    .p2. ==   .p38.    2     0     0    0     NA   0
## 75 75    .p3. ==   .p39.    2     0     0    0     NA   0
## 76 76    .p4. ==   .p40.    2     0     0    0     NA   0
## 77 77    .p5. ==   .p41.    2     0     0    0     NA   0
## 78 78    .p6. ==   .p42.    2     0     0    0     NA   0
## 79 79    .p7. ==   .p43.    2     0     0    0     NA   0
## 80 80    .p8. ==   .p44.    2     0     0    0     NA   0
## 81 81    .p9. ==   .p45.    2     0     0    0     NA   0
## 82 82   .p19. ==   .p55.    2     0     0    0     NA   0
## 83 83   .p20. ==   .p56.    2     0     0    0     NA   0
## 84 84   .p21. ==   .p57.    2     0     0    0     NA   0
## 85 85   .p22. ==   .p58.    2     0     0    0     NA   0
## 86 86   .p23. ==   .p59.    2     0     0    0     NA   0
## 87 87   .p24. ==   .p60.    2     0     0    0     NA   0
## 88 88   .p25. ==   .p61.    2     0     0    0     NA   0
## 89 89   .p26. ==   .p62.    2     0     0    0     NA   0
## 90 90   .p27. ==   .p63.    2     0     0    0     NA   0
## 91 91   .p28. ==   .p64.    2     0     0    0     NA   0
## 92 92   .p29. ==   .p65.    2     0     0    0     NA   0
## 93 93   .p30. ==   .p66.    2     0     0    0     NA   0
## 94 94   .p31. ==   .p67.    2     0     0    0     NA   0
## 95 95   .p32. ==   .p68.    2     0     0    0     NA   0
## 96 96   .p33. ==   .p69.    2     0     0    0     NA   0
## 97 97   .p34. ==   .p70.    2     0     0    0     NA   0
## 98 98   .p35. ==   .p71.    2     0     0    0     NA   0
## 99 99   .p36. ==   .p72.    2     0     0    0     NA   0
```

14.5.7.3 Fit the Model

```
residualInvarianceModel_fit <- cfa(residualInvarianceSyntax,
                                   data = HolzingerSwineford1939,
                                   group = "school",
                                   missing = "ML",
                                   estimator = "MLR")
```

```
residualInvarianceModelFullSyntax_fit <- lavaan(cfaModel_residualInvariance,
                data = HolzingerSwineford1939,
                group = "school",
                missing = "ML",
                estimator = "MLR")
```

14.5.7.4 Model Summary

```
summary(residualInvarianceModel_fit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 62 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        66
##   Number of equality constraints                    27
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               138.270     148.444
##   Degrees of freedom                                69          69
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.931
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                           70.226      75.394
##     1                                           68.044      73.051
##
## Model Test Baseline Model:
##
##   Test statistic                              609.050     638.762
##   Degrees of freedom                               72          72
##   P-value                                       0.000       0.000
```

```
##   Scaling correction factor                                  0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.871       0.860
##   Tucker-Lewis Index (TLI)                       0.865       0.854
##
##   Robust Comparative Fit Index (CFI)                         0.863
##   Robust Tucker-Lewis Index (TLI)                            0.857
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2613.096   -2613.096
##   Scaling correction factor                                  0.621
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -2543.961   -2543.961
##   Scaling correction factor                                  0.975
##       for the MLR correction
##
##   Akaike (AIC)                                5304.193    5304.193
##   Bayesian (BIC)                              5442.149    5442.149
##   Sample-size adjusted Bayesian (BIC)         5318.510    5318.510
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.089       0.095
##   90 Percent confidence interval - lower         0.067       0.073
##   90 Percent confidence interval - upper         0.110       0.117
##   P-value RMSEA <= 0.05                          0.003       0.001
##
##   Robust RMSEA                                               0.092
##   90 Percent confidence interval - lower                     0.072
##   90 Percent confidence interval - upper                     0.112
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.096       0.096
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
```

```
##
## Latent Variables:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    0.961    0.153    6.263    0.000    0.961    0.755
##     x2      (l.2_)    0.489    0.127    3.846    0.000    0.489    0.413
##     x3      (l.3_)    0.736    0.129    5.723    0.000    0.736    0.633
##   textual =~
##     x4      (l.4_)    0.955    0.090   10.584    0.000    0.955    0.831
##     x5      (l.5_)    1.140    0.106   10.747    0.000    1.140    0.885
##     x6      (l.6_)    0.895    0.082   10.959    0.000    0.895    0.846
##   speed =~
##     x7      (l.7_)    0.645    0.105    6.159    0.000    0.645    0.609
##     x8      (l.8_)    0.679    0.101    6.729    0.000    0.679    0.708
##     x9      (l.9_)    0.617    0.116    5.343    0.000    0.617    0.628
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.501    0.128    3.921    0.000    0.501    0.501
##     speed  (p.3_1)    0.419    0.156    2.688    0.007    0.419    0.419
##   textual ~~
##     speed  (p.3_2)    0.316    0.098    3.238    0.001    0.316    0.316
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (nu.1)    5.026    0.121   41.695    0.000    5.026    3.948
##    .x2      (nu.2)    6.167    0.098   62.827    0.000    6.167    5.209
##    .x3      (nu.3)    2.317    0.110   21.004    0.000    2.317    1.992
##    .x4      (nu.4)    2.758    0.098   28.024    0.000    2.758    2.400
##    .x5      (nu.5)    3.908    0.118   33.200    0.000    3.908    3.035
##    .x6      (nu.6)    1.907    0.084   22.590    0.000    1.907    1.803
##    .x7      (nu.7)    4.282    0.095   45.282    0.000    4.282    4.048
##    .x8      (nu.8)    5.635    0.091   61.767    0.000    5.635    5.879
##    .x9      (nu.9)    5.471    0.084   64.963    0.000    5.471    5.566
##     visual  (a.1.)    0.000                                0.000    0.000
##     textual (a.2.)    0.000                                0.000    0.000
##     speed   (a.3.)    0.000                                0.000    0.000
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.697    0.207    3.364    0.001    0.697    0.430
##    .x2      (t.2_)    1.163    0.129    9.009    0.000    1.163    0.830
##    .x3      (t.3_)    0.811    0.128    6.325    0.000    0.811    0.600
##    .x4      (t.4_)    0.408    0.062    6.592    0.000    0.408    0.309
##    .x5      (t.5_)    0.359    0.067    5.381    0.000    0.359    0.216
```

```
##    .x6      (t.6_)    0.318    0.052    6.083    0.000    0.318    0.284
##    .x7      (t.7_)    0.704    0.120    5.874    0.000    0.704    0.629
##    .x8      (t.8_)    0.458    0.090    5.102    0.000    0.458    0.499
##    .x9      (t.9_)    0.585    0.111    5.260    0.000    0.585    0.606
##     visual  (p.1_)    1.000                                1.000    1.000
##     textual (p.2_)    1.000                                1.000    1.000
##     speed   (p.3_)    1.000                                1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.570
##     x2                0.170
##     x3                0.400
##     x4                0.691
##     x5                0.784
##     x6                0.716
##     x7                0.371
##     x8                0.501
##     x9                0.394
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (l.1_)    0.961    0.153    6.263    0.000    0.684    0.634
##     x2      (l.2_)    0.489    0.127    3.846    0.000    0.348    0.307
##     x3      (l.3_)    0.736    0.129    5.723    0.000    0.524    0.503
##   textual =~
##     x4      (l.4_)    0.955    0.090   10.584    0.000    0.956    0.832
##     x5      (l.5_)    1.140    0.106   10.747    0.000    1.141    0.885
##     x6      (l.6_)    0.895    0.082   10.959    0.000    0.896    0.846
##   speed =~
##     x7      (l.7_)    0.645    0.105    6.159    0.000    0.599    0.581
##     x8      (l.8_)    0.679    0.101    6.729    0.000    0.631    0.682
##     x9      (l.9_)    0.617    0.116    5.343    0.000    0.574    0.600
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textul  (p.2_)    0.465    0.142    3.270    0.001    0.653    0.653
##     speed  (p.3_1)    0.362    0.160    2.261    0.024    0.547    0.547
##   textual ~~
##     speed  (p.3_2)    0.287    0.160    1.796    0.073    0.308    0.308
##
```

```
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (nu.1)    5.026    0.121   41.695    0.000    5.026    4.657
##    .x2      (nu.2)    6.167    0.098   62.827    0.000    6.167    5.443
##    .x3      (nu.3)    2.317    0.110   21.004    0.000    2.317    2.224
##    .x4      (nu.4)    2.758    0.098   28.024    0.000    2.758    2.398
##    .x5      (nu.5)    3.908    0.118   33.200    0.000    3.908    3.033
##    .x6      (nu.6)    1.907    0.084   22.590    0.000    1.907    1.802
##    .x7      (nu.7)    4.282    0.095   45.282    0.000    4.282    4.154
##    .x8      (nu.8)    5.635    0.091   61.767    0.000    5.635    6.090
##    .x9      (nu.9)    5.471    0.084   64.963    0.000    5.471    5.722
##     visual  (a.1.)   -0.132    0.196   -0.673    0.501   -0.185   -0.185
##     textual (a.2.)    0.637    0.156    4.091    0.000    0.636    0.636
##     speed   (a.3.)   -0.355    0.181   -1.956    0.050   -0.382   -0.382
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1      (t.1_)    0.697    0.207    3.364    0.001    0.697    0.598
##    .x2      (t.2_)    1.163    0.129    9.009    0.000    1.163    0.906
##    .x3      (t.3_)    0.811    0.128    6.325    0.000    0.811    0.747
##    .x4      (t.4_)    0.408    0.062    6.592    0.000    0.408    0.308
##    .x5      (t.5_)    0.359    0.067    5.381    0.000    0.359    0.216
##    .x6      (t.6_)    0.318    0.052    6.083    0.000    0.318    0.284
##    .x7      (t.7_)    0.704    0.120    5.874    0.000    0.704    0.662
##    .x8      (t.8_)    0.458    0.090    5.102    0.000    0.458    0.535
##    .x9      (t.9_)    0.585    0.111    5.260    0.000    0.585    0.640
##     visual  (p.1_)    0.507    0.199    2.544    0.011    1.000    1.000
##     textual (p.2_)    1.002    0.226    4.434    0.000    1.000    1.000
##     speed   (p.3_)    0.864    0.260    3.329    0.001    1.000    1.000
##
## R-Square:
##                    Estimate
##     x1                0.402
##     x2                0.094
##     x3                0.253
##     x4                0.692
##     x5                0.784
##     x6                0.716
##     x7                0.338
##     x8                0.465
##     x9                0.360
```

```
summary(residualInvarianceModelFullSyntax_fit,
        fit.measures = TRUE,
        standardized = TRUE,
```

```
        rsquare = TRUE)
```

```
## lavaan 0.6-8 ended normally after 64 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        66
##   Number of equality constraints                    27
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               138.270     148.444
##   Degrees of freedom                                69          69
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.931
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                           70.226      75.394
##     1                                           68.044      73.051
##
## Model Test Baseline Model:
##
##   Test statistic                               609.050     638.762
##   Degrees of freedom                                72          72
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  0.953
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.871       0.860
##   Tucker-Lewis Index (TLI)                       0.865       0.854
##
##   Robust Comparative Fit Index (CFI)                         0.863
##   Robust Tucker-Lewis Index (TLI)                            0.857
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2613.096   -2613.096
```

```
##   Scaling correction factor                                  0.621
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -2543.961   -2543.961
##   Scaling correction factor                                  0.975
##       for the MLR correction
##
##   Akaike (AIC)                                5304.193    5304.193
##   Bayesian (BIC)                              5442.149    5442.149
##   Sample-size adjusted Bayesian (BIC)         5318.510    5318.510
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.089       0.095
##   90 Percent confidence interval - lower         0.067       0.073
##   90 Percent confidence interval - upper         0.110       0.117
##   P-value RMSEA <= 0.05                          0.003       0.001
##
##   Robust RMSEA                                               0.092
##   90 Percent confidence interval - lower                     0.072
##   90 Percent confidence interval - upper                     0.112
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.096       0.096
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (lmb1)    0.961    0.153    6.263    0.000    0.961    0.755
##     x2      (lmb2)    0.489    0.127    3.846    0.000    0.489    0.413
##     x3      (lmb3)    0.736    0.129    5.723    0.000    0.736    0.633
##   textual =~
##     x4      (lmb4)    0.955    0.090   10.584    0.000    0.955    0.831
##     x5      (lmb5)    1.140    0.106   10.747    0.000    1.140    0.885
##     x6      (lmb6)    0.895    0.082   10.959    0.000    0.895    0.846
##   speed =~
##     x7      (lmb7)    0.645    0.105    6.159    0.000    0.645    0.609
```

```
##     x8      (lmb8)    0.679    0.101    6.729    0.000    0.679    0.708
##     x9      (lmb9)    0.617    0.116    5.343    0.000    0.617    0.628
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.501    0.128    3.921    0.000    0.501    0.501
##     speed             0.419    0.156    2.688    0.007    0.419    0.419
##   textual ~~
##     speed             0.316    0.098    3.238    0.001    0.316    0.316
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.000                               0.000    0.000
##     textual           0.000                               0.000    0.000
##     speed             0.000                               0.000    0.000
##    .x1      (int1)    5.026    0.121   41.695    0.000    5.026    3.948
##    .x2      (int2)    6.167    0.098   62.827    0.000    6.167    5.209
##    .x3      (int3)    2.317    0.110   21.004    0.000    2.317    1.992
##    .x4      (int4)    2.758    0.098   28.024    0.000    2.758    2.400
##    .x5      (int5)    3.908    0.118   33.200    0.000    3.908    3.035
##    .x6      (int6)    1.907    0.084   22.590    0.000    1.907    1.803
##    .x7      (int7)    4.282    0.095   45.281    0.000    4.282    4.048
##    .x8      (int8)    5.635    0.091   61.767    0.000    5.635    5.879
##    .x9      (int9)    5.471    0.084   64.963    0.000    5.471    5.566
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            1.000                               1.000    1.000
##     textual           1.000                               1.000    1.000
##     speed             1.000                               1.000    1.000
##    .x1      (rsd1)    0.697    0.207    3.364    0.001    0.697    0.430
##    .x2      (rsd2)    1.163    0.129    9.009    0.000    1.163    0.830
##    .x3      (rsd3)    0.811    0.128    6.325    0.000    0.811    0.600
##    .x4      (rsd4)    0.408    0.062    6.592    0.000    0.408    0.309
##    .x5      (rsd5)    0.359    0.067    5.381    0.000    0.359    0.216
##    .x6      (rsd6)    0.318    0.052    6.083    0.000    0.318    0.284
##    .x7      (rsd7)    0.704    0.120    5.874    0.000    0.704    0.629
##    .x8      (rsd8)    0.458    0.090    5.102    0.000    0.458    0.499
##    .x9      (rsd9)    0.585    0.111    5.260    0.000    0.585    0.606
##
## R-Square:
##                    Estimate
##     x1                0.570
##     x2                0.170
##     x3                0.400
```

```
##     x4                0.691
##     x5                0.784
##     x6                0.716
##     x7                0.371
##     x8                0.501
##     x9                0.394
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1      (lmb1)    0.961    0.153    6.263    0.000    0.684    0.634
##     x2      (lmb2)    0.489    0.127    3.846    0.000    0.348    0.307
##     x3      (lmb3)    0.736    0.129    5.723    0.000    0.524    0.503
##   textual =~
##     x4      (lmb4)    0.955    0.090   10.584    0.000    0.956    0.832
##     x5      (lmb5)    1.140    0.106   10.747    0.000    1.141    0.885
##     x6      (lmb6)    0.895    0.082   10.959    0.000    0.896    0.846
##   speed =~
##     x7      (lmb7)    0.645    0.105    6.159    0.000    0.599    0.581
##     x8      (lmb8)    0.679    0.101    6.729    0.000    0.631    0.682
##     x9      (lmb9)    0.617    0.116    5.343    0.000    0.574    0.600
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual ~~
##     textual           0.465    0.142    3.270    0.001    0.653    0.653
##     speed             0.362    0.160    2.261    0.024    0.547    0.547
##   textual ~~
##     speed             0.287    0.160    1.796    0.073    0.308    0.308
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual           -0.132    0.196   -0.673    0.501   -0.185   -0.185
##     textual           0.637    0.156    4.091    0.000    0.636    0.636
##     speed            -0.355    0.181   -1.956    0.050   -0.382   -0.382
##    .x1      (int1)    5.026    0.121   41.695    0.000    5.026    4.657
##    .x2      (int2)    6.167    0.098   62.827    0.000    6.167    5.443
##    .x3      (int3)    2.317    0.110   21.004    0.000    2.317    2.224
##    .x4      (int4)    2.758    0.098   28.024    0.000    2.758    2.398
##    .x5      (int5)    3.908    0.118   33.200    0.000    3.908    3.033
##    .x6      (int6)    1.907    0.084   22.590    0.000    1.907    1.802
##    .x7      (int7)    4.282    0.095   45.281    0.000    4.282    4.154
##    .x8      (int8)    5.635    0.091   61.767    0.000    5.635    6.090
```

```
##    .x9      (int9)    5.471    0.084   64.963    0.000    5.471    5.722
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     visual            0.507    0.199    2.544    0.011    1.000    1.000
##     textual           1.002    0.226    4.434    0.000    1.000    1.000
##     speed             0.864    0.260    3.329    0.001    1.000    1.000
##    .x1      (rsd1)    0.697    0.207    3.364    0.001    0.697    0.598
##    .x2      (rsd2)    1.163    0.129    9.009    0.000    1.163    0.906
##    .x3      (rsd3)    0.811    0.128    6.325    0.000    0.811    0.747
##    .x4      (rsd4)    0.408    0.062    6.592    0.000    0.408    0.308
##    .x5      (rsd5)    0.359    0.067    5.381    0.000    0.359    0.216
##    .x6      (rsd6)    0.318    0.052    6.083    0.000    0.318    0.284
##    .x7      (rsd7)    0.704    0.120    5.874    0.000    0.704    0.662
##    .x8      (rsd8)    0.458    0.090    5.102    0.000    0.458    0.535
##    .x9      (rsd9)    0.585    0.111    5.260    0.000    0.585    0.640
##
## R-Square:
##                    Estimate
##     x1                0.402
##     x2                0.094
##     x3                0.253
##     x4                0.692
##     x5                0.784
##     x6                0.716
##     x7                0.338
##     x8                0.465
##     x9                0.360
```

14.5.7.5 Model Fit

You can specify the null model as the baseline model using: `baseline.model = nullModelFit`

```
fitMeasures(residualInvarianceModel_fit)
```

```
##                          npar                          fmin
##                        39.000                         0.272
##                         chisq                            df
##                       138.270                        69.000
##                        pvalue                  chisq.scaled
##                         0.000                       148.444
##                     df.scaled                 pvalue.scaled
##                        69.000                         0.000
##          chisq.scaling.factor                 baseline.chisq
```

```
##                         0.931                       609.050
##                   baseline.df               baseline.pvalue
##                        72.000                         0.000
##         baseline.chisq.scaled            baseline.df.scaled
##                       638.762                        72.000
##        baseline.pvalue.scaled baseline.chisq.scaling.factor
##                         0.000                         0.953
##                           cfi                           tli
##                         0.871                         0.865
##                          nnfi                           rfi
##                         0.865                         0.763
##                           nfi                          pnfi
##                         0.773                         0.741
##                           ifi                           rni
##                         0.872                         0.871
##                    cfi.scaled                    tli.scaled
##                         0.860                         0.854
##                    cfi.robust                    tli.robust
##                         0.863                         0.857
##                   nnfi.scaled                   nnfi.robust
##                         0.854                         0.857
##                    rfi.scaled                    nfi.scaled
##                         0.758                         0.768
##                    ifi.scaled                    rni.scaled
##                         0.861                         0.860
##                    rni.robust                          logl
##                         0.863                      -2613.096
##             unrestricted.logl                           aic
##                      -2543.961                      5304.193
##                           bic                        ntotal
##                       5442.149                       254.000
##                          bic2             scaling.factor.h1
##                       5318.510                         0.975
##             scaling.factor.h0                         rmsea
##                         0.621                         0.089
##                rmsea.ci.lower                rmsea.ci.upper
##                         0.067                         0.110
##                  rmsea.pvalue                  rmsea.scaled
##                         0.003                         0.095
##         rmsea.ci.lower.scaled         rmsea.ci.upper.scaled
##                         0.073                         0.117
##           rmsea.pvalue.scaled                  rmsea.robust
##                         0.001                         0.092
##         rmsea.ci.lower.robust         rmsea.ci.upper.robust
##                         0.072                         0.112
##           rmsea.pvalue.robust                           rmr
```

```
##                           NA                        0.113
##                    rmr_nomean                         srmr
##                         0.111                        0.096
##                  srmr_bentler           srmr_bentler_nomean
##                         0.096                        0.092
##                          crmr                   crmr_nomean
##                         0.142                        0.092
##                     srmr_mplus             srmr_mplus_nomean
##                         0.135                        0.094
##                         cn_05                        cn_01
##                       165.210                      183.279
##                           gfi                         agfi
##                         0.990                        0.985
##                          pgfi                          mfi
##                         0.633                        0.873
```

```
residualInvarianceModelFitIndices <- fitMeasures(residualInvarianceModel_fit)[c("cfi.robust", "rm
```

14.5.7.6 Compare Model Fit

14.5.7.6.1 Nested Model (χ^2) Difference Test The scalar invariance model and the residual invariance model are considered "nested" models. The residual invariance model is nested within the scalar invariance model because the scalar invariance model includes all of the terms of the residual invariance model along with additional terms. Model fit of nested models can be compared with a chi-square difference test, also known as a likelihood ratio test or deviance test. A significant chi-square difference test would indicate that the simplified model with additional constraints (the residual invariance model) is significantly worse fitting than the more complex model that has fewer constraints (the scalar invariance model).

```
anova(scalarInvarianceModel_fit, residualInvarianceModel_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                             Df    AIC    BIC  Chisq Chisq diff Df diff
## scalarInvarianceModel_fit   60 5303.9 5473.7 119.97
## residualInvarianceModel_fit 69 5304.2 5442.1 138.27      21.32       9
##                             Pr(>Chisq)
```

```
## scalarInvarianceModel_fit
## residualInvarianceModel_fit     0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this instance, you do not need to examine whether residual invariance held because the preceding level of measurement invariance (scalar invariance) did not hold.

```
round(residualInvarianceModelFitIndices - scalarInvarianceModelFitIndices, digits = 3)
```

14.5.7.6.2 Compare Other Fit Criteria

```
##   cfi.robust rmsea.robust         srmr
##       -0.020        0.001        0.006
```

14.5.7.6.3 Permutation Test Permutation procedures for testing measurement invariance are described in [Jorgensen et al., 2018]. For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed. You can specify the null model as the baseline model using: `baseline.model = nullModelFit`.

Warning: this code takes a while to run based on 1000 iterations. You can reduce the number of iterations to be faster.

```
residualInvarianceTest <- permuteMeasEq(nPermute = numPermutations,
                                        modelType = "mgcfa",
                                        con = residualInvarianceModel_fit,
                                        uncon = scalarInvarianceModel_fit,
                                        AFIs = myAFIs,
                                        moreAFIs = moreAFIs,
                                        iseed = 52242)
```

```
residualInvarianceTest
```

```
## Omnibus p value based on parametric chi-squared difference test:
##
## Chisq diff    Df diff Pr(>Chisq)
##     21.320      9.000      0.011
##
```

```
##
## Omnibus p values based on nonparametric permutation method:
##
##              AFI.Difference p.value
## chisq                18.304   0.033
## chisq.scaled         21.146   0.036
## rmsea                 0.000   0.122
## cfi                  -0.017   0.032
## tli                  -0.001   0.120
## srmr                  0.006   0.207
## rmsea.robust          0.001   0.120
## cfi.robust           -0.020   0.033
## tli.robust           -0.002   0.111
```

14.5.7.7 Reliability

Reliability of factor scores for the latent factors, as quantified by Cronbach's alpha (α) and omega (ω), was estimated using the `semTools` package [Jorgensen et al., 2021].

```
semTools::reliability(residualInvarianceModel_fit)
```

```
## $`2`
##           visual   textual     speed
## alpha  0.6524461 0.8841445 0.6430042
## omega  0.6414761 0.8918065 0.6830386
## omega2 0.6414761 0.8918065 0.6830386
## omega3 0.6110057 0.8854437 0.6866394
## avevar 0.3896054 0.7352841 0.4184035
##
## $`1`
##           visual   textual     speed
## alpha  0.4648456 0.8955126 0.7327296
## omega  0.4754752 0.8920077 0.6505292
## omega2 0.4754752 0.8920077 0.6505292
## omega3 0.5056994 0.8927268 0.6344718
## avevar 0.2443595 0.7356902 0.3832594
```

14.5.7.8 Path Diagram

Below is a path diagram of the model generated using the `semPlot` package [Epskamp, 2019].

```
semPaths(residualInvarianceModel_fit,
         what = "est",
         layout = "tree2",
         optimizeLatRes = FALSE,
         exoVar = FALSE,
         ask = FALSE,
         edge.label.cex = 1.2)
```

14.5.8 Addressing Measurement Non-Invariance

In the example above, we detected measurement non-invariance of intercepts, but not factor loadings. Thus, we would want to identify which item(s) show non-invariant intercepts across groups. When detecting measurement non-invariance in a given parameter (e.g., factor loadings, intercepts, or residuals), one can identify the specific items that show measurement non-invariance in one of two primary ways: 1) starting with a model that allows the given parameter to differ across groups, a researcher can iteratively add constraints to identify the item(s) for which measurement invariance fails, or 2) starting with a model that constrains the given parameter to be the same across groups, a researcher can iteratively remove constraints to identify the item(s) for which measurement invariance becomes established (and by process of elimination, the items for which measurement invariance does not become established).

14.5.8.1 Iteratively add constraints to identify measurement non-invariance

In the example above, we detected measurement non-invariance of intercepts. One approach to identify measurement non-invariance is to iteratively add constraints to a model that allows the given parameter (intercepts) to differ across groups. So, we will use the metric invariance model as the baseline model (in which item intercepts are allowed to differ across groups), and iteratively add constraints to identify which item(s) show non-invariant intercepts across groups.

14.5.8.1.1 Variable 1 Variable 1 does not have non-invariant intercepts

```
cfaModel_metricInvarianceV1 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9
```

Figure 14.38: Residual invariance model in confirmatory factor analysis

Figure 14.39: Residual invariance model in confirmatory factor analysis

```
 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV1_fit <- lavaan(cfaModel_metricInvarianceV1,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV1_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                 Df    AIC    BIC  Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4 81.456
## cfaModel_metricInvarianceV1_fit 55 5275.7 5463.1 81.743    0.29909       1
##                                 Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV1_fit     0.5845
```

14.5.8.1.2 Variable 2

Variable 2 has non-invariant intercepts.

```
cfaModel_metricInvarianceV2 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
```

```
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ c(intx1, intx1)*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV2_fit <- lavaan(cfaModel_metricInvarianceV2,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV2_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4  81.456
## cfaModel_metricInvarianceV2_fit 57 5390.4 5570.8 200.518     192.38       3
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV2_fit  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Variable 2 appears to have larger intercepts in group 2 than in group 1:

```
summary(metricInvarianceModel_fit)
```

```
## lavaan 0.6-8 ended normally after 81 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        63
##   Number of equality constraints                     9
##
##   Number of observations per group:
##     2                                              132
##     1                                              122
##   Number of missing patterns per group:
##     2                                               51
##     1                                               56
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                81.456      86.907
##   Degrees of freedom                                54          54
##   P-value (Chi-square)                           0.009       0.003
##   Scaling correction factor                                  0.937
##        Yuan-Bentler correction (Mplus variant)
##   Test statistic for each group:
##     2                                            43.580      46.496
##     1                                            37.876      40.411
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
##
## Group 1 [2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1      (l.1_)    1.093    0.151    7.242    0.000
##     x2      (l.2_)    0.483    0.121    4.008    0.000
##     x3      (l.3_)    0.710    0.102    6.970    0.000
##   textual =~
##     x4      (l.4_)    0.992    0.090   11.037    0.000
##     x5      (l.5_)    1.097    0.106   10.304    0.000
```

```
##     x6      (l.6_)    0.872    0.090    9.693    0.000
##   speed =~
##     x7      (l.7_)    0.602    0.085    7.084    0.000
##     x8      (l.8_)    0.671    0.090    7.490    0.000
##     x9      (l.9_)    0.587    0.108    5.441    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textul  (p.2_)    0.545    0.083    6.536    0.000
##     speed  (p.3_1)    0.411    0.130    3.156    0.002
##   textual ~~
##     speed  (p.3_2)    0.347    0.097    3.560    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1      (n.1.)    4.922    0.113   43.524    0.000
##    .x2      (n.2.)    6.024    0.119   50.541    0.000
##    .x3      (n.3.)    2.525    0.106   23.818    0.000
##    .x4      (n.4.)    2.837    0.109   26.090    0.000
##    .x5      (n.5.)    3.871    0.119   32.489    0.000
##    .x6      (n.6.)    1.876    0.088   21.269    0.000
##    .x7      (n.7.)    4.442    0.096   46.284    0.000
##    .x8      (n.8.)    5.586    0.092   60.978    0.000
##    .x9      (n.9.)    5.394    0.096   56.397    0.000
##     visual  (a.1.)    0.000
##     textual (a.2.)    0.000
##     speed   (a.3.)    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1      (t.1_)    0.320    0.230    1.388    0.165
##    .x2      (t.2_)    1.378    0.219    6.299    0.000
##    .x3      (t.3_)    0.857    0.139    6.168    0.000
##    .x4      (t.4_)    0.439    0.086    5.087    0.000
##    .x5      (t.5_)    0.448    0.095    4.736    0.000
##    .x6      (t.6_)    0.241    0.063    3.857    0.000
##    .x7      (t.7_)    0.757    0.126    6.003    0.000
##    .x8      (t.8_)    0.542    0.101    5.360    0.000
##    .x9      (t.9_)    0.663    0.144    4.605    0.000
##     visual  (p.1_)    1.000
##     textual (p.2_)    1.000
##     speed   (p.3_)    1.000
##
##
## Group 2 [1]:
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1      (l.1_)    1.093    0.151    7.242    0.000
##     x2      (l.2_)    0.483    0.121    4.008    0.000
##     x3      (l.3_)    0.710    0.102    6.970    0.000
##   textual =~
##     x4      (l.4_)    0.992    0.090   11.037    0.000
##     x5      (l.5_)    1.097    0.106   10.304    0.000
##     x6      (l.6_)    0.872    0.090    9.693    0.000
##   speed =~
##     x7      (l.7_)    0.602    0.085    7.084    0.000
##     x8      (l.8_)    0.671    0.090    7.490    0.000
##     x9      (l.9_)    0.587    0.108    5.441    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textul  (p.2_)    0.428    0.133    3.222    0.001
##     speed  (p.3_1)    0.343    0.152    2.264    0.024
##   textual ~~
##     speed  (p.3_2)    0.286    0.173    1.657    0.098
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1      (n.1.)    5.008    0.109   46.007    0.000
##    .x2      (n.2.)    6.263    0.104   60.369    0.000
##    .x3      (n.3.)    1.993    0.098   20.317    0.000
##    .x4      (n.4.)    3.280    0.108   30.417    0.000
##    .x5      (n.5.)    4.680    0.116   40.342    0.000
##    .x6      (n.6.)    2.505    0.107   23.426    0.000
##    .x7      (n.7.)    3.864    0.095   40.541    0.000
##    .x8      (n.8.)    5.452    0.085   63.893    0.000
##    .x9      (n.9.)    5.344    0.091   58.974    0.000
##     visual  (a.1.)    0.000
##     textual (a.2.)    0.000
##     speed   (a.3.)    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1      (t.1_)    0.794    0.231    3.437    0.001
##    .x2      (t.2_)    0.939    0.139    6.776    0.000
##    .x3      (t.3_)    0.755    0.142    5.309    0.000
##    .x4      (t.4_)    0.322    0.079    4.059    0.000
##    .x5      (t.5_)    0.322    0.078    4.098    0.000
```

```
##    .x6      (t.6_)    0.397    0.082    4.839    0.000
##    .x7      (t.7_)    0.538    0.110    4.878    0.000
##    .x8      (t.8_)    0.302    0.110    2.740    0.006
##    .x9      (t.9_)    0.517    0.119    4.350    0.000
##     visual  (p.1_)    0.420    0.183    2.289    0.022
##     textual (p.2_)    1.013    0.225    4.513    0.000
##     speed   (p.3_)    1.091    0.332    3.288    0.001
```

Thus, in subsequent models, we would allow variable 2 to have different intercepts across groups.

14.5.8.1.3 Variable 3 Variable 3 has non-invariant intercepts.

```
cfaModel_metricInvarianceV3 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9
```

```
 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ c(intx1, intx1)*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV3_fit <- lavaan(cfaModel_metricInvarianceV3,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV3_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4  81.456
## cfaModel_metricInvarianceV3_fit 57 5690.3 5870.7 500.409     222.92       3
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV3_fit  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.1.4 Variable 4 Variable 4 has non-invariant intercepts.

```
cfaModel_metricInvarianceV4 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9
```

```
 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ c(intx1, intx1)*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV4_fit <- lavaan(cfaModel_metricInvarianceV4,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV4_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4  81.456
## cfaModel_metricInvarianceV4_fit 57 5541.2 5721.6 351.231     197.98       3
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV4_fit  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.1.5 Variable 5 Variable 5 has non-invariant intercepts.

```
cfaModel_metricInvarianceV5 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
```

```
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ c(intx1, intx1)*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV5_fit <- lavaan(cfaModel_metricInvarianceV5,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV5_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4  81.456
## cfaModel_metricInvarianceV5_fit 57 5338.8 5519.2 148.923     69.762       3
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV5_fit  4.799e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.1.6 Variable 6 Variable 6 has non-invariant intercepts.

```
cfaModel_metricInvarianceV6 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ c(intx1, intx1)*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV6_fit <- lavaan(cfaModel_metricInvarianceV6,
```

```
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV6_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4  81.456
## cfaModel_metricInvarianceV6_fit 57 5699.4 5879.8 509.521      236.9       3
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV6_fit  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.1.7 Variable 7 Variable 7 has non-invariant intercepts.

```
cfaModel_metricInvarianceV7 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
```

```
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ c(intx1, intx1)*1
 x8 ~ NA*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV7_fit <- lavaan(cfaModel_metricInvarianceV7,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV7_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                 Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4  81.456
## cfaModel_metricInvarianceV7_fit 57 5349.9 5530.4 160.025     66.451       3
##                                 Pr(>Chisq)
## metricInvarianceModel_fit
```

```
## cfaModel_metricInvarianceV7_fit  2.454e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.1.8 Variable 8 Variable 8 has non-invariant intercepts.

```
cfaModel_metricInvarianceV8 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
```

```
 x7 ~ NA*1
 x8 ~ c(intx1, intx1)*1
 x9 ~ NA*1
'

cfaModel_metricInvarianceV8_fit <- lavaan(cfaModel_metricInvarianceV8,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV8_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4  81.456
## cfaModel_metricInvarianceV8_fit 57 5308.0 5488.4 118.090     52.587       3
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV8_fit  2.245e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.1.9 Variable 9 Variable 9 has non-invariant intercepts.

```
cfaModel_metricInvarianceV9 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
```

```
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively fix intercepts of manifest variables across groups
 x1 ~ c(intx1, intx1)*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ c(intx1, intx1)*1
'

cfaModel_metricInvarianceV9_fit <- lavaan(cfaModel_metricInvarianceV9,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_metricInvarianceV9_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
```

```
##
##                                Df    AIC    BIC  Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4 81.456
## cfaModel_metricInvarianceV9_fit 57 5289.7 5470.1 99.817     22.151       3
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_metricInvarianceV9_fit  6.067e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.1.10 Summary When iteratively adding constraints, item 1 showed measurement invariance of intercepts, but all other items showed measurement non-invariance of intercepts. This could be due to the selection of starting with item 1 as the first constraint. We could try applying constraints in a different order to see the extent to which the order influences which items are identified as showing measurement non-invariance.

14.5.8.2 Iteratively drop constraints to identify measurement non-invariance

Another approach to identify measurement non-invariance is to iteratively drop constraints to a model that constrains the given parameter (intercepts) to be the same across groups. So, we will use the scalar invariance model as the baseline model (in which item intercepts are constrained to be the same across groups), and we will iteratively drop constraints to identify which the items for which measurement invariance becomes established (relative to the metric invariance model which allows item intercepts to differ across groups), and therefore, identifies which item(s) show non-invariant intercepts across groups.

14.5.8.2.1 Variable 1 Measurement invariance does not yet become established when freeing intercepts of variable 1 across groups. This suggests that, whether or not variable 1 shows non-invariant intercepts, there are other items that show non-invariant intercepts.

```
cfaModel_scalarInvarianceV1 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0
```

```
 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ c(intx2, intx2)*1
 x3 ~ c(intx3, intx3)*1
 x4 ~ c(intx4, intx4)*1
 x5 ~ c(intx5, intx5)*1
 x6 ~ c(intx6, intx6)*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV1_fit <- lavaan(cfaModel_scalarInvarianceV1,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV1_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
```

```
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                 Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4  81.456
## cfaModel_scalarInvarianceV1_fit 62 5335.3 5498.0 155.352      72.63       8
##                                 Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV1_fit  1.469e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.2.2 Variable 2 Measurement invariance does not yet become established when freeing intercepts of variable 2 across groups. This suggests that, whether or not variables 1 and 2 show non-invariant intercepts, there are other items that show non-invariant intercepts.

```
cfaModel_scalarInvarianceV2 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
```

```
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ c(intx3, intx3)*1
 x4 ~ c(intx4, intx4)*1
 x5 ~ c(intx5, intx5)*1
 x6 ~ c(intx6, intx6)*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV2_fit <- lavaan(cfaModel_scalarInvarianceV2,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV2_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4  81.456
## cfaModel_scalarInvarianceV2_fit 61 5335.0 5501.2 153.031     71.976       7
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV2_fit  5.884e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.2.3 Variable 3 Measurement invariance does not yet become established when freeing intercepts of variable 3 across groups. This suggests that, whether or not variables 1, 2, and 3 show non-invariant intercepts, there are other items that show non-invariant intercepts.

```
cfaModel_scalarInvarianceV3 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ c(intx4, intx4)*1
 x5 ~ c(intx5, intx5)*1
 x6 ~ c(intx6, intx6)*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV3_fit <- lavaan(cfaModel_scalarInvarianceV3,
```

```
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV3_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4  81.456
## cfaModel_scalarInvarianceV3_fit 60 5316.4 5486.2 132.449     51.848       6
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV3_fit  2.001e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.2.4 Variable 4 Measurement invariance does not yet become established when freeing intercepts of variable 4 across groups. This suggests that, whether or not variables 1, 2, 3, and 4 show non-invariant intercepts, there are other items that show non-invariant intercepts.

```
cfaModel_scalarInvarianceV4 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed
```

```
 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ c(intx5, intx5)*1
 x6 ~ c(intx6, intx6)*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV4_fit <- lavaan(cfaModel_scalarInvarianceV4,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV4_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                 Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4  81.456
```

```
## cfaModel_scalarInvarianceV4_fit 59 5317.2 5490.5 131.249     49.035       5
##                                 Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV4_fit  2.183e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.2.5 Variable 5 Measurement invariance does not yet become established when freeing intercepts of variable 5 across groups. This suggests that, whether or not variables 1, 2, 3, 4, and 5 show non-invariant intercepts, there are other items that show non-invariant intercepts.

```
cfaModel_scalarInvarianceV5 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
```

```
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ c(intx6, intx6)*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV5_fit <- lavaan(cfaModel_scalarInvarianceV5,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV5_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4  81.456
## cfaModel_scalarInvarianceV5_fit 58 5313.2 5490.0 125.250      43.05       4
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV5_fit   1.01e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.2.6 Variable 6 Measurement invariance does not yet become established when freeing intercepts of variable 6 across groups. This suggests that, whether or not variables 1, 2, 3, 4, 5, and 6 show non-invariant intercepts, there are other items that show non-invariant intercepts.

```
cfaModel_scalarInvarianceV6 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
```

```
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ c(intx7, intx7)*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV6_fit <- lavaan(cfaModel_scalarInvarianceV6,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")
```

```
anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV6_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                 Df    AIC    BIC   Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4  81.456
## cfaModel_scalarInvarianceV6_fit 57 5290.1 5470.5 100.156      19.02       3
##                                 Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV6_fit  0.0002708 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

14.5.8.2.7 Variable 7 Measurement invariance becomes established when freeing intercepts of variable 7 across groups. This suggests that item 7 shows non-invariant intercepts across groups, and that measurement invariance holds when constraining items 8 and 9 to be invariant across groups. There may also be other items, especially among the preceding items (variables 1–6), that show non-invariant intercepts across groups.

```
cfaModel_scalarInvarianceV7 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
```

```
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV7_fit <- lavaan(cfaModel_scalarInvarianceV7,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV7_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC  Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4 81.456
## cfaModel_scalarInvarianceV7_fit 56 5274.5 5458.5 82.609     1.1536       2
##                                Pr(>Chisq)
## metricInvarianceModel_fit
```

```
## cfaModel_scalarInvarianceV7_fit     0.5617
```

14.5.8.2.8 Variable 8 Measurement invariance continues to hold when freeing intercepts of variable 8 across groups. This suggests that variable 8 does not show invariant intercepts across groups, at least when the other intercepts have been freed.

```
cfaModel_scalarInvarianceV8 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0

 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
```

```
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ NA*1
 x9 ~ c(intx9, intx9)*1
'

cfaModel_scalarInvarianceV8_fit <- lavaan(cfaModel_scalarInvarianceV8,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV8_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df    AIC    BIC  Chisq Chisq diff Df diff
## metricInvarianceModel_fit      54 5277.4 5468.4 81.456
## cfaModel_scalarInvarianceV8_fit 55 5275.5 5463.0 81.601    0.14607       1
##                                Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV8_fit     0.7023
```

14.5.8.2.9 Variable 9 Measurement invariance continues to hold when freeing intercepts of variable 9 across groups (even after constraining intercepts of variable 8 across groups). This suggests that variable 9 does not show invariant intercepts across groups, at least when the other intercepts have been freed.

```
cfaModel_scalarInvarianceV9 <- '
 #Fix factor loadings to be the same across groups
 visual  =~ c(lambdax1,lambdax1)*x1 + c(lambdax2,lambdax2)*x2 + c(lambdax3,lambdax3)*x3
 textual =~ c(lambdax4,lambdax4)*x4 + c(lambdax5,lambdax5)*x5 + c(lambdax6,lambdax6)*x6
 speed   =~ c(lambdax7,lambdax7)*x7 + c(lambdax8,lambdax8)*x8 + c(lambdax9,lambdax9)*x9

 #Fix latent means to zero
 visual ~ 0
 textual ~ 0
 speed ~ 0
```

```
 #Fix latent variances to one in group 1; free latent variances in group 2
 visual ~~ c(1, NA)*visual
 textual ~~ c(1, NA)*textual
 speed ~~ c(1, NA)*speed

 #Estimate covariances among latent variables
 visual ~~ textual
 visual ~~ speed
 textual ~~ speed

 #Estimate residual variances of manifest variables
 x1 ~~ x1
 x2 ~~ x2
 x3 ~~ x3
 x4 ~~ x4
 x5 ~~ x5
 x6 ~~ x6
 x7 ~~ x7
 x8 ~~ x8
 x9 ~~ x9

 #Iteratively free intercepts across groups
 x1 ~ NA*1
 x2 ~ NA*1
 x3 ~ NA*1
 x4 ~ NA*1
 x5 ~ NA*1
 x6 ~ NA*1
 x7 ~ NA*1
 x8 ~ c(intx8, intx8)*1
 x9 ~ NA*1
'

cfaModel_scalarInvarianceV9_fit <- lavaan(cfaModel_scalarInvarianceV9,
               data = HolzingerSwineford1939,
               group = "school",
               missing = "ML",
               estimator = "MLR")

anova(metricInvarianceModel_fit, cfaModel_scalarInvarianceV9_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
```

```
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                 Df    AIC    BIC  Chisq Chisq diff Df diff
## metricInvarianceModel_fit       54 5277.4 5468.4 81.456
## cfaModel_scalarInvarianceV9_fit 55 5276.5 5464.0 82.607     1.2013       1
##                                 Pr(>Chisq)
## metricInvarianceModel_fit
## cfaModel_scalarInvarianceV9_fit     0.2731
```

14.5.8.2.10 Summary When iteratively dropping constraints, measurement invariance became established when freeing the intercepts for items 1–7 across groups. This suggests that items 7 and 8 show measurement invariance of intercepts across groups. This could be due to the selection of starting with item 1 as the first constraint to be freed. We could try freeing constraints in a different order to see the extent to which the order influences which items are identified as showing measurement non-invariance.

14.5.8.3 Next Steps

After identifiying which item(s) show measurement non-invariance, we would evaluate the measurement non-invariance according to theory and effect sizes. We might start with the item with the largest measurement non-invariance. If we deem the measurement non-invariance for this item is non-negligible, we have three primary options: 1) drop the for both groups, 2) drop the item for one group but keep it for the other group, or 3) freely estimate parameters for the item across groups. We would do this iteratively for the remaining items, by magnitude, that show non-negligible measurement non-invariance. Thus, our model might show partial scalar invariance—some items might have intercepts that are constrained to be the same across groups, whereas other items might have intercepts that are allowed to differ across groups. Then, we would test subsequent measurement invariance models (e.g., residual invariance) while keeping the partially freed constaints from the partial scalar invairance model. Once we establish the best fitting model that makes as many constraints and theoretically and empirically justified for the purposes of the study, we would use that model in subsequent tests. As a reminder, full or partial metric invariance is helpful for comparing associations across groups, whereas full or partial scalar invariance is helpful for comparing mean levels across groups.

14.6 Conclusion

14.7 Exercises

14.7.1 Questions

Note: several of the following questions use data from the Children of the National Longitudinal Survey of Youth Survey (CNLSY). The CNLSY is a publicly available longitudinal dataset provided by the Bureau of Labor Statistics (https://www.bls.gov/nls/nlsy79-children.htm#topical-guide). The CNLSY data file for these exercises is located on the course's page of the Open Science Framework (https://osf.io/3pwza). Children's behavior problems were rated in 1988 (time 1: T1) and then again in 1990 (time 2: T2) on the Behavior Problems Index (BPI). Below are the items corresponding to the Antisocial subscale of the BPI: 1) cheats or tells lies 2) bullies or is cruel/mean to others 3) does not seem to feel sorry after misbehaving 4) breaks things deliberately 5) is disobedient at school 6) has trouble getting along with teachers 7) has sudden changes in mood or feeling

1. [different types of bias]
2. [different ways to evaluate fairness]
3. [fairness vs bias]
4. [DIF]
5. [measurement invariance]

14.7.2 Answers

Chapter 15

The Interview and the DSM

[**Articles on pfactor: Caspi; Smith et al, 2020**]

[**Articles on network analysis: McNally, 2021**]

15.1 Conclusion

Chapter 16

Objective Personality Testing

[**Wright et al, 2019 citation in psych assessment**; **Bensch et al, 2019 citation** ;**Tackett et al, 2019 citation on MTMM**; **Simms et al. 2019 citation**; **Clark & Watson, 2019 citation**; **Sellbom & Tellegen, 2019**; **Wright & Zimmerman, 2019 citation**; **Burchett et al, 2019 citation**; **Chandler et al, 2020 citation**]

[**Costa et al, 2019 citation**]

[**Markon, 2019 article on bifactor models**]

16.1 Conclusion

Chapter 17

Projective Personality Testing

17.1 Conclusion

Chapter 18

Psychophysiological Measures

[**Calamia, 2019 citation**; **Holmlund et al, 2019 citation**, **Raugh et al, 2019 citation**, **Roberts et al, 2019, Psych Assessment citation on VR**: https://psycnet.apa.org/PsycARTICLES/journal/pas/31/3]

[**Himmelstein et al, 2019 citation**; **Patrick et al, 2019 citation**; **Stanton et al, 2020 citation**; **Trull & Ebner-Priemer citation**]

18.1 Conclusion

Chapter 19

Computers and Adaptive Testing

[**Granziol et al, in press citation Assessment**; **Lee Meeuw Kjoe et al, in press citation**]

19.1 Getting Started

19.1.1 Load Libraries

```
library("mirtCAT")
library("tinytex")
```

19.2 Example of Uni-dimensional CAT

The computerized adaptive test (CAT) was fit using the `mirtCAT` package [Chalmers, 2020b]. The example of a CAT with uni-dimensional data is adapted from `mirtCAT` documentation: http://philchalmers.github.io/mirtCAT/html/unidim-exampleGUI.html

19.2.1 Define population IRT parameters

For reproducibility, I set the seed below. Using the same seed will yield the same answer every time. There is nothing special about this particular seed.

```
set.seed(52242)
nitems <- 100
itemnames <- paste("Item.", 1:nitems, sep = "")
a <- matrix(rlnorm(nitems, meanlog = .2, sdlog = .3))
d <- matrix(rnorm(nitems, mean = 0, sd = 1))
pars <- data.frame(a1 = a, d = d, g = 0.2)
```

19.2.2 Fit IRT model

```
mod <- generate.mirt_object(pars, "3PL")
```

19.2.3 Model Summary

```
summary(mod)
```

```
##             F1     h2
## Item.1   0.483 0.2331
## Item.2   0.374 0.1401
## Item.3   0.557 0.3099
## Item.4   0.491 0.2407
## Item.5   0.454 0.2057
## Item.6   0.483 0.2332
## Item.7   0.515 0.2654
## Item.8   0.590 0.3475
## Item.9   0.549 0.3014
## Item.10  0.595 0.3539
## Item.11  0.616 0.3799
## Item.12  0.571 0.3265
## Item.13  0.516 0.2659
## Item.14  0.760 0.5777
## Item.15  0.536 0.2870
## Item.16  0.503 0.2530
## Item.17  0.723 0.5233
## Item.18  0.677 0.4587
## Item.19  0.786 0.6179
## Item.20  0.590 0.3480
## Item.21  0.449 0.2014
## Item.22  0.589 0.3469
## Item.23  0.794 0.6297
## Item.24  0.580 0.3363
```

```
## Item.25  0.600 0.3601
## Item.26  0.507 0.2569
## Item.27  0.705 0.4969
## Item.28  0.620 0.3842
## Item.29  0.603 0.3640
## Item.30  0.615 0.3780
## Item.31  0.442 0.1950
## Item.32  0.437 0.1913
## Item.33  0.502 0.2524
## Item.34  0.541 0.2926
## Item.35  0.815 0.6640
## Item.36  0.784 0.6139
## Item.37  0.666 0.4432
## Item.38  0.598 0.3571
## Item.39  0.631 0.3978
## Item.40  0.701 0.4912
## Item.41  0.482 0.2321
## Item.42  0.618 0.3819
## Item.43  0.702 0.4929
## Item.44  0.799 0.6385
## Item.45  0.662 0.4383
## Item.46  0.466 0.2169
## Item.47  0.482 0.2328
## Item.48  0.629 0.3952
## Item.49  0.381 0.1448
## Item.50  0.594 0.3530
## Item.51  0.414 0.1711
## Item.52  0.741 0.5489
## Item.53  0.527 0.2777
## Item.54  0.507 0.2575
## Item.55  0.425 0.1804
## Item.56  0.684 0.4674
## Item.57  0.496 0.2463
## Item.58  0.726 0.5266
## Item.59  0.769 0.5909
## Item.60  0.700 0.4907
## Item.61  0.602 0.3627
## Item.62  0.462 0.2137
## Item.63  0.662 0.4382
## Item.64  0.544 0.2959
## Item.65  0.769 0.5919
## Item.66  0.599 0.3589
## Item.67  0.636 0.4047
## Item.68  0.490 0.2401
## Item.69  0.715 0.5112
## Item.70  0.642 0.4122
```

```
## Item.71  0.420 0.1766
## Item.72  0.417 0.1735
## Item.73  0.309 0.0954
## Item.74  0.603 0.3641
## Item.75  0.520 0.2706
## Item.76  0.609 0.3703
## Item.77  0.697 0.4852
## Item.78  0.665 0.4426
## Item.79  0.753 0.5670
## Item.80  0.706 0.4988
## Item.81  0.476 0.2265
## Item.82  0.585 0.3420
## Item.83  0.585 0.3421
## Item.84  0.584 0.3411
## Item.85  0.638 0.4075
## Item.86  0.641 0.4114
## Item.87  0.669 0.4472
## Item.88  0.626 0.3916
## Item.89  0.618 0.3823
## Item.90  0.572 0.3270
## Item.91  0.510 0.2601
## Item.92  0.724 0.5236
## Item.93  0.659 0.4349
## Item.94  0.505 0.2548
## Item.95  0.615 0.3781
## Item.96  0.601 0.3610
## Item.97  0.765 0.5846
## Item.98  0.469 0.2199
## Item.99  0.755 0.5696
## Item.100 0.480 0.2309
##
## SS loadings:  36.315
## Proportion Var:  0.363
##
## Factor correlations:
##
##    F1
## F1  1
```

```
coef(mod, simplify = TRUE, IRTpars = TRUE)
```

```
## $items
##              a      b   g u
## Item.1   0.938  0.165 0.2 1
## Item.2   0.687  1.157 0.2 1
```

```
## Item.3   1.140 -0.768 0.2 1
## Item.4   0.958 -1.138 0.2 1
## Item.5   0.866  0.105 0.2 1
## Item.6   0.939  1.212 0.2 1
## Item.7   1.023  1.854 0.2 1
## Item.8   1.242  0.461 0.2 1
## Item.9   1.118  0.576 0.2 1
## Item.10  1.260 -0.461 0.2 1
## Item.11  1.332  1.277 0.2 1
## Item.12  1.185  0.085 0.2 1
## Item.13  1.024 -0.127 0.2 1
## Item.14  1.990 -1.220 0.2 1
## Item.15  1.080  0.891 0.2 1
## Item.16  0.991 -0.937 0.2 1
## Item.17  1.783 -1.119 0.2 1
## Item.18  1.567  1.604 0.2 1
## Item.19  2.164 -0.428 0.2 1
## Item.20  1.244 -0.163 0.2 1
## Item.21  0.855 -2.272 0.2 1
## Item.22  1.241  0.018 0.2 1
## Item.23  2.219  0.165 0.2 1
## Item.24  1.211  0.795 0.2 1
## Item.25  1.277 -0.595 0.2 1
## Item.26  1.001  0.676 0.2 1
## Item.27  1.692  0.020 0.2 1
## Item.28  1.344 -0.196 0.2 1
## Item.29  1.288 -0.439 0.2 1
## Item.30  1.327 -0.141 0.2 1
## Item.31  0.838 -0.876 0.2 1
## Item.32  0.828  0.123 0.2 1
## Item.33  0.989 -0.746 0.2 1
## Item.34  1.095 -0.594 0.2 1
## Item.35  2.393 -0.447 0.2 1
## Item.36  2.146  0.437 0.2 1
## Item.37  1.519  0.499 0.2 1
## Item.38  1.269  0.327 0.2 1
## Item.39  1.383  0.992 0.2 1
## Item.40  1.672  0.720 0.2 1
## Item.41  0.936  1.007 0.2 1
## Item.42  1.338 -0.207 0.2 1
## Item.43  1.678 -0.608 0.2 1
## Item.44  2.262  0.928 0.2 1
## Item.45  1.503  1.118 0.2 1
## Item.46  0.896 -0.338 0.2 1
## Item.47  0.937 -0.601 0.2 1
## Item.48  1.376  0.514 0.2 1
```

```
## Item.49  0.700  2.349 0.2 1
## Item.50  1.257 -0.675 0.2 1
## Item.51  0.773 -1.738 0.2 1
## Item.52  1.877 -0.793 0.2 1
## Item.53  1.055 -0.508 0.2 1
## Item.54  1.002 -0.855 0.2 1
## Item.55  0.798 -0.678 0.2 1
## Item.56  1.595  0.474 0.2 1
## Item.57  0.973 -0.201 0.2 1
## Item.58  1.795 -0.427 0.2 1
## Item.59  2.046 -0.460 0.2 1
## Item.60  1.670  0.655 0.2 1
## Item.61  1.284 -0.889 0.2 1
## Item.62  0.887  0.843 0.2 1
## Item.63  1.503 -1.005 0.2 1
## Item.64  1.103 -1.380 0.2 1
## Item.65  2.050  0.289 0.2 1
## Item.66  1.274  1.156 0.2 1
## Item.67  1.403 -0.586 0.2 1
## Item.68  0.957 -1.629 0.2 1
## Item.69  1.741  0.711 0.2 1
## Item.70  1.425  0.379 0.2 1
## Item.71  0.788  0.048 0.2 1
## Item.72  0.780  0.100 0.2 1
## Item.73  0.553 -2.557 0.2 1
## Item.74  1.288  0.156 0.2 1
## Item.75  1.037  0.113 0.2 1
## Item.76  1.305 -0.076 0.2 1
## Item.77  1.652 -0.153 0.2 1
## Item.78  1.517  0.645 0.2 1
## Item.79  1.947 -0.039 0.2 1
## Item.80  1.698 -0.532 0.2 1
## Item.81  0.921  0.859 0.2 1
## Item.82  1.227 -0.030 0.2 1
## Item.83  1.227  0.316 0.2 1
## Item.84  1.225 -1.201 0.2 1
## Item.85  1.411 -0.725 0.2 1
## Item.86  1.423 -0.319 0.2 1
## Item.87  1.531 -0.457 0.2 1
## Item.88  1.365  0.741 0.2 1
## Item.89  1.339  1.596 0.2 1
## Item.90  1.186 -0.187 0.2 1
## Item.91  1.009 -0.499 0.2 1
## Item.92  1.784 -0.046 0.2 1
## Item.93  1.493 -0.302 0.2 1
## Item.94  0.995  0.126 0.2 1
```

```
## Item.95  1.327  1.568 0.2 1
## Item.96  1.279 -0.199 0.2 1
## Item.97  2.019 -0.611 0.2 1
## Item.98  0.904  0.673 0.2 1
## Item.99  1.958 -0.079 0.2 1
## Item.100 0.933  0.953 0.2 1
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
```

```
modItemParameters <- coef(mod, simplify = TRUE, IRTpars = TRUE)$items
```

19.2.4 Model Plots

The test appears to measure theta (θ) with $SE < .4$ within -2 to 2 range

```
plot(mod)
```

```
plot(mod, type = "infoSE", theta_lim = c(-3, 3))
```

19.2.5 Item Plots

```
plot(mod, type = "trace", theta_lim = c(-3, 3))
```

```
plot(mod, type = "infotrace", theta_lim = c(-3, 3))
```

- Item 30 would be a good starting item; it has the difficulty closest to the 50th percentile ($b = -0.141$) and a high discrimination ($a = 1.327$)
- Item 70 would be a good starting item; it has a difficulty closest to the 50th percentile ($b = 0.379$) and a high discrimination ($a = 1.425$)
- Item 86 is an easy item ($b = -0.319$)
- Item 16 is a difficult item ($b = -0.937$)
- Item 53 has a low discrimination ($a = 1.055$)

Figure 19.1: Test characteristic curve

```
itemplot(object = mod, item = "Item.30", "infotrace")
```

```
itemplot(object = mod, item = "Item.70", "infotrace")
```

```
itemplot(object = mod, item = "Item.86", "infotrace")
```

```
itemplot(object = mod, item = "Item.16", "infotrace")
```

```
itemplot(object = mod, item = "Item.53", "infotrace")
```

19.2.6 Create math items

```
questions <- answers <- character(nitems)
choices <- matrix("a", nitems, 5)
```

Figure 19.2: Test information and standard error of measurement

```
spacing <- floor(d - min(d)) + 1 #easier items have more variation

for(i in 1:nitems){
  n1 <- sample(1:100, 1)
  n2 <- sample(101:200, 1)
  ans <- n1 + n2
  questions[i] <- paste(n1, " + ", n2, " = ?", sep = "")
  answers[i] <- as.character(ans)
  ch <- ans + sample(c(-5:-1, 1:5) * spacing[i,], 5)
  ch[sample(1:5, 1)] <- ans
  choices[i,] <- as.character(ch)
}

df <- data.frame(Questions = questions, Answer = answers, Option = choices, Type = "radio")
```

19.2.7 Run computerized adaptive test (CAT)

Set the minimum standard error of measurement for the latent trait (theta; θ) that must be reached before stopping the CAT. You can lengthen the CAT by

Figure 19.3: Item characteristic curves

lowering the minimum standard error of measurement, or can shorten the CAT by raising the minimum standard error of measurement.

```
minimum_SEM <- .6
```

Run the CAT and stop once the standard error of measurement for the latent trait (theta; θ) becomes 0.6 or lower.

```
result <- mirtCAT(df, mod, start_item = "MI", method = "EAP", criteria = "MI", design =
```

19.2.8 CAT results

```
print(result)
```

```
##  n.items.answered Theta_1 SE.Theta_1
##                 0       0          1
```

Figure 19.4: Item information functions

```
summary(result)
```

```
## $final_estimates
##           Theta_1
## Estimates       0
## SEs             1
##
## $raw_responses
## character(0)
##
## $items_answered
## integer(0)
##
## $terminated_sucessfully
## [1] FALSE
```

Figure 19.5: Item characteristic curves and information functions

19.2.8.1 CAT Standard Errors

```
plot(result, SE = 1)
```

```
## Error: plot not available for non-adaptive tests
```

19.2.8.2 CAT 95% Confidence Interval

```
plot(result, SE = qnorm(.975))
```

```
## Error: plot not available for non-adaptive tests
```

19.3 Conclusion

19.4 Exercises

Figure 19.6: Item characteristic curves and information functions

19.4.1 Questions

19.4.2 Answers

Figure 19.7: Item characteristic curves and information functions

Figure 19.8: Item characteristic curves and information functions

Figure 19.9: Item characteristic curves and information functions

Chapter 20

Behavioral Assessment

20.1 Conclusion

Chapter 21

Repeated Assessments across Time

[**Wright & Woods citation on personalized modeling of psychopathology**]

21.1 Types of Research Designs

Schaie [1965], Baltes [1968], and Schaie and Baltes [1975] proposed various types of research designs, in terms of the various combinations of age, period (i.e., time of measurement), and cohort. Age refers to a person's chronological age at the time of measurement. Period refers to the time of measurement (e.g., April 12, 2014 at 7:53 AM). Cohort refers to a person' year of birth (e.g., 2003). If you know two of these (e.g., age and period), you can determine the third (e.g., cohort). Here are the types of research designs based on combinations of age, period, and cohort: [**CITE LITTLE; SCHAIE; BALTES; MASCHE**]

- cross-sectional
- cross-sectional sequences
- longitudinal
- longitudinal sequences
- time-lag
- time-sequential
- cross-sequential
- cohort-sequential

These research designs are depicted by age and cohort in Figure 21.1 and by time of measurement and cohort in Figure 21.2. Although the depictions of

cohort-sequential and cross-sequential designs differ from the depictions by Schaie [2005], they are consistent with contemporary definitions of these designs [Little, 2013, Masche and van Dulmen, 2004, Whitbourne, 2019].

21.1.1 Cross-sectional design

A *cross-sectional* design involves multiple participants, often spanning different ages, assessed at one time of measurement. A cross-sectional design is a single factor design, where the researcher is interested in comparing effects of one factor: age. However, in a cross-sectional design, cohort (i.e., birth year) is confounded with age differences. Thus, observed age-related differences could reflect cohort differences rather than true age differences. Additionally, sampling differences at each age could yield non-comparable age groups.

21.1.2 Cross-sectional sequences design

A *cross-sectional sequences* design involves successive cross-sectional studies of different participants at different times of measurement.

21.1.3 Longitudinal design

A *longitudinal* design involves following the same participants over time. The single-cohort longitudinal design is a single factor design: the researcher specifies one factor: age. However, in a single-cohort longitudinal design, cohort and time of measurement are confounded with age. Thus, any observed changes with age could be due to time-of-measurement effects or cohort effects (instead of people's true change).

21.1.4 Longitudinal sequences design

A *longitudinal sequences* design involves following multiple cohorts across time.

21.1.5 Time-lag design

A time-lag design involves participants from different cohorts assessed at the same age. The time-lag design is a single factor design: the researcher specifies one factor: cohort. However, time of measurement is confounded with cohort. Thus, any cohort differences could reflect different times of measurement.

Figure 21.1: Research designs by age and cohort. Values in the cells are the times (years) of measurement. Dashed line indicates different participants were assessed at each time of measurement.

Figure 21.2: Research designs by time of measurement and cohort. Values in the cells are ages of the participants. Dashed line indicates different participants were assessed at each time of measurement.

21.1.6 Time-sequential design

A *time-sequential* design is depicted in Figure 21.3. In a time-sequential design, multiple age groups are assessed at multiple times [Whitbourne, 2019]. It is a repeated cross-sectional design, with some participants followed longitudinally. Additional age groups are added to a time-lag design, with some of the participants assessed at more than time point. In other words, the age range is kept the same and is repeatedly assessed (with only some participants being repeatedly assessed). The two factors a researcher specifies in a time-sequential design are age and time of measurement. A time-sequential design can be helpful for identifying age differences while controlling for time of measurement. However, cohort effects are confounded with the interaction of age and time of measurement [Whitbourne, 2019]. Thus, observed differences as a function of age or time of measurement could reflect cohort effects.

Age of Participant	Time (year) of Measurement	
	2030	2040
5 years old	2025	2035
10 years old	2020	2030
	Cohort (birth year)	

Figure 21.3: Time-sequential research design.

21.2 Cross-sequential design

A *cross-sequential* design is depicted in Figure 21.4. In a cross-sequential design, multiple cohorts are assessed at multiple times [Whitbourne, 2019]. It is is a cross of a cross-sectional and longitudinal design. It starts as a cross-sectional study with participants from multiple cohorts, and then all participants are followed longitudinally (typically across the same duration). It is also called an accelerated longitudinal design. The two factors a researcher specifies in a cross-sequential design are time of measurement and cohort. A cross-sequential design can be helpful for identifying cohort differences while controlling for time of measurement. However, age differences are confounded with the interaction between cohort and time of measurement.

Thus, observed differences as a function of cohort or time of measurement could reflect age effects.

Time (year) of Measurement	Cohort (birth year)	
	2010	2020
2030	20 years old	10 years old
2040	30 years old	20 years old
	Age of Participant	

Figure 21.4: Cross-sequential research design.

21.2.1 Cohort-sequential design

A *cohort-sequential* design is depicted in Figure 21.5. In a cohort-sequential design, multiple cohorts are assessed at multiple ages [Whitbourne, 2019]. It starts multiple cohorts at the same age and then follows them longitudinally (typically across the same duration). It is like starting a longitudinal study at the same age over and over again. The two factors a researcher specifies in a cohort-sequential design are age and cohort. A cohort-sequential design can be helpful for identifying age differences while controlling for cohort differences. However, time-of-measurement effects are confounded with the interaction of age and cohort [Whitbourne, 2019]. Thus, observed differences as a function of age or cohort could reflect time-of-measurement effects.

21.3 Using Sequential Designs To Make Developmental Inferences

To have greater confidence that age-related differences reflect true change (development) rather than effects of cohort or time of measurement, we would conduct all three sequential designs. To the extent that the age-related effects in the time-sequential and cohort-sequential designs are stronger than the cohort- and time-of-measurement effects in the cross-sequential design, we have confidence that the observed age-related differences reflect development [Whitbourne, 2019]. Moreover, there is an entire set of analytic approaches,

Age of Participant	Cohort (birth year)	
	2020	2030
5 years old	2025	2035
10 years old	2030	2040
	Time (year) of Measurement	

Figure 21.5: Cohort-sequential research design.

known as age-period-cohort analysis [Yang and Land, 2013], whose goal is to disentangle the effects of age, period, and cohort.

21.4 Conclusion

Chapter 22

Assessment of Cognition

22.1 Conclusion

Chapter 23

Cultural and Individual Diversity

Like ethics, considerations of cultural and individual diversity are relevant to all of our domains as psychologists, including (but not limited to) research, teaching, assessment, intervention, and supervision. As a result, I have discussed important considerations about diversity throughout the book. At the same time, given the importance of cultural and individual diversity, I also provide this section, which is devoted more fully to these issues.

[**Han et al, 2019 citation**]

See Figure 23.1.

See Figure 23.2.

23.1 ADDRESSING

Remember the acryonym *ADDRESSING* [Hays, 2016], which stands for:

- **A**ge and generational differences
- **D**evelopmental or other **D**isability
- **R**eligion and spiritual orientation
- **E**thnic and racial identity
- **S**ocioeconomic status
- **S**exual orientation
- **I**ndigenous heritage
- **N**ational origin
- **G**ender

➜ **NOTE: Please answer BOTH Question 5 about Hispanic origin and Question 6 about race. For this survey, Hispanic origins are not races.**

5 Is Person 1 of Hispanic, Latino, or Spanish origin?

- ☐ **No,** not of Hispanic, Latino, or Spanish origin
- ☐ Yes, Mexican, Mexican Am., Chicano
- ☐ Yes, Puerto Rican
- ☐ Yes, Cuban
- ☐ Yes, another Hispanic, Latino, or Spanish origin – *Print origin, for example, Argentinean, Colombian, Dominican, Nicaraguan, Salvadoran, Spaniard, and so on.*

Figure 23.1: Question asking about one's Hispanic origin in the 2020 U.S. Census. Figure reprinted from the American Community Survey (2020): https://censusreporter.org/topics/race-hispanic/

6 What is Person 1's race? *Mark (X) one or more boxes.*

- ☐ White
- ☐ Black or African Am.
- ☐ American Indian or Alaska Native — *Print name of enrolled or principal tribe.*

☐ Asian Indian	☐ Japanese	☐ Native Hawaiian
☐ Chinese	☐ Korean	☐ Guamanian or Chamorro
☐ Filipino	☐ Vietnamese	☐ Samoan
☐ Other Asian – *Print race, for example, Hmong, Laotian, Thai, Pakistani, Cambodian, and so on.*		☐ Other Pacific Islander – *Print race, for example, Fijian, Tongan, and so on.*

- ☐ Some other race – *Print race.*

Figure 23.2: Question asking about one's race in the 2020 U.S. Census. Figure reprinted from the American Community Survey (2020): https://censusreporter.org/topics/race-hispanic/

23.2 Conclusion

Chapter 24

Reproducibility

It is important for work to be reproducible by others. Reproducibility and replicability are key goals of the open science movement [Gandrud, 2020, Open Science Collaboration, 2015, Tackett et al., 2019a,b]. To that aim, I provide information below on my setup that was used to generate the results found in this book. If you run (all of) my code in the same order with the exact same setup, you should get the same results—but let me know if not!

24.1 Session Info

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## Random number generation:
##  RNG:     L'Ecuyer-CMRG
##  Normal:  Inversion
##  Sample:  Rejection
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
```

```
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
##  [1] grid      stats4    splines   stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] mirtCAT_1.10            shiny_1.6.0             dmacs_0.1.0
##  [4] dagitty_0.3-1           nFactors_2.4.1          corrplot_0.89
##  [7] msir_1.3.3              ggpubr_0.4.0            gridExtra_2.3
## [10] uroc_0.1.0              PredictABEL_1.2-4       ResourceSelection_0.3-5
## [13] rms_6.2-0               SparseM_1.81            Hmisc_4.5-0
## [16] Formula_1.2-4           survival_3.2-11         pROC_1.17.0.1
## [19] magrittr_2.0.1          mirt_1.33.2             lattice_0.20-44
## [22] drc_3.0-1               semTools_0.5-4          quantreg_5.86
## [25] mice_3.13.0             snow_0.4-3              simsem_0.5-16
## [28] DT_0.18                 MASS_7.3-54             kableExtra_1.3.4
## [31] rockchalk_1.8.144       semPlot_1.1.2           MOTE_1.0.2
## [34] here_1.0.1              bookdown_0.22           rmarkdown_2.8
## [37] knitr_1.33              tinytex_0.32            gtheory_0.1.2
## [40] lme4_1.1-27             Matrix_1.3-4            psychmeta_2.6.0
## [43] lavaan_0.6-8            MBESS_4.8.0             blandr_0.5.1
## [46] psych_2.1.3             viridis_0.6.1           viridisLite_0.4.0
## [49] forcats_0.5.1           stringr_1.4.0           dplyr_1.0.6
## [52] purrr_0.3.4             readr_1.4.0             tidyr_1.1.3
## [55] tibble_3.1.2            ggplot2_3.3.3           tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##   [1] estimability_1.3      coda_0.19-4           multcomp_1.4-17
##   [4] data.table_1.14.0     rpart_4.1-15          generics_0.1.0
##   [7] TH.data_1.0-10        polspline_1.1.19      webshot_0.5.2
##  [10] xml2_1.3.2            lubridate_1.7.10      httpuv_1.6.1
##  [13] assertthat_0.2.1      xfun_0.23             hms_1.1.0
##  [16] evaluate_0.14         promises_1.2.0.1      fansi_0.5.0
##  [19] progress_1.2.2        dbplyr_2.1.1          readxl_1.3.1
##  [22] igraph_1.2.6          DBI_1.1.1             tmvnsim_1.0-2
##  [25] Rsolnp_1.16           htmlwidgets_1.5.3     reshape_0.8.8
##  [28] ellipsis_0.3.2        backports_1.2.1       V8_3.4.2
##  [31] pbivnorm_0.6.0        permute_0.9-5         jmvcore_1.2.23
##  [34] vctrs_0.3.8           ROCR_1.0-11           abind_1.4-5
##  [37] withr_2.4.2           PBSmodelling_2.68.8   checkmate_2.0.0
##  [40] emmeans_1.6.1         vegan_2.5-7           fdrtool_1.2.16
##  [43] prettyunits_1.1.1     mclust_5.4.7          mnormt_2.0.2
##  [46] svglite_2.0.0         cluster_2.1.2         mi_1.0
##  [49] crayon_1.4.1          pkgconfig_2.0.3       labeling_0.4.2
```

```
##  [52] nlme_3.1-152          nnet_7.3-16           rlang_0.4.11
##  [55] lifecycle_1.0.0       MatrixModels_0.5-0    sandwich_3.0-1
##  [58] kutils_1.70           modelr_0.1.8          cellranger_1.1.0
##  [61] tcltk_4.0.5           rprojroot_2.0.2       matrixStats_0.59.0
##  [64] regsem_1.8.0          carData_3.0-4         boot_1.3-28
##  [67] zoo_1.8-9             reprex_2.0.0          base64enc_0.1-3
##  [70] png_0.1-7             arm_1.11-2            jpeg_0.1-8.1
##  [73] rstatix_0.7.0         ggsignif_0.6.1        scales_1.1.1
##  [76] lpSolve_5.6.15        plyr_1.8.6            compiler_4.0.5
##  [79] RColorBrewer_1.1-2    plotrix_3.8-1         cli_2.5.0
##  [82] pbapply_1.4-3         ps_1.6.0              htmlTable_2.2.1
##  [85] mgcv_1.8-36           tidyselect_1.1.1      stringi_1.6.2
##  [88] lisrelToR_0.1.4       sem_3.1-11            highr_0.9
##  [91] yaml_2.2.1            OpenMx_2.19.5         latticeExtra_0.6-29
##  [94] polynom_1.4-0         tools_4.0.5           animation_2.6
##  [97] parallel_4.0.5        rio_0.5.26            matrixcalc_1.0-4
## [100] rstudioapi_0.13       foreign_0.8-81        GPArotation_2014.11-1
## [103] farver_2.1.0          digest_0.6.27         Rcpp_1.0.6
## [106] car_3.0-10            broom_0.7.6           ez_4.4-0
## [109] later_1.2.0           httr_1.4.2            Deriv_4.1.3
## [112] colorspace_2.0-1      rvest_1.0.0           XML_3.99-0.6
## [115] fs_1.5.0              truncnorm_1.0-8       conquer_1.0.2
## [118] systemfonts_1.0.2     xtable_1.8-4          jsonlite_1.7.2
## [121] nloptr_1.2.2.2        corpcor_1.6.9         glasso_1.11
## [124] dcurver_0.9.2         R6_2.5.0              pillar_1.6.1
## [127] htmltools_0.5.1.1     mime_0.10             glue_1.4.2
## [130] fastmap_1.1.0         minqa_1.2.4           codetools_0.2-18
## [133] mvtnorm_1.1-2         utf8_1.2.1            curl_4.3.1
## [136] gtools_3.9.2          zip_2.2.0             openxlsx_4.2.3
## [139] qgraph_1.6.9          munsell_0.5.0         haven_2.4.1
## [142] reshape2_1.4.4        gtable_0.3.0
```

Bibliography

Thomas M. Achenbach. What are norms and why do we need valid ones? *Clinical Psychology: Science and Practice*, 8(4):446–450, 2001. doi: 10.1093/clipsy.8.4.446. URL https://login.proxy.lib.uiowa.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=2001-09266-006Thomas.Achenbach@uvm.edu. U Vermont, Dept of Psychiatry, Burlington, VT, US. Other Publishers: Educational Publishing Foundation; Wiley-Blackwell Publishing Ltd. Release Date: 20210111. Publication Type: Journal (0100), Peer Reviewed Journal (0110). Format Covered: Print. Document Type: Comment/Reply. Language: EnglishMajor Descriptor: Conduct Disorder; Psychotherapy; Statistical Analysis; Treatment Effectiveness Evaluation; Treatment Outcomes. Classification: Psychotherapy & Psychotherapeutic Counseling (3310). Population: Human (10); Male (30); Female (40). Age Group: Childhood (birth-12 yrs) (100); Preschool Age (2-5 yrs) (160); School Age (6-12 yrs) (180); Adolescence (13-17 yrs) (200). Page Count: 5. Issue Publication Date: Win 2001.

JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. *rmarkdown: Dynamic documents for R*, 2020. URL https://github.com/rstudio/rmarkdown. R package version 2.6.

American Educational Research Association, American Psychological Association, and National Council on Measurement in Education. *Standards for educational and psychological testing*. STANDARDS for Educational and Psychological Testing. American Educational Research Association, Washington, DC, US, 2014.

Pavel Atanasov, Jens Witkowski, Lyle Ungar, Barbara Mellers, and Philip Tetlock. Small steps to accuracy: Incremental belief updaters are better forecasters. *Organizational Behavior and Human Decision Processes*, 160:19–35, 2020. doi: https://doi.org/10.1016/j.obhdp.2020.02.001. URL https://www.sciencedirect.com/science/article/pii/S0749597819300949.

Peter C. Austin and Ewout W. Steyerberg. Graphical assessment of internal and external calibration of logistic regression models by using loess smoothers.

Statistics in Medicine, 33(3):517–535, 2014. doi: https://doi.org/10.1002/sim.5941. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.5941.

Roger Bakeman and Sherryl H. Goodman. Interobserver reliability in clinical research: Current issues and discussion of how to establish best practices. *Journal of Abnormal Psychology*, 129(1):5–13, 2020. doi: 10.1037/abn0000487.

Pablo Ballesteros-Pérez, M Carmen González-Cruz, and Daniel Mora-Melià. Explaining the Bayes' theorem graphically. In *Proceedings of the International Technology, Education and Development Conference*, 2018.

Paul B. Baltes. Longitudinal and cross-sectional sequences in the study of age and generation effects. *Human Development*, 11(3):145–171, 1968. URL http://www.jstor.org/stable/26761719.

Daniel J. Bauer, William C. M. Belzak, and Veronica T. Cole. Simplifying the assessment of measurement invariance over multiple background variables: Using regularized moderated nonlinear factor analysis to detect differential item functioning. *Structural Equation Modeling: A Multidisciplinary Journal*, 27(1):43–55, 2020. doi: 10.1080/10705511.2019.1642754. URL https://doi.org/10.1080/10705511.2019.1642754.

A Alexander Beaujean. *Latent variable modeling using R: A step-by-step guide*. Routledge, 2014.

William C. M. Belzak and Daniel J. Bauer. Improving the assessment of measurement invariance: Using regularization to select anchor items and identify differential item functioning. *Psychological Methods*, in press. doi: 10.1037/met0000253. URL http://proxy.lib.uiowa.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=2020-01276-001ORCID:0000-0001-6594-1651wbelzak@live.unc.edu. Department of Psychology and Neuroscience, University of North Carolina at Chapel Hill, Chapel Hill, NC, US. Release Date: 20200109. Publication Type: Journal (0100), Peer Reviewed Journal (0110). Format Covered: Electronic. Document Type: Journal Article. Language: English. Grant Information: Bauer, Daniel J. Major Descriptor: Item Analysis (Test); Item Response Theory; Machine Learning; Differential Item Functioning; Measurement Invariance. Minor Descriptor: Behavioral Sciences; Computer Simulation. Classification: Research Methods & Experimental Design (2260). Methodology: Empirical Study; Quantitative Study. Supplemental Data: Appendixes Internet; Computer Software Internet; Tables and Figures Internet; Text Internet. Publication History: Accepted Date: Nov 11, 2019; Revised Date: Oct 31, 2019; First Submitted Date: May 5, 2019. Copyright Statement: American Psychological Association. 2020. Sponsor: National Institutes of Health, US. Grant: R01 DA034636. Recipients: Bauer, Daniel J. (Prin Inv) Sponsor: Department of Defense, US. Other Details: National Defense Science and Engineering Graduate (NDSEG) Fellowship. Recipients: Belzak, William C. M. 10.1037/met0000253.supp (Supplemental).

J Eric Bickel and Seong Dae Kim. Verification of the weather channel probability of precipitation forecasts. *Monthly Weather Review*, 136(12):4867–4881, 2008. doi: 10.1175/2008MWR2547.1.

J. Martin Bland and Douglas G. Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The Lancet*, 327 (8476):307–310, 1986. doi: 10.1016/S0140-6736(86)90837-8. URL http://www.sciencedirect.com/science/article/pii/S0140673686908378.

J. Martin Bland and Douglas G. Altman. Measuring agreement in method comparison studies. *Statistical Methods in Medical Research*, 8(2):135–160, 1999. doi: 10.1177/096228029900800204. URL https://journals.sagepub.com/doi/abs/10.1177/096228029900800204.

Fergus Bolger and Dilek Önkal Atay. The effects of feedback on judgmental interval predictions. *International Journal of Forecasting*, 20(1):29–39, 2004. doi: 10.1016/S0169-2070(03)00009-8. URL http://www.sciencedirect.com/science/article/pii/S0169207003000098.

Kenneth A. Bollen and Shawn Bauldry. Three cs in measurement models: Causal indicators, composite indicators, and covariates. *Psychological Methods*, 16(3):265–284, 2011. doi: 10.1037/a0024448.

Kenneth A. Bollen and Adamantios Diamantopoulos. In defense of causal-formative indicators: A minority report. *Psychological Methods*, 22(3):581–596, 2017. doi: 10.1037/met0000056.

Kenneth A. Bollen and Richard D. Lennox. Conventional wisdom on measurement: A structural equation perspective. *Psychological Bulletin*, 110(2): 305–314, 1991. doi: 10.1037/0033-2909.110.2.305.

Robert F. Bornstein. Toward a process-focused model of test score validity: Improving psychological assessment in science and practice. *Psychological Assessment*, 23(2):532–544, 2011. doi: 10.1037/a0022402.

Denny Borsboom. *Conceptual issues in psychological measurement*. Universiteit van Amsterdam, 2003.

George E.P. Box. *Robustness in the strategy of scientific model building*. Academic Press, New York, 1979. Wisconsin Univ-Madison Mathematics Research Center.

Robert L. Brennan. Generalizability theory. *Educational Measurement: Issues and Practice*, 11(4):27–34, 1992. doi: https://doi.org/10.1111/j.1745-3992.1992.tb00260.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1745-3992.1992.tb00260.x.

Buros Center for Testing. *The twenty-first mental measurements yearbook*. Buros Center for Testing, Lincoln, NE, 2021.

Paul-Christian Bürkner. Bayesian item response modelling in r with brms and stan. *arXiv*, 2019. doi: 1905.09501.

Donald T. Campbell and Donald W. Fiske. Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin*, 56(2): 81–105, 1959. doi: 10.1037/h0046016. US: American Psychological Association.

Phil Chalmers. *mirt: Multidimensional item response theory*, 2020a. URL https://CRAN.R-project.org/package=mirt. R package version 1.33.2.

Phil Chalmers. *mirtCAT: Computerized adaptive testing with multidimensional item response theory*, 2020b. URL https://CRAN.R-project.org/package=mirtCAT. R package version 1.10.

Jerome P Charba and William H Klein. Skill in precipitation forecasting in the national weather service. *Bulletin of the American Meteorological Society*, 61 (12):1546–1555, 1980. doi: 10.1175/1520-0477(1980)061<1546:SIPFIT>2.0.CO;2.

Fang Fang Chen. Sensitivity of goodness of fit indexes to lack of measurement invariance. *Structural Equation Modeling: A Multidisciplinary Journal*, 14 (3):464–504, 2007. doi: 10.1080/10705510701301834. URL https://doi.org/10.1080/10705510701301834.

Yu Chen, Ricardo B. C. Prudêncio, Tom Diethe, and Peter Flach. 3-irt: A new item response model and its applications. *arXiv:1903.04016*, 2019. URL https://arxiv.org/abs/1903.04016.

Ying Cheng, Can Shao, and Quinn N. Lathrop. The mediated mimic model for understanding the underlying mechanism of dif. *Educational and Psychological Measurement*, 76(1):43–63, 2016. doi: 10.1177/0013164415576187. URL https://journals.sagepub.com/doi/abs/10.1177/0013164415576187.

Gordon W. Cheung and Roger B. Rensvold. Evaluating goodness-of-fit indexes for testing measurement invariance. *Structural Equation Modeling: A Multidisciplinary Journal*, 9(2):233–255, 2002. doi: 10.1207/s15328007sem0902_5. URL http://dx.doi.org/10.1207/S15328007SEM0902_5.

Mustafa Emre Civelek. *Essentials of structural equation modeling*. Zea E-Books, Lincoln, Nebraska, 2018.

Veronica Cole, Nisha Gottfredson, and Michael Giordano. *aMNLFA: Automated Fitting of Moderated Nonlinear Factor Analysis Through the Mplus Program*, 2018. URL https://CRAN.R-project.org/package=aMNLFA. R package version 0.1.

Harris Cooper, Larry V Hedges, and Jeffrey C Valentine. *The handbook of research synthesis and meta-analysis*. Russell Sage Foundation, 2009.

Jose M. Cortina. What is coefficient alpha? an examination of theory and applications. *Journal of Applied Psychology*, 78:98–104, 1993. doi: 10.1037/0021-9010.78.1.98.

Patrick J. Curran, Andrea L. Howard, Sierra A. Bainter, Stephanie T. Lane, and James S. McGinley. The separation of between-person and within-person components of individual change over time: A latent curve model with structured residuals. *Journal of Consulting and Clinical Psychology*, 82:8–94, 2014. doi: 10.1037/a0035297.

Deepankar Datta. *blandr: Bland-Altman method comparison*, 2018. URL https://github.com/deepankardatta/blandr/. R package version 0.5.1.

Adamantios Diamantopoulos, Petra Riefler, and Katharina P. Roth. Advancing formative measurement models. *Journal of Business Research*, 61(12): 1203–1218, 2008. doi: 10.1016/j.jbusres.2008.01.009. URL http://www.sciencedirect.com/science/article/pii/S0148296308000118.

David Dueber. *dmacs: Measurement nonequivalence effect size calculator*, 2019. URL https://github.com/ddueber/dmacs. R package version 0.1.0.

Greg J. Duncan, Mimi Engel, Amy Claessens, and Chantelle J. Dowsett. Replication and robustness in developmental research. *Developmental Psychology*, 50(11):2417–2425, 2014. doi: 10.1037/a0037996.

Thomas J. Dunn, Thom Baguley, and Vivienne Brunsden. From alpha to omega: A practical solution to the pervasive problem of internal consistency estimation. *British Journal of Psychology*, 105(3):399–412, 2014. doi: 10.1111/bjop.12046. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/bjop.12046.

David M. Eddy. *Probabilistic reasoning in clinical medicine: Problems and opportunities*, pages 249–267. Cambridge University Press, 1982.

Jeffrey R. Edwards. The fallacy of formative measurement. *Organizational Research Methods*, 14(2):370–388, 2011. doi: 10.1177/1094428110378369. URL https://doi.org/10.1177/1094428110378369.

Jeffrey R. Edwards and Richard P. Bagozzi. On the nature and direction of relationships between constructs and measures. *Psychological Methods*, 5(2): 155–174, 2000. doi: 10.1037/1082-989X.5.2.155. Journal Article Jun.

Susan E. Embretson. The new rules of measurement. *Psychological Assessment*, 8:341–349, 1996. doi: 10.1037/1040-3590.8.4.341.

Susan E. Embretson and Steven Paul Reise. *Item response theory for psychologists*, volume 4. Lawrence Erlbaum Associates, Mahwah, NJ, US, 2000.

Sacha Epskamp. *semPlot: Path diagrams and visual analysis of various SEM packages' output*, 2019. URL https://github.com/SachaEpskamp/semPlot. R package version 1.1.2.

David P. Farrington and Rolf Loeber. Relative improvement over chance (RIOC) and phi as measures of predictive efficiency and strength of association in 2×2 tables. *Journal of Quantitative Criminology*, 5(3):201–213, 1989. doi: 10.1007/BF01062737. URL https://doi.org/10.1007/BF01062737.

Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. Statistical power analyses using g*power 3.1: Tests for correlation and regression analyses. *Behavior Research Methods*, 41(4):1149–1160, 2009. doi: 10.3758/brm.41.4.1149. URL http://dx.doi.org/10.3758/BRM.41.4.1149.

Donald W. Fiske and Donald T. Campbell. Citations do not solve problems. *Psychological Bulletin*, 112(3):393–395, 1992. doi: 10.1037/0033-2909.112.3.393.

David B. Flora. Your coefficient alpha is probably wrong, but which coefficient omega is right? a tutorial on using r to obtain better reliability estimates. *Advances in Methods and Practices in Psychological Science*, 3(4):484–501, 2020. doi: 10.1177/2515245920951747. URL https://journals.sagepub.com/doi/abs/10.1177/2515245920951747.

Jeremy Freese and David Peterson. Replication in social science. *Annual Review of Sociology*, 2017.

R. Michael Furr. *Psychometrics: An introduction*. SAGE publications, 2017.

R. Michael Furr and Sarah Heuckeroth. The "quantifying construct validity" procedure: Its role, value, interpretations, and computation. *Assessment*, 26 (4):555–566, 2019. doi: 10.1177/1073191118820638. URL https://journals.sagepub.com/doi/abs/10.1177/1073191118820638.

Christopher Gandrud. *Reproducible Research with R and R Studio*. CRC Press, 3rd edition, 2020. URL https://www.routledge.com/Reproducible-Research-with-R-and-RStudio/Gandrud/p/book/9780367143985.

Andrew Gelman and Eric Loken. The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time. *Department of Statistics, Columbia University*, 2013.

Robert D. Gibbons, David J. Weiss, Ellen Frank, and David Kupfer. Computerized adaptive diagnosis and testing of mental health disorders. *Annual Review of Clinical Psychology*, 12(1):83–104, 2016. doi: 10.1146/annurev-clinpsy-021815-093634. URL https://www.annualreviews.org/doi/abs/10.1146/annurev-clinpsy-021815-093634.

Tilmann Gneiting and Eva-Maria Walz. Receiver operating characteristic (ROC) movies, universal ROC (UROC) curves, and coefficient of predictive ability (CPA). *arXiv:1912.01956v2*, 2020.

Laura D. Goodwin and Nancy L. Leech. Understanding correlation: Factors that affect the size of r. *The Journal of Experimental Education*, 74(3):249–266, 2006. doi: 10.3200/JEXE.74.3.249-266. URL https://doi.org/10.3200/JEXE.74.3.249-266.

Nisha C. Gottfredson, Veronica T. Cole, Michael L. Giordano, Daniel J. Bauer, Andrea M. Hussong, and Susan T. Ennett. Simplifying the implementation of modern scale scoring methods with an automated R package: Automated moderated nonlinear factor analysis (aMNLFA). *Addictive Behaviors*, 94:65–73, 2019. doi: 10.1016/j.addbeh.2018.10.031. URL http://www.sciencedirect.com/science/article/pii/S0306460318312310.

John Graham, Allison Olchowski, and Tamika Gilreath. How many imputations are really needed? some practical clarifications of multiple imputation theory. *Prevention Science*, 8(3):206–213, 2007. doi: 10.1007/s11121-007-0070-9. URL http://dx.doi.org/10.1007/s11121-007-0070-9.

Samuel B. Green and Yanyun Yang. Evaluation of dimensionality in the assessment of internal consistency reliability: Coefficient alpha and omega coefficients. *Educational Measurement: Issues and Practice*, 34(4):14–20, 2015. doi: 10.1111/emip.12100. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/emip.12100.

Curt Hagquist. Explaining differential item functioning focusing on the crucial role of external information – an example from the measurement of adolescent mental health. *BMC Medical Research Methodology*, 19(1):185, 2019. doi: 10.1186/s12874-019-0828-3. URL https://doi.org/10.1186/s12874-019-0828-3.

Curt Hagquist and David Andrich. Recent advances in analysis of differential item functioning in health research using the rasch model. *Health and Quality of Life Outcomes*, 15(1):181, 2017. doi: 10.1186/s12955-017-0755-0. URL https://doi.org/10.1186/s12955-017-0755-0.

Gregory R. Hancock and Brian F. French. *Power analysis in structural equation modeling*, pages 117–159. Quantitative methods in education and the behavioral sciences: Issues, research, and teaching. IAP Information Age Publishing, Charlotte, NC, US, 2013.

Andrew M. Hardin, Jerry Cha-Jan Chang, Mark A. Fuller, and Gholamreza Torkzadeh. Formative measurement and academic research: In search of measurement theory. *Educational and Psychological Measurement*, 71(2):281–305, 2011. doi: 10.1177/0013164410370208. URL https://journals.sagepub.com/doi/abs/10.1177/0013164410370208.

Frank Harrell. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer, 2015.

Frank E. Harrell, Jr. *rms: Regression modeling strategies*, 2021. URL https://CRAN.R-project.org/package=rms. R package version 6.1-1.

Andrew F. Hayes and Jacob J. Coutts. Use omega rather than cronbach's alpha for estimating reliability. but…. *Communication Methods and Measures*, 14 (1):1–24, 2020. doi: 10.1080/19312458.2020.1718629. URL https://doi.org/10.1080/19312458.2020.1718629.

S. C. Hayes, R. O. Nelson, and R. B. Jarrett. The treatment utility of assessment: A functional approach to evaluating assessment quality. *American Psychologist*, 42:963–974, 1987. doi: 10.1037/0003-066X.42.11.963.

P.A. Hays. *Addressing Cultural Complexities in Practice: Assessment, Diagnosis, and Therapy*. American Psychological Association, 2016.

Jörg Henseler, Christian M. Ringle, and Marko Sarstedt. A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science*, 43(1):115–135, 2015. doi: 10.1007/s11747-014-0403-8. URL https://doi.org/10.1007/s11747-014-0403-8.

Stephen J. Hoch. Counterfactual reasoning and accuracy in predicting personal events. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(4):719–731, 1985. doi: 10.1037/0278-7393.11.1-4.719. URL http://proxyiub.uits.iu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1986-10967-001&site=ehost-live&scope=site. U Chicago, Graduate School of Business, Ctr for Decision Research. Release Date: 20060710. Correction Date: 20100412. Publication Type: Journal (0100), Peer Reviewed Journal (0110). Format Covered: Print. Document Type: Journal Article. Language: English. Major Descriptor: Life Experiences; Prediction; Reasoning. Minor Descriptor: Cognitive Processes. Classification: Cognitive Processes (2340). Population: Human (10). Age Group: Adulthood (18 yrs & older) (300). Methodology: Empirical Study. References Available: Y. Page Count: 13. Issue Publication Date: Oct, 1985. Publication History: Revised Date: Feb 18, 1985; First Submitted Date: Aug 29, 1984. Copyright Statement: American Psychological Association. 1985.

Susan Elizabeth Hough. *Predicting the unpredictable: The tumultuous science of earthquake prediction*. Princeton University Press, 2016.

Roy D. Howell, Einar Breivik, and James B. Wilcox. Reconsidering formative measurement. *Psychological Methods*, 12(2):205–218, 2007. doi: 10.1037/1082-989X.12.2.205.

Alan Huebner and Marisa Lucht. Generalizability theory in R. *Practical Assessment, Research & Evaluation*, 24(5):2, 2019. doi: 10.7275/5065-gc10.

John Hunsley, Catherine M. Lee, James M. Wood, and Whitney Taylor. *Controversial and questionable assessment techniques*, pages 42–82. The Guilford Press, New York, NY, US, 2nd edition, 2015.

Andrea M. Hussong, Patrick J. Curran, and Daniel J. Bauer. Integrative data analysis in clinical psychology research. *Annual Review of Clinical Psychology*, 9(1):61–89, 2013. doi: 10.1146/annurev-clinpsy-050212-185522. URL https://www.annualreviews.org/doi/abs/10.1146/annurev-clinpsy-050212-185522.

Andrea M. Hussong, Daniel J. Bauer, Michael L. Giordano, and Patrick J. Curran. Harmonizing altered measures in integrative data analysis: A methods analogue study. *Behavior Research Methods*, 2020. doi: 10.3758/s13428-020-01472-7. URL https://doi.org/10.3758/s13428-020-01472-7.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, Melbourne, Australia, 2nd edition, 2018.

Zhehan Jiang. Using the linear mixed-effect model framework to estimate generalizability variance components in R. *Methodology*, 14(3):133–142, 2018. doi: 10.1027/1614-2241/a000149. URL https://econtent.hogrefe.com/doi/abs/10.1027/1614-2241/a000149.

Leslie K. John, George Loewenstein, and Drazen Prelec. Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science*, 23(5):524–532, 2012. doi: 10.1177/0956797611430953. URL https://journals.sagepub.com/doi/abs/10.1177/0956797611430953.

Johnnie E. V. Johnson and Alistair C. Bruce. Calibration of subjective probability judgments in a naturalistic setting. *Organizational Behavior and Human Decision Processes*, 85(2):265–290, 2001. doi: 10.1006/obhd.2000.2949. URL http://www.sciencedirect.com/science/article/pii/S0749597800929492.

Paul E. Johnson. *rockchalk: Regression estimation and presentation*, 2019. URL https://CRAN.R-project.org/package=rockchalk. R package version 1.8.144.

Terrence D. Jorgensen, Benjamin A. Kite, Po-Yi Chen, and Stephen D. Short. Permutation randomization methods for testing measurement equivalence and detecting differential item functioning in multiple-group confirmatory factor analysis. *Psychological Methods*, 23(4):708–728, 2018. doi: 10.1037/met0000152.

Terrence D. Jorgensen, Sunthud Pornprasertmanit, Alexander M. Schoemann, and Yves Rosseel. *semTools: Useful tools for structural equation modeling*, 2021. URL https://github.com/simsem/semTools/wiki. R package version 0.5-4.

Ken Kelley. *MBESS: The MBESS R Package*, 2020. URL http://nd.edu/~kkelley/site/MBESS.html. R package version 4.8.0.

Ken Kelley and Sunthud Pornprasertmanit. Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods*, 21(1):69–92, 2016. doi: 10.1037/a0040086.

Gideon Keren. Facing uncertainty in the game of bridge: A calibration study. *Organizational Behavior and Human Decision Processes*, 39(1):98–114, 1987. doi: 10.1016/0749-5978(87)90047-1. URL http://www.sciencedirect.com/science/article/pii/0749597887900471.

Ronald C. Kessler, Robert M. Bossarte, Alex Luedtke, Alan M. Zaslavsky, and Jose R. Zubizarreta. Suicide prediction models: a critical review of recent research with recommendations for the way forward. *Molecular Psychiatry*, 25(1):168–179, 2020. doi: 10.1038/s41380-019-0531-0. URL https://doi.org/10.1038/s41380-019-0531-0.

Rogier Kievit, Willem Frankenhuis, Lourens Waldorp, and Denny Borsboom. Simpson's paradox in psychological science: a practical guide. *Frontiers in Psychology*, 4(513), 2013. doi: 10.3389/fpsyg.2013.00513. URL https://www.frontiersin.org/article/10.3389/fpsyg.2013.00513.

Donald F. Klein and T. Anne Cleary. Platonic true scores: Further comment. *Psychological Bulletin*, 71(4):278–280, 1969. doi: 10.1037/h0026852.

Derek J Koehler, Lyle Brenner, and Dale Griffin. *The calibration of expert judgment: Heuristics and biases beyond the laboratory*, chapter 39. Cambridge University Press, New York, NY, 2002.

Asher Koriat, Sarah Lichtenstein, and Baruch Fischhoff. Reasons for confidence. *Journal of Experimental Psychology: Human Learning and Memory*, 6(2):107–118, 1980. doi: 10.1037/0278-7393.6.2.107. URL http://proxyiub.uits.iu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1981-07049-001&site=ehost-live&scope=site. U Haifa, Israel. Release Date: 20060329. Publication Type: Journal (0100), Peer Reviewed Journal (0110). Format Covered: Print. Document Type: Journal Article. Language: English. Major Descriptor: Decision Making; Judgment; Prediction. Classification: Cognitive Processes (2340). Population: Human (10). References Available: Y. Page Count: 12. Issue Publication Date: Mar, 1980. Copyright Statement: American Psychological Association. 1980.

Robert F. Krueger, Penny E. Nichol, Brian M. Hicks, Kristian E. Markon, Christopher J. Patrick, William G. lacono, and Matt McGue. Using latent trait modeling to conceptualize an alcohol problems continuum. *Psychological Assessment*, 16(2):107–119, 2004. doi: 10.1037/1040-3590.16.2.107. URL http://ezproxy.lib.indiana.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=pas-16-2-107&site=ehost-livekrueg038@umn.edu. Department of Psychology, University of Minnesota, Minneapolis, MN, US. Other Journal Titles: Psychological Assessment: A Journal of Consulting and Clinical Psychology. Release Date: 20060710. Correction Date: 20090209. Publication Type: Journal, Peer Reviewed Journal. Media Covered: Print. Media Available: Electronic; Print. Document Type: Journal Article. Language: English. Major Descriptor: Alcoholism;

Item Response Theory; Severity (Disorders); Simulation; Symptoms. Minor Descriptor: Alcohol Withdrawal; Drug Tolerance. Classification: Substance Abuse & Addiction (3233); Population: Human (10); Male (30); Location: US. Age Group: Adulthood (18 yrs & older) (300); . Tests & Measures: Substance Abuse Module; Methodology: Empirical Study; Qualitative Study. References Available: Y.. Issue Publication Date: Jun, 2004. Publication History: Accepted Date: Dec 8, 2003; Revised Date: Nov 14, 2003; First Submitted Date: Apr 22, 2003. Copyright: American Psychological Association. 2004.;.

Suman Kundu, Yurii S. Aulchenko, and A. Cecile J.W. Janssens. *PredictABEL: Assessment of risk prediction models*, 2020. URL https://CRAN.R-project.org/package=PredictABEL. R package version 1.2-4.

Michael J. Larson and Kaylie A. Carbine. Sample size calculations in human electrophysiology (eeg and erp) studies: A systematic review and recommendations for increased rigor. *International Journal of Psychophysiology*, 111:33–41, 2017. doi: 10.1016/j.ijpsycho.2016.06.015. URL http://www.sciencedirect.com/science/article/pii/S0167876016301180.

Kimberley M. Lek and Rens Van De Schoot. A comparison of the single, conditional and person-specific standard error of measurement: What do they measure and when to use them? *Frontiers in Applied Mathematics and Statistics*, 4(40), 2018. doi: 10.3389/fams.2018.00040. URL https://www.frontiersin.org/article/10.3389/fams.2018.00040.

Subhash R. Lele, Jonah L. Keim, and Peter Solymos. *ResourceSelection: Resource selection (probability) functions for use-svailability data*, 2019. URL https://github.com/psolymos/ResourceSelection. R package version 0.3-5.

Scott O. Lilienfeld. Psychological treatments that cause harm. *Perspectives on Psychological Science*, 2(1):53–70, 2007. doi: 10.1111/j.1745-6916.2007.00029.x. United Kingdom: Blackwell Publishing.

Scott O. Lilienfeld. Psychology's replication crisis and the grant culture: Righting the ship. *Perspectives on Psychological Science*, 12(4):660–664, 2017. doi: 10.1177/1745691616687745. URL https://journals.sagepub.com/doi/abs/10.1177/1745691616687745.

Scott O. Lilienfeld, Katheryn Sauvigne, Steven Jay Lynn, Robert D Latzman, Robin Cautin, and Irwin D. Waldman. Fifty psychological and psychiatric terms to avoid: A list of inaccurate, misleading, misused, ambiguous, and logically confused words and phrases. *Frontiers in Psychology*, 6, 2015. doi: 10.3389/fpsyg.2015.01100. URL http://www.frontiersin.org/Journal/Abstract.aspx?s=346&name=educational_psychology&ART_DOI=10.3389/fpsyg.2015.01100.

Oliver Lindhiem, Isaac T. Petersen, Lucas K. Mentch, and Eric A. Youngstrom. The importance of calibration in clinical psychology. *Assessment*, 27(4):840–854, 2020. doi: 10.1177/1073191117752055. URL https://doi.org/10.1177/1073191117752055.

Todd D. Little. *Longitudinal structural equation modeling*. Methodology in the social sciences. The Guilford Press, New York, NY, US, 2013.

Todd D. Little, David W. Slegers, and Noel A. Card. A non-arbitrary method of identifying and scaling latent variables in sem and macs models. *Structural Equation Modeling*, 13(1):59–72, 2006. doi: 10.1207/s15328007sem1301_3. URL http://ezproxy.lib.indiana.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=20573424&site=ehost-live. Little, Todd D. 1; Email Address: yhat@ku.edu Slegers, David W. 1 Card, Noel A. 1; Affiliation: 1: University of Kansas; Source Info: 2006, Vol. 13 Issue 1, p59; Subject Term: STRUCTURAL equation modeling; Subject Term: LATENT variables; Subject Term: SCALING laws (Statistical physics); Subject Term: LATENT structure analysis; Subject Term: ANALYSIS of covariance; Subject Term: VARIABLES (Mathematics); Subject Term: MULTIVARIATE analysis; Subject Term: FACTOR analysis; Subject Term: MATHEMATICAL models; Subject Term: MATHEMATICAL statistics; Number of Pages: 14p; Illustrations: 2 Charts; Document Type: Article.

Todd D. Little, Kristopher J. Preacher, James P. Selig, and Noel A. Card. New developments in latent variable panel analyses of longitudinal data. *International Journal of Behavioral Development*, 31(4):357–365, 2007. doi: 10.1177/0165025407077757. URL http://jbd.sagepub.com/content/31/4/357.abstract.

Yu Liu, Roger E. Millsap, Stephen G. West, Jenn-Yun Tein, Rika Tanaka, and Kevin J. Grimm. Testing measurement invariance in longitudinal data with ordered-categorical measures. *Psychological Methods*, 22(3):486–506, 2017. doi: 10.1037/met0000075.

Eric Loken and Andrew Gelman. Measurement error and the replication crisis. *Science*, 355(6325):584–585, 2017. doi: 10.1126/science.aal3618. URL https://science.sciencemag.org/content/sci/355/6325/584.full.pdf.

Spyros Makridakis, Robin M. Hogarth, and Anil Gaba. Forecasting and uncertainty in the economic and business world. *International Journal of Forecasting*, 25(4):794–812, 2009. doi: https://doi.org/10.1016/j.ijforecast.2009.05.012. URL https://www.sciencedirect.com/science/article/pii/S0169207009000855.

Keith A. Markus. Three conceptual impediments to developing scale theory for formative scales. *Methodology*, 14(4):156–164, 2018. doi: 10.1027/1614-2241/a000154. URL https://econtent.hogrefe.com/doi/abs/10.1027/1614-2241/a000154.

J. Gowert Masche and Manfred H. M. van Dulmen. Advances in disentangling age, cohort, and time effects: No quadrature of the circle, but a help. *Developmental Review*, 24(3):322–342, 2004. doi: https://doi.org/10.1016/j.dr.2004.04.002. URL https://www.sciencedirect.com/science/article/pii/S0273229704000206.

Richard M. McFall. Manifesto for a science of clinical psychology. *The Clinical Psychologist*, 44(6):75–91, 1991.

Richard M. McFall. Elaborate reflections on a simple manifesto. *Applied & Preventive Psychology*, 9(1):5–21, 2000. doi: 10.1016/s0962-1849(05)80035-6. Netherlands: Elsevier Science.

Daniel McNeish. Thanks coefficient alpha, we'll take it from here. *Psychological Methods*, 23(3):412–433, 2018. doi: 10.1037/met0000144.

Adam W. Meade. A taxonomy of effect size measures for the differential functioning of items and scales. *Journal of Applied Psychology*, 95(4):728–743, 2010. doi: 10.1037/a0018966.

Paul E. Meehl. When shall we use our heads instead of the formula? *Journal of Counseling Psychology*, 4(4):268–273, 1957. doi: 10.1037/h0047554. US: Wm. C. Brown Co.

Paul E. Meehl. Theoretical risks and tabular asterisks: Sir karl, sir ronald, and the slow progress of soft psychology. *Journal of Consulting and Clinical Psychology*, 46(4):806–834, 1978. doi: 10.1037/0022-006x.46.4.806. US: American Psychological Association.

Paul E. Meehl and Albert Rosen. Antecedent probability and the efficiency of psychometric signs, patterns, or cutting scores. *Psychological Bulletin*, 52(3):194–216, 1955. doi: 10.1037/h0048070. US: American Psychological Association.

Charles E. Metz, David J. Goodenough, and Kurt Rossmann. Evaluation of receiver operating characteristic curve data in terms of information theory, with applications in radiography. *Radiology*, 109(2):297–303, 1973. doi: 10.1148/109.2.297. URL https://pubs.rsna.org/doi/abs/10.1148/109.2.297.

Roger E. Millsap. *Statistical approaches to measurement invariance*. Taylor & Francis, New York, NY, US, 2011.

Christopher T. Moore. *gtheory: Apply generalizability theory with R*, 2016. URL http://EvaluationDashboard.com. R package version 0.1.2.

S. K. Morley, T. V. Brito, and D. T. Welling. Measures of model performance based on the log accuracy ratio. *Space Weather*, 16(1):69–88, 2018. doi: 10.1002/2017SW001669. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017SW001669.

Allan H. Murphy and Robert L. Winkler. Probability forecasting in meterology. *Journal of the American Statistical Association*, 79(387):489–500, 1984. doi: 10.2307/2288395. URL http://www.jstor.org/stable/2288395.

Linda K. Muthén and Bengt O. Muthén. How to use a Monte Carlo study to decide on sample size and determine power. *Structural Equation Modeling: A Multidisciplinary Journal*, 9(4):599–620, 2002. doi: 10.1207/s15328007sem0904_8. URL http://dx.doi.org/10.1207/S15328007SEM0904_8.

Linda K. Muthén and Bengt O. Muthén. Mplus version 8.4, 2019.

Jum C Nunnally and Ira H. Bernstein. *Psychometric theory*. McGraw-Hill, New York, 3rd edition, 1994.

Christopher D. Nye, Jacob Bradburn, Jeffrey Olenick, Christopher Bialko, and Fritz Drasgow. How big are my effects? examining the magnitude of effect sizes in studies of measurement equivalence. *Organizational Research Methods*, 22(3):678–709, 2019. doi: 10.1177/1094428118761122. URL https://journals.sagepub.com/doi/abs/10.1177/1094428118761122.

Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251), 2015. doi: 10.1126/science.aac4716. URL http://www.sciencemag.org/content/349/6251/aac4716.abstract.

Stuart Oskamp. Overconfidence in case-study judgments. *Journal of Consulting Psychology*, 29(3):261–265, 1965. doi: 10.1037/h0022125. URL http://proxyiub.uits.iu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1965-12467-001&site=ehost-live&scope=site. Claremont Graduate Sch. Other Publishers: American Association for Applied Psychology; Dentan Printing Company; Science Press Printing Company. Release Date: 20060329. Publication Type: Journal (0100), Peer Reviewed Journal (0110). Format Covered: Print. Document Type: Journal Article. Language: English. Major Descriptor: Medical Diagnosis; Psychologists; Self Confidence. Minor Descriptor: Judgment. Classification: Professional Personnel Attitudes & Characteristics (3430). Population: Human (10). Age Group: Adulthood (18 yrs & older) (300). Methodology: Empirical Study; Quantitative Study. References Available: Y. Page Count: 5. Issue Publication Date: Jun, 1965. Copyright Statement: American Psychological Association. 1965.

Judea Pearl. Linear models: A useful "microscope" for causal analysis. *Journal of Causal Inference*, 1(1):155–170, 2013. doi: 10.1515/jci-2013-0003. URL https://doi.org/10.1515/jci-2013-0003.

Gjalt-Jorn Peters. The alpha and the omega of scale reliability and validity: Why and how to abandon Cronbach's alpha and the route towards more comprehensive assessment of scale quality. *European Health Psychologist*, 16 (2):56–69, 2014.

Philip M. Podsakoff, Scott B. MacKenzie, and Nathan P. Podsakoff. Sources of method bias in social science research and recommendations on how to control it. *Annual Review of Psychology*, 63(1):539–569, 2012. doi: 10.1146/annurev-psych-120710-100452. URL https://www.annualreviews.org/doi/abs/10.1146/annurev-psych-120710-100452.

Sunthud Pornprasertmanit, Patrick Miller, Alexander Schoemann, and Terrence D. Jorgensen. *simsem: SIMulated structural equation modeling*, 2020. URL http://www.simsem.org. R package version 0.5-15.905.

Diane L. Putnick and Marc H. Bornstein. Measurement invariance conventions and reporting: The state of the art and future directions for psychological research. *Developmental Review*, 41:71–90, 2016. doi: 10.1016/j.dr.2016.06.004. URL http://www.sciencedirect.com/science/article/pii/S0273229716300351.

R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL https://www.R-project.org/.

Gilles Raiche and David Magis. *nFactors: Parallel analysis and other non graphical solutions to the Cattell scree test*, 2020. URL https://CRAN.R-project.org/package=nFactors. R package version 2.4.1.

Tenko Raykov. Bias of coefficient for fixed congeneric measures with correlated errors. 25(1):69–76, 2001. doi: 10.1177/01466216010251005. URL http://journals.sagepub.com/doi/abs/10.1177/01466216010251005.

Tenko Raykov and George A. Marcoulides. Thanks coefficient alpha, we still need you! *Educational and Psychological Measurement*, 79(1):200–210, 2019. doi: 10.1177/0013164417725127. URL https://journals.sagepub.com/doi/abs/10.1177/0013164417725127.

Tenko Raykov, George A. Marcoulides, Michael Harrison, and Mingcai Zhang. On the dependability of a popular procedure for studying measurement invariance: A cause for concern? *Structural Equation Modeling: A Multidisciplinary Journal*, in press. doi: 10.1080/10705511.2019.1610409. URL https://doi.org/10.1080/10705511.2019.1610409.

Steven P. Reise and Niels G. Waller. Item response theory and clinical measurement. *Annual Review of Clinical Psychology*, 5(1):27–48, 2009. doi: 10.1146/annurev.clinpsy.032408.153553. URL http://www.annualreviews.org/doi/abs/10.1146/annurev.clinpsy.032408.153553.

William Revelle. *psych: Procedures for psychological, psychometric, and personality research*, 2020. URL https://personality-project.org/r/psych/https://personality-project.org/r/psych-manual.pdf. R package version 2.0.12.

William Revelle and David M. Condon. Reliability from to : A tutorial. *Psychological Assessment*, 31(12):1395–1411, 2019. doi: 10.1037/pas0000754.

Edward E. Rigdon. *Polychoric correlation coefficient*. SAGE Publications, 2010. doi: 10.4135/9781412961288.

Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. *pROC: Display and analyze ROC curves*, 2021. URL http://expasy.org/tools/pROC/. R package version 1.17.0.1.

Alexander Robitzsch. *mnlfa: Moderated nonlinear factor analysis*, 2019. URL https://CRAN.R-project.org/package=mnlfa. R package version 0.1-53.

Yves Rosseel, Terrence D. Jorgensen, and Nicholas Rockwood. *lavaan: Latent variable analysis*, 2020. URL http://lavaan.org. R package version 0.6-7.

J Edward Russo and Paul JH Schoemaker. Managing overconfidence. *Sloan management review*, 33(2):7, 1992.

K. Warner Schaie. A general model for the study of developmental problems. *Psychological Bulletin*, 64(2):92–107, 1965. doi: 10.1037/h0022371.

K. Warner Schaie. *Developmental influences on adult intelligence: The Seattle Longitudinal Study*. Oxford University Press, 2005.

K. Warner Schaie and P. B. Baltes. On sequential strategies in developmental research. *Human Development*, 18(5):384–390, 1975. doi: 10.1159/000271498. URL https://www.karger.com/DOI/10.1159/000271498.

Frank L. Schmidt and John E. Hunter. Measurement error in psychological research: Lessons from 26 research scenarios. *Psychological Methods*, 1(2): 199–223, 1996. doi: 10.1037/1082-989X.1.2.199.

James B. Schreiber, Amaury Nora, Frances K. Stage, Elizabeth A. Barlow, and Jamie King. Reporting structural equation modeling and confirmatory factor analysis results: A review. *Journal of Educational Research*, 99(6):323–337, 2006. doi: 10.3200/JOER.99.6.323-338. URL http://ezproxy.lib.indiana.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=tfh&AN=21615216&site=ehost-live&scope=site. Journal of Educational Research Source Information: Jul/Aug2006, Vol. 99 Issue 6, p323; Subject Term: FACTOR analysis; Subject Term: EQUATIONS; Subject Term: STATISTICS; Subject Term: CORRELATION (Statistics); Subject Term: ALGEBRA; Subject Term: ; Number of Pages: 15p; ; Illustrations: 6 Diagrams, 5 Charts; ; Document Type: Article;.

L Sechrest. Incremental validity: A recommendation. *Educational and Psychological Measurement*, 23:153–158, 1963. doi: 10.1177/001316446302300113.

R. J. Shavelson, N. M. Webb, and R. L. Rawley. Generalizability theory. *American Psychologist*, 44:922–932, 1989. doi: 10.1037/0003-066X.44.6.922.

Patrick E. Shrout and Joseph L. Rodgers. Psychology, science, and knowledge construction: Broadening perspectives from the replication crisis. *Annual Review of Psychology*, 69(1):487–510, 2018. doi: 10.1146/annurev-psych-122216-011845. URL http://www.annualreviews.org/doi/abs/10.1146/annurev-psych-122216-011845.

Klaas Sijtsma. On the use, the misuse, and the very limited usefulness of cronbach's alpha. *Psychometrika*, 74(1):107, 2008. doi: 10.1007/s11336-008-9101-0. URL https://doi.org/10.1007/s11336-008-9101-0.

Nate Silver. *The signal and the noise: why so many predictions fail–but some don't*. Penguin, 2012.

Dorota Skala. Overconfidence in psychology and finance-an interdisciplinary literature review. *Bank i Kredyt*, (4):33–50, 2008.

Gregory T. Smith, Denis M. McCarthy, and Kristen G. Anderson. On the sins of short-form development. *Psychological Assessment*, 12(1):102–111, 2000. doi: 10.1037/1040-3590.12.1.102. URL http://www.apa.org. English.

Eugene Somoza, Luis Soutullo-Esperon, and Douglas Mossman. Evaluation and optimization of diagnostic tests using receiver operating characteristic analysis and information theory. *International Journal of Bio-Medical Computing*, 24 (3):153–189, 1989. doi: 10.1016/0020-7101(89)90029-9. URL http://www.sciencedirect.com/science/article/pii/0020710189900299.

Harold Stanislaw and Natasha Todorov. Calculation of signal detection theory measures. *Behavior Research Methods, Instruments, & Computers*, 31(1): 137–149, 1999. doi: 10.3758/bf03207704. URL http://dx.doi.org/10.3758/BF03207704.

Richard J. Stevens and Katrina K. Poppe. Validation of clinical prediction models: what does the "calibration slope" really measure? *Journal of Clinical Epidemiology*, 118:93–99, 2020. doi: https://doi.org/10.1016/j.jclinepi.2019.09.016. URL https://www.sciencedirect.com/science/article/pii/S0895435619303579.

Ewout W. Steyerberg and Yvonne Vergouwe. Towards better clinical prediction models: seven steps for development and an abcd for validation. *European Heart Journal*, 35(29):1925–1931, 2014. doi: 10.1093/eurheartj/ehu207. URL https://doi.org/10.1093/eurheartj/ehu207.

Jennifer L. Tackett, Cassandra M. Brandes, Kevin M. King, and Kristian E. Markon. Psychology's replication crisis and clinical psychological science. *Annual Review of Clinical Psychology*, 15(1):579–604, 2019a. doi: 10.1146/annurev-clinpsy-050718-095710. URL https://doi.org/10.1146/annurev-clinpsy-050718-095710.

Jennifer L. Tackett, Cassandra M. Brandes, and Kathleen W. Reardon. Leveraging the open science framework in clinical psychological assessment research. *Psychological Assessment*, 31(12):1386–1394, 2019b. doi: 10.1037/pas0000583.

Philip E. Tetlock. *Expert political judgment: How good is it? How can we know? - New edition*. Princeton University Press, 2017.

Johannes Textor, Benito van der Zander, and Ankur Ankan. *dagitty: Graphical analysis of structural causal models*, 2021. URL https://CRAN.R-project.org/package=dagitty. R package version 0.3-1.

Michael L. Thomas. Advances in applications of item response theory to clinical assessment. *Psychological Assessment*, 31(12):1442–1455, 2019. doi: 10.1037/pas0000597.

Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66 (8):1352–1362, 2015. doi: 10.1057/jors.2014.103. URL http://dx.doi.org/10.1057/jors.2014.103.

David Trafimow. A defense against the alleged unreliability of difference scores. *Cogent Mathematics*, 2(1):1064626, 2015. doi: 10.1080/23311835.2015.1064626. URL https://doi.org/10.1080/23311835.2015.1064626.

T. A. Treat and Richard J. Viken. *Measuring test performance with signal detection theory techniques*, volume 1, chapter 37, pages 723–744. American Psychological Association, Washington, D.C., 2012.

Horst Treiblmaier, Peter M. Bentler, and Patrick Mair. Formative constructs implemented via common factors. *Structural Equation Modeling: A Multidisciplinary Journal*, 18(1):1–17, 2011. doi: 10.1080/10705511.2011.532693. URL https://doi.org/10.1080/10705511.2011.532693.

Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974. doi: 10.1126/science.185.4157.1124. URL http://science.sciencemag.org/content/sci/185/4157/1124.full.pdf.

Rens Van De Schoot, Anouck Kluytmans, Lars Tummers, Peter Lugtig, Joop Hox, and Bengt Muthen. Facing off with scylla and charybdis: a comparison of scalar, partial, and the novel possibility of approximate measurement invariance. *Frontiers in Psychology*, 4(770), 2013. doi: 10.3389/fpsyg.2013.00770. URL https://www.frontiersin.org/article/10.3389/fpsyg.2013.00770.

Rens Van De Schoot, Peter Schmidt, Alain De Beuckelaer, Kimberley Lek, and Marielle Zondervan-Zwijnenburg. Editorial: Measurement invariance. *Frontiers in Psychology*, 6(1064), 2015. doi: 10.3389/fpsyg.2015.01064. URL https://www.frontiersin.org/article/10.3389/fpsyg.2015.01064.

Walter P. Vispoel, Carrie A. Morris, and Murat Kilinc. Applications of generalizability theory and their relations to classical test theory and structural equation modeling. *Psychological Methods*, 23(1):1–26, 2018. doi: 10.1037/met0000107.

Clay M. Voorhees, Michael K. Brady, Roger Calantone, and Edward Ramirez. Discriminant validity testing in marketing: An analysis, causes for concern, and proposed remedies. *Journal of the Academy of Marketing Science*, 44 (1):119–134, 2016. doi: 10.1007/s11747-015-0455-4. URL https://doi.org/10.1007/s11747-015-0455-4.

Wen-Chung Wang, Ching-Lin Shih, and Chih-Chien Yang. The mimic method with scale purification for detecting differential item functioning. *Educational and Psychological Measurement*, 69(5):713–731, 2009. doi: 10.1177/0013164409332228. URL https://journals.sagepub.com/doi/abs/10.1177/0013164409332228.

Yilin A. Wang and Mijke Rhemtulla. Power analysis for parameter estimation in structural equation modeling: A discussion and tutorial. *Advances in Methods and Practices in Psychological Science*, in press.

Taiyun Wei and Viliam Simko. *corrplot: Visualization of a correlation matrix*, 2017. URL https://github.com/taiyun/corrplot. R package version 0.84.

Susan Krauss Whitbourne. Longitudinal, cross-sectional, and sequential designs in lifespan developmental psychology, 2019-01-25 2019.

Jerry S. Wiggins. *Personality and prediction: Principles of personality assessment*. Addison-Wesley, Reading, MA, US, 1973.

Yihui Xie. *Dynamic documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015.

Yihui Xie. *bookdown: Authoring books and technical documents with R Markdown*, 2020. URL https://github.com/rstudio/bookdown. R package version 0.21.

Yihui Xie. *knitr: A general-purpose package for dynamic report generation in R*, 2021. URL https://yihui.org/knitr/. R package version 1.31.

Yang Yang and Kenneth C. Land. *Age-period-cohort analysis: New models, methods, and empirical applications*. Taylor & Francis, 2013.

Jun Zhang and Shane T. Mueller. A note on ROC analysis and non-parametric estimate of sensitivity. *Psychometrika*, 70(1):203–212, 2005. doi: 10.1007/s11336-003-1119-8. URL https://doi.org/10.1007/s11336-003-1119-8.