Applying Conditional Distributions To Individuals: Using Latent Variable Models

Feng Ji
Illinois State University, jifenghi@gmail.com
This study proposes a new method to interpret individual results of psychological test batteries. The Mahalanobis distance is a commonly-used measure of how unusual an individual’s profile of scores is compared to a population of score profiles. In models in which there is a set of predictors and a set of dependent variables (e.g., cognitive abilities predicting academic abilities), it is useful to distinguish between a profile of dependent scores that is unusual because its profile of predictor scores is unusual and a profile of dependent scores that is unusual even after controlling for the predictors. The conditional Mahalanobis distance measures the unusualness of a profile shape after controlling for a set of predictors. In psychological assessments, one only has access to observed scores, but the goal is to understand a person’s profile of latent construct scores. Factor score estimates can be calculated, but with measurement error. Using simulations studies, I investigate the accuracy of the conditional Mahalanobis distance when it is used with estimated factor scores. The conditional Mahalanobis distance used with factor scores is more accurate when the factor scores are accurately measured (i.e., the factor loadings are high or the number of indicators increases) and less accurate when the constructs in the latent structure model are highly correlated. I created an R package to assist researchers and practitioners who wish to use the conditional Mahalanobis distance. I illustrate its use with several case studies.

KEYWORDS: Latent Variable Modelling, Psychological Assessment, Unusualness, Mahalanobis Distance, Monte-Carlo Simulation, Statistical Software
APPLYING CONDITIONAL DISTRIBUTIONS TO INDIVIDUALS:
USING LATENT VARIABLE MODELS

FENG JI

COMMITTEE MEMBERS:
Matthew Hesson-McInnis, Chair
W. Joel Schneider
Thomas Critchfield
ACKNOWLEDGMENTS

Finally, I am writing the last part of this thesis – acknowledgements. I have pictured so many times how I would write it, However, when it comes true in reality, I suddenly find I do not know where to start – there are so many people and things that fed me along the way.

First and foremost, enormous gratitude is due to professor W. Joel Schneider for his mentorship during the last two years. I would like to express my most sincere gratitude and deepest appreciation to Joel for all he has done in completing this project, without him it would not exist. I can always vividly recall how Joel and I spent hours and hours in his office or via video calls, working out bit by bit of this project. You taught me so many beyond this project that I cannot possibly list them all here. I also want to thank professor Matthew Hesson-McInnis for his willingness to take over Joel’s role as my committee chair on paper after Joel moved to Temple University and providing me valuable feedback. I thank professor Thomas Critchfield and professor J. Scott Jordan for their inspiring and critical comments.

I would also like to thank the department of psychology and the student counseling center for providing me with financial support and research experience. How fortunate and proud I am to have studied here and been part of it.

Most importantly, I would also like to thank my Choupipi, Xiaoya Zhang, for her love and caring. We are guardian angels for each other and I fear nothing because of your love. Our love is beyond words and everyone's understanding. I am also deeply indebted to my parents, DENG Jing and JI Shu-min: being your son is my fortune of life and you raised me to be independent and positive.

F. J.
## TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Cognitive Ability Test Scores from a Hypothetical Case</td>
<td>2</td>
</tr>
<tr>
<td>2. The Process of Generating Two Parameters of a Beta Distribution</td>
<td>16</td>
</tr>
<tr>
<td>3. Mahalanobis and Conditional Mahalanobis Probabilities for Example A</td>
<td>29</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.</td>
<td>A simplified prediction model that could be used in psychoeducational assessment (Model A in the later simulation study)</td>
</tr>
<tr>
<td>2.</td>
<td>Model A: A simplified prediction model that could be used in psychoeducational assessment</td>
</tr>
<tr>
<td>3.</td>
<td>Model B: Three cognitive predictors of math and reading ability</td>
</tr>
<tr>
<td>4.</td>
<td>For Model A, the observed distribution of $d_{CM}^2$ (red) overlaps almost perfectly with the theoretically expected distribution $\chi^2$ (black)</td>
</tr>
<tr>
<td>5.</td>
<td>For Model B, the observed distribution of $d_{CM}^2$ (red) overlaps almost perfectly with the theoretically expected distribution $\chi^2$ (black)</td>
</tr>
<tr>
<td>6.</td>
<td>The relationship between $d_{CM}$ estimated from factor scores and the $d_{CM}$ calculated from latent construct scores is nearly linear</td>
</tr>
<tr>
<td>7.</td>
<td>The accuracy of the Conditional Mahalanobis distance on Model A increases as a function of the average factor loading of the measurement model and decreases as a function of the average coefficients of the structural model</td>
</tr>
<tr>
<td>8.</td>
<td>The accuracy of the Conditional Mahalanobis distance on Model B increases as a function of the average factor loading of the measurement model and decreases as a function of the average coefficients of the structural model</td>
</tr>
<tr>
<td>9.</td>
<td>Comparison of the accuracy of $d_{CM}$ using Thurstone and composite scores on Model A</td>
</tr>
<tr>
<td>10.</td>
<td>The accuracy of $d_{CM}$ using Thurstone and composite scores on Model B</td>
</tr>
<tr>
<td>11.</td>
<td>Comparison of the performance of the conditional Mahalanobis distance using different numbers of indicators on Model A</td>
</tr>
<tr>
<td>12.</td>
<td>Comparison of the performance of the conditional Mahalanobis distance using different numbers of indicators on Model B</td>
</tr>
<tr>
<td>13.</td>
<td>Hypothetical Example A</td>
</tr>
<tr>
<td>14.</td>
<td>Hypothetical Example B</td>
</tr>
<tr>
<td>15.</td>
<td>Hypothetical Example C</td>
</tr>
</tbody>
</table>
CHAPTER I: THE PROBLEM AND ITS BACKGROUND

Statement of the Problem

In psychological assessments, practitioners not only collect measures of outcome variables (e.g., depression, reading ability, academic achievements, and job performance), but also measures of variables that predict or explain those outcomes. For example, if a child is a poor reader, the practitioner would assess a variety of potential explanatory variables (e.g., intelligence, auditory processing, instruction quality and attitudes toward education in the home). These explanatory variables are used in conjunction to understand why the child is having difficulty reading. These multiple-variable results are often obtained by collecting data from multiple batteries (e.g. Woodcock-Johnson Tests of Cognitive Abilities, Woodcock, McGrew, & Mather, 2001; Wechsler Adult Intelligence Scale, Wechsler, 1991).

Suppose that a child is referred to a clinician because the child is a slow reader despite adequate instruction and support for learning in the home. Table 1 shows cognitive and academic ability test scores from a hypothetical case, similar to the kinds of test data that clinicians encounter in their daily practice. Each of the six abilities in the table was measured with 3 independent tests from which a composite score was calculated. The child’s reading comprehension (RC) and reading decoding (RD; the ability to read single words accurately) was average, but the child’s reading fluency (RF; the speed at which text can be read accurately) was low. To try to understand the nature of the child’s reading difficulties, the clinician measured crystallized intelligence (Gc; knowledge and language development), fluid intelligence (Gf; reasoning ability and novel problem-solving) and processing speed (Gs; the speed at which attention can fluently shift while performing an attention-demanding task). Although the child has average or better language skills (Gc) and reasoning ability (Gf), her processing speed (Gs) was a little below average.
Table 1. Cognitive Ability Test Scores from a Hypothetical Case

<table>
<thead>
<tr>
<th>Test</th>
<th>Score</th>
<th>Percentile</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystallized Intelligence (Gc)</td>
<td>119</td>
<td>90</td>
<td>High Average</td>
</tr>
<tr>
<td>Gc1</td>
<td>109</td>
<td>73</td>
<td>Average</td>
</tr>
<tr>
<td>Gc2</td>
<td>126</td>
<td>96</td>
<td>High</td>
</tr>
<tr>
<td>Gc3</td>
<td>116</td>
<td>86</td>
<td>High Average</td>
</tr>
<tr>
<td>Fluid Intelligence (Gf)</td>
<td>101</td>
<td>53</td>
<td>Average</td>
</tr>
<tr>
<td>Gf1</td>
<td>94</td>
<td>34</td>
<td>Average</td>
</tr>
<tr>
<td>Gf2</td>
<td>109</td>
<td>73</td>
<td>Average</td>
</tr>
<tr>
<td>Gf3</td>
<td>100</td>
<td>50</td>
<td>Average</td>
</tr>
<tr>
<td>Processing Speed (Gs)</td>
<td>90</td>
<td>25</td>
<td>Low Average</td>
</tr>
<tr>
<td>Gs1</td>
<td>82</td>
<td>12</td>
<td>Low Average</td>
</tr>
<tr>
<td>Gs2</td>
<td>97</td>
<td>42</td>
<td>Average</td>
</tr>
<tr>
<td>Gs3</td>
<td>96</td>
<td>39</td>
<td>Average</td>
</tr>
<tr>
<td>Reading Decoding (RD)</td>
<td>100</td>
<td>50</td>
<td>Average</td>
</tr>
<tr>
<td>RD1</td>
<td>104</td>
<td>61</td>
<td>Average</td>
</tr>
<tr>
<td>RD2</td>
<td>91</td>
<td>27</td>
<td>Average</td>
</tr>
<tr>
<td>RD3</td>
<td>105</td>
<td>63</td>
<td>Average</td>
</tr>
<tr>
<td>Reading Comprehension (RC)</td>
<td>105</td>
<td>63</td>
<td>Average</td>
</tr>
<tr>
<td>RC1</td>
<td>110</td>
<td>75</td>
<td>High Average</td>
</tr>
<tr>
<td>RC2</td>
<td>100</td>
<td>50</td>
<td>Average</td>
</tr>
<tr>
<td>RC3</td>
<td>103</td>
<td>58</td>
<td>Average</td>
</tr>
<tr>
<td>Reading Fluency (RF)</td>
<td>77</td>
<td>6</td>
<td>Low</td>
</tr>
<tr>
<td>RF1</td>
<td>73</td>
<td>4</td>
<td>Low</td>
</tr>
<tr>
<td>RF2</td>
<td>82</td>
<td>12</td>
<td>Low Average</td>
</tr>
<tr>
<td>RF3</td>
<td>86</td>
<td>18</td>
<td>Low Average</td>
</tr>
</tbody>
</table>

Note: All standard scores have a mean of 100 and a standard deviation of 15. Composite scores are bolded.

At first glance, it may seem that an explanation for the child’s reading difficulty has been found. That is, the child’s core cognitive abilities (Gf and Gc) are average or better, which is consistent with average reading decoding and reading comprehension skills. The child has lower than average processing speed (Gs), which is consistent with low reading fluency (i.e., slow cognitive processing contributes to slow reading). However, not all intuitively plausible explanations are consistent with reality.

As will be shown, the seemingly adequate intuitive explanation from this hypothetical case is misleading. To make a more statistically plausible explanation, clinicians would need to be able to know first whether the profile of academic scores is unusual. If it is not, then there is nothing to explain. If the
academic scores need to be explained, the clinician would need to know if the profile of cognitive scores is unusual. If the profile of cognitive scores is ordinary, then it cannot be an explanation for unusual academic outcomes. If an atypical profile of academic scores is typical for people with an unusual profile of cognitive scores, then the cognitive scores have in a sense explained the atypicality of the academic scores.

Suppose that the scores from Table 1 were consistent with the structural model shown in Figure 1 and that all scores exhibit multivariate normality. Following the procedures described in this thesis, the clinician would discover that fewer than 3% percent of children with the same profile of cognitive ability scores as this child would have a score of 77 or less in reading fluency. Thus, although this child has a below-average processing speed (Gs), his/her reading fluency is unusually low even after the low processing speed has been accounted for. In this case, further investigation is needed to understand why the child reads slowly. At best, the child’s mild weakness in processing speed is only a partial explanation for the poor reading fluency performance. Unfortunately, in practice, few clinicians have access to the tools or knowledge to make statistically informed decisions like this, and thus must rely on error-prone heuristics and untrained intuition.

To make it possible for clinicians to be able to make statistically plausible explanatory statements about individuals, I will outline a novel set of procedures intended to provide a new way for clinicians to make sense of multivariate data (such as test results from cognitive assessment batteries such as the Wechsler Individual Achievement Test–Third Edition; WIAT-III). I will explain the underlying mathematics behind these procedures and will explain how to use newly developed software that implements and automates the processes outlined in this thesis.
Figure 1. A simplified prediction model that could be used in psychoeducational assessment (Model A in the later simulation study)
CHAPTER II: BACKGROUND KNOWLEDGE

The Superiority of Statistical-based Judgment

There have been debates on how to explain and use multivariate data about individuals properly (Grove & Meehl, 1954/1996; White, 2006). To process information effectively and correctly to make proper diagnoses or predictions, one can combine information using clinical judgment or applying formal algorithms (Grove & Meehl, 1996). Clinical judgment refers to the practice of thinking about the evidence in a holistic manner and coming to a decision, usually without being able to articulate each step in one’s reasoning. To be able to give a full accounting of one’s reasoning requires that the practitioner not only understands how each explanatory variable is related to the outcome but also how each explanatory variable is related to every other explanatory variable. Statistical procedures use algorithms and methods (e.g. multiple regression, discriminant analysis) to combine test data in terms of, usually, probabilistic-based indicators to quantify the results.

Meehl (1954) was an early entrant into the debate about the merits of formal, algorithmic decisions over clinical judgement. He evaluated twenty studies that used both methods and found in all cases but one that the statistical method equaled or outperformed the clinical method. In the following years, there have been many studies indicating a consistent conclusion that statistical methods typically outperforms clinical methods (White, 2006). Meehl (1996) explained why the statistical method is generally more accurate: the influence of unreliable data combination, which was illustrated by the Goldberg paradox. Goldberg (1970) formed least-squares regression and calculated the regression weights by treating the clinician’s ratings as the criterions and the eleven MMPI scores as the predictors. These “strangely” derived weights were then used to differentiate psychotic from neurotic patients, and he found this method outperformed the judgements of clinicians themselves for each clinician. The reason why it occurs is because of rate unreliability: when clinicians make judgements on their own, they apply their weights inconsistently every time, which leads to a worse performance on average. The phenomenon of the Goldberg paradox has been replicated in 15 subsequent studies for a variety of predictive purposes (Camerer, 1981). What made this result less interesting was that Dawes and Corrigan (1974) also found
the randomly-generated weights perform equally well as the models with clinicians’ judgements. Meehl (1996) pointed out the reason behind this situation is that the human brain is not designed for optimizing, selecting, assigning and manipulating information to make predictions; rather, it is a fairly weak device for these types of tasks. The superiority of formal, actuarial, statistically-based methods has been obvious from the historical point of view, as the scientific method of accumulating knowledge in the post-Galilean era has contributed to valid knowledge and astonishing achievements.

The conclusion from these studies illustrates the point that usually we require the aid of a statistical model to summarize and simplify the multivariate relations among the variables and the statistical methods can bring us a better way to make clinical judgements.

As the evidence that statistical methods’ superiority starts to accumulate, there have been many studies to advocate the usage of statistical methods for clinicians to make clinical judgements to improve the accuracy and efficacy of their treatment (Grove & Meehl, 1996). However, one technical problem that prevents practitioners from applying statistical methods is that the statistical knowledge behind the scene, which intimidates many of practitioners to still stick with their own judgement (Grove & Meehl, 1996).

**Unusualness as a Measure of “Effect Size” for Individual Data**

As stated in the last section, statistical-based judgements have a clear advantage. However, there are plenty of ways to represent statistical results. I chose what Schneider (2013) described as unusualness, which is a way for clinicians to make sense of statistical results, because a percentage-based (e.g. how likely we observe what we observe) approach is both straightforward, intuitive, and congruent with the nature of intelligence tests. The example from chapter one, the unusualness of reading fluency for that person was the percentage of people with the same Gc,Gs and Gf scores who will have a score of reading fluency as or more extreme.

**The Problem of Diagnosis in Psychological Assessment**

Structural equation modeling (SEM) is one widely-used method that was designed to impute multivariate relations among latent variables, which is especially useful in psychological assessment to
describe multiple relations between cognitive abilities and academic abilities. Problems arise when practitioners use SEM-based test batteries: Specifically, there are two key problems:

The first problem is that scholars frequently present new structural models to illustrate new theoretical advances, for example, Niilekesia (2016) found new validity evidence for the Woodcock-Johnson IV (WJ IV; Schrank, McGrew, & Mather, 2014). Scholars sometimes argue that the structural models they present are practically meaningful and helpful for psychological assessment professionals. However, being able to understand SEM-based research advances and translate and apply them into applied work presents a steep learning curve that is prohibitive to many if not most practitioners. In this paper, I proposed a set of methods that are intended to make it easier for practitioners to use latent variable models without needing to understand the methodological details. Specifically, I developed methods of predicting individual outcomes with SEM and evaluating how unusual a given pattern of outcome profiles is, contingent on a set of predictor variables.

The second problem is that the focus of most SEM methods is on population-level statistics, whereas most practitioners deal with individuals, whose profile of individual differences may or may not resemble the population model. For example, at the population level, vocabulary and reading ability are strongly correlated, but many individuals have a strong vocabulary but poor reading ability (or vice versa). Practitioners often wonder why the population trend is relevant when working with a person with an unusual pattern of individual difference variables. One benefit of applying structural equation models to such an individual is that the model can help estimate just how unusual the person is. Without the aid of the model, practitioners can mistakenly assume that common patterns are rare or that rare patterns are common. When practitioners make diagnostic errors, intervention resources can be misallocated from those more in need of help to those less in need of help or the wrong resources are provided, ones that that fail to address the clients’ needs.
CHAPTER III: METHODOLOGY

Schneider (2013) proposed an approach that provides clinicians a user-friendly way to extract useful information from confirmatory factor analysis (CFA) models that can be applied to individuals—CFA is a special case of SEM. I extend this method to structural equation models so that practitioners can use estimated latent variable scores as explanations of outcome variables.

**Factor Scores Estimation**

To make use of SEM for individual cases, factor scores are needed to conduct subsequent analyses. However, because any set of observed scores can arise from many combinations of factor scores, factor scores are said to be *indeterminate*. That is, one never knows which combination of latent factor scores produced the scores one observes, though some combinations are more likely than others.

There are many different ways to extract factor scores, including Thurstone’s (1935) regression-based scores, Bartlett’s approach (Bartlett, 1937), and empirical Bayes random effects estimates from mixed effects modeling (Estabrook & Neale, 2013). Each of these methods have distinct advantages and disadvantages. For example, using Bartlett’s approach, one can produce factor score estimates that have nearly the same covariance matrix as the actual factor scores, which is desirable if one wants to use the factor score estimates in subsequent group-level analyses. The disadvantage of these scores is that they are relatively less accurate estimates of latent variables at the individual level. Thurstone’s regression method has the opposite profile of advantages and disadvantages. Thurstone’s method maximizes the factor scores’ validity coefficient, which is defined as the correlation between the factor score estimates and the true scores in simulation studies (Grice, 2001; DiStefano et al., 2009; Mulaik, 2009). This advantage comes at a cost: The factor score estimates no longer have the same covariance matrix as the original factor scores, making them less suitable for subsequent group-level analyses. Empirical Bayes estimates are generally less accurate than Thurstone scores but perform better in cases in which data are missing (Estabrook & Neale, 2013).
For this study, I chose to use Thurstone’s method because its advantages align with the goals of practitioners: to estimate an individual’s scores as accurately as possible. The Thurstone regression method is implemented by the formula:

\[
\hat{x}_i = R_{XX} \Lambda R^{-1}_{YY} y_i = R_{XY} R^{-1}_{YY} y_i,
\]

where

\( \hat{x}_i \) is a random vector of the \( i \)th subject’s factor scores estimate on the common factors

\( R_{XX} \) is the correlation matrix among the latent factors

\( \Lambda \) is the factor-pattern matrix

\( R_{XY} \) is a correlation matrix between the common factors and the observed variables

\( R^{-1}_{YY} \) is the inverse of the correlation matrix among the observed variables

\( y_i \) is a random vector of the \( i \)th subject’s standardized scores on the observed variables

**The Mahalanobis Distance as a Measure of Profile Usualness**

If we know the probability distribution of the factor scores, it is possible to calculate how unusual each individual score is. However, the information for one score could be limited, as in the example in Figure 1, we know the information of how likely a person with the same Gc, Gs and Gf would have a reading fluency of 77. We sometimes want to know, not only the single outcome (reading fluency), but also a profile that consists of multiple variables (reading fluency, reading decoding and reading comprehension). That is, we would like to know how likely in the population a person with the same Gc, Gs and Gf would have the reading profile that we observed. When practitioners need make predictions or arrive at a diagnosis, it would be useful for them to also know how unusual an observed profile of scores is. I will refer to *profile usualness* as how frequently a given profile is observed in the population.

To obtain probabilistic information of multiple scores in one profile, the Mahalanobis Distance (hereinafter to be referred as \( d_M \), Mahalanobis, 1937) is of help, which is a generalized multidimensional way to measure the distance between two vectors in standard deviation units.

\[
d_M = \sqrt{(X - \mu)'C^{-1}(X - \mu)}
\]
where \( \mathbf{X} \) is a column vector of an individual’s scores on \( k \) tests, \( \mathbf{\mu} \) is a column vector of the population means of the \( k \) tests, and \( \mathbf{C} \) is the covariance matrix of the \( k \) tests.

For multivariate normal data with \( k \) variables, we can assume \( d_M^2 \) follows a \( \chi^2 \) distribution with \( k \) degrees of freedom. If the scores have a multivariate normal distribution, we can use the Mahalanobis distance to estimate how unusual a particular set of scores is. For the IQ example above, we would be able to know how likely to observe a person with the math and reading scores.

**Conditional Mahalanobis Distances**

Sometimes we wish to know more than just the overall profile usualness, but the profile usualness for people from a specific population (e.g. the people with a learning disability) or people with a particular set of cognitive strengths and weaknesses. For example, Figure 2 illustrates the relations between the measures of outcome variables (i.e., Decoding, Reading Comprehension and Reading Fluency) and potential explanatory variables (i.e., Crystallized Intelligence (Gc), Fluid Intelligence (Gf), Processing Speed (Gs)) in a structural equation model. A conditional version of Mahalanobis distance measures the unusualness of the profile of measures of outcome variables conditioned on the values of a set of explanatory variables.

In this sense, it may be more useful to look at the Conditional \( d_M \), which measures the profile usualness of a set of variables of interest conditioning on another set of explanatory variables (hereinafter to be referred as \( d_{CM} \), Schneider, 2017). The \( d_{CM} \) statistics is useful to form conditional distributions with multiple constraints. With \( d_{CM} \), we will be able to form the distributions conditioned on some of estimated predictor factor scores and make inferences from the conditional distribution of outcome factor scores in that we can calculate probabilities of a given profile for a given individual and the structural model. Put another way, provided a particular structural model, we can estimate the distributions of estimated endogenous factor scores conditioned on multiple estimated exogenous factor scores. **Exogenous variables** refer to variables that are not influenced by other variables in the model, whereas **endogenous variables** are influenced by other variables in the model. Note that an endogenous variable may also cause
another endogenous variable in the model. With $d_{CM}$, we are able to estimate the profile usualness of endogenous factor scores.

![Diagram of Model A]

Figure 2. Model A: A simplified prediction model that could be used in psychoeducational assessment

We assume that all exogenous factor scores ($\mathbf{v}_x$) and exogenous factor scores ($\mathbf{v}_x$) have a multivariate normal distribution $\mathbf{V} \sim \mathcal{N} (\mathbf{u}, \Sigma)$

$$
\mathbf{V} = \begin{bmatrix} \mathbf{v}_y \\ \mathbf{v}_x \end{bmatrix}
$$

$$
\mathbf{u} = \begin{bmatrix} \mathbf{u}_y \\ \mathbf{u}_x \end{bmatrix}
$$

$$
\Sigma = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{bmatrix}
$$
When $\mathbf{v}_x$ equals to some profile $\mathbf{x}$, we could find the conditional multivariate distribution of $\mathbf{V}$ given $\mathbf{v}_x = \mathbf{k}$, that is, $(\mathbf{v}_y | \mathbf{v}_x = \mathbf{x})$ with mean

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_y + \Sigma_{yx} \Sigma_{xx}^{-1} \mathbf{x}$$

And conditional covariance matrix

$$\bar{\Sigma} = \Sigma_{yy} - \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy}$$

So, in general, the conditional Mahalanobis distance is

$$d_{CM} = \sqrt{\mathbf{v} - \bar{\mathbf{u}})^{\top} \bar{\Sigma}^{-1} (\mathbf{v} - \bar{\mathbf{u})}}$$

Where

- $\mathbf{v}$ the vector of observed scores.
- $\bar{\mathbf{u}}$ is the mean vector of the conditional normal distribution.
- $\bar{\Sigma}$ is the matrix of the variance of the conditional normal distribution.

For our specific question in psychological assessment, we could regard the set of factor scores of cognitive abilities as $\mathbf{v}_x$, and the set of factor scores of academic abilities as $\mathbf{v}_y$ so that we could examine where the predicted factor scores are located in the multivariate normal conditional distribution.

If multivariate normality can be assumed and there are $k$ outcome scores

$$d_{CM} = \sqrt{(\mathbf{z}_o - \bar{\mathbf{z}}_o)^{\top} \bar{\mathbf{R}}_o^{-1} (\mathbf{z}_o - \bar{\mathbf{z}}_o)}$$

Where

- $\bar{\mathbf{z}}_o$ is the vector of predicted outcome scores (i.e., the predicted academic abilities predicted by the factor scores of cognitive abilities).
- $\mathbf{z}_o$ is the vector of outcome scores (i.e., the factor scores of academic abilities estimated from our SEM).
\( R_g^{-1} \) is the matrix of conditional variance among the factor scores (the composite correlation between factor scores calculated using population correlation among observed scores, that is, \( \beta'\Sigma_{xx}\beta \), where \( \beta = \Sigma_{yx}\Sigma_{xx}^{-1} \).

If multivariate normality can be assumed and there are \( c \) outcome scores,

\[ d^2_{cM} \sim \chi^2(k) \]

Returning to the IQ, reading and math score question example, we can find the unusualness by calculating \( d_{cM} \) and form \( \chi^2 \) distribution to understand how unusual the observed individual profile is in the population with an IQ at 15\textsuperscript{th} percentile.

It is my hope to create an easily interpretable and generalized approach to extend Schneider’s (2013, 2017) method to the more generalized level into the context of SEM and being able to create distributions condition on multiple constraints. Although this idea was created originally for solving the problems in psychological assessment, it is also my hope that this method could be utilized beyond psychological assessment as a generalized approach to give individualized explanations in SEM so that it is not specifically for clinicians in psychological assessment but also for practitioners in the other fields where structural equation models are employed to advance understanding and illustrate theories.

**The Current Study**

The unusualprofile package was created in the R programming environment (R Core Team, 2017) to automate calculating factor scores, and conditional Mahalanobis distances \( d_{cM} \) in a user-specified structural equation model.

Because of the complexity and ongoing arguments over technical details of SEM, we focus on recursive models in our current study, which refer to the models with no feedback loops or correlated disturbances.

Simulation studies were conducted to test how the conditional Mahalanobis distance perform under different conditions and models. Comparisons of conditional Mahalanobis distance calculated using
Thurstone (regression) method and commonly-used equally-weighted method (composite scores) were conducted to examine the superiority of using the proposed method over the common practice.

**Data Generation**

Data was generated using the RAM model (the Reticular Action Model, McArdle & MacDonald, 1984), which is a simplified SEM notation that consists of three matrices $A$, $S$, and $F$ matrices, where:

- $A$: The Asymmetric matrix contains all direct paths.
- $S$: The Symmetric matrix contains all correlations/covariance and residual variances.
- $F$: The Filter matrix selects all the observed variables in a matrix.

If the number of total variables in a model is $m$, and $n$ out of $m$ are measured, the dimensions of the three matrices are $A = m \times m$, $S = m \times m$, $F = n \times m$. In most applications, the $F$ matrix filters out the latent variables. In this context, we need to leave the latent variables in place, and thus only the $A$ and $S$ matrix from the RAM model are used.

The implied correlation matrix $R$ is calculated by the equation

$$ R = (I - A)^{-1} S (I - A)^{-1} $$

Let $\mathbf{v}$ be a $t \times 1$ vector of random variables, and let

$$ \mathbf{v} = A \times \mathbf{v} + \mathbf{u} $$

Let $\alpha_{ij}$ refer the elements of $A$ matrix and be the coefficient of the variable on column $i$ on row $j$.

Let $\mathbf{u}$ refer the residual of $\mathbf{v}$.

A and $S$ have a relation:

$$ S = E \{ \mathbf{uu}' \} $$

$$ \mathbf{u} = \mathbf{v} - A \mathbf{v} = (I - A) \mathbf{v} $$

so $\mathbf{v} = (I - A)^{-1} \mathbf{u}$

$\mathbf{u}$ can be generated using the `mvtnorm` package (Genz et al., 2015) in the R statistical programing environment for a given $S$ and $A$. 

14
Two models were examined in the current study: a hypothetical model that consists of both academic abilities and cognitive abilities, Model A (Figure 2) and Model B (Figure 3).

![Figure 3. Model B: Three cognitive predictors of math and reading ability](image)

**Parameter Generation**

Parameters in models for simulation were generated using the beta distribution because it always generates numbers between 0 and 1, which is useful for generating random standardized coefficients.

The beta distribution with two parameters: shape1 ($\alpha$) and shape2 ($\beta$) is given by:

\[
f(x) = \frac{x^{\alpha-1} (1-x)^{\beta-1}}{B(\alpha, \beta)}
\]

where $B$ is the beta function, i.e., $B(x, y) = \int_0^1 t^{x-1} (1 - t)^{y-1} dt$
Every structural equation model can be divided into a measurement model (the factor loadings of the observed variables) and the structural model (the predictive relations among the variables of interest). For each model, parameters were generated for the measurement model and structural model separately (i.e., factor loadings and standardized latent structural coefficients, respectively). The beta distribution was used to produce factor loadings and standardized latent structural coefficients. The two parameters of the beta distribution were generated as follows (Table 1) and which yields a range of factor loadings whose average value from 0.1 to 0.9. For every condition, observed scores of 1000 people were generated and the correlation between estimated Mahalanobis distance and true Mahalanobis distance were calculated as the accuracy index, totaling 1,000,000 (1000 conditions 1000 × individuals) individual data points for one model.

Table 2. The Process of Generating Two Parameters of a Beta Distribution

<table>
<thead>
<tr>
<th>Conditions</th>
<th>α</th>
<th>β</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition 1</td>
<td>$1000 + \frac{8000}{1000} \times 0$</td>
<td>$9000 - \frac{8000}{1000} \times 0$</td>
</tr>
<tr>
<td>Condition 2</td>
<td>$1000 + \frac{8000}{1000} \times 1$</td>
<td>$9000 - \frac{8000}{1000} \times 1$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Condition n</td>
<td>$1000 + \frac{8000}{1000} \times n$</td>
<td>$9000 - \frac{8000}{1000} \times n$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Condition 999</td>
<td>$1000 + \frac{8000}{1000} \times 999$</td>
<td>$9000 - \frac{8000}{1000} \times 999$</td>
</tr>
<tr>
<td>Condition 1000</td>
<td>$1000 + \frac{8000}{1000} \times 1000$</td>
<td>$9000 - \frac{8000}{1000} \times 1000$</td>
</tr>
</tbody>
</table>

Standard

The simulation study was designed to evaluate the accuracy index of conditional Mahalanobis distance under different situations for continuous data, which is defined as the correlation between the true conditional Mahalanobis distance and the one calculated from factor scores.
CHAPTER IV: RESULTS

Do the Simulated $d_{CM}$ Distributions Conform to Theoretical $d_{CM}$?

To make valid inferences, we expected to see consistency between empirical distributions for the $d_{CM}^2$ generated by the simulated data and theoretical $\chi^2$ distributions. For Model A, there are 3 latent exogenous variables and 3 latent endogenous variables. The distribution of the $d_{CM}^2$ on the simulated endogenous latent scores after controlling for the exogenous latent scores should have a $\chi^2$ distribution with 3 degrees of freedom. Simulated latent scores for 1,000,000 cases were generated according to the specifications of Model A, and $d_{CM}$ was calculated for each case. In Figure 4, the simulated distribution of $d_{CM}^2$ matches the theoretical distribution almost perfectly. The same procedure was completed for Model B, which has 2 latent endogenous variables. As seen in Figure 5, the simulated $d_{CM}^2$ scores have a $\chi^2$ distribution with 2 degrees of freedom.

![Figure 4](image)

Figure 4. For Model A, the observed distribution of $d_{CM}^2$ (red) overlaps almost perfectly with the theoretically expected distribution $\chi^2$ (black)
Figure 5. For Model B, the observed distribution of $d_{CM}^2$ (red) overlaps almost perfectly with the theoretically expected distribution $\chi^2$ (black)
How Accurate is the $d_{CM}$ Computed from Estimated Factor Scores?

If scores are generated according to Model A in Figure 1, the $d_{CM}$ can be computed from both the latent construct scores and also from the estimated factor scores. The correlation between these sets of $d_{CM}$ values represents the accuracy of $d_{CM}$ computed from estimated factor scores. In this case, the correlation was 0.85. As seen in Figure 6, the relationship between the true $d_{CM}$ and $d_{CM}$ estimated from factor scores is nearly linear. Although not shown, this relationship was also nearly linear in every other model I tested.

![Figure 6. The relationship between $d_{CM}$ estimated from factor scores and the $d_{CM}$ calculated from latent construct scores is nearly linear.](image)

The accuracy of the $d_{CM}$ computed from estimated factor scores depends on several features of the latent model. If scores are generated for Model A, and the path coefficients are systematically varied, one can see how the accuracy of $d_{CM}$ computed from estimated factor scores changes. In Model A, Processing Speed (Gs), Fluid Reasoning (Gf), and Crystallized Intelligence (Gc) are cognitive predictors,
and Reading Decoding, Reading Fluency, and Reading Comprehension are academic outcomes. For this
analysis, I will distinguish between three types of structural coefficients:

- Cognitive-to-cognitive relationships (e.g., Gs → Gf)
- Cognitive-to-academic relationships (e.g., Gc → Reading Decoding)
- Academic-to-academic relationships (e.g., Reading Decoding → Reading Comprehension)

Each of these structural coefficient types, along with each construct’s factor loadings, was varied
systematically. Using each model, data were simulated, factor scores were estimated, and the accuracy of
the $d_{CM}$ was computed. This process was duplicated for Model B resulting in similar findings. Informally,
many other structural models were tested, and the pattern of results was the same each time.

Not all combinations of structural coefficients could be evaluated because they resulted in more
than 100% of the variance of a construct being explained, which is impossible. For example, if Gc and Gf
correlate at 0.9, they cannot both have a direct effect of 0.9 on Reading Decoding. With large, complex
models with many predictors, many outcomes, and many structural paths, the direct effects cannot rise
very high on average before the model explains more than 100% of the variance in at least one construct.
For example, Model A has three predictor constructs. If all three predictors are uncorrelated and have
equal direct effects on an academic outcome, the direct effect can be no higher than $\sqrt{1/3} \approx 0.577$. If the
predictors are positively correlated, the maximum average direct effect decreases quickly.

For Model A (see Figure 7) and Model B (see Figure 8), the accuracy of $d_{CM}$ increases if the
factor loadings increase. The reason for this relationship is straightforward: if the latent construct scores
are more accurately estimated, the accuracy of $d_{CM}$ increases. Counterintuitively, if the structural
coefficients increase, the accuracy of the $d_{CM}$ computed from estimated factor scores decreases. There are
two reasons for this effect:

1. When the cognitive-to-academic paths are high, the predictors explain more variance in the
   outcomes. Likewise, as the cognitive-to-cognitive paths increase, and all other paths are held
constant, the predictors explain increasing amounts of variance in the outcomes (i.e., correlated predictors explain more variance than analogous uncorrelated predictors with the same direct effects). Estimated factor scores consist of construct-relevant variance and measurement error. The $d_{CM}$ statistic measures the pattern of outcome variance after controlling for a set of predictors. That is, it measures the unusualness of the residualized outcome profiles. Because predictors can only explain construct-relevant variance in the outcomes, the residualized outcome scores increasingly consist of measurement error as the explained variance in the outcomes approaches 100%. Because measurement error cannot be predicted, the $d_{CM}$ becomes less accurate as the residualized outcome scores increasingly consist of measurement error.

2. When the academic-to-academic paths increase, the outcomes become more correlated. A well-known psychometric phenomenon is that the reliability of a difference score decreases when the correlation between the scores being compared increases (Guilford, 1954). From a certain point of view, a Mahalanobis distance is a kind of multivariate difference score. That is, a Mahalanobis distance measures the shape of a profile, and in doing so, the scores must be compared to each other, and their differences are summarized. Thus, except when the reliability of all scores is perfect, the accuracy with which one can estimate the shape of a latent profile decreases if the correlations among the profile scores increase.
Figure 7. The accuracy of the Conditional Mahalanobis distance on Model A increases as a function of the average factor loading of the measurement model and decreases as a function of the average coefficients of the structural model.
Figure 8. The accuracy of the Conditional Mahalanobis distance on Model B increases as a function of the average factor loading of the measurement model and decreases as a function of the average coefficients of the structural model.

**Comparing the Accuracy of \(d_{CM}\)**

The comparison of performance of conditional Mahalanobis distance using Thurstone factor score estimates and equally-weighted composite scores was conducted to illustrate the superior performance of our proposed method (Thurstone) over the commonly-used method (composite scores). Data were simulated for models A and B. When factor loadings of each construct are approximately equal, there is little advantage to using the Thurstone method over the simpler, equally-weighted composite score method because the Thurstone method creates estimates with nearly equal weights. However, when the within-construct variability of factor loadings is high, the Thurstone factor score method performs better than the composite method.

For example, in Figure 9, Thurstone factor score estimates and equally-weighted composite scores were created for each construct in Model A. The cognitive-to-cognitive effects and the cognitive-to-academic effects (labeled as the cognitive effects) were varied from 0 to 0.3 as were the academic-to-academic effects (labeled as the academic effects). Factor loadings were varied to have a mean of 0.01 to 0.99 but were sampled from the beta distribution with 10 total trials to create factor loadings with a high level of variability. For example, when the mean of beta distribution is 0.5 with 10 total trials (i.e., with shape parameters of 5 and 5), the average standard deviation of three random scores is about .13.
The results presented in Figure 9 and Figure 10 illustrate the small increase in precision one obtains when one uses Thurstone factor estimates instead of equally-weighted composites to estimate the $d_{CM}$. From the figures, we can see Thurstone’s method outperforms the equally-weighted composite scores in all conditions, particularly as the predictor variables have stronger effects on the outcome variables. If the variability of factor loadings were to be increased, the superiority of the Thurstone method would be greater than displayed in Figure 9 and Figure 10. If the factor loadings are made more equal, the superiority of the Thurstone method would disappear.

**How the Number of Factor Indicators Influences the Accuracy of $d_{CM}$**

The comparison of the accuracy of $d_{CM}$ using different numbers of indicators was conducted to examine how the accuracy changes under five conditions of numbers of indicators (i.e., 1, 2, 3, 4, 5). Five conditions of factor loadings were generated the same way described in Chapter III.

The results were presented in Figure 11 and Figure 12 for model A and B, respectively. From the figures, it is evident that the $d_{CM}$ become more accurate as the number of indicators increases. That is, as the latent constructs are more reliably estimated with more indicators per construct, the $d_{CM}$ is also more reliability estimated.
Figure 9. Comparison of the accuracy of $d_{CM}$ using Thurstone and composite scores on Model A
Figure 10. The accuracy of $d_{CM}$ using Thurstone and composite scores on Model B
Figure 11. Comparison of the performance of the conditional Mahalanobis distance using different numbers of indicators on Model A
Figure 12. Comparison of the performance of the conditional Mahalanobis distance using different numbers of indicators on Model B
CHAPTER V: CASE STUDIES

Purpose of Case Studies

A series of hypothetical case studies are presented in this section illustrating how different conditional Mahalanobis distances are used to help practitioners understand the unusualness of test profiles. All the following illustrations use observed and latent scores that are multivariate normal.

Table 3. Mahalanobis and Conditional Mahalanobis Probabilities for Example A

<table>
<thead>
<tr>
<th>Case</th>
<th>Gc_1</th>
<th>Gc_2</th>
<th>Gc_3</th>
<th>Estimated Gc</th>
<th>( p(d_M) )</th>
<th>( p(d_{CM}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130.92</td>
<td>.795</td>
<td>.046</td>
</tr>
<tr>
<td>Case 2</td>
<td>115</td>
<td>130</td>
<td>145</td>
<td>130.92</td>
<td>.998</td>
<td>.987</td>
</tr>
</tbody>
</table>

We can use the proposed method to make sense of observed scores. If the observed scores of the three tests measuring Gc are all 130 (Figure 13, Case 1), we can calculate the (unconditional) Mahalanobis distance to show that the profile of the three scores would be less usual than about 80% of the population (see Table 3). If the observed scores of the three tests measuring Gc are 115, 130 and 145 (Figure 13, Case 2), the profile would be less usual than 99.8% of population. We can see from this case that although they have equal averages, their unusualness is quite different. The problem with the traditional
Mahalanobis distance is that both profiles are unusual because the scores, on average, are 2 standard deviations above the mean. With the proposed conditional Mahalanobis distance (conditioned on the estimated factor score of 130.92), we would see in Table 3 that Case 2, with a variable profile, is still more unusual than 98.5% of profiles with the same factor score. Case 1—with equal scores conditioned on an equally weighted factor score—is the least unusual profile possible.¹ Thus, with the conditional Mahalanobis distance, one can distinguish between profiles that are unusual because of elevation and those that are unusual because of variability in their shape.

Profile B represents the result of an individual assessed on a cognitive and academic test battery. The estimated cognitive factor scores are all 115 and academic factor scores (i.e., reading) are all 85. With Mahalanobis distances and correspondent Chi-square distribution, we can know that the cognitive profile (i.e., 115 for Gs, Gf and Gc) by itself is more unusual than 47% of the population’s and this academic profile (i.e., 85 for Reading Comprehension, Reading Fluency and Reading Decoding) by itself is more unusual than 28% of population. When we only focus on the subpopulation with the cognitive profile above, however, the probability that an individual would have an academic profile this extreme or more extreme is less than 0.001%, which is exceedingly rare. So, although the cognitive profile and academic profile in and of themselves are not so rare, the two occurring in the same individual is extremely unlikely. That is, although the elevation of scores are not unusual, the configuration of scores given the relations between the constructs is extremely unusual. Be more explicit, it might be the time for

¹ The reason that the conditional Mahalanobis is not exactly zero in Case 1 is that the estimated factor score is 130.92, but the true factor score is unknown because the subtest errors are unknown. If the subtest errors were their most likely values, they would be exactly 0. In that case, the true factor score would be 133.33 (i.e., divide a subtest z-score by its loading and convert to the standard score metric). If 133.33 were the score on which the profile was conditioned, the conditional Mahalanobis distance would be exactly zero because each subtest would be perfectly predicted. Note that the correlation between the true factor score and the estimated factor score is 0.927. If we use the true factor score of 133.33 to “predict” the factor score, we get $0.927 \cdot (133.33 - 100) + 100 = 130.92$. Thus, the factor score estimate does not provide the lowest possible conditional Mahalanobis distance because of imperfect estimation (i.e., regression to the mean).
practitioners to delve into the potential explanations of the pattern of scores for individuals to find a meaningful explanation.

By contrast, the cognitive profile (Gc = 63, Gf = 78, Gs = 85) of Example C by itself is quite unusual, which is more unusual than 95% of population’s and the academic profile (Reading Decoding = 86, Reading Comprehension = 75, Reading Fluency = 84) is also unusual, more unusual than 85% of population. If we only take the people who have the cognitive profile above into account, however, the academic profile is only more unusual than 1% of the subpopulation (i.e., the subpopulation where people all have the cognitive profile above). That is, the academic profile is actually within expectations when we knew the cognitive profile of that person. In this case, the clinician could be satisfied that an explanation for the academic profile has been found. If the conditional academic profile were not within expectations, additional non-cognitive explanations would need to be considered.
Figure 14. Hypothetical Example B
Figure 15. Hypothetical Example C
CHAPTER VI: DISCUSSION AND LIMITATIONS

I developed a new method to help clinicians make sense of complicated multivariate relations among assessment results along with the software that automates the whole process. I conducted a series of simulations to examine how the proposed method works empirically and compared the performance with commonly-used method. It is found that:

1. The empirical distribution of Mahalanobis distance matches the theoretical chi-square distribution as intended, so it provides us evidence for using this method to make judgments.

2. The reliability of the conditional Mahalanobis distance is higher using the proposed Thurstone regression score method than the method using equally-weighted composite scores.

3. The reliability of the conditional Mahalanobis distance increases as a function of the average factor loading of the measurement model. Larger factor loadings allow for constructs to be estimated more accurately, which increases the accuracy of the conditional Mahalanobis distance.

4. When constructs are estimated with more indicators per construct, the constructs are estimated more accurately, which allows the conditional Mahalanobis distance to be estimated more accurately.

5. The accuracy of the conditional Mahalanobis distance decreases as a function of the average path coefficients of the structural model. The seemingly counterintuitive finding results has two causes.
   a. When factor score estimates of the predictor variables explain more variance in outcome variables, the remaining unexplained variance in the outcome factor score estimates consists of more error variance. Thus, the conditional Mahalanobis distance is less accurate because the conditional profile consists of more error variance.
   b. As is the case with traditional Mahalanobis distances, the conditional Mahalanobis distance is less accurately estimated when the outcome variables are highly...
correlated. The Mahalanobis distance can be thought of as a multivariate difference score. Just as difference scores of pairs of highly correlated variables are—all else equal—less reliable than pairs of less correlated variables (Overall & Woodward, 1975; Rogosa & Willett, 1983), the multivariate differences measured in Mahalanobis distances are less reliable when the variables are highly correlated than when they are less highly correlated. Although this feature of Mahalanobis distances might seem counterintuitive, one might consider the fact that highly correlated profiles tend to be flat. As the scores in a profile become increasingly correlated but the reliability is held constant, the variability in the profile is increasingly due to measurement error, which causes the conditional Mahalanobis distance based on factor score estimates to be less accurate.

**Advantages of This Method**

The proposed method provides an easy way for practitioners to comprehend the multivariate relations, which, as stated before, causes confusion and misunderstanding in explanations. Percentage based results summarizing multivariate relations were provided. This probabilistic summarizing information is believed to be more intuitive and easier to understand based on past literature in Chapter II.

As we can see from the results, the proposed model clearly outperforms the commonly-used way of summarizing multivariate relations (i.e., composite scores) in terms of accuracy of conditional Mahalanobis distance in the simulation study where different configurations of SEM with different latent structure coefficients and factor loadings were specified and the accuracy index were compared. In case studies, we can see that practitioners could gain extra information besides the commonly-seen single-value-based result.

The proposed method was created with the hope that clinicians can apply it to their daily practice. With more and multivariate-based information provided by this method, they can make clinical judgements as usual first and then apply the proposed method to test the plausibility of their case conceptualizations. If the results of their analyses are consistent with their case conceptualization, they
would have more confidence in implanting a treatment plan based on their case conceptualization; if not, they can seek additional information that would help them reformulate their case conceptualization. Where feasible, the procedures I have described could be repeated to test the plausibility of the new case conceptualization (e.g., with additional predictor variables added to the model).

The proposed method in this sense is designed to provide additional protection from making poor treatment decisions based on statistically-implausible case conceptualizations. If this method results in better case conceptualizations, future studies might reveal that practitioners who use these procedures allocate treatment resources more prudently because they are less likely to waste time, effort, and money implementing unnecessary interventions and more likely to select effective interventions that address the problems.

**Suggestions of Usage of the Method**

Although a consistent trend was seen in the three simulation studies I conducted for Model A and B, practitioners work with a wide variety of theoretical models with different specifications of factor loadings and latent structural coefficients. So, it is suggested to use the `boot_dCM` function in the `unusualprofile` package to get a sense of how accurate the conditional Mahalanobis distance could be and what the range of the 95% confidence interval of Mahalanobis distance accuracy is for the set of factor loadings and latent structural coefficients of the given population model. This `boot_dCM` function simulates data from a user-specified model and calculates the 95% confidence interval of the accuracy index, which gives us information about how well the proposed method works for any given model.

**Limitations**

There are several limitations in the current project. First, unusualness of the Conditional Mahalanobis distance is known with precision only when the data are multivariate normal. How it performs and how robust it is that when the assumption has been violated is unknown. Second, because there are any infinite number of possible structural equation models, it is impossible to test how the method performs in all situations. However, it is always good to know how well it performs in some widely-used models under different conditions of factor loadings and standardized latent structural
coefficients. Third, although the software accompanied is designed so that practitioners can use it to the minimum knowledge of programming and statistics, it still requires the installment of the R programming environment and some test types. A future version of the package with a user-friendly online interface is needed to make it more accessible to practitioners. Fourth, the non-linear relations between the accuracy and the factor loadings should be better addressed. A rigorous look using mathematics should be taken in the future. Fifth, because SEM has been developing and new extensions are frequently proposed, more research needs to be done to examine the performance and the explanation of conditional Mahalanobis distance in these situations (e.g. multilevel SEM, Rabe-Hesketh, Skrondal & Pickles, 2004; Bayesian SEM, Muthén & Asparouhov, 2010; Nonparametric SEM, Song & Lee, 2012).
REFERENCES


APPENDIX: PACKAGE SOURCE CODE AND SIMULATION CODE

Functions of the unusualprofile package

#' Function that takes a lavaan model with standardized parameters and
#' simulates latent scores, errors, disturbances, and observed scores
#' @export
#' @param m Structural model represented by lavaan Syntax
#' @param n Number of simulated cases
#' @param ObservedOnly Return only observed data
#' @return Latent scores, errors, disturbances, and observed scores
#' @examples
#' # Lavaan model
#' m = "Latent_1 =~ 0.8 * Ob_1 + 0.8 * Ob_2"
#' # simulate 10 cases
#' simStandardized(m, n = 10)

simStandardized < - function(m, n = 1000, ObservedOnly = FALSE)
{
  # Parameter Table
  pt <- lavParTable(m, fixed.x = F)
  
  # Variable Names
  vObserved <- lavNames(pt, "ov")
  vLatent <- lavNames(pt, "lv")
  vLatentExogenous <- lavNames(pt, "lv.x")
  vLatentEndogenous <- lavNames(pt, "lv.nox")
vObservedExogenous <- lavNames(pt, "ov.x")

vObservedEndogenous <- lavNames(pt, "ov.nox")

if (length(vLatentEndogenous) > 0) {
  vDisturbance <- paste0("d_", vLatentEndogenous)
}
else {
  vDisturbance <- character(0)
}

if (length(vObservedEndogenous) > 0) {
  vError <- paste0("e_", vObservedEndogenous)
}
else {
  vError <- character(0)
}

# Names for A, S and new S matrices
vA <- c(vLatentExogenous,
        vLatentEndogenous,
        vObservedExogenous,
        vObservedEndogenous)

vS <- c(vLatentExogenous,
        vLatentEndogenous,
        vObservedExogenous,
        vObservedEndogenous)

vnewS <- c(vLatentExogenous,
           vLatentEndogenous,
vObservedExogenous,
vObservedEndogenous,
vDisturbance,
vError)

# Number of Variables
k <- length(vA)

# Initialize A matrix and exogenous correlation matrix
ExoCor <- A <- matrix(0, k, k, dimnames = list(vA, vA))

# Assign loadings to A
for (i in pt[pt[, "op"] == "=~", "id"] ) {
  A[pt$rhs[i], pt$lhs[i]] <- pt$ustart[i]
}

# Assign regressions to A
for (i in pt[pt[, "op"] == "~", "id"] ) {
  A[pt$lhs[i], pt$rhs[i]] <- pt$ustart[i]
}

# Assign correlations to ExoCor
diag(ExoCor) <- 1
for (i in pt[pt[, "op"] == "~~", "id"] ) {
  if (pt$lhs[i] != pt$rhs[i]) {
    ExoCor[pt$lhs[i], pt$rhs[i]] <- ifelse(is.na(pt$ustart[i]),
    ...
\[
\theta,
\]
\[
\text{pt$u$tart[i])}
\]
\[
\text{ExoCor[pt$rhs[i], pt$lhs[i]] <- ExoCor[pt$lhs[i], pt$rhs[i]]}
\]

#Solving for error variances and correlation matrix

#Column of k ones
v1 <- matrix(1, k)

#Initial estimate of error variances
varS <- as.vector(v1 - (A * A) %*% v1)
S <- diag(varS) %*% ExoCor %*% diag(varS)

#Initial estimate of the correlation matrix
R <- solve(diag(k) - A) %*% S %*% t(solve(diag(k) - A))

# Set interaction count at 0
iterations <- 0

# Find values for S matrix
while ((round(sum(diag(R)), 10) != k) * (iterations < 100)) {
    iA <- solve(diag(k) - A)
R <- iA %*% S %*% t(iA)
sdS <- diag(diag(S) ^ 0.5)
S <- diag(diag(diag(k) - R)) + (sdS %*% ExoCor %*% sdS)
    diag(S)[diag(S) < 0] <- 0.0000001
iterations <- iterations + 1
}
if (iterations == 100) {
    stop("Maximum iterations reached (100).")
}

# Assign variable names to S
dimnames(S) <- list(vS, vS)

# Generate data frame

# Exogenous data
u <- rmvnorm(n, sigma = S)
colnames(u) <- vS

v <- u %*% t(iA)

#Simulated dataset

d <- as_tibble(cbind(v,
                u[, c(-1 *
match(c(vLatentExogenous,vObservedExogenous), vS))))
    colnames(d) <- vnewS
`dimnames(S) <- list(vS, vS)`

# Created extended A matrix

`extendA <- diag(diag(S) ^ 0.5)`

`dimnames(extendA) <- list(nlist(
  c(vLatentExogenous, vLatentEndogenous, vObservedExogenous, vObservedEndogenous),
  c(vLatentExogenous, vDisturbance, vObservedExogenous, vError)
))`

# Remove exogenous variables

`extendA <- extendA[, c(vDisturbance, vError)]`

# bind A and extended A

`extCol <- cbind(A, extendA)`

# Append zeros so that new A will be square

`extRow <- matrix(0, nrow = ncol(extendA),`
\[ \text{ncol} = \text{ncol}(A) + \text{ncol}(\text{extendA}) \]

\[ \text{newA} \leftarrow \text{rbind}(\text{extCol}, \text{extRow}) \]

\[ \text{dimnames(newA)} \leftarrow \text{list}(c(\text{colnames}(A), \text{colnames}(\text{extendA})), \]
\[ c(\text{colnames}(A), \text{colnames}(\text{extendA}))) \]

# build a S matrix with 1s and 0s on the diag

\[ \text{newS} \leftarrow \text{diag}(c( \]
\[ \text{rep}(1, \text{length}(\text{vLatentExogenous})), \]
\[ \text{rep}(0, \text{length}(\text{vLatentEndogenous})), \]
\[ \text{rep}(1, \text{length}(\text{vObservedExogenous})), \]
\[ \text{rep}(0, \text{length}(\text{vObservedEndogenous})), \]
\[ \text{rep}(1, \text{length}(\text{vDisturbance})), \]
\[ \text{rep}(1, \text{length}(\text{vError}))) \]

\[ \text{dimnames(newS)} \leftarrow \text{list}(\text{vnewS}, \text{vnewS}) \]

# Insert all off-diagonal covariances

\[ \text{ExoCor} \leftarrow \text{newS}[c(\text{vLatentExogenous}, \]
\[ \text{vObservedExogenous}, \]
\[ \text{vDisturbance}, \]
\[ \text{vError}), \]
\[ c(\text{vLatentExogenous}, \]
\[ \text{vObservedExogenous}, \]
\[ \text{vDisturbance}, \]
R <- \texttt{solve(diag(nrow(newA)) - newA)} \%\% 
newS \%\% 
= \texttt{t(solve(diag(nrow(newA)) - newA))} 

Rxx <- R[vObserved, vObserved] 
Rxy <- R[vObserved, c(vLatent, vDisturbance, vError)] 
iRxx <- \texttt{solve(Rxx)}

\textbf{if} (ObservedOnly) {
    d[, vObserved]
} \textbf{else} {
    l <- \texttt{list(}
    \hspace{1em} \texttt{Data} = d,
    \hspace{1em} \texttt{vObserved} = vObserved,
    \hspace{1em} \texttt{vError} = vError,
    \hspace{1em} \texttt{R} = R,
    \hspace{1em} \texttt{A} = A,
    \hspace{1em} \texttt{S} = S,
    \hspace{1em} \texttt{iterations} = iterations
    \texttt{)}
    \textbf{if} (\texttt{length(vLatent) > 0}) {
        FScoef <- iRxx \%\% Rxy
        FactorScores <- \texttt{as.matrix(d[, vObserved])} \%\% FScoef
        \texttt{colnames(FactorScores) <- paste0(c(vLatent,
vDisturbance,
 vError),

"_FS")

# Add factor scores to the R matrix

# Initialise factor weight matrix
fw <- matrix(0,

  nrow = nrow(R),
  ncol = ncol(FScoef),
  dimnames = list(rownames(R), colnames(FScoef)))

# Assign values to factor weight matrix
fw[rownames(fw) %in% rownames(FScoef), colnames(FScoef)] <- FScoef

# Combine factor weight matrix with identify matrix of all variables
w <- cbind(diag(nrow(R)), fw)

colnames(w) <- c(colnames(R), paste0(colnames(FScoef), "_FS"))

# Correlation all variables, include factor scores
R_all <- cov2cor(t(w) %*% R %*% w)

# Validity coefficient (% Latent variance in factor scores)
FSValidity <- diag(t(FScoef) %*% Rxy)
# Standard Errors of factor scores

FSStandardError <- sqrt(rep(1, 

    length(c(vLatent, 
            vDisturbance, 
            vError))) - FSValidity)

paste0("se.", names(FSStandardError)) -> names(FSStandardError)

l$Data <- cbind(d, FactorScores)
l$vLatent <- vLatent
l$vDisturbance <- vDisturbance
l$vError <- vError
l$vFactorScores <- colnames(FactorScores)
l$FactorScoreCoef <- FScoef
l$R_all <- R_all
l$R_FS <- cov2cor(t(FScoef) %*% Rxx %*% FScoef)
l$FactorScoreValidity <- FSValidity
l$FactorScoreSE <- FSStandardError
l$Model <- m

}
' @export

#' @export

#' @importFrom lavaan simStandardized

#' @param d observed z-scores in matrix or data.frame

#' @param m Structural model represented by lavaan Syntax

#' @return simStandardized output list with estimated factor scores from d

#' @examples

#' m = "latent_1 =~ 0.8 * ob_1 + 0.8 * ob_2 + 0.8 * ob_3"

#' d <- data.frame(ob_1 = 1, ob_2 = -0.5, ob_3 = 1.2)

#' # Estimate factor scores based on this case

#' estStandardized(d = d, m = m)

estStandardized <- function(d, m){
  d_sim <- simStandardized(m = m, n = 100)
  d_FS <- as.matrix(d[,d_sim$vObserved]) %*% d_sim$FactorScoreCoef
  colnames(d_FS) <- paste0(colnames(d_FS), "_FS")
  d_sim$Data = cbind(d, d_FS)
  d_sim
}

#' Calculate the conditional Mahalanobis distance for any variables.

#'

#' @export

#' @param d Data.frame with the independent and dependent variables.

#' @param R Conditional correlation among variables.

#' @param Dep Vector of names of the dependent variables in your profile.
# @param Ind Vector of names of independent variables you would like to control for.
# @param UseFactorScores_Dep Use the factor scores for the dependent variables
# @param UseFactorScores_Ind Use the factor scores for the independent variables
# @param IncludeDiagnostics Return additional diagnostic information
# @importFrom matrixcalc is.singular.matrix
# @importFrom matrixcalc matrix.rank
# @return conditional Mahalanobis distance, percentiles for each case based on the Chi-square distribution formed by conditional Mahalanobis distance and predicted Deps based on Inds.

# @examples
# m <- "
# Gc =~ 0.85 * Gc1 + 0.68 * Gc2 + 0.8 * Gc3
# Gf =~ 0.8 * Gf1 + 0.9 * Gf2 + 0.8 * Gf3
# Gs =~ 0.7 * Gs1 + 0.8 * Gs2 + 0.8 * Gs3
# Read =~ 0.66 * Read1 + 0.85 * Read2 + 0.91 * Read3
# Math =~ 0.4 * Math1 + 0.9 * Math2 + 0.7 * Math3
# Gc ~ 0.6 * Gf + 0.1 * Gs
# Gf ~ 0.5 * Gs
# Read ~ 0.4 * Gc + 0.1 * Gf
# Math ~ 0.2 * Gc + 0.3 * Gf + 0.1 * Gs"
# d_demo <- simStandardized(m = m, n = 10)
# CMahalanobis(d = d_demo$Data,
CMahalanobis <- function(d,

    R,

    Dep,

    Ind = NULL,

    UseFactorScores_Ind = F,

    UseFactorScores_Dep = F,

    IncludeDiagnostics = F) {

    if (is.list(Dep)) Dep <- unlist(Dep)
    if (is.list(Ind)) Ind <- unlist(Ind)

    # Initialize singular predictors
    Ind_Sing <- character(0)

    # Number of independent and dependent measures
    k_Ind <- length(Ind)
    k_Dep <- length(Dep)

    if (UseFactorScores_Ind) Ind_Use <- paste0(Ind, "_FS") else Ind_Use <- Ind
    if (UseFactorScores_Dep) Dep_Use <- paste0(Dep, "_FS") else Dep_Use <-
Dep

Ryy <- R[Dep, Dep]

if (is.singular.matrix(Ryy)) stop("Dependent measures are collinear. Cannot calculate the Mahalanobis Distance")

LastCondCov <- Ryy

if (!is.null(Ind)) {
  Rxx <- R[Ind, Ind, drop = F]
  Rxy <- R[Ind, Dep, drop = F]
  Ryx <- R[Dep, Ind, drop = F]

  iRxx <- solve(Rxx)

  RegBeta <- iRxx %*% Rxy
  R2 <- colSums(RegBeta * Rxy)

  # change the name to select cases
  PredictedSubtests <- as.matrix(d[, Ind_Use]) %*% RegBeta
  SubtestDeviations <- d[, Dep_Use, drop = F] - PredictedSubtests

  # conditional variance
CondCov <- Ryy - Ryx %*% iRxx %*% Rxy
df <- k_Dep

if (is.singular.matrix(CondCov)) {
  removepredictor <- function(p, Ind, Dep, R, mRank, LastCondCov) {
    if (k_Ind > 1) {
      vInd <- Ind[!p %in% Ind]
      Rxx <- R[vInd, vInd]
      Rxy <- R[vInd, Dep]
      Ryx <- R[Dep, vInd]
      iRxx <- solve(Rxx)
      CondCov <- Ryy - Ryx %*% iRxx %*% Rxy
      list(Remove = mRank != matrix.rank(CondCov),
            CondCov = CondCov)
    } else {
      list(Remove = TRUE,
            CondCov = LastCondCov)
    }
  }

  mRank <- matrix.rank(CondCov)
  i <- 1
  OldInd <- Ind

  while (is.singular.matrix(CondCov) | i < length(OldInd) + 1) {
    rp <- removepredictor(p = Ind[i], Ind, Dep, R, mRank, LastCondCov = CondCov)
if (rp$Remove) {
    Ind <- Ind[!(Ind[i] %in% Ind)]
    CondCov <- rp$CondCov
    LastCondCov <- CondCov
    Ind_Sing <- c(Ind_Sing,Ind[i])
    df <- df - 1
}
i <- i + 1
}

# Conditional Mahalanobis Distance
dCM <- (((as.matrix(SubtestDeviations) %*% solve(CondCov)) * as.matrix(SubtestDeviations))) %*% matrix(1,nrow = k_Dep)) >>=
sqrt %>%
as.vector

# Probability
p <- pchisq(dCM ^ 2, df)

if (IncludeDiagnostics) {
    if (k_Ind > 1) {
        # Calculate Mahalanobis distance of independent variables
\[ \text{d}_\text{IndUse} \leftarrow \text{as.matrix}(d[, \text{Ind}\_\text{Use}, \text{drop} = \text{F}]) \]

\[ \text{dM}_\text{Ind} \leftarrow \left( (\text{d}_\text{IndUse} \odot \text{solve}(\text{Rxx})) \ast \text{d}_\text{IndUse} \right)^{\frac{1}{2}} \text{as.vector} \]

\[ \text{dM}_\text{Ind} \leftarrow \text{NA} \]}

\# Calculate Mahalanobis distance of dependent variables

\[ \text{d}_\text{Dep\_Use} \leftarrow \text{as.matrix}(d[, \text{Dep}\_\text{Use}, \text{drop} = \text{F}]) \]

\[ \text{dM}_\text{Dep} \leftarrow \left( (\text{d}_\text{Dep\_Use} \odot \text{solve}(\text{Ryy})) \ast \text{d}_\text{Dep\_Use} \right)^{\frac{1}{2}} \text{as.vector} \]

\text{list}(\text{dCM} = \text{dCM}, \text{df} = \text{df}, \text{p} = \text{p}, \text{Dep} = \text{Dep\_Use}, \text{Ind} = \text{Ind\_Use}, \text{d}_\text{Dep} = d[, \text{Dep}\_\text{Use}, \text{drop} = \text{F}], \text{d}_\text{Ind} = d[, \text{Ind}\_\text{Use}, \text{drop} = \text{F}], \text{PredictedSubtests} = \text{PredictedSubtests}, \text{SubtestDeviations} = \text{SubtestDeviations}, \text{R2} = \text{R2}, \text{ConditionalCovariance} = \text{CondCov}, \)
\[ dM_{\text{Ind}} = dM_{\text{Ind}}, \]
\[ p_{dM_{\text{Ind}}} = \text{pchisq}(dM_{\text{Ind}}^2, \text{df} = k_{\text{Ind}}), \]
\[ dM_{\text{Dep}} = dM_{\text{Dep}}, \]
\[ p_{dM_{\text{Dep}}} = \text{pchisq}(dM_{\text{Dep}}^2, \text{df} = k_{\text{Dep}}). \]

} else {
  list(dCM = dCM, df = df, p = p)
}

} else {
  d_DepUse <- as.matrix(d[, Dep_Use, drop = F])
  dCM <- (((d_DepUse %*% solve(Ryy)) * d_DepUse) %*% matrix(1, nrow = length(Dep))) %>%
            sqrt %>%
            as.vector
  df <- length(Dep)
  p <- pchisq(dCM^2, df)
  list(dCM = dCM, df = df, p = p)
}

}

#' Wrapper for finding out Mahalanobis distance between variables: this one gives everything for practitioners to use when they only have population relations and their clients' data
#'
#' @export
#' @param m Structural model represented by lavaan Syntax.
#' @param Dep The names of variables you would like to condition on.
#' @param Ind The names of variables of your interest.
#' @param d Profiles of interest.
#' @return conditional Mahalanobis distance, percentiles for each case
# based on the Chi-square distribution formed by conditional Mahalanobis
distance and predicted Deps based on Inds.

#' @examples

# ' # Standardized structural model in lavaan syntax
# '
m <- 
#' Gc =~ 0.85 * Gc_1 + 0.68 * Gc_2 + 0.80 * Gc_3
#' Gf =~ 0.80 * Gf_1 + 0.90 * Gf_2 + 0.80 * Gf_3
#' Read =~ 0.66 * Read_1 + 0.85 * Read_2 + 0.91 * Read_3
#' Math =~ 0.40 * Math_1 + 0.90 * Math_2 + 0.70 * Math_3
#' Gc ~ 0.60 * Gf
#' Read ~ 0.40 * Gc + 0.10 * Gf
#' Math ~ 0.20 * Gc + 0.30 * Gf
#' 
#'

# ' # Put observed scores in data.frame
# ' d_demo <- data.frame(
#'   Gc_1 = -1,
#'   Gc_2 = 0.5,
#'   Gc_3 = -0.2,
#'   Gf_1 = 1.1,
#' Gf_2 = 1.3,
#' Gf_3 = 2,
#' Read_1 = -0.5,
#' Read_2 = -1,
#' Read_3 = -1.4,
#' Math_1 = 1.1,
#' Math_2 = 1.3,
#' Math_3 = 0.7
#' )

#' maha(d = d_demo,
#'   m = m,
#'   Dep = c("Math", "Read"),
#'   Ind = c("Gc", "Gf")
#
# maha <- function(d = d_demo,
#   m = m,
#   Dep = c("Math", "Read"),
#   Ind = c("Gc", "Gf")
#
maha <- function(d, m, Dep, Ind = NULL) {
  Output <- estStandardized(d = d, m = m)

  CMahalanobis(
    Dep = Dep,
    Ind = Ind,
    R = Output$R_all,
    d = Output$Data,
    UseFactorScores_Ind = sum(Output$vLatent %in% Ind) > 1,
    UseFactorScores_Dep = sum(Output$vLatent %in% Dep) > 1
  )
}
' Function to evaluate the accuracy of dCM with estimated factor scores 
'
'# @export
'# @param m Lavaan Syntax Object
'# @param Dep The names of variables you would like to condition on. 
'# @param Ind The names of variables of your interest. 
'# @param n Sample size of simulated data 
'# @return Correlation between the conditional Mahalanobis distance calculated by using the true scores and the conditional Mahalanobis calculated by using estimated factor scores 
'# @examples 
'# m <- "
'# Gc =~ 0.85 * Gc_1 + 0.68 * Gc_2 + 0.80 * Gc_3
'# Gf =~ 0.80 * Gf_1 + 0.90 * Gf_2 + 0.80 * Gf_3
'# Gs =~ 0.70 * Gs_1 + 0.80 * Gs_2 + 0.80 * Gs_3
'# Read =~ 0.66 * Read_1 + 0.85 * Read_2 + 0.91 * Read_3
'# Math =~ 0.40 * Math_1 + 0.90 * Math_2 + 0.70 * Math_3
'# Gc ~ 0.60 * Gf + 0.10 * Gs
'# Gf ~ 0.50 * Gs
'# Read ~ 0.40 * Gc + 0.10 * Gf
'# Math ~ 0.20 * Gc + 0.30 * Gf + 0.10 * Gs
'# "
'# dCM_cor(m, Dep = c("Math", "Read"), Ind = c("Gc", "Gf", "Gs"), n = 100)
```r
dCM_cor <- function(m, Dep, Ind, n = 10000) {
  tryCatch({
    # extract simulated data
    sm <- simStandardized(m, n = n)

    # get the true CMahalanobis
    dCM_true <- CMahalanobis(d = sm$Data,
      R = sm$R_all,
      Dep = Dep,
      Ind = Ind
    )$dCM

    # get the CMahalanobis of FS
    dCM_estimated <- CMahalanobis(d = sm$Data,
      R = sm$R_all,
      Dep = Dep,
      Ind = Ind,
      UseFactorScores_Ind = T,
      UseFactorScores_Dep = T
    )$dCM
    as.vector(cor(dCM_true, dCM_estimated))
  }, error = function(e) NA)
}
```

#' Confidence interval of the reliability (accuracy index)
#'
#' @export
#' @param m Population relations among variables represented by Lavaan Syntax
#' @param Dep The names of variables you would like to condition on
#' @param Ind The names of variables of your interest
#' @param replications The number of trials
#' @param sample_size The number of cases
#' @return simulated 95% confidence interval
#' @examples
#' m <- "
#' Gc =~ 0.85 * Gc_1 + 0.68 * Gc_2 + 0.80 * Gc_3
#' Gf =~ 0.80 * Gf_1 + 0.90 * Gf_2 + 0.80 * Gf_3
#' Gs =~ 0.70 * Gs_1 + 0.80 * Gs_2 + 0.80 * Gs_3
#' Read =~ 0.66 * Read_1 + 0.85 * Read_2 + 0.91 * Read_3
#' Math =~ 0.40 * Math_1 + 0.90 * Math_2 + 0.70 * Math3
#' Gc ~ 0.60 * Gf + 0.10 * Gs
#' Gf ~ 0.50 * Gs
#' Read ~ 0.40 * Gc + 0.10 * Gf
#' Math ~ 0.20 * Gc + 0.30 * Gf + 0.10 * Gs
#' "
#' boot_dCM(m,
#'       Dep = c("Math", "Read"),
#
```r
# Ind = c("Gc", "Gf", "Gs"),
# sample_size = 100,
# replications = 100)

boot_dCM <- function(m, Dep, Ind = NULL, sample_size = 10000, replications = 100) {
  out <- replicate(replications, dCM_cor(
    m = m, Dep = Dep, Ind = Ind,
    n = sample_size
  ))
  stats::quantile(out, probs = c(0.025, 0.5, 0.975))
}

Setup Code for Analyses, Figures, and Tables

Packages to Load and Setup Options

library(unusualprofile)
library(lavaan)
library(magrittr)
library(tidyverse)
library(glue)
library(mvtnorm)
```
library(purrrlyr)
library(psych)
library(knitr)
library(viridis)
library(rlang)

axislabels <- c("0",".2",".4",".6",".8","1")

Helper Functions for Analyses

# Create z-scores
z_score <- function(x) {
  (x - mean(x)) / sd(x)
}

# Function to set a beta variate
makeBeta <- function(p, Trials = 10, n = 1) {
  rbeta(n, p * Trials, (1 - p) * Trials)
}

# Function to create simple sums of latent scores
indicator_sum <- function(o) {
  require(dplyr)
  require(tibble)
  require(magrittr)
  require(rlang)
  require(lavaan)
# Indicators for each latent variable

```r
lvIndicators <- lavParTable(o$Model, fixed.x = F) %>%
  filter(op == "=~") %>%
  select(lhs, rhs) %>%
  group_by(lhs) %>%
  nest()
```

```r
myData <- o$Data
pmap_dfc(lvIndicators,
  function(lhs, data) {
    tibble(! paste0(lhs,"_sum") := myData[,unlist(data)] %>%
      rowSums)
  }) %>%
mutate_all(z_score)
```

# Function to compare accuracy of dCM made with
# Thurstone factor scores and composites

```r
CM_cor_sum <- function(m, Dep, Ind, n){
  tryCatch({
    d <- simStandardized(m, n = n)
    true_dCM <- CMahalanobis(
      d = d$Data,
```
\[
R = dR_all,
\]
\[
\text{Dep} = \text{Dep},
\]
\[
\text{Ind} = \text{Ind}
\]
\[
\]
\[
\]
Thurstone_dCM <- CMahalanobis(
  d = d$Data,
  R = d$R_all,
  Dep = Dep,
  Ind = Ind,
  UseFactorScores_Ind = T,
  UseFactorScores_Dep = T
)

r FS <- cor(true_dCM, Thurstone_dCM)

# extract simulated data

cor_data <- d$Data

vLatent <- d$vLatent

# get the true cor

cor_True <- d$R[c(Dep, Ind), c(Dep, Ind)]

cor_data_sum <- bind_cols(cor_data,
                           indicator_sum(d)) %>%
  as_tibble

colnames(cor_True) <- paste0(colnames(cor_True), "_sum")
row.names(cor_True) <- colnames(cor_True)

composite_dCM <- CMahalanobis(
    d = cor_data_sum,
    R = cor_True,
    Dep = paste0(Dep, "_sum"),
    Ind = paste0(Ind, "_sum")
)

# calculate the reliability of composite dCM
r_Composite <- cor(true_dCM, composite_dCM)
list(r_FS = r_FS, r_Composite = r_Composite)

error = function(e) list(r_FS = NA, r_Composite = NA))

# Function to create lavaan indicators with loadings
makeLoadings <- function(Latent, k, loading) {
    tibble(Latent,k,loading) %>%
        pmap_chr(function(Latent,k,loading) {
            paste0(loading,
                " * ",
                Latent,
                " _ ",
                seq(1:k),
                collapse = " + ")
# Function to create lavaan indicators
makeIndicator <- function(Latent, k_indicators, loading) {
  paste0(
    Latent,
    " =~ ",
    makeLoadings(Latent,k_indicators,loading), collapse = "\n"
  )
}

# Function to evaluate the acuracy of a model
# with a specific number of indicators per factor
IndicatorModelAccuracy <- function(mStructural,
  k_indicators,
  loading,
  Ind_Ind,
  Ind_Dep,
  Dep_Dep,
  Inds,
  Deps,
  n) {
  paste0(makeIndicator(c(Inds,Deps),

k_indicators,loading),

"\n",

glue(mStructural),
collapse = "\n") %>%
dCM_cor(Dep = Deps, Ind = Inds, n = n)

Figure 4

Conditional Mahalanobis distribution: Model A

SimModel_A <- "
Gc =~ 0.85 * Gc_1 + 0.68 * Gc_2 + 0.80 * Gc_3
Gf =~ 0.80 * Gf_1 + 0.90 * Gf_2 + 0.80 * Gf_3
Gs =~ 0.70 * Gs_1 + 0.80 * Gs_2 + 0.80 * Gs_3
RD =~ 0.66 * RD_1 + 0.85 * RD_2 + 0.91 * RD_3
RC =~ 0.40 * RC_1 + 0.90 * RC_2 + 0.70 * RC_3
RF =~ 0.40 * RF_1 + 0.90 * RF_2 + 0.70 * RF_3
Gc ~ 0.30 * Gf
Gf ~ 0.19 * Gs
RD ~ 0.21 * Gc + 0.16 * Gf + 0.16 * Gs
RC ~ 0.15 * Gc + 0.15 * Gf + 0.16 * Gs + 0.50 * RD + 0.42 * RF
RF ~ 0.10 * Gc + 0.14 * Gf + 0.12 * Gs + 0.50 * RD
"

# simulate data using the new function
d_A <- simStandardized(SimModel_A, 1000000)

Deps <- c("RD", "RC", "RF")

Inds <- c("Gc", "Gf", "Gs")
```
d_A.latent <- CMahalanobis(d = d_A$Data, 
                         R = d_A$R_all, 
                         Dep = Deps, 
                         Ind = Inds)

tibble(d_CM = d_A.latent$dCM^2) %>% 
  ggplot(aes(x = d_CM)) + 
  geom_density(fill = "black", 
               alpha = 0.5, 
               color = NA) + 
  stat_function(fun = dchisq, 
                args = list(df = d_A.latent$df), 
                colour = "red", 
                n = 1000) + 
  scale_x_continuous(expression(Conditional ~ Mahalanobis ~ Distance ~ italic((d[CM] ^ 2)))) + 
  scale_y_continuous(name = "Density") + 
  theme_gray(base_size = 14, base_family = "serif")

ggsave("Figure 4 Model A Chi Square.png", 
        dpi = 600, 
        width = 6.5, 
        height = 7.5)
```
Figure 5

Conditional Mahalanobis distribution: Model B

\[
\text{SimModel}_B \leftarrow "
\]

\[
\begin{align*}
\text{Gc} &\sim 0.85 \times \text{Gc1} + 0.68 \times \text{Gc2} + 0.8 \times \text{Gc3} \\
\text{Gf} &\sim 0.8 \times \text{Gf1} + 0.9 \times \text{Gf2} + 0.8 \times \text{Gf3} \\
\text{Gs} &\sim 0.7 \times \text{Gs1} + 0.8 \times \text{Gs2} + 0.8 \times \text{Gs3} \\
\text{Read} &\sim 0.66 \times \text{Read1} + 0.85 \times \text{Read2} + 0.91 \times \text{Read3} \\
\text{Math} &\sim 0.4 \times \text{Math1} + 0.9 \times \text{Math2} + 0.7 \times \text{Math3} \\
\text{Gc} &\sim 0.6 \times \text{Gf} + 0.1 \times \text{Gs} \\
\text{Gf} &\sim 0.5 \times \text{Gs} \\
\text{Read} &\sim 0.4 \times \text{Gc} + 0.1 \times \text{Gf} \\
\text{Math} &\sim 0.2 \times \text{Gc} + 0.3 \times \text{Gf} + 0.1 \times \text{Gs} \\
" \\
\#	ext{test the distribution of Conditional Mahalanobis distance}
\]

\#	ext{simulate 1000 data using the new function}

d_B \leftarrow \text{simStandardized}(\text{SimModel}_B, \text{1000000})

\text{d_B.test} \leftarrow \text{CMahalanobis}(d = \text{d_B}\$\text{Data},

\begin{align*}
\text{R} &= \text{d_B}\$\text{R}, \\
\text{Dep} &= \text{c}(\text{"Math"}, \text{"Read"}), \\
\text{Ind} &= \text{c}(\text{"Gf"}, \text{"Gs"}, \text{"Gc"}))
\end{align*}

\text{tibble(d_CM = d_B.test}\$\text{dCM}^{2})\%\%"
Figure 6
 Scatterplot of estimated and true conditional Mahalanobis

# simulate 1000 data using the new function

d_A <- simStandardized(SimModel_A, 100000)
d_B <- simStandardized(SimModel_B, 100000)
Deps <- c("RD", "RC", "RF")
Inds <- c("Gc", "Gf", "Gs")

d_A.latent <- CMahalanobis(d = d_A$Data,
\textbf{R} = d_A$R$\_all, \\
\textbf{Dep} = Deps, \\
\textbf{Ind} = Inds)

d_A.estimated \leftarrow \textbf{CMahalanobis}(d = d_A$Data, \\
\textbf{R} = d_A$R$\_all, \\
\textbf{Dep} = Deps, \\
\textbf{Ind} = Inds, \\
\text{UseFactorScores}\_\text{Ind} = T, \\
\text{UseFactorScores}\_\text{Dep} = T)

d_B.latent \leftarrow \textbf{CMahalanobis}(d = d_B$Data, \\
\textbf{R} = d_B$R, \\
\textbf{Dep} = c("Math", "Read"), \\
\textbf{Ind} = c("Gf", "Gs", "Gc"))

d_B.estimated \leftarrow \textbf{CMahalanobis}(d = d_B$Data, \\
\textbf{R} = d_B$R$\_all, \\
\textbf{Dep} = c("Math", "Read"), \\
\textbf{Ind} = c("Gf", "Gs", "Gc"), \\
\text{UseFactorScores}\_\text{Ind} = T, \\
\text{UseFactorScores}\_\text{Dep} = T)

tibble(\textbf{Estimated} = d_A.estimated$\text{dCM}, \\
\textbf{Latent} = d_A.latent$\text{dCM},
Factors that influence accuracy of conditional Mahalanobis (Model A)
Deps <- c("RD", "RC", "RF")

Inds <- c("Gc", "Gf", "Gs")

ptable <- crossing(fl = seq(0.01,0.99,0.02),
                   Ind_Ind = seq(0,0.4,0.1),
                   Ind_Dep = seq(0,0.5,0.1),
                   Dep_Dep = seq(0,0.2,0.1))

d_Accuracy <- ptable %>%
  mutate(r = pmap(list(fl, Ind_Ind, Ind_Dep, Dep_Dep),
               function(fl, Ind_Ind, Ind_Dep, Dep_Dep) {
                 paste0("Gc =~ ", fl, " * Gc_1 + ", fl, " * Gc_2 + ", fl, " * Gc_3
                 Gf =~ ", fl, " * Gf_1 + ", fl, " * Gf_2 + ", fl, " * Gf_3
                 Gs =~ ", fl, " * Gs_1 + ", fl, " * Gs_2 + ", fl, " * Gs_3
                 RD =~ ", fl, " * RD_1 + ", fl, " * RD_2 + ", fl, " * RD_3
                 RC =~ ", fl, " * RC_1 + ", fl, " * RC_2 + ", fl, " * RC_3
                 RF =~ ", fl, " * RF_1 + ", fl, " * RF_2 + ", fl, " * RF_3
                 Gc ~ ", Ind_Ind, " * Gf
                 Gf ~ ", Ind_Ind, " * Gs
               })}
\[ RD \sim " \text{Ind}_\text{Dep}, " \text{* Gc } + " \], \\
\text{Ind}_\text{Dep}, " \text{* Gf } + " \\
\text{Ind}_\text{Dep}, " \text{* Gs} \\
\]
\[ RC \sim " \text{Ind}_\text{Dep}, " \text{* Gc } + " \], \\
\text{Ind}_\text{Dep}, " \text{* Gf } + " \\
\text{Ind}_\text{Dep}, " \text{* Gs} \\
\text{Dep}_\text{Dep}, " \text{* RC } + " \\
\text{Dep}_\text{Dep}, " \text{* RF} \\
\]
\[ RF \sim " \text{Ind}_\text{Dep}, " \text{* Gc } + " \], \\
\text{Ind}_\text{Dep}, " \text{* Gf } + " \\
\text{Ind}_\text{Dep}, " \text{* Gs} \\
\text{Dep}_\text{Dep}, " \text{* RF} \\
"
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
\)
geom_smooth(se = F) +

scale_x_continuous("Factor Loadings",
    limits = c(0,1),
    breaks = seq(0,1,0.2),
    labels = axislabels) +

scale_y_continuous(expression(Correlation~of~True~italic(d[CM])~
    and~Estimated~italic(d[CM])),
    limits = c(0,1),
    breaks = seq(0,1,0.2),
    labels = axislabels) +

facet_grid(`Cognitive to\nCognitive` ~ `Academic to\nAcademic`,
    labeller = label_both) +

scale_color_viridis("Cognitive\nto Academic",
    discrete = T,
    direction = -1) +

theme_gray(12,"serif") +
theme(legend.position = "right") +
guides(color = guide_legend(ncol = 1)) +
coord_fixed()

ggsave("Figure 7.png",
    dpi = 600,
    width = 6.5,
    height = 8.5)
Factors that influence accuracy of conditional Mahalanobis (Model B)

```r
Deps <- c("Reading", "Math")
Inds <- c("Gc", "Gf", "Gs")

ptable <- crossing(fl = c(0.05, seq(0.1,0.9,0.1), .95, .99),
                  Ind_Ind = seq(0,0.5,0.1),
                  Ind_Dep = seq(0,0.5,0.1))

d_Accuracy <- ptable %>%
  mutate(r = pmap(list(fl, Ind_Ind, Ind_Dep),
                function(fl, Ind_Ind, Ind_Dep) {
                  paste0("Gc =~ ", fl, " * Gc_1 + ", fl, " * Gc_2 + ", fl, " * Gc_3
                    Gf =~ ", fl, " * Gf_1 + ", fl, " * Gf_2 + ", fl, " * Gf_3
                    Gs =~ ", fl, " * Gs_1 + ", fl, " * Gs_2 + ", fl, " * Gs_3
                    Reading =~ ",
                    fl, " * Reading_1 + ",
```
fl, " * Reading_2 + ",
fl, " * Reading_3
Gc ~ ", Ind_Ind, " * Gf + ", Ind_Ind, " * Gs
Gf ~ ", Ind_Ind, " * Gs
Math ~ ", Ind_Dep, " * Gc + ", Ind_Dep, " * Gf + ", Ind_Dep, " * Gs
Reading ~ ", Ind_Dep, " * Gc + ", Ind_Dep, " * Gf
"
})
}) %>%
  map_dbl(dCM_cor,
    Dep = Deps,
    Ind = Inds,
    n = 100000)

d_Accuracy %>%
  filter(!is.na(r)) %>%
  rename(`Cognitive to
Cognitive` = Ind_Ind,
    `Cognitive to
Academic` = Ind_Dep) %>%
  ggplot(aes(fl , r ,color = factor(`Cognitive to
Academic`))) +
  geom_smooth(se = F) +
  scale_x_continuous("Factor Loadings",
    limits = c(0,1),
    breaks = seq(0,1,0.2),
    labels = axislabels) +
  scale_y_continuous(expression(atop(Correlation~of~True~italic(d[CM]),
    and~Estimated~italic(d[CM]))),
    limits = c(-1,1),
    breaks = seq(-1,1,0.2),
    labels = axislabels)
Comparison of accuracy of conditional Mahalanobis distance using Thurstone factor score and equally weighted composite scores: Model A

```r
n <- 100000
Dep <- c("RD", "RC", "RF")
Inds <- c("Gc", "Gf", "Gs")
# Glueable model for Model A
mStructural <- "
```
Gc =~
{round(makeBeta(loading), 4}) * Gc_1 +
{round(makeBeta(loading), 4}) * Gc_2 +
{round(makeBeta(loading), 4}) * Gc_3

Gf =~
{round(makeBeta(loading), 4}) * Gf_1 +
{round(makeBeta(loading), 4}) * Gf_2 +
{round(makeBeta(loading), 4}) * Gf_3

Gs =~
{round(makeBeta(loading), 4}) * Gs_1 +
{round(makeBeta(loading), 4}) * Gs_2 +
{round(makeBeta(loading), 4}) * Gs_3

RD =~
{round(makeBeta(loading), 4}) * RD_1 +
{round(makeBeta(loading), 4}) * RD_2 +
{round(makeBeta(loading), 4}) * RD_3

RC =~
{round(makeBeta(loading), 4}) * RC_1 +
{round(makeBeta(loading), 4}) * RC_2 +
{round(makeBeta(loading), 4}) * RC_3

RF =~
{round(makeBeta(loading), 4}) * RF_1 +
{round(makeBeta(loading), 4}) * RF_2 +
{round(makeBeta(loading), 4}) * RF_3

Gc ~ {Ind_Ind} * Gf
Gf ~ {Ind_Ind} * Gs
RD ~
{Ind_Dep} * Gc +
{Ind_Dep} * Gf +
{Ind_Dep} * Gs
RC ~
{Ind_Dep} * Gc +
{Ind_Dep} * Gf +
{Ind_Dep} * Gs +
{Dep_Dep} * RD +
{Dep_Dep} * RF
RF ~
{Ind_Dep} * Gc +
{Ind_Dep} * Gf +
{Ind_Dep} * Gs +
{Dep_Dep} * RD
"

ptable <- crossing(reps = 1,

    loading = c(seq(0.01, 0.998, 0.001), 0.999),
    Ind_Ind = seq(0, 0.3, 0.1),
    Dep_Dep = seq(0, 0.3, 0.1)) %>%

    mutate(Ind_Dep = Ind_Ind)

m <- ptable %>%
pmap_chr(function(reps,loading,
    Ind_Ind,
    Ind_Dep,
    Dep_Dep) glue(mStructural))

d_composite <- bind_cols(ptable,
    map_df(m,
        CM_cor_sum,
        Dep = Deps ,
        Ind = Inds,
        n = n)) %>%
gather(Type,r,r_FS,r_Composite)

d_composite %>%
    filter(Ind_Ind < 0.4) %>%
    rename(`Cognitive Effects` = Ind_Ind,
        `Academic Effects` = Dep_Dep) %>%
    mutate(Type = factor(Type,
        levels = c("r_Composite","r_FS"),
        labels = c("Composite","Thurstone"))) %>%
ggplot(aes(loading, r, color = Type)) +
    geom_point(size = 0.01, alpha = 0.15) +
    geom_smooth(se = F, size = 0.5) +
    facet_grid(`Cognitive Effects` ~ `Academic Effects`,
        labeller = label_both) +
Comparison of accuracy of conditional Mahalanobis distance using Thurstone factor score and equally weighted composite scores: Model B

```r
Deps <- c("Reading", "Math")
Inds <- c("Gc", "Gf", "Gs")
```

Figure 10
n <- 1000

# Glueable model for Model B

mStructural <- "

Gc =~
{round(makeBeta(loading), 4)} * Gc_1 +
{round(makeBeta(loading), 4)} * Gc_2 +
{round(makeBeta(loading), 4)} * Gc_3

Gf =~
{round(makeBeta(loading), 4)} * Gf_1 +
{round(makeBeta(loading), 4)} * Gf_2 +
{round(makeBeta(loading), 4)} * Gf_3

Gs =~
{round(makeBeta(loading), 4)} * Gs_1 +
{round(makeBeta(loading), 4)} * Gs_2 +
{round(makeBeta(loading), 4)} * Gs_3

Math =~
{round(makeBeta(loading), 4)} * Math_1 +
{round(makeBeta(loading), 4)} * Math_2 +
{round(makeBeta(loading), 4)} * Math_3

Reading =~
{round(makeBeta(loading), 4)} * Reading_1 +
{round(makeBeta(loading), 4)} * Reading_2 +
{round(makeBeta(loading), 4)} * Reading_3

Gc ~ {Ind_Ind} * Gf + {Ind_Ind} * Gs
Gf ~ {Ind_Ind} * Gs
Math ~ {Ind_Dep} * Gc + {Ind_Dep} * Gf + {Ind_Dep} * Gs
Reading ~ {Ind_Dep} * Gc + {Ind_Dep} * Gf

ptable <- crossing(reps = 1,
                   loading = c(seq(0.01, 0.998, 0.001), 0.999),
                   Ind_Ind = seq(0, 0.3, 0.1),
                   Ind_Dep = seq(0, 0.5, 0.1)) %>%
mutate(Dep_Dep = Ind_Ind)

m <- ptable %>%
pmap_chr(function(reps,
                   loading,
                   Ind_Ind,
                   Ind_Dep,
                   Dep_Dep) glue(mStructural))

d_composite <- bind_cols(ptable,
                          map_df(m,
                          CM_cor_sum,
                          Dep = Deps,
                          Ind = Inds,
                          n = 10000)) %>%
```r
gather(Type, r, r_FS, r_Composite)

d_composite %>%
  filter(Ind_Ind < 0.4) %>%
  rename(`Cognitive to
         Cognitive` = Ind_Ind,
         `Cognitive to
         Academic` = Ind_Dep) %>%
  mutate(Type = factor(Type,
                   levels = c("r_Composite","r_FS"),
                   labels = c("Composite","Thurstone"))) %>%
  ggplot(aes(loading, r, color = Type)) +
  geom_point(size = 0.01, alpha = 0.15) +
  geom_smooth(se = F, size = 0.5) +
  facet_grid(`Cognitive to
            Academic` ~ `Cognitive to
            Cognitive`,
            labeller = label_both) +
  scale_color_manual("Estimate Type",
                   values = c("#449148", "#3B528B")) +
  theme_gray(12,"serif") +
  theme(legend.position = "top") +
  guides(color = guide_legend(nrow = 1)) +
  coord_fixed() +
  scale_x_continuous("Average Factor Loadings",
                    limits = c(0,1),
                    breaks = seq(0,1,0.2),
                    labels = axislabels) +
  scale_y_continuous(expression(Correlation~of~True~italic(d[CM])~
                               Correlation~of~True~italic(d[CM])~)
```

and~Estimated~italic(d[CM])),

limits = c(0,1),
breaks = seq(0,1,0.2),
labels = axislabels)

ggsave("Figure 10.png",
    dpi = 1200,
    width = 6.5,
    height = 8)

**Figure 11**

The effect of the number of indicators on the accuracy of the conditional Mahalanobis Distance:

Model A

```r
Deps <- c("RD", "RC", "RF")
Inds <- c("Gc", "Gf", "Gs")

# Glueable structural model for Model A
mStructural <- "
Gc ~ {Ind_Ind} * Gf
Gf ~ {Ind_Ind} * Gs
RD ~ {Ind_Dep} * Gc + {Ind_Dep} * Gf + {Ind_Dep} * Gs
RC ~
{Ind_Dep} * Gc +
{Ind_Dep} * Gf +
{Ind_Dep} * Gs +
{Dep_Dep} * RD +
```

88
\{\text{Dep}_{\text{Dep}}} \ast \text{RF}

\text{RF} \sim \{\text{Ind}_{\text{Dep}}} \ast \text{Gc} + \{\text{Ind}_{\text{Dep}}} \ast \text{Gf} + \{\text{Ind}_{\text{Dep}}} \ast \text{Gs} + \{\text{Dep}_{\text{Dep}}} \ast \text{RD} \\
\end{align}

"# Create dataframe with parameters

\text{ptable} <- \text{crossing}(k_{\text{indicators}} = 1:5, \\
\hspace{1cm} \text{loading} = \\
\hspace{2cm} \text{c}(0.05, \\
\hspace{3.2cm} \text{seq}(0.1,0.9,0.1), \\
\hspace{3.2cm} 0.95, \\
\hspace{3.2cm} 0.9999), \\
\hspace{1cm} \text{Ind}_{\text{Ind}} = \text{seq}(0,0.4,0.1), \\
\hspace{1cm} \text{Dep}_{\text{Dep}} = \text{seq}(0,0.4,0.1)) \%>\%

\text{mutate}(\text{Ind}_{\text{Dep}} = \text{Ind}_{\text{Ind}})

# calculate accuracy correlations

\text{r} <- \text{ptable} \%>\%

\text{pmap_dbl}(\text{IndicatorModelAccuracy}, \\
\hspace{1cm} \text{Inds} = \text{Inds}, \\
\hspace{1cm} \text{Deps} = \text{Deps}, \\
\hspace{1cm} \text{n} = 100000, \\
\hspace{1cm} \text{mStructural} = \text{mStructural})

# Plot accuracy correlations

\text{ptable} \%>\%
```r
code
```
The effect of the number of indicators on the accuracy of the conditional Mahalanobis Distance:

Model B

```r
# Glueable structural model for Model B
mStructural <- "
Gc ~ {Ind_Ind} * Gf + {Ind_Ind} * Gs
Gf ~ {Ind_Ind} * Gs
Math ~ {Ind_Dep} * Gc + {Ind_Dep} * Gf + {Ind_Dep} * Gs
Reading ~ {Ind_Dep} * Gc + {Ind_Dep} * Gf"

Deps <- c("Reading", "Math")
Inds <- c("Gc", "Gf", "Gs")

# Create dataframe with parameters
ptable <- crossing(k_indicators = 1:5,
                     loading =
                     c(0.05, seq(0.05, 0.9, 0.05), 0.95, 0.9999),
                     Ind_Ind = seq(0, 0.3, 0.1),
                     Ind_Dep = seq(0, 0.4, 0.1))

# calculate accuracy correlations
```
r <- ptable %>%
  pmap_dbl(IndicatorModelAccuracy,
    Dep_Dep = 0,
    Inds = Inds,
    Deps = Deps,
    n = 100000,
    mStructural = mStructural)

# Plot accuracy correlations
ptable %>%
  mutate(r = r) %>%
  filter(!is.na(r), Ind_Ind < 0.4) %>%
  rename(`Cognitive\nto Cognitive` = Ind_Ind,
        `Cognitive\nto Academic` = Ind_Dep) %>%
  mutate(Indictors = factor(k_indicators)) %>%
  ggplot(aes(loading, r, color = Indictors)) +
  geom_smooth(se = F) +
  facet_grid(`Cognitive\nto Academic` ~ `Cognitive\nto Cognitive`,
             labeller = label_both) +
  scale_color_viridis("Indictors per Factor",
                     discrete = T) +
  scale_x_continuous("Factor Loadings",
                    limits = c(0, 1),
                    breaks = seq(0, 1, 0.2),
                    labels = axislabels) +
Case for Figure 13 and Table 3

# Cases for Figure 13

# Make data

m = "Gc =~ 0.9 * Gc_1 + 0.9 * Gc_2 + 0.9 * Gc_3"

d_data <- simStandardized(m = m,
                          n = 10000)

# Make case 1 and 2

d_case <- matrix(c(2,2,2,
                  1,2,3),2, byrow = T) %*%
```r
cbind(diag(3), d_data$FactorScoreCoef[, "Gc"]) %>%
set_colnames(c(paste0("Gc_", 1:3), "Gc_FS")) %>%
as_tibble() %>%
mutate(Gc = Gc_FS / d_data$FactorScoreValidity["Gc"])

# Calculate d_CM

d_CM <- CMahalanobis(d = d_case,
                     R = d_data$R_all,
                     Dep = paste0("Gc_", 1:3),
                     Ind = "Gc",
                     UseFactorScores_Ind = T)

# Make Table 3

d_case %>%
select(-Gc) %>%
mutate_all(function(x) x * 15 + 100) %>%
mutate(pdM = as.vector(d_M$p),
        pdCM = as.vector(d_CM$p),
        Case = 1:2) %>%
select(Case, everything()) %>%
knitr::kable(digits = 3)

Cases for Figure 14 and 15

d_data_B <- simStandardized(
    m = "
```
\[ Gc = 0.90 \cdot Gc1 + 0.90 \cdot Gc2 + 0.90 \cdot Gc3 \]
\[ Gf = 0.90 \cdot Gf1 + 0.90 \cdot Gf2 + 0.90 \cdot Gf3 \]
\[ Gs = 0.90 \cdot Gs1 + 0.90 \cdot Gs2 + 0.90 \cdot Gs3 \]
\[ RD = 0.86 \cdot RD1 + 0.88 \cdot RD2 + 0.89 \cdot RD3 \]
\[ RC = 0.78 \cdot RC1 + 0.88 \cdot RC2 + 0.84 \cdot RC3 \]
\[ RF = 0.83 \cdot RF1 + 0.85 \cdot RF2 + 0.89 \cdot RF3 \]
\[ Gc \sim 0.30 \cdot Gf \]
\[ Gf \sim 0.19 \cdot Gs \]
\[ RD \sim 0.21 \cdot Gc + 0.16 \cdot Gf + 0.16 \cdot Gs \]
\[ RC \sim 0.15 \cdot Gc + 0.15 \cdot Gf + 0.16 \cdot Gs + 0.50 \cdot RD + 0.42 \cdot RF \]
\[ RF \sim 0.10 \cdot Gc + 0.14 \cdot Gf + 0.12 \cdot Gs + 0.50 \cdot RD \]

# Case data

d_case_B <- tibble::tribble(
  ~Gc, ~Gf, ~Gs, ~RD, ~RC, ~RF,
  115L, 115L, 115L, 85L, 85L, 85L,
  61L, 78L, 85L, 86L, 75L, 84L
) %>%
  mutate_all(function(x) (x - 100) / 15)

dCM_B <- CMahalanobis(d = d_case_B, 
  R = d_data_B$R)
Dep = c("RD","RC","RF"),
Ind = c("Gc","Gf","Gs"),
IncludeDiagnostics = T)

# p(dCM) for cases

tibble(d_CM = dCM_B$p,
       d_M_Academic = dCM_B$p_dM_Dep,
       d_M_Cognitive = dCM_B$p_dM_Ind) %>%
kable(digits = 3)