

# Composite-based Structural Equation Modeling



## INAUGURAL-DISSERTATION

zur Erlangung des akademischen Grades eines Doktors der  
Wirtschaftswissenschaften an der Wirtschaftswissenschaftlichen  
Fakultät der Julius-Maximilians-Universität Würzburg

vorgelegt von

Manuel Elias Rademaker

Würzburg, Juni 2020

Erstgutachter: Prof. Dr. Martin Kukuk

Zweitgutachter: Prof. Toker Doganoglu, Ph.D.

*To my wonderful wife.*

---

# Zusammenfassung

---

Strukturgleichungsmodellierung (SEM) wird seit Jahrzehnten in verschiedenen Bereichen und Forschungsgebieten wie zum Beispiel der Psychologie, der Soziologie und den Wirtschaftswissenschaften verwendet und weiterentwickelt. SEM umfasst dabei die Gesamtheit einer Reihe verwandter Theorien, mathematischer Modelle, Methoden, Algorithmen und Terminologien im Zusammenhang mit der Analyse der Beziehungen zwischen theoretischen Entitäten – so genannten Konzepten –, ihrer statistischen Repräsentation – als Konstrukte bezeichnet – und Beobachtungsgrößen – üblicherweise Indikatoren, Items oder manifeste Variablen genannt.

Diese Arbeit befasst sich mit Aspekten eines bestimmten Forschungszweigs innerhalb der SEM, den Komposit-basierten SEM Verfahren. Komposit-basiertes SEM ist ein Überbegriff für alle SEM Methoden, die Kompositen – d.h. gewichtete Linearkombinationen aus Beobachtungen – zur Schätzung unbekannter Größen verwenden.

Der Inhalt der Arbeit basiert auf einem Arbeitspapier (Kapitel 2), einem veröffentlichten referierten Zeitschriftenartikel (Kapitel 3), einem weiteren Arbeitspapier (Kapitel 4) und einer stetig wachsenden Dokumentation, die ich für das R-Paket `cSEM` geschrieben habe, bzw. kontinuierlich weiter schreibe (Kapitel 5). Das Paket `cSEM` – geschrieben von mir und Florian Schuberth, meinem ehemaligen Kollegen an der Universität Würzburg – stellt Funktionen zur Verfügung, um lineare, nichtlineare, hierarchische und Multigruppen-Strukturgleichungsmodelle mit Hilfe von

Komposit-basierten Ansätzen und Verfahren zu schätzen, zu analysieren, zu bewerten, zu testen und zu untersuchen.

In Kapitel 1 gehe ich zunächst kurz auf einige der wichtigsten SEM Begriffe ein. Kapitel 2 basiert auf einem Arbeitspapier mit dem Titel *Assessing overall model fit of composite models in structural equation modeling*, das im *Journal of Business Research* eingereicht werden wird. Der Artikel befasst sich mit dem Thema der Bewertung der Gesamtgüte des Modells (eng. overall model fit) im Kontext des Komposit-Modells. Das Papier leistet drei zentrale Beiträge zur Literatur zu diesem Thema. Erstens wird das Konzept der Modellgüte in der SEM im Allgemeinen und der Komposit-basierten SEM im Besonderen eingehend erörtert. Zweitens wird auf gängige Fit-Indizes eingegangen und erläutert, ob und wie sie zur Beurteilung eines Komposit-Modells angewendet werden können. Drittens wird gezeigt, dass der *root mean square outer residual covariance* ( $RMS_{\theta}$ ) identisch mit einem anderen bekannten Index, dem *standardized root mean square residual* (SRMR) ist, falls der  $RMS_{\theta}$  als Maß für die Modellanpassungsgüte verwendet werden soll.

Kapitel 3 basiert auf einem in *Internet Research* veröffentlichten Zeitschriftenartikel mit dem Titel *Measurement error correlation within blocks of indicators in consistent partial least squares: Issues and remedies*. Der Artikel entwickelt das consistent partial least squares (PLSc) Verfahren weiter, um konsistente Parameterschätzungen für Populationsmodelle zu erhalten, deren Indikatorblöcke korrelierte Messfehler enthalten. Dies wird erreicht, indem die Korrektur für die Dämpfung (eng. attenuation), wie sie ursprünglich von PLSc angewandt wird, so modifiziert wird, dass sie a priori Annahmen über die Struktur der Messfehlerkorrelationen innerhalb der Indikatorblöcke enthält. Um die statistische Gültigkeit der Modifikation zu beurteilen, wird eine Monte Carlo Simulation durchgeführt. Das Papier wurde gemeinsam mit Florian Schubert und Theo Dijkstra verfasst.

Kapitel 4 basiert auf einem zur Veröffentlichung in *Industrial Management & Data Systems* anstehenden Zeitschriftenartikel mit dem Titel *Estimating and testing second order constructs using PLS-PM: the case of composites of composites*. Der Zweck dieses Artikels ist ein dreifacher: (i) Bewertung und Vergleich gängiger Ansätze zur Schätzung von Modellen, die Konstrukte zweiter Ordnung enthalten, die als Komposite von Kompositen modelliert wurden; (ii) ein zweistufiges Testverfahren vorzustellen und statistisch zu bewerten, um die allgemeine Modellanpassung solcher

Modelle zu testen und (iii) auf der Grundlage unserer Ergebnisse Empfehlungen für Praktiker zu formulieren. Darüber hinaus wurde eine Monte Carlo Simulation durchgeführt, um die Ansätze in Bezug auf Fisher-Konsistenz, geschätzte Verzerrung und RMSE zu vergleichen. Das Papier wurde gemeinsam mit Florian Schuberth und Jörg Henseler verfasst.

Kapitel 5 stellt das R-Paket `cSEM` vor. Zum Zeitpunkt der Einreichung dieser Dissertation zur Begutachtung ist `cSEM` im Comprehensive R Archive Network (CRAN) als Version 0.2.0 verfügbar.

---

# Abstract

---

Structural equation modeling (SEM) has been used and developed for decades across various domains and research fields such as, among others, psychology, sociology, and business research. Although no unique definition exists, SEM is best understood as the entirety of a set of related theories, mathematical models, methods, algorithms, and terminologies related to analyzing the relationships between theoretical entities – so-called concepts –, their statistical representations – referred to as constructs –, and observables – usually called indicators, items or manifest variables.

This thesis is concerned with aspects of a particular strain of research within SEM – namely, composite-based SEM. Composite-based SEM is defined as SEM involving linear compounds, i.e., linear combinations of observables when estimating parameters of interest.

The content of the thesis is based on a working paper (Chapter 2), a published refereed journal article (Chapter 3), a working paper that is, at the time of submission of this thesis, under review for publication (Chapter 4), and a steadily growing documentation that I am writing for the R package `cSEM` (Chapter 5). The `cSEM` package – written by myself and my former colleague at the University of Wuerzburg, Florian Schubert – provides functions to estimate, analyse, assess, and test nonlinear, hierarchical and multigroup structural equation models using composite-based approaches and procedures.

In Chapter 1 I briefly discuss some of the key SEM terminology.

Chapter 2 is based on a working paper to be submitted to the *Journal of Business Research* titled *Assessing overall model fit of composite models in structural equation modeling*. The article is concerned with the topic of overall model fit assessment of the composite model. Three main contributions to the literature are made. First, we discuss the concept of model fit in SEM in general and composite-based SEM in particular. Second, we review common fit indices and explain if and how they can be applied to assess composite models. Third, we show that, if used for overall model fit assessment, the *root mean square outer residual covariance* ( $\text{RMS}_\theta$ ) is identical to another well-known index called the standardized root mean square residual (SRMR).

Chapter 3 is based on a journal article published in *Internet Research* called *Measurement error correlation within blocks of indicators in consistent partial least squares: Issues and remedies*. The article enhances consistent partial least squares (PLSc) to yield consistent parameter estimates for population models whose indicator blocks contain a subset of correlated measurement errors. This is achieved by modifying the correction for attenuation as originally applied by PLSc to include a priori assumptions on the structure of the measurement error correlations within blocks of indicators. To assess the efficacy of the modification, a Monte Carlo simulation is conducted. The paper is joint work with Florian Schuberth and Theo Dijkstra.

Chapter 4 is based on a journal article under review for publication in *Industrial Management & Data Systems* called *Estimating and testing second-order constructs using PLS-PM: the case of composites of composites*. The purpose of this article is threefold: (i) evaluate and compare common approaches to estimate models containing second-order constructs modeled as composites of composites, (ii) provide and statistically assess a two-step testing procedure to test the overall model fit of such models, and (iii) formulate recommendation for practitioners based on our findings. Moreover, a Monte Carlo simulation to compare the approaches in terms of Fisher consistency, estimated bias, and RMSE is conducted. The paper is joint work with Florian Schuberth and Jörg Henseler.

Chapter 5 introduces the R package `cSEM`. By the time this dissertation was submitted, `cSEM` was available on the Comprehensive R Archive Network (CRAN) as version 0.2.0.



---

# Acknowledgment

---

This dissertation was written while I was employed as a research assistant at the Chair of Econometrics at the University of Wuerzburg. Above all, I therefore express my gratitude to my supervisor and head of the Chair of Econometrics, Prof. Dr. Martin Kukuk for fostering my interest in econometrics in several lectures and numerous discussions. Moreover, I thank him for creating an open and encouraging research environment which allowed me to pursue my scientific priorities and thus develop my own scientific ideas.

Furthermore, I particularly like to thank my former colleague at the University of Wuerzburg, co-author, and professional go-to support, Florian Schubert, who gently convinced me to focus on composite-based structural equation modeling in my dissertation and guided me through the – at times frustrating – process of scientific publication. He has been the driving force behind many of the projects we jointly worked on. It cannot be overstated how important it is to have a partner with an undeniable professional expertise to turn to for answers, discussions, and inspiration! I am therefore deeply grateful to him for all the conversations and probably thousand phone call hours we had – even if we did not always share the same opinion.

I would also like to thank my colleague, Tamara Schamberger, for keeping as much work off by back as possible while I was concentrating on my thesis. More importantly, however, I thank her for her support, her time she spent reviewing what I had written, her patience, and all the interesting conversations we had – both professional and private; in short, I thank her for being a dear friend!

In addition, I would like to thank my co-authors Jörg Henseler and Theo Dijkstra for

all the time they took to answer my questions and the valuable comments they made when reviewing drafts I had written.

I also thank my mother Roswitha Fydrich-Steiner and my father Bernhard Steiner, as well as my sisters Verena Jenkin and Tamara Steiner for being a family that truly is a place to retreat, relax and regain strength. Moreover, I express my gratitude towards all my friends and colleagues at University of Wuerzburg and around the world who accompanied me during the last 6 years.

Lastly, I am deeply grateful to my wife, Iris Rademaker, who I met while writing this thesis. Her unconditional love and support have created the positive mental environment that has generated the drive necessary to complete a project of the magnitude of this dissertation.

Thank you all!

Wuerzburg, June 2020

Manuel E. Rademaker

---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Structural equation modeling . . . . .	1
1.2	The latent variable or common factor model . . . . .	1
1.3	The composite model . . . . .	3
1.4	Composite-based SEM and the partial least squares algorithm (PLS) . . . . .	5
<b>2</b>	<b>Assessing overall model fit of composite models in structural equation modeling</b>	<b>6</b>
2.1	Introduction . . . . .	6
2.2	The composite model and composite-based estimation . . . . .	12
2.2.1	Estimation . . . . .	14
2.3	The concept of model fit . . . . .	14
2.4	Distance measures, fit functions and fit indices . . . . .	18
2.4.1	Measures of absolute fit . . . . .	19
2.4.2	Measures of relative fit . . . . .	28
2.4.3	Fit indices and cut-off . . . . .	32
2.5	Model fit tests . . . . .	34
2.6	Discussion . . . . .	35
<b>3</b>	<b>Measurement error correlation within blocks of indicators in consistent partial least squares: Issues and remedies</b>	<b>39</b>
3.1	Introduction . . . . .	39

3.2	The partial least squares path modeling algorithm . . . . .	42
3.2.1	Traditional partial least squares path modeling . . . . .	43
3.2.2	Consistent partial least squares . . . . .	45
3.3	Correlated measurement errors . . . . .	46
3.4	Monte Carlo simulation . . . . .	48
3.5	Results . . . . .	51
3.6	Discussion and future research . . . . .	56
<b>4</b>	<b>Estimating and testing second-order constructs using partial least squares path modeling: the case of composites of composites</b>	<b>60</b>
4.1	Introduction . . . . .	60
4.2	Approaches to estimate models containing second-order constructs . . .	66
4.2.1	Repeated indicators approach . . . . .	69
4.2.2	Two-stage approach . . . . .	70
4.2.3	Hybrid approach . . . . .	71
4.3	Assessment of the overall model fit . . . . .	71
4.3.1	Assessing the fit of higher-order constructs . . . . .	74
4.4	A Monte Carlo simulation . . . . .	75
4.4.1	Simulation design . . . . .	76
4.4.2	Simulation results . . . . .	82
4.4.3	Simulation insights . . . . .	87
4.5	Guidelines for estimating and assessing models containing composites of composites . . . . .	87
4.6	Discussion and future research . . . . .	91
4.7	Appendix A to Chapter 4 . . . . .	94
4.8	Appendix B to Chapter 4 . . . . .	104
4.8.1	Data generation . . . . .	104
4.8.2	Procedure to obtain the alternative DGPs . . . . .	111
4.8.3	Models . . . . .	115
4.8.4	Simulation . . . . .	117
4.8.5	Helper functions . . . . .	122
<b>5</b>	<b>cSEM: An R package for composite-based structural equation modeling</b>	<b>127</b>

5.1	Introduction . . . . .	127
5.2	Preparing the data . . . . .	131
5.3	Specifying a model . . . . .	133
5.3.1	Linear models . . . . .	134
5.3.2	Models containing nonlinear terms . . . . .	134
5.3.3	Hierarchical (second-order) models . . . . .	135
5.4	Estimation using <code>csem()</code> . . . . .	136
5.4.1	Weighting approaches . . . . .	139
5.4.2	The <code>cSEMResults</code> class . . . . .	141
5.4.3	Inference . . . . .	144
5.5	Postestimation . . . . .	151
5.5.1	<code>assess()</code> . . . . .	153
5.5.2	<code>infer()</code> . . . . .	161
5.5.3	<code>predict()</code> . . . . .	163
5.5.4	<code>summarize()</code> . . . . .	166
5.5.5	<code>verify()</code> . . . . .	169
5.5.6	<code>testOMF()</code> . . . . .	171
5.5.7	<code>testMICOM()</code> . . . . .	173
5.5.8	<code>testMGD()</code> . . . . .	178
5.5.9	<code>testHausman()</code> . . . . .	185
5.6	Technical details . . . . .	187
	<b>Bibliography</b>	<b>191</b>

---

# List of Figures

---

1.1	Measurement models . . . . .	2
1.2	A typical structural model . . . . .	3
1.3	The composite model . . . . .	4
2.1	Histogramms of the simulated $\chi^2$ statistic and theoretical $\chi^2$ distribution. .	20
2.2	Misspecified models . . . . .	28
2.3	Histogramms of the simulated $\chi^2$ statistic and theoretical noncentral $\chi^2$ distribution. . . . .	29
3.1	Baseline population model . . . . .	49
4.1	The four main types of second-order constructs . . . . .	64
4.2	Example of a model containing a second-order composite formed by composites . . . . .	67
4.3	Second-order DGP . . . . .	76
4.4	Structural specification of the alternative DGPs. . . . .	79
4.5	Tests' rejection rates in case of no misspecification . . . . .	85
4.6	Tests' rejection rates for various degrees of misspecification . . . . .	86
4.7	Tests' rejection rates for the alternative population models estimated by a correctly specified model . . . . .	102
4.8	Tests' rejection rates for the baseline population model estimated by a specification that matches the alternative population models . . . . .	103
4.9	Second-order DGP . . . . .	104

4.10	Structural specification for the alternative DGPs . . . . .	107
5.1	The cSEM workflow . . . . .	130
5.2	Structure of a <i>cSEMResults_default</i> object . . . . .	141
5.3	Structure of a <i>cSEMResults_multi</i> object . . . . .	142
5.4	Structure of a <i>cSEMResults_2ndorder</i> object . . . . .	143

---

# List of Tables

---

3.1	Estimated bias . . . . .	54
3.2	Root mean squared error (RMSE) . . . . .	55
4.1	Approaches to estimate models containing composites of composites . . . . .	68
4.2	Population path coefficients in case of misspecifications . . . . .	80
4.3	Results for the estimated bias, estimated RMSE, Fisher consistency, and number of converged estimations . . . . .	83
4.4	Guidelines for estimating and assessing models containing second-order composites built by composites . . . . .	88
4.5	Indicator population covariance matrix of the baseline model . . . . .	95
4.6	Indicator population covariance matrix in case of the small model misspecification . . . . .	96
4.7	Indicator population covariance matrix in case of the medium model misspecification . . . . .	97
4.8	Indicator population covariance matrix in case of the large model misspecification . . . . .	98
4.9	Approaches' Fisher consistency - Part I . . . . .	99
4.10	Approaches' Fisher consistency - Part II . . . . .	100
4.11	Smallest weight, smallest loading, and largest VIF for the population models used in the context of model misspecification . . . . .	101
4.12	Population path coefficients in case of misspecifications . . . . .	108
5.1	Weighting approaches available via <code>.approach_weights</code> . . . . .	140



5.2	Available postestimation functions (cSEM, version 0.2.0) . . . . .	151
5.3	State of the postestimation functions as of version 0.2.0. . . . .	152
5.4	Quality criteria computed by <code>assess()</code> - Part I . . . . .	155
5.5	Quality criteria computed by <code>assess()</code> - Part II . . . . .	156
5.6	Inferential quantities computed by <code>infer()</code> . . . . .	162
5.7	Path estimates . . . . .	169

---

# Chapter 1

## Introduction

---

### 1.1 Structural equation modeling

Structural equation modeling (SEM) is concerned with analyzing, i.e., modeling, estimating, assessing, and testing the (causal) relationships between concepts – entities defined by a conceptual definition (Rigdon, 2016) – with other concepts and/or observable quantities generally referred to as indicators, manifest variables or items.

Broadly speaking, two statistical modeling approaches for the concepts and their relationship exist. The first is known as the latent variable or common factor model and the second as the composite model. Each approach entails a different view of "how the world functions" and, consequently, a potentially unique set of methods, test, evaluation criteria, and a specific terminology that may or may not be adequate within the realm of the other approach.

### 1.2 The latent variable or common factor model

Assuming a researcher identifies  $J$  concepts and  $K$  indicators, the fundamental feature of the latent variable model is the assumption of the existence of a set of  $J$  latent variables (common factors) that each serve as a statistical representation of one

of the  $J$  concepts to be studied. Depending on the assumed theoretical relationship between the latent variable  $J$  and its corresponding  $K_j$  indicators, the latter are modeled as causes of the latent variable (reflective measurement) or as causing the variable itself (causal-formative measurement).

The entirety of these measurement relations is captured by the measurement model which relates indicators to latent variables according to the researchers theory of how observables are related to the concepts in question. Figure 1.1 illustrates the two types of measurement relations of the latent variable model using the standard SEM symbols and notation (e.g., [Bollen, 1989b](#)).

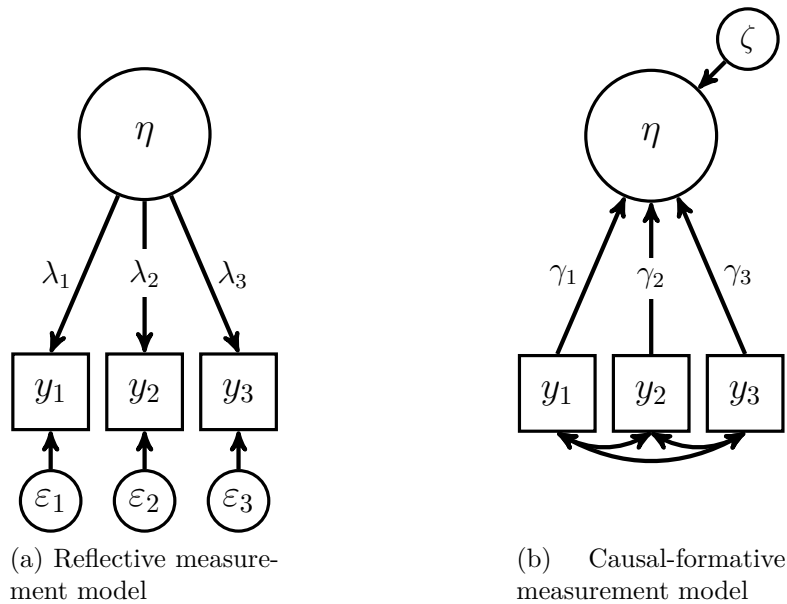


Figure 1.1: Measurement models

Accordingly, unobserved entities such as the latent variable  $\eta$ , the measurement error terms  $\varepsilon_1$ ,  $\varepsilon_2$ , and  $\varepsilon_3$ , and the structural error term  $\zeta$  are graphically illustrated by circles. Observables, i.e., the indicators  $y_1$ ,  $y_2$ , and  $y_3$  are depicted as squares. The one-sided arrows either indicate the assumed causal relationship or adherence. In Figure (a), for instance, the latent variable  $\eta$  is modeled as the underlying cause of the indicators  $y_1$ ,  $y_2$ , and  $y_3$ . The strength of the relationship is captured by the (factor) loadings  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ . In Figure (b), on the other hand, the latent variable is assumed to be caused by the indicators, with corresponding parameters  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  capturing the strength of the relationship. The one-sided arrows from the error terms to the indicators and latent variable indicate adherence. Two-sided arrows indicate covariance or correlation.

The entirety of the relationships between concepts – i.e., its representation in a statistical model, the construct – is captured by the structural model, whose parameters are usually at the center of the researchers interest. Using standard notation and symbols again, a typical structural model would look like this:

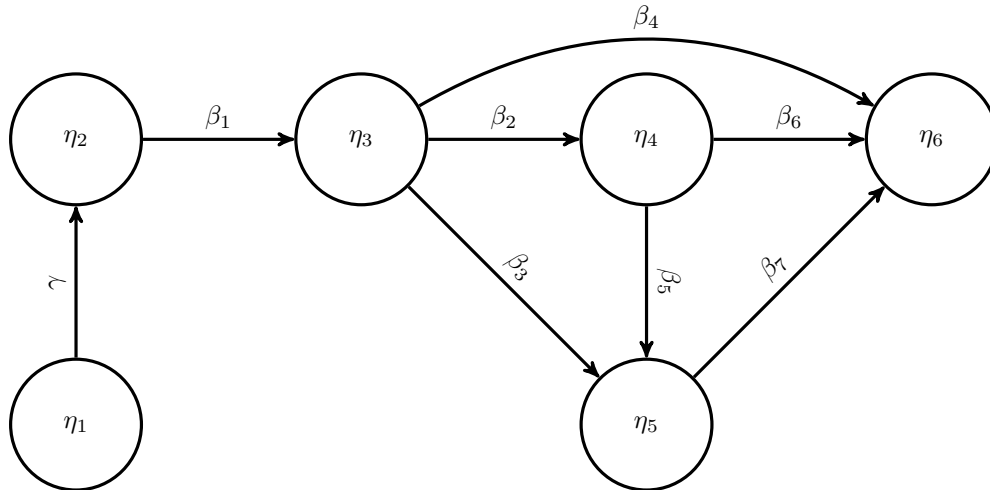


Figure 1.2: A typical structural model

Parameters  $\gamma$  and  $\beta$  are referred to as path or structural coefficients. Variables that have an arrow pointing towards them are called endogenous; otherwise they are called exogenous.

The unknown parameters of the latent variable model are usually estimated by maximum likelihood (ML; [Jöreskog, 1967, 1970b](#)). The basic idea of ML is to find parameters such that the difference between the empirical and the covariance matrix implied by the model is minimized. Such estimation methods are therefore often referred to as covariance-based methods. For details on the different measurement models, estimation, and model assessment the reader is referred to the standard literature of the area e.g., [Bollen \(1989b\)](#), [Kline \(2015\)](#) or [Brown \(2015\)](#).

### 1.3 The composite model

The second approach is known as the composite model (e.g., [Dijkstra, 2017](#); [Henseler, 2017](#); [Schuberth et al., 2018a](#)). As opposed to the latent variable or common factor model, composites do not presuppose the existence of a latent variable. Hence, designed entities (artifacts) such as a particular therapy in psychology or a "leadership workshop" that arguably have no latent counterpart but rather emerge

may be adequately described by a composite, i.e., the linear combination of observables defining the composite. Figure 1.3 shows a generic composite model. Note that contrary to the causal formative measurement model, there is no error term  $\zeta$  on the construct level.

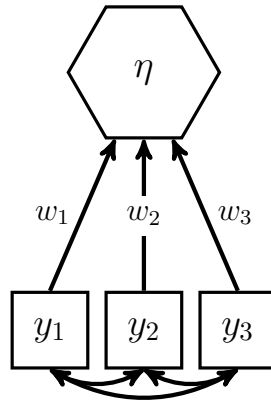


Figure 1.3: The composite model

Composites are usually depicted by hexagons and the parameters  $w_1$ ,  $w_2$ , and  $w_3$  capturing the magnitude of the relationship between the indicators and the composites are called weights. Again, several composites can be placed in a structural model as in Figure 1.2.

Composites may also be formed to represent latent variables/common factors, in which case the composite serves as a proxy or stand-in for the latent variable (e.g., Rigdon, 2016; Rigdon et al., 2017, 2019).

Parameters in composite models are typically retrieved by a composite-based approach such as the partial least squares (PLS) algorithm, generalized structured component analysis (GSCA), or dimension reduction techniques such as principal component analysis (PCA). However, in principal, ML estimation is possible as well.<sup>1</sup> The basic idea of any composite-based approach is to build scores, i.e., composites for each concept and subsequently retrieve structural model parameters by a series of (linear) regressions. Such estimation methods are therefore often referred to as variance-based methods, as regression maximizes the explained variance of the dependent variable. See e.g., Dijkstra (2017) or Schuberth et al. (2018a) for details.

<sup>1</sup>This is ongoing research by Jörg Henseler, Florian Schuberth, and Tamara Schamberger.

## 1.4 Composite-based SEM and the partial least squares algorithm (PLS)

Composite-based SEM is the entirety of methods, approaches, procedures, and algorithms that involve linear compounds (composites/proxies/scores), i.e., linear combinations of observables when retrieving (estimating) quantities of interest such as the coefficients of the structural model. It is crucial to clearly distinguish between the *composite model* and *composite-based SEM*. They are not the same. While the former is a statistical model relating constructs to observables, the latter simply states that composites – linear compounds, i.e., weighted linear combinations of observables – are used to estimate parameters of interest. Hence, composite-based SEM as a way of obtaining/estimating parameters of interest may thus be used for the latent variable or common factor model as well as the composite model. However, interpretation of the parameter estimates is fundamentally different since the underlying models differ. In principal, any weighted combination of appropriately chosen observables can be used to estimate structural relationships between these linear compounds. Hence, any conceptual or methodological issue discussed based on a composite build by a given (weighting) approach may equally well be discussed for any other potential weighting scheme. The appropriateness or potential superiority of a specific weighting approach or algorithm such as the partial least squares (PLS) algorithm (Wold, 1975, 1982; Lohmöller, 1989) over another such as unit weights (e.g., Devlieger and Rosseel, 2017), or generalized structured component analysis (GSCA; Hwang and Takane, 2014) is therefore a question of *relative* appropriateness and *relative* superiority (Rönkkö, 2014).

As a notable consequence, approaches such as the PLS algorithm or GSCA are best exclusively understood as prescriptions for forming linear compounds based on observables, i.e., as weighting approaches. In fact, labels such as PLS-PM and even more so PLS-SEM as propagated by e.g., Hair et al. (2017b); Sarstedt et al. (2019); Cheah et al. (2018); Hair et al. (2019b,a, 2020) are misleading as they create the impression that PLS(-SEM) is somehow capable of more than other composite-based approaches. While among composite-based approaches, research surrounding *composites formed using weights obtained by the PLS algorithm* is most vivid, the PLS algorithm remains a weighting scheme in its core (Rönkkö et al., 2016).

---

## Chapter 2

# Assessing overall model fit of composite models in structural equation modeling

---

### 2.1 Introduction <sup>1</sup>

Structural equation modeling (SEM) is a widely used technique to examine relationships between concepts – entities defined by a theoretical definition (Rigdon, 2016). Within SEM several strains of research may be distinguished. One distinction is typically made with respect to the operationalization of the concepts as constructs within a statistical model.

According to the latent variable or common factor model, concepts are operationalized as latent variables, i.e., statistical entities for which there are no direct observations but observable measures, the indicators (Bollen, 1989b).

Depending on the assumed theoretical relationship between the latent variable and its corresponding indicators, the latter are either modeled as causes of the latent variable

---

<sup>1</sup>This chapter is based on a working paper to be submitted to the *Journal of Business Research*. I thank Sebastian Groß, Florian Schuberth, and Tamara Schamberger for helpful comments on prior versions of the manuscript.

(reflective measurement) or as causing the variable itself (causal-formative measurement). The latent variable model has been the dominant operationalization framework for over a century and has been discussed extensively in numerous books and articles (e.g., [Jöreskog, 1969](#); [Bollen, 1989b](#); [Muthén, 2002](#); [Kline, 2015](#); [Brown, 2015](#), too name just a few).

An alternative, albeit less prominent, operationalization is the composite model (e.g., [Henseler et al., 2014](#); [Dijkstra, 2017](#); [Henseler, 2017](#); [Schuberth et al., 2018a](#)). In the composite model, concepts are modeled as emergent variables, i.e., a composite build by a weighted sum of indicators. Hence, designed entities (artifacts) that arguably have no latent counterpart could be adequately operationalized by a composite. For example, concepts like marketing mix known from Marketing research ([Fornell and Bookstein, 1982](#)) and concepts from Information Systems research such as IT infrastructure flexibility ([Benitez et al., 2018c](#)) and IT integration capability ([Braojos et al., forthcoming](#)) have been modeled as emergent variable. Besides using the composite model to operationalize artifacts, the current literature argues that composites may also be used as statistical proxies for latent variables (e.g., [Rigdon, 2016](#); [Rigdon et al., 2017, 2019](#); [Hair and Sarstedt, 2019](#)). Regardless of the intention of application, concepts in the composite model are operationalized as weighted sums of the indicators in either cases.

While parameters in the latent variable model are typically estimated by maximum likelihood (ML; e.g., [Jöreskog, 1969, 1970b,a](#)), generalized least squares (GLS; [Browne, 1984](#)), or variants thereof, parameters in composite models are naturally estimated by a composite-based approach (e.g., [Dijkstra, 2017](#); [Schuberth et al., 2018a](#)). Composite-based approaches involve linear compounds, i.e., weighted linear combinations of observables when retrieving (estimating) parameters of interest. The basic idea of any composite-based approach is to build weighted scores/composites for each concept and, subsequently, retrieve model parameters by a series of – usually linear – regressions. Such estimation methods are therefore often referred to as variance-based methods, as regression maximizes the explained variance of the dependent variable. Similarly, estimators such as ML and GLS are sometimes referred to as covariance-based estimators in the SEM context, as they obtain model parameter estimates such that the discrepancy between the observed and the model-implied indicator covariance is minimized. In this paper, we use the partial



least squares algorithm (PLS; [Wold, 1975, 1982](#); [Lohmöller, 1989](#)), which is arguably the most common composite-based method.<sup>2</sup>

Irrespective of the choice of concept operationalization and estimator, overall model fit assessment is considered an integral part of SEM (e.g., [Bollen, 1989b](#); [Hu and Bentler, 1999](#); [Yuan, 2005](#); [Barrett, 2007](#); [Hayduk, 2014](#); [Kline, 2015](#); [Greiff and Heene, 2017](#); [Henseler, 2018](#)). On a theoretical level, overall model fit in SEM refers to the closeness of a given model to the underlying process or distribution assumed to give rise to the observed data – the data generating process (DGP). While closeness may be defined in several ways, it is generally operationalized by a distance measure, i.e., a mathematical loss function, capturing the discrepancy between the estimated model-implied indicator covariance matrix and its sample counterpart.

The most common statistical framework to empirically assess model fit – as captured by a distance measure – is the exact model fit testing framework. For covariance-based estimators such as ML a  $\chi^2$  test is readily available (e.g., [Jöreskog, 1969](#); [Bollen, 1989b](#)). For composite-based estimators no  $\chi^2$  test has been derived yet. Composite-based estimators therefore rely on the bootstrap-based test for exact overall model fit proposed by [Dijkstra and Henseler \(2015a\)](#) in the context of PLS.<sup>3</sup> Both, the  $\chi^2$  and the bootstrap-based test for exact overall model fit assess the null hypothesis that the population covariance matrix ( $\Sigma$ ) is *exactly* equal to the indicator covariance matrix implied by the postulated model ( $\Sigma(\theta)$ ).

While the statistical testing framework is theoretically appealing, exact model fit testing may not always be the adequate decisions framework for applied research questions. The epistemological basis of the claim is rooted in [Box \(1976\)](#)'s famous “all models are wrong, but some are useful” remark which implies that the null hypothesis of *perfect* fit is, by definition of a model as inherently approximate, *always* wrong.

Accordingly, exact fit is therefore typically not of actual interest to researchers trying

---

<sup>2</sup>It is crucial at this point to clearly distinguish between the composite model and composite-based estimators. While the former is a statistical model relating concepts to observables (indicators) via composites, the latter simply states that composites - linear compounds, i.e., weighted linear combinations of observables - are used to *estimate* parameters of the postulated statistical model. Whether these estimates are “good” in a statistical sense – i.e., consistent and/or efficient – depends on the statistical model to be estimated – e.g, composite-model vs. latent variable model. Composite-based estimators essentially only differ in how they determine weights and consequently composite scores. Possible contenders are, among others, the partial least squares algorithm (PLS; [Wold, 1975, 1982](#); [Lohmöller, 1989](#)), generalized structured component analysis (GSCA; [Hwang and Takane, 2004](#)), principal component analysis, or generalized canonical correlation analysis (GCCA; [Kettenring, 1971](#)).

<sup>3</sup>See [Beran and Srivastava \(1985\)](#) for mathematical proofs of the procedure and [Bollen and Stine \(1992\)](#) for illustration and intuitive motivation.

to study a certain phenomenon by means of a model, i.e., a deliberate approximation to reality (e.g., Jöreskog, 1969; Steiger and Lind, 1980; Bentler and Bonett, 1980; Browne and Cudeck, 1992; Mulaik et al., 1989; MacCallum, 2003; Cudeck and Henly, 2003; Steiger, 2007; Hoyle, 2012). Technically, dependence of the  $\chi^2$  statistic on the sample size will render the smallest numerical difference between  $\Sigma(\theta)$  and  $\Sigma$  statistically significant with a probability approaching 1 (Steiger and Lind, 1980). As a consequence, virtually all models will have to be rejected for increasing sample sizes. Hence, although a rejection of the null hypothesis of perfect fit has clear statistical implications – namely that an alternative, typically more parameterized model could be found which mimics the structure of the data more adequately – the implications in practical terms are less clear: discard the model completely; or is there anything to be learned from the model despite its lack of perfect fit? Against this background, researchers in the early 1980s started popularizing a number of fit indices as an alternative or supplement to binary reject/non-reject exact model fit testing in an attempt to overcome the practical issues related to its rational (e.g., Bentler and Bonett, 1980; Steiger and Lind, 1980; Jöreskog and Sörbom, 1982). Since then a large number of fit indices has been proposed and studied (see, e.g., Hoyle (2012) for an overview). Following McDonald and Ho (2002) these indices can roughly be classified as absolute or relative.

The principal idea of absolute model fit indices is to measure the correspondence between model and data along a continuum to gauge how well the postulated model corresponds to the data instead of merely defining a perfect correspondence as the (only) desirable objective (e.g. Mulaik et al., 1989, p. 431). Relative fit indices, on the other hand, compare the postulated model to an adequate reference model to assess the relative increase in model fit. In contrast to model fit testing, model fit assessment via fit indices is consequently rather a question of “close” (Browne and Cudeck, 1992) or “comparative” (Bentler and Bonett, 1980; Bentler, 1990) fit than that of perfect fit.

To classify the magnitude of a given fit index as “sufficiently close” or “comparatively better”, conventional recommendations rely on simple one-size-fits-all cut-offs that essentially dichotomize the continuum of fit values into an acceptable/non-acceptable region. While the index-cut-off decision strategy seems to convey similarity to classical hypothesis testing, both frameworks are, in fact, distinctly different. First,

the magnitudes of all indices are influenced, to different degrees, by model and data characteristics other than the misspecification they attempt to quantify (e.g. [Sivo et al., 2006](#)). The major factors identified by the literature are sample size, number of constructs and number of indicators (i.e., model complexity), the distribution of the data, and the sizes of the population parameters – see e.g., [Niemand and Mai \(2018\)](#) for an overview. Second, the distinction between “good” and “bad” models in terms of usefulness, parsimony, interpretability, and generalizability is widely recognized as an inherently case-specific and subjective endeavor that requires to take into account any kind of reliable evidence (e.g., [Jöreskog, 1969](#); [Bentler and Bonett, 1980](#); [Bentler, 1990](#); [Browne and Cudeck, 1992](#); [Bollen, 1993](#); [Marsh et al., 2004](#); [Yuan, 2005](#); [Bentler, 2006](#); [Lai, 2019](#)). Hence, scientifically meaningful decisions about fit and misfit are conceptually non-binary. Consequently, using fit indices as measures of approximate fit and one-size-fits-all cut-offs to translate these values back into a binary reject/non-reject framework is, to some extent, logically incoherent.

This paper makes three contributions to the literature on model assessment of composite models estimated by PLS.

First, we provide a thorough discussion of the concept “model fit” in the context of SEM in general and the composite model estimated by PLS in particular. The discussion is motivated by a number of claims that have recently been brought forward by e.g., [Hair et al. \(2017b\)](#), [Hair et al. \(2019a\)](#), and [Hair et al. \(2019b\)](#) discouraging model fit assessment by means of model fit testing or model fit indices in the context of a model estimated by PLS. These claims seem to warrant a correction. Moreover, we clarify the role of exact model fit testing in the context of the composite model estimated by PLS, explain the reasoning behind approximate model fit assessment by means of fit indices, and discuss the (in)appropriateness of conventional cut-offs for these indices.

Second, we review common fit indices known from the latent variable model and explain their intuition and applicability to a composite model estimated by a composite-based approach such as PLS. The number of fit indices available for discussion is large, however, only a subset of indices is regularly used in practice. In this paper, we therefore examine a subset of eight fit indices which, based on our review, feature most prominently in SEM research. On the one hand, we examine those fit indices whose construction logic is independent of the construct

operationalization, namely the root mean square residual (RMR; [Jöreskog and Sörbom, 1982](#)), the standardized root mean square residual (SRMR; [Bentler, 1995](#)) and the normed fit index (NFI; [Bentler and Bonett, 1980](#)). On the other hand, we consider five indices/statistics known from covariance-based SEM, namely the  $\chi^2$  statistic and the  $\chi^2$  statistic divided by its degrees of freedom ([Jöreskog, 1969](#)), the nonnormed fit index (NNFI; [Bentler and Bonett, 1980](#)), the comparative fit index (CFI; [Bentler, 1990](#)), and the root mean square error of approximation (RMSEA; [Steiger and Lind, 1980](#)). The latter three indices rely on the  $\chi^2$  statistic and properties of its asymptotic  $\chi^2$  distribution. In this context we show by means of a small Monte Carlo simulation that the  $\chi^2$  test statistic constructed using the PLS estimator asymptotically follows a  $\chi^2$  distribution with degrees of freedom equal to the number of non-redundant elements of the empirical indicator covariance matrix minus the number of model parameters.

Third, we thoroughly examine a lesser known index called the root mean square outer residual covariance ( $\text{RMS}_\theta$ ) which was first defined by [Lohmöller \(1989\)](#) in the context of PLS and has been suggested as a suitable model fit criterion (e.g., [Henseler et al., 2014](#)) – although evidence of the claim is scarce with no study providing insights into its theoretical and statistical justification. In particular, we discuss the different ways the  $\text{RMS}_\theta$  can be constructed and show that only one of the ways of constructing the  $\text{RMS}_\theta$  is meaningful for model fit assessment; however, precisely in this case, the  $\text{RMS}_\theta$  is numerically identical to the SRMR, which leads us to conclude that the  $\text{RMS}_\theta$  is, in fact, simply the SRMR, unless it is defined in a way that makes it unsuitable for model fit assessment.

The remainder of the paper is structured as followed. In Section 2 we review the composite model and composite-based estimation. Section 3 is concerned with the concept of model fit. In Section 4 we discuss distance measures, fit functions, fit indices, recommended cut-offs, and explain their intuition and applicability to composite models estimated by PLS. In Section 5 we discuss model fit tests. The paper closes with a discussion in Section 6.

## 2.2 The composite model and composite-based estimation

Let  $\eta_j$  with  $j = 1, \dots, J$  be a construct, i.e., a statistical representation of a concept and  $\mathbf{x}_j$  be a  $(K_j \times 1)$  random vector of standardized indicators related to  $\eta_j$ . Let  $\Sigma_j(\boldsymbol{\theta})$  be the model-implied covariance matrix of these indicators based on a generic model governed by a vector of generic population model parameters  $\boldsymbol{\theta}$ . Further, let  $E(\mathbf{x}_j \mathbf{x}_j') = \Sigma_j$  be the population indicator covariance matrix of the same indicator block.<sup>4</sup> In the composite model, constructs are operationalized by composites, i.e., linear combinations of indicators

$$\eta_j \equiv c_j = \mathbf{w}_j' \mathbf{x}_j, \quad (2.1)$$

where  $c_j$  is called a composite and  $\mathbf{w}_j$  is a  $(K_j \times 1)$  vector of weights scaled such that the variance of the composite is one.<sup>5</sup>

The indicators of composite  $c_j$  are allowed to freely covary. Hence, the model-implied indicator covariance matrix  $\Sigma_j(\boldsymbol{\theta})$  carries no restrictions regarding the off-diagonal elements and is taken to be equal to  $\Sigma_j$ .

The composites may be assumed to freely correlate in which case the model-implied composite covariance matrix  $\Sigma_c$  carries no restrictions other than the unit variance restriction, i.e.,  $\text{diag}(\Sigma_c) = \boldsymbol{\iota}$ , where  $\boldsymbol{\iota}$  is a vector of ones.

Alternatively, structural model restrictions may be imposed. Let  $\mathbf{c}_{endo}$  be the vector of endogenous composites of the reduced form structural model and  $\mathbf{c}_{exo}$  a vector of exogenous composites. Following common notation (e.g., [Bollen, 1989b](#)), the structural relationship between composites can be written in general form as

$$\begin{aligned} \mathbf{c}_{endo} &= \mathbf{B} \mathbf{c}_{endo} + \mathbf{\Gamma} \mathbf{c}_{exo} + \boldsymbol{\zeta} \\ &= \mathbf{\Pi} \mathbf{c}_{exo} + (\mathbf{I} - \mathbf{B})^{-1} \boldsymbol{\zeta}, \end{aligned} \quad (2.2)$$

where  $\mathbf{B}$  and  $\mathbf{\Gamma}$  are parameter matrices. The matrix  $\mathbf{\Pi}$  is known as the reduced form matrix containing the model-implied total effects of the exogenous on the endogenous composites.  $\mathbf{I}$  is the identity matrix and  $\boldsymbol{\zeta}$  a vector of structural errors with

$E(\boldsymbol{\zeta} | \mathbf{c}_{exo}) = \mathbf{0}$ . Theoretically,  $\mathbf{B}$  and  $\mathbf{\Gamma}$  can take on any form, however, practically,

<sup>4</sup>Since indicators are standardized the terms covariance and correlation can be used interchangeably.

<sup>5</sup>Since by definition of the composite as  $c_j = \mathbf{w}_j' \mathbf{x}_j$  the variance is  $\text{Var}(c_j) = \mathbf{w}_j' \Sigma_j \mathbf{w}_j$ , weights are scaled by  $(\mathbf{w}_j' \Sigma_j \mathbf{w}_j)^{-\frac{1}{2}}$  such that  $\text{Var}(c_j) = 1$ .

model parameter identification typically necessitates several restrictions. To avoid identification issues due to feedback loops and correlated structural error terms, we only consider recursive models, and models with uncorrelated structural error terms in this paper. Technically, recursiveness implies that the vector  $\mathbf{c}_{endo}$  can be rearranged such that  $\mathbf{B}$  is lower triangular with zeros on its main diagonal.<sup>6</sup> In addition we focus on linear structural relations; models containing interactions, quadratic terms or higher order constructs are thus excluded.

Rearranging the vector of composites such that  $\mathbf{c} = (\mathbf{c}'_{exo} \ \mathbf{c}'_{endo})'$ , the model-implied covariance matrix of the composites is given by

$$\Sigma_{\mathbf{c}} = \begin{pmatrix} \Phi & \Phi\Pi' \\ \Pi\Phi & \Pi\Phi\Pi' + (\mathbf{I} - \mathbf{B})^{-1}\Psi((\mathbf{I} - \mathbf{B})^{-1})' \end{pmatrix}, \quad (2.3)$$

where  $\Phi$  the composite covariance matrix of the exogenous composites and  $\Psi$  is the covariance matrix of the error terms which is assumed to be diagonal.

The covariances between the composite  $c_j$  and the corresponding indicators  $\mathbf{x}_j$ , i.e., the vector of composite loadings is captured by the  $(K_j \times 1)$  vector  $\boldsymbol{\lambda}_j = \Sigma_j \mathbf{w}_j$ . The covariances between  $\mathbf{x}_j$  and  $\mathbf{x}_s$  ( $j \neq s$ ) are fully attributed to the covariance between its corresponding composites, i.e., the  $js$ 'th element of the matrix  $\Sigma_{\mathbf{c}}$ , and the vector of composite loadings:

$$\Sigma_{js}(\boldsymbol{\theta}) = \sigma_{c;js} \boldsymbol{\lambda}_j \boldsymbol{\lambda}'_s \quad (2.4)$$

Technically, this is known as the rank 1 constraint, as the model-implied indicator covariance matrix  $\Sigma_{js}(\boldsymbol{\theta})$  has rank 1 by construction. More intuitively, the constraint is essentially a reformulation of Wolds *fundamental principle of soft modeling* in terms of composites stating that all information between constructs is conveyed solely by the composites (e.g., [Wold, 1982](#); [Dijkstra, 2010, 2017](#)).

The complete model-implied indicator covariance matrix is thus given by:

$$\Sigma(\boldsymbol{\theta}) = \begin{pmatrix} \Sigma_1 & \Sigma_{12} & \Sigma_{13} & \dots & \Sigma_{1J} \\ & \Sigma_2 & \Sigma_{23} & \dots & \Sigma_{2J} \\ & & \ddots & \vdots & \vdots \\ & & & \Sigma_{J-1;J-1} & \Sigma_{J-1;J} \\ & & & & \Sigma_J \end{pmatrix} \quad (2.5)$$

---

<sup>6</sup>Ruling out feedback loops in the structural model essentially ensures identification, since recursive models are almost always identified ([Bollen, 1989b](#)).

Similar to latent variable models, degrees of freedom are calculated as the difference between the number of non-redundant off-diagonal elements of the population indicator covariance matrix  $\Sigma$  and the number of free model parameters in  $\theta$ . If the model contains only linear terms  $\theta$  is:

$$\theta = \theta(\Sigma_1, \Sigma_2, \dots, \Sigma_J, \mathbf{W}, \Phi, \mathbf{B}, \Gamma) \quad (2.6)$$

The degrees of freedom are thus given by:<sup>7</sup>

$$\begin{aligned} df &= \# \text{ of non-redundant off-diagonal elements of } \Sigma \\ &\quad - \# \text{ of non-redundant off-diagonal elements of } \Sigma_j \text{ for } j = 1, \dots, J. \\ &\quad - \# \text{ of weights} \\ &\quad - \# \text{ of free covariances between exogenous composites} \\ &\quad - \# \text{ of structural parameters (non-zero elements of } \mathbf{B} \text{ and } \Gamma) \\ &\quad + \# \text{ of composites} \end{aligned} \quad (2.7)$$

### 2.2.1 Estimation

The vector  $\theta$  contains all model parameters of the composite model. To obtain an estimate  $\hat{\theta}$  of  $\theta$  any consistent estimator may be used.

In this paper, we use the PLS algorithm with mode B.<sup>8</sup> As detailed by [Dijkstra \(2010, 2017\)](#) weight estimates obtained by PLS are continuously differentiable functions of the sample indicator covariance matrix  $\mathbf{S}$ . Moreover, any composite model parameter estimate – e.g., the elements of  $\hat{\Sigma}_j$ ,  $\hat{\mathbf{B}}$  or  $\hat{\Gamma}$  – can be shown to depend smoothly on elements of  $\mathbf{W}$  and  $\mathbf{S}$  only. Assuming  $\mathbf{S}$  is a consistent estimate of  $\Sigma$ , the estimator  $\hat{\theta}_{PLS} = f(\mathbf{S}, \hat{\mathbf{W}}(\mathbf{S}))$  is consistent for  $\theta$  and asymptotically jointly normal. Similar results apply to other composite-based algorithms such as the class of generalized canonical correlation analysis (GCCA) algorithms ([Kettenring, 1971](#); [Dijkstra, 2017](#)).

## 2.3 The concept of model fit

Model fit and statistical modeling in general is based on the idea that any population whose features a researcher seeks to learn about is governed by a mechanism that

<sup>7</sup>Indicators are assumed to be standardized, i.e., their variance is fixed to one. Hence, in contrast to the degrees of freedom calculation in the latent variable model, the main diagonal elements of  $\Sigma$  are ignored.

<sup>8</sup>For algorithmic details interested readers are referred to e.g., [Wold \(1982\)](#), [Lohmöller \(1989\)](#), or [Rademaker et al. \(2019\)](#).

describes the population in its entirety. If the researcher knew the mechanism, any feature of the population would be known, any (causal) relationships in the population completely traceable, and any realization of that population completely predictable. Hence, neither a model nor statistical inference is required. Practically, the precise mechanism is, bearing some trivial exceptions, virtually never known.<sup>9</sup> To make inferential, causal, or even mechanistic statements about the population, quantitative researchers formulate a statistical model that imposes restrictions on how theoretical (unobserved) and observable features of the population are thought to behave according to theory. A classical example of a statistical model is the multiple linear regression model; the conditional mean of the dependent variable is modeled as a linear function of a set of explanatory variables and an error term assumed to be at least uncorrelated with the explanatory variables. Likewise, the composite model as discussed in the previous section constitutes a statistical model that imposes restrictions on how (mixed) moments of observables and unobservables are thought to behave in the population.

Model fit refers to the closeness of a given statistical model to the underlying mechanism assumed to govern the population and any realizations based thereon. In statistics this mechanism is generically referred to as the data generating process (DGP). Model fit assessment in quantitative research is therefore concerned with the evaluation of statistical evidence indicating whether a statistical model is in line with the (empirical) world, i.e., the data produced by the DGP the researcher intends to model. Clearly, statistical evidence is not the only possible piece of evidence that could be brought forward to judge whether a given model is “good” or “bad” in terms of model fit, however, it is compelling and widely accepted for two reasons. First, when judging a statistical model like the composite model in terms of model fit, it seems reasonable to rely on statistical evidence to do so; after all, ignoring statistical evidence begs the question of why statistical methodology was used in the first place. Second, non-statistical model fit assessment does not adhere to an objective framework with verifiable mathematical implications making scientific debates about implications of a given model notoriously difficult (e.g. [Browne and Cudeck, 1992](#)). Consequently, non-statistical statements about model fit are hard to generalize and

---

<sup>9</sup>Unless, of course, the process is designed by the researcher as in e.g., a simulation study.



falsify.<sup>10</sup>

Since the term “fit” may convey a different meaning to readers of different scientific backgrounds it is crucial to avoid ambiguity. Three aspects of model assessment have explicitly or implicitly been associated with model fit: in-sample fit (fit in the regression sense), out-of-sample “fit” – more commonly referred to as predictive accuracy –, and fit in the SEM sense. Fit in the regression sense is the amount of variation in a target variable that is explained by the model. Evidence of this type of fit, as measured by the  $R^2$ , is a measure of in-sample (predictive) fit. Out-of-sample predictive accuracy, as measured by, for instance, the root mean square error (RMSE), is the closeness between predicted and actual values of a subset of one or several target variables whose values have not been used in the estimation. Note that neither in-sample nor out-of-sample predictive accuracy require the model to be causally correctly specified. In fact, a model *may* happen to be causally correctly specified with respect to the population, however, neither causal misspecification nor correct causal specification are sufficient for model-to-population fit in the SEM sense. Fit in the SEM sense is distinctly different. In contrast to predictive fit (accuracy) assessment, model fit in SEM is concerned with the closeness of the mechanism implied by the researcher’s model to the DGP. While closeness may be defined in several ways it is typically operationalized as the difference between the model-implied indicator covariance matrix  $\Sigma(\theta)$  of the postulated model and the population indicator covariance matrix  $\Sigma$  as captured by a generic distance or fit function  $F$ :

$$F = F(\Sigma, \Sigma(\theta)) \quad (2.8)$$

In fact, model fit in SEM is used exclusively to refer to this concept of fit. The implicit assumption motivating the use of covariance matrices is the idea that the causal structure of the underlying DGP is thought to be reflected in these matrices. While perfect fit in the population as indicated by some distance measure is

---

<sup>10</sup>We consider as non-statistical model fit assessment any assessment about the goodness/adequacy of a model based on qualitative statements. The defining feature of statistical evidence on the other hand is evidence obtained based on an objective framework with verifiable mathematical implications. While most statistical frameworks such as the null hypothesis significance testing (NHST) framework clearly involve subjectivity (which significance level  $\alpha$  is appropriate?), the implications once  $\alpha$  is chosen are mathematically undeniable. Labeling qualitative statements as non-statistical should not be misunderstood as dismissive, nor should the fact that NHST has clear *mathematical* implications be interpreted as reckless support for NHST. Qualitative statements based on domain knowledge and subjective judgment are a vital part of any scientific endeavor. Moreover, NHST does suffer from its own set of challenges (e.g., [Mertens and Recker, 2019](#); [Wasserstein et al., 2019](#)). Nevertheless, non-statistical evidence must be labeled as such and not be confused with or sold as statistical evidence.

necessary but not sufficient for a correctly causally specified model, observing a misfit according to  $F$  is sufficient for causal misspecification.<sup>11</sup> To put it differently: a model-implied indicator covariance matrix that perfectly fits its population counterpart according to the distance function  $F$  may still be causally misspecified, however, models whose fit is non-zero can never be fully correctly specified (e.g. Jöreskog and Sörbom, 1993, p.114).

As an important consequence of the scope of the different concepts of fit, statistical evidence about fit according to one of the three concepts, does not necessarily imply fit according to the other. Regression fit, for example, is no evidence of correct causal specification; an  $R^2$  of 0.99 merely implies an almost perfect in-sample fit – typically a sign of overfitting. Likewise, causally correctly specified models may exhibit high out-of-sample predictive accuracy, yet reversing the argument is a known logical fallacy (Saylor and Trafimow, 2020, p.10). Consequently, predictive accuracy cannot serve as *statistical* evidence of correct causal specification, i.e., evidence of model fit in terms of SEM.

No matter the choice of the distance function  $F$ , four outcomes are of particular interest:  $F = 0$ ,  $F > 0$ ,  $0 < F \leq c$ , and  $F > c$ . Mathematically, models either fit, or they do not. A given model-implied indicator covariance matrix may, for instance, reproduce the population covariance matrix such that both matrices are numerically identical. Mathematically speaking, model fit assessment is therefore an assessment of  $F = 0$  vs.  $F > 0$ . While mathematically appealing, a large body of literature argues that this type of fit may only be of limited interest if applied to answer real-world questions. The epistemological basis of the argument is conceptually rooted in Box (1976)’s famous “all models are wrong, but some are useful” remark which implies that *perfect* fit is, by definition of a model as inherently approximate, *always* wrong. Accordingly, exact fit is therefore typically not of actual interest to researchers trying to study a certain phenomena by means of a model, i.e., an approximation to reality (e.g., Mulaik et al., 1989; Browne and Cudeck, 1992; Hoyle, 2012). Consequently, fit assessment becomes a question of the assessment of *approximate* fit where  $F \neq 0$  is a priori accepted as given (e.g., Steiger, 2007; Bentler, 1990). Hence, any value of  $F$  in the interval  $[0; c]$  is regarded as minor in a sense that 1.) findings and implications derived from the misspecified model are “in line with”

---

<sup>11</sup>Note that Equation (2.8) is on the population level, hence sampling variation is irrelevant here.

the findings and implications that would have been derived from the correctly specified model, and 2.) the model is still useful. While perfect fit is mathematically incontestable and theoretically unambiguous, approximate fit obviously critically hinges on the notation of approximate fit and usefulness.

Mirroring the theoretical concepts of perfect and approximate model fit, distance measures, fit functions, fit indices and model fit testing in the composite model are discussed next.

## 2.4 Distance measures, fit functions and fit indices

The term fit or fitting function is used to refer to a loss function that is minimized by an associated estimator whose name is typically used to determine the name of the fit function. By far the most famous fit function in SEM is the maximum likelihood (ML) fit function

$$F_{ML} = F_{ML}(\mathbf{S}, \Sigma(\boldsymbol{\theta})) = \ln(|\Sigma(\boldsymbol{\theta})|) + \text{tr}(\mathbf{S}\Sigma(\boldsymbol{\theta})^{-1}) - \ln(|\mathbf{S}|) - p, \quad (2.9)$$

where  $p$  is the number indicators. The function is minimized by the ML estimator  $\hat{\boldsymbol{\theta}}_{ML}$  with  $\Sigma(\boldsymbol{\theta})$  replaced by its estimated counterpart  $\hat{\Sigma} = \Sigma(\hat{\boldsymbol{\theta}}_{ML})$  – provided the necessary assumptions for consistency of the ML estimator are met. Other fitting functions known from covariance-based SEM include, the asymptotically distribution free ( $F_{ADF}$ ), the generalized least squares ( $F_{GLS}$ ), and the unweighted least squares ( $F_{ULS}$ ) fitting functions minimized by respective estimators of the same name (e.g., [Bollen, 1989b](#), pp. 111–115 and pp. 426–427).

Fit functions such as  $F_{ML}$ ,  $F_{GLS}$ , and  $F_{ULS}$  are also called distance or discrepancy functions. We refer to a distance measure as mathematical function capturing the discrepancy between the population and the postulated model on a continuous scale. Technically, any feasible discrepancy function is theoretically applicable provided it is increasing in the size of the discrepancy. Moreover, a fixed value (typically zero) should imply perfect fit.

For composite models [Dijkstra and Henseler \(2015a\)](#) propose two distance functions, namely the squared Euclidian distance

$$d_L = \frac{1}{2} \text{tr}[(\mathbf{S} - \Sigma(\boldsymbol{\theta}))^2] = \frac{1}{2} \sum_{i=1}^K \sum_{j=1}^K (s_{ij} - \sigma(\boldsymbol{\theta})_{ij})^2, \quad (2.10)$$

which is identical to  $F_{ULS}$  and the geodesic distance

$$d_G = \frac{1}{2} \sum_{i=1}^K \log(\psi_i)^2, \quad (2.11)$$

which is asymptotically identical to  $F_{ML}$ , where  $\psi_i$  is the  $i$ -th eigenvalue of the matrix  $\mathbf{S}^{-1}\boldsymbol{\Sigma}(\boldsymbol{\theta})$ .

Distance functions are the basis for exact model fit tests discussed in the next section. Moreover, they constitute the basis for a large range of fit indices. These indices can be broadly categorized as absolute or relative (McDonald and Ho, 2002). There are more refined classifications (e.g., Marsh and Hau, 1996; Yuan, 2005; Kline, 2015) that further distinguished between e.g., parsimonious indices, noncentrality-based indices, and goodness-of-fit/badness-of-fit indices, although the classification is fuzzy, as most indices can be characterized according to either category. For the paper at hand a rough distinction is sufficient. Note that in principle, fit indices may also be regarded as specialized distance measures – although a distinction with respect to the specific distance a given fit index actually captures must be made.

## 2.4.1 Measures of absolute fit

### The $\chi^2$ and the $\chi^2/df$ statistic

The  $\chi^2$  statistic was introduced by Jöreskog (1969) and is typically not used as an individual fit index, but rather as the basis for a number of other fit indices. The statistic is defined as:

$$\chi^2 = (N - 1) \cdot F, \quad (2.12)$$

where  $F = F(\mathbf{S}, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$  is a generic distance function capturing the distance between  $\mathbf{S}$  and  $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ . Recognizing the practical limitations of exact model fit testing, Jöreskog (1969, p. 202) suggested to divide the statistic by its respective degrees of freedom and interpret the value as “... an indication that more information can be extracted from the data” or “... an indication that the model fits too well”.

Theoretically, the  $\chi^2$  statistic and the  $\chi^2$  statistic divided by its degrees of freedom can be computed using any distance measure  $F$ , although the  $\chi^2$  statistic is historically closely tied to the ML fitting function  $F_{ML}$  and, consequently, ML. The reason is that the test statistic  $\chi_{ML}^2 = (N - 1) \cdot F_{ML}(\mathbf{S}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_{ML}))$  asymptotically follows a  $\chi^2$  distribution with  $p(p - 1) - q$  degrees of freedom, where  $p$  is the number

of indicators in  $\mathbf{S}$  and  $q$  the number of parameter in  $\boldsymbol{\theta}_{ML}$  (Wilks, 1938). The fact that  $\chi_{ML}^2 \stackrel{a}{\sim} \chi_{df}^2$  can be used to obtain an approximate distribution of the same test statistic with  $\boldsymbol{\theta}_{ML}$  replaced by  $\boldsymbol{\theta}_{PLS}$ . Since PLS using mode B is a consistent and asymptotically normally distributed estimator for the composite model, it asymptotically minimizes the maximum likelihood distance function  $F_{ML}$  from Equation (2.9). The modified test statistic is therefore approximately  $\chi^2$  distributed with  $p(p-1) - q$  degrees of freedom:

$$\chi_{PLS}^2 = (N-1) \cdot F_{ML}(\mathbf{S}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_{PLS})), \quad (2.13)$$

Figure 2.1 shows the results of a small simulation study illustrating the efficacy of the approximation.

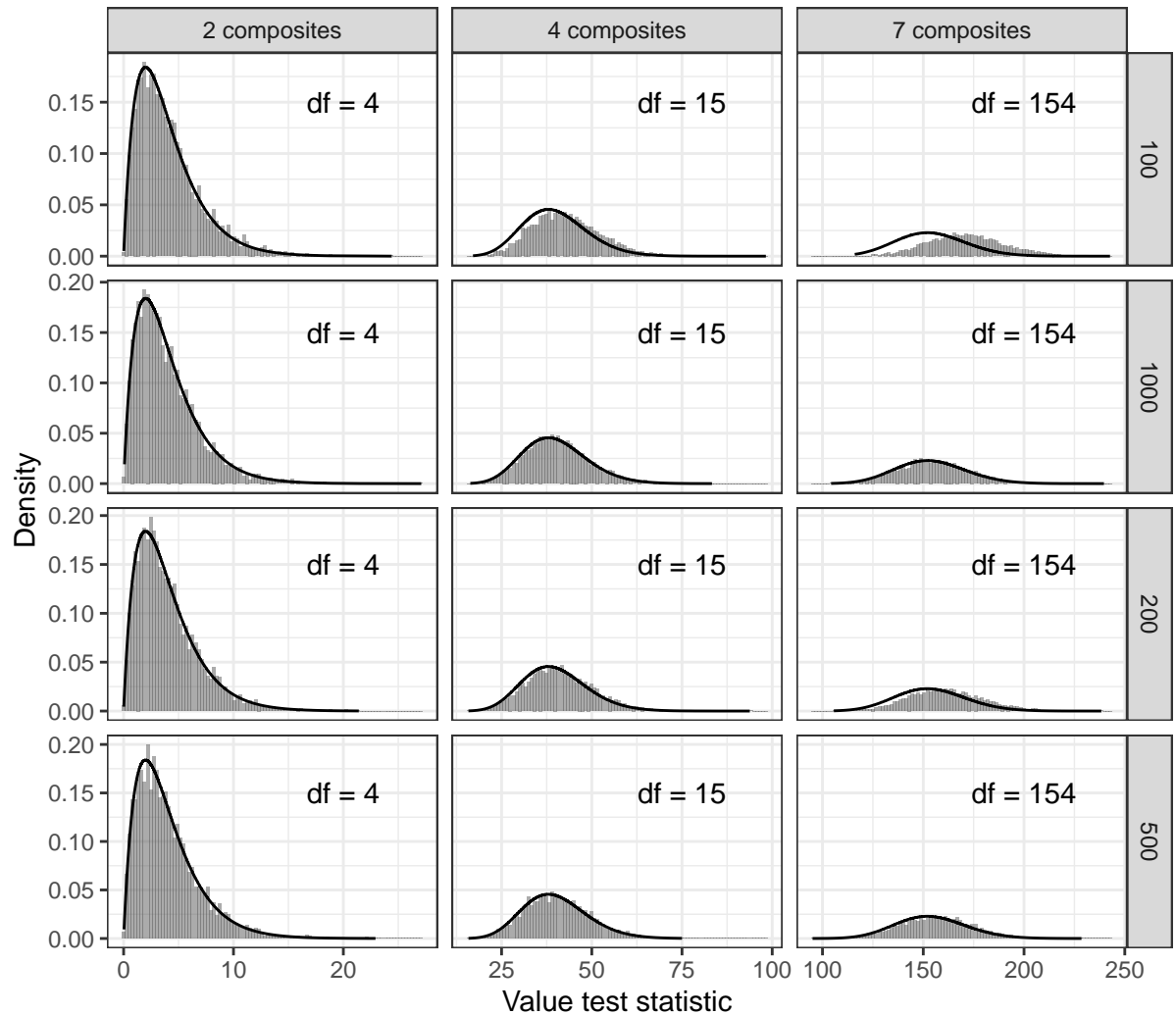


Figure 2.1: Histogramms of the simulated  $\chi^2$  statistic and theoretical  $\chi^2$  distribution.

Each of the 12 tiles show a histogram of the simulated values of the test statistic from Equation (2.13). The solid black line is the theoretical  $\chi^2$  distribution with the

respective number of degrees of freedom calculated according to Equation (2.7). For the simulation, we drew  $M = 5000$  samples of size  $N = \{100, 200, 500, 1000\}$  for 3 data generating processes with varying number of composites. Each composite is build by 3 indicators with randomly assigned unscaled weights of 1, 2, or 3.

Within-block indicator correlations are set to 0.5. For simplicity no structural model was assumed. Instead all composites are correlated with a correlation of 0.3.

For each DGP and sample size, a model was estimated that perfectly corresponds to the DGP the data was drawn from. Hence, the null hypothesis of perfect fit is true.

In this case we expect  $F_{ML}(\mathbf{S}, \Sigma(\hat{\boldsymbol{\theta}}_{PLS}))$  to be close to zero and  $\chi^2_{PLS}$  to follow a  $\chi^2$  distribution with degrees of freedom as shown in Figure 2.1.

Figure 2.1 illustrates that the approximation is extremely accurate for larger sample sizes. In smaller samples the simulated distribution of the test statistic and the theoretical  $\chi^2$  distribution slightly diverge, in particular for more complex models with many degrees of freedom.

Note that contrary to what authors such as e.g., [Hair et al. \(2019b\)](#) suggest, the fact that PLS does not minimize any known distance metric during estimation – but rather maximizes explained variance of the dependent variable –, is irrelevant for the rational of the test statistic; only the choice of the distance function is less obvious – in contrast to ML where the choice is given directly by the minimized function. The only prerequisites are consistency and asymptotic normality of the estimator that is used to derive  $\Sigma(\hat{\boldsymbol{\theta}})$ .

Hence, although PLS’s estimation objective is not to minimize  $F_{ML}$ , it still does so asymptotically when the model is correctly specified with respect to the DGP; this is, of course, if PLS is consistent for the model it estimates.

### **The root mean square residual (RMR) and the standardized root mean square residual (SRMR)**

The root mean square residual (RMR) was first proposed by [Jöreskog and Sörbom \(1982, p.408\)](#) and is given as:

$$\text{RMR} = \sqrt{\frac{\sum_{i=1}^K \sum_{j=1}^i (s_{ij} - \sigma(\boldsymbol{\theta})_{ij})^2}{\frac{K(K+1)}{2}}} = \sqrt{\frac{d_L}{\frac{K(K+1)}{2}}} \quad (2.14)$$

The RMR is the square root of the average of the squared element-wise deviations between all elements – including the main-diagonal elements – of  $\mathbf{S}$  and  $\Sigma(\boldsymbol{\theta})$ , where

$\mathbf{S}$  is the empirical indicator *covariance* matrix and  $\mathbf{\Sigma}(\boldsymbol{\theta})$  is the model-implied *covariance* matrix. The RMR is therefore equal to the square root of the squared Euclidian distance  $d_L$  divided by  $0.5 \cdot (K(K + 1))$ . If  $\mathbf{S}$  and  $\mathbf{\Sigma}(\boldsymbol{\theta})$  represent *correlation* matrices, the RMR and its standardized version, the standardized root mean square residual (SRMR; Bentler, 1995) defined as

$$\text{SRMR} = \sqrt{\frac{\sum_{i=1}^K \sum_{j=1}^i (s_{ij} - \sigma(\boldsymbol{\theta})_{ij}) / \sqrt{s_{ii}^2 s_{jj}^2})^2}{\frac{K(K+1)}{2}}}, \quad (2.15)$$

are identical since in this case the variances are:  $s_{ii}^2 = s_{jj}^2 = 1$ . As it is common to assume standardized indicators in PLS, the RMR and the SRMR are usually identical.

In principal, both the RMR and the SRMR are conceptually meaningful to assess (mis)fit in the composite model, however, a change to the original formula should be considered. As detailed in Section 2.2, indicators of composite  $c_j$  are allowed to freely covary, rendering the difference  $\mathbf{S}_j - \mathbf{\Sigma}(\boldsymbol{\theta})_j$  equal to zero by definition of the composite model. While, the numerator of the (S)RMR remains unchanged, the number of elements to divide by should only be based on the number of non-redundant elements in  $\mathbf{S}_{j_s}$ . Failing to do so essentially artificially deflates the (S)RMR.

### The root mean square outer residual covariance (RMS $_{\theta}$ )

The root mean square outer residual covariance was first proposed by Lohmöller (1989, p. 53; Eq. 2.118) under the same name as a potential candidate “... to evaluate different aspects of the estimates” (Lohmöller, 1989, p. 53). Alluding to the fact that the outer residual covariance is sometimes symbolically referred to by the Greek letter  $\theta$ , Henseler et al. (2014, p. 192) labeled the index RMS $_{\theta}$ , however, they did not provide mathematical details on how its computation was conducted.

The RMS outer residual covariance or RMS $_{\theta}$  as proposed by Lohmöller (1989, p. 53; Eq. 2.118) may be understood and implemented in at least two different ways, each implying a different index. This confusion is likely due to four reasons.

First, Lohmöller (1989) did not give a precise formula for its computation leaving room for interpretation as to what is meant by “*rms outer residual covariance*” (Lohmöller, 1989, p. 53). Second, Lohmöller (1989) uses the term “residual” to refer to the error term (a theoretical quantity), the difference between an observed score

and its predicted counterpart (an empirical, i.e., estimated quantity), and the difference between observed and model-implied covariances which adds scope for interpretation (confusion).<sup>12</sup> Third, Lohmöller (1989) did not motivate the intended use of the RMS outer residual covariance beyond its basic use as a candidate “... to evaluate different aspects of the estimates” (Lohmöller, 1989, p.53), nor did he investigate its statistical properties. Fourth, to the best of our knowledge, since its introduction, the  $\text{RMS}_\theta$  has never been subject to a thorough analysis of its statistical properties – which could have clarified its precise implementation.

To understand where the different readings of the  $\text{RMS}_\theta$  come from, why and how they differ, a brief review of the relevant sections of Lohmöller (1989)’s book are necessary. The root mean square outer residual covariance is defined in Equation (2.118) of the book as

$$c := \text{rms } \mathbf{C}, \quad (2.16)$$

where according to Lohmöller (1989, p. 24):  $\mathbf{C} = \text{cov}(\mathbf{e})$  is the outer residual covariance matrix.<sup>13</sup> The *outer residual* is given as

$$\mathbf{e} = \mathbf{y} - \mathbf{P}\mathbf{Y}, \quad (2.17)$$

with  $\mathbf{y} :=$  manifest variables, data;  $\mathbf{P} :=$  loading pattern coefficients;

$\mathbf{Y} :=$  latent variables. Lohmöller (1989, p. 24) defines the symbols  $\mathbf{y}$ ,  $\mathbf{P}$ ,  $\mathbf{Y}$  and  $\mathbf{e}$  in a column labeled “estimated”. Translating the notation to be consistent with the notation of the current paper, Equation (2.17) is thus identical to

$$\hat{\mathbf{E}}' = \mathbf{X}' - \hat{\mathbf{\Lambda}} \cdot \mathbf{H}', \quad (2.18)$$

---

<sup>12</sup>In SEM a “residual” is usually defined as the difference between observed and *model-implied covariances* (e.g., Hoyle, 2012, p.216).

<sup>13</sup>Lohmöller (1989, p. 52) defines the operator  $\text{rms}(\cdot)$  in Equation (2.108) as:

$$\text{rms } \mathbf{A} = \sqrt{\sum_i \sum_j a_{ij}^2}$$

and notes: “ ... [the operator] is a shortcut for the root mean squared function of a matrix  $\mathbf{A}$ , where the diagonal is excluded if  $\mathbf{A}$  is symmetric”. The definition is a bit unusual in a sense that, 1.) the operator is commonly called the root mean *square* and 2.) the fact that the rms is a mean over squared elements is typically made explicit. While the first point is probably just a somewhat careless use of terminology, the second point is clarified by Equation (1.14) on page 26 stating that:

$$\frac{1}{N} \sum_{n=1}^N y_{kn} = \sum_n y_{kn}$$

Hence,  $\sum_n a_{kn}$  is the arithmetic mean over all  $n$ .



where  $\hat{\mathbf{E}}$  is a  $(N \times K)$  matrix of residuals,  $\mathbf{X}$  the  $(N \times K)$  matrix of standardized data,  $\hat{\mathbf{\Lambda}}$  the  $(K \times J)$  block-diagonal matrix of estimated composite loadings, and  $\mathbf{H}$  the  $(N \times J)$  matrix of composite scores. The dimensions of the matrices follow from pages 25, 28, and 35 of the book.

To compute the  $\text{RMS}_\theta$  the relevant  $(K \times K)$  covariance matrix  $\mathbf{C}$  must be derived. Depending on what Equation (2.17) actually constitutes and what concept of “residual” Lohmöller (1989) had in mind, two different computations approaches of  $\mathbf{C}$  of are conceivable. We label the first the *empirical* and the second the *model-implied* approach.

The empirical approach basically reads “*rms outer residual covariance*” as the root mean square of the *covariances of residuals*. Residual in this case denotes the difference between the observed and the fitted indicator value. Hence, the  $\text{RMS}_\theta$  is simply the root mean square of the elements of

$$\text{Cov}(\hat{\mathbf{E}}) = \frac{\hat{\mathbf{E}}' \hat{\mathbf{E}}}{N - 1}, \quad (2.19)$$

where the means of the columns of  $\hat{\mathbf{E}}$  are by definition equal to zero.<sup>14</sup> Diagonal elements of  $\text{Cov}(\hat{\mathbf{E}})$  are included in this case. Writing the covariances in terms of right hand side variables of Equation (2.18) gives

$$\begin{aligned} \hat{c}_1 \equiv \text{Cov}(\hat{\mathbf{E}}) &= \frac{1}{N - 1} (\mathbf{X}' - \hat{\mathbf{\Lambda}} \mathbf{H}') \cdot (\mathbf{X}' - \hat{\mathbf{\Lambda}} \mathbf{H}')' \\ &= \frac{1}{N - 1} \mathbf{X}' \mathbf{X} - \frac{1}{N - 1} \mathbf{X}' \mathbf{H} \hat{\mathbf{\Lambda}}' - \frac{1}{N - 1} \hat{\mathbf{\Lambda}} \mathbf{H}' \mathbf{X} + \frac{1}{N - 1} \hat{\mathbf{\Lambda}} \mathbf{H}' \mathbf{H} \hat{\mathbf{\Lambda}}' \\ &= \mathbf{S} - \mathbf{T} \hat{\mathbf{\Lambda}}' - \hat{\mathbf{\Lambda}} \mathbf{T}' + \hat{\mathbf{\Lambda}} V(\mathbf{H}) \hat{\mathbf{\Lambda}}' \end{aligned} \quad (2.20)$$

where  $V(\mathbf{H})$  is the  $(J \times J)$  matrix of empirical composite correlations and  $\mathbf{S}$  the  $(K \times K)$  empirical indicator covariance matrix.  $\mathbf{T}$  is the  $(K \times J)$  “loadings structure coefficient matrix” Lohmöller (1989, p.24) which is the indicator-composite covariance matrix sometimes referred to as the matrix of “cross-loadings”. Since the composite score  $\mathbf{H}$  is equal to  $\mathbf{X} \hat{\mathbf{W}}$ , where  $\hat{\mathbf{W}}$  is the estimated  $(K \times J)$  weight matrix of the same structure as  $\hat{\mathbf{\Lambda}}$ , we have:

$$\mathbf{T} = \frac{1}{N - 1} \mathbf{X}' \mathbf{H} = \frac{1}{N - 1} \mathbf{X}' \mathbf{X} \hat{\mathbf{W}} = \mathbf{S} \hat{\mathbf{W}} \quad (2.21)$$

---

<sup>14</sup>Whether to divide by  $N$  or  $N - 1$  makes hardly any difference.

The  $\text{RMS}_\theta$  follows as

$$\text{RMS}_\theta^{\text{emp}} = \sqrt{\frac{1}{K(K+1)/2} \sum_i^K \sum_j^i \hat{c}_{1;ij}^2}, \quad (2.22)$$

where  $\hat{c}_{1;ij}$  is the  $ij$ -th element of  $\hat{c}_1$ .

The model-implied approach considers Equation (2.17) as a model equation and  $\mathbf{e}$  as the estimated model-implied residual. The model in this case is the reflective measurement model. Accordingly, Equation (2.18) is written on the population level as:

$$\boldsymbol{\varepsilon} = \mathbf{x} - \boldsymbol{\Lambda}\mathbf{c} \quad (2.23)$$

The rms outer residual covariance is thus given as the root mean square of the *conditional, model-implied, covariances of errors*, where the error is the difference between the random matrix  $\mathbf{x}$  and its conditional expectation implied by the reflective measurement model  $E(\mathbf{x}|\mathbf{c}) = \boldsymbol{\Lambda}\mathbf{c}$ . For the covariance expressed in terms of the right hand side variables of Equation (2.23) we obtain:

$$\begin{aligned} c_2 \equiv \text{cov}(\boldsymbol{\varepsilon}) &= E(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'|\mathbf{c}) = E[(\mathbf{x} - \boldsymbol{\Lambda}\mathbf{c}) \cdot (\mathbf{x} - \boldsymbol{\Lambda}\mathbf{c})'|\mathbf{c}] \\ &= E(\mathbf{x}\mathbf{x}'|\mathbf{c}) - E(\mathbf{x}|\mathbf{c})\mathbf{c}'\boldsymbol{\Lambda}' - \boldsymbol{\Lambda}\mathbf{c}E(\mathbf{x}'|\mathbf{c}) + \boldsymbol{\Lambda}E(\mathbf{c}\mathbf{c}'|\mathbf{c})\boldsymbol{\Lambda}' \\ &= \boldsymbol{\Sigma} - \boldsymbol{\Lambda}\mathbf{c}\mathbf{c}'\boldsymbol{\Lambda}' - \boldsymbol{\Lambda}\mathbf{c}\mathbf{c}'\boldsymbol{\Lambda}' + \boldsymbol{\Lambda}\mathbf{c}\mathbf{c}'\boldsymbol{\Lambda}' \\ &= \boldsymbol{\Sigma} - \boldsymbol{\Lambda}\boldsymbol{\Sigma}_c\boldsymbol{\Lambda}' \\ &= \boldsymbol{\Sigma} - \boldsymbol{\Sigma}(\boldsymbol{\theta}) \end{aligned} \quad (2.24)$$

Here,  $\boldsymbol{\Sigma}_c$  is the conditional (model-implied) composite covariance matrix and  $\boldsymbol{\Sigma}(\boldsymbol{\theta})$  the model-implied indicator covariance matrix. Since  $\boldsymbol{\Sigma}$  and  $\boldsymbol{\Sigma}(\boldsymbol{\theta})$  are theoretical quantities they are replaced by their empirical/estimated counterparts  $\mathbf{S}$  and  $\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}})$  to obtain the (model-implied) outer residual covariance  $\hat{c}_2 = \text{Cov}(\mathbf{e}) = \mathbf{S} - \hat{\boldsymbol{\Sigma}}$ . The computation of the  $\text{RMS}_\theta$  in this case is given by

$$\text{RMS}_\theta^{\text{mi}} = \sqrt{\frac{1}{K(K+1)/2} \sum_i^K \sum_j^i \hat{c}_{2;ij}^2}, \quad (2.25)$$

where  $\hat{c}_{2;ij}^2 = (s_{ij} - \hat{\sigma}_{ij})^2$  and therefore:

$$\text{RMS}_\theta^{\text{mi}} = \sqrt{\frac{1}{K(K+1)/2} \sum_i^K \sum_j^i (s_{ij} - \hat{\sigma}_{ij})^2} = (\text{S})\text{RMR} \quad (2.26)$$

Without context, neither  $\hat{c}_1$  nor  $\hat{c}_2$  is statistically preferable over the other. However, if – as it is reasonable to assume – the intended use of the  $\text{RMS}_\theta$  is as a criteria for

model fit assessment only  $\text{RMS}_\theta^{mi}$  adequately serves this purpose. Since  $\hat{c}_1$  cannot generally be expected to be zero when the postulated model perfectly reproduces the population indicator covariance matrix. In fact, even if  $\mathbf{S} - \hat{\mathbf{\Lambda}}V(\mathbf{H})\hat{\mathbf{\Lambda}}'$  in Equation (2.20) is zero, i.e., the estimated model-implied indicator covariance matrix is identical to the empirical indicator covariance matrix, the remaining terms  $-\mathbf{T}\hat{\mathbf{\Lambda}}'$  and  $-\hat{\mathbf{\Lambda}}\mathbf{T}'$  do not necessarily cancel. Hence,  $\text{RMS}_\theta^{emp}$  is non-zero for perfectly fitting models. This lack of reference to perfect fit renders  $\text{RMS}_\theta^{emp}$  essentially unusable for model fit assessment.

The  $\text{RMS}_\theta^{mi}$  is by definition equal to zero if  $\mathbf{\Sigma} = \mathbf{\Sigma}(\boldsymbol{\theta})$  in the population or – if replaced by their empirical counterparts –  $\mathbf{S} = \hat{\mathbf{\Sigma}}$ . However, in this case the  $\text{RMS}_\theta$  is identical to the (S)RMR as defined in Equation (2.14).

We therefore conclude that the  $\text{RMS}_\theta$  is not a useful fit index unless implemented such that it equals the (S)RMR in which both indices coincide.

### **The root mean square error of approximation (RMSEA)**

The RMSEA is based on two related insights. First, the overall lack of fit between the population indicator covariance matrix  $\mathbf{\Sigma}$  and the estimated model-implied indicator covariance matrix  $\hat{\mathbf{\Sigma}}$  can be decomposed into an error of approximation and an error of estimation (Browne and Cudeck, 1992, p. 236f.). The error of approximation is captured by the discrepancy  $F(\mathbf{\Sigma}, \tilde{\mathbf{\Sigma}})$  – the so-called *discrepancy due to approximation* (Browne and Cudeck, 1992, p. 236) –, where  $\tilde{\mathbf{\Sigma}} = \mathbf{\Sigma}(\boldsymbol{\theta}_0)$  is the model-implied indicator covariance matrix that would theoretically yield the best fit to  $\mathbf{\Sigma}$ ; this is, the discrepancy that were to arise if the model were fitted using the population covariance matrix. Hence, if the discrepancy  $F(\mathbf{\Sigma}, \tilde{\mathbf{\Sigma}})$  is non-zero, no  $\boldsymbol{\theta}$  exists such that  $\mathbf{\Sigma} = \mathbf{\Sigma}(\boldsymbol{\theta})$ . Of course, both  $\mathbf{\Sigma}$  and  $\tilde{\mathbf{\Sigma}}$  are non-stochastic unknown matrices. The error of estimation refers to the difference between  $\tilde{\mathbf{\Sigma}}$  – the best possible matrix closest to  $\mathbf{\Sigma}$  – and the estimated model-implied indicator covariance matrix  $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma}(\hat{\boldsymbol{\theta}})$  obtained using the sample covariance matrix  $\mathbf{S}$ . The error of estimation is captured by the discrepancy  $F(\tilde{\mathbf{\Sigma}}, \hat{\mathbf{\Sigma}})$  which is referred to as the *discrepancy due to estimation* (Browne and Cudeck, 1992, p. 236). Formally:

$$F(\mathbf{\Sigma}, \hat{\mathbf{\Sigma}}) = F(\mathbf{\Sigma}, \tilde{\mathbf{\Sigma}}) + F(\tilde{\mathbf{\Sigma}}, \hat{\mathbf{\Sigma}}) \quad (2.27)$$

The second insight refers to the  $\chi^2$  test statistic (Equation (2.12)). The observed discrepancy  $F(\mathbf{S}, \hat{\Sigma})$  asymptotically either follows a central – if  $\Sigma = \Sigma(\theta)$  – or noncentral – if  $\Sigma \neq \Sigma(\theta)$  –  $\chi^2$  distribution with noncentrality parameter  $\lambda = (N - 1) \cdot F(\Sigma, \tilde{\Sigma})$ . Using Browne and Cudeck (1992), Proposition 1, an estimate of  $F(\Sigma, \tilde{\Sigma})$  and  $\hat{\lambda}$  is given by:

$$\hat{F} = \max\left(F(\mathbf{S}, \hat{\Sigma}) - \frac{df}{N-1}; 0\right) \quad \text{and} \quad \hat{\lambda} = \max\left((N-1) \cdot F(\mathbf{S}, \hat{\Sigma}) + df; 0\right) \quad (2.28)$$

The RMSEA is the root mean square of the error of approximation per degree of freedom (Steiger and Lind, 1980; Browne and Cudeck, 1992). Using  $\hat{F}$  as an estimate of  $F(\mathbf{S}, \hat{\Sigma})$ , the estimated RMSEA is given by:

$$\text{RMSEA} = \sqrt{\frac{\hat{F}}{df}} \quad (2.29)$$

To obtain the RMSEA, an appropriate fit or distance function is required. For the common factor model  $F_{ML}$  is usually the best choice. For the composite model estimated by a composite-based approach such as PLS, we use the same argument as for the central  $\chi^2$  distribution from Section 2.4.1. Plugging  $\hat{\theta}_{PLS}$  into  $F(\mathbf{S}, \hat{\Sigma})$  approximately minimizes the distance measure. Consequently,  $\chi_{PLS}^2 = (N - 1) \cdot F(\mathbf{S}, \hat{\Sigma}_{PLS})$  is expected to follow a noncentral chi square distribution if the model is misspecified with respect to the population, i.e.,  $F(\Sigma, \tilde{\Sigma}) > 0$ .

To illustrate, we conducted a small Monte Carlo simulation. The simulation setup is similar to Section 2.4.1. Samples of size  $N = \{100, 200, 500, 1000\}$  for two DGPs with 4 or 6 composites are drawn. Each composite is build by 3 indicators with randomly assigned unscaled weights of 1, 2, or 3. Intra-block indicator correlations are all equal to 0.5. All composites are correlated with a correlation of 0.3.

For each DGP and sample size,  $M = 5000$  data sets are drawn. For each data set a misspecified model was estimated. As misspecification we imposed a path model on the relationship between composites. Since all composites are correlated with a correlation of 0.3 in the DGP, restricting certain paths to zero introduces an error of approximation. The path models are given in Figure 2.2.<sup>15</sup>

<sup>15</sup>For our purpose the nature of the misspecification is irrelevant, as we only require the model to exhibit some kind of misspecification. Alternative misspecifications such as a different path model, falsely assigned indicators, falsely omitting a composite etc. led to virtually identical results.

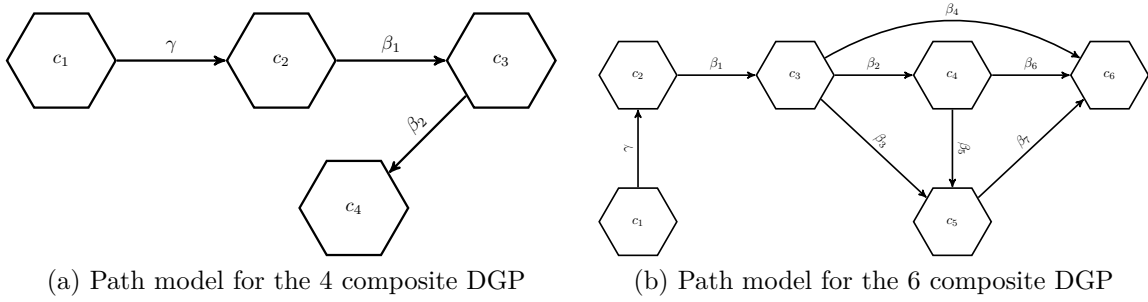


Figure 2.2: Misspecified models

In each case we expect  $F_{ML}(\mathbf{S}, \boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}}_{PLS}))$  to be systematically non-zero and  $\chi_{PLS}^2$  to follow a noncentral  $\chi^2$  distribution with estimated noncentrality parameter

$$\hat{\lambda} = \frac{1}{M} \sum_{i=1}^M (\chi_{PLS;i}^2 - df_i) \quad (2.30)$$

and degrees of freedom computed according to Equation (2.7). Figure 2.3 illustrates that – similar to Figure 2.1 – the approximation is extremely accurate even in small samples.

## 2.4.2 Measures of relative fit

In contrast to absolute fit indices, relative fit indices evaluate the goodness of fit of a postulated model relative to a competing model. Let  $M^0$  denote the null,  $M^p$  the postulated,  $M^k$  an alternative, and  $M^s$  the saturated model. We assume that models are nested, i.e.,  $M^0 \subset M^p \subset M^k \subset M^s$ . For each model corresponding distance measures  $F^0$ ,  $F^k$ ,  $F^p$ , and  $F^s$  with degrees of freedom  $df_0$ ,  $df_k$ ,  $df_p$ , and  $df_s$  are available. Of all possible models, the saturated model is the least restricted model possible in a sense that there are as many model parameters as there are off-diagonal elements in  $\mathbf{S}$ ; hence, only the identifying restrictions are placed on the model and no testable restrictions remain, i.e.,  $F^s = 0$  and  $df_s = 0$ . Likewise the “suitably framed null model” (Bentler and Bonett, 1980, p. 595) is the most restricted theoretically defensible model. Moreover, we assume that the alternative model is more restricted than the postulated model. Hence:  $df_0 > df_k > df_p > df_s = 0$  and – if evaluated with the same distance metric –  $F^0 \geq F^k \geq F^p \geq F^s = 0$ .

Note that the null model does not necessarily have to be the most restricted model possible but rather “...one that would reasonably be considered in practice” (Bentler, 1990, p. 239). In the context of the latent variable model three such null models have

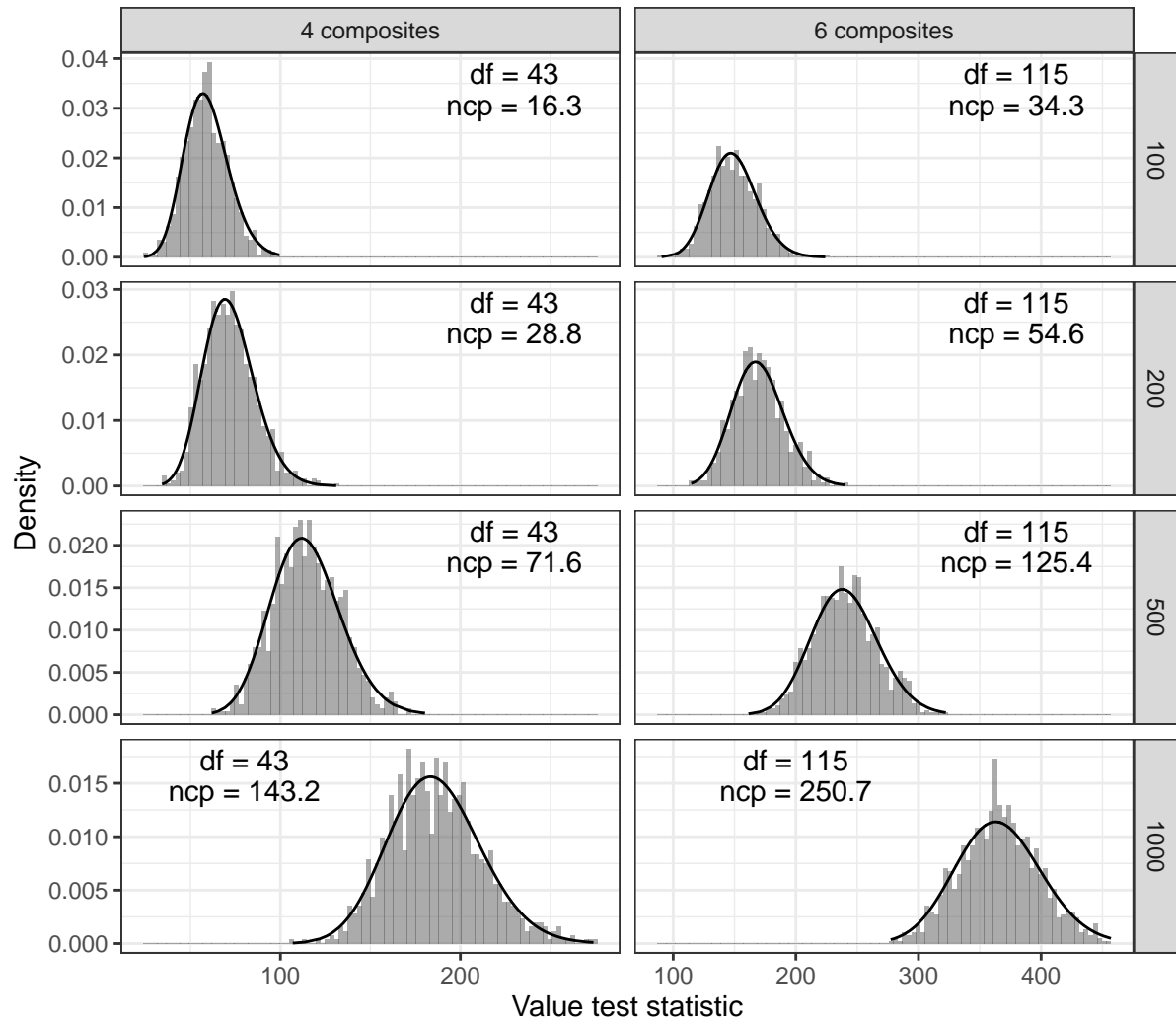


Figure 2.3: Histograms of the simulated  $\chi^2$  statistic and theoretical noncentral  $\chi^2$  distribution.

been suggested. The first null model was referred to as “no model at all” by [Jöreskog and Sörbom \(1993, p. 122\)](#). The “model-implied” indicator covariance matrix is accordingly equal to the null matrix:

$$\Sigma(\theta) = \mathbf{0} \tag{2.31}$$

Such a “model” is unlikely to ever qualify as theoretically defensible in practice and has, to the best of our knowledge, not been used in practice as a realistic null model. The second null model, commonly called the independence model (e.g., [Bentler and Bonett, 1980, p. 596](#)), restricts all indicator covariances to zero, yielding a model-implied indicator covariance equal to the identity matrix:<sup>16</sup>

$$\Sigma(\theta) = \mathbf{I} \tag{2.32}$$

<sup>16</sup>We assume standardized indicators again.

This is the most common operationalization of the null model. The third null model is called the “general” null model (Bentler and Bonett, 1980, p. 596). Such a model would be similar to the independence model with selected indicator covariances left unrestricted.

For the composite model, we suggest a modified independence model as a suitable benchmark null model that accounts for the fact that, by assumption of the composite model, indicators are allowed to freely covary within blocks. Hence, the model-implied indicator covariance matrix of the modified independence model is given by the the block diagonal matrix

$$\Sigma(\boldsymbol{\theta}) = \text{diag}(\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_J), \quad (2.33)$$

where  $\mathbf{S}_j$  is the  $(K_j \times K_j)$  empirical covariance matrix of the indicators belonging to composite  $c_j$ .

### The normed fit index (NFI)

The index first appeared in Bentler and Bonett (1980) and is defined as:

$$\text{NFI} = \frac{F^0 - F^p}{F^0} \quad \text{or alternatively} \quad \text{NFI}_{kp} = \frac{F^k - F^p}{F^0} \quad (2.34)$$

The NFI measures the increase in fit relative to the fit of the null mode (NFI) or an alternative model ( $\text{NFI}_{kp}$ ). The index can take values between 0 and 1 with 0 indicating no improvement in fit over the null (or alternative) model. A value of 1 of the NFI implies perfect fit of the postulated model, i.e.  $F^p = 0$ . Its intuition is directly applicable to the composite model.

### The nonnormed fit index (NNFI)

The NNFI first appeared in Bentler and Bonett (1980) by that name. The index is identical to an index proposed by Tucker and Lewis (1973) which is commonly labeled the Tucker-Lewis index (TLI). The NNFI is defined as:

$$\text{NNFI} = \frac{\frac{F^0}{df_0} - \frac{F^p}{df_p}}{\frac{F^0}{df_0} - \frac{1}{N-1}} \quad \text{or alternatively} \quad \text{NNFI}_{kp} = \frac{\frac{F^k}{df_k} - \frac{F^p}{df_p}}{\frac{F^0}{df_0} - \frac{1}{N-1}} \quad (2.35)$$

The NNFI measures the discrepancy between the fit of two competing models per degree of freedom relative to the null model.

The construction logic of the index is based on properties that are directly related to the (noncentral)  $\chi^2$  distributed test statistic; the denominator  $F^0/df_0 - 1/(N - 1)$ , for instance, can be rewritten as  $(N - 1)^{-1}df_0^{-1}(\chi_0^2 - df_0)$ . When rewriting the numerator as well, the term  $(N - 1)^{-1}df_0^{-1}$  cancels and  $\chi_0^2 - df_0$  is simply the difference between the  $\chi_0^2$  test statistic of the null model and its expected value  $E(\chi_0^2) = df_0$  which, in turn, is used as an estimate of the noncentrality parameter  $\lambda$ . The same applies to other closely related  $\chi^2$ -dependent indices such as the modified NNFI (Bollen, 1986), the relative noncentrality index (RNI; Bentler, 1990), and the incremental fit index (IFI; Bollen, 1989a). Consequently, the NNFI – and related  $\chi^2$ -based indices – may be used for composite model estimated by PLS, as long as the alternative test statistic  $\chi_{PLS}^2$  exhibits the same properties as  $\chi^2$  test statistic based on ML.

### The comparative fit index (CFI)

The CFI was first proposed by Bentler (1990). It is defined as:

$$\text{CFI} = 1 - \frac{\max(0, (N - 1) \cdot F^p - df_p)}{\max(0, (N - 1) \cdot F^p - df_p, (N - 1) \cdot F^0 - df_0)} \quad (2.36)$$

Like the RMSEA, the CFI is a non-centrality based index. It measures the increase in fit – that is, the reduction in non-centrality – of the postulated model relative to the fit of the null or another alternative model. The CFI is a normed index with a value of 1 indicating the best fit. Like the NNFI, its construction rationale is based on the noncentrality of the  $\chi^2$  statistic under violation of the null hypothesis of perfect fit. Its intuition therefore does apply to the composite model estimated by a composite-based estimator, provided  $\chi_{PLS}^2$  exhibits the same properties as the  $\chi^2$  test statistic based on ML.

Note, the relative fit indices discussed have originally been proposed using the value of the  $\chi^2$  test statistic from Equation (2.12) but have been redefined to be a function of the distance function  $F$  (e.g., Bollen, 1986). We use this notation to emphasize that the relative indices can be computed for arbitrary fitting or distance functions although the construction rationale of, for instance, the NNFI or the CFI is inherently tied to properties of the  $\chi^2$  test statistic and its corresponding noncentral  $\chi^2$  distribution.



### 2.4.3 Fit indices and cut-off

Once a particular value for a given fit index and model has been obtained, researchers want to know whether the value can be considered as evidencing “sufficiently good fit” or “bad fit”. To classify the magnitude of a given fit index such as the RMSEA or the SRMR as “sufficiently good”, conventional recommendations rely on one-size-fits-all cut-offs that essentially dichotomize the continuum of possible fit values into an acceptable/non-acceptable region, similar to the hypothesis testing framework. While certainly appealing for practitioners struggling to decide how to handle a given fit index value, the use of cut-offs as the decision reference suffers from statistical, conceptual, and logical shortcomings. To understand these issues it helps to remember why fit indices are used in the first place and how current cut-off recommendations have historically been derived.

Using fit indices with the intention to statistically assess the goodness/badness of the postulated model is implicitly based on the idea that an unknown interval  $[0, c]$  of fit values exists that indicates “acceptable misspecification” (approximate fit). Whereas the lower bound of the interval has a clear interpretation – perfect fit in the SEM sense –, the upper bound, and thus the size of the interval, is difficult to determine. In fact, unless the distribution  $D$  of the entirety of misspecifications that could theoretically be detected by a given fit index is known,  $c$  can only be determined based on domain knowledge and subjective judgment. If the distribution  $D$  was known, researchers could compare the fit index value obtained based on the postulated model to a quantile of  $D$  in order to judge whether a given index constitutes a comparatively small (if the fit value is smaller than, say, the 5% quantile of  $D$ ) or large misspecification (e.g., fit value  $>$  95% quantile). Clearly, there is also subjectivity involved when determining a reference quantile below which fit index values are taken to indicate a small misspecification; however, the decision rule is objective since  $D$  is fix and logically in line with the idea of approximate fit. Yet, since  $D$  will never be known, researchers face the conundrum of determining  $c$  without knowing the distribution  $D$ .

But are common cut-off values such as “SRMR  $<$  0.08” not the upper bound  $c$  we are looking for? Considering how these cut-offs historically arose, the answer is no. The popularity of most common cut-off recommendations can be traced back to [Hu and Bentler \(1999\)](#)’s seminal work. The authors essentially applied the hypothesis testing

method box to determine optimal cut-offs that balance the type I error rate – falsely rejecting the null hypothesis when it is true – and the type II error rate – falsely “accepting” the null hypothesis when it is false. The idea of [Hu and Bentler \(1999\)](#) was to determine the type I error rate for a range of fit indices with respect to several model complexities, sample sizes and cut-offs. The type II error rate for the same fit indices was determined with respect to models of varying magnitude of misspecification using different sample sizes and cut-offs. Accordingly, the type I error rate was determined as the  $\alpha\%$  quantile of the simulated distribution of the fit index *under the hypothesis of perfect fit*. Hence, for a goodness-of-fit measure,  $c_\alpha$  is interpreted in accordance with the interpretation of a quantile as the value that subsumes  $\alpha\%$  of the probability mass to its left. Likewise, the type II error rate is the share of misspecified models that have not been rejected. There are at least three issues with this approach (e.g. [Marsh et al., 2004](#); [Sivo et al., 2006](#); [Greiff and Heene, 2017](#); [Niemand and Mai, 2018](#))

First, the distribution of the fit indices is influenced by factors other than the degree of misspecification they attempt to quantify. The major factors identified as influencing the distribution are sample size, number of constructs and number of indicators (i.e., model complexity), the distribution of the data, and the sizes of the population parameters – see e.g., [Yuan \(2005\)](#) or [Niemand and Mai \(2018\)](#) for an overview. In their paper, [Hu and Bentler \(1999\)](#) were careful not to generalize their results beyond the specific simulation design noting that “...it is difficult to designate a specific cut-off value for each fit index because it does not work equally well with various conditions”. Practically, however, – despite mantra like calls to the opposite – cut-offs are de facto interpreted as universal rules.

Second, acknowledging sensibility of cut-offs to factors other than the magnitude of misspecification, proponents argue that the issue is not the cut-off value per se but how it is used and interpreted. Clearly, mindless applications of cut-off rules (of thumb) will never constitute good scientific practice, however, even if applied “carefully” the actual values [Hu and Bentler \(1999\)](#) proposed are partly based on a flawed logic as illustrated by [Marsh et al. \(2004\)](#). The authors illustrate that the cut-off values derived by [Hu and Bentler \(1999\)](#) heavily hinge on so-called acceptable-misspecified models – essentially models that are misspecified but should have been classified as acceptable according to the cut-off the authors themselves

propose. In other words, the “optimal cut-offs” found by [Hu and Bentler \(1999\)](#) may, in fact, not be optimal when applied to a real-world situation.

Third, the concept of approximate fit implicitly or explicitly comprises the idea of the statistical model as an approximation to reality. The model is accepted as wrong and the hypothesis of the perfect fit is rejected from the outset. Against this background, even if the cut-offs were based on a sound statistical procedure and independent of factors other than the misspecification they attempt to quantify, determining a cut-off with reference to a critical quantile of the distribution of an index under perfect fit contradicts the very concept of approximate fit. Hence, using a decision criterion obtained under the assumption of perfect fit for an index whose purpose is to be applied precisely when perfect fit does not hold, seems logically incoherent. In fact, [Marsh et al. \(2004\)](#) show that in order to correctly distinguish between correctly and incorrectly specified models, a conventional hypothesis test based on the  $\chi^2$  test statistic, as discussed next, always outperforms the index plus cut-off decision strategy.

## 2.5 Model fit tests

Model fit tests assess the null hypothesis of perfect (exact) fit, i.e.,  $H_0 : F = 0$ .

Hence, tests are by construction agnostic to the concept of approximate fit. By far the most common statistical test to empirically assess exact model fit in the context of a structural equation model estimated by a covariance-based estimator such as ML is the  $\chi^2$  test (e.g., [Jöreskog, 1969](#)) based on the  $\chi^2$  test statistic (Equation 2.12).

The test typically involves the ML fit function  $F_{ML}$  (Equation 2.9) although other fit or distance measures, given a set of assumptions hold, may be used as well ([Bollen, 1989b](#)). Since  $\Sigma$  and  $\Sigma(\theta)$  are unknown, the estimated fit function  $\hat{F} = F(\mathbf{S}, \hat{\Sigma})$  is used, where  $\hat{\Sigma} = \Sigma(\hat{\theta})$  represents the estimated model-implied indicator covariance matrix of the postulated model based on the empirical indicator covariance matrix  $\mathbf{S}$  – which is assumed to be representative of the population indicator covariance matrix  $\Sigma$ .<sup>17</sup> As discussed in Section 2.4.1, using  $\hat{F} = F_{ML}(\mathbf{S}, \Sigma(\hat{\theta}_{PLS}))$  asymptotically minimizes the ML fit function. The resulting test statistic is thus approximately  $\chi^2$  distributed under correctness of the null hypothesis – this is, if observations are

---

<sup>17</sup>Mathematically speaking we must have  $\text{plim } \mathbf{S} = \Sigma$ .

independent draws from a multivariate normal distribution. Hence, a simple  $\chi^2$  test could be used to test exact model fit of composite models estimated by PLS.

For a structural equation model estimated using PLS, [Dijkstra and Henseler \(2015a\)](#) introduced the bootstrap-based test for exact overall model fit. Similarly to the  $\chi^2$  test, the bootstrap-based test utilizes a discrepancy function capturing the distance between  $\mathbf{S}$  and  $\Sigma(\hat{\boldsymbol{\theta}})$  to infer how well  $\Sigma$  and  $\Sigma(\boldsymbol{\theta})$  fit in the population. [Dijkstra and Henseler \(2015a\)](#) use the squared Euclidian distance  $d_L$  (Equation (2.10)) and the geodesic distance (2.11). In addition, [Schuberth et al. \(2018a\)](#) propose to use the SRMR. In principal, several distance measures qualify.

Instead of relying on the  $\chi^2$  as the reference distribution, the distribution of the test statistic *under the null* is obtained via bootstrap. Bootstrap based on the original sample  $\mathbf{X}$  is unfeasible since  $\mathbf{S} - \Sigma(\boldsymbol{\theta})$  does not satisfy the null hypothesis. To obtain the distribution under the null hypothesis, [Bollen and Stine \(1992\)](#) propose to use a general transformation procedure based on the work of [Beran and Srivastava \(1985\)](#). Instead of using  $\mathbf{X}$  directly, bootstrap is based on the transformed data:

$$\mathbf{X}^* = \mathbf{X} \mathbf{S}^{-\frac{1}{2}} \hat{\Sigma}^{\frac{1}{2}} \quad (2.37)$$

Note that now by definition:

$$\mathbf{S}^* = \text{E}(\mathbf{X}'^* \mathbf{X}^*) = \hat{\Sigma} \quad \text{and consequently} \quad \mathbf{S}^* - \hat{\Sigma} = \mathbf{0}.$$

## 2.6 Discussion

Model fit assessment is widely recognized as crucial part in quantitative causal research. In the context of the latent variable model estimated by e.g., maximum likelihood (ML), the amount of literature written on the subject is huge. In contrast, for the composite model estimated by a composite-based approach such as the partial least squares algorithm (PLS), research is limited with some authors (e.g., [Hair et al., 2019a,b](#)) even questioning its necessity for models estimated by PLS. We believe that the controversy surrounding model fit assessment, in particular in the context models estimated by PLS, is due to different apprehensions of the concepts, assumptions and terminology surrounding the issue of model fit assessment in general.

To clarify, we thoroughly discussed the concept of model fit in SEM and argued that fit assessment is both a question of what kind of fit a particular fit metric assesses –

fit in the SEM sense, regression fit or predictive fit – and what kind of fit is called for given the scientific goal of the researcher. If the goal is to answer any kind of inferential, causal or mechanistic question, model fit assessment in the SEM sense is the only known way to obtain *statistical* evidence with respect to the validity of answers to such questions. If, on the other hand, prediction is the goal, model fit in the SEM sense is virtually irrelevant, if not a nuisance.

Fit in the SEM sense is traditionally assessed using a null hypothesis significance test or a range of fit indices. Both are discussed in detail against the background of the composite model estimated by PLS. We find that the fit indices discussed in this paper are, in theory, indeed applicable to the composite model estimated by PLS. For the root mean square residual (RMR), the standardized root mean square residual (SRMR) and the normed fit index (NFI) application and interpretation of the index values is analogous to traditional SEM; however, to better reflect the restrictions of the composite model, we suggest to divide by the number of off-diagonal elements of the empirical indicator covariance matrix, excluding the number of non-redundant intra-block indicator covariances when computing the RMR and SRMR. In contrast to RMR, SRMR, and NFI, the root mean square error of approximation (RMSEA), the nonnormed fit index (NNFI), and the comparative fit index (CFI) rely on the  $\chi^2$  test statistic and its properties. For composite models estimated by PLS no  $\chi^2$  statistic is available. To this end, we suggest a slightly modified  $\chi^2$  test statistic which is based on the ML fit function with PLS estimates instead of ML estimates. A small simulation confirmed that the test statistic approximately follows a  $\chi^2$  distribution with degrees of freedom equal to the number of non-redundant elements of the empirical indicator covariance matrix minus the number of model parameters. Likewise, for misspecified models the test statistic approximately follows a noncentral  $\chi^2$  distribution with estimated noncentrality parameter

$$\hat{\lambda} = \max \left( (N - 1) \cdot F(\mathbf{S}, \hat{\Sigma}) + df; 0 \right).$$

Concerning the root mean square outer residual covariance ( $\text{RMS}_\theta$ ) we showed that the index is numerically identical to the SRMR, unless it is implemented in a way that makes it inadequate for model fit assessment in the SEM sense. We therefore recommend not to use the  $\text{RMS}_\theta$  unless it is computed identically to the SRMR.

With respect to the exact test of overall model fit, the bootstrap-based test suggested by [Dijkstra and Henseler \(2015a\)](#) is available. Alternatively, a classical  $\chi^2$  test based

on the modified test statistic could be used. Whether a fully parametric theoretical or a non-parametric bootstrap reference distribution is preferable largely depends on how certain a researcher is with respect to the assumptions required for the  $\chi^2$  test statistic to indeed follow a  $\chi^2$  distribution. In any case, the relevant hypothesis is exact fit. Similarly, whether a rejection of the hypothesis of perfect fit therefore necessarily implies that a given model should be discarded depends on whether a researcher believes that a model can still be useful even if there is evidence against perfect fit.

As usual results based on a Monte Carlo simulation using a particular design do not automatically generalize; simulation-based findings are a piece of statistical evidence whose informative value grows with similar evidence. However, in this particular setup, results are indeed likely to generalize to other kinds of model setups and misspecifications as long as observations are assumed to be independent draws from a normal distribution. Under non-normality the  $\chi^2$  test statistic generally does not follow a  $\chi^2$  distribution. Further research could investigate whether the modified  $\chi^2$  test statistic based on the maximum likelihood fit function with PLS estimates instead of maximum likelihood estimates behaves similarly to its ML sibling for different kinds of non-normal data. Moreover, it would be interesting to learn, whether common correction such as the well-known Satorra-Bentler correction for non-normality ([Satorra and Bentler, 2001](#)) could be applied in this case.

Although, fit indices can be helpful, we remain skeptical regarding their current use, in particular with respect to the pervasive use of cut-off values as decision criteria. Clearly, cut-off values have great appeal for practitioners, as they essentially augment a decision based on a continuous metric with undefined scale (the index) into a binary test-like reject/not reject decision. Yet, their statistical and logical shortcomings lead us to conclude that decisions based on cut-offs are, at best, rough guesses that lack a statistical basis.

We use the PLS estimator because it features most prominently in the literature on composite models, however, we stress that in principal any consistent estimator may be used to estimate the composite model. Moreover the modified test statistic is likely to behave similar for any consistent and asymptotically normally distributed estimator. In fact, we used alternative estimators such as generalized structured component analysis (GSCA) or several of [Kettenring \(1971\)](#)'s algorithms and

obtained similar results. It seems reasonable to conclude that the choice of estimator is of least importance when assessing model fit. In fact, the choice of the appropriate estimator for model parameters is determined by the purpose of the model and the resulting demands on the estimators' statistical properties. If the model is used to derive inferential or causal statements, possibly involving fit assessment, estimators are expected to be unbiased or consistent for all model parameters. Ideally, the estimator is also efficient in a sense that the information (sample) based upon which estimates are derived is used most effectively thereby maximizing estimation precision. If out-of-sample prediction is at the center of the researchers interest statistical properties such as unbiasedness and consistency for a population parameter become less important, or even a nuisance, as the sole objective is to minimize a loss function such as the RMSE to maximize the out-of-sample predictive quality of the model – the end justifies the means. In fact, population parameter estimates are essentially just by-products of the loss function minimization in this case. Hence, estimates may happen to be consistent estimates for their population parameter counterpart, however, it is not a requirement as inferential or causal statements are not to be made.

---

## Chapter 3

# Measurement error correlation within blocks of indicators in consistent partial least squares: Issues and remedies

---

### 3.1 Introduction <sup>1</sup>

Structural equation modeling (SEM) is a versatile, widely used analytical techniques to statistically examine relationships between unobserved constructs. In SEM constructs are predominantly operationalized by latent variables; so-called common factors assumed to be measured by a set of observable indicators within the measurement model framework.

To estimate the measurement model parameters and the postulated structural relationship between latent variables, two conceptually different estimation approaches have been established: covariance-based (CB) estimation and

---

<sup>1</sup>This chapter is based on a peer-reviewed, published article written with Florian Schuberth and Theo Dijkstra. Compared to the published article, I rewrote parts of the introduction and updated the terminology to reflect my improved understanding since the paper was published.



variance-based (VB) estimation. CB parameter estimates are obtained by minimizing a distance measure capturing the discrepancy between the empirical indicator covariance matrix and its model-implied counterpart. Usually, maximum likelihood is used for this purpose (e.g., Jöreskog, 1978; Bollen, 1989b). VB estimators, on the other hand, use weighted linear combinations of the indicators to represent or approximate the constructs; subsequently, model parameters are estimated – e.g., by means of linear regressions – based on these linear combinations.

Among VB estimators, the partial least squares path modeling (PLS) algorithm is arguably most wide-spread. It has been used for research in numerous fields, including strategic management (e.g., Hair et al., 2012a), marketing (e.g., Hair et al., 2012b), information systems (e.g., Ringle et al., 2012), tourism research (e.g., Müller et al., 2018), and internet research (e.g., Chiang and Hsiao, 2015; Yan et al., 2017; Wu and Li, 2018). For a recent overview of the methodological research on PLS see e.g. Khan et al. (2019).

However, despite its popularity, PLS has been subject to intense debate in recent years (see e.g., Rigdon et al., 2017, for a recent stocktaking of the debate) that helped show its limitations. Most notably, the PLS algorithm is only *consistent at large* (e.g., Dijkstra, 1981; Schneeweiss, 1993), hence yielding generally inconsistent parameter estimates for common factor models. In fact, unless all measurement errors are zero in the population, linear combinations of indicators, i.e. composites, cannot generally be expected to be a perfect substitute for the underlying common factor. As a consequence, the probability limit of the estimated correlation between composites is smaller than the population correlation between their corresponding common factors. Hence, path coefficients and factor loadings based on estimated composite correlations are inconsistent estimates for their underlying latent variable counterpart (Dijkstra and Henseler, 2015a).

To correct for these shortcomings, Dijkstra and Henseler (2015a) introduced consistent partial least squares (PLSc). Like PLS, PLSc uses the PLS algorithm to obtain composites. However, instead of using the raw composite-indicator and composite-composite correlations to obtain parameter estimates, PLSc applies adequate corrections yielding consistent and asymptotically normally distributed parameter estimates for common factor models in line with Wold (1975)'s basic design (Dijkstra, 1981; Dijkstra and Henseler, 2015a,b). As one of the defining

assumptions of the basic design, uncorrelated measurement errors within and across blocks of indicators are thus necessary in theory for PLS<sub>c</sub> to maintain consistency. Practically, however, there are a number of cases in empirical research in which uncorrelatedness of measurement errors may not hold (e.g., [Gerbing and Anderson, 1984](#); [Rubio and Gillespie, 1995](#); [Chin et al., 2003](#); [Saris and Aalberts, 2003](#); [Henseler and Chin, 2010](#); [Brown, 2015](#)). Depending on the magnitude of the unobserved correlation between measurement errors, the number of indicators, and their quality, ignoring measurement error correlations leads to inconsistent structural parameter estimates and, therefore, to potentially erroneous conclusions (e.g., [Podsakoff et al., 2012](#); [Westfall et al., 2012](#); [Gu et al., 2017](#)).

Different remedies have been proposed to prevent correlated measurement errors through a careful study design (e.g., [MacKenzie and Podsakoff, 2012](#); [Podsakoff et al., 2012](#)). However, in practice, aspects such as study design, item quality, and wording are often beyond the researchers' control, essentially leaving modeling approaches as the only alternative. Several researchers therefore suggest addressing the problem indirectly, e.g., by means of bifactor models and associated hierarchical reliability indices (e.g., [McNeish, 2018](#)). Others propose explicitly specifying the measurement error correlation structure in the model (e.g., [Rubio and Gillespie, 1995](#); [Brown, 2015](#), pp. 175 – 162) - although there is some controversy as to the conceptual justification of such an approach (e.g., [Landis et al., 2009](#); [Hermida, 2015](#)).

Against this background, we contribute to the literature by extending PLS<sub>c</sub> to yield consistent parameter estimates for population models whose indicator blocks contain a subset of correlated measurement errors. Based on an idea outlined in [Dijkstra \(2013a\)](#) and mentioned in [Dijkstra and Henseler \(2015a\)](#), this is achieved by modifying the calculation of the correction factors as defined by PLS<sub>c</sub> to include a priori assumptions on the structure of the within-block measurement error correlations.

The remainder of the paper is structured as follows: Section [3.2](#) briefly reviews the PLS algorithm and its consistent version PLS<sub>c</sub>. Section [3.3](#) presents the methodological contribution to obtain consistent and asymptotically normally distributed parameter estimates if within-block measurement error correlation is present. The design and results of a Monte Carlo simulation to assess the approach are described in Sections [3.4](#) and [3.5](#). The paper closes with a discussion and an

outline for potential future research in Section 3.6.

## 3.2 The partial least squares path modeling algorithm

PLS was developed by Herman O. A. Wold for the analysis of high-dimensional data in a low-structure environment but has been extended and modified in recent years to accommodate a wide variety of analytical needs (e.g., Wold, 1975, 1982). PLS, which may be regarded as similar to generalized canonical correlation analysis, is capable of emulating several of Kettenring's (1971) techniques for the canonical correlation analysis of several sets of indicators (Tenenhaus et al., 2005).

The following section briefly reviews the notation and main aspects of the PLS algorithm, PLS<sub>c</sub>, and their underlying model setup: the basic design.

Consider a model with  $J$  latent variables  $\eta_1, \eta_2, \dots, \eta_J$  with unit variance related via a set of structural equations and the existence of corresponding vectors of indicators  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_J$ ; indicators are defined as measurement error-prone manifestations of their respective latent variable:

$$\mathbf{x}_j = \boldsymbol{\lambda}_j \eta_j + \boldsymbol{\varepsilon}_j \quad \forall j = 1, \dots, J, \quad (3.1)$$

where the vector of loadings  $\boldsymbol{\lambda}_j$  contains as many components as there are indicators in  $\mathbf{x}_j$ . All variables involved are centered at their mean, and all second-order moments are assumed to exist. The measurement errors  $\boldsymbol{\varepsilon}_j$  are assumed to satisfy  $E(\boldsymbol{\varepsilon}_j | \eta_j) = \mathbf{0}$  such that the conditional mean of  $\mathbf{x}_j$  is given by  $\boldsymbol{\lambda}_j \eta_j$ . Furthermore, measurement errors are taken as mutually uncorrelated within blocks and between blocks such that the within-block measurement error covariance matrix  $\boldsymbol{\Theta}_{jj} \equiv E(\boldsymbol{\varepsilon}_j \boldsymbol{\varepsilon}_j')$  is diagonal and the measurement error covariance matrix across blocks  $\boldsymbol{\Theta}_{ij} \equiv E(\boldsymbol{\varepsilon}_i \boldsymbol{\varepsilon}_j')$  is zero. Based on these assumptions, we have the following covariance matrices:

$$\boldsymbol{\Sigma}_{ij} \equiv E(\mathbf{x}_i \mathbf{x}_j') = \rho_{ij} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_j' \quad \text{and} \quad (3.2)$$

$$\boldsymbol{\Sigma}_{jj} \equiv E(\mathbf{x}_j \mathbf{x}_j') = \boldsymbol{\lambda}_j \boldsymbol{\lambda}_j' + \boldsymbol{\Theta}_{jj}, \quad (3.3)$$

where  $\rho_{ij}$  is the correlation between latent variables  $\eta_i$  and  $\eta_j$ . The correlation matrix ( $\rho_{ij}$ ) will generally be positive definite. It can satisfy rank constraints on sub-matrices as induced by (non-recursive) simultaneous equations for the latent variables

(Dijkstra, 1981). In this paper, we work with recursive systems only, so each equation for a latent variable is a regression equation.

### 3.2.1 Traditional partial least squares path modeling

In addition to the setup given above, assume that there are  $K_j$  column vectors of standardized indicator observations of length  $N$  denoted by  $\mathbf{x}_{1j}, \mathbf{x}_{2j}, \dots, \mathbf{x}_{K_jj}$ . For ease of notation, all  $K_j$  indicators are stacked in the  $(N \times K_j)$  matrix  $\mathbf{X}_j$ . In PLS, proxies for each latent variable are built as the weighted sum of its related indicators. The unknown weight vector  $\mathbf{w}_j$  is determined in an iterative three-step procedure.

At the outset, initial arbitrary outer weights  $\hat{\mathbf{w}}_j^{(0)}$  are chosen such that the unit variance condition  $\hat{\mathbf{w}}_j^{(0)'} \mathbf{S}_{jj} \hat{\mathbf{w}}_j^{(0)} = 1$  holds, where the  $(K_j \times K_j)$  matrix  $\mathbf{S}_{jj}$  is a consistent estimate of the population correlation matrix  $\Sigma_{jj}$ .<sup>2</sup> After initialization, the iterative algorithm begins with Step 1, the outer estimation of  $\eta_j$ :

$$\hat{\boldsymbol{\eta}}_j^{(h)} = \mathbf{X}_j \hat{\mathbf{w}}_j^{(h)} \quad \text{with} \quad \hat{\mathbf{w}}_j^{(h)'} \mathbf{S}_{jj} \hat{\mathbf{w}}_j^{(h)} = 1 \quad \forall j = 1, \dots, J, \quad (3.4)$$

where  $\hat{\boldsymbol{\eta}}_j^{(h)}$  is the  $(N \times 1)$  vector of outer estimates and  $\hat{\mathbf{w}}_j^{(h)}$  the  $(K_j \times 1)$  estimated weight vector. The superscript indicates the  $h$ -th iteration step. Since outer weights are scaled, the outer estimates are scaled as well.

Based on the outer estimates from Step 1, so-called inner estimates of latent variable  $\eta_j$  are computed according to the inner weighting scheme:

$$\tilde{\boldsymbol{\eta}}_j^{(h)} = \sum_{i=1}^J e_{ji}^{(h)} \hat{\boldsymbol{\eta}}_i^{(h)}, \quad (3.5)$$

where  $e_{ji}^{(h)} = \text{sign}(\hat{\mathbf{w}}_j^{(h)'} \mathbf{S}_{ji} \hat{\mathbf{w}}_i^{(h)})$  is the inner weight with  $\text{plim} \mathbf{S}_{ij} = \Sigma_{ij}$ .<sup>3</sup> All inner estimates  $\tilde{\boldsymbol{\eta}}_j^{(h)}$  are again scaled such that their variance is one.

In the third step of each iteration, new outer weights are calculated according to *mode A*. For *mode A*, the new estimated outer weights, also known as correlation weights,  $\hat{\mathbf{w}}_j^{(h+1)}$  are equal to the coefficients resulting from a sequence of univariate

<sup>2</sup>Throughout the iteration, the unit variance condition is maintained by using the scaling factor  $(\hat{\mathbf{w}}_j^{(h)'} \mathbf{S}_{jj} \hat{\mathbf{w}}_j^{(h)})^{-\frac{1}{2}}$  for the outer weights  $\hat{\mathbf{w}}_j^{(h)}$  in each iteration step  $h$ .

<sup>3</sup>The inner weight  $e_{ji}$  defines how the inner estimates are built. Three inner weighting schemes are common: the *centroid*, the *factorial*, and the *path weighting* scheme. For linear structural models, however, all schemes yield essentially the same results (Noonan and Wold, 1982) and therefore do not affect our proposed approach. For the purpose of our simulation, we employed the centroid scheme. For more details on the schemes, see e.g., Tenenhaus et al. (2005).

ordinary least squares (OLS) regressions of  $\mathbf{X}_j$  on  $\tilde{\boldsymbol{\eta}}_j^{(h)}$ .<sup>4</sup> As a crucial result of *mode A*, the following proportionality relation is obtained:

$$\hat{\boldsymbol{w}}_j^{(h+1)} \propto \sum_{i=1}^J e_{ij}^{(h)} \mathbf{S}_{ij} \hat{\boldsymbol{w}}_i^{(h)} \quad \text{with} \quad \hat{\boldsymbol{w}}_j^{(h+1)'} \mathbf{S}_{jj} \hat{\boldsymbol{w}}_j^{(h+1)} = 1. \quad (3.6)$$

New outer weights  $\hat{\boldsymbol{w}}_j^{(h+1)}$  are checked for notable changes compared to the outer weights from the previous iteration step  $\hat{\boldsymbol{w}}_j^{(h)}$ . If there is a significant change in the weights, the algorithm continues by building new outer proxies based on the newly obtained weights; otherwise, it stops. Assuming that the established model is correct, it can be shown that the PLS algorithms will converge with a probability tending to one as the sample size increases (Dijkstra, 1981). For smaller samples and misspecified models, however, convergence may be an issue (Henseler and Chin, 2010). The resulting weights satisfy Equation (3.6) with all superscripts removed. Moreover, their probability limits satisfy the same equations, with  $\mathbf{S}_{ij}$  replaced by  $\boldsymbol{\Sigma}_{ij}$ . Thus, the probability limits of the weights obtained by PLS and PLS<sub>c</sub> can be obtained by applying them to the population indicator covariance matrix  $\boldsymbol{\Sigma}$ . Notably, the proof of numerical and probabilistic convergence does not require that the measurement errors within blocks are uncorrelated. To see this, it is crucial to note, that the population weights are unaffected of the precise nature of  $\boldsymbol{\Sigma}_{jj}$ . Using the final weights  $\hat{\boldsymbol{w}}_j$  and taking probability limits on both sides of Equation (3.6), we have:

$$\text{plim } \hat{\boldsymbol{w}}_j \propto \text{plim } \sum_{i=1}^J e_{ij} \mathbf{S}_{ij} \hat{\boldsymbol{w}}_i \longrightarrow \boldsymbol{w}_j \propto \sum_{i=1}^J e_{ij} \boldsymbol{\Sigma}_{ij} \boldsymbol{w}_i = \sum_{i=1}^J e_{ij} \rho_{ij} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i' \boldsymbol{w}_i, \quad (3.7)$$

where the last equality crucially assumes uncorrelated measurement errors *across* blocks, i.e.,  $\boldsymbol{\Theta}_{ij} = \mathbf{0}$ , but not within blocks of indicators.<sup>5</sup>

Once convergence is reached, the resulting stable outer weights  $\hat{\boldsymbol{w}}_j$  are used to build the final proxy for the latent variables:  $\hat{\boldsymbol{\eta}}_j = \mathbf{X}_j \hat{\boldsymbol{w}}_j$ . Finally, factor loadings for each block are obtained as the OLS solution of a sequence of regressions of  $\mathbf{X}_j$  on  $\hat{\boldsymbol{\eta}}_j$ .

Similarly, the path coefficients are the OLS estimates of the equations postulated by the structural model.

---

<sup>4</sup>Only correlation weights are considered, as these were originally used by Dijkstra and Henseler (2015a) to obtain consistent parameter estimates. However, consistent parameter estimates can be also obtained from the weights calculated by *mode B* or *mode C* (Dijkstra, 1981, Chap.2 par. 5.2). Moreover, weights obtained by *mode A* are generally more stable, since those from *mode B* (regression weights) tend to suffer from multicollinearity. For an overview of outer weighting schemes and their properties, see Dijkstra (1981).

<sup>5</sup>In fact, Equation (3.7) is not tied to using “converged” weights such as those obtained by PLS. Dijkstra and Schermelleh-Engel (2014) for example discuss what they call “one-step” weight (essentially weight obtained after one iteration). In theory, any weight vector obtained after an arbitrary number of iterations (converged or not) will satisfy Equation (3.7).

### 3.2.2 Consistent partial least squares

The principal idea of PLS is to build proxies as stand-ins for the latent variables and subsequently estimate model parameters based on these proxies. Naturally, it cannot be expected that these stand-ins perfectly reflect the underlying latent variables unless all measurement errors are assumed to be zero in the population. As a consequence, the probability limit of the estimated correlation between proxies is smaller in absolute value than the population correlation between their corresponding common factors. Hence, path coefficients and factor loadings based on estimated proxy correlations are inconsistent estimates for their population counterpart. PLSc addresses this shortcoming by consistently estimating the composite reliability and subsequently correcting the correlations among the proxies for attenuation (Cohen et al., 2003). Provided that each latent variable is connected to at least two indicators, the population composite reliability of the population proxy  $\bar{\eta}_j$  as defined in Dijkstra and Henseler (2015b) is given by:

$$\rho_{A,j} := (\mathbf{w}'_j \mathbf{w}_j)^2 \cdot c_j^2, \quad (3.8)$$

where  $c_j := \sqrt{\boldsymbol{\lambda}'_j \boldsymbol{\Sigma}_{jj} \boldsymbol{\lambda}_j}$  is the factor that relates population weights  $\mathbf{w}_j = \text{plim } \hat{\mathbf{w}}_j$  to their corresponding population loadings  $\boldsymbol{\lambda}_j$  (Dijkstra, 1981; Dijkstra and Henseler, 2015a):

$$\mathbf{w}_j = \frac{\boldsymbol{\lambda}_j}{\sqrt{\boldsymbol{\lambda}'_j \boldsymbol{\Sigma}_{jj} \boldsymbol{\lambda}_j}}. \quad (3.9)$$

It is crucial to note, that this relationship holds independent of the form of  $\boldsymbol{\Sigma}_{jj}$ . To see this, note that based on Equation (3.7), the population relation between weights and loadings may simply be written as  $\mathbf{w}_j = c_j^{-1} \boldsymbol{\lambda}_j$  since  $\sum_{i=1}^J e_{ij} \rho_{ij} \boldsymbol{\lambda}'_i \mathbf{w}_i$  is a scalar. Using the population normalization condition  $\mathbf{w}'_j \boldsymbol{\Sigma}_{jj} \mathbf{w}_j = 1$  now yields the population value  $c_j$ :

$$\mathbf{w}'_j \boldsymbol{\Sigma}_{jj} \mathbf{w}_j = 1 \quad (3.10)$$

$$\frac{\boldsymbol{\lambda}'_j \boldsymbol{\Sigma}_{jj} \boldsymbol{\lambda}_j}{c_j^2} = 1 \quad (3.11)$$

$$c_j^2 = \boldsymbol{\lambda}'_j \boldsymbol{\Sigma}_{jj} \boldsymbol{\lambda}_j. \quad (3.12)$$

Consequently, population weights and the proportionality constant  $c_j$  clearly vary with  $\boldsymbol{\Sigma}_{jj}$ , however the fundamental relationship given by Equation (3.7) is unaffected by  $\boldsymbol{\Sigma}_{jj}$  (and therefore also unaffected by potential within-block error correlation).

To obtain the estimated correction factor  $\hat{c}_j$ , a variety of approaches are possible (Dijkstra, 2013a). Usually,  $\hat{c}_j$  is chosen for block  $j$  such that the squared Euclidean distance between the off-diagonal elements of the empirical covariance matrix  $\mathbf{S}_{jj}$  and the matrix  $(c_j \hat{\mathbf{w}}_j)(c_j \hat{\mathbf{w}}_j)'$  is minimized. In this case, the squared estimated correction factor is given by:

$$\hat{c}_j^2 = \frac{\hat{\mathbf{w}}_j'(\mathbf{S}_{jj} - \text{diag}(\mathbf{S}_{jj}))\hat{\mathbf{w}}_j}{\hat{\mathbf{w}}_j'(\hat{\mathbf{w}}_j\hat{\mathbf{w}}_j' - \text{diag}(\hat{\mathbf{w}}_j\hat{\mathbf{w}}_j'))\hat{\mathbf{w}}_j}. \quad (3.13)$$

Since  $\text{plim } \hat{\mathbf{w}}_j = \mathbf{w}_j$  and  $\text{plim } \mathbf{S}_{jj} = \Sigma_{jj}$  and since the functions involved are continuous, the probability limit directly follows:

$$\text{plim } \hat{c}_j^2 = \frac{\mathbf{w}_j'(\Sigma_{jj} - \text{diag}(\Sigma_{jj}))\mathbf{w}_j}{\mathbf{w}_j'(\mathbf{w}_j\mathbf{w}_j' - \text{diag}(\mathbf{w}_j\mathbf{w}_j'))\mathbf{w}_j} \quad (3.14)$$

$$= \boldsymbol{\lambda}_j' \Sigma_{jj} \boldsymbol{\lambda}_j + \boldsymbol{\lambda}_j' \Sigma_{jj} \boldsymbol{\lambda}_j \cdot \frac{\boldsymbol{\lambda}_j'(\Theta_{jj} - \text{diag}(\Theta_{jj}))\boldsymbol{\lambda}_j}{\boldsymbol{\lambda}_j'(\boldsymbol{\lambda}_j\boldsymbol{\lambda}_j' - \text{diag}(\boldsymbol{\lambda}_j\boldsymbol{\lambda}_j'))\boldsymbol{\lambda}_j}. \quad (3.15)$$

The numerator of the last term in Equation (3.15) is zero when all the measurement errors are uncorrelated in the population since, in this case,  $\Theta_{jj} = \text{diag}(\Theta_{jj})$ .

Assuming that  $\Theta_{jj}$  is indeed a diagonal matrix, the resulting probability limit of the squared estimated correction factor equals the squared correction factor from Equation (3.12), i.e., the squared distortion of the population weights to population loadings. Hence, consistent factor loading estimates and attenuation-corrected correlations between common factors  $j$  and  $i$  are readily given by

$$\hat{\boldsymbol{\lambda}}_j = \hat{c}_j \hat{\mathbf{w}}_j \quad \text{and} \quad \widehat{\text{Cor}(\eta_j, \eta_i)} = \frac{\hat{\mathbf{w}}_j' \mathbf{S}_{ji} \hat{\mathbf{w}}_i}{\sqrt{\hat{\rho}_{A,j} \cdot \hat{\rho}_{A,i}}}. \quad (3.16)$$

Depending on the underlying structural model, consistent path coefficient estimates may be obtained by OLS or two-staged least squares using the estimated disattenuated correlation given above.

### 3.3 Correlated measurement errors

As suggested by Equation (3.15), the consistency of original PLSc was established based on the assumptions of the basic design, including measurement errors that are uncorrelated across and within blocks of indicators; i.e.,  $\Theta_{jj}$  is indeed a diagonal matrix. In fact, if measurement errors in the population are correlated within blocks of indicators, then original PLSc using the correction factor from Equation (3.13) leads to inconsistent parameter estimates for both factor loadings and path

coefficients, where the magnitude of the inconsistency is positively related to the strength of the measurement error correlation and negatively affected by the composite reliability. However, taking into account measurement errors is straightforward provided that the correlation is confined to be within the indicator blocks.

Given a presumption on the measurement error correlation structure, define the set of uncorrelated measurement error pairs as  $U_j := \{(k, m) | \theta_{km;jj} = 0\}$ , where  $\theta_{km;jj}$  denotes the population covariance between the  $k$ -th and  $m$ -th measurement error of block  $j$ . An immediate extension to original PLS is to minimize the squared Euclidean distance between the off-diagonal elements of the empirical covariance matrix  $\mathbf{S}_{jj}$  and the matrix  $(c_j \hat{\mathbf{w}}_j)(c_j \hat{\mathbf{w}}_j)'$  with respect to  $c_j$ , including only those elements contained in the set  $U_j$ :

$$\hat{c}_j^2 = \operatorname{argmin}_{c_j^2} \sum_{k,m \in U_j} [s_{km;jj} - c_j^2 \hat{w}_{kj} \hat{w}_{mj}]^2, \quad (3.17)$$

where  $\hat{w}_{kj}$  and  $\hat{w}_{mj}$  are the  $k$ -th and  $m$ -th elements of the weight vector  $\hat{\mathbf{w}}_j$  and  $s_{km;jj}$  is the empirical covariance between the  $k$ -th and  $m$ -th indicators of block  $j$ .<sup>6</sup>

Provided that the set of uncorrelated measurement error pairs is nonempty, minimization yields:

$$\hat{c}_j^{*2} = \frac{\sum_{k,m \in U_j} \hat{w}_{kj} \hat{w}_{mj} s_{km;jj}}{\sum_{k,m \in U_j} \hat{w}_{kj}^2 \hat{w}_{mj}^2}. \quad (3.18)$$

Because of the continuity of the functions involved, the consistency of the sample moments, and the fact that the probability limits of the PLS weight vectors, as given in [Dijkstra \(1981\)](#), are effectively independent of the assumed structure within the blocks, the probability limit of the estimated adjusted correction factor is again equal to  $\boldsymbol{\lambda}'_j \boldsymbol{\Sigma}_{jj} \boldsymbol{\lambda}_j$ . Indeed, replacing the terms in Equation (3.18) by their population

---

<sup>6</sup>The extension suggested here is not necessarily tied to using the Euclidean distance. As pointed out by [Dijkstra \(2013a\)](#), weights could be introduced in Equation (3.17) to potentially reap efficiency gains. More generally, functions of ratios may be minimized; however, the solution will require iterative procedures. In this paper, the simplest approach was chosen to keep the main focus on our enhancement.



counterparts yields:

$$\text{plim } \hat{c}_j^{*2} = \frac{\sum_{k,m \in U_j} w_{k,j} w_{m,j} \sigma_{km;jj}}{\sum_{k,m \in U_j} w_{k,j}^2 w_{m,j}^2} \quad (3.19)$$

$$= \lambda_j' \Sigma_{jj} \lambda_j \cdot \frac{\sum_{k,m \in U_j} \lambda_{k,j}^2 \lambda_{m,j}^2 + \sum_{k,m \in U_j} \lambda_{k,j} \lambda_{m,j} \theta_{km;jj}}{\sum_{k,m \in U_j} \lambda_{k,j}^2 \lambda_{m,j}^2} \quad (3.20)$$

$$= \lambda_j' \Sigma_{jj} \lambda_j. \quad (3.21)$$

where the last term in Equation (3.20) is one since  $\theta_{km;jj}$  is zero by assumption for all elements contained in  $U_j$ . As a consequence, consistent estimates for the attenuation-corrected correlations between common factors, loadings, and path coefficients may be obtained along the same lines described in the preceding section.

### 3.4 Monte Carlo simulation

To assess the efficacy of the modification, a Monte Carlo simulation is conducted.

To this end, six population models are investigated.<sup>7</sup> The baseline population model to be considered is illustrated in Figure 3.1.

---

<sup>7</sup>To draw a comprehensive picture of each modeling decision's influence on the results, we examined numerous alternative setups where we varied, for instance, the number of indicators, the number of observations, the indicator block whose errors were correlated and the magnitude of different loadings. Additionally, as a robustness check, we conducted the simulation using non-normally distributed data as in [Dijkstra and Henseler \(2015a\)](#) and applied all of the alternative approaches to obtain the correction factor described in [Dijkstra \(2013a\)](#). Here, we describe only those setups that we deem most informative and most general, but note that none of the results of any other specifications were contrary to the central findings of the paper at hand. The results for the alternative specifications or the necessary R-files to reproduce these can be obtained from the authors upon request.

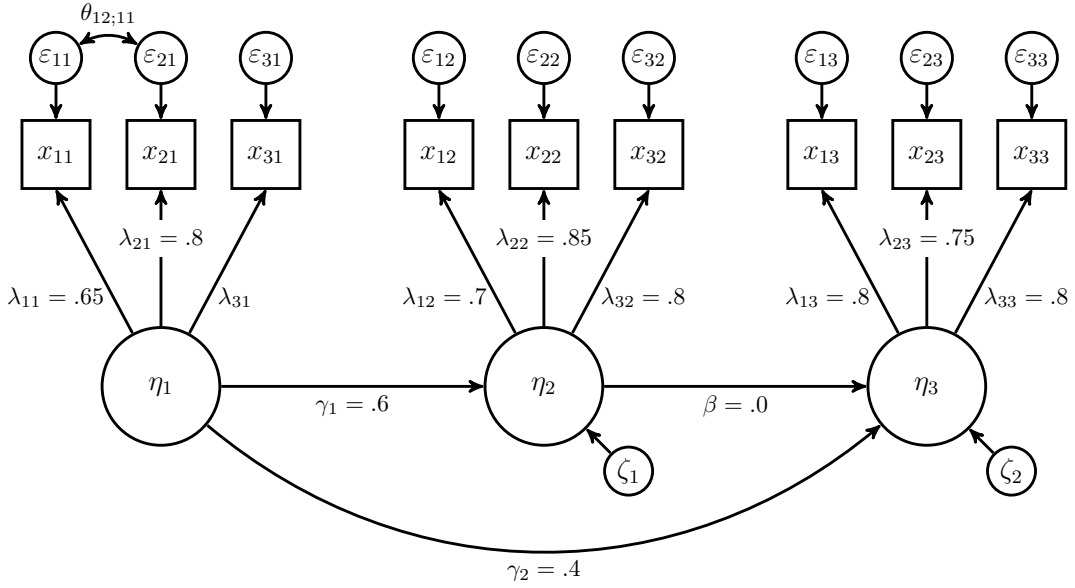


Figure 3.1: Baseline population model

The structural population model contains three latent variables:

$$\eta_2 = \gamma_1 \eta_1 + \zeta_1 \quad (3.22)$$

$$\eta_3 = \gamma_2 \eta_1 + \beta \eta_2 + \zeta_2, \quad (3.23)$$

where  $\gamma_1 = 0.6$ ,  $\gamma_2 = 0.4$ ,  $\beta = 0$ ,  $\text{Var}(\zeta_1) = 0.64$ ,  $\text{Var}(\zeta_2) = 0.84$ , and

$\text{Cov}(\eta_1 \cdot \zeta_1) = \text{Cov}(\eta_1 \cdot \zeta_2) = \text{Cov}(\eta_2 \cdot \zeta_2) = \text{Cov}(\zeta_1, \zeta_2) = 0$ . The structural model

remains identical across all six population models and is similar to structural models typically applied in the literature (e.g., Paxton et al., 2001; Hwang et al., 2010).

For each population model, the exogenous latent variable  $\eta_1$  and the two endogenous latent variables  $\eta_2$  and  $\eta_3$  are each connected to three indicators, the minimum requirement for our approach to be feasible since the additional indicator ensures that  $U_j \neq \emptyset$  if a correlation between any two measurement errors is allowed. Loadings for  $\eta_2$  and  $\eta_3$  are fixed at  $\lambda_{12} = 0.7$ ,  $\lambda_{22} = 0.85$ ,  $\lambda_{32} = 0.8$  and  $\lambda_{13} = 0.8$ ,  $\lambda_{23} = 0.75$ ,  $\lambda_{33} = 0.8$ , reflecting average indicator reliabilities. Furthermore, the first two loadings of  $\eta_1$  are set to  $\lambda_{11} = 0.65$  and  $\lambda_{21} = 0.8$ , respectively. To investigate how different composite reliabilities affect parameter estimates, both the number of indicators per block and the size of the loadings may be varied. Here, we chose the latter by varying  $\lambda_{31}$  within a range of 0.5 to 0.9 in steps of 0.2.

All measurement errors ( $\epsilon_{kj}$ ) have a mean of zero and are uncorrelated across and within blocks except for the first and the second measurement errors of the first indicator block:  $\theta_{12;11} = \sqrt{0.360 \cdot 0.578} \cdot \rho_{12;11}$ , where  $\rho_{12;11}$  denotes the correlation

between  $\varepsilon_{11}$  and  $\varepsilon_{21}$ . To assess how the strength of the correlation affects parameter estimates, we include a case with comparatively low ( $\rho_{12;11} = 0.1$ ) and high correlation ( $\rho_{12;11} = 0.6$ ).

The simulation is conducted in the statistical software environment R (R Core Team, 2017). The datasets for each of the six resulting population models (= 3 different loading magnitudes  $\times$  2 different measurement error correlations) are drawn according to the following baseline population indicator correlation matrix using the *MASS* package (Venables and Ripley, 2002). Samples of size  $N = 100, 200$ , and 1000 are drawn from a multivariate normal distribution with the mean of each indicator set to zero and the covariance matrix displayed in Equation (3.24).

$$\Sigma = \begin{pmatrix} \underline{x_{11}} & \underline{x_{21}} & \underline{x_{31}} & \underline{x_{12}} & \underline{x_{22}} & \underline{x_{32}} & \underline{x_{13}} & \underline{x_{23}} & \underline{x_{33}} \\ 1.00 & 0.52 + \theta_{12;11} & 0.65 \cdot \lambda_{31} & 0.27 & 0.33 & 0.31 & 0.21 & 0.20 & 0.21 \\ & 1.00 & 0.80 \cdot \lambda_{31} & 0.34 & 0.41 & 0.38 & 0.26 & 0.24 & 0.26 \\ & & 1.00 & 0.21 & 0.26 & 0.24 & 0.16 & 0.15 & 0.16 \\ & & & 1.00 & 0.60 & 0.56 & 0.13 & 0.13 & 0.13 \\ & & & & 1.00 & 0.68 & 0.16 & 0.15 & 0.16 \\ & & & & & 1.00 & 0.15 & 0.14 & 0.15 \\ & & & & & & 1.00 & 0.60 & 0.64 \\ & & & & & & & 1.00 & 0.60 \\ & & & & & & & & 1.00 \end{pmatrix}. \quad (3.24)$$

The number of replications per population model is set to 1000, resulting in a total of 18,000 datasets (6 population models  $\times$  3 sample sizes  $\times$  1,000 replications).

To estimate the underlying population parameters for each data set, two models were specified. The first model  $M_1$  correctly reflects the corresponding underlying population model in terms of the structural and the measurement model but does not explicitly account for the correlation between the measurement errors  $\varepsilon_{11}$  and  $\varepsilon_{21}$ .

Here, estimation by traditional PLS is expected to yield estimates that systematically deviate from their corresponding population values. The second model  $M_2$  is similar to the first model but acknowledges the measurement error correlation as present in the population models. Estimation is performed using our contributed modification. To this end, we use the `MoMpoly` function provided by the *MoMpoly* package (Schuberth et al., 2017), which implements the procedure as described in this

paper.<sup>8</sup> Here, the enhanced procedure is expected to yield estimates close to the corresponding population parameters. However, this is likely to come at the cost of a loss in precision, as the calculation of the correction factor is based on less information. In addition to the estimations based on the simulated datasets, we retrieve the parameters for each population model using the population covariance matrix as input. This serves to verify Fisher consistency, i.e., whether a given estimator is in fact able to yield population parameters if supplied by the population covariance matrix.

To compare the estimates across the different designs, two common quality measures are considered: the estimated bias and the root mean squared error (RMSE). The bias is estimated as

$$\widehat{\text{Bias}} = \frac{1}{M} \sum_{i=1}^M (\hat{\psi}_i - \psi), \quad (3.25)$$

where  $\psi$  denotes a generic population parameter and  $\hat{\psi}$  is its corresponding estimate for a given model and sample size. The number of elements  $M$  is equal to the number of replications corrected for the number of Heywood cases and outliers.<sup>9</sup> The latter is defined as all estimates larger than the median  $+/- 3$  times the interquartile range. Consistency of our modification is essentially achieved by discarding information. Hence, finite sample comparisons between modified PLS<sub>c</sub> and original PLS<sub>c</sub> should take the expected trade-off between bias and variability into account. A well-established measure in this respect is the (estimated) RMSE given by:

$$\widehat{\text{RMSE}} = \sqrt{\frac{1}{M} \sum_{i=1}^M (\hat{\psi}_i - \psi)^2}. \quad (3.26)$$

The population RMSE essentially combines standard deviation and bias. For an unbiased estimator, it equals to the standard deviation.

### 3.5 Results

Below, we present the results of the simulation study. We report the results for the path coefficients  $\gamma_1$ ,  $\gamma_2$ , and  $\beta$  and the factor loadings  $\lambda_{21}$  and  $\lambda_{31}$  of the indicator

---

<sup>8</sup>The *MoMpoly* package is currently not on the Comprehensive R Archive Network. To replicate the results, a development version is available upon request.

<sup>9</sup>Heywood cases in PLS<sub>c</sub> may occur for three reasons: (i) the attenuation-corrected or uncorrected estimated covariance matrix between proxies is not semi-positive definite, (ii) standardized absolute loading estimates are larger than one, and (iii) the PLS algorithm has not converged.

block affected by measurement error correlation. In addition, the share of Heywood cases and the share of outliers are given for each setup. Omission of the other loadings is justified because the results for  $\lambda_{11}$  are virtually identical to those of  $\lambda_{21}$  and  $\lambda_{31}$ , while the loadings of those indicator blocks whose measurement errors are assumed to be uncorrelated are by construction unaffected by the correlated measurement errors of other blocks within the structural model.

Tables 3.1 and 3.2 summarize the results. Each major column contains the results for a given population  $\lambda_{31}$  (i.e., 0.5, 0.7, 0.9) spread across two minor columns representing the varying population measurement error correlation  $\rho_{12;11}$ , where  $\rho_{12;11} \in \{0.1, 0.6\}$ . Each major-minor combination is again split by model (i.e., model  $M_1$  and model  $M_2$ ) to facilitate the comparison.

Table 3.1 displays the simulation results with respect to the estimated bias. Each row displays the average deviation of the estimated parameters from their corresponding population values split by sample size  $N = 100, 200, 1000$  (across rows), population, and estimated model (across columns). In the presence of unmodeled measurement error correlation within a block of indicators, parameter estimates obtained by PLS<sub>c</sub> using the traditional correction factor (model  $M_1$ ) systematically deviate on average from their pre-specified population value, where the deviation per population model and parameter is stable across sample sizes. This finding is in line with the fact that original PLS<sub>c</sub> is indeed unable to retrieve population parameters when supplied with the corresponding population indicator covariance matrix, as displayed at the bottom of Table 3.1. Comparing results for a given sample size, the magnitude of the deviations varies between virtually no bias (e.g., for  $\lambda_{31} = 0.9$  and  $\rho_{12;11} = 0.1$ ) and values of up to 0.1 (e.g., for  $\lambda_{31} = 0.5$  and  $\rho_{12;11} = 0.6$ ), depending on the strength of the measurement error correlation  $\rho_{12;11}$  and the size of the population loading  $\lambda_{31}$ . In this respect, the effect of the strength of the correlation between measurement errors on the estimated bias is most pronounced with higher error correlations leading to increased deviation.

Looking across columns for a given measurement error correlation, deviations vary only marginally, although an increasing reliability – as induced by the higher loadings – slightly decreases bias overall. These findings are again supported by the parameters obtained based on the corresponding population covariance matrix shown in the last four rows of Table 3.1: deviations for all parameters are lowest for estimates based on

population models with a higher composite reliability, i.e.,  $\lambda_{31} = 0.9$ .

Table 3.1: Estimated bias

		$\lambda_{31} = 0.5$				$\lambda_{31} = 0.7$				$\lambda_{31} = 0.9$			
		$\rho_{12;11} = 0.1$		$\rho_{12;11} = 0.6$		$\rho_{12;11} = 0.1$		$\rho_{12;11} = 0.6$		$\rho_{12;11} = 0.1$		$\rho_{12;11} = 0.6$	
$N$	Parameter	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)
100	$\gamma_1$	0.00	0.01	-0.06	0.01	0.00	0.01	-0.04	0.00	0.00	0.01	-0.01	0.01
100	$\gamma_2$	0.02	0.05	-0.06	0.04	0.02	0.02	-0.04	0.00	0.00	0.01	-0.02	0.02
100	$\beta$	0.02	0.00	0.08	0.00	0.02	0.02	0.07	0.03	0.04	0.01	0.06	0.02
100	$\lambda_{21}$	0.01	-0.01	0.08	-0.02	0.01	0.00	0.06	0.00	0.01	0.00	0.06	0.01
100	$\lambda_{31}$	0.01	0.01	0.06	0.01	0.01	0.00	0.06	0.00	-0.01	-0.01	0.02	-0.01
Heywood cases (%)		28.70	34.00	33.30	36.30	27.00	31.20	30.60	27.60	30.50	27.50	46.10	32.80
Outliers (%)		1.96	3.64	3.75	5.81	4.11	2.47	2.59	1.93	2.45	3.45	2.97	2.53
200	$\gamma_1$	-0.01	0.01	-0.06	0.01	-0.01	0.00	-0.04	0.01	-0.01	0.00	-0.03	0.00
200	$\gamma_2$	-0.01	0.02	-0.08	0.02	-0.01	0.00	-0.05	0.02	-0.01	0.00	-0.03	0.00
200	$\beta$	0.02	0.00	0.08	-0.01	0.02	0.01	0.06	-0.01	0.01	0.01	0.04	0.01
200	$\lambda_{21}$	0.02	-0.01	0.09	0.00	0.01	0.00	0.07	0.00	0.01	0.00	0.06	0.00
200	$\lambda_{31}$	0.01	0.00	0.06	0.00	0.01	0.00	0.06	0.00	0.00	0.00	0.03	0.00
Heywood cases (%)		9.00	13.70	14.20	13.20	10.40	8.30	10.10	10.70	12.50	8.00	25.90	10.80
Outliers (%)		0.88	0.81	0.93	0.92	0.67	1.31	0.67	0.78	1.14	0.65	0.27	0.56
1000	$\gamma_1$	-0.01	0.00	-0.07	0.00	-0.01	0.00	-0.05	0.00	-0.01	0.00	-0.03	0.00
1000	$\gamma_2$	-0.02	0.00	-0.08	0.00	-0.01	0.00	-0.06	0.00	-0.01	0.00	-0.05	0.00
1000	$\beta$	0.02	0.00	0.07	0.00	0.01	0.00	0.05	0.01	0.01	0.00	0.04	0.00
1000	$\lambda_{21}$	0.02	0.00	0.10	0.00	0.01	0.00	0.07	0.00	0.01	0.00	0.05	0.00
1000	$\lambda_{31}$	0.01	0.00	0.06	0.00	0.01	0.00	0.06	0.00	0.01	0.00	0.05	0.00
Heywood cases (%)		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.50	0.10
Outliers (%)		0.30	0.10	0.10	0.30	0.00	0.10	0.40	0.00	0.00	0.00	0.10	0.00
Pop.	$\gamma_1$	-0.01	0.00	-0.07	0.00	-0.01	0.00	-0.05	0.00	-0.01	0.00	-0.03	0.00
Pop.	$\gamma_2$	-0.02	0.00	-0.08	0.00	-0.01	0.00	-0.06	0.00	-0.01	0.00	-0.05	0.00
Pop.	$\beta$	0.02	0.00	0.07	0.00	0.01	0.00	0.05	0.00	0.01	0.00	0.04	0.00
Pop.	$\lambda_{21}$	0.02	0.00	0.10	0.00	0.01	0.00	0.07	0.00	0.01	0.00	0.05	0.00
Pop.	$\lambda_{31}$	0.01	0.00	0.06	0.00	0.01	0.00	0.06	0.00	0.01	0.00	0.05	0.00

Table 3.2: Root mean squared error (RMSE)

		$\lambda_{31} = 0.5$						$\lambda_{31} = 0.7$						$\lambda_{31} = 0.9$					
		$\rho_{12;11} = 0.1$		$\rho_{12;11} = 0.6$		$\rho_{12;11} = 0.1$		$\rho_{12;11} = 0.6$		$\rho_{12;11} = 0.1$		$\rho_{12;11} = 0.6$		$\rho_{12;11} = 0.1$		$\rho_{12;11} = 0.6$			
$N$	Parameter	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)	M1 (-)	M2 (+)		
100	$\gamma_1$	0.09	0.10	0.11	0.11	0.09	0.10	0.10	0.10	0.10	0.10	0.08	0.08	0.08	0.08	0.08	0.09		
100	$\gamma_2$	0.16	0.20	0.16	0.22	0.17	0.17	0.17	0.17	0.15	0.15	0.15	0.14	0.14	0.14	0.14	0.15		
100	$\beta$	0.17	0.19	0.17	0.20	0.17	0.17	0.18	0.17	0.17	0.17	0.16	0.15	0.15	0.16	0.16	0.17		
100	$\lambda_{21}$	0.09	0.10	0.11	0.10	0.08	0.09	0.09	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.07		
100	$\lambda_{31}$	0.12	0.09	0.14	0.10	0.10	0.10	0.08	0.11	0.09	0.06	0.06	0.05	0.06	0.06	0.06	0.06		
200	$\gamma_1$	0.07	0.08	0.09	0.08	0.07	0.07	0.07	0.08	0.08	0.07	0.06	0.06	0.06	0.06	0.06	0.06		
200	$\gamma_2$	0.12	0.15	0.13	0.15	0.11	0.11	0.11	0.12	0.12	0.13	0.11	0.11	0.10	0.11	0.10	0.11		
200	$\beta$	0.12	0.14	0.13	0.14	0.11	0.11	0.11	0.12	0.12	0.12	0.11	0.11	0.11	0.11	0.11	0.11		
200	$\lambda_{21}$	0.07	0.08	0.11	0.08	0.06	0.07	0.07	0.08	0.08	0.06	0.05	0.05	0.07	0.05	0.07	0.05		
200	$\lambda_{31}$	0.08	0.07	0.11	0.07	0.07	0.06	0.06	0.09	0.09	0.06	0.04	0.04	0.05	0.04	0.05	0.04		
1000	$\gamma_1$	0.03	0.04	0.07	0.04	0.03	0.03	0.03	0.06	0.06	0.03	0.03	0.03	0.04	0.03	0.04	0.03		
1000	$\gamma_2$	0.05	0.06	0.10	0.06	0.05	0.05	0.05	0.08	0.08	0.05	0.05	0.05	0.06	0.05	0.06	0.05		
1000	$\beta$	0.05	0.06	0.08	0.06	0.05	0.05	0.05	0.07	0.07	0.05	0.05	0.05	0.06	0.05	0.06	0.05		
1000	$\lambda_{21}$	0.03	0.04	0.10	0.04	0.03	0.03	0.03	0.07	0.07	0.03	0.02	0.02	0.05	0.02	0.05	0.02		
1000	$\lambda_{31}$	0.04	0.03	0.07	0.03	0.03	0.03	0.03	0.07	0.07	0.03	0.02	0.02	0.06	0.02	0.06	0.02		



In contrast to Model  $M_1$ , population model parameters are retrieved when errors are taken into account along the lines described in Section 3.3 (model  $M_2$ ). The finite sample results for model  $M_2$  are largely in line with these findings, although small deviations are found; e.g., with values of 0.04 and 0.05, the estimated bias for path coefficient  $\gamma_2$  is comparatively high.

For a given parameter, the sign of the deviations is relatively stable across sample sizes, population model, and estimated model. The results show a small but almost consistently negative deviation for  $\gamma_1$  and  $\gamma_2$ , while  $\beta$ , the path coefficient connecting the two endogenous latent variables  $\eta_2$  and  $\eta_3$ , as well as the loadings  $\lambda_{21}$  and  $\lambda_{31}$  are uniformly overestimated.

Overall, the difference between  $M_1$  and  $M_2$  is most pronounced for the estimated loadings, while deviations for the path coefficients are generally small, with modified PLS<sub>c</sub> outperforming original PLS<sub>c</sub> for large sample sizes and strong measurement error correlation only.

Table 3.2 reports the results for the RMSE. Here, the picture is mixed. For medium ( $\lambda_{31} = 0.7$ ) and high ( $\lambda_{31} = 0.9$ ) composite reliability, the RMSE for both loading and path coefficient estimates is virtually identical for  $M_1$  and  $M_2$ . In contrast to the results in Table 3.1, the RMSE does not differ systematically with the magnitude of the error correlation. For  $\lambda_{31} = 0.5$ , however, original PLS<sub>c</sub> is superior to the modified approach in small samples ( $N = 100, 200$ ). Only for a large sample size and a high composite reliability does  $M_2$  produce strictly smaller RMSEs compared to the values produced by  $M_1$ .

Regarding Heywood cases and outliers, no significant difference between  $M_1$  and  $M_2$  is visible. While the number of Heywood cases is close to zero or is zero for large samples, roughly 300 of the 1000 replications were discarded for a sample size of  $N = 100$ . In each instance, Heywood cases occur because of the loading estimates that are larger than one in absolute value.

## 3.6 Discussion and future research

Correlated measurement errors are a common feature in SEM. However, research regarding issues and potential remedies related to measurement error correlations in the context of VB estimation is scarce. While prior research papers (e.g., Charles,

2005; Zimmerman, 2007; Padilla and Veprinsky, 2012; Raykov et al., 2014) have discussed and addressed the issue of correlated measurement errors in the common factor framework, none of these are based on a VB approach like PLS. Against this background, we contribute to the ongoing development and assessment of VB estimation approaches by filling two gaps in the literature.

First, this study enhanced PLS<sub>c</sub> to yield consistent parameter estimates for population models whose indicator blocks contain a subset of correlated measurement errors – provided that all correlated errors are accounted for in the estimated model. Since PLS and PLS<sub>c</sub> are viable options for estimating interactions and other nonlinear relationships between constructs (e.g., Dijkstra and Henseler, 2011; Dijkstra and Schermelleh-Engel, 2014), our findings may help in advancing current approaches in this field. Notable examples of this kind would be the product-indicator approach (Chin et al., 2003) and the orthogonalizing approach (Henseler and Chin, 2010) – both of which rely on indicators whose errors can safely be assumed to be correlated for technical reasons. The proposed correction can help to make these two approaches consistent.

Second, initial evidence on the implications of neglecting measurement error correlation in PLS<sub>c</sub> was provided. To this end, a Monte Carlo simulation was conducted to investigate the average difference between estimated parameters and their respective population counterpart as well as the RMSE across a range of pre-specified population models for original and modified PLS<sub>c</sub>.

For original PLS<sub>c</sub>, the simulation results showed a generally small yet persistent average deviation between the estimated parameters and their corresponding population value (estimated bias) across all population models if measurement error correlation was neglected in the estimated model (model  $M_1$ ). For our proposed approach (model  $M_2$ ), the average deviation between the estimated parameters and their corresponding population value was virtually zero across all samples sizes, indicating that the procedure works well in finite samples. These findings were in line with theoretical considerations regarding the inconsistency of original PLS<sub>c</sub> when measurement errors within indicator blocks are ignored. Overall, however, differences were generally rather small. In particular, when efficiency is considered with respect to the RMSE,  $M_1$  and  $M_2$  produce virtually identical results unless both the sample size and the population error correlation are high and the population composite

reliability is low.

Regarding the magnitude of the estimated bias, we found a positive relation with the strength of the measurement error correlation, while higher composite reliability can be seen as a catalyst that essentially mitigates the effect of a given neglected measurement error correlation. The latter is intuitively appealing since an increase in composite reliability implies a decrease in attenuation of the latent variable correlation. Hence, correction for attenuation and, by the same token, any inconsistency caused by unmodeled measurement error correlation becomes less and less influential. Regarding the RMSE, the relation is less clear, although the RMSE for both the modified approach and original PLS<sub>c</sub> is higher when the population measurement error is comparatively high.

These findings are regarded as initial evidence that – although our approach is theoretically superior – original PLS<sub>c</sub> is comparatively robust with respect to misspecification of the structure of the measurement error correlations within blocks of indicators. Indeed, some preliminary simulation results by the authors confirm that PLS<sub>c</sub> outperforms common CB estimators (including maximum likelihood) in terms of bias if measurement error correlation within blocks of indicators is neglected. However, a generalization of these findings requires separate attention.

The observed tendency of PLS<sub>c</sub> to produce Heywood cases (loadings larger than one in absolute value), or incorrect signs of regression coefficients in PLS, should be addressed. We chose the simplest method to demonstrate our modification, but more robust approaches for estimating the correction factor may be applied. In fact, initial Monte Carlo evidence confirms that using, e.g., Equation (11) of [Dijkstra \(2013a\)](#), does indeed improve the share of admissible results by roughly 10 percentage points without affecting any of the results described above. Whether these findings hold in general, however, is an open question. Furthermore, we have developed a simple approach – essentially empirically Bayes – where we use a posterior mean, median or mode that does lie in the appropriate range to address these issues. The merits of this approach, however, are not yet fully investigated ([Dijkstra, 2018](#)).

This study provided initial evidence on the implications of neglecting measurement error correlation in terms of parameter accuracy. Clearly, this is of limited scope. Future research should investigate the consequences of our modified approach for model fit. Critics have repeatedly cautioned against pre-specifying measurement error

correlations, claiming that these correlations often lack a substantive meaning, which would in turn only obfuscate a meaningful interpretation of the specified model. In fact, for CB estimators such as maximum likelihood freeing, one or more measurement error correlations naturally leads to an increase in model fit, as the estimated model-implied covariance matrix is closer to its empirical counterpart. Similarly, common fit indices based on the distance between the estimated model-implied and empirical covariance matrix - such as the standardized root mean squared residual or the geodesic distance - generally indicate a better fit. The focus of this paper was on within indicator block measurement error correlation only. In the presence of unmodeled population measurement errors across blocks, the modification does not yield consistent estimates because the proportionality between weights and loadings as used to derive the correction factor no longer holds. As a consequence, loadings, reliabilities, and path coefficients pertaining to the blocks affected by measurement error correlation are generally inconsistent. Strategies to address unmodeled population measurement errors across blocks within the PLS/PLSc framework are thus needed.

---

## Chapter 4

# Estimating and testing second-order constructs using partial least squares path modeling: the case of composites of composites

---

### 4.1 Introduction <sup>1</sup>

Information systems (IS) research examines the impact of information technologies (ITs) on society, e.g., organizations or individuals. Consequently, it must combine the research paradigms of design and behavioral research. On the one hand, IS research deals with human-made objects such as smart home services (Yang et al., 2017) or transactional e-government services (Venkatesh et al., 2012), wherefore it can be regarded as “a science of the artificial” (Simon, 1969). On the other hand, it

---

<sup>1</sup>This chapter is based on joint work with Florian Schuberth and Jörg Henseler. It forms the basis for a paper with the same title. By the time this dissertation was submitted, the paper was under review for publication in the journal *Industrial Management & Data Systems*.

examines the interplay between such human-made objects and individuals in terms of, e.g., perceived usefulness of a hospital information system (Park et al., 2015) or the intention to use a particular smartphone app (Hew et al., 2015). Hence, it can also be understood as a behavioral science.

To meet the demands of both design and behavioral sciences in empirical research, structural equation modeling (SEM) is eminently suitable. SEM allows for operationalizing unobserved concepts by a set of observable variables and to connect these concepts as statistical entities in a structural model (Bollen, 1989b). In doing so, complex relationships such as reciprocal (e.g., Dijkstra and Henseler, 2015a) or nonlinear relationships (e.g., Klein and Moosbrugger, 2000; Dijkstra and Schermelleh-Engel, 2014) among the concepts can be taken into account. Hence, SEM allows researchers to statistically model and assess theories. Moreover, it can be used to test these theories which makes it a favorable tool for disciplines whose main objective is confirmatory research (e.g. Chin and Todd, 1995; Green and Inman, 2007; Henseler et al., 2016a).

In SEM, two ways of operationalizing unobserved concepts have been established: the latent variable model – also known as the common factor model/framework (e.g., Bollen, 1989b) – and the composite model (Fornell and Bookstein, 1982; Rigdon, 2016; Henseler, 2017; Schubert et al., 2018a).

Within the latent variable model framework, the reflective measurement model is the prevalent modeling approach in social and behavioral sciences; typically used to operationalize unobserved concepts such as abilities or attitudes. Researcher using a reflective measurement seek to explore whether the construct – i.e., a latent variable representing the unobserved concept –, exists by testing whether the relationships between its indicators – collected measures of that construct – are consistent with the implied restrictions of the reflective measurement model framework. By virtue of this type of operationalization, each indicator is modeled as a measurement-error-prone consequence of the latent variable. Consequently, there is a presumed causal relationship from the latent variable representing the unobserved concept on its connected indicators. Potential unobserved concepts from the field of IS that have been operationalized in a reflective measurement model are trust in mobile applications (Hajiheydari and Ashkani, 2018) and purchase intention (Hsu, 2017). Apart from the reflective measurement model, there is a highly controversial

alternative measurement approach called the causal-formative measurement model (e.g., [Diamantopoulos, 2008](#); [Edwards, 2011](#); [Aguirre-Urreta et al., 2016](#); [Bollen and Diamantopoulos, 2017](#)). Similar to the reflective measurement model, the concept in a causal-formative measurement framework is modeled as a latent variable. However, the presumed causality between indicators and the latent variable is reversed. Hence, the latent variable is assumed to be causally affected by its connected indicators. The remaining variance of the latent variable which is not explained by its indicators is captured by an error term. The causal-formative measurement model requires at least two paths to other variables as a sufficient condition for identification, e.g., additional indicators reflectively measuring other constructs affected by the causal-formative construct ([Bollen and Bauldry, 2011](#)).

In the composite model, the relationship between the indicators and the construct is modeled from the indicators to the construct. Unlike the causal-formative measurement model, however, the relationship between the indicators and the construct in a composite model is a definitorial rather than a causal one. Thus, an aggregate, i.e., an emergent variable, of – typically measurement-error-free – indicators is built to represent the underlying unobserved concept. Although the composite model as a way of operationalizing unobserved concepts is not as prevalent as the reflective measurement model in social and behavioral sciences, it is acknowledged by researchers across several disciplines ([Diamantopoulos, 2008](#); [Grace and Bollen, 2008](#); [Rossiter, 2002](#)). Examples from IS research are the unobserved concepts organizational Internet use ([Brock and Zhou, 2005](#)) and IT infrastructure capabilities ([Benitez et al., 2018b](#)). Therefore, unobserved concepts that are conceived as constructions – i.e., a human-made object composed by its ingredients to fulfill a certain purpose, a so-called artifact – ([Henseler, 2017](#); [Schuberth et al., 2018a](#)) are natural candidates for a composite operationalization.<sup>2</sup> Compared to the latent variable model, the composite operationalization is more pragmatic in a sense

---

<sup>2</sup>Besides using the composite model to operationalize artifacts, the current literature on composite-based SEM argues that both common factors and composites can be used to operationalize unobserved concepts of behavioral sciences ([Rigdon, 2016](#); [Rigdon et al., 2017, 2019](#)). While we agree with the idea in principal, in the following, we distinguish between unobserved concepts of behavioral sciences and unobserved concepts of design science (artifacts) to motivate the use of composites.

that it seeks to explore whether building an artifact is useful at all.<sup>3</sup>

Based on a given operationalization – latent variable or composite – two types of estimation methods may be distinguished: (i) factor-based methods/estimators and (ii) composite-based methods/estimators.<sup>4</sup> While factor-based estimators such as maximum likelihood minimize the discrepancy between the model-implied and the empirical indicator covariance matrix to obtain model parameter estimates (e.g., Jöreskog, 1970a; Bollen, 1989b), composite-based estimators always use a composite, i.e., a weighted sum of indicators as representation of a concept, and subsequently retrieve model parameters based on these composites – typically by running a series of regressions. The latter has important implications for interpretation; when the composite used in the estimation represents a potentially error-prone proxy for an underlying latent variable, parameters estimates based on regressions between proxies are likely to be inconsistent estimates for their latent variable counterpart. In contrast, regressions between composites representing composites in their own right, i.e., entities in a composite model, generally yield consistent estimates for the population path between these composites. It is therefore crucial to distinguish between composites as entities of the composite *model* and composites as entities of a composite-based *estimator* such as the partial least squares path modeling (PLS-PM) algorithm (e.g., Wold, 1982; Lohmöller, 1989) which we use in this paper.

While the PLS-PM algorithm features most prominently among composite-based methods, factor score regression using Bartlett, unit or regression weights (Devlieger and Rosseel, 2017), principal components, techniques for the canonical correlation analysis of several sets of indicators (Kettenring, 1971), and generalized structured component analysis (GSCA; Hwang and Takane, 2004) constitute alternative composite-based approaches.

In recent decades, hierarchical constructs – so-called higher-order constructs – have gained popularity in SEM (e.g., Law and Wong, 1999; Johnson et al., 2012; Polites et al., 2012). In contrast to unidimensional constructs assumed to represent one

---

<sup>3</sup>It is important to note here that the traditional nomenclature *formative* or *formative measurement* has an ambiguous meaning as, although fundamentally different, both the composite model and the (causal-)formative measurement model haven been referred to as *formative measurement* by different researchers. Hence, we deliberately refrain from using these words to denote a composite model to avoid any ambiguity.

<sup>4</sup>Factor-based methods are often called covariance-based methods as focal parameters are retrieved such that the difference between the model-implied and the empirical indicator covariance matrix is minimized. Composite-based methods are also called variance-based methods as focal parameters are usually retrieved such that the explained *variances* are maximized.



layered concepts, a higher-order construct contains several layered structures of constructs and therefore involves several dimensions, i.e. represents a multidimensional concept. Although, in general, any number of levels of abstraction is conceivable, the overwhelming majority of empirical studies involving higher-order constructs utilizes a second-order construct. In the remainder of this chapter, we therefore focus on these types of higher-order constructs.

Figure 4.1 contrasts the four main types of second-order constructs commonly encountered in applied research.

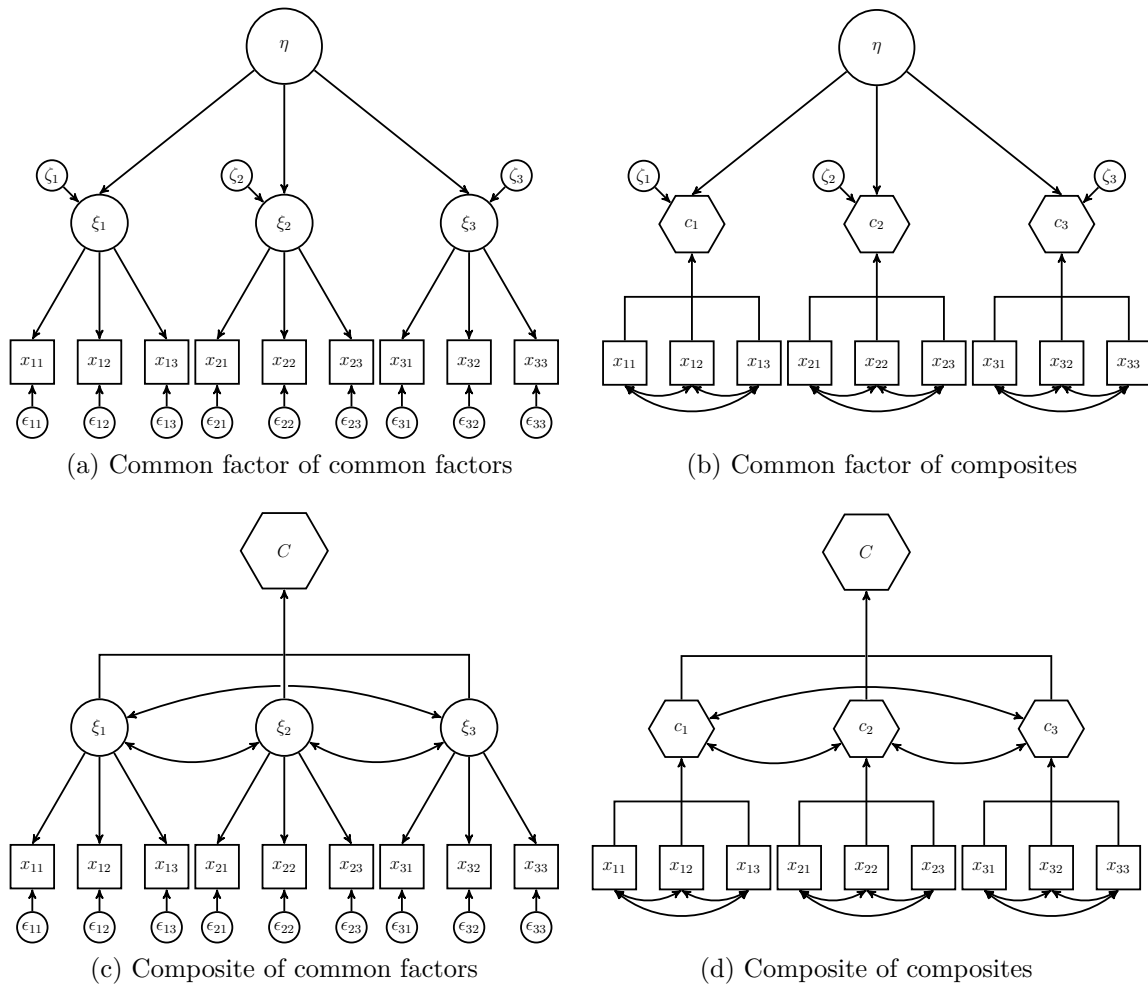


Figure 4.1: The four main types of second-order constructs

In line with common symbol usage, we use circles for latent variables or common factors and hexagons for composites. Squares are observables. A second-order construct modeled as a common factor measured by common factors is generally referred to as reflective-reflective second-order construct (Figure 4.1a). This type of

operationalization is typically used to model a multidimensional concept assumed to be measured by several unobserved concepts each represented by a unidimensional construct which is itself measured by a set of observable variables.

Reflective-reflective second-order constructs are preferably estimated by factor-based estimators – typically maximum likelihood. Examples include job satisfaction (Edwards, 2001) or a recent study by Nunkoo et al. (2017) investigating service quality. A second-order construct modeled as a common factor measured by composites as shown in Figure 4.1b is practically absent in empirical research (Becker et al., 2012). Although such second-order constructs might be useful for item parceling (Little et al., 2002, 2013) by using weights obtained by PLS-PM instead of unit weights, this type of hierarchical construct is currently of least practical relevance among the four types. In contrast, second-order constructs modeled as composites build by common factors (Figure 4.1c) are frequently found and studied in the literature (e.g., van Riel et al., 2017; Sarstedt et al., 2019).

The last type of second-order construct shown in Figure 4.1d constitutes a second-order construct modeled as a composite build by composites which are in turn build by a weighted linear combination of observable indicators. Such composite-composite operationizations are an obvious candidate to statistically examine artifacts which are build by other artifacts. The practical relevance of this kind of second-order construct is highlighted by Becker et al. (2012) who reviewed 25 models containing hierarchical constructs in the journal 'Management Information Systems Quarterly' and identified that composite-composite second-order constructs are the second most often employed. Concrete empirical examples for this kind of second-order construct are student satisfaction (Rueda et al., 2017), social media capability (Benitez et al., 2018a), and IT infrastructure capability (Benitez et al., 2018b).

To estimate a model containing second-order constructs, several approaches have been applied in empirical research, namely the repeated indicators approach (Wold, 1982), the two-stage approach (Agarwal and Karahanna, 2000), and the hybrid approach (Wilson and Henseler, 2007). However, while there are some theoretical research papers investigating these approaches (Wilson and Henseler, 2007; Becker et al., 2012; van Riel et al., 2017; Duarte and Amaro, 2018), their statistical evaluation has received comparatively little attention in the past. This is particularly

true with respect to composite-composite second-order constructs. While most studies focus on composite or common factor second-order constructs built by common factors, none of the studies has investigated the approaches' performance for composite-composite second-order constructs. Moreover, since a test for assessing the overall model fit was introduced only recently (Dijkstra and Henseler, 2015a), the current literature lacks guidance on how to statistically assess the overall model fit for models containing second-order constructs. Recognizing this fact, van Riel et al. (2017) provided a guideline for testing the overall model fit for models containing second-order constructs modeled as composites of common factors. However, recommendations for estimating and assessing models containing composite-composite second-order constructs have not been provided, nor has the testing procedure proposed by van Riel et al. (2017) been subject to a simulation study. The paper at hand seeks to close these gaps in the literature.

The remainder of the paper is organized as follows. Section 4.2 reviews approaches used to estimate models containing second-order constructs. Section 4.3 discusses a testing procedure to examine whether the restrictions implied by including composite-composite second-order constructs into the model are in line with the data. In Section 4.4 the performance of commonly used estimators for composite-composite second-order constructs is evaluated and compared by means of a Monte Carlo simulation. Additionally, the performance of the proposed test is evaluated both in terms of power (the probability of rejecting the null hypothesis when it is false i.e., the ability to detect misspecification) and type-I error rate (the probability of rejecting the null hypothesis when it is true). The paper closes with a discussion and an outlook for future research in Section 4.6.

## 4.2 Approaches to estimate models containing second-order constructs

To review commonly used approaches to estimate models containing second-order constructs we use the minimal model depicted in Figure 4.2. The model exhibits only the features necessary to illustrate these approaches.

The simple model consists of one second-order construct modeled as a composite ( $C$ ) that is built by two first-order constructs which are themselves modeled as

composites ( $c_1$  and  $c_2$ ). Empirically, this setup may be used to model a situation where two human-made objects make up a second artifact. The two sets of indicators ( $y_{1i}$  and  $y_{2i}$ ,  $i = 1, 2$ ) completely define/build the respective first-order composites which, in turn, fully determine the second-order composite. Furthermore, the model contains an exogenous common factor ( $\xi$ ) that, in contrast to the first-order composites, causally affects the second-order composite ( $C$ ). The common factor is connected to two indicators ( $x_1$  and  $x_2$ ). Consequently, the covariances between the two sets of indicators  $y_{1i}$  and  $y_{2i}$  with  $\xi$  are fully mediated by  $C$ . Similarly, the covariance between the indicators  $y_{1i}$  and  $y_{2i}$  with those of the indicators  $x_1$  and  $x_2$  are fully mediated by the first-order and the second-order composites, respectively. To preserve clarity, the structural error, measurement errors, and covariances among the variables are not depicted.

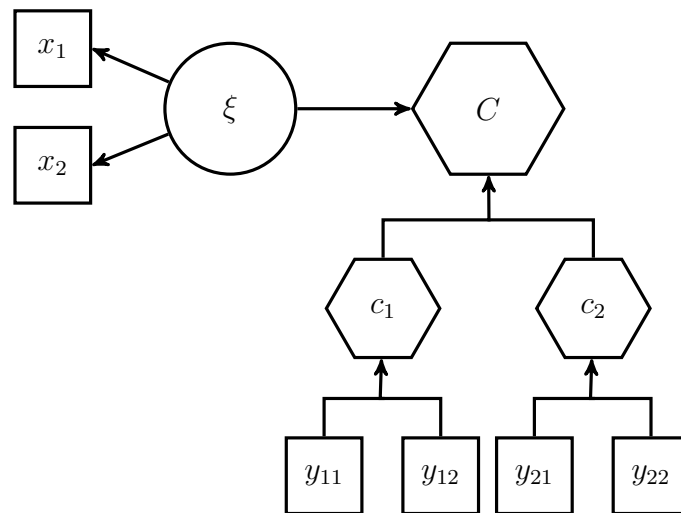


Figure 4.2: Example of a model containing a second-order composite formed by composites

The existing literature suggests three approaches to estimate models containing higher-order constructs: (i) the *repeated indicators approach*, (ii) the *two-stage approach*, and (iii) the *hybrid approach*. Each approach has sub-approaches that are best seen as extensions or variations of the main approaches.<sup>5</sup> Table 4.1 summarizes and illustrates the approaches and extensions relevant for this paper by means of the model example illustrated in Figure 4.2.

<sup>5</sup>Most theoretical and empirical literature discusses or applies these approaches in the PLS-PM framework. It is worth noting, however, that none of these approaches is conceptually tied to PLS-PM. Researchers should be aware that, while PLS-PM is indeed most prominent among composite-based methods, it may be more adequate to consider e.g., the repeated indicators approach as a composite-based approach to estimate a structural equation model containing higher-order constructs.

Table 4.1: Approaches to estimate models containing composites of composites

Approach:	(Extended) Repeated indicators approach (Wold, 1982)	Two-stage approach (Agarwal and Karahanna, 2000)	Embedded two-stage approach (Wilson, 2010)	(Extended) Hybrid approach (Wilson and Henseler, 2007)
Illustration:		<p>First stage:</p> <p>Second stage:</p>	<p>First stage:</p> <p>Second stage:</p>	
Description:	<p>All indicators of the first-order constructs building the second-order construct are used as indicators for the second-order construct. Originally, it was proposed to specify only the direct effect of the antecedent construct on the second-order construct. As a consequence, the second-order construct is fully explained by the first-order constructs <math>c_1</math> and <math>c_2</math> and all path from antecedent constructs (here the path from <math>\xi</math> on <math>C</math>) are estimated towards zero. To address this shortcoming, it was proposed to additionally specify the indirect effects of the antecedent constructs on the second-order construct through the first-order constructs (dotted lines) to estimate the effect of the respective antecedent construct on the second-order construct as the total effects (Becker et al., 2012).</p>	<p>Stage 1: Construct scores (<math>\hat{c}_1</math>, <math>\hat{c}_2</math>, and <math>\hat{\xi}</math>) are calculated based on a (saturated) structural model without the second-order construct.</p> <p>Stage 2: Construct scores are used as indicators to estimate the structural model containing the second-order construct.</p>	<p>Stage 1: Construct scores (<math>\hat{c}_1</math>, <math>\hat{c}_2</math>, and <math>\hat{\xi}</math>) are calculated by the repeated indicators approach.</p> <p>Stage 2: Construct scores are used as indicators to estimate the structural model containing the second-order construct.</p>	<p>Split the sets of original indicators of the first-order constructs in half, such that one half of each set is used as indicators of the first-order constructs, while the other half of each set is used as indicators of the second-order construct. Originally, it was proposed to specify only the direct effect of the antecedent construct on the hierarchical construct. As a consequence, the second-order construct is fully explained by the first-order constructs <math>c_1</math> and <math>c_2</math> and all path from antecedent constructs (here the path from <math>\xi</math> to <math>C</math>) are estimated towards zero. To address this shortcoming, it was proposed to additionally specify the indirect effects of the antecedent constructs on the second-order construct through the first-order constructs (dotted lines) to estimate the effect of the respective antecedent construct on the second-order construct as the total effects (Becker et al., 2012).</p>

### 4.2.1 Repeated indicators approach

The repeated indicators approach was originally proposed by [Wold \(1982\)](#). According to the repeated indicators approach all indicators of the first-order constructs building the second-order construct are reused as indicators of the second-order construct ([Wold, 1982](#); [Lohmöller, 1989](#)). This essentially reduces the second-order construct to a first-order construct. Although easy to implement, there are both practical and conceptual downsides tied to the repeated indicators approach.

Being build by  $y_{1i}$  and  $y_{2i}$  respectively, the first-order constructs  $c_1$  and  $c_2$  are by construction perfect predictors of  $C$ . Practically, retrieving the path coefficient between an antecedent construct of the second-order construct (here: between  $\xi$  and  $C$ ) in a multiple regression based on composite scores obtained by any composite-based approach will fail (i.e., will be close to zero) as all the variation in  $C$  is fully explained by  $c_1$  and  $c_2$ . To overcome this shortcoming, [Becker et al. \(2012\)](#) proposed to additionally specify indirect effects of the lower-order antecedent construct on the second-order construct through the first-order constructs forming the second-order construct and recover a consistent estimate of the path from  $\xi$  to  $C$  as the total effect of the antecedent construct on the second-order construct.

According to the commonly called *extended repeated indicator approach* researcher would therefore add a path from  $\xi$  to  $c_1$  and one from  $\xi$  to  $c_2$ , compute the total effect and interpret this as the path from  $\xi$  to  $C$ .

If (out-of-sample) prediction is the sole purpose of the analysis, specifying or removing path is unproblematic as long as the prediction error is improved. However, if any kind of confirmatory or inferential statement is to be made, adding additional paths or reducing the higher-order construct to a first-order construct is conceptually hard to conceive as it challenges the very idea of the model the researcher theorized in the first place. By changing the model, the model being estimated by the repeated indicators approach no longer corresponds to the theoretical model the researchers had in mind. This is particularly true for the extended repeated indicators approach. By specifying additional path, researchers are likely to introduce new, potentially unintended, restrictions which alter the model-implied indicator covariance matrix and, subsequently, any quantity based thereon. In the simple model above,  $c_1$  and  $c_2$  are exogenous and therefore allowed to freely correlate. Introducing path from  $\xi$  to  $c_1$  and  $c_2$ , say  $\pi_1$  and  $\pi_2$ , renders  $c_1$  and  $c_2$  endogenous, essentially restricting the

correlation between  $c_1$  and  $c_2$  to be  $\pi_1\pi_2$ . This would alter the model-implied indicator covariance matrix. Since most common fit measures and model fit tests are based on some kind of discrepancy between the empirical and the model-implied indicator covariance matrix, any such measure will thus be affected. Hence, it is unclear how such fit measures are to be interpreted in this case. To conclude: while it is generally possible to leave the model-implied indicator covariance matrix unchanged by adding yet another set of additional path, researchers would have to be aware of the precise mechanism giving rise to the model-implied indicator covariance matrix. In the simple case above, adding a path from  $c_1$  to  $c_2$  (or vice-versa) would lift the restriction thereby reinstating the original covariance structure.

## 4.2.2 Two-stage approach

As the name suggests, the two-stage approach consists of two stages ([Agarwal and Karahanna, 2000](#); [Henseler et al., 2007](#)). In the first stage, the aim is to obtain construct scores. Two strategies can be applied. First, all second-order constructs are removed from the original model; the remaining model is estimated and construct scores are saved. Technically, after removing the second-order constructs, researchers may use any recursive structural model specification involving the remaining constructs to obtain construct scores – provided all constructs are related in a nomological net ([van Riel et al., 2017](#)). Practically, however, for testing purposes there are advantages to specifying a saturated structural model instead – i.e., a structural model with zero degrees of freedom. See [Section 4.3](#) for explanation. A second strategy proposed by [Wilson \(2010\)](#) and [Ringle et al. \(2012\)](#) uses the repeated indicators approach to obtain construct scores in the first stage for all constructs, including the second-order construct. Again, any recursive structural model specification involving the remaining constructs could be used – provided all constructs are related in a nomological net.

In the second stage, the original structural model involving all constructs is estimated using the scores obtained in the first stage as indicators of the second stage constructs. Practically, all non-second-order constructs therefore become single indicator constructs. If the second-order construct is modeled as a common factor or if any of the first-order constructs forming/measuring the second-order construct are modeled as common factors, a correction for attenuation is necessary to obtain

consistent parameter estimates (van Riel et al., 2017); a process van Riel et al. (2017) label the *the three-stage approach*.

Consequently, in contrast to the repeated indicators approach, at least two estimation steps are required to estimate the model parameters using the two-step approach. To distinguish between the two versions of the two-stage approach, we refer to the approach involving the repeated indicators approach in the first step as the *embedded two-stage approach*. The alternative involving a saturated first stage model is labeled the *two-stage approach*.

### 4.2.3 Hybrid approach

The hybrid approach (Wilson and Henseler, 2007) is similar to the repeated indicators approach but avoids reusing the indicators of the first-order constructs connected to the second-order construct. Instead, the sets of indicators of the first-order constructs connected to the second-order construct are split in half: one half of the indicators is used as indicators of the second-order construct, while the other half remains as indicators of the respective first-order constructs. The primary intention of this procedure is to avoid artificially correlated residuals (Wilson and Henseler, 2007) although it has never been formally investigated whether the hybrid approach adequately serves this purpose. For constructs modeled as composites, correlated residuals are, by definition of the composite, not an issue. Moreover, using only a subset of the indicators to define the composites contradicts the idea of a composite as an artifact assumed to be fully defined by a unique set of indicators (Bollen and Bauldry, 2011). Because of the conceptual shortcoming and the fact that the approach is rarely applied in practice, the Hybrid approach is not considered in the Monte Carlo simulation.

## 4.3 Assessment of the overall model fit

Many empirical questions are inferential, causal or even mechanistic in nature (Leek and Peng, 2015). Statistically valid answers to these types of questions critically hinge on the correctness of both the postulated model and the properties of the estimates of focal model parameters. The latter typically comprises the need for consistent and/or unbiased model parameter estimates and an adequate measure of the uncertainty tied



to these estimates, i.e., consistent standard errors. Assessing the correctness of the postulated model involves both theoretical and statistical considerations. On the one hand, researchers must, for instance, assess whether to model a certain concept as a higher-order construct is theoretically reasonable or whether restricting a path from one construct to another to zero has theoretical support. On the other hand, researchers can draw on a variety of statistical quality criteria and tests generally referred to as model fit criteria and model fit tests respectively. Statistical assessment of overall model fit in SEM can therefore be divided in (i) measures of model fit, i.e. model fit indices and (ii) tests for (exact) overall model fit.

Model fit indices are broadly categorized as absolute or relative (McDonald and Ho, 2002). The principal idea of absolute model fit indices is to measure the correspondence between model and data along a continuum to gauge how well the postulated model corresponds to the data, instead of merely defining a perfect correspondence as the (only) desirable objective (e.g., Mulaik et al., 1989, p. 431). Relative fit indices, on the other hand, compare the postulated model to a reference model (the “null model”) to assess the relative increase in model fit. Accordingly, model fit assessment is rather a question of “close” (Browne and Cudeck, 1992) or “comparative” (Bentler and Bonett, 1980; Bentler, 1990) fit than that of perfect fit. Classification of the magnitude of a given fit index as “sufficiently close” or “comparatively better” is based on heuristic rules; fit index values are compared to a threshold to decide whether a model shows an acceptable model fit. The most prominent fit index in the context of PLS-PM is the standardized root mean square residual (SRMR, Bentler, 1995), where a value smaller than 0.08 is commonly taken to indicate an acceptable model fit.<sup>6</sup> Moreover, a multitude of fit measures such as the comparative fit index (CFI, Bentler, 1990) the normed fit index (NFI, Bentler and Bonett, 1980) or the root mean square outer residual covariance ( $RMS_{\theta}$ , Lohmöller, 1989) could be applied. While there is plethora of research on the (in)adequacy of fit indices in the context of factor-based SEM, we are unaware of any study in the context of the composite model. However, the principal critique surrounding test-like decision-making according to simulation-based cut-off values (e.g., Marsh et al., 2004) is likely to carry over to the composite model.

---

<sup>6</sup>The value of 0.08 is usually justified with reference to Hu and Bentler (1999). Its adequacy, however, has not been assessed for composite models yet.

In contrast to fit measures, tests for exact overall model fit test the null hypotheses that the population indicator covariance matrix equals the population covariance matrix implied by the model. Such tests follow the well-known hypothesis testing logic, i.e., postulate a null hypothesis, compute a test statistic using empirical data, and decide based on a comparison of the value of the test statistic to the distribution of the test statistic that were to arise if the null hypothesis were true.

Taking into account the usual limitations of hypothesis testing – i.e., falsely finding no difference when, in fact, there is one – a non-rejection of perfect fit is therefore statistical evidence that data and model do not stand in contradiction. Accordingly, a model that fits – within the limits of sampling error – is evidence that the restrictions implied by the postulated model – i.e. the theory –, are consistent with the data.

In analogy to the  $\chi^2$ -test in factor-based SEM (Jöreskog, 1967), any measure of discrepancy between the empirical indicator covariance matrix and the indicator covariance matrix implied by the model constitutes a reasonable test statistic.

Schuberth et al. (2018a) propose three such measures, namely the geodesic distance ( $d_G$ ), the squared Euclidean distance ( $d_L$ ), and the standardized root mean square residual (SRMR). As opposed to the  $\chi^2$ -test, however, non of these measures follow a known statistical distribution under the null hypothesis. To obtain the reference distribution in this case, it has become standard practice to use a bootstrap procedure based on Beran and Srivastava (1985) and Bollen and Stine (1992).

According to the procedure, the original indicator data is transformed such that the new data set satisfies the null hypothesis; i.e., the sample indicator covariance matrix of the transformed data set equals the estimated model-implied indicator covariance matrix under the null hypothesis. The reference distribution is now obtained by the bootstrap based on the transformed data. Finally, the test statistic is compared to a critical quantile of the reference distribution to obtain evidence in favor or against the null hypothesis. Usually, values larger than the 95% or 99% quantiles are considered sufficient evidence against the null hypothesis (Henseler et al., 2016b).

Assuming correct operationlization of all concepts in a structural equation model, the type of misfit can roughly be divided into misfit pertaining to the measurement and misfit pertaining to the structural model. Typical sources of the former are indicators which are assigned to the wrong construct or (unmodeled) measurement error correlation. Structural model misspecification, on the other hand, most frequently

arises when paths between constructs are falsely restricted to zero or when a relevant construct is incorrectly omitted from the postulated model (endogeneity). Since the model fit test is an overall test in a sense that all sources of misspecification are simultaneously taken into account, a rejection of the null hypothesis carries no specific information as to what constitutes the source of misspecification. Hence, researchers must find ways to investigate separate parts of the model in isolation. With regard to this issue, we propose a two-step strategy allowing researchers to distinguish between measurement and structural model misspecification.

In a first step, the entire model is estimated assuming a saturated structural model instead of the structural model postulated by the researcher. In this case, any systematic misspecification detected when testing the model fit must inevitably be due to misspecification in the measurement model – ignoring sample variation for now. This fact is most easily understood when counting the structural model degrees of freedom. In a model with  $J$  constructs there are  $J(J - 1)/2$  unique construct covariances (the number of non-redundant off-diagonal elements of the covariance matrix). In a saturated model each construct has one path to all other constructs such that the model is still recursive, i.e. there are no feedback loops. Hence, there are  $J(J - 1)/2$  unknown path coefficients. Consequently, by definition of the structural model degrees of freedom as the number of unique construct covariances minus the number of unknown parameters such a model has zero degrees of freedom. In other words: the saturated model contains no testable restrictions, consequently – ignoring sampling variation once again – any misfit found in a model with a saturated structural model must be due to measurement model misspecification.

In the second step, assuming that overall model fit in the first step was not rejected at the conventional significance levels, the original structural model specification is estimated. A rejection of the test for overall model fit in this case can now be interpreted as evidence against the null hypothesis of a correctly specified structural model.

### **4.3.1 Assessing the fit of higher-order constructs**

A specific kind of structural model misspecification occurs when higher-order constructs are used; for example, a researcher might assume that a concept of her model is multidimensional, i.e., is correctly represented in a statistical model as a

second-order construct.

Due to its two-step design, the two-stage approach with a saturated first-stage model naturally lends itself to the two-step testing procedure discussed above. Testing the first step therefore provides evidence as to whether the measurement model is correctly specified. This allows the researcher to decide whether it makes sense to continue building and estimating the second-order construct. If no evidence of systematic measurement model misspecification is found in the first stage, a rejection of the second stage model is evidence against the postulated higher-order specification. Consequently, to obtain empirical evidence for the usefulness of the second-order composite, it must affect or be affected by at least two other constructs; otherwise, the model is saturated and would always perfectly fit the data.

By design, alternative approaches such as the (extended) repeated indicators approach make model fit testing difficult, since it is unclear which model should be tested at all. From a testing point of view, the two-stage approach with a saturated structural model is therefore conceptually superior to other approaches discussed in the previous section – at least as long as confirmatory or inferential statements are to be made.

## 4.4 A Monte Carlo simulation

A Monte Carlo simulation is conducted to further contribute to the understanding of the approaches used to estimate models containing second-order composites and to investigate the performance of two proposed ways of model fit assessment. Due to the well-known drawback of the original repeated indicators approach in case of potential predictors of the second-order constructs (Becker et al., 2012; Ringle et al., 2012) and the inherent inappropriateness of the hybrid approach for concepts modeled as composites, we only investigate the two-stage approach based on a saturated structural model in the first stage, the embedded two-stage approach that combines the repeated indicators and the two-stage approach, and the extended repeated indicators approach that specifies additional effects.

### 4.4.1 Simulation design

To assess the performance of the various approaches, we choose a data generating process (DGP) that consists of one second-order composite ( $\eta_3$ ) built by three first-order composites  $c_i = \mathbf{w}'_i \mathbf{x}_i$ ,  $i = 1, 2, 3$ . Figure 4.3 shows the second-order DGP. For clarity, we omit error terms and correlations among variables forming a composite.<sup>7</sup>

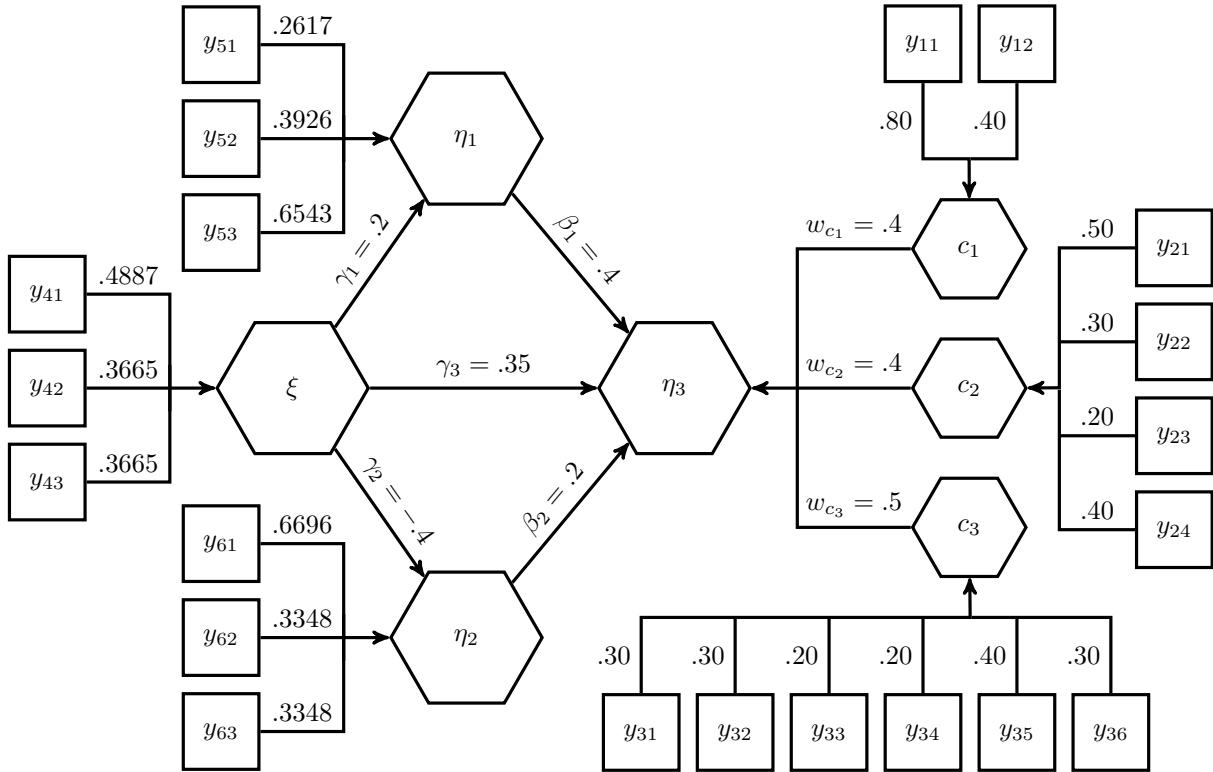


Figure 4.3: Second-order DGP

To be in line with recent guidelines regarding the assessment of models containing this type of second-order construct (Sarstedt et al., 2019), the population weights used to build the second-order composite are  $\mathbf{w}'_c = \begin{pmatrix} 0.4 & 0.4 & 0.5 \end{pmatrix}$  and the population correlations among the first-order composites are set to:  $r_{c_1 c_2} = 0.49$ ,  $r_{c_1 c_3} = 0.27$ , and  $r_{c_2 c_3} = 0.413$ . Thus, each first-order composite contributes substantially to the second-order composite and multicollinearity among the first-order composites is not an issue, i.e., first-order composites' correlations are below the recommended threshold of 0.5 and the VIF values of the weights are below 3 (Hair et al., 2017a). The number of indicators of the first-order composites is deliberately chosen unequally to investigate the claim that an unequal number of

<sup>7</sup>The complete population indicator covariance matrix implied by the DGP is provided in Figure 4.5 of the Appendix.

indicators has an adverse effect on the parameter estimates of the repeated indicators approach. The first-order composite  $c_1$  is built by two indicators ( $y_{11}$  and  $y_{12}$ ), the first-order composite  $c_2$  is built by four indicators ( $y_{21}$  to  $y_{24}$ ), and the first-order composite  $c_3$  is built by six indicators ( $y_{31}$  to  $y_{36}$ ). To build the first-order composites  $c_1$  to  $c_3$ , we again choose indicator weights and correlations that are conform with current guidelines on the assessment of models containing second-order composites:

$$\mathbf{w}'_1 = \begin{pmatrix} 0.8 & 0.4 \end{pmatrix} \quad (4.1)$$

$$\mathbf{w}'_2 = \begin{pmatrix} 0.5 & 0.3 & 0.2 & 0.4 \end{pmatrix} \quad (4.2)$$

$$\mathbf{w}'_3 = \begin{pmatrix} 0.3 & 0.3 & 0.2 & 0.2 & 0.4 & 0.3 \end{pmatrix} \quad (4.3)$$

Moreover, the population correlation between the two indicators of  $c_1$  is set to 0.3125, the population correlation matrices of the indicators forming  $c_2$  and  $c_3$  respectively are given in Equations 4.4 and 4.5. To preserve clarity, the correlations are rounded to the second decimal place.

$$\Sigma_{c_2} = \begin{pmatrix} \underline{y_{21}} & \underline{y_{22}} & \underline{y_{23}} & \underline{y_{24}} \\ 1.00 & 0.40 & 0.30 & 0.31 \\ & 1.00 & 0.28 & 0.31 \\ & & 1.00 & 0.30 \\ & & & 1.00 \end{pmatrix} \quad (4.4)$$

and

$$\Sigma_{c_3} = \begin{pmatrix} \underline{y_{31}} & \underline{y_{32}} & \underline{y_{33}} & \underline{y_{34}} & \underline{y_{35}} & \underline{y_{36}} \\ 1.00 & 0.10 & 0.25 & 0.13 & 0.10 & 0.30 \\ & 1.00 & 0.20 & 0.40 & 0.30 & 0.20 \\ & & 1.00 & 0.30 & 0.10 & 0.30 \\ & & & 1.00 & 0.20 & 0.20 \\ & & & & 1.00 & 0.10 \\ & & & & & 1.00 \end{pmatrix} . \quad (4.5)$$

Besides the first-order composites and the second-order composite, the structural model consists of three composites ( $\xi$ ,  $\eta_1$ , and  $\eta_2$ ) that are antecedents of the second-order composite  $\eta_3$ . The second-order composite is deliberately in an endogenous position to investigate the efficacy of the extended repeated indicators approach that was proposed to overcome the drawbacks of the original repeated

indicators approach in such a situation. In designing the structural model, we choose a complexity that is similar to models found in the empirical literature (e.g., [Ainin et al., 2015](#); [Yim and Leem, 2013](#); [Hsieh et al., 2006](#)). To ensure sufficient degrees of freedom of the structural model – which is required for model fit assessment of the structural model –, we opt for a multiple mediation structure with population path coefficients:  $\gamma_1 = 0.2$ ,  $\gamma_2 = -0.4$ ,  $\gamma_3 = 0.35$ ,  $\beta_1 = 0.4$ , and  $\beta_2 = 0.2$ . As a consequence, the effect sizes ( $f^2$ ) range from 0.04 to 0.22 indicating small and medium effects ([Cohen, 1992](#)). Moreover, the structural error terms are mutually independent. To assess the performance of the various approaches, we consider the estimated bias and the estimated root mean square error (RSME) which are defined as

$$\widehat{\text{Bias}} = \frac{1}{N} \sum_{i=1}^N \hat{\theta}_i - \theta \quad \text{and} \quad \widehat{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{\theta}_i - \theta)^2}, \quad (4.6)$$

where  $\theta$  represents a generic population parameter;  $\hat{\theta}_i$  is its corresponding estimate from the  $i$ -th Monte Carlo simulation run.  $N$  denotes the total number of Monte Carlo simulation runs. While the estimated bias indicates how much an estimate differs on average from its population counterpart, the RMSE combines the bias and the uncertainty involved in an estimate, namely its standard deviation. Additionally, we assess whether the approaches are Fisher consistent, i.e., whether they are able to retrieve the population parameters when given the indicator population covariance matrix as input.

To study the performance of the two presented testing procedures, we assess their type I error rate, i.e., the probability of falsely rejecting the null hypothesis, and the power, i.e., the probability of correctly rejecting the null hypothesis, under various conditions. When assessing the complete model, the estimated model-implied indicator covariance matrix of the model that matches the second-order DGP from [Figure 4.3](#) is compared to the sample indicator covariance matrix. For the two-step testing procedure, the fit of the model without the second-order composite and a saturated structural model is assessed; in the second step, the fit of the structural model containing the second-order composite and the corresponding first-order composites is tested. Estimations that were rejected in the first step are excluded in the second step.

Assessment of the power of the overall model fit can be conducted in two ways. According to the first approach a given data set drawn from a given (known) DGP is

used to estimate a model which does not represent the DGP (e.g., a path that is non zero in the population is falsely omitted). Consequently, the model is misspecified with respect to the DGP. Depending on the source and the strength of the misspecification a test should be able to detect this misspecification, i.e., correctly reject the null hypothesis of no misspecification. The second approach takes the model as given but varies the DGP such that it does not correspond to the postulated model. Again, a test should be able to detect this misspecification with sufficient power. We use the second approach in our simulation.

Clearly, the source and the magnitude of the misspecification are critical to power assessment. With respect to the source of the misspecification, we focus on the misspecification that arises when a researcher incorrectly imposes a second-order structure on the relationship between a set of constructs. Practically, we therefore take the second-order DGP in Figure 4.3, remove the second-order  $\eta_3$ , and draw direct path from all constructs antecedent to  $\eta_3$  to the respective lower dimensional constructs  $c_1 - c_3$ ; hence, we obtain an alternative DGP as shown in Figure 4.4. Indicators and weights are omitted as they remain unchanged.

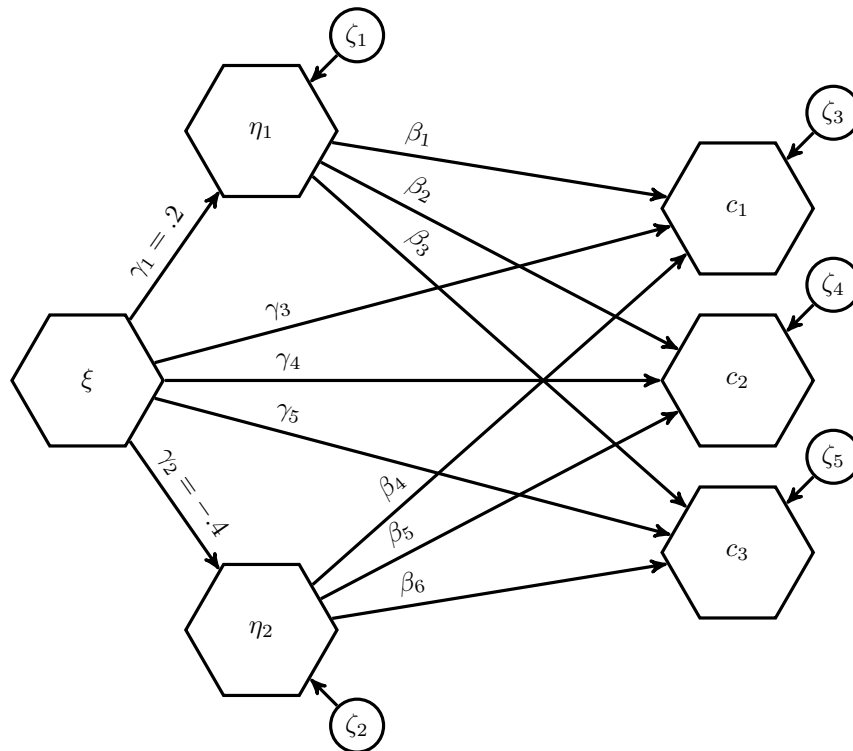


Figure 4.4: Structural specification of the alternative DGPs.

To be able to investigate power for varying levels of misspecification we choose three different parameter combinations for the path from  $\xi$ ,  $\eta_1$ , and  $\eta_2$  to  $c_1$ ,  $c_2$ , and  $c_3$ ,



respectively. The first parameter combination yields a small, the second a medium, and the third a large misspecification. Of course, small, medium, and large are to be understood in relative terms. We quantify misspecification in terms of the standardized root mean square residual (SRMR).<sup>8</sup> The parameter combinations that yielded the respective misspecifications are obtained by a procedure detailed in Section 4.8.2 of Appendix B.

The sets of path coefficients and the corresponding SRMR values are displayed in Table 4.2. Additionally, Table 4.2 reports the SRMR for a model without the second-order composite and a saturated structural model, i.e., the model from the first stage of the two-stage approach (SRMR<sub>1st</sub>) and the SRMR for the structural model containing the second-order composites with its first-order composites, i.e., the model from the second stage of two-stage approach (SRMR<sub>2nd</sub>). As can be seen, the misfit is fully caused by the structural model, since the SRMR for the model from the first stage is zero, while the SRMR for the model from the second stage is larger than zero. For larger misspecifications and larger sample sizes, we expect larger rejection rates. Rejection rates above 80% are desired (Cohen, 1988).

Table 4.2: Population path coefficients in case of misspecifications

Misspecification	SRMR	SRMR <sub>1st</sub>	SRMR <sub>2nd</sub>	$\gamma_3$	$\gamma_4$	$\gamma_5$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$	$\beta_6$
Small	0.0302	0.000	0.0626	0.4	0.1	0.5	0.1	0.5	0.6	0.1	0.2	0.5
Medium	0.0629	0.000	0.1262	-0.4	0.1	0.5	0.1	0.5	0.6	0.2	0.2	0.5
Large	0.0949	0.000	0.1718	0.4	0.1	0.5	0.5	0.4	0.6	-0.4	0.2	0.5

Our simulation design additionally highlights the importance of model fit assessment and emphasizes the need for updated guidelines on the assessment of models containing second-order composites. Regardless of whether the two-stage approach, the repeated indicators approach or embedded two-stage approach is employed, current guidelines for the assessment of formative measurement models<sup>9</sup> (Sarstedt et al., 2019; Hair et al., 2017b,a; Ramayah et al., 2016; Becker et al., 2012) fail to distinguish correctly from incorrectly specified models. On a population level, all four

<sup>8</sup>Alternatively, one could use the geodesic distance ( $d_G$ ) or the Euclidian distance ( $d_L$ ) to quantify the amount of misspecification. SRMR and  $d_L$  are identical except for a constant scaling factor. SRMR and  $d_G$  may slightly differ, however, the difference is negligible in a sense that categorizing misspecifications as small, medium, and large will be highly similar for the SRMR or  $d_G$ -based misspecification measure.

<sup>9</sup>In PLS-SEM parlance a formative construct is a composite whose weights are estimated by mode B.

estimated model, i.e., the correctly specified and the three misspecified models, are not indicated as problematic by current guidelines on the assessment of models containing composite of composites. In all cases, all indicators/first-order composites contributed substantially to their composite, i.e., all weights are sizable. Moreover, multicollinearity is not an issue as all VIF values are below the conservative rule of thumb of 3 suggested by e.g. [Hair et al. \(2017a\)](#) and the correlations among the indicators and first-order composites forming a composite are all below 0.5. Convergent validity cannot be assessed, as the model does not contain reflective measurement models. The results can be found in Table 4.11 in Appendix A. The simulation is conducted in the statistical programming environment R ([R Core Team, 2020](#)). To investigate the performance of the three approaches, the model that matches the second-order DGP from Figure 4.3 is estimated 1,000 times by each approach based on data of size:  $n = 100, 300, 500,$  and 1,000 drawn from the same DGP. We check each estimation run for admissibility and instantly replace inadmissible results such that in each case exactly 1,000 admissible results are available. To examine the type I error rate, we repeatedly draw data set from the second-order DGP of Figure 4.3, estimate the model which does indeed represent the DGP, and perform the test for exact overall model fit. As the model is correctly specified with respect to the DGP, we expect rejection rates close to the predefined significance level  $\alpha$ . The observations are drawn from a multivariate normal distribution with means of zero using the `mvrnorm()` function of the MASS package ([Venables and Ripley, 2002](#)). To examine the Fisher consistency of the approaches, the indicator population covariance matrix is used as input for the various approaches. To assess the power of the two testing strategies, the datasets are generated based on population covariance matrices implied by the different degrees of model misspecification displayed in Table 4.2. The PLS-PM parameter estimates are obtained by the `csem()` function of the cSEM package ([Rademaker and Schubert, 2020](#)). As outer weighting scheme Mode B is employed for all constructs since it has been shown to produce consistent estimates for composite models ([Dijkstra, 2017](#)). As stopping criterion the absolute change of the weights is considered, i.e., during the PLS-PM algorithm the largest absolute difference between the current weights and the weights from the previous iteration step are compared. In case that the largest absolute difference is smaller than  $10^{-5}$  the algorithm stops. Furthermore, the

maximum number of iteration is set to 100.

While for models without second-order constructs, all inner weighting schemes produce similar results (Noonan and Wold, 1982), there is only little advice on which scheme to use for models containing second-order composites (Becker et al., 2012; Sarstedt et al., 2019). Although the path weighting scheme seems to be favored for models containing composites of composites (Becker et al., 2012), we further examine whether the choice of the inner weighting scheme affects the behavior of the different approaches. Therefore, the path, the factorial, and the centroid weighting scheme are employed for every approach. Finally, the exact overall model fit test is conducted using the `testOMF()` function of the `cSEM` package. We use 1,000 bootstrap runs per test and replace non-converged estimations during the bootstrap. For the complete R code, see Appendix B.

## 4.4.2 Simulation results

### Behavior of the various approaches

Table 4.3 displays the results for the estimated bias, the estimated RMSE, Fisher consistency, and share of converged estimations of the three investigated approaches using the path, factorial, and centroid weighting scheme during PLS-PM's inner estimation. Due to space constraints, we present the results only for the estimates of two path coefficients ( $\gamma_1$  and  $\beta_1$ ), the first weight of the first-order composite  $c_1$  ( $w_{11}$ ), and the three weights used to build the second-order composite ( $w_{c_1}$ ,  $w_{c_2}$ , and  $w_{c_3}$ ). The remaining estimates behave similarly to those presented.

The two-stage approach based on a saturated structural model in the first stage appears to produce consistent estimates. The average estimates of all parameters converge to their population counterparts for an increasing sample size. Moreover, the decrease in the estimated RMSE reflects that the standard errors of the parameter estimates decrease when the sample size increases. This result is hardly affected by the choice of the inner weighting scheme. Furthermore, it produced the smallest number of non-converged estimations. In none of the conditions more than 0.6% of the estimations have not converged. Finally, it returns the population parameters when provided with the indicator population covariance matrix showing the Fisher consistency of the two-stage approach.

Table 4.3: Results for the estimated bias, estimated RMSE, Fisher consistency, and number of converged estimations

Inner weighting scheme	Sample size	Approach	% converged estimations	Estimated bias and Fisher consistency							Estimated RMSE						
				$\gamma_1$	$\beta_1$	$w_{11}$	$w_{c_1}$	$w_{c_2}$	$w_{c_3}$	$\kappa_2$	$\gamma_1$	$\beta_1$	$w_{11}$	$w_{c_1}$	$w_{c_2}$	$w_{c_3}$	$\kappa_2$
Centroid scheme	100	two-stage	100.00	0.037	0.003	-0.024	-0.055	0.008	-0.016	0.003	0.101	0.081	0.147	0.197	0.209	0.181	0.081
		embedded TS	99.80	0.048	-0.002	-0.021	-0.061	0.000	0.004	-0.002	0.108	0.083	0.149	0.194	0.193	0.158	0.083
		extended RI	100.00	0.033	-0.389	-0.031	-0.000	0.022	-0.045	0.017	0.099	0.389	0.171	0.054	0.063	0.069	0.081
	500	two-stage	100.00	0.005	0.000	-0.000	-0.014	0.009	-0.008	0.000	0.042	0.035	0.057	0.097	0.096	0.085	0.035
		embedded TS	100.00	0.007	-0.002	0.001	-0.015	0.006	-0.002	-0.002	0.042	0.035	0.056	0.090	0.091	0.076	0.035
		extended RI	100.00	0.003	-0.398	-0.001	0.019	0.025	-0.048	0.002	0.042	0.398	0.067	0.032	0.036	0.054	0.035
	1,000	two-stage	100.00	0.004	0.003	-0.005	-0.005	-0.002	0.000	0.003	0.030	0.026	0.042	0.064	0.070	0.060	0.026
		embedded TS	100.00	0.005	0.002	-0.005	-0.006	-0.001	0.001	0.002	0.030	0.026	0.042	0.061	0.067	0.056	0.026
		extended RI	100.00	0.003	-0.399	-0.006	0.022	0.023	-0.048	0.004	0.030	0.399	0.049	0.028	0.030	0.051	0.026
$\Sigma$	two-stage	-	-3.89e-16	5.00e-16	7.77e-16	-1.55e-15	2.11e-15	-1.22e-15	5.00e-16	-	-	-	-	-	-	-	
embedded TS	-	-3.33e-16	-2.27e-11	-1.24e-05	-6.66e-11	1.39e-10	-3.03e-11	-2.27e-11	-	-	-	-	-	-	-	-	
extended RI	-	-3.61e-16	-4.00e-01	-6.21e-06	2.41e-02	2.41e-02	-4.94e-02	-5.09e-04	-	-	-	-	-	-	-	-	
Factorial scheme	100	two-stage	99.80	0.031	0.014	-0.005	-0.056	-0.001	-0.004	0.014	0.110	0.086	0.134	0.193	0.196	0.180	0.086
		embedded TS	99.50	0.035	0.008	-0.003	-0.054	0.011	-0.021	0.008	0.112	0.086	0.133	0.189	0.193	0.172	0.086
		extended RI	98.70	0.027	-0.388	-0.007	-0.004	0.045	-0.075	0.022	0.107	0.388	0.145	0.047	0.064	0.090	0.086
	500	two-stage	100.00	0.005	0.005	-0.002	-0.010	-0.000	-0.001	0.005	0.042	0.036	0.053	0.079	0.085	0.074	0.036
		embedded TS	100.00	0.006	0.003	-0.002	-0.009	0.002	-0.005	0.003	0.043	0.036	0.055	0.079	0.085	0.073	0.036
		extended RI	100.00	0.004	-0.398	-0.002	0.015	0.055	-0.081	0.005	0.042	0.398	0.056	0.025	0.059	0.083	0.036
	1,000	two-stage	100.00	0.004	0.002	-0.000	-0.005	-0.000	-0.001	0.002	0.030	0.025	0.037	0.054	0.056	0.051	0.025
		embedded TS	100.00	0.004	0.001	0.000	-0.005	0.001	-0.004	0.001	0.030	0.025	0.038	0.054	0.056	0.052	0.025
		extended RI	100.00	0.003	-0.399	-0.000	0.017	0.056	-0.082	0.001	0.030	0.399	0.041	0.022	0.057	0.083	0.025
$\Sigma$	two-stage	-	-6.66e-16	2.78e-16	5.55e-16	6.66e-16	1.44e-15	-2.33e-15	2.78e-16	-	-	-	-	-	-	-	
embedded TS	-	-6.94e-16	-1.44e-11	-8.52e-06	-1.75e-12	3.05e-11	5.66e-12	-1.44e-11	-	-	-	-	-	-	-	-	
extended RI	-	-6.94e-16	-4.00e-01	-5.33e-06	1.97e-02	5.69e-02	-8.20e-02	-1.34e-03	-	-	-	-	-	-	-	-	
Path scheme	100	two-stage	99.40	0.022	0.031	-0.008	-0.045	-0.024	0.009	0.031	0.113	0.093	0.128	0.181	0.196	0.178	0.093
		embedded TS	89.10	0.031	-0.002	-0.016	-0.022	0.003	-0.051	-0.002	0.113	0.087	0.152	0.179	0.193	0.181	0.087
		extended RI	98.20	0.023	-0.400	-0.010	-0.153	0.045	0.058	0.012	0.105	0.400	0.137	0.156	0.057	0.064	0.085
	500	two-stage	100.00	0.005	0.008	-0.005	-0.011	-0.001	0.002	0.008	0.045	0.037	0.057	0.077	0.075	0.065	0.037
		embedded TS	100.00	0.007	0.000	-0.005	-0.006	0.005	-0.011	0.000	0.045	0.036	0.068	0.077	0.076	0.067	0.036
		extended RI	100.00	0.005	-0.400	-0.005	-0.144	0.050	0.069	0.001	0.045	0.400	0.060	0.144	0.051	0.070	0.036
	1,000	two-stage	100.00	0.000	0.004	-0.003	-0.001	-0.003	-0.001	0.004	0.030	0.025	0.039	0.052	0.056	0.048	0.025
		embedded TS	100.00	0.001	0.000	-0.002	0.001	0.000	-0.007	0.000	0.030	0.025	0.047	0.052	0.056	0.049	0.025
		extended RI	100.00	0.000	-0.400	-0.003	-0.143	0.050	0.069	-0.001	0.030	0.400	0.042	0.143	0.051	0.070	0.025
$\Sigma$	two-stage	-	-6.94e-16	2.78e-16	0.00e+00	9.44e-16	-1.67e-16	-1.11e-15	2.78e-16	-	-	-	-	-	-	-	
embedded TS	-	-6.94e-16	-1.05e-10	-4.58e-10	2.18e-10	5.74e-10	-6.17e-10	-1.05e-10	-	-	-	-	-	-	-	-	
extended RI	-	-6.94e-16	-4.00e-01	-1.35e-09	-1.42e-01	5.05e-02	7.01e-02	-3.63e-03	-	-	-	-	-	-	-	-	

Note: embedded TS: embedded two-stage approach; extended RI: extended repeated indicators approach; two-stage: two-stage approach based on a saturated structural model in the first stage;  $\Sigma$ : the indicator population covariance matrix is provided as input;  $\kappa_2$ : the total effect of  $\eta_1$  on  $\eta_3$ .

The embedded two-stage approach behaves similarly to the original two-stage approach, but it produces a larger share of non-converged estimations. The highest share of 10.9% was observed for small sample sizes ( $n = 100$ ) in combination with the path weighting scheme. However, the number of converged estimations decreases for an increasing sample size and diminishes for sample sizes  $n \geq 300$ . Considering the estimated bias, it decreases for all parameters with an increasing sample size and almost diminishes for a sample size of 1,000 observations. The estimated RMSE also decreases for all parameters with an increasing sample size, as the standard errors of the estimates decrease. In contrast to the two-stage approach, the embedded two-stage approach cannot retrieve all population parameters from the indicator population covariance matrix. The weights to build the first-order composites slightly differ from their population counterparts.<sup>10</sup> Among the three inner weighting schemes, the path weighting scheme produces for the weights used to build the first-order composites the smallest deviation from their population counterpart. The extended repeated indicators approach, i.e., indirect effects of  $\xi$ ,  $\eta_1$ , and  $\eta_2$  on the second-order composite  $\eta_3$  are specified through the first-order composites, behaves similar to the two other approaches for the considered sample sizes when the centroid or the factorial schemes are employed. In case of the path weighting scheme, the weight estimates used to build the second-order composite considerably deviate on average from their population counterparts. In contrast to the other model parameters, the estimated bias of these weights does not diminish for an increasing sample size. The large values of the estimated RMSE for these weights show that the estimates have larger standard errors compared to the other approaches. As expected, the direct effects of constructs on the second-order composites are biased towards zero, however, the results confirm that the total effects can be used to capture these effects. Moreover, in none of the conditions the share of non-converged estimations is above 1.8%. However, not all population parameters can be retrieved when the indicator population covariance matrix is provided as input. This result is unaffected by the employed inner weighting scheme.

---

<sup>10</sup>Although not presented, in case of the path weighting scheme, the retrieved weights used to build  $c_3$  deviate from their population counterpart. A table containing all estimated parameters can be found in Tables 4.9 and 4.10 of the Appendix.

## Performance of the two testing procedures

In the following, we present the results of our simulation to assess the performance of the two presented testing procedures based on the two-stage approach employing the path weighting scheme and using a saturated structural model in the first stage. We additionally employed the centroid and the factorial weighting scheme, however, the results are very similar.

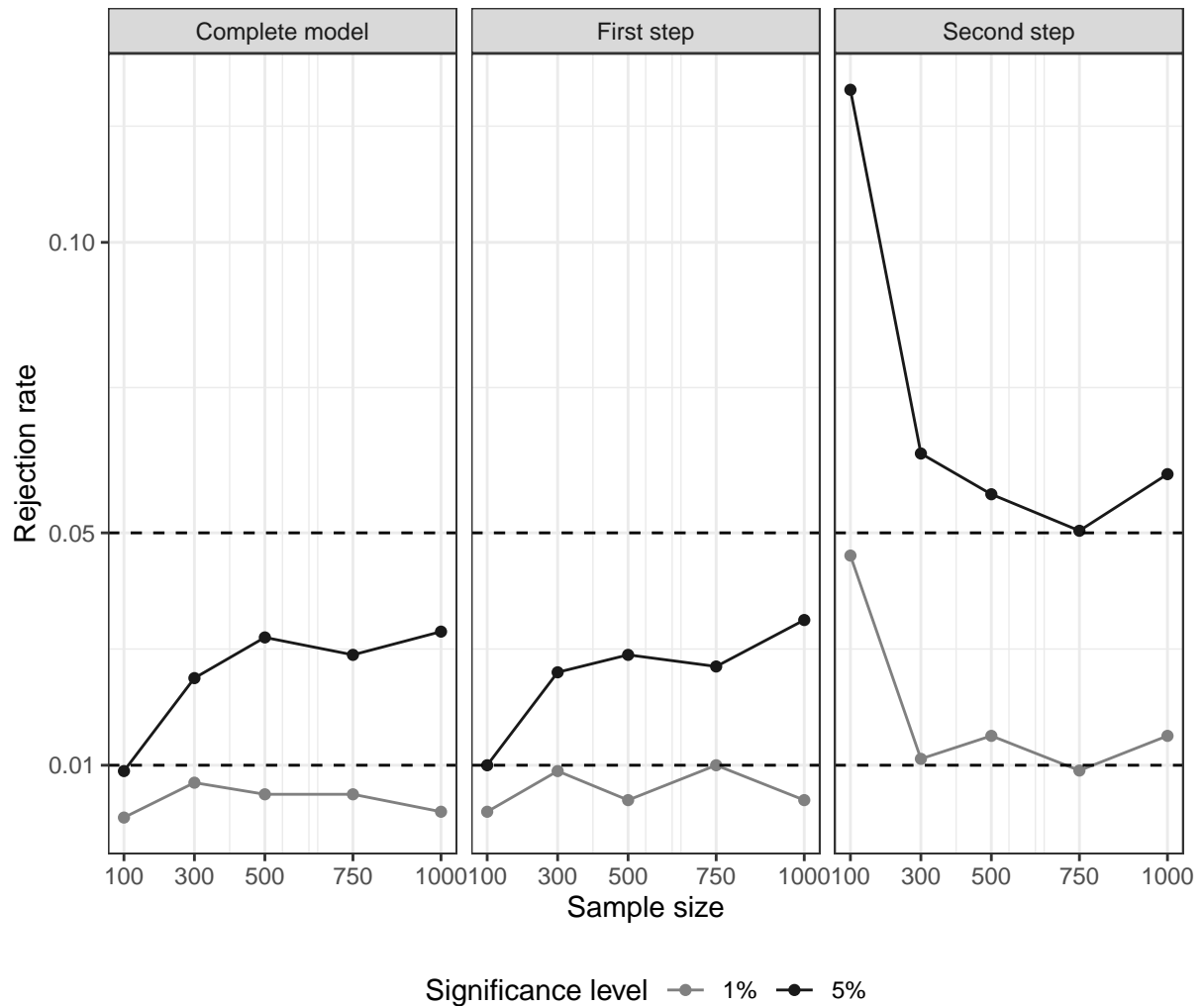


Figure 4.5: Tests' rejection rates in case of no misspecification

Figure 4.5 displays the tests' rejection rates in case of no model misspecification, i.e., the null hypothesis is true. The complete model testing procedure produces rejection slightly below the predefined significance level. Particularly, for a small sample size and a significance level of 5%, the rejection rates are too small. However, the rejection rates converge to the predefined significance level for an increasing sample size. Similar can be observed for the first step of the two-step testing procedure. In

contrast, in the second step of the two-step testing procedure, the rejection rates are too conservative, i.e., the model is rejected too often. However, for an increasing sample size, the rejection rates converge to predefined significance level.

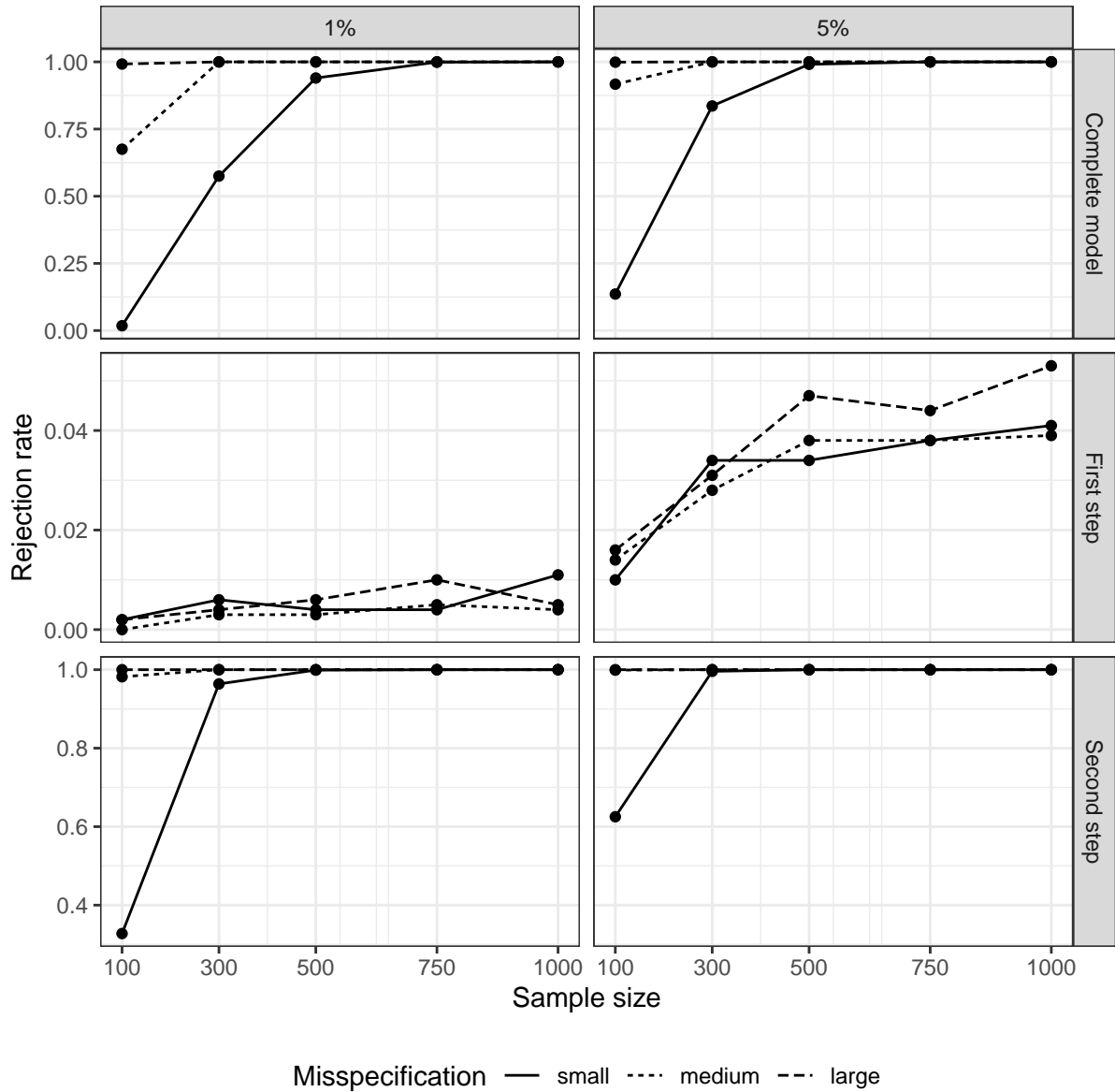


Figure 4.6: Tests' rejection rates for various degrees of misspecification

Figure 4.6 depicts the rejection rates in case of the various misspecifications and the two considered significance levels of 1% and 5%. For the complete model testing, large and medium misspecified models are detected in all cases, except for small sample sizes. However, in this case, it still reliably detects large and medium misspecifications. For models with small misspecification, larger sample sizes are required to reject the null hypothesis. In case of small sample sizes, small misspecification cannot be reliably detected.

In the first step of the two-step testing procedure, the rejection rates are close to the predefined significance level. This is not surprising, as the model from the first stage is not misspecified, since misspecifications in the structural model are ignored because the specified structural model is saturated. In contrast, in the second step of the two-step testing procedure almost all misspecifications are reliably detected, except small misspecification in case of small sample sizes.

### **4.4.3 Simulation insights**

The Monte Carlo simulation shows that for finite sample sizes all three approaches perform similarly in terms of parameter recovery, i.e., estimated bias and RMSE. An exception is the extended repeated indicators approach in combination with the path weighting scheme, which produced considerably biased estimates for those weights that are used to build the second-order composite. Moreover, our simulation confirms that the extended repeated indicators approach is able to estimate the constructs' direct effects on the second-order composite by their total effects. Considering Fisher consistency, only the two-stage approach returns all population parameters when the indicator population covariance matrix is applied. Additionally, its results are hardly affected by the inner weighting scheme.

Considering the results of the two presented testing procedures, both produce rejection rates slightly off the predefined significance level in case of no model misspecification. However, the rejection rates converge to the significance level along with an increasing sample size. Moreover, both procedures reliably detect most of the misspecification. As expected, small misspecifications require larger sample sizes to be reliably detected and the tests' power increase for an increasing sample size.

## **4.5 Guidelines for estimating and assessing models containing composites of composites**

To enable researchers to appropriately estimate and assess models containing composites of composites, particularly in the context of confirmatory and explanatory research, we provide guidelines embracing the two-stage approach and the two-step testing procedure. Although our simulation study shows that both testing procedures

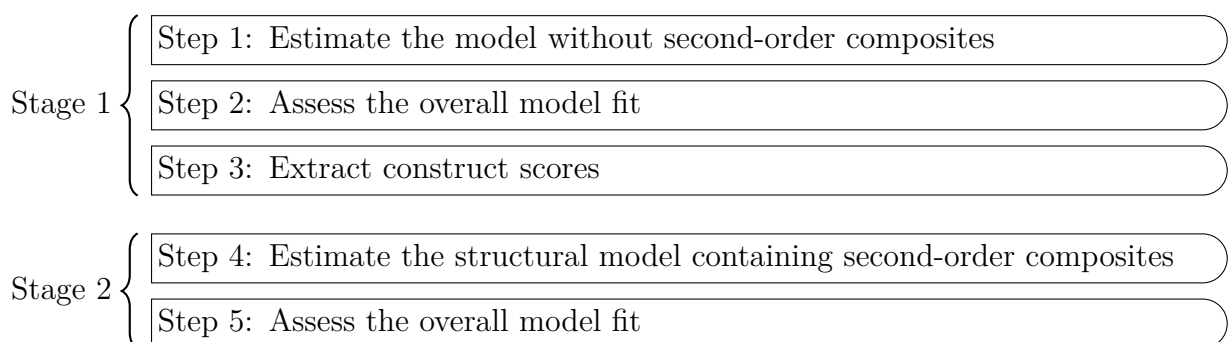


detect similarly reliably misspecification in the structural model, the two-step testing procedure is recommended in the following as it allows for a separate assessment of the concepts' operationalization and the structural model. To fully leverage the strength of the two-step testing procedure, a saturated structural model should be specified in the first stage of the two-stage approach. This allows for spotting the location of the misspecification, i.e., in the operationalization of the concepts and/or in the structural model.

Prior to the estimation of the model and its assessment, the researcher should ensure that the model is identified, guaranteeing that the model parameters can be uniquely retrieved from the system of equations. Indeed, model identification is straightforward in the case of composite models (Dijkstra, 2017; Schubert et al., 2018a); however, it must not be disregarded. A necessary condition for the identification of a composite model is that each (higher-order) composite is built by at least one indicator/lower-order composite and that no (higher-order) composite is isolated in the structural model. Apart from all composites being connected via the structural model, it must be additionally ensured that the structural model is also identified. Since the study at hand focuses only on recursive models, we refer to the 'recursive rule', which states that recursive models with uncorrelated structural error terms are always identified (Bollen, 1989b, p. 104).

After model identification is ensured, the model can be estimated, and its fit can be assessed. For this purpose, guidelines are developed and depicted in Table 4.4. In the following, we elaborate on each step of our proposed guidelines to estimate and assess recursive models containing composites of composites. In doing so, the two-stage approach is employed for model estimation based on a saturated structural model in the first stage.

Table 4.4: Guidelines for estimating and assessing models containing second-order composites built by composites



## Stage 1

In the first stage, as suggested by the two-stage approach, the model is estimated without second-order composites. The goal of this stage is to calculate the construct scores and to assess the operationalization of the concepts represented in the structural model. If all concepts are modeled as composites, this stage largely resembles a CCA.

### **Step 1: Estimating the model without second-order composites**

In Step 1, the model without the second-order composites is estimated. In doing so, a model with a saturated structural model should be used. For concepts modeled as common factors, the use of correlation weights (mode A) in combination with a correction for attenuation is recommended, i.e., PLSc, while for concepts conceptualized as composites, regression weights (mode B) should be employed.

### **Step 2: Assessing the overall model fit**

Since the model is estimated, it is necessary to evaluate whether it makes sense to continue forming second-order composites from the first-order composites. Thus, the fit of the estimated model from Step 1 needs to be assessed to obtain empirical evidence for the operationalization of the concepts. For this purpose, as our simulation has shown, the bootstrap-based test based on a discrepancy measure – such as the SRMR – can be used (see Section 4.3). For decision making, the 95% or 99% quantiles should be used as critical values, i.e., the value of the discrepancy measure based on the original dataset is compared to the 95% or 99% quantiles of its reference distribution, to decide whether the null hypothesis that the indicator population covariance matrix equals the model-implied counterpart is rejected. Next to the tests for exact overall model fit, fit measures can be employed to judge the fit of the model. Yet, fit measures for composite models have yet not been properly examined, therefore they should be used cautiously (Hair et al., 2019b). For guidelines on assessing models estimated by PLS-PM in the context of explanatory and confirmatory research, refer to Benitez et al. (2020).

### **Step 3: Extracting construct scores**

Once the estimated model from Step 1 has shown acceptable fit, the construct scores are extracted for the first-order composites and the constructs not belonging to a second-order composite. The standardized scores are appended as new variables to the file containing the observations of the original variables.

## **Stage 2**

In the second stage, the second-order construct is specified in the model by means of a second-order composite. Moreover, first-order composites and the blocks of indicators of constructs not belonging to a second-order composite are replaced by the corresponding construct scores. The purpose of this stage is to obtain consistent estimates for the parameters of the structural model, i.e., the path coefficients and the weights of the first-order composites used to build the second-order composite. Moreover, the overall fit of the structural model is assessed.

### **Step 4: Estimating the structural model containing second-order composites**

In Step 4, the second-order composite is included in the structural model and the first-order composites are replaced by their construct scores. In doing so, the construct scores of the first-order composites are utilized as indicators to build the second-order composite. Similarly, the indicators of the other constructs are also replaced by their construct scores, i.e., the construct become single-indicator constructs. For the estimation of the weights used to build the second-order composite, the use of mode B is recommended. However, in some situations the use of predefined weights or weights obtained by other techniques can be beneficial ([Dijkstra, 2013b, 2016](#)).

### **Step 5: Assessing model fit**

In the last step, which is similar to Step 3, the overall model fit needs to be assessed to obtain empirical support for the structural model containing composites of composites. In other words, we investigate whether the structural model is misspecified, and whether it is useful at all to build a second-order composite from

the first-order composites. Supported by our simulation results, for this purpose the bootstrap-based test of exact overall model fit from Section 4.3 can be used again. As in Step 2, fit measures should be used cautiously as they are hardly explored in the context of composite models.

## 4.6 Discussion and future research

Supported by the results of our Monte Carlo simulation, the study at hand introduces step-by-step user-guidelines embracing the two-stage approach and a two-step testing procedure on estimating and statistically assessing models containing composites of composites in the context of PLS-PM. This enables researchers from the field of IS to appropriately model artifacts which are again built by human-made objects and to obtain empirical evidence for the specified model. The latter is particularly important in causal research, i.e., explanatory and confirmatory research, as it supports the researcher's postulated theory. While the study at hand limits the use of the composite to model artifacts, it is noted that the composite model can also be used to operationalize behavioral concepts (Rigdon, 2016; Rigdon et al., 2017, 2019). This notion assumes that both composites and common factors serve as proxy for the behavioral concept (Sarstedt et al., 2016; Hair and Sarstedt, 2019). Moreover, recent research identified instances in which the composite model appears to have advantages over the common factor model (Rhemtulla et al., 2020). These findings are promising and do not limit our guidelines; on contrary, they extend their application.

The need for updated guidelines for models containing composites of composites in the context of explanatory and confirmatory research, is emphasized by our simulation design. It shows that existing guidelines, which neglect overall model fit assessment and focus mainly on measurement quality evaluation (e.g., Sarstedt et al., 2019; Hair et al., 2020, 2019a, 2017b,a; Ramayah et al., 2016; Becker et al., 2012), are unable to distinguish correctly from incorrectly specified models. These guidelines ignore a crucial cornerstone of SEM, namely to investigate whether the specified model fits to the collected data and as a consequence an important step to detect potential weaknesses in the model is neglected. Finally, it has been suggested to evaluate the predictive power of models estimated by PLS (Carrión et al., 2016; Shmueli et al., 2019). Although we generally agree that knowledge about the

predictive capacities of a model can provide additional insights, it cannot replace the assessment steps of causal modeling involving model fit assessment. This is highlighted by the fact that wrong models can predict well and that well-specified models can predict poorly ([Shmueli, 2010](#)). Therefore, guidelines in the context of causal research that currently ignore overall model fit assessment, should be extended by it.

To support our proposed guidelines and to further shed light on the performance of the commonly applied approaches in the context of PLS-PM to estimate models containing composites of composites, a Monte Carlo simulation was conducted. The results of our simulation showed that for finite sample sizes none of the approaches considered is superior in terms of estimated bias and RMSE. For most approaches, the average parameter estimates converged to the population counterpart and the standard errors of the estimates decreased for an increasing sample size. However, for the repeated indicators approach, we observed a slightly different behavior compared to the findings of previous simulation studies. While for example [Becker et al. \(2012\)](#) conclude that the path weighting scheme is the preferable option for the repeated indicators approach, we observed in our simulation that this inner weighting scheme produced more biased first-order composite weight estimates compared to the centroid and the factorial inner weighting scheme. Overall, only the two-stage approach based on a saturated structural model in the first stage is Fisher consistent, i.e., it retrieves the population parameters when the indicator population covariance matrix is used as input. Moreover, its results are hardly affected by the choice of the inner weighting scheme and it produced a low share of non-converged estimations. Additionally, the two-stage approach's way to estimate models containing composites of composites has advantages over the other two approaches in terms of model testing as the source of misspecification can be located. Therefore, for models containing a composite of composites, we recommend to use the two-stage approach based on a saturated structural model in the first stage.

Similarly to investigate the efficacy of two presented testing procedures, namely the complete testing and the two-step testing, a Monte Carlo simulation was conducted. The results showed that, particularly for small sample sizes, both testing procedures produce rejection rates slightly off the predefined significance. However, for an increasing number of observations, the rejection rates converge to the predefined

significance level. Moreover, both tests are capable to reliably detect misspecifications, except for small misspecification and a small sample size. In this case, the null hypothesis was rejected too rarely.

Since a simulation study is always limited to its design, modified replications and extensions are always helpful. For instance, researchers could vary distributional assumption and/or model complexity to broaden the understanding of the behavior of the different approaches. Particularly, for the repeated indicators approach more simulation studies would be beneficial, as our study observed a different behavior – compared to previous studies (e.g., [Becker et al., 2012](#)) – in terms of which inner PLS-PM weighting scheme should be statistically preferable.

Similarly, future research should strive for explanations why the repeated indicators approach and its modifications are not Fisher consistent to mathematically support our findings. Moreover, the performance of the approaches discussed in [Section 4.2](#) based on weights obtained by PLS-PM could also be assessed using a different weight estimators such as generalized structured component analysis (GSCA). Considering the testing procedures, our simulation study is limited to misspecifications in the structural model. However, there is a simulation study showing that the test is also capable to investigate misspecifications in the composite models ([Schuberth et al., 2018a](#)). This provides a starting point to further investigate the testing procedures' behavior for misspecifications in the structural model and the concepts' operationalization.

Moreover, we investigate the approaches' performance only for models containing second-order constructs specified as composites of composites. However, second-order constructs specified as a mixture of different constructs are conceivable. Although to our knowledge, this type of second-order constructs is not applied yet in the literature, it may open new avenues for modeling theories in IS and beyond. In particular, disciplines dealing with both behavioral concepts and artifacts, may benefit from its examination. Potential candidates are second-order constructs specified as a common factor of a mixture of composites and common factors and second-order constructs specified as a composite of a mixture of composites and common factors. An investigation of the performance of the commonly employed approaches and the assessment of such models would further contribute to the literature.

## 4.7 Appendix A to Chapter 4

Table 4.5: Indicator population covariance matrix of the baseline model

$y_{11}$	$y_{12}$	$y_{21}$	$y_{22}$	$y_{23}$	$y_{24}$	$y_{31}$	$y_{32}$	$y_{33}$	$y_{34}$	$y_{35}$	$y_{36}$	$y_{41}$	$y_{42}$	$y_{43}$	$y_{51}$	$y_{52}$	$y_{53}$	$y_{61}$	$y_{62}$	$y_{63}$
1.00	0.31	0.36	0.31	0.25	0.32	0.13	0.16	0.13	0.14	0.15	0.15	0.19	0.18	0.18	0.16	0.22	0.27	0.02	0.01	0.01
0.31	1.00	0.26	0.22	0.18	0.23	0.09	0.11	0.09	0.10	0.11	0.10	0.14	0.12	0.12	0.12	0.15	0.19	0.01	0.01	0.01
0.36	0.26	1.00	0.40	0.30	0.31	0.18	0.21	0.18	0.19	0.20	0.20	0.19	0.17	0.17	0.16	0.21	0.26	0.02	0.01	0.01
0.31	0.22	0.40	1.00	0.28	0.31	0.15	0.18	0.15	0.16	0.17	0.17	0.16	0.14	0.14	0.13	0.18	0.22	0.01	0.01	0.01
0.25	0.18	0.30	0.28	1.00	0.30	0.12	0.15	0.12	0.13	0.14	0.14	0.13	0.12	0.12	0.11	0.14	0.18	0.01	0.01	0.01
0.32	0.23	0.31	0.31	0.30	1.00	0.16	0.19	0.15	0.16	0.18	0.17	0.16	0.15	0.15	0.14	0.18	0.23	0.01	0.01	0.01
0.13	0.09	0.18	0.15	0.12	0.16	1.00	0.10	0.25	0.13	0.10	0.30	0.12	0.11	0.11	0.10	0.13	0.17	0.01	0.01	0.01
0.16	0.11	0.21	0.18	0.15	0.19	0.10	1.00	0.20	0.40	0.30	0.20	0.14	0.13	0.13	0.12	0.16	0.20	0.01	0.01	0.01
0.13	0.09	0.18	0.15	0.12	0.15	0.25	0.20	1.00	0.30	0.10	0.30	0.12	0.11	0.11	0.10	0.13	0.17	0.01	0.01	0.01
0.14	0.10	0.19	0.16	0.13	0.16	0.13	0.40	0.30	1.00	0.20	0.20	0.13	0.11	0.11	0.11	0.14	0.18	0.01	0.01	0.01
0.15	0.11	0.20	0.17	0.14	0.18	0.10	0.30	0.10	0.20	1.00	0.10	0.14	0.12	0.12	0.12	0.15	0.19	0.01	0.01	0.01
0.15	0.10	0.20	0.17	0.14	0.17	0.30	0.20	0.30	0.20	0.10	1.00	0.13	0.12	0.12	0.11	0.15	0.19	0.01	0.01	0.01
0.19	0.14	0.19	0.16	0.13	0.16	0.12	0.14	0.12	0.13	0.14	0.13	1.00	0.40	0.40	0.09	0.12	0.15	-0.29	-0.18	-0.22
0.18	0.12	0.17	0.14	0.12	0.15	0.11	0.13	0.11	0.11	0.12	0.12	0.40	1.00	0.40	0.08	0.11	0.13	-0.26	-0.17	-0.20
0.18	0.12	0.17	0.14	0.12	0.15	0.11	0.13	0.11	0.11	0.12	0.12	0.40	0.40	1.00	0.08	0.11	0.13	-0.26	-0.17	-0.20
0.16	0.12	0.16	0.13	0.11	0.14	0.10	0.12	0.10	0.11	0.12	0.11	0.09	0.08	0.08	1.00	0.20	0.30	-0.04	-0.02	-0.03
0.22	0.15	0.21	0.18	0.14	0.18	0.13	0.16	0.13	0.14	0.15	0.15	0.12	0.11	0.11	0.20	1.00	0.40	-0.05	-0.03	-0.04
0.27	0.19	0.26	0.22	0.18	0.23	0.17	0.20	0.17	0.18	0.19	0.19	0.15	0.13	0.13	0.30	0.40	1.00	-0.06	-0.04	-0.05
0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.29	-0.26	-0.26	-0.04	-0.05	-0.06	1.00	0.25	0.40
0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.18	-0.17	-0.17	-0.02	-0.03	-0.04	0.25	1.00	0.16
0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.22	-0.20	-0.20	-0.03	-0.04	-0.05	0.40	0.16	1.00



Table 4.6: Indicator population covariance matrix in case of the small model misspecification

$y_{11}$	$y_{12}$	$y_{21}$	$y_{22}$	$y_{23}$	$y_{24}$	$y_{31}$	$y_{32}$	$y_{33}$	$y_{34}$	$y_{35}$	$y_{36}$	$y_{41}$	$y_{42}$	$y_{43}$	$y_{51}$	$y_{52}$	$y_{53}$	$y_{61}$	$y_{62}$	$y_{63}$
1.00	0.31	0.22	0.18	0.15	0.19	0.11	0.13	0.11	0.11	0.12	0.12	0.17	0.15	0.15	0.07	0.09	0.12	0.27	0.17	0.20
0.31	1.00	0.15	0.13	0.10	0.13	0.08	0.09	0.07	0.08	0.09	0.08	0.12	0.11	0.11	0.05	0.06	0.08	0.19	0.12	0.14
0.22	0.15	1.00	0.40	0.30	0.31	0.15	0.18	0.15	0.16	0.17	0.17	-0.03	-0.02	-0.02	0.20	0.27	0.34	0.37	0.23	0.27
0.18	0.13	0.40	1.00	0.28	0.31	0.13	0.15	0.13	0.13	0.15	0.14	-0.02	-0.02	-0.02	0.17	0.23	0.29	0.31	0.20	0.23
0.15	0.10	0.30	0.28	1.00	0.30	0.10	0.12	0.10	0.11	0.12	0.12	-0.02	-0.02	-0.02	0.14	0.18	0.23	0.26	0.16	0.19
0.19	0.13	0.31	0.31	0.30	1.00	0.13	0.16	0.13	0.14	0.15	0.15	-0.02	-0.02	-0.02	0.18	0.24	0.30	0.33	0.20	0.24
0.11	0.08	0.15	0.13	0.10	0.13	1.00	0.10	0.25	0.13	0.10	0.30	-0.03	-0.02	-0.02	0.05	0.07	0.09	0.21	0.13	0.16
0.13	0.09	0.18	0.15	0.12	0.16	0.10	1.00	0.20	0.40	0.30	0.20	-0.03	-0.03	-0.03	0.06	0.08	0.10	0.25	0.16	0.18
0.11	0.07	0.15	0.13	0.10	0.13	0.25	0.20	1.00	0.30	0.10	0.30	-0.03	-0.02	-0.02	0.05	0.07	0.08	0.21	0.13	0.15
0.11	0.08	0.16	0.13	0.11	0.14	0.13	0.40	0.30	1.00	0.20	0.20	-0.03	-0.03	-0.03	0.05	0.07	0.09	0.22	0.14	0.16
0.12	0.09	0.17	0.15	0.12	0.15	0.10	0.30	0.10	0.20	1.00	0.10	-0.03	-0.03	-0.03	0.06	0.08	0.10	0.24	0.15	0.18
0.12	0.08	0.17	0.14	0.12	0.15	0.30	0.20	0.30	0.20	0.10	1.00	-0.03	-0.03	-0.03	0.06	0.08	0.10	0.23	0.15	0.17
0.17	0.12	-0.03	-0.02	-0.02	-0.02	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	1.00	0.40	0.40	0.09	0.12	0.15	-0.29	-0.18	-0.22
0.15	0.11	-0.02	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02	-0.03	-0.03	-0.03	0.40	1.00	0.40	0.08	0.11	0.13	-0.26	-0.17	-0.20
0.15	0.11	-0.02	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02	-0.03	-0.03	-0.03	0.40	0.40	1.00	0.08	0.11	0.13	-0.26	-0.17	-0.20
0.07	0.05	0.20	0.17	0.14	0.18	0.05	0.06	0.05	0.05	0.06	0.06	0.09	0.08	0.08	1.00	0.20	0.30	-0.04	-0.02	-0.03
0.09	0.06	0.27	0.23	0.18	0.24	0.07	0.08	0.07	0.07	0.08	0.08	0.12	0.11	0.11	0.20	1.00	0.40	-0.05	-0.03	-0.04
0.12	0.08	0.34	0.29	0.23	0.30	0.09	0.10	0.08	0.09	0.10	0.10	0.15	0.13	0.13	0.30	0.40	1.00	-0.06	-0.04	-0.05
0.27	0.19	0.37	0.31	0.26	0.33	0.21	0.25	0.21	0.22	0.24	0.23	-0.29	-0.26	-0.26	-0.04	-0.05	-0.06	1.00	0.25	0.40
0.17	0.12	0.23	0.20	0.16	0.20	0.13	0.16	0.13	0.14	0.15	0.15	-0.18	-0.17	-0.17	-0.02	-0.03	-0.04	0.25	1.00	0.16
0.20	0.14	0.27	0.23	0.19	0.24	0.16	0.18	0.15	0.16	0.18	0.17	-0.22	-0.20	-0.20	-0.03	-0.04	-0.05	0.40	0.16	1.00

Table 4.7: Indicator population covariance matrix in case of the medium model misspecification

$y_{11}$	$y_{12}$	$y_{21}$	$y_{22}$	$y_{23}$	$y_{24}$	$y_{31}$	$y_{32}$	$y_{33}$	$y_{34}$	$y_{35}$	$y_{36}$	$y_{41}$	$y_{42}$	$y_{43}$	$y_{51}$	$y_{52}$	$y_{53}$	$y_{61}$	$y_{62}$	$y_{63}$
1.00	0.31	0.24	0.20	0.17	0.21	0.10	0.12	0.10	0.11	0.12	0.11	-0.44	-0.40	-0.40	-0.01	-0.01	-0.02	0.54	0.34	0.40
0.31	1.00	0.17	0.14	0.12	0.15	0.07	0.08	0.07	0.08	0.08	0.08	-0.31	-0.28	-0.28	-0.01	-0.01	-0.01	0.38	0.24	0.28
0.24	0.17	1.00	0.40	0.30	0.31	0.15	0.18	0.15	0.16	0.17	0.17	-0.03	-0.02	-0.02	0.20	0.27	0.34	0.37	0.23	0.27
0.20	0.14	0.40	1.00	0.28	0.31	0.13	0.15	0.12	0.13	0.14	0.14	-0.02	-0.02	-0.02	0.17	0.23	0.29	0.31	0.20	0.23
0.17	0.12	0.30	0.28	1.00	0.30	0.10	0.12	0.10	0.11	0.12	0.11	-0.02	-0.02	-0.02	0.14	0.18	0.23	0.26	0.16	0.19
0.21	0.15	0.31	0.31	0.30	1.00	0.13	0.16	0.13	0.14	0.15	0.15	-0.02	-0.02	-0.02	0.18	0.24	0.30	0.33	0.20	0.24
0.10	0.07	0.15	0.13	0.10	0.13	1.00	0.10	0.25	0.13	0.10	0.30	0.02	0.02	0.02	0.06	0.08	0.10	0.19	0.12	0.14
0.12	0.08	0.18	0.15	0.12	0.16	0.10	1.00	0.20	0.40	0.30	0.20	0.02	0.02	0.02	0.07	0.09	0.11	0.23	0.14	0.17
0.10	0.07	0.15	0.12	0.10	0.13	0.25	0.20	1.00	0.30	0.10	0.30	0.02	0.02	0.02	0.06	0.07	0.09	0.19	0.12	0.14
0.11	0.08	0.16	0.13	0.11	0.14	0.13	0.40	0.30	1.00	0.20	0.20	0.02	0.02	0.02	0.06	0.08	0.10	0.20	0.13	0.15
0.12	0.08	0.17	0.14	0.12	0.15	0.10	0.30	0.10	0.20	1.00	0.10	0.02	0.02	0.02	0.07	0.09	0.11	0.22	0.14	0.16
0.11	0.08	0.17	0.14	0.11	0.15	0.30	0.20	0.30	0.20	0.10	1.00	0.02	0.02	0.02	0.06	0.08	0.11	0.21	0.13	0.16
-0.44	-0.31	-0.03	-0.02	-0.02	-0.02	0.02	0.02	0.02	0.02	0.02	0.02	1.00	0.40	0.40	0.09	0.12	0.15	-0.29	-0.18	-0.22
-0.40	-0.28	-0.02	-0.02	-0.02	-0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.40	1.00	0.40	0.08	0.11	0.13	-0.26	-0.17	-0.20
-0.40	-0.28	-0.02	-0.02	-0.02	-0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.40	0.40	1.00	0.08	0.11	0.13	-0.26	-0.17	-0.20
-0.01	-0.01	0.20	0.17	0.14	0.18	0.06	0.07	0.06	0.06	0.07	0.06	0.09	0.08	0.08	1.00	0.20	0.30	-0.04	-0.02	-0.03
-0.01	-0.01	0.27	0.23	0.18	0.24	0.08	0.09	0.07	0.08	0.09	0.08	0.12	0.11	0.11	0.20	1.00	0.40	-0.05	-0.03	-0.04
-0.02	-0.01	0.34	0.29	0.23	0.30	0.10	0.11	0.09	0.10	0.11	0.11	0.15	0.13	0.13	0.30	0.40	1.00	-0.06	-0.04	-0.05
0.54	0.38	0.37	0.31	0.26	0.33	0.19	0.23	0.19	0.20	0.22	0.21	-0.29	-0.26	-0.26	-0.04	-0.05	-0.06	1.00	0.25	0.40
0.34	0.24	0.23	0.20	0.16	0.20	0.12	0.14	0.12	0.13	0.14	0.13	-0.18	-0.17	-0.17	-0.02	-0.03	-0.04	0.25	1.00	0.16
0.40	0.28	0.27	0.23	0.19	0.24	0.14	0.17	0.14	0.15	0.16	0.16	-0.22	-0.20	-0.20	-0.03	-0.04	-0.05	0.40	0.16	1.00

Table 4.8: Indicator population covariance matrix in case of the large model misspecification

$y_{11}$	$y_{12}$	$y_{21}$	$y_{22}$	$y_{23}$	$y_{24}$	$y_{31}$	$y_{32}$	$y_{33}$	$y_{34}$	$y_{35}$	$y_{36}$	$y_{41}$	$y_{42}$	$y_{43}$	$y_{51}$	$y_{52}$	$y_{53}$	$y_{61}$	$y_{62}$	$y_{63}$
1.00	0.31	0.27	0.23	0.19	0.24	0.05	0.06	0.05	0.06	0.06	0.06	0.17	0.15	0.15	0.07	0.09	0.12	0.27	0.17	0.20
0.31	1.00	0.19	0.16	0.13	0.17	0.04	0.04	0.04	0.04	0.04	0.04	0.12	0.11	0.11	0.05	0.06	0.08	0.19	0.12	0.14
0.27	0.19	1.00	0.40	0.30	0.31	0.06	0.07	0.06	0.06	0.07	0.07	0.23	0.20	0.20	0.19	0.26	0.32	0.26	0.16	0.19
0.23	0.16	0.40	1.00	0.28	0.31	0.05	0.06	0.05	0.05	0.06	0.06	0.19	0.17	0.17	0.16	0.22	0.27	0.22	0.14	0.16
0.19	0.13	0.30	0.28	1.00	0.30	0.04	0.05	0.04	0.04	0.05	0.05	0.16	0.14	0.14	0.13	0.18	0.22	0.18	0.11	0.13
0.24	0.17	0.31	0.31	0.30	1.00	0.05	0.06	0.05	0.06	0.06	0.06	0.20	0.18	0.18	0.17	0.23	0.28	0.23	0.14	0.17
0.05	0.04	0.06	0.05	0.04	0.05	1.00	0.10	0.25	0.13	0.10	0.30	-0.25	-0.23	-0.23	0.02	0.03	0.04	0.31	0.19	0.23
0.06	0.04	0.07	0.06	0.05	0.06	0.10	1.00	0.20	0.40	0.30	0.20	-0.29	-0.26	-0.26	0.03	0.04	0.05	0.36	0.23	0.27
0.05	0.04	0.06	0.05	0.04	0.05	0.25	0.20	1.00	0.30	0.10	0.30	-0.24	-0.22	-0.22	0.02	0.03	0.04	0.30	0.19	0.22
0.06	0.04	0.06	0.05	0.04	0.06	0.13	0.40	0.30	1.00	0.20	0.20	-0.26	-0.24	-0.24	0.02	0.03	0.04	0.32	0.20	0.24
0.06	0.04	0.07	0.06	0.05	0.06	0.10	0.30	0.10	0.20	1.00	0.10	-0.28	-0.26	-0.26	0.03	0.03	0.04	0.35	0.22	0.26
0.06	0.04	0.07	0.06	0.05	0.06	0.30	0.20	0.30	0.20	0.10	1.00	-0.27	-0.25	-0.25	0.03	0.03	0.04	0.34	0.21	0.25
0.17	0.12	0.23	0.19	0.16	0.20	-0.25	-0.29	-0.24	-0.26	-0.28	-0.27	1.00	0.40	0.40	0.09	0.12	0.15	-0.29	-0.18	-0.22
0.15	0.11	0.20	0.17	0.14	0.18	-0.23	-0.26	-0.22	-0.24	-0.26	-0.25	0.40	1.00	0.40	0.08	0.11	0.13	-0.26	-0.17	-0.20
0.15	0.11	0.20	0.17	0.14	0.18	-0.23	-0.26	-0.22	-0.24	-0.26	-0.25	0.40	0.40	1.00	0.08	0.11	0.13	-0.26	-0.17	-0.20
0.07	0.05	0.19	0.16	0.13	0.17	0.02	0.03	0.02	0.02	0.03	0.03	0.09	0.08	0.08	1.00	0.20	0.30	-0.04	-0.02	-0.03
0.09	0.06	0.26	0.22	0.18	0.23	0.03	0.04	0.03	0.03	0.03	0.03	0.12	0.11	0.11	0.20	1.00	0.40	-0.05	-0.03	-0.04
0.12	0.08	0.32	0.27	0.22	0.28	0.04	0.05	0.04	0.04	0.04	0.04	0.15	0.13	0.13	0.30	0.40	1.00	-0.06	-0.04	-0.05
0.27	0.19	0.26	0.22	0.18	0.23	0.31	0.36	0.30	0.32	0.35	0.34	-0.29	-0.26	-0.26	-0.04	-0.05	-0.06	1.00	0.25	0.40
0.17	0.12	0.16	0.14	0.11	0.14	0.19	0.23	0.19	0.20	0.22	0.21	-0.18	-0.17	-0.17	-0.02	-0.03	-0.04	0.25	1.00	0.16
0.20	0.14	0.19	0.16	0.13	0.17	0.23	0.27	0.22	0.24	0.26	0.25	-0.22	-0.20	-0.20	-0.03	-0.04	-0.05	0.40	0.16	1.00

Table 4.9: Approaches' Fisher consistency - Part I

Parameter	Centroid scheme			Factorial scheme			Path scheme		
	two-stage	embedded TS	extended RI	two-stage	embedded TS	extended RI	two-stage	embedded TS	extended RI
$\gamma_1$	-3.89e-16	-3.33e-16	-3.61e-16	-6.66e-16	-6.94e-16	-6.94e-16	-6.94e-16	-6.94e-16	-6.94e-16
$\gamma_2$	6.66e-16	6.11e-16	5.55e-16	-2.22e-16	-1.11e-16	-2.22e-16	-1.11e-16	-1.11e-16	-1.67e-16
$\gamma_3$	-7.77e-16	-1.98e-11	-3.50e-01	-1.67e-16	-1.26e-11	-3.50e-01	0.00e+00	-9.16e-11	-3.50e-01
$\beta_1$	5.00e-16	-2.27e-11	-4.00e-01	2.78e-16	-1.44e-11	-4.00e-01	2.78e-16	-1.05e-10	-4.00e-01
$\beta_2$	-1.39e-16	-1.13e-11	-2.00e-01	-8.33e-17	-7.22e-12	-2.00e-01	5.55e-17	-5.24e-11	-2.00e-01
$w_{11}$	7.77e-16	-1.24e-05	-6.21e-06	5.55e-16	-8.52e-06	-5.33e-06	0.00e+00	-4.58e-10	-1.35e-09
$w_{12}$	-8.33e-16	1.77e-05	8.83e-06	-4.44e-16	1.21e-05	7.59e-06	0.00e+00	6.52e-10	1.92e-09
$w_{21}$	-3.33e-16	-4.20e-06	-2.80e-06	3.89e-16	-7.99e-06	-5.29e-06	9.99e-16	-3.14e-07	-3.77e-07
$w_{22}$	5.00e-16	1.98e-06	1.32e-06	4.44e-16	3.78e-06	2.50e-06	1.67e-16	1.48e-07	1.78e-07
$w_{23}$	-5.55e-16	5.07e-06	3.39e-06	-5.83e-16	9.66e-06	6.39e-06	-1.30e-15	3.80e-07	4.56e-07
$w_{24}$	-5.55e-17	-1.11e-06	-7.40e-07	-3.89e-16	-2.11e-06	-1.39e-06	-3.33e-16	-8.29e-08	-9.94e-08
$w_{31}$	-3.33e-16	-1.28e-06	-4.71e-07	-2.22e-15	-9.20e-07	-3.60e-07	-8.33e-16	-3.96e-06	-1.46e-06
$w_{32}$	-1.67e-16	-1.28e-06	-4.71e-07	1.61e-15	-9.20e-07	-3.60e-07	1.11e-15	-3.96e-06	-1.46e-06
$w_{33}$	1.11e-16	8.07e-06	2.96e-06	-8.33e-16	5.78e-06	2.26e-06	-1.30e-15	2.48e-05	9.19e-06
$w_{34}$	2.66e-15	8.07e-06	2.96e-06	3.61e-15	5.78e-06	2.26e-06	1.17e-15	2.48e-05	9.19e-06
$w_{35}$	-1.17e-15	-1.06e-05	-3.90e-06	-1.39e-15	-7.62e-06	-2.98e-06	2.78e-16	-3.28e-05	-1.21e-05
$w_{36}$	-1.17e-15	-1.28e-06	-4.71e-07	-8.88e-16	-9.20e-07	-3.60e-07	-4.44e-16	-3.96e-06	-1.46e-06
$w_{41}$	4.44e-16	4.44e-16	2.22e-16	3.33e-16	6.66e-16	0.00e+00	-5.55e-16	-3.33e-16	-6.66e-16

Note: embedded TS: embedded two-stage approach; extended RI: extended repeated indicators approach;

two-stage: two-stage approach based on a saturated structural model in the first stage;

$\kappa_1$ : the total effect of  $\xi$  on  $\eta_3$ ;  $\kappa_2$ : the total effect of  $\eta_1$  on  $\eta_3$ ;  $\kappa_3$ : the total effect of  $\eta_2$  on  $\eta_3$ .

Table 4.10: Approaches' Fisher consistency - Part II

Parameter	Centroid scheme			Factorial scheme			Path scheme		
	two-stage	embedded TS	extended RI	two-stage	embedded TS	extended RI	two-stage	embedded TS	extended RI
$w_{42}$	-5.00e-16	-3.33e-16	-6.11e-16	6.11e-16	5.55e-17	5.00e-16	-6.11e-16	-9.44e-16	-7.22e-16
$w_{43}$	1.11e-16	1.67e-16	6.66e-16	-8.88e-16	-8.88e-16	-5.55e-16	1.11e-15	1.28e-15	1.39e-15
$w_{51}$	2.00e-15	2.16e-15	1.67e-15	1.05e-15	1.33e-15	1.28e-15	7.77e-16	1.17e-15	1.11e-15
$w_{52}$	-2.16e-15	-1.28e-15	-1.94e-15	-1.44e-15	-1.28e-15	-1.94e-15	-2.11e-15	-1.89e-15	-2.05e-15
$w_{53}$	3.33e-16	-3.33e-16	2.22e-16	3.33e-16	-1.11e-16	6.66e-16	1.33e-15	7.77e-16	9.99e-16
$w_{61}$	-3.11e-15	-2.89e-15	-7.88e-15	1.11e-15	7.77e-16	9.99e-16	-1.11e-16	-6.66e-16	-3.33e-16
$w_{62}$	3.66e-15	3.89e-15	6.16e-15	-7.22e-16	-1.11e-16	-5.00e-16	-1.72e-15	-1.44e-15	-1.78e-15
$w_{63}$	1.61e-15	1.17e-15	6.16e-15	-7.77e-16	-4.44e-16	-5.55e-16	2.22e-15	2.61e-15	2.61e-15
$w_{c_1}$	-1.55e-15	-6.66e-11	-	6.66e-16	-1.75e-12	-	9.44e-16	2.18e-10	-
$w_{c_2}$	2.11e-15	1.39e-10	-	1.44e-15	3.05e-11	-	-1.67e-16	5.74e-10	-
$w_{c_3}$	-1.22e-15	-3.03e-11	-	-2.33e-15	5.66e-12	-	-1.11e-15	-6.17e-10	-
$\kappa_2$	5.00e-16	-2.27e-11	-5.09e-04	2.78e-16	-1.44e-11	-1.34e-03	2.78e-16	-1.05e-10	-3.63e-03
$\kappa_3$	-1.39e-16	-1.13e-11	-2.55e-04	-8.33e-17	-7.22e-12	-6.72e-04	5.55e-17	-5.24e-11	-1.81e-03
$\kappa_1$	-3.89e-16	-1.98e-11	-4.46e-04	-1.67e-16	-1.26e-11	-1.18e-03	0.00e+00	-9.16e-11	-3.17e-03

Note: embedded TS: embedded two-stage approach; extended RI: extended repeated indicators approach;  
two-stage: two-stage approach based on a saturated structural model in the first stage;  
 $\kappa_1$ : the total effect of  $\xi$  on  $\eta_3$ ;  $\kappa_2$ : the total effect of  $\eta_1$  on  $\eta_3$ ;  $\kappa_3$ : the total effect of  $\eta_2$  on  $\eta_3$ .

Table 4.11: Smallest weight, smallest loading, and largest VIF for the population models used in the context of model misspecification

	small misspecification									medium misspecification									large misspecification								
	TS			Emb. TS			Ext. RI			TS			Emb. TS			Ext. RI			TS			Emb. TS			Ext. RI		
	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$	$w$	VIF	$\lambda$
$\xi$	.39	1.9	.75	.39	1.9	.75	.39	1.9	.75	.39	1.9	.75	.39	1.9	.75	.39	1.9	.75	.39	1.9	.75	.39	1.9	.75	.39	1.9	.75
$\eta_1$	.26	1.8	.54	.26	1.8	.54	.26	1.8	.54	.26	1.8	.54	.26	1.8	.54	.26	1.8	.54	.26	1.8	.54	.26	1.8	.54	.26	1.8	.54
$\eta_2$	.33	1.8	.56	.33	1.8	.56	.33	1.8	.56	.33	1.8	.56	.33	1.8	.56	.33	1.8	.56	.33	1.8	.56	.33	1.8	.56	.33	1.8	.56
$c_1$	.40	1.5	.65	.40	1.5	.65	.40	1.5	.65	.40	1.5	.65	.40	1.5	.65	.40	1.5	.65	.40	1.5	.65	.40	1.5	.65	.40	1.5	.65
$c_2$	.20	1.9	.55	.20	1.9	.55	.20	1.9	.55	.20	1.9	.55	.20	1.9	.55	.20	1.9	.55	.20	1.9	.55	.20	1.9	.55	.20	1.9	.55
$c_3$	.20	1.9	.53	.20	1.9	.53	.20	1.9	.53	.20	1.9	.53	.20	1.9	.53	.20	1.9	.53	.20	1.9	.53	.20	1.9	.53	.20	1.9	.53
$\eta_3$	.27	1.7	.55	.27	1.7	.55	.27	1.7	.55	.27	1.8	.53	.22	1.8	.53	.27	2.7	.54	.27	2.7	.54	.27	2.7	.55	.27	2.7	.55

Note: values in columns  $w$  and  $\lambda$ , respectively, show the smallest weight and loading, respectively, among the indicators attached to the corresponding composite; Values in column VIF show the largest VIF value of the indicators' weight used to build the corresponding composite.

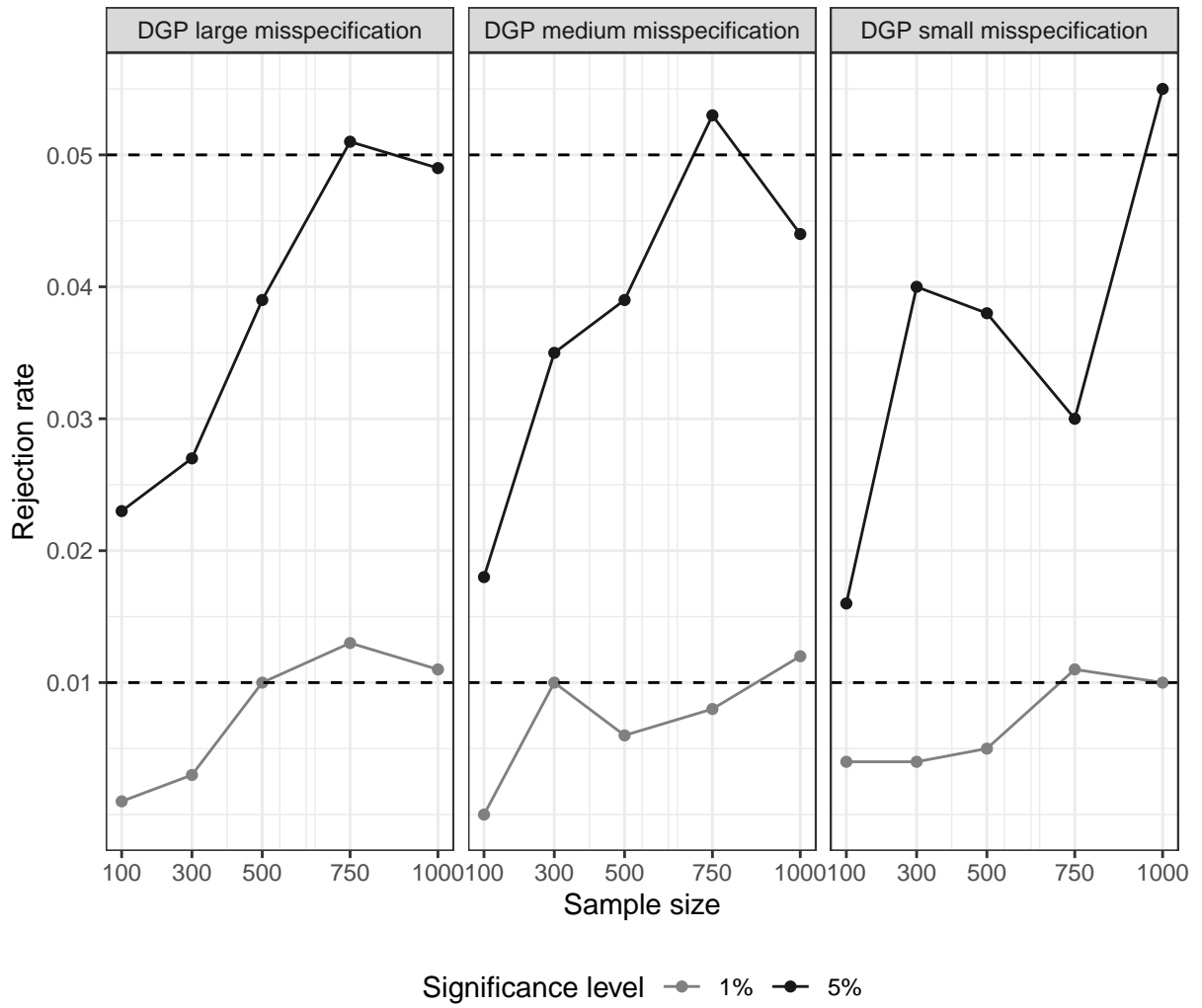


Figure 4.7: Tests' rejection rates for the alternative population models estimated by a correctly specified model

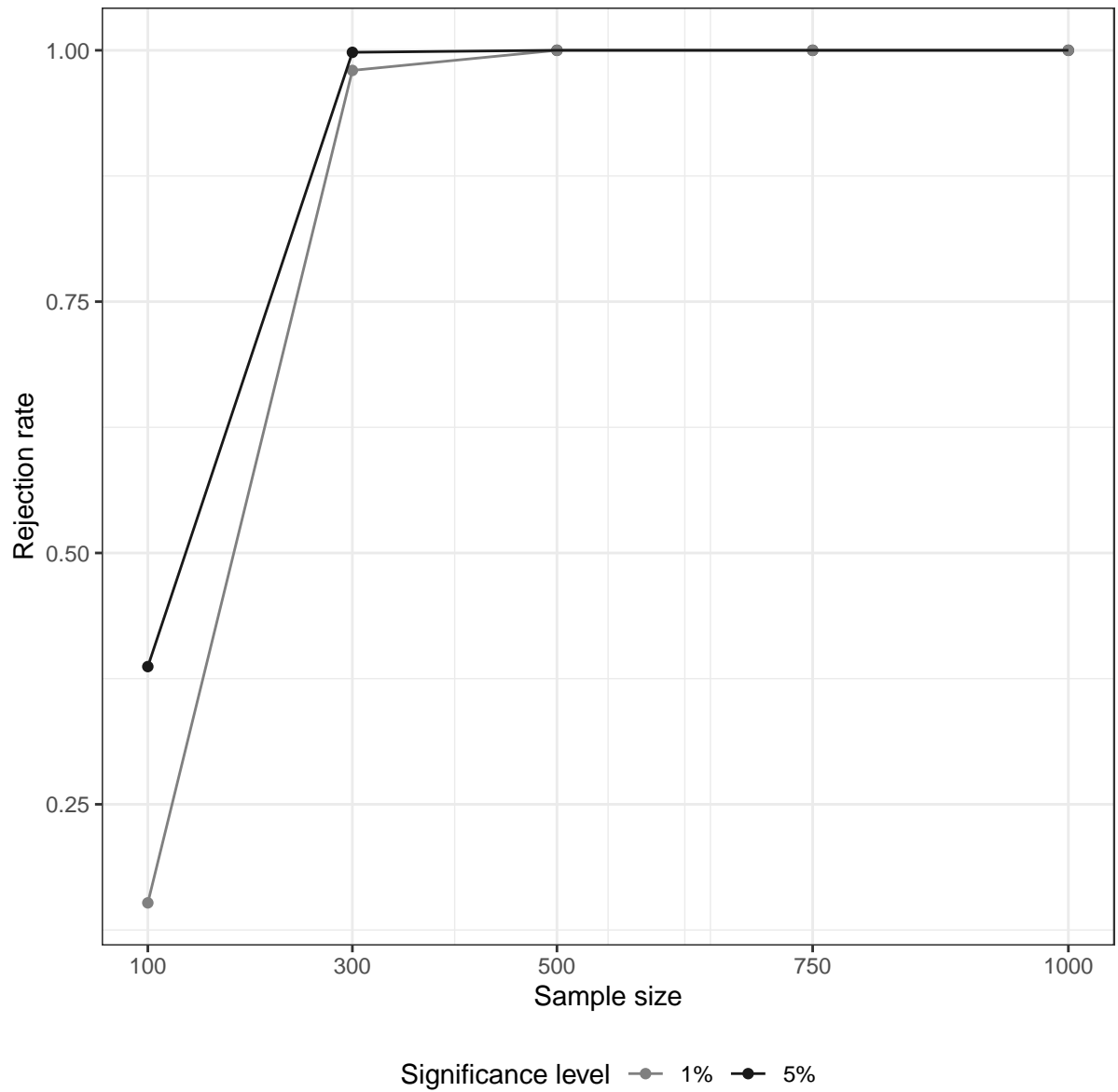


Figure 4.8: Tests' rejection rates for the baseline population model estimated by a specification that matches the alternative population models



## 4.8 Appendix B to Chapter 4

### 4.8.1 Data generation

We consider four different data generating processes (DGPs) in the paper:

- `dgp_2ndorder`
- `dgp_alt_small`
- `dgp_alt_med`
- `dgp_alt_large`

#### The second-order DGP (`dgp_2ndorder`)

This DGP includes the second-order construct  $\eta_3$ . The DGP is displayed in Figure 4.3 in the paper but repeated here for convenience.

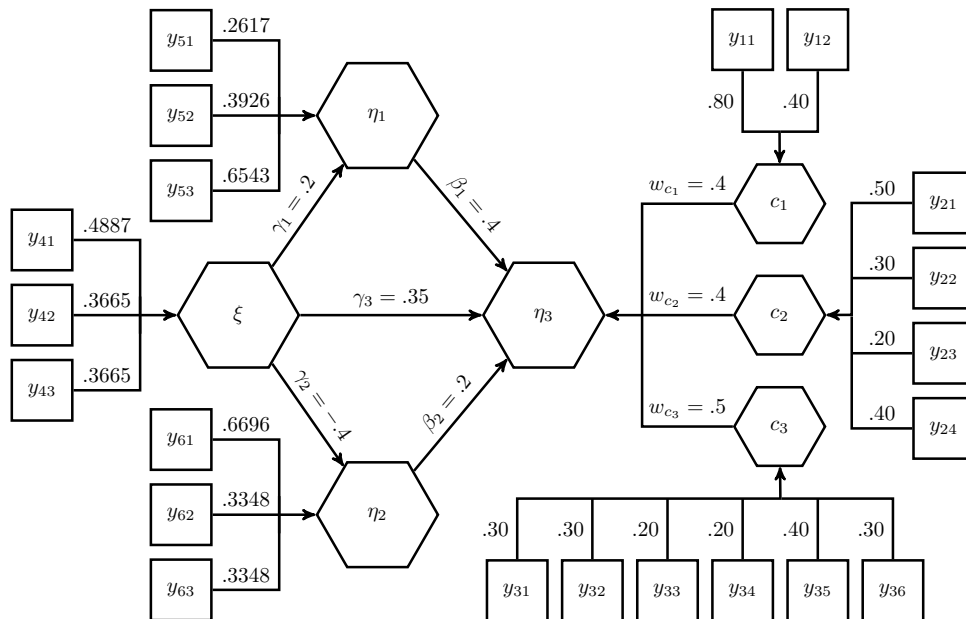


Figure 4.9: Second-order DGP

We use the `cSEM.DGP` package (Rademaker and Schamberger, 2020) to generate the data. The package was written by Tamara Schamberger and myself and is available on [GitHub](#). The main purpose of the package is to provide a quick way to generate model-implied population covariance matrices or data for structural equation models in R.

```

dgp_2ndorder <- "
## Structural model
eta1 ~ 0.2*xi
eta2 ~ -0.4*xi
eta3 ~ 0.35*xi + 0.4*eta1 + 0.2*eta2

## Composite models
xi <~ 0.8*y41 + 0.6*y42 + 0.6*y43
eta1 <~ 2*y51 + 3*y52 + 5*y53
eta2 <~ 2*y61 + 1*y62 + 1*y63
c1 <~ 0.8*y11 + 0.4*y12
c2 <~ 0.5*y21 + 0.3*y22 + 0.2*y23 + 0.4*y24
c3 <~ 0.3*y31 + 0.3*y32 + 0.2*y33 + 0.2*y34 + 0.4*y35 + 0.3*y36

## Second-order composite
eta3 <~ 0.4*c1 + 0.4*c2 + 0.5*c3

## Composite indicator correlations
# xi
y41 ~~ 0.4*y42 + 0.4*y43
y42 ~~ 0.4*y43

# eta1
y51 ~~ 0.2*y52 + 0.3*y53
y52 ~~ 0.4*y53

# eta2
y61 ~~ 0.25*y62 + 0.4*y63
y62 ~~ 0.16*y63

# eta3 (the 2nd order construct)
c1 ~~ 0.49*c2 + 0.27*c3
c2 ~~ 0.413*c3

# c1-c3
y11 ~~ 0.3125*y12

y21 ~~ 0.4*y22 + 0.3*y23 + 0.31*y24
y22 ~~ 0.28*y23 + 0.31*y24
y23 ~~ 0.3*y24

y31 ~~ 0.1*y32 + 0.25*y33 + 0.13*y34 + 0.1*y35 + 0.3*y36
y32 ~~ 0.2*y33 + 0.4*y34 + 0.3*y35 + 0.2*y36
y33 ~~ 0.3*y34 + 0.1*y35 + 0.3*y36
y34 ~~ 0.2*y35 + 0.2*y36
y35 ~~ 0.1*y36
"

```

```

## Generate data
dat <- generateData(dgp_2ndorder, .return_type = "data.frame", .empirical = TRUE)

## Estimate and summarize
out <- cSEM::summarize(csem(dat, dgp_2ndorder))

## Save the population parameters (to be able to compare them to the estimates)
pop_params_2ndorder <- list(
  "Loadings" = {
    l <- c(out$First_stage$Estimates>Loading_estimates$Estimate,
          out$Second_stage$Estimates>Loading_estimates$Estimate)
    names(l) <- c(out$First_stage$Estimates>Loading_estimates$Name,
                 out$Second_stage$Estimates>Loading_estimates$Name)
    l
  },
  "Path_coefficients" = {
    p <- out$Second_stage$Estimates$Path_estimates$Estimate
    names(p) <- out$Second_stage$Estimates$Path_estimates$Name
    p
  },
  "Weights" = {
    w <- c(out$First_stage$Estimates$Weight_estimates$Estimate,
          out$Second_stage$Estimates$Weight_estimates$Estimate)
    names(w) <- c(out$First_stage$Estimates$Weight_estimates$Name,
                 out$Second_stage$Estimates$Weight_estimates$Name)
    w
  }
)

```

## The alternative DGPs

The alternative DGPs are identical to the second-order DGP except for the structural model. Hence, the population weights and intra-block correlations are identical to those from the second-order DGP; however, unlike the second-order DGP, the constructs  $\xi$ ,  $\eta_1$ , and  $\eta_2$  directly affect the constructs  $c_1$ ,  $c_2$ , and  $c_3$  without there being a second-order construct. The structural model is shown in Figure 4.4 of the paper but repeated here for convenience.

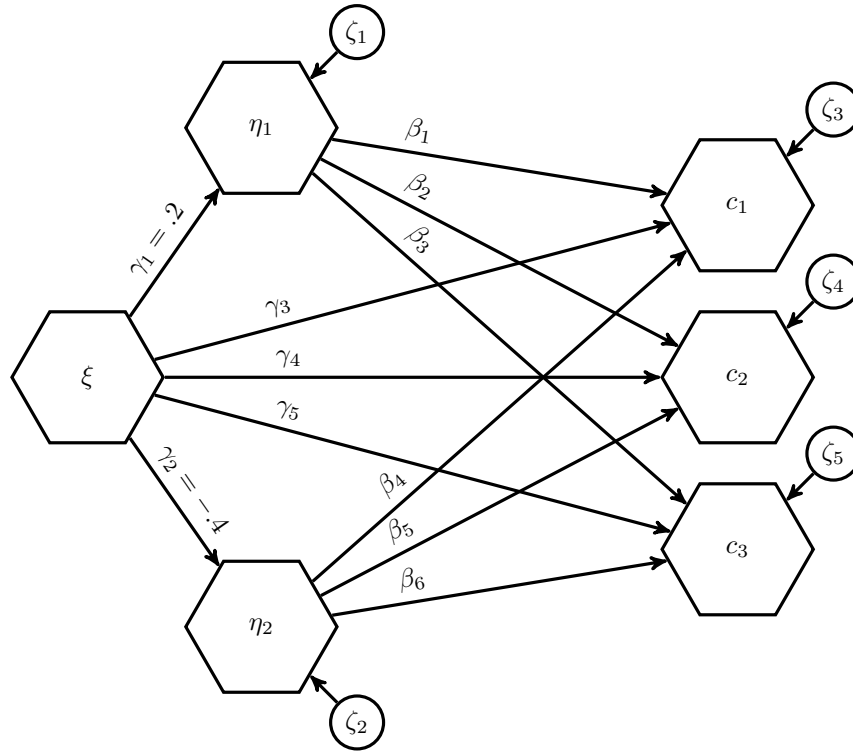


Figure 4.10: Structural specification for the alternative DGPs

To assess the power of the testing procedures proposed in the paper, three different alternative DGPs are considered. They only differ in the absolute magnitude of the direct effects of  $\xi$ ,  $\eta_1$ , and  $\eta_2$  on  $c_1$ ,  $c_2$ , and  $c_3$ . In the following, these DGPs are called:

- `dgp_alt_small`
- `dgp_alt_med`
- `dgp_alt_large`

The model that matches the second-order DGP (`model_2ndorder` in Subsection 4.8.3) is misspecified with respect to the alternative DGPs. The degree of misspecification is measured by means of the standardized root mean square residual (SRMR). Every test's power to detect misspecification is a function of the magnitude of misspecification; the larger the misspecification, the more likely the test is to detect that misspecification. In order to obtain different amounts of misspecification – while keeping the structure of the model unchanged –, we vary the magnitude of the path coefficients of  $\xi$ ,  $\eta_1$ , and  $\eta_2$  on  $c_1$ ,  $c_2$ , and  $c_3$ . Depending on the combination of population values for these paths, different amounts of misspecification with respect to the second-order model arise.

Technically, we use the `generateData()` function's ability to generate data based on variable parameter values. To achieve this, the package makes use of `lavaan`'s labeling capabilities. Users may replace a given parameter in, i.e., the structural model by a symbolic name – e.g., `a`, `b`, `c` etc. – and assign a value or a vector of values to that name as an argument of `generateData()`. These values will be used to generate data for all possible combinations of these values with the remaining fixed parameters.

Table 4.2 lists the parameter combinations that we used for the alternative DGPs, including the corresponding magnitude of misspecification. We repeat the table here for convenience.

Table 4.12: Population path coefficients in case of misspecifications

Misspecification	SRMR	SRMR <sub>1st</sub>	SRMR <sub>2nd</sub>	$\gamma_3$	$\gamma_4$	$\gamma_5$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$	$\beta_6$
Small	0.0302	0.000	0.0626	0.4	0.1	0.5	0.1	0.5	0.6	0.1	0.2	0.5
Medium	0.0629	0.000	0.1262	-0.4	0.1	0.5	0.1	0.5	0.6	0.2	0.2	0.5
Large	0.0949	0.000	0.1718	0.4	0.1	0.5	0.5	0.4	0.6	-0.4	0.2	0.5

The next subsection details how the parameter values are determined.

```
dgp_alt <- "
## Structural model
eta1 ~ gamma1*xi
eta2 ~ gamma2*xi
c1 ~ a*xi + b*eta1 + c*eta2
c2 ~ d*xi + e*eta1 + f*eta2
c3 ~ g*xi + h*eta1 + i*eta2

## Composite model
xi <- 0.8*y41 + 0.6*y42 + 0.6*y43
eta1 <- 2*y51 + 3*y52 + 5*y53
eta2 <- 2*y61 + 1*y62 + 1*y63
c1 <- 0.8*y11 + 0.4*y12
c2 <- 0.5*y21 + 0.3*y22 + 0.2*y23 + 0.4*y24
c3 <- 0.3*y31 + 0.3*y32 + 0.2*y33 + 0.2*y34 + 0.4*y35 + 0.3*y36

## Composite indicator correlations
# xi
y41 ~~ 0.4*y42 + 0.4*y43
y42 ~~ 0.4*y43

# eta1
y51 ~~ 0.2*y52 + 0.3*y53
```

```

y52 ~~ 0.4*y53

# eta2
y61 ~~ 0.25*y62 + 0.4*y63
y62 ~~ 0.16*y63

# eta3 (the 2nd order construct)
c1 ~~ 0.49*c2 + 0.27*c3
c2 ~~ 0.413*c3

# c1-c3
y11 ~~ 0.3125*y12

y21 ~~ 0.4*y22 + 0.3*y23 + 0.31*y24
y22 ~~ 0.28*y23 + 0.31*y24
y23 ~~ 0.3*y24

y31 ~~ 0.1*y32 + 0.25*y33 + 0.13*y34 + 0.1*y35 + 0.3*y36
y32 ~~ 0.2*y33 + 0.4*y34 + 0.3*y35 + 0.2*y36
y33 ~~ 0.3*y34 + 0.1*y35 + 0.3*y36
y34 ~~ 0.2*y35 + 0.2*y36
y35 ~~ 0.1*y36
"

## Generate population indicator covariance matrices (Sigma matrices)
Sigma_dgp_alt_small <- generateData(.model = dgp_alt, .return_type = "cor",
                                   gamma1 = 0.2, gamma2 = -0.4,
                                   a = 0.4, b = 0.1, c = 0.5,
                                   d = 0.1, e = 0.5, f = 0.6,
                                   g = 0.1, h = 0.2, i = 0.5)

Sigma_dgp_alt_med <- generateData(.model = dgp_alt, .return_type = "cor",
                                  gamma1 = 0.2, gamma2 = -0.4,
                                  a = -0.4, b = 0.1, c = 0.5,
                                  d = 0.1, e = 0.5, f = 0.6,
                                  g = 0.2, h = 0.2, i = 0.5)

Sigma_dgp_alt_large <- generateData(.model = dgp_alt, .return_type = "cor",
                                    gamma1 = 0.2, gamma2 = -0.4,
                                    a = 0.4, b = 0.1, c = 0.5,
                                    d = 0.5, e = 0.4, f = 0.6,
                                    g = -0.4, h = 0.2, i = 0.5)

## Get Sigma matrix
Sigma_dgp_alt_small <- Sigma_dgp_alt_small$dgp[[1]]
Sigma_dgp_alt_med <- Sigma_dgp_alt_med$dgp[[1]]

```

```

Sigma_dgp_alt_large <- Sigma_dgp_alt_large$dgp[[1]]

## Generate data -----
dat1 <- MASS::mvrnorm(200, mu = rep(0, nrow(Sigma_dgp_alt_small)),
                     Sigma = Sigma_dgp_alt_small, empirical = TRUE)
dat2 <- MASS::mvrnorm(200, mu = rep(0, nrow(Sigma_dgp_alt_med)),
                     Sigma = Sigma_dgp_alt_med, empirical = TRUE)
dat3 <- MASS::mvrnorm(200, mu = rep(0, nrow(Sigma_dgp_alt_large)),
                     Sigma = Sigma_dgp_alt_large, empirical = TRUE)

### Collect and save population parameters -----
sout1 <- cSEM::summarize(csem(dat1, dgp_alt))
sout2 <- cSEM::summarize(csem(dat2, dgp_alt))
sout3 <- cSEM::summarize(csem(dat3, dgp_alt))

pop_params_alt_small <- list(
  "Loadings" = {
    l <- sout1$Estimates$Loading_estimates$Estimate
    names(l) <- sout1$Estimates$Loading_estimates$Name
    l
  },
  "Path_coefficients" = {
    p <- sout1$Estimates$Path_estimates$Estimate
    names(p) <- sout1$Estimates$Path_estimates$Name
    p
  },
  "Weights" = {
    w <- sout1$Estimates$Weight_estimates$Estimate
    names(w) <- sout1$Estimates$Weight_estimates$Name
    w
  }
)

pop_params_alt_med <- list(
  "Loadings" = {
    l <- sout2$Estimates$Loading_estimates$Estimate
    names(l) <- sout2$Estimates$Loading_estimates$Name
    l
  },
  "Path_coefficients" = {
    p <- sout2$Estimates$Path_estimates$Estimate
    names(p) <- sout2$Estimates$Path_estimates$Name
    p
  },
  "Weights" = {
    w <- sout2$Estimates$Weight_estimates$Estimate

```

```

    names(w) <- sout2$Estimates$Weight_estimates$Name
    w
  }
)

pop_params_alt_large <- list(
  "Loadings" = {
    l <- sout3$Estimates$Loading_estimates$Estimate
    names(l) <- sout3$Estimates$Loading_estimates$Name
    l
  },
  "Path_coefficients" = {
    p <- sout3$Estimates$Path_estimates$Estimate
    names(p) <- sout3$Estimates$Path_estimates$Name
    p
  },
  "Weights" = {
    w <- sout3$Estimates$Weight_estimates$Estimate
    names(w) <- sout3$Estimates$Weight_estimates$Name
    w
  }
)

### Save -----
save(list = c("Sigma_dgp_2ndorder", "Sigma_dgp_alt_small", "Sigma_dgp_alt_med",
             "Sigma_dgp_alt_large", "pop_params_2ndorder", "pop_params_alt_small",
             "pop_params_alt_med", "pop_params_alt_large"),
     file = "DGPs/DGPs.RData")

```

## 4.8.2 Procedure to obtain the alternative DGPs

The alternative DGPs are obtained by the following procedure:

1. Randomly vary the value of the paths of the constructs  $\xi$ ,  $\eta_1$ , and  $\eta_2$  on  $c1 - c3$ .
2. Generate population covariance matrices and normally distributed data for each parameter combination using the `generateData()` function of the `cSEM.DGP` package. Make sure to set `.empirical = TRUE` such that the covariance between variables of the data we generate perfectly reproduces the population covariance matrix.
3. For each data set, estimate the model containing the second-order construct (i.e., the “wrong” model with respect to the alternative DGPs).



4. If an estimation produces admissible results, compute the SRMR as a measure of misspecification.
5. Compute the empirical 5%, the 50% and 95% quantile of the SRMR values.
6. Obtain parameter combinations that produce SRMRs close to that of the 5%, 50% and 95% quantile.
7. Save DGPs for reproducibility.
8. Select combinations that satisfy the required guidelines on what parameter values should be (i.e., loadings  $> 0.5$ , see below).
9. Randomly pick one population model from each selection. The path coefficients of the three selected alternative population models are presented in the paper.

We want a data generating process that – if estimated using the wrong model, i.e., the model containing a second-order construct – should still be in line with the guidelines for models containing composites as postulated, for instance, by [Sarstedt et al. \(2019\)](#); [Hair et al. \(2017b,a\)](#); [Ramayah et al. \(2016\)](#); [Becker et al. \(2012\)](#). Guidelines for a pure composite model (formative model in PLS-SEM parlance) amount to:

1. All variance inflation factors (VIFs) between indicators should be smaller than 3.
2. All Composite loadings should be larger than 0.5.
3. Indicator correlations should be smaller than 0.5
4. Weights should not be “too small”.

The structural specification of the alternative DGP in general form is:

```
dgp_alt <- "
## Path model / Regressions
eta1 ~ gamma1*xi
eta2 ~ gamma2*xi
c1 ~ a*xi + b*eta1 + c*eta2
c2 ~ d*xi + e*eta1 + f*eta2
c3 ~ g*xi + h*eta1 + i*eta2
```

```
... omitted ...  
"
```

The structural and second-order part of the second-order model is:

```
model_2ndorder <- "  
# Path model / Regressions  
eta1 ~ xi  
eta2 ~ xi  
eta3 ~ xi + eta1 + eta2  
  
# Higher order composite  
eta3 <~ c1 + c2 + c3  
  
... omitted ...  
"
```

The model is clearly misspecified with respect to the alternative DGP no matter what parameters we pick for  $\mathbf{a} - \mathbf{i}$ .

## The procedure

```
## 1. and 2. Vary population path coefficients and generate data with  
## a correlation matrix identical to the population correlation matrix  
  
dat_table <- generateData(  
  .model = dgp_alt,  
  .return_type = "data.frame",  
  .empirical = TRUE,  
  .N = 200,  
  gamma1 = 0.2,  
  gamma2 = -0.4,  
  a = c(-0.4, 0.2, 0.4, 0.8),  
  b = c(0.1, 0.4, 0.6),  
  c = c(0.1, 0.5),  
  d = c(0.1, 0.5),  
  e = c(-0.5, 0.1, 0.4, 0.5),  
  f = c(-0.3, 0.2, 0.4, 0.6),  
  g = c(-0.4, 0.1, 0.2, 0.5),  
  h = c(-0.3, 0.2, 0.5),  
  i = c(-0.4, 0.1, 0.2, 0.5),  
  .handle_negative_definite = "drop") # creates about 180.000 data sets  
  
## 3. and 4. Estimate and compute SRMR if estimates are admissible  
  
res_table <- dat_table %>%
```

```

mutate("SRMR" = sapply(dgp, function(x) {

  out <- csem(x, model_2ndorder)

  if(sum(unlist(verify(out))) == 0) {
    cSEM:::calculateSRMR(out)
  } else {
    NA
  }
}))

res_table2 <- res_table[!is.na(res_table$SRMR), ]

## 5. Compute quantiles of the SRMRs
quants <- quantile(res_table2$SRMR, probs = c(0.05, 0.5, 0.95), na.rm = TRUE)

## 6. Obtain parameter combinations that produce SRMRs close to the
## 5%, 50%, and 95% quantiles of the generated SRMRs
selected <- filter(res_table2,
  (near(quants[1], res_table2$SRMR, tol = 0.001) |
   near(quants[2], res_table2$SRMR, tol = 0.001) |
   near(quants[3], res_table2$SRMR, tol = 0.001)))

## 7. Save the selected parameter combinations for replicability.
save(selected, file = "DGPs/selected_alt_dgps.RData")

## 8. Keep only those parameter combinations that produces estimates that
# are conform with existing guidelines

selected2 <- selected %>%
  mutate("Passed_guidelines" = sapply(dgp, function(x) {
    ## The (extended) repeated indicators approach requires the repeated indicators to
    ## be attached to the data set
    coln <- c(colnames(x), paste0(colnames(x), "_temp"))
    x_RI <- x[, c(1:ncol(x), 1:ncol(x))]
    colnames(x_RI) <- coln

    # Two-stage
    out_TS <- cSEM::summarize(csem(x, model_2ndorder))
    loadings_TS <- min(out_TS$Second_stage$Estimates$Loading_estimates$Estimate)
    weights_TS <- min(out_TS$Second_stage$Estimates$Weight_estimates$Estimate)
    VIF_TS <- unlist(lapply(
      cSEM::calculateVIFModeB(csem(x, model_2ndorder)$Second_stage), max))["eta3"]

    # RI extended
    out_RI <- csem(x_RI, model_RI_extended)
    loadings_RI <- min(cSEM::summarize(out_RI)$Estimates$Loading_estimates$Estimate[1:21])
  })

```

```

loadings_RI_2ndorder <- min(cor(out_RI$Estimate$Construct_scores)[3:5, 7])
weights_RI <- min(cSEM::summarize(out_RI)$Estimates$Weight_estimates$Estimate[1:21])
weights_RI_2ndorder <- min(cSEM::summarize(out_RI)$Estimates$Path_estimates$Estimate[15:17])

VIF_RI <- max(out_RI$Estimates$VIF$eta3)

if(loadings_TS > 0.51 & weights_TS > 0.1 & VIF_TS < 3 &
    loadings_RI > 0.51 & loadings_RI_2ndorder > 0.51 & weights_RI > 0.1 &
    weights_RI_2ndorder > 0.1 & VIF_RI < 3) {
  TRUE
} else {
  FALSE
}
})) %>%
filter(Passed_guidlines == TRUE)

## 9. For the simulation, we pick the following parameter combinations:
# - small misspecification: Id = 31431
# - medium misspecification: Id = 32989
# - large misspecification: Id = 32223
filter(selected2, Id %in% c(32223, 31431, 32989))

```

Note: Other parameter combinations producing similar misspecification could have been picked as well. This is purely random. Results will not differ.

### 4.8.3 Models

We have three models:

- `model_2ndorder`: A specification that includes the second-order composite. If such a model is provided to the `csem()` function, the two-stage approach with a saturated structural model in the first stage is applied by default. To estimate the model using the embedded two-stage approach set argument `.approach_2ndorder` to `"mixed"` in `csem()`. Alternatively, the embedded two-stage approach can be used (called “mixed” in `cSEM`).
- `model_alt`: This specification matches the structure of the alternative DGPs.
- `model_RI_extended`: This specification is used for the extended repeated indicators approach.

```

model_2ndorder <- "
# Structural model
eta1 ~ xi
eta2 ~ xi
eta3 ~ xi + eta1 + eta2

# Composite model
xi <~ y41 + y42 + y43
eta1 <~ y51 + y52 + y53
eta2 <~ y61 + y62 + y63
c1 <~ y11 + y12
c2 <~ y21 + y22 + y23 + y24
c3 <~ y31 + y32 + y33 + y34 + y35 + y36

# Second-order composite
eta3 <~ c1 + c2 + c3
"

model_alt <- "
# Structural model
eta1 ~ xi
eta2 ~ xi
c1 ~ xi + eta1 + eta2
c2 ~ xi + eta1 + eta2
c3 ~ xi + eta1 + eta2

# Composite model
xi <~ y41 + y42 + y43
eta1 <~ y51 + y52 + y53
eta2 <~ y61 + y62 + y63
c1 <~ y11 + y12
c2 <~ y21 + y22 + y23 + y24
c3 <~ y31 + y32 + y33 + y34 + y35 + y36"

model_RI_extended <- "
# Structural model
eta1 ~ xi
eta2 ~ xi
eta3 ~ xi + eta1 + eta2 + c1 + c2 + c3

# Composite model
xi <~ y41 + y42 + y43
eta1 <~ y51 + y52 + y53
eta2 <~ y61 + y62 + y63
c1 <~ y11 + y12
c2 <~ y21 + y22 + y23 + y24
c3 <~ y31 + y32 + y33 + y34 + y35 + y36

```

```

# second-order composite
eta3 <~ y11_temp + y12_temp + y21_temp + y22_temp + y23_temp + y24_temp +
      y31_temp + y32_temp + y33_temp + y34_temp + y35_temp + y36_temp

# Additional indirect effects
c1 ~ xi
c2 ~ xi
c3 ~ xi

c1 ~ eta1
c2 ~ eta1
c3 ~ eta1

c1 ~ eta2
c2 ~ eta2
c3 ~ eta2
"

### Save -----
save(list = c("model_2ndorder", "model_alt", "model_RI_original", "model_RI_extended"),
     file = "Models/Models.RData")

```

## 4.8.4 Simulation

This is the R code that runs the actual simulation.

```

# General Preparation =====
## Install packages if necessary
if(!require(foreach)) install.packages("foreach")
if(!require(doParallel)) install.packages("doParallel")
if(!require(parallel)) install.packages("parallel")
if(!require(cSEM)) install.packages("cSEM")

## Load packages
library(cSEM)      # (version: 0.1.0)
library(foreach)

## Load DGPs and Models
load("DGPs/DGPs.RData") # see Subsection "Data generation" on how to obtain these.
load("Models/Models.RData") # see Subsection "Models" on how to obtain these.

## Source helper functions and models to estimate
source("O_0_HelperFunctions.R") # see Subsection "Helper functions"

# Simulation =====

```

```

### Preparation -----
# Things to loop over
sample_size      <- list(100, 300, 500, 750, 1000)
number_of_draws  <- 1000
number_boot_reps <- 1000
dgp              <- list(
  "Sigma_dgp_2ndorder" = Sigma_dgp_2ndorder,
  "Sigma_dgp_alt_low"  = Sigma_dgp_alt_small,
  "Sigma_dgp_alt_med"  = Sigma_dgp_alt_med,
  "Sigma_dgp_alt_high" = Sigma_dgp_alt_large)
model            <- mget(c("model_2ndorder", "model_alt"))
weighting_scheme <- c("centroid", "factorial", "path")

number_cores     <- parallel::detectCores()
### Monte Carlo simulation -----
## Create cluster
cl <- parallel::makeCluster(number_cores)
doParallel::registerDoParallel(cl)

sim <-
  foreach(j = seq_along(dgp), .packages = c("MASS", "dplyr", "cSEM")) %:%
  foreach(m = seq_along(model)) %:%
  foreach(l = weighting_scheme) %:%
  foreach(i = sample_size) %dopar% {

  # Create dummy structures to make data processing easier

  out1 <- data.frame(
    "Estimate_type" = NA,
    "Name"          = NA,
    "Estimate"      = NA
  )
  out2 <- data.frame(
    "Stage"          = NA,
    "Distance_measure" = NA,
    "99%"           = NA,
    "95%"           = NA,
    "90%"           = NA,
    check.names     = FALSE
  )

  out_dummy <- list(
    "Parameter_estimates" = out1,
    "Test"                 = out2
  )

  out_dummy_status <- data.frame(

```

```

"Status_code" = NA,
"Status_code_1" = NA,
"Status_code_2" = NA
)

out_status <- list(
  list(
    out_dummy_status,
    out_dummy_status,
    out_dummy_status
  )
)

## Note:
## j := the index for the DGP
## m := the index for the model
out_estimation <- list()
counter <- 0
repeat{

  counter <- counter + 1

  ## Generate data
  samp <- DataGeneration(
    .sample_size = i,
    .Sigma = dgp[[j]],
    .ndraws = 1,
    .scale_factor = "normal.kurt",
    .empirical = FALSE
  )[[1]]

  ## The (extended) repeated indicators approach requires the repeated indicators to
## be attached to the data set
  coln <- c(colnames(samp), paste0(colnames(samp), "_temp"))
  samp_RI <- samp[, c(1:ncol(samp), 1:ncol(samp))]
  colnames(samp_RI) <- coln

  ### Two-stage approach -----
  ## Do the estimation (Result is a list of length number_of_draws with
## elements First_stage and Second_stage)
  out_2stage <- csem(
    .data = samp,
    .model = model[[m]],
    .approach_2ndorder = "2stage",
    .PLS_weight_scheme_inner = 1
  )
}

```



```

status_2stage <- sum(unlist(verify(out_2stage)))

## Only compute extended repeated indicators approach and mixed approach
## for the 2nd order model (model_pop_Sigma_2ndorder) and if the DGP is
## the second order DGP (Sigma_2ndorder)
status_mixed <- 0
status_RI_extended <- 0
if(j == 1 & m == 1) {

  ### Mixed two-stage approach -----
  ## Do the estimation (Result is a list of length number_of_draws with
  ## elements First_stage and Second_stage)
  out_mixed <- csem(
    .data = samp,
    .model = model[[m]],
    .approach_2ndorder = "mixed",
    .PLS_weight_scheme_inner = 1
  )

  status_mixed <- sum(unlist(verify(out_mixed)))

  ### Extended repeated indicators approach -----
  out_RI_extended <- csem(
    .data = samp_RI,
    .model = model_RI_extended,
    .PLS_weight_scheme_inner = 1
  )

  status_RI_extended <- sum(unlist(verify(out_RI_extended)))

} else {
  ## Set up dummy structure to facilitate data processing
  out_mixed <- out_dummy
  out_RI_extended <- out_dummy

} # END if j == 1 & m == 1

if(status_2stage == 0 & status_mixed == 0 & status_RI_extended == 0) {

  out_2stage <- computeRelevant(out_2stage, .R = number_boot_reps)
  if(j == 1 & m == 1) {
    out_mixed <- computeRelevant(out_mixed, .R = number_boot_reps)
    out_RI_extended <- computeRelevant(out_RI_extended, .R = number_boot_reps)
  }

  out_estimation[[counter]] <- list(
    "two-stage" = out_2stage,

```

```

    "embedded_TS" = out_mixed,
    "extended_RI" = out_RI_extended
  )
  cat("Simulation run: ", counter)
} else {# status is not ok
  ll <- list(out_2stage, out_mixed, out_RI_extended)

  out_status[[length(out_status) + 1]] <- lapply(ll, function(x) {
    if(inherits(x, "cSEMResults_default")) {
      data.frame(
        "Status_code" = names(verify(x)),
        "Status_code_1" = c(verify(x)),
        "Status_code_2" = NA
      )
    } else if(inherits(x, "cSEMResults_2ndorder")) {
      data.frame(
        "Status_code" = names(verify(x)[[1]]),
        "Status_code_1" = c(verify(x)[[1]]),
        "Status_code_2" = c(verify(x)[[2]])
      )
    } else {
      data.frame(
        "Status_code" = NA,
        "Status_code_1" = NA,
        "Status_code_2" = NA
      )
    }
  })
  # Reset counter
  counter <- counter - 1
}

# Break repeat loop if .R results have been created.
if(length(out_estimation) == number_of_draws) {
  ## Give names
  names(out_status) <- 1:length(out_status)
  break
}
} # END repeat

out <- list(
  "out_estimation" = out_estimation,
  "out_status" = out_status
)
} # END simulation

closeAllConnections() # close connection to relase RAM

```

```
# Save objects =====
save(list = c("sim", "sample_size", "number_of_draws"),
     file = "Data_simulation/sim.RData")
```

## 4.8.5 Helper functions

These functions are required for to run the estimation. The purpose of the two functions is:

- `DataGeneration()`: The function is essentially a wrapper around the `mvrnorm()` function of the `MASS` package. It is used to generate data.
- The function computes the relevant quantities for each Monte Carlo run.

```
DataGeneration <- function(.sample_size, .Sigma, .ndraws, .empirical = F,
                          .scale_factor = c("normal.kurt", "medium.kurt", "high.kurt")) {
  #----- Arguments:
  # .sample_size : numeric value indicating the sample size to draw
  # .Sigma       : (symmetric) indicator correlation matrix
  # .ndraws      : number of replications that should be drawn
  # .empirical   : see ?mvrnorm; defaults to "FALSE"
  # .scale_factor: the scaling factor to be used for each vector of indicator values.
  #               The scaling factor is used to induce non-normality in the data. If
  #               "normal.kurt" (default) the data has a normal kurtosis (kurtosis of 3).
  #               "medium.kurt" increases the kurtosis by 1.7124.
  #               "high.kurt" increases the kurtosis by 6.

  #----- What it does:
  # Draws .ndraws samples from a multivariate normal distribution for a given
  # sample_size and Sigma matrix. The resulting data is either multivariate
  # normal or non-normal depending on the .scale_factor argument. The function
  # creates a list of length .ndraw (=the number of replications)
  # where each list element is a matrix of simulated data based on .Sigma.
  # The matrix has .sample_size rows and ncol columns.

  type      <- match.arg(.scale_factor)
  scale_fac <- if(type == "normal.kurt") {

    1
  } else if(type == "medium.kurt") {

    sqrt(abs(rnorm(.sample_size))*sqrt(pi/2))
  }
}
```

```

} else if(type == "high.kurt") {

  sqrt(3)*(rchisq(.sample_size, 5))(-1/2)
}

replicate(.ndraws,
  scale(scale_fac*MASS::mvrnorm(.sample_size,
                                mu = rep(0, ncol(.Sigma)),
                                Sigma = .Sigma,
                                empirical = .empirical)),
  simplify = FALSE)
}

```

```

computeRelevant <- function(.object, .R) {

  #----- Arguments:
  # .object           : a cSEMResults object

  ## Which columns are relevant in Path_ , Weight_ , and Effect_estimates?
  relevant <- c("Name", "Estimate")

  ## Which parameters are of interest for the simulation
  # All path           : gamma1, gamma2, gamma3, beta1, beta2
  # All weights        : w_y11, w_y21, w_y31, w_y41, w_y51, w_y61, w_c1, w_c2, w_c3
  # All total_effects : te_eta1_eta4, te_eta2_eta4, te_eta3_eta4

  selector_path <- c(
    "eta1 ~ xi",
    "eta2 ~ xi",
    "eta3 ~ xi",
    "eta3 ~ eta1",
    "eta3 ~ eta2"
  )

  selector_weights_1stage <- c(
    "c1 <~ y11", "c1 <~ y12",
    "c2 <~ y21", "c2 <~ y22", "c2 <~ y23", "c2 <~ y24",
    "c3 <~ y31", "c3 <~ y32", "c3 <~ y33", "c3 <~ y34",
    "c3 <~ y35", "c3 <~ y36",
    "xi <~ y41", "xi <~ y42", "xi <~ y43",
    "eta1 <~ y51", "eta1 <~ y52", "eta1 <~ y53",
    "eta2 <~ y61", "eta2 <~ y62", "eta2 <~ y63"
  )

  selector_weights_2ndstage <- c("eta3 <~ c1", "eta3 <~ c2", "eta3 <~ c3")
  selector_te <- c("eta3 ~ xi", "eta3 ~ eta1", "eta3 ~ eta2")

  if(inherits(.object, "cSEMResults_2ndorder")) {

```

```

### Parameter estimates -----
sum_res <- cSEM::summarize(.object)
path <- sum_res$Second_stage$Estimates$Path_estimates[
  sum_res$Second_stage$Estimates$Path_estimates$Name %in% selector_path,
  relevant, drop = FALSE]
weights_1stage <- sum_res$First_stage$Estimates$Weight_estimates[
  sum_res$First_stage$Estimates$Weight_estimates$Name %in% selector_weights_1stage,
  relevant, drop = FALSE]
weights_2stage <- sum_res$Second_stage$Estimates$Weight_estimates[
  sum_res$Second_stage$Estimates$Weight_estimates$Name %in% selector_weights_2ndstage,
  relevant, drop = FALSE]
total_effect <- sum_res$Second_stage$Estimates$Effect_estimates$Total_effect[
  sum_res$Second_stage$Estimates$Effect_estimates$Total_effect$Name %in% selector_te,
  relevant, drop = FALSE]

out1 <- dplyr::bind_rows(
  list("Path" = path,
       "Weights" = dplyr::bind_rows(weights_1stage, weights_2stage),
       "Total_effect" = total_effect),
  .id = "Estimate_type")

### Overall model fit test -----
## Compute test only for two-stage approach
if(.object$Second_stage$Information$Approach_2ndorder == "2stage") {

  ## Joint test
  out2a <- testOMF(.object,
    .handle_inadmissibles = "replace",
    .alpha = c(0.1, 0.05, 0.01),
    .verbose = FALSE, .R = .R)

  out2b <- data.frame(
    "Distance_measure" = names(out2a$Test_statistic),
    stringsAsFactors = FALSE
  )

  out2a <- cbind(out2b, out2a$Decision)
  rownames(out2a) <- NULL
  out2 <- cbind("Stage" = "Both", out2a)

  ## Test first and second stage separately

  out2c <- lapply(.object, function(x) {
    x <- testOMF(x, .handle_inadmissibles = "replace", .alpha = c(0.1, 0.05, 0.01),
      .verbose = FALSE, .R = .R)
    xa <- data.frame(
      "Distance_measure" = names(x$Test_statistic),

```

```

    stringsAsFactors = FALSE
  )

  x <- cbind(xa, x$Decision)
  rownames(x) <- NULL
  x
})
## Combine
out2c[["Both"]] <- out2a
out2 <- dplyr::bind_rows(out2c, .id = "Stage")
} else {
  ## Set up empty data set to retain the same structure as for the second
  ## case
  out2 <- data.frame(
    "Stage"          = NA,
    "Distance_measure" = NA,
    "99%"           = NA,
    "95%"           = NA,
    "90%"           = NA,
    check.names = FALSE
  )
}
} else {

  selector_weights2 <- c("eta3 ~ c1", "eta3 ~ c2", "eta3 ~ c3")

  sum_res <- cSEM::summarize(.object)
  path <- sum_res$Estimates$Path_estimates[
    sum_res$Estimates$Path_estimates$Name %in% selector_path, relevant,
    drop = FALSE]
  weights <- sum_res$Estimates$Weight_estimates[
    sum_res$Estimates$Weight_estimates$Name %in% c(selector_weights_1stage,
    selector_weights_2ndstage),
    relevant, drop = FALSE]
  weights2 <- sum_res$Estimates$Path_estimates[
    sum_res$Estimates$Path_estimates$Name %in% selector_weights2, relevant,
    drop = FALSE]

  if(nrow(weights2) == 3) {
    weights2$Name <- c("eta3 <~ c1", "eta3 <~ c2", "eta3 <~ c3")
    weights <- dplyr::bind_rows(weights, weights2)
  }
  total_effect <- sum_res$Estimates$Effect_estimates$Total_effect[
    sum_res$Estimates$Effect_estimates$Total_effect$Name %in% selector_te,
    relevant, drop = FALSE]

  out1 <- dplyr::bind_rows(

```

```

list("Path" = path,
     "Weights" = weights,
     "Total_effect" = total_effect),
     .id = "Estimate_type")

## Only compute test if no repeated indicators are present (in this case the
## length of data would be 35)
if(ncol(.object$Information$Data) == 21) {
  ## Joint test
  out2a <- testOMF(.object,
                  .handle_inadmissibles = "replace",
                  .alpha = c(0.1, 0.05, 0.01),
                  .verbose = FALSE, .R = .R)

  out2b <- data.frame(
    "Distance_measure" = names(out2a$Test_statistic),
    stringsAsFactors = FALSE
  )

  out2a <- cbind(out2b, out2a$Decision)
  rownames(out2a) <- NULL
  out2 <- cbind("Stage" = "Both", out2a)

} else {
  ## Set up empty data set to retain the same structure as for the second-order
  ## case
  out2 <- data.frame(
    "Stage" = NA,
    "Distance_measure" = NA,
    "99%" = NA,
    "95%" = NA,
    "90%" = NA,
    check.names = FALSE
  )
}
}

## Return list
out <- list(
  "Parameter_estimates" = out1,
  "Test" = out2
  # "Status_code" = out3
)
}

```

---

## Chapter 5

# cSEM: An R package for composite-based structural equation modeling

---

### 5.1 Introduction<sup>1</sup>

cSEM ([Rademaker and Schubert, 2020](#)) is an R package that provides functions to estimate, analyze, assess, test, and study linear, nonlinear, hierarchical, and multi-group structural equation models using composite-based approaches and procedures. The primary focus of the package is on those procedures centered around the partial least squares path modeling (PLS-PM) algorithm (e.g., [Wold, 1975, 1982](#)), including extensions such as consistent partial least squares (PLSc; [Dijkstra and Henseler, 2015b](#)), ordinal (consistent) partial least squares (OrdPLSc; [Schubert et al., 2018b](#)), and robust (consistent) partial least squares (robustPLSc; [Schamberger et al., 2020](#)). Besides PLS-PM, cSEM includes estimation approaches such as generalized structured component analysis (GSCA; e.g., [Hwang and Takane, 2004](#),

---

<sup>1</sup>This chapter is based on package help files, vignettes, as well as a working paper. By the time this dissertation was submitted for review, cSEM was available on the Comprehensive R Archive Network (CRAN) as version 0.2.0. cSEM is co-authored by Florian Schubert. Development started in January 2018.



2014), generalized structured component analysis with uniqueness terms (GSCAm; Hwang et al., 2017b), generalized canonical correlation analysis (GCCA; Kettenring, 1971), principal component analysis (PCA), and factor score regression (FSR) using sum, regression, or Bartlett scores – including bias correction using Croon (2002)’s approach (e.g., Devlieger and Rosseel, 2017).

The intended principal purpose of `cSEM` is twofold:

1. Provide a unified framework to common composite-based structural equation modeling (SEM) approaches, including typical postestimation procedures. Ideally, the package should be similar to what the R package `lavaan` (Rosseel, 2012) constitutes for researchers and practitioners conducting factor-based SEM analysis in R.
2. Make sure that the user experience is as simple and friendly as possible.

The first point is an always ongoing task; approaches are constantly evolving with new developments appearing at a pace that my co-author, Florian Schubert, and I, the package author, will not be able to keep up with given the limited resources we have at our disposal. However, as of version 0.2.0, the package includes most of the existing capabilities that other R packages such as `semPLS` (Monecke and Leisch, 2012), `plspm` (Sanchez et al., 2017), `matrixpls` (Rönkkö, 2017), `gesca` (Hwang et al., 2017a), `seminr` (Ray et al., 2019) and commercial software (e.g., smartPLS (Ringle et al., 2015), ADANCO (Henseler, 2019)) addressing similar content offer. In addition, `cSEM` offers approaches, methods, and ease-of-use that, to the best of my knowledge, are not offered by any other R package related to composite-based SEM. The second point, the user experience, has been particularly important to me as I have regularly been frustrated by how technical and user-unfriendly packages in R can be. From the very beginning of the package development in January 2018, I therefore envisioned a consistent, intuitive workflow that essentially only comprises three steps:

1. Import the data into R and specify a model.
2. Estimate using a single function that only requires a data set and a model.
3. Apply simple “postestimation verbs” which require nothing but the results of the estimation, yet cover all the analysis steps a typical user may need.

Moreover, I paid particular attention to informative and visually appealing print and plot methods, as I believe that presentation of results is vital for a friendly user-experience.

Before I explain in detail what I consider to be the *cSEM workflow*, a brief overview of the three steps is given first.

1. **Data and model:** no estimator or approach works without data, a specification of what parameters are to be estimated, and how data is related to these parameters, i.e., a model. To do anything in *cSEM*, a data set and a model specification are thus mandatory. Model specification is done in *lavaan model syntax*<sup>2</sup>; model specification in *lavaan model syntax* is arguably the most intuitive way of specifying structural equation models in R – and well known to R users that have an interest in SEM. Consequently, the syntax qualified as the most appealing way for users to specify their model. As for the data, experience tells, for R beginners the biggest obstacle has been to get the data into R. However, largely thanks to the *tidyverse* (Wickham et al., 2019) and *RStudio* (RStudio Team, 2015), data import and data transformation are nowadays relatively easy to handle. To mitigate potential problems caused by different data types and classes, *cSEM* is designed to be flexible as to the type of data it accepts.
2. **Estimation:** regardless of the model and type of data, estimation is always done using the central function of the package: `csem()`. The function takes a data set as its first and the model as its second argument:

```
csem(.data = my_data, .model = my_model)
```

The result of a call to `csem()` is an object of class *cSEMResults*. Naturally, the `csem()` function has a number of additional arguments to fine-tune the estimation, most notably, arguments to change the estimation approach. Since `csem()` automatically recognizes, for instance, whether a concept was modeled as a common factor or composite and automatically applies appropriate correction for attenuation, only a small number of arguments need to be changed in practical applications.

---

<sup>2</sup>See: <http://lavaan.ugent.be/tutorial/syntax1.html>; last accessed: 31.03.2020.

3. **Postestimation:** inspired by the [grammar of data manipulation](#)<sup>3</sup> underlying widely-known R packages such as `dplyr` (Wickham et al., 2020) and `tidyr` (Wickham and Henry, 2020), `cSEM` provides a verb-only postestimation framework that concisely covers all common postestimation tasks. The purpose of each function should be largely self-explanatory thanks to its name. Currently there are five major postestimation verbs, four additional `test`-commands and four `do`-commands:

- `assess()`
- `infer()`
- `predict()`
- `summarize()`
- `verify()`
- `testOMF()`
- `testMICOM()`
- `testMGD()`
- `testHausman()`
- `doFloodlightAnalysis()`
- `doSurfaceAnalysis`
- `doRedundancyAnalysis()`
- `doIPMA()`

All verbs accept the result of a call to `csem()`, i.e., a `cSEMResults` object, as input and do not require further mandatory arguments, which makes working with these function as simple as possible – only remember the verb, not any specific syntax or arguments. In addition, all functions have a number of additional arguments to fine-tune the postestimation.

Figure 5.1 summarizes the workflow.

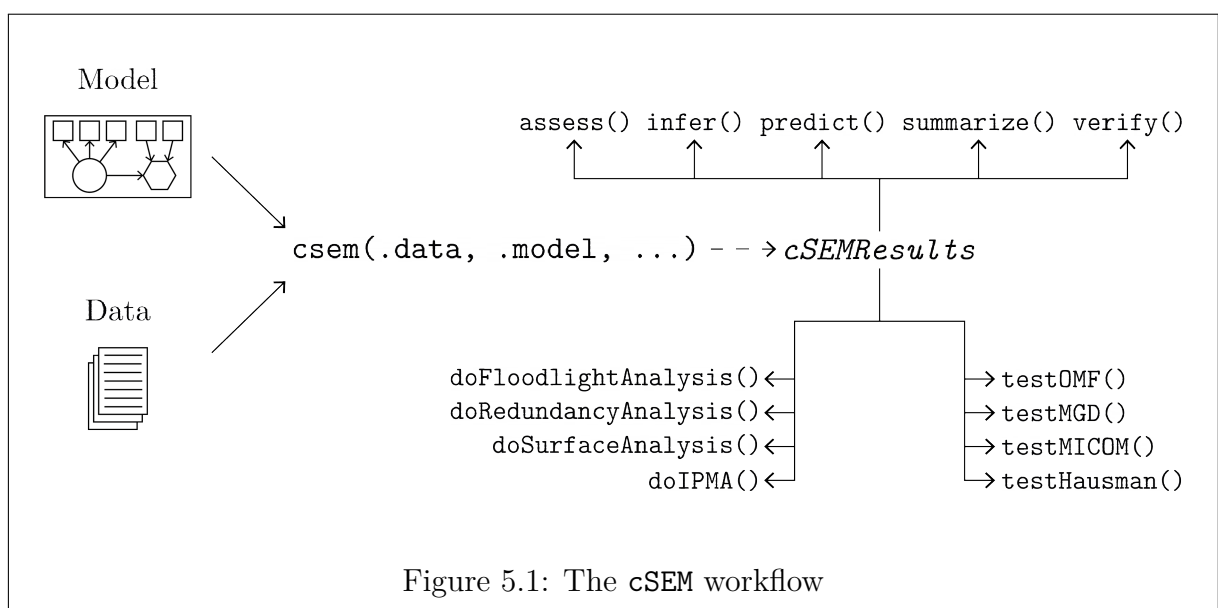


Figure 5.1: The cSEM workflow

<sup>3</sup>See: <https://dplyr.tidyverse.org/>; last accessed: 31.03.2020

As a consequence of the design choice, the price for an increase in flexibility is primarily a, mostly minor, loss in computational speed, in particular, when large-scale resampling is involved – i.e., 10,000 bootstrap runs for a complex model with, say, 1,000 observations.<sup>4</sup> Users looking for the most efficient implementation of common resampling routines may find faster implementations. In particular, implementations that exhaust the full capacity of C or C++ code are likely to excel in terms of speed. That said, I believe, the time saved when using a standardized estimate-postestimate workflow, no matter the model or data used, well outweighs the occasional computational inefficiencies.

For illustration, several example data sets are used throughout this chapter. The data sets and corresponding models are either generated, used in [Henseler \(forthcoming\)](#)<sup>5</sup>, or imported from other R packages. All data sets are available in `cSEM`. Since data and models are used purely for illustration purposes, readers are referred to the package help files and [Henseler \(forthcoming\)](#) for details on the data and the reasoning behind the corresponding models.

## 5.2 Preparing the data

Technically, preparing the data does not require `cSEM` and is therefore better considered a preparation task, i.e., a “pre-`cSEM`” task. The reason why this step is nevertheless considered an explicit part of the `cSEM`-workflow is motivated by the experience that applied users tend to shy away from software like R because “just getting the data in” and understanding how to show, manipulate, and work with data can be frustrating – at least, if one is not aware of R’s rich and easy to learn data import and data processing capabilities. While these topics may have been overwhelming for newcomers several years ago, data import and data transformation have become relatively simple and user-friendly if the right packages are used. A good place to start is the [Rstudio cheat sheet webpage](#)<sup>6</sup>, especially the **Data Import** and the **Data Transformation** cheat sheets.

---

<sup>4</sup>Two years of experience in developing and maintaining the package have also shown that an increase in flexibility for the user translates to an increase in complexity for the developer. This has caused a significant amount of complications and extra hours during the development.

<sup>5</sup>By the time this thesis was submitted for review, the book had not been published yet. I am grateful to Jörg Henseler for allowing me to use the data sets nonetheless.

<sup>6</sup>See: <https://www.rstudio.com/resources/cheatsheets/>; last accessed: 31.03.2020.

To avoid potential frustrations, `cSEM` is relatively flexible as to the type of data accepted. Currently the following data types and structures are accepted:

1. A data frame or `tibble` (Müller and Wickham, 2020), a type of data frame, with column names matching the indicator names used in the lavaan model syntax description. Possible column types or classes of the data provided are: "logical" (TRUE/FALSE), "numeric" ("double" or "integer"), "factor" ("ordered" and/or "unordered"), or a mix of several types. Additionally, the data may also include **a single** character column whose column name must be given to the `.id` argument of `csem()`. Values of this column are interpreted as group identifiers; `csem()` will split the data by levels of that column and run the estimation for each level separately. This is particularly useful if the user intends to perform a multigroup analysis as discussed in more detail in Section 5.5.

The order of the columns in the data set is irrelevant as variable names are reordered internally to assure each column of the data is matched to the correct indicator of the model description. Moreover, unused columns are simply ignored.

2. A matrix with column names matching the indicator names used in the lavaan model description of the measurement model or composite model description. Matrices must be atomic, i.e., must have only one column-type. Typically, matrices are used when the data is all-numeric.
3. A list of data frames or matrices. In this case estimation is repeated for each data frame or matrix separately. This is useful if the user intends to perform a multigroup analysis as discussed in more detail in Section 5.5.

**Example 5.2.1.** Assuming the following simple linear model involving three constructs modeled as composites is to be estimated. In lavaan model syntax:

```
model <- "  
# Composite model  
AgrIneq <~ gini + farm + rent  
IndDev <~ gnpr + labo  
PolInst <~ inst + ecks + deat + stab + dict
```

```
# Structural model
PolInst ~ AgrIneq + IndDev
"
```

To estimate the model a data frame, a tibble, a matrix, or a list of any of the three with  $N$  rows (the observations) and at least  $K = 10$  columns with column names `gini`, `farm`, `rent`, `gnpr`, `labo`, `inst`, `ecks`, `deat`, `stab`, `dict` is required. The order of the columns in the data does not matter as they are reordered internally.

The current version 0.2.0 of `cSEM` does not provide any tools to handle missing values; hence, data must not contain missing values. Although, `cSEM` detects missing values and warns the user, this is nevertheless a limitation, I intend to address. Future versions are likely to include a framework to address missing values.

### 5.3 Specifying a model

Models are defined using [lavaan model syntax](#). Currently, only the “standard” lavaan model syntax is supported. This comprises:

1. The definition of a latent variable/common factor by the “`=~`” operator.
2. The definition of a composite by the “`<~`” operator.
3. The specification of regression (structural) equations by the “`~`” operator.
4. The definition of error (co)variances, indicator correlations, or correlations between constructs using the “`~~`” operator.

Other lavaan model syntax is ignored by `cSEM`. Specifying, for instance, a threshold using the “`|`” operator or a scaling factor using the “`~ * ~`” operator would thus be silently ignored.

`cSEM` handles linear, nonlinear, and hierarchical models. Syntax for each model is illustrated next. I use variables of the build-in `satisfaction` data set for linear models, the `Switching` data set for nonlinear models, and the `Yooetal2000` data set for second-order models.

### 5.3.1 Linear models

A full linear model specification in lavaan model syntax could look like this:

```
model <- "  
# Structural model  
EXPE ~ IMAG  
QUAL ~ EXPE  
VAL ~ EXPE + QUAL  
SAT ~ IMAG + EXPE + QUAL + VAL  
LOY ~ IMAG + SAT  
  
# Composite model  
IMAG <- imag1 + imag2 + imag3           # composite  
EXPE <- expe1 + expe2 + expe3           # composite  
QUAL <- qual1 + qual2 + qual3 + qual4 + qual5 # composite  
VAL <- val1 + val2 + val3               # composite  
  
# Reflective measurement model  
SAT =~ sat1 + sat2 + sat3 + sat4        # common factor  
LOY =~ loy1 + loy2 + loy3 + loy4       # common factor  
  
# Measurement error correlation  
sat1 ~~ sat2  
"
```

Note that the operator  $<\sim$  tells cSEM that the construct to its left is modeled as a composite; the operator  $=\sim$  tells cSEM that the construct to its left is modeled as a common factor. The operator  $\sim\sim$  tells cSEM that the *measurement errors* of the indicators `sat1` and `sat2` are allowed to correlate. The `#` sign indicates a comment that is ignored when parsing the model.

### 5.3.2 Models containing nonlinear terms

Nonlinear terms are specified as interactions using the dot operator `"."`.<sup>7</sup> Nonlinear terms include interactions and exponential terms. The latter is described in model syntax as an “interaction with itself”, e.g.,  $x_1^3 = x1.x1.x1$ . Currently, the following terms are supported:

- Single, e.g., INV

---

<sup>7</sup>Interactions in `lavaan` are specified using the colon operator `":"`. Unfortunately, I only became aware of this feature when I had already fully implemented the dot operator in `cSEM`. Although, this may occasionally confuse users jumping between `lavaan` and `cSEM`, my co-author and I decided to accept this “inconsistency”; reversing it would be highly time consuming for relatively little gain.

- Quadratic, e.g., `INV.INV`
- Cubic, e.g., `INV.INV.INV`
- Two-way interaction, e.g., `INV.SAT`
- Three-way interaction, e.g., `INV.SAT.VAR3`
- Quadratic and two-way interaction, e.g., `INV.INV.SAT`

Using the `Switching` data set, a simple example model involving an interaction term could look like this:

```
model <- "
# Measurement models
INV =~ INV1 + INV2 + INV3 + INV4
SAT =~ SAT1 + SAT2 + SAT3
INT =~ INT1 + INT2

# Structural model containing an interaction term
INT ~ INV + SAT + INV.SAT
"
```

### 5.3.3 Hierarchical (second-order) models

With regards to hierarchical models, only second-order constructs are currently supported, i.e., models containing at least one concept which is theorized to be a second-order construct. Specification of the second-order construct takes place in the measurement/composite model. Second-order models are therefore specified using the operators `=~` or `<~`. These operators are usually used with indicators on their right-hand side. For second-order models the right-hand side variables are constructs instead.

An example is the model used for the `Yooetal2000` data set.

```
model <- "
# Measurement models
PR =~ PR1 + PR2 + PR3
IM =~ IM1 + IM2 + IM3
DI =~ DI1 + DI2 + DI3
AD =~ AD1 + AD2 + AD3
DL =~ DL1 + DL2 + DL3
AA =~ AA1 + AA2 + AA3 + AA4 + AA5 + AA6
"
```



```

LO =~ LO1 + LO3
QL =~ QL1 + QL2 + QL3 + QL4 + QL5 + QL6

# Composite model for second-order construct BR
BR <- QL + LO + AA

# Structural model
BR ~ PR + IM + DI + AD + DL
"

```

In this case, BR is modeled as a second-order composite formed by three common factors QL, LO, and AA.

## 5.4 Estimation using `csem()`

`csem()` is *the* central function of the package. Although, it is possible to estimate a model “by hand” using individual functions called by `csem()` (e.g., `parseModel()`, `processData()`, `calculateWeightsPLS()`, `estimatePath()` etc.) , it is, by design, easier, safer, and quicker to use `csem()` instead.<sup>8</sup>

**Example 5.4.1.** The simplest possible call to `csem()` involves a data set and a model. For illustration, I use a very simple model. The model involves three constructs modeled as common factors each measuring three indicators. The corresponding `threecommonfactors` data is also available from the package. Note that the `threecommonfactors` data set contains 500 observations drawn from a population indicator covariance matrix that matches the postulated model. Sample variation aside, the model thus represents the underlying structure of the data and estimation is expected to yield parameters close to their population value. Population path coefficients, for instance, are 0.6 for the path of  $\eta_1$  on  $\eta_2$ , 0.4 for the path of  $\eta_1$  on  $\eta_3$ , and 0.35 for the path of  $\eta_2$  on  $\eta_3$ .

```

## Load the cSEM package
require(cSEM)

## Specify model in lavaan model syntax
model <- "

```

---

<sup>8</sup>Functions such as `estimatePath()` are internal, i.e., not directly available to the user once the package is loaded in R. Internal functions can, however, still be called using R’s so called “:::” (dotdotdot)-mechanism for non-exported functions. For details, see <https://cran.r-project.org/doc/manuals/R-exts.html#Using-pthreads>; Section 1.5.1 “Specifying imports and exports”.

```

# Path model / Regressions
eta2 ~ eta1
eta3 ~ eta1 + eta2

# Reflective measurement model
eta1 =~ y11 + y12 + y13
eta2 =~ y21 + y22 + y23
eta3 =~ y31 + y32 + y33
"

```

Using the `threecommonfactors` data set and the model saved under `model`, a minimal call to `csem()` is now possible.

```
res1 <- csem(.data = threecommonfactors, .model = model)
```

Calling the saved object `res1` tells the user whether `csem()` encountered any issues – as checked by the postestimation function `verify()` discussed in Subsection 5.5.5. Moreover, basic information about the structure of the resulting object is printed and a hint at possible next steps is given.

```

res1

## -----
## ----- Overview -----
##
## Estimation was successful.
##
## The result is a list of class cSEMResults with list elements:
##
## - Estimates
## - Information
##
## To get an overview or help type:
##
## - ?cSEMResults
## - str(<object-name>)
## - listviewer::jsondedit(<object-name>, mode = 'view')
##
## If you wish to access the list elements directly type e.g.
##
## - <object-name>$Estimates
##
## Available postestimation commands:
##
## - assess(<object-name>)

```

```

## - infer(<object-name>)
## - predict(<object-name>)
## - summarize(<object-name>)
## - verify(<object-name>)
## -----

```

When calling `csem()`, all arguments not explicitly named are set to their defaults. The simplest call is thus equivalent to:

```

csem(
  .data                = threecommonfactors,
  .model               = model,
  .approach_cor_robust = "none",
  .approach_nl         = "sequential",
  .approach_paths      = "OLS",
  .approach_weights    = "PLS-PM",
  .conv_criterion      = "diff_absolute",
  .disattenuate        = TRUE,
  .dominant_indicators = NULL,
  .estimate_structural = TRUE,
  .id                  = NULL,
  .iter_max            = 100,
  .normality           = FALSE,
  .PLS_approach_cf     = "dist_squared_euclid",
  .PLS_ignore_structural_model = FALSE,
  .PLS_modes           = NULL,
  .PLS_weight_scheme_inner = "path",
  .reliabilities       = NULL,
  .starting_values     = NULL,
  .tolerance           = 1e-05,
  .resample_method     = "none",
  .resample_method2    = "none",
  .R                   = 499,
  .R2                  = 199,
  .handle_inadmissibles = "drop",
  .user_funs           = NULL,
  .eval_plan           = "sequential",
  .seed                = NULL,
  .sign_change_option  = "no"
)

```

Most of the arguments are discussed in the following sections and subsections. For details on arguments not explicitly discussed, the reader is referred to the `csem()`

reference on the [cSEM website](#).<sup>9</sup>

### 5.4.1 Weighting approaches

Of particular importance is the `.approach_weights` argument; the third most important argument after `.data` and `.model`.

By virtue of the package, `cSEM` uses composite-based approaches only. The defining feature of a composite-based approach according to my understanding is that an approach is “composite-based” if it involves linear compounds, i.e., a weighted linear combination of observables when retrieving (estimating) quantities of interest such as the coefficients of the structural model. Indeed, the way the weights forming the compound are obtained is, in fact, the crucial distinguishing feature between all composite-based approaches.

Consequently, composite-based approaches such as PLS-PM or GSCA are best exclusively understood as prescriptions for forming linear compounds based on observables, i.e., as weighting approaches. In `cSEM`, this is reflected by the fact that e.g., *"PLS-PM"* and *"GSCA"* are choices of the `.approach_weights` argument.

Table 5.1 on the next page lists the available choices, offers a brief description, and provides the relevant references. Note that as of version 0.2.0, applicability of some of the postestimation functions discussed in Section 5.5 below are limited to PLS-PM.

Limitations are due to two reasons:

1. As the package is still developing, some routines have simply not been addressed yet. Trying to estimate a model containing nonlinear terms using GSCA, for instance, causes an error informing the user that the approach has not yet been implemented.
2. Some postestimation approaches are, by the nature of their intended purpose, not applicable to particular weighting approaches. PCA, for instance, does not actually estimate a measurement model. Quality criteria such as the average variance extracted (AVE) or the heterotrait-monotrait ratio of correlations (HTMT) – as computed by `assess()` discussed in Subsection 5.5.1 – are therefore not meaningfully applicable.

---

<sup>9</sup>See: <https://m-e-rademaker.github.io/cSEM/reference/csem.html>; last accessed: 06.04.2020.

Table 5.1: Weighting approaches available via `.approach_weights`

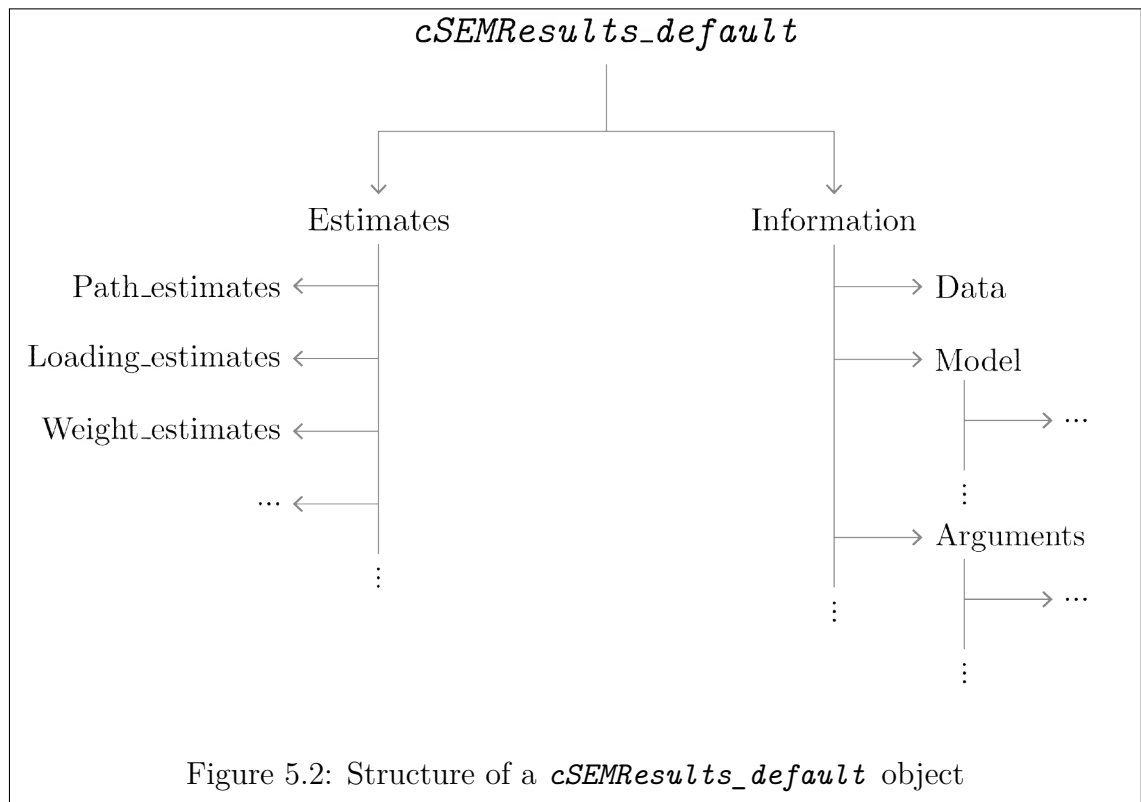
Approach	Description & Arguments	cSEM name	References
Partial least squares path modeling (PLS-PM)	Important additional arguments typically required: <code>.PLS_weights_scheme_inner</code> to pick the inner weighting scheme (" <i>factorial</i> ", " <i>centroid</i> ", or " <i>path</i> " (the default)); <code>.PLS_modes</code> to pick the mode (" <i>modeA</i> ", " <i>modeB</i> ", " <i>unit</i> ", " <i>modeBNNLS</i> ", " <i>PCA</i> "). Default is " <i>modeA</i> ". If <code>.disattenuate = TRUE</code> (the default), consistent partial least squares (PLSc; <a href="#">Dijkstra and Henseler, 2015b</a> ) is used for models containing common factors.	"PLS-PM"	<a href="#">Wold (1975)</a> <a href="#">Wold (1982)</a> <a href="#">Lohmöller (1989)</a> <a href="#">Dijkstra and Henseler (2015b)</a>
Generalized structured component analysis (GSCA)	If <code>.disattenuate = TRUE</code> (the default), generalized structured component analysis with uniqueness terms (GSCAm; <a href="#">Hwang et al., 2017b</a> ) is used for models containing common factors.	"GSCA"	<a href="#">Hwang and Takane (2004)</a> <a href="#">Hwang and Takane (2014)</a> <a href="#">Hwang et al. (2017b)</a>
Kettenring approaches (GCCA)	The approaches proposed by <a href="#">Kettenring (1971)</a> . No additional arguments required.	"MAXVAR", "MINVAR", "SUMCORR", "SSQCOR", "GENVAR"	<a href="#">Kettenring (1971)</a>
Principal component analysis (PCA)	Obtains the weights as the first eigenvector of the empirical indicator correlation matrix. The eigenvector is scaled s.t. the corresponding composite has a variance of one.	"PCA"	–
Factor score regression (FSR)	Factor score regression as explained in e.g., <a href="#">Devlieger and Rosseel (2017)</a> . For unit weights, weights are set to 1 and scaled s.t. the corresponding composite has a variance of one.	"unit", "bartlett", "regression"	<a href="#">Thomson (1934)</a> <a href="#">Bartlett (1937)</a> <a href="#">Devlieger and Rosseel (2017)</a>

## 5.4.2 The *cSEMResults* class

`csem()` accepts all model and data types described in Section 5.2. Every call to `csem()` results in an object of class *cSEMResults*. To be precise, `csem()` technically results in an object with at least two class attributes. The first class attribute is always *cSEMResults*, no matter the type of data or model provided. The second is one of *cSEMResults\_default*, *cSEMResults\_multi*, or *cSEMResults\_2ndorder* and depends on the estimated model and/or the type of data provided. The third class attribute, *cSEMResults\_resampled*, is only added if resampling as described in Subsection 5.4.3 was conducted.

Depending on the class of the resulting object, three different output structures exist. Technically, these output structures are organized as named list; all R functions that work on lists, therefore work on *cSEMResults* objects as well.

**The default structure** All objects that inherit the *cSEMResults\_default* class have the default structure. Figure 5.2 illustrates the default structure. This will be the structure for the vast majority of applications.



Users can access list elements by name using the `$` operator. To get, for

example, the path estimates or information on the number of iterations that were necessary for convergence of the default algorithm in Example 5.4.1 (PLS-PM in this case), type:

```
res1$Estimates$Path_estimates

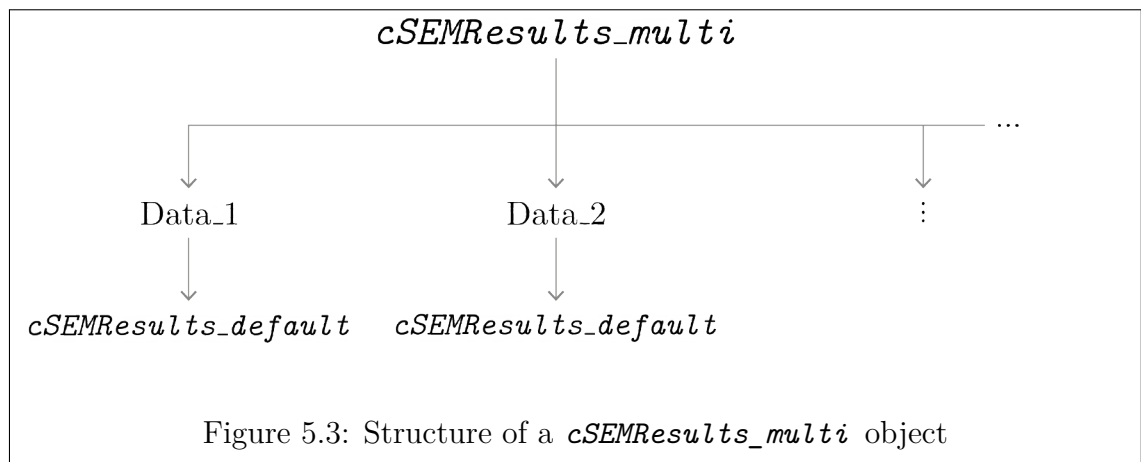
##           eta1           eta2 eta3
## eta1 0.0000000 0.0000000    0
## eta2 0.6713334 0.0000000    0
## eta3 0.4585068 0.3051511    0

res1$Information$Weight_info$Number_iterations

## [1] 4
```

**The multi-group structure** All objects that inherit the *cSEMResults\_multi* class have the multi-group structure. If the data provided to *csem()* is, a) a single matrix or data frame containing an id-column to split the data by its *G* group levels, or, b) a list of *G* data sets, the resulting object is a list of *G* lists each of class *cSEMResults\_default*. Hence their structure is identical to default structure. Figure 5.3 illustrates the structure.

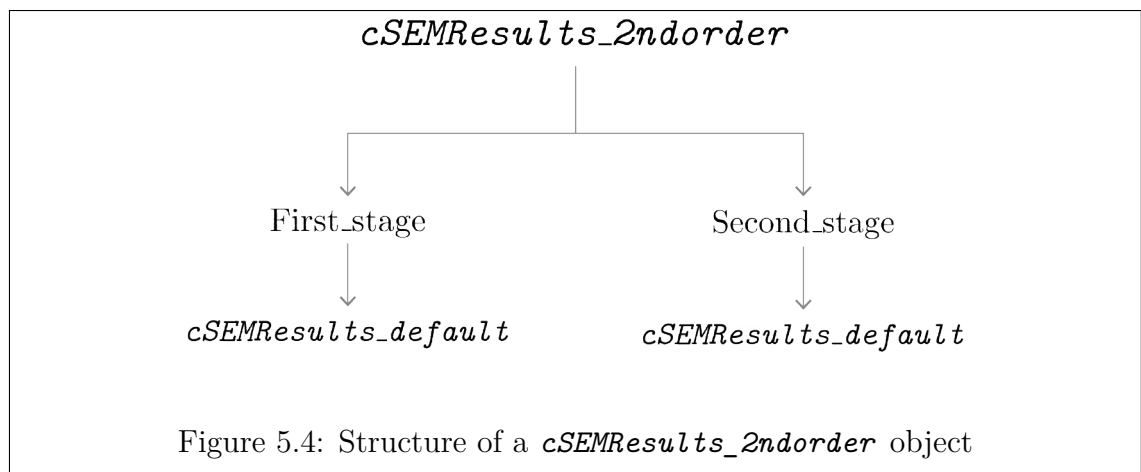
By default, the *G* list elements are named according to the *G* group levels of the id-column or the names of the *G* data sets. If the data sets have no name attributes, list elements are labeled Data\_1, Data\_2, etc.



Again, list elements can be accessed using the \$ operator. Since each list (group or data set) has an identical structure, functions such as *lapply()* can be used

to easily apply a function on each group simultaneously.

**The two-stage structure** A special output is generated if the model to be estimated contains hierarchical constructs and the two-stage or the embedded two-stage approach – also called the “mixed approach” – is used to estimate the model (`.approach_2ndorder = "2stage"` or `.approach_2ndorder = "mixed"` in `csem()`). In this case, the resulting object is, similar to the multi-group structure, a list containing two elements `First_stage` and `Second_stage`. While each list element has class `cSEMResults_default` again, the “groups” differ compared to a `cSEMResults_multi` object in that a different model is estimated in each stage. Figure 5.4 illustrates the structure.



If `.resample_method = "bootstrap"` or `.resample_method = "jackknife"` in `csem()`, resamples are attached to each object. For objects of class ...

- `cSEMResults_default` the resamples are attached to `.object$Estimates$Estimates_resample`.
- `cSEMResults_multi` resamples are attached to `.object$<group_name>$Estimates$Estimates_resample` for each group/data.
- `cSEMResults_2ndorder` resamples are attached to `.object$Second_stage$Information$Resamples`.

All objects containing these elements gain the `cSEMResults_resampled` class.



### 5.4.3 Inference

By default, no inferential quantities are calculated since composite-based approaches, generally, do not have closed-form solutions for standard errors.

To obtain inferential quantities like standard errors, test statistics, critical quantiles, and confidence intervals, `cSEM` relies on resampling techniques such as bootstrap, jackknife, and permutation. Depending on the complexity of the model, the method chosen, the size of the data, and the hardware available, resampling can be time consuming. Consequently, `csem()` and postestimation functions such as `summarize()` and `assess()` do not compute and display any inferential quantities unless explicitly requested by the user. Using `summarize()` on the `cSEMResults` object `res1` computed in Example 5.4.1 in the previous subsection therefore displays NA for standard errors, t-statistics and p-values.

```
summarize(res1)

## -----
## ----- Overview -----
##
## General information:
## -----
## Estimation status           = Ok
## Number of observations      = 500
## Weight estimator           = PLS-PM
## Inner weighting scheme     = "path"
## Type of indicator correlation = Pearson
## Path model estimator       = OLS
## Second-order approach      = NA
## Type of path model         = Linear
## Disattenuated              = Yes (PLSc)
##
## Construct details:
## -----
## Name  Modeled as  Order  Mode
##
## eta1  Common factor  First order  "modeA"
## eta2  Common factor  First order  "modeA"
## eta3  Common factor  First order  "modeA"
##
## ----- Estimates -----
##
## Estimated path coefficients:
## =====
```

```

## Path          Estimate Std. error  t-stat.  p-value
## eta2 ~ eta1    0.6713      NA       NA       NA
## eta3 ~ eta1    0.4585      NA       NA       NA
## eta3 ~ eta2    0.3052      NA       NA       NA
##
## Estimated loadings:
## =====
## Loading        Estimate Std. error  t-stat.  p-value
## eta1 =~ y11    0.6631      NA       NA       NA
## eta1 =~ y12    0.6493      NA       NA       NA
## eta1 =~ y13    0.7613      NA       NA       NA
## eta2 =~ y21    0.5165      NA       NA       NA
## eta2 =~ y22    0.7554      NA       NA       NA
## eta2 =~ y23    0.7997      NA       NA       NA
## eta3 =~ y31    0.8223      NA       NA       NA
## eta3 =~ y32    0.6581      NA       NA       NA
## eta3 =~ y33    0.7474      NA       NA       NA
##
## Estimated weights:
## =====
## Weight          Estimate Std. error  t-stat.  p-value
## eta1 <~ y11    0.3956      NA       NA       NA
## eta1 <~ y12    0.3873      NA       NA       NA
## eta1 <~ y13    0.4542      NA       NA       NA
## eta2 <~ y21    0.3058      NA       NA       NA
## eta2 <~ y22    0.4473      NA       NA       NA
## eta2 <~ y23    0.4735      NA       NA       NA
## eta3 <~ y31    0.4400      NA       NA       NA
## eta3 <~ y32    0.3521      NA       NA       NA
## eta3 <~ y33    0.3999      NA       NA       NA
##
## ----- Effects -----
##
## Estimated total effects:
## =====
## Total effect    Estimate Std. error  t-stat.  p-value
## eta2 ~ eta1     0.6713      NA       NA       NA
## eta3 ~ eta1     0.6634      NA       NA       NA
## eta3 ~ eta2     0.3052      NA       NA       NA
##
## Estimated indirect effects:
## =====
## Indirect effect Estimate Std. error  t-stat.  p-value
## eta3 ~ eta1     0.2049      NA       NA       NA
## -----

```

Indeed, the most common use case where inferential quantities are required is

assessment of statistical significance. `cSEM` offers two ways to compute standard errors, p-values, and confidence intervals:

1. Pass a `cSEMResults` object that does not contain resamples to `resamplecSEMResults()` and subsequently use `summarize()` or `infer()`.
2. Set argument `.resample_method` of `csem()` to `"jackknife"` or `"bootstrap"` to perform resampling and subsequently use `infer()` on the resulting object. Typically, users are only interested in statistical significance of the coefficient estimates in which case it is more convenient to call `summarize()` directly on the resulting object as `summarize()` offers a much more user-friendly print method. This is the recommend way. In fact, `csem()` internally calls `resamplecSEMResults()` which does the actual computation and `summarize()` internally calls `infer()` to compute the actual inferential quantities of interest but has the added benefit of a visually appealing print method.

Resampling is eventually always executed by `resamplecSEMResults()`. When the function is called,  $M$  resamples of the data used to compute the `cSEMResults` object to be resampled are drawn. Based on these  $M$  resample data sets, `resamplecSEMResults()` essentially calls `csem()` on each resample data set using the arguments of the original call. By default, the function returns estimates for a subset of the resampled parameters/statistics computed by `csem()`. Currently, the following estimates are computed and returned by default based on each resample: path estimates, loading estimates, weight estimates, measurement error correlations, indicator correlations, construct correlations, total effects, and indirect effects. In some applications, users may need the standard errors or confidence intervals for a specific statistic – e.g, the differences between path coefficients  $\beta_1$  and  $\beta_2$ . Such statistics may be provided to `resamplecSEMResults()` or `csem()` as a function `fun(.object, ...)` or a list of such functions via the `.user_funs` argument. The first argument of these functions must always be a `cSEMResults` object. Internally, the function will be applied on each resample to produce the desired statistic. Hence, resamples for arbitrary complicated statistics may be computed – as long as the body of the function draws on elements contained in the `cSEMResults` object. Output of the function `fun(.object, ...)` should preferably be a (named) vector, but matrices are also accepted; the output will be vectorized (columnwise) in this case.

Both, resampling the original `cSEMResults` object ("first resample") and resampling based on a resampled `cSEMResults` object ("second resample") are supported. Resampling based on a resample is suppressed by default (`.resample_method2 = "none"`) as it *significantly* increases computation time; there are now  $M \cdot M2$  resamples to compute. Resamples of a resample are required, e.g., for the studentized confidence interval discussed in Subsection 5.5.2 (e.g., [Hesterberg, 2015](#)). Typically, bootstrap resamples are used in this case ([Davison, 1997](#)).

As `csem()` accepts a single data set, a list of  $G$  data sets, as well as data sets that contain a column name used to split the data into  $G$  groups, the `cSEMResults` object may contain multiple data sets. In this case, resampling is done by data set or group. Note that depending on the number of data sets/groups, the computation will be considerably slower as resampling will be repeated for each data set/group, resulting in a total of  $G \cdot M$  resample runs. However, apart from speed considerations, users do not need to worry about the type of input used to compute the `cSEMResults` object; `resamplecSEMResults()` is able to deal with each case.

The number of bootstrap runs for the first and second run are controlled by the `.R` and `.R2` argument. The default is 499 for the first and 199 for the second run but should be increased in real applications. See e.g. [Hesterberg \(2015\)](#), [Davison \(1997\)](#) or [Efron and Hastie \(2016\)](#) for recommendations. For jackknife `.R` and `.R2` are ignored. Resampling may produce inadmissible results (as checked by `verify()`). By default these results are dropped however users may choose to *"ignore"* or *"replace"* inadmissible results in which case resampling continues until the necessary number of admissible results is reached. The behavior is controlled by `csem()`'s the `.handle_inadmissibles` argument.

Both bootstrap and jackknife resampling support platform-independent multiprocessing as well as random seeds via the *future framework* ([Bengtsson, 2018](#)). To enable multiprocessing set `.eval_plan = "multiprocess"`. In this case, the maximum number of available cores is used if on a Unix-type system. On Windows, as many separate R instances are opened in the background as there are cores available instead. Note that this naturally has some overhead. Consequently, for a small number of resamples multiprocessing will generally not be faster compared to sequential (single core) processing (`.eval_plan = "sequential"`; the default). Seeds are set via the `.seed` argument. If no seed is set, `csem()` randomly picks a

seed, and saves it to be able to replicate resampling at a later point.

```
# Option 1
res_boot <- csem(.data = threecommonfactors, .model = model,
               .resample_method = "bootstrap",
               .R = 599,
               .handle_inadmissibles = "replace")

# Option 2
res_boot2 <- resamplecSEMResults(res1) # res1 was computed at the beginning of Section 5.4.
```

```
summarize(res_boot)

## -----
## ----- Overview -----
##
## General information:
## -----
## Estimation status           = Ok
## Number of observations      = 500
## Weight estimator           = PLS-PM
## Inner weighting scheme     = "path"
## Type of indicator correlation = Pearson
## Path model estimator       = OLS
## Second-order approach      = NA
## Type of path model         = Linear
## Disattenuated              = Yes (PLSc)
##
## Resample information:
## -----
## Resample method             = "bootstrap"
## Number of resamples        = 599
## Number of admissible results = 599
## Approach to handle inadmissibles = "replace"
## Sign change option         = "none"
## Random seed                 = -378747585
##
## Construct details:
## -----
## Name  Modeled as   Order   Mode
##
## eta1  Common factor First order "modeA"
## eta2  Common factor First order "modeA"
## eta3  Common factor First order "modeA"
##
## ----- Estimates -----
##
## Estimated path coefficients:
```

```

## =====
##
##                                     CI_percentile
## Path           Estimate Std. error  t-stat.  p-value    95%
## eta2 ~ eta1     0.6713    0.0435  15.4209  0.0000 [ 0.5874; 0.7548 ]
## eta3 ~ eta1     0.4585    0.0821   5.5879  0.0000 [ 0.2755; 0.6119 ]
## eta3 ~ eta2     0.3052    0.0848   3.5973  0.0003 [ 0.1368; 0.4793 ]
##
## Estimated loadings:
## =====
##                                     CI_percentile
## Loading         Estimate Std. error  t-stat.  p-value    95%
## eta1 =~ y11     0.6631    0.0382  17.3720  0.0000 [ 0.5877; 0.7321 ]
## eta1 =~ y12     0.6493    0.0402  16.1432  0.0000 [ 0.5690; 0.7284 ]
## eta1 =~ y13     0.7613    0.0306  24.8648  0.0000 [ 0.6985; 0.8172 ]
## eta2 =~ y21     0.5165    0.0552   9.3508  0.0000 [ 0.4098; 0.6157 ]
## eta2 =~ y22     0.7554    0.0350  21.5743  0.0000 [ 0.6885; 0.8207 ]
## eta2 =~ y23     0.7997    0.0388  20.6265  0.0000 [ 0.7173; 0.8687 ]
## eta3 =~ y31     0.8223    0.0326  25.2295  0.0000 [ 0.7577; 0.8813 ]
## eta3 =~ y32     0.6581    0.0395  16.6622  0.0000 [ 0.5756; 0.7290 ]
## eta3 =~ y33     0.7474    0.0399  18.7099  0.0000 [ 0.6643; 0.8181 ]
##
## Estimated weights:
## =====
##                                     CI_percentile
## Weight          Estimate Std. error  t-stat.  p-value    95%
## eta1 <~ y11     0.3956    0.0215  18.4025  0.0000 [ 0.3552; 0.4361 ]
## eta1 <~ y12     0.3873    0.0190  20.3482  0.0000 [ 0.3501; 0.4272 ]
## eta1 <~ y13     0.4542    0.0190  23.8970  0.0000 [ 0.4182; 0.4921 ]
## eta2 <~ y21     0.3058    0.0290  10.5440  0.0000 [ 0.2483; 0.3576 ]
## eta2 <~ y22     0.4473    0.0219  20.3788  0.0000 [ 0.4065; 0.4899 ]
## eta2 <~ y23     0.4735    0.0214  22.1343  0.0000 [ 0.4321; 0.5201 ]
## eta3 <~ y31     0.4400    0.0182  24.1682  0.0000 [ 0.4067; 0.4760 ]
## eta3 <~ y32     0.3521    0.0188  18.7126  0.0000 [ 0.3139; 0.3876 ]
## eta3 <~ y33     0.3999    0.0195  20.5096  0.0000 [ 0.3634; 0.4389 ]
##
## ----- Effects -----
##
## Estimated total effects:
## =====
##                                     CI_percentile
## Total effect    Estimate Std. error  t-stat.  p-value    95%
## eta2 ~ eta1     0.6713    0.0435  15.4209  0.0000 [ 0.5874; 0.7548 ]
## eta3 ~ eta1     0.6634    0.0410  16.1811  0.0000 [ 0.5898; 0.7416 ]
## eta3 ~ eta2     0.3052    0.0848   3.5973  0.0003 [ 0.1368; 0.4793 ]
##
## Estimated indirect effects:
## =====

```

```
##
##      Indirect effect      Estimate  Std. error   t-stat.   p-value      CI_percentile
##      eta3 ~ eta1           0.2049     0.0577     3.5531    0.0004 [ 0.0951; 0.3291 ]
## -----
```

As discussed in Subsection 5.5.2, several confidence intervals are implemented and available via the `.ci` argument of `summarize()`. By default, the percentile confidence interval is computed. Using `summariz()` is the most convenient way to obtain a confidence interval, however, it is also possible to compute one or several confidence intervals directly via `infer()`. We may, for instance, want to compare the 95% and the 99% normal-based ( $z$ ) confidence interval with the percentile confidence interval of the path coefficient estimates.

```
ii <- infer(res_boot, .quantity = c("CI_standard_z", "CI_percentile"), .alpha = c(0.01, 0.05))
ii$Path_estimates

## $CI_standard_z
##      eta2 ~ eta1 eta3 ~ eta1 eta3 ~ eta2
## 99%L  0.5589022  0.2446665  0.08803619
## 99%U  0.7831749  0.6673769  0.52503894
## 95%L  0.5857133  0.2952003  0.14027856
## 95%U  0.7563638  0.6168431  0.47279657
##
## $CI_percentile
##      eta2 ~ eta1 eta3 ~ eta1 eta3 ~ eta2
## 99%L  0.5556889  0.2333521  0.07165315
## 99%U  0.7877641  0.6785145  0.53492055
## 95%L  0.5874035  0.2754634  0.13678684
## 95%U  0.7547567  0.6118670  0.47927965
```

## 5.5 Postestimation

As of version 0.2.0, `cSEM` provides five major postestimation functions, four test-family functions, and four do-type functions:

Table 5.2: Available postestimation functions (`cSEM`, version 0.2.0)

Major functions	Test-family	Do-family
<ul style="list-style-type: none"><li>• <code>assess()</code></li><li>• <code>infer()</code></li><li>• <code>predict()</code></li><li>• <code>summarize()</code></li><li>• <code>verify()</code></li></ul>	<ul style="list-style-type: none"><li>• <code>testOMF()</code></li><li>• <code>testMICOM()</code></li><li>• <code>testMGD()</code></li><li>• <code>testHausman()</code></li></ul>	<ul style="list-style-type: none"><li>• <code>doFloodlightAnalysis()</code></li><li>• <code>doSurfaceAnalysis</code></li><li>• <code>doRedundancyAnalysis()</code></li><li>• <code>doIPMA()</code></li></ul>

Some of the postestimation functions have already been mentioned in the previous sections. This section covers the major postestimation functions and the test-family functions in more detail. The do-family of functions are rather specialized and therefore not discussed here.<sup>10</sup>

All postestimations functions expect a `cSEMResults` object. Technically, however, the precise behavior of each postestimation function is determined by the second and third class attribute, namely `cSEMResults_default`, `cSEMResults_multi`, `cSEMResults_2ndorder`, and `cSEMResults_resampled`. Currently, some of the postestimation functions can not (yet) deal with models containing nonlinear terms. Similarly, not all functions accept objects of class `cSEMResults_2ndorder`.

Eventually, postestimation function will accept all possible types of models. Table 5.3 shows the current state of the functions in terms of models they currently accept.

For illustration, I use the `Switching` data set throughout this section. The model to be estimated is:

```
model <- "  
# Measurement models  
INV =~ INV1 + INV2 + INV3 + INV4  
SAT =~ SAT1 + SAT2 + SAT3  
INT =~ INT1 + INT2  
  
# Structural model  
INT ~ INV + SAT  
"
```

<sup>10</sup>The reader is referred to the package website: <https://m-e-rademaker.github.io/cSEM/> for details; last accessed: 22.04.2020.



Table 5.3: State of the postestimation functions as of version 0.2.0.

Postestimation function	Linear model	Class			
		<i>cSEMResults_default</i>	<i>cSEMResults_multi</i>	<i>cSEMResults_2ndorder</i>	
<code>assess()</code>	Yes	✓	✓/✓	✓	✓
	No	not yet	not yet / not yet	not yet	not yet
<code>infer()</code>	Yes	✓	✓/✓	✓	✓
	No	✓	✓/✓	✓	✓
<code>predict()</code>	Yes	✓	✓/ not yet	not yet	not yet
	No	not yet	not yet / not yet	not yet	not yet
<code>summarize()</code>	Yes	✓	✓/✓	✓	✓
	No	✓	✓/✓	✓	✓
<code>verify()</code>	Yes	✓	✓/✓	✓	✓
	No	✓	✓/✓	✓	✓
<code>testOMF()</code>	Yes	✓	✓/✓	✓	✓
	No	✓	✓/✓	✓	✓
<code>testMICOM()</code>	Yes	-	✓/ not yet	-	-
	No	-	✓/ not yet	-	-
<code>testMGD()</code>	Yes	-	✓/✓	-	-
	No	-	✓/✓	-	-
<code>testHausman()</code>	Yes	✓	not yet / not yet	not yet	not yet
	No	not yet	not yet / not yet	not yet	not yet
<code>doFloodlightAnalysis()</code>	Yes	-	-/-	-	-
	No	✓	✓/✓	✓	✓
<code>doSurfaceAnalysis()</code>	Yes	-	-/-	-	-
	No	✓	✓/✓	✓	✓
<code>doIPMA()</code>	Yes	✓	✓/ not yet	not yet	not yet
	No	not yet	not yet / not yet	not yet	not yet

Notes: “✓” := the postestimation function handles this case with a meaningful computation; “not yet” := not yet implemented, but the user is informed by an error message; “-” := not meaningful due to the nature of the task the postestimation function performs. In column *cSEMResults\_multi*: values before the “/” refer to the case where all list elements are of class *cSEMResults\_default*; values after the “/” refer to the case where all list elements are of class *cSEMResults\_2ndorder*.

To obtain inferential quantities, bootstrap with the default of 499 runs is used. Inadmissible results are replaced on the fly to ensure that the number of admissible results is sufficient for reliable inference. Moreover, for reproducibility, I use the random number 170818 as the seed.

```
out_switching <- csem(  
  .data           = Switching,  
  .model          = model,  
  .resample_method = "bootstrap",  
  .handle_inadmissibles = "replace",  
  .seed           = 170818  
)
```

### 5.5.1 `assess()`

```
assess(  
  .object           = NULL,  
  .quality_criterion = c(  
    "all", "ave", "chi_square", "chi_square_df",  
    "cfi", "dg", "df", "dl", "dml", "effects", "f2", "fl_criterion",  
    "gfi", "gof", "htmt", "ifi", "nfi", "nnfi", "r2", "r2_adj",  
    "reliability", "rho_C", "rho_C_mm", "rho_C_weighted",  
    "rho_C_weighted_mm", "rho_T", "rho_T_weighted",  
    "rmsea", "rms_theta", "srmr", "vif", "vifmodeB"),  
  .only_common_factors = TRUE,  
  ...  
)
```

As indicated by the name, `assess()` is used to assess an estimated model.

In `cSEM`, model assessment is considered to be any task that in some way or another seeks to assess the quality of the estimated model *without conducting a statistical test* – tests are covered by the test-family of functions. Quality, in this case, is taken to be a catch-all term for all common aspects of model assessment; this mainly comprises fit indices, reliability estimates, common validity assessment criteria, effect sizes, and other related quality measures/indices that do not rely on a formal test procedure. In `cSEM` a generic (fit) index, quality, or assessment measure is referred to as a **quality criterion**.

Technically, the `assess()` function is essentially a wrapper around a number of functions that perform a particular assessment task. These functions can be called

directly, however, by design its more convenient to use `assess()` instead. An exception are the tasks implemented via a do-family function. Due to the nature of the task these procedures perform – for example `doIMPA()` relies on graphical analysis –, they are not meaningfully integrated into a wrapper function like `assess()`. They are therefore treated as independent functions, which must be called separately if required.

By default, every possible quality criterion is calculated (`.quality_criterion = "all"`). If only a subset of quality criteria are needed, a single character string or a vector of character strings naming the criteria to be computed may be supplied to `assess()` via the `.quality_criterion` argument. Tables 5.4 and 5.5 list the available quality criteria alongside a brief description.

Some assessment measures are inherently tied to the common factor model. It is therefore unclear how to interpret their results in the context of a composite model. Consequently, their computation is suppressed by default for constructs modeled as composites. Currently, this applies to the following quality criteria:

- AVE and validity assessment based theron (i.e., the Fornell-Larcker criterion)
- HTMT and validity assessment based theron
- All reliability measures

It is possible to force computation of all quality criteria for constructs modeled as composites using `.only_common_factors = FALSE`, however, interpreting results in analogy to a common factor is conceptually questionable in this case.

It is possible to control the exact behavior of the functions which are called by `assess()` by passing arguments down to these functions via `assess()`'s `"..."` argument. To see what arguments are available, the reader is referred to the [online help file](#) for `assess()`.<sup>11</sup> In particular the description of the `"..."` argument.

Lastly, all quality criteria assume that the estimated loadings, construct correlations and path coefficients involved in the computation of a specific quality criterion are consistent estimates for their theoretical population counterpart. If the user deliberately chooses an approach that yields inconsistent estimates – by setting

---

<sup>11</sup>See: <https://m-e-rademaker.github.io/cSEM/reference/assess.html>; last accessed: 12.04.2020.

Table 5.4: Quality criteria computed by `assess()` - Part I

Quality criterion	Description	cSEM name	Calculated by
Average variance extracted (AVE)	An estimate of the amount of variation in the indicators that is due to the latent variable (Formell and Larcker, 1981).	"ave"	<code>calculateAVE()</code>
Degrees of freedom	The difference between the number of non-redundant off-diagonal elements of the indicator correlation matrix $\mathbf{S}$ and the model parameters. Details for common factor models are discussed in e.g., Bollen (1989b). For the composite model, see Schubert et al. (2018a) and Chapter 2, Section 2.2.	"df"	<code>calculateDf()</code>
Distance measures	Mathematical functions that assess the distance between two matrices (typically the empirical indicators correlation matrix $\mathbf{S}$ and the estimated model-implied indicator correlation matrix $\hat{\Sigma}$ ). Implemented distance measures: the geodesic distance ( $d_G$ ), the squared Euclidian distance ( $d_L$ ), the maximum likelihood distance ( $d_{ML}$ ). See Chapter 2, Section 2.4 for details.	"dg", "dl", "dml"	<code>calculateDG()</code> <code>calculateDL()</code> <code>calculateDML()</code>
Effect estimates	The estimated indirect and total effects of the variables of the structural model. See e.g., Bollen (1987).	"effects"	<code>calculateEffects()</code>
Effect sizes (Cohens $f^2$ )	An index of the effect size of an independent variable on a dependent variable in a structural regression equation. Commonly known as Cohen's $f^2$ (Cohen, 1992).	"f2"	<code>calculatef2()</code>
Fit measures	Measures of (mis)fit commonly encountered in the SEM literature. Currently, supported indices: $\chi^2$ , $\chi^2/df$ , comparative fit index (CFI), goodness of fit index (GFI), incremental fit index (IFI), normed fit index (NFI), non-normed fit index / Tucker-Lewis index (NNFI/TLI), root mean square error of approximation (RMSEA), root mean square outer residual covariance (RMS $\theta$ ), standardized root mean square residual (SRMR). For detail, see Chapter 2, Section 2.4.	"chi_square", "chi_square_df", "cfi", "gfi", "ifi", "nfi", "nnfi", "rmsea", "rms_theta", "srmr"	<code>calculateChiSquare()</code> <code>calculateChiSquareDf()</code> <code>calculateCFI()</code> <code>calculateGFI()</code> <code>calculateIFI()</code> <code>calculateNFI()</code> <code>calculateNNFI()</code> <code>calculateRMSTheta()</code> <code>calculateSRMR()</code>

Table 5.5: Quality criteria computed by `assess()` - Part II

Quality criterion	Description	cSEM name	Calculated by
Fornell-Larcker criterion	A rule suggested by <a href="#">Fornell and Larcker (1981)</a> to assess discriminant validity.	<code>"fl_criterion"</code>	<code>calculateFLCriterion()</code>
Goodness-of-Fit (GoF)	The GoF was suggested by <a href="#">Tenenhaus et al. (2004)</a> as a measure of model fit. Contrary to what the name suggests, the GoF is <b>not</b> a measure of overall model fit in the SEM sense (e.g., <a href="#">Henseler et al., 2012</a> ).	<code>"gof"</code>	<code>calculateGoF()</code>
The heterotrait-monotrait ratio of correlations (HTMT)	An estimate of the correlation between latent variables. The HTMT is used to assess convergent and/or discriminant validity of a construct. It was introduced by <a href="#">Henseler et al. (2015)</a> . It is also possible to obtain confidence intervals for the HTMT estimates.	<code>"htmt"</code>	<code>calculateHTMT()</code>
$R^2$ and $\bar{R}^2$	The $R^2$ and the adjusted $R^2$ of the structural equations. Computed by <code>csem()</code> but, being a quality criteria, available via <code>assess()</code> .	<code>"r2"</code> , <code>"r2_adj"</code>	<code>csem()</code>
Reliability	Computes several reliability estimates such as the Jöreskogs $\rho$ ( <a href="#">Jöreskog, 1971</a> ), Cronbach's $\alpha$ ( <a href="#">Cronbach, 1951</a> ) and Dijkstra-Henseler's $\rho_A$ ( <a href="#">Dijkstra and Henseler, 2015b</a> ). See <a href="#">Cho (2016)</a> for the terminology that cSEM's reliability terminology is based on.	<code>"reliability"</code> , <code>"rho_C"</code> , <code>"rho_C_mm"</code> , <code>"rho_C_weighted"</code> , <code>"rho_C_weighted_mm"</code> , <code>"rho_T"</code> , <code>"rho_T_weighted"</code>	<code>calculateRhoC()</code> , <code>calculateRhoT()</code>
Variance inflation factors (VIFs)	An index for the amount of (multi-)collinearity between independent variables of a regression equation. Computed by <code>csem()</code> but, being a quality criteria, available via <code>assess()</code> .	<code>"vif"</code>	<code>csem()</code>
VIFs for PLS mode B weights	An index for the amount of (multi-)collinearity between independent variables (indicators) in mode B regression equations.	<code>"vifmodeB"</code>	<code>calculateVifModeB()</code>

`.disattenuate = FALSE` in `csem()` when the estimated model contains common factors –, `assess()` will still estimate all quantities; however, quantities such as the AVE or the congeneric reliability ( $\rho_C$ ) inherit inconsistency and should thus not be interpreted unless there is evidence indicating the inconsistency is not severe.

## Example

To compute all possible quality criteria for the `out_switching` object, simply call `assess()` on that object without any additional arguments.:

```
assess(out_switching)

## -----
##
## Construct      AVE      R2      R2_adj
## INV           0.4391    NA      NA
## SAT           0.6297    NA      NA
## INT           0.5972    0.4519  0.4505
##
## ----- Common (internal consistency) reliability estimates -----
##
## Construct Cronbachs_alpha  Joereskogs_rho  Dijkstra-Henselers_rho_A
## INV       0.7452           0.7483           0.7827
## SAT       0.8362           0.8334           0.8561
## INT       0.7408           0.7464           0.7561
##
## ----- Alternative (internal consistency) reliability estimates -----
##
## Construct      RhoC      RhoC_mm  RhoC_weighted
## INV           0.7483     0.7357    0.7827
## SAT           0.8334     0.8197    0.8561
## INT           0.7464     0.7464    0.7561
##
## Construct      RhoC_weighted_mm  RhoT  RhoT_weighted
## INV           0.7827           0.7452  0.7148
## SAT           0.8561           0.8362  0.8397
## INT           0.7561           0.7408  0.7394
##
## ----- Distance and fit measures -----
##
## Geodesic distance      = 0.08620259
## Squared Euclidian distance = 0.104884
## ML distance            = 0.4604675
##
## Chi_square            = 352.7181
## Chi_square_df        = 14.69659
```

```

## CFI = 0.8588499
## CN = 80.08274
## GFI = 0.881781
## IFI = 0.8595735
## NFI = 0.85085
## NNFI = 0.7882748
## RMSEA = 0.1337186
## RMS_theta = 0.08900008
## SRMR = 0.0482779
##
## Degrees of freedom = 24
##
## ----- Model selection criteria -----
##
## Construct      AIC      AICc      AICu
## INT      -456.2132    312.8393    -453.2073
##
## Construct      BIC      FPE      GM
## INT      -442.2857    0.5517    785.4465
##
## Construct      HQ      HQc      Mallows_Cp
## INT      -450.8523    -450.7777    4.5190
##
## ----- Variance inflation factors (VIFs) -----
##
## Dependent construct: 'INT'
##
## Independent construct  VIF value
## INV                    1.1071
## SAT                    1.1071
##
## ----- Effect sizes (Cohen's f^2) -----
##
## Dependent construct: 'INT'
##
## Independent construct  f^2
## INV                    0.3024
## SAT                    0.2660
##
## ----- Validity assessment -----
##
## Heterotrait-monotrait ratio of correlations matrix (HTMT matrix)
##
##          INV      SAT INT
## INV 1.0000000 0.0000000 0
## SAT 0.3025275 1.0000000 0
## INT 0.5549424 0.5293116 1

```

```

##
##
## Fornell-Larcker matrix
##
##          INV          SAT          INT
## INV 0.43913216 0.09672899 0.3061243
## SAT 0.09672899 0.62965408 0.2861872
## INT 0.30612425 0.28618719 0.5971781
##
##
## ----- Effects -----
##
## Estimated total effects:
## =====
## Total effect   Estimate   Std. error   t-stat.   p-value
## INT ~ INV      0.4283     0.0430      9.9694    0.0000
## INT ~ SAT     -0.4017     0.0424     -9.4832    0.0000
## -----

```

To pick only a subset of quality criteria, use a vector of names:

```

assess(out_switching, .quality_criterion = c("dl", "dg", "dml"))
## -----
##
## ----- Distance and fit measures -----
##
## Geodesic distance           = 0.08620259
## Squared Euclidian distance = 0.104884
## ML distance                 = 0.4604675
##
## -----

```

To compute, for instance, the critical 99%-quantiles for the HTMT, supply the relevant arguments in place of the “...” argument:

```

assess(out_switching, .quality_criterion = "htmt", .inference = TRUE, .alpha = 0.01)
## -----
##
## ----- Validity assessment -----
##
## Heterotrait-monotrait ratio of correlations matrix (HTMT matrix)
##
## Values in the upper triangular part are the 99%-quantile of the
## bootstrap distribution (using .ci = 'CI_percentile').
##

```



```
##          INV          SAT          INT
## INV 1.0000000 0.4043013 0.6515852
## SAT 0.3025275 1.0000000 0.6219331
## INT 0.5549424 0.5293116 1.0000000
##
## -----
```

If the user wants to access the quality criteria, she must first assign the result of the call to `assess()` to a name. The resulting object is a list of class `cSEMAssess`. As usual, list elements can be accessed using the `$` operator:

```
res_assess <- assess(out_switching)
names(res_assess)

## [1] "Information"      "AVE"              "AIC"              "AICc"
## [5] "AICu"             "BIC"              "FPE"              "GM"
## [9] "HQ"               "HQc"              "Mallows_Cp"       "RhoC"
## [13] "RhoC_mm"          "RhoC_weighted"    "RhoC_weighted_mm" "DG"
## [17] "DL"               "DML"              "Df"               "Effects"
## [21] "F2"               "Chi_square"       "Chi_square_df"    "CFI"
## [25] "GFI"              "CN"               "IFI"              "NFI"
## [29] "NNFI"             "RMSEA"            "RMS_theta"        "SRMR"
## [33] "Fornell-Larcker" "GoF"              "HTMT"             "R2"
## [37] "R2_adj"           "Reliability"      "RhoT"             "RhoT_weighted"
## [41] "VIF"              "VIF_modeB"
```

And now for example:

```
res_assess$SRMR

## [1] 0.0482779

res_assess$HTMT

##          INV          SAT          INT
## INV 1.0000000 0.0000000 0
## SAT 0.3025275 1.0000000 0
## INT 0.5549424 0.5293116 1
```

## 5.5.2 infer()

```
infer(  
  .object      = NULL,  
  .quantity    = c(  
    "all", "mean", "sd", "bias", "CI_standard_z", "CI_standard_t",  
    "CI_percentile", "CI_basic", "CI_bc", "CI_bca", "CI_t_interval"),  
  .alpha       = 0.05,  
  .bias_corrected = TRUE  
)
```

Similar to `assess()`, `infer()` is a convenience wrapper around a number of internal, i.e., non-exported functions. These functions compute a particular inferential quantity, i.e., a value or set of values to be used in statistical inference. As mentioned when discussing statistical inference in Section 5.4, `cSEM` relies on resampling (bootstrap and jackknife) as the basis for the computation of e.g., standard errors or confidence intervals. Consequently, `infer()` requires resamples to work. Technically, the `cSEMResults` object used in the call to `infer()` must therefore also have class attribute `cSEMResults_resampled`. If the object provided by the user does not contain resamples yet, `infer()` will obtain bootstrap resamples first; a warning is given to inform the user since computation will take longer in this case.

With regard to the goal of user-friendliness, `infer()` does as much as possible in the background. Hence, every time `infer()` is called on a `cSEMResults` object, the quantities chosen by the user are automatically computed for every estimated parameter contained in the object. By default, all possible quantities are computed (`.quantity = all`). Table 5.6 lists the available inferential quantities alongside a brief description. Implementation of the confidence intervals is based on [Hesterberg \(2015\)](#) and [Davison \(1997\)](#).

By default, all but the studentized t-interval confidence interval and the bias-corrected and accelerated confidence interval are calculated. The reason for excluding these quantities by default is that both require an additional resampling step.

As of version 0.2.0 of the package, `infer()` does not have a very user-friendly print method yet – at least according to what I consider as user-friendly. Future versions

Table 5.6: Inferential quantities computed by `infer()`

Inferential quantity	Description	cSEM name
Resample mean	The mean over all $M$ resample estimates of a generic statistic or parameter.	<code>"mean"</code>
Resample standard deviation	The standard deviation based on $M$ resample estimates of a generic statistic or parameter.	<code>"sd"</code>
Resample bias	The difference between the resample mean and the original estimate of a generic statistic or parameter.	<code>"bias"</code>
Standard confidence interval	The standard confidence interval of a generic statistic or parameter with standard errors estimated by the resample standard deviation. While <code>"CI_standard_z"</code> assumes a standard normally distributed statistic, <code>"CI_standard_t"</code> assumes a t-statistic with $N - 1$ degrees of freedom.	<code>"CI_standard_z"</code> <code>"CI_standard_t"</code>
Percentile confidence interval	The lower and upper bounds of the confidence interval are estimated as the $\alpha$ and $1 - \alpha$ quantiles of the distribution of the resample estimates.	<code>"CI_percentile"</code>
Basic confidence interval	The basic confidence interval is also called the reverse bootstrap percentile confidence interval. See <a href="#">Hesterberg (2015, p.381)</a> for details.	<code>"CI_basic"</code>
Bias-corrected confidence interval	The bias-corrected confidence interval. See <a href="#">Davison (1997)</a> for details.	<code>"CI_bc"</code>
Bias-corrected and accelerated confidence interval	The bias-corrected and accelerated confidence interval. Requires additional jackknife resampling to compute the influence values. See <a href="#">Davison (1997)</a> for details.	<code>"CI_bca"</code>
Studentized confidence interval	If based on bootstrap resamples the interval is also called the bootstrap t-interval confidence interval ( <a href="#">Hesterberg, 2015, p.381</a> ). Requires resamples of resamples. See Section <a href="#">5.4</a> for how to obtain these.	<code>"CI_t_interval"</code>

will likely contain improvements.<sup>12</sup>

<sup>12</sup>As already mentioned in Section [5.4](#), for users interested in the estimated standard errors, t-values, p-values and/or confidence intervals of the path, weight, or loading estimates, calling `summarize()` directly will usually be more convenient. Hence, the typical user may never actually invoke `infer()` directly.

## Example

To compute, for example, the percentile and the bias-corrected and accelerated (Bca) confidence interval for the estimates in the `out_switching` object, call `infer()` on the object and select the confidence intervals using the `.quantity` argument. Note that `out_switching` already contains resamples, however, the Bca confidence interval requires an additional round of jackknife resampling to compute the leverage values necessary for its computation.

```
out_infer <- infer(out_switching, .quantity = c("CI_percentile", "CI_bca"))
out_infer$Path_estimates

## $CI_percentile
##      INT ~ INV  INT ~ SAT
## 95%L 0.3494161 -0.4974419
## 95%U 0.5124118 -0.3244157
##
## $CI_bca
##      INT ~ INV  INT ~ SAT
## 95%L 0.3483501 -0.4934969
## 95%U 0.5112962 -0.3211290
```

### 5.5.3 `predict()`

```
predict(
  .object          = NULL,
  .benchmark      = c("lm", "unit", "PLS-PM", "GSCA", "PCA", "MAXVAR"),
  .cv_folds       = 10,
  .handle_inadmissibles = c("stop", "ignore", "set_NA"),
  .r              = 10,
  .test_data      = NULL
)
```

`predict()` uses the procedure introduced by [Shmueli et al. \(2016\)](#) in the context of PLS-PM – commonly called “PLSPredict” by the authors (e.g., [Shmueli et al., 2019](#)). Upon providing a `cSEMResults` object to `predict()`,  $k$ -fold cross-validation is conducted to randomly split the data which was used to obtain the `cSEMResults` object into training and test data. Subsequently, the relevant values in the test data are predicted based on the model parameter estimates obtained using the training data. The number of cross-validation folds is `.cv_folds = 10` by default. Likewise,

the procedure is repeated `.r = 10` times to avoid irregularities due to a particular split. See [Shmueli et al. \(2019\)](#) for details.

Alternatively, users may supply a matrix of `.test_data` with the same column names as those in the data used to obtain the `cSEMResults` object (the training data). In this case, arguments `.cv_folds` and `.r` are ignored and `predict()` uses the estimated coefficients from the `cSEMResults` object to predict the values in the columns of `.test_data`.

In [Shmueli et al. \(2016\)](#), PLS-based predictions for a generic indicator are compared to the predictions based on a multiple regression of the indicator on all available exogenous indicators (`.benchmark = "lm"`) and a simple mean-based prediction summarized in the so-called  $Q^2_{\text{predict}}$  metric ([Shmueli et al., 2019](#), p.2338). `predict()` is more general in that it allows users to compare the predictions based on a so-called target model/specification to predictions based on an alternative benchmark. Besides multiple regression, available benchmarks include predictions based on a model estimated using PLS-PM weights, unit weights (i.e., sum scores), GSCA weights, PCA weights, and MAXVAR weights. In all cases, the available comparison metrics are the mean absolute error (MAE) and the root mean square error (RMSE).

When conducting cross-validation, each estimation run is checked for admissibility using `verify()`. If an estimation yields inadmissible results, `predict()` stops with an error (`.handle_inadmissibles = "stop"`). Users may choose to *"ignore"* inadmissible results in which case they are included, or to simply set predictions to NA (`.handle_inadmissibles = "set_NA"`) for the particular run that failed. In this case, some predictions may not be available. Use with care.

## Example

For illustration, I use the `Switching` data set again. In this case, we compare the predictions of the indicator values for indicators INT1 and INT2 based on the postulated model with those of a simple multiple regression of INT1 and INT2 on all other indicators (INV1, ..., SAT3).

```
predict(out_switching)

## -----
## ----- Overview -----
##
```

```

## Number of obs. training      = 693
## Number of obs. test         = 74
## Number of cv folds          = 10
## Number of repetitions       = 10
## Handle inadmissibles        = stop
## Target                       = 'PLS-PM'
## Benchmark                    = 'lm'
##
## ----- Prediction metrics -----
##
##
## Name      MAE target  MAE benchmark  RMSE target  RMSE benchmark  Q2_predict
## INT1      0.5995     0.6422         0.8346       0.8849        0.1845
## INT2      0.6326     0.6972         0.8470       0.9426        0.2618
## -----

```

Both predictions metrics show lower values for the target model-based predictions – compared to a benchmark simple linear regression model. According to [Shmueli et al. \(2019\)](#), this can be taken as evidence that imposing a structure on the relationship between observables has some merits in terms of out-of-sample predictive accuracy.<sup>13</sup> There is also a method for the generic function `plot()` available which plots actual vs. predicted values. Moreover, the density of the prediction residuals and of the predicted values can be displayed.

```

pred_switching <- predict(out_switching)
plot_switching <- plot(pred_switching)

```

And now for example:

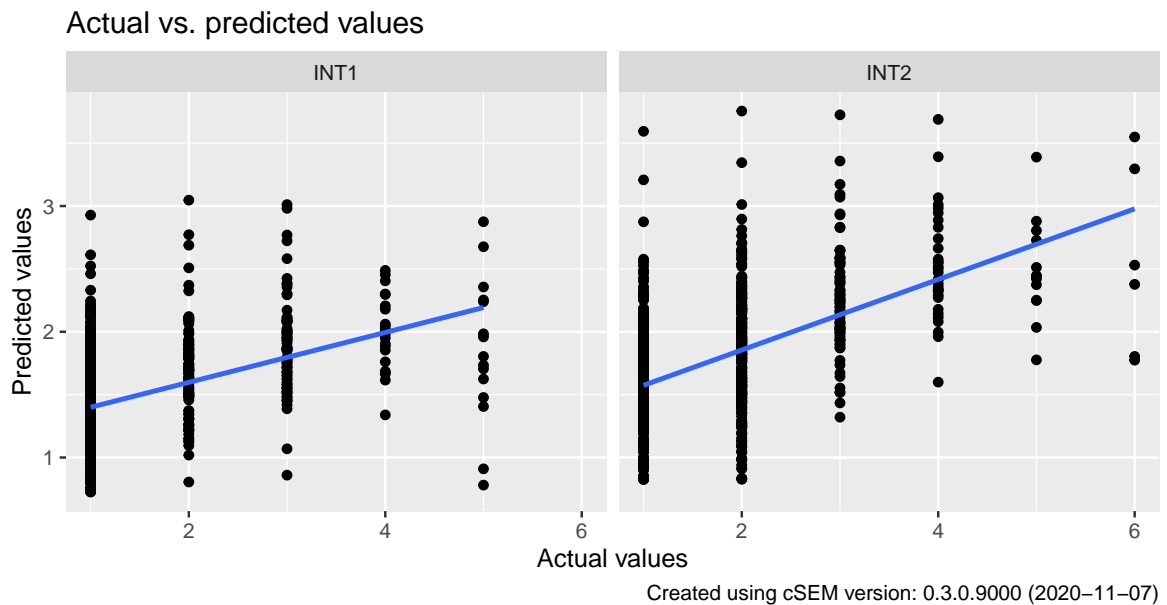
```

plot_switching$`Actual vs. predicted`

```

---

<sup>13</sup>In fact, the authors go much further by claiming that out-of-sample predictive accuracy somehow translates into causal evidence. They position the procedure they label “PLSPredict” as a “...’causal-predictive’ approach to structural equation modeling (SEM), designed to overcome the apparent dichotomy between explanation and prediction.” ([Shmueli et al., 2019](#), p.2322). This is a strong assertion that, as I have discussed in [Chapter 2](#), lacks a theoretical foundation.



#### 5.5.4 `summarize()`

```
summarize(
  .object = NULL,
  .alpha = 0.05,
  .ci     = NULL,
  ...
)
```

The function is mainly called for its side effect: the printing of a structured summary of the estimates. The resulting object has class *cSEMSummarize*. Objects of that class are virtually identical to *cSEMResults* objects. The main difference is that path, loading, weight, and effect estimates are returned as data frames instead of matrices. The data frame format is usually much more convenient if users intend to present the results in e.g., a paper or a presentation.

If `.object` contains resamples, standard errors, t-values and p-values – assuming estimates are standard normally distributed – are computed and printed. By default, the percentile confidence interval is given as well. For other confidence intervals the `.ci` argument is used. See Table 5.6 in Subsection 5.5.2 for possible choices.

#### Example

We use the `Switching` data set again.

```
sum_csem <- summarize(out_switching, .ci = "CI_standard_z")
```

Users are often only interested in path, loading, and weight estimates. It is possible to print a smaller output which comprises only these estimates by setting the `.full_output` argument of `print()` to `FALSE`.

```
print(sum_csem, .full_output = FALSE)

## -----
## ----- Overview -----
##
## General information:
## -----
## Estimation status           = Ok
## Number of observations      = 767
## Weight estimator           = PLS-PM
## Inner weighting scheme     = "path"
## Type of indicator correlation = Pearson
## Path model estimator       = OLS
## Second-order approach      = NA
## Type of path model         = Linear
## Disattenuated              = Yes (PLSc)
##
## Resample information:
## -----
## Resample method            = "bootstrap"
## Number of resamples        = 499
## Number of admissible results = 499
## Approach to handle inadmissibles = "replace"
## Sign change option         = "none"
## Random seed                 = 170818
##
## Construct details:
## -----
## Name  Modeled as   Order   Mode
##
## INV   Common factor First order "modeA"
## SAT   Common factor First order "modeA"
## INT   Common factor First order "modeA"
##
## ----- Estimates -----
##
## Estimated path coefficients:
## =====
##
##                                     CI_standard_z
## Path      Estimate Std. error  t-stat.  p-value  95%
```



```

## INT ~ INV      0.4283      0.0430      9.9694      0.0000 [ 0.3426; 0.5110 ]
## INT ~ SAT     -0.4017      0.0424     -9.4832      0.0000 [-0.4810;-0.3150 ]
##
## Estimated loadings:
## =====
##
##                                     CI_standard_z
## Loading      Estimate Std. error  t-stat.  p-value      95%
## INV =~ INV1   0.6626    0.0512   12.9388  0.0000 [ 0.5630; 0.7637 ]
## INV =~ INV2   0.8173    0.0434   18.8280  0.0000 [ 0.7346; 0.9048 ]
## INV =~ INV3   0.6950    0.0442   15.7194  0.0000 [ 0.6103; 0.7836 ]
## INV =~ INV4   0.4080    0.0636    6.4190  0.0000 [ 0.2906; 0.5397 ]
## SAT =~ SAT1   0.6984    0.0384   18.1724  0.0000 [ 0.6196; 0.7703 ]
## SAT =~ SAT2   0.7170    0.0436   16.4393  0.0000 [ 0.6296; 0.8006 ]
## SAT =~ SAT3   0.9418    0.0355   26.5665  0.0000 [ 0.8799; 1.0189 ]
## INT =~ INT1   0.7033    0.0409   17.1850  0.0000 [ 0.6213; 0.7817 ]
## INT =~ INT2   0.8365    0.0341   24.5367  0.0000 [ 0.7698; 0.9034 ]
##
## Estimated weights:
## =====
##
##                                     CI_standard_z
## Weight      Estimate Std. error  t-stat.  p-value      95%
## INV <~ INV1  0.3337    0.0254   13.1462  0.0000 [ 0.2831; 0.3826 ]
## INV <~ INV2  0.4117    0.0233   17.6962  0.0000 [ 0.3656; 0.4568 ]
## INV <~ INV3  0.3500    0.0211   16.5574  0.0000 [ 0.3083; 0.3912 ]
## INV <~ INV4  0.2055    0.0305    6.7398  0.0000 [ 0.1488; 0.2684 ]
## SAT <~ SAT1  0.3421    0.0166   20.6161  0.0000 [ 0.3075; 0.3726 ]
## SAT <~ SAT2  0.3512    0.0175   20.0746  0.0000 [ 0.3157; 0.3843 ]
## SAT <~ SAT3  0.4613    0.0215   21.4359  0.0000 [ 0.4220; 0.5064 ]
## INT <~ INT1  0.5120    0.0187   27.4357  0.0000 [ 0.4752; 0.5483 ]
## INT <~ INT2  0.6090    0.0230   26.4507  0.0000 [ 0.5646; 0.6548 ]
## -----

```

The data frame output is much more convenient if results are to be reported in e.g., a publication. With a few adjustments, the data frame output can be converted into a near publication-ready table using e.g., the `kable` function of the `knitr` (Xie, 2020) package.

```

require(kableExtra)
kable_styling(kable(sum_csem$Estimates$Path_estimates,
  digits = 4,
  caption = "Path estimates",
  format = "latex",
  booktabs = TRUE), latex_options = c("scale_down", "hold_position"))

```

Table 5.7: Path estimates

Name	Construct_type	Estimate	Std_err	t_stat	p_value	CI_standard_z.95%L	CI_standard_z.95%U
INT ~ INV	Common factor	0.4283	0.0430	9.9694	0	0.3426	0.511
INT ~ SAT	Common factor	-0.4017	0.0424	-9.4832	0	-0.4810	-0.315

### 5.5.5 `verify()`

```
verify(.object)
```

Verify is a small, one-argument function that verifies admissibility of the results obtained during the estimation. In `cSEM` there are five defects that are considered to render the results inadmissible.

**Non-convergence of the weight algorithm** Non-convergence can be an issue for all iterative algorithms such as PLS, GSCA, SUMCORR, GENVAR, and SSQCORR (the other GCCA algorithms MAXVAR and MINVAR do not iterate). For PLS and GSCA the maximum number of iterations is set to 100 by default, but may be changed by the user via `csem()`'s `.iter_max` argument. For SUMCORR, GENVAR, and SSQCORR optimization is done using the Augmented Lagrangian Minimization Algorithm implemented in the `auglag()` function of the `alabama` (Varadhan, 2015) package. In this case, the user has no influence on the convergence behavior.

**Standardized loading estimates larger than 1** A violation implies either negative measurement error variances or construct-indicator correlations larger than 1. Loadings larger than 1 primarily occur after disattenuating the composite-indicator correlations for constructs modeled as common factors to obtain consistent loading estimates.

**Estimated construct correlation is not positive semi-definite** Similar to the issue of loadings larger than 1, this primarily occurs after disattenuating the composite correlations for constructs modeled as common factors to obtain consistent construct-correlation estimates. Correlation estimates may be larger than 1 which renders the construct correlation matrix non-positive semi-definite.

**Congeneric reliability estimates larger than 1** Congeneric reliability of construct  $j$  is estimated as the square of the sum of the estimated loadings

times the corresponding weights of that construct. If loading estimates are larger than 1, this is likely, but not necessary, to cause reliability estimates to be larger than 1 as well. In turn, being a sum, to have all loadings smaller than 1 does not imply reliability estimates smaller than 1.

### **Estimated model-implied correlation matrix is not positive semi-definite**

Can have many causes. It is the most commonly encountered reason for inadmissibility.

If the object to be verified is of class `cSEMResults_2ndorder`, i.e., estimates are based on a model containing second-order constructs, both the first and the second stage are checked separately.

Currently, a model-implied indicator correlation matrix for models containing nonlinear terms is not available. `verify()` therefore skips the check for positive definiteness of the model-implied indicator correlation matrix for nonlinear models and returns “ok”.

### **Example**

For the Switching data and model all 5 issues are ok.

```
verify(out_switching)

## -----
##
## Verify admissibility:
##
##   admissible
##
## Details:
##
##   Code   Status   Description
##   1      ok       Convergence achieved
##   2      ok       All absolute standardized loading estimates <= 1
##   3      ok       Construct VCV is positive semi-definite
##   4      ok       All reliability estimates <= 1
##   5      ok       Model-implied indicator VCV is positive semi-definite
## -----
```

## 5.5.6 testOMF()

```
testOMF(  
  .object           = NULL,  
  .alpha           = 0.05,  
  .fit_measures    = FALSE,  
  .handle_inadmissibles = c("drop", "ignore", "replace"),  
  .R               = 499,  
  .saturated       = FALSE,  
  .seed            = NULL,  
  .verbose         = TRUE  
)
```

The function performs the bootstrap-based exact test for overall model fit originally proposed by [Beran and Srivastava \(1985\)](#). See also [Dijkstra and Henseler \(2015a\)](#) who first suggested the test in the context of PLS-PM.

`testOMF()` tests the null hypothesis that the population indicator correlation matrix equals the population model-implied indicator correlation matrix. In principal, several discrepancy measures may be used as test statistics; hence, `testOMF()` computes four such discrepancy measures, namely the geodesic distance ( $dG$ ), the squared Euclidean distance ( $dL$ ), the standardized root mean square residual (SRMR), and the distance based on the maximum likelihood fit function (dML). The reference distribution for each test statistic is obtained by the bootstrap as proposed by [Beran and Srivastava \(1985\)](#). See Chapter 2 for details on the distance measures (Section 2.4) and on the test in general (Section 2.5).

It is possible to perform the bootstrap-based test using fit measures such as the GFI or the RMSEA as the test statistic if `.fit_measures = TRUE`. This is experimental. To the best of my knowledge, the applicability and usefulness of these fit measures as test statistics in exact model fit assessment have not been formally, statistically assessed yet. The logic of the test theoretically applies to some fit indices as well. Hence, their applicability seems theoretically justified, although it is unclear whether it is of any practical merit.

If `.saturated = TRUE`, the original structural model is ignored and replaced by a saturated model, i.e., a model in which all constructs are allowed to correlate freely.

This is useful to test misspecification of the measurement model in isolation.

The other function arguments are used to set the number of bootstrap replication (`.R`), the  $\alpha$ -level or a vector of  $\alpha$ -levels based upon which to compute the critical

quantile(s) of the bootstrap distribution (`.alpha`), the random number seed (`.seed`), instructions how inadmissible results are to be handled (`handle_inadmissibles`), and whether the function should be verbose in a sense that progress is printed to the console.

The resulting object is of class `cSEMTtestOMF` containing the following list elements:

**Test\_statistic** The value of the the geodesic distance, the squared Euclidean distance, the standardized root mean square residual (SRMR), and the distance based on the maximum likelihood fit function.

**Critical\_value** The corresponding critical values computed as the  $1 - \alpha$  quantile of the bootstrap distribution of the respective test statistics.

**Decision** The test decision for each test statistic. One of: `FALSE` (Reject) or `TRUE` (Do not reject).

**Information** Additional test information such as the number of bootstrap runs, the number of admissible results, the number of total runs, and the seed used.

## Example

Like all postestimation functions, `testOMF()` is designed to do as much as possible in the background. Hence, typically it suffices to call `testOMF()` on the object of interest without setting any additional arguments. For illustration some arguments are used nonetheless:

```
out_testOMF <- testOMF(out_switching,
  .alpha = c(0.01, 0.05, 0.1),
  .seed = 2311,
  .verbose = FALSE) # .verbose only set to FALSE
                    # in order to suppress console output such as
                    # the progress bar
```

```
out_testOMF
## -----
## ----- Test for overall model fit based on Beran & Srivastava (1985) -----
##
## Null hypothesis:
##
```

```

##      +-----+
##      |
##      |  H0: The model-implied indicator covariance matrix equals the  |
##      |  population indicator covariance matrix.                      |
##      |
##      +-----+
##
## Test statistic and critical value:
##
##                                     Critical value
## Distance measure   Test statistic   99%   95%   90%
## dG                 0.0862         0.0285 0.0228 0.0201
## SRMR              0.0483         0.0373 0.0326 0.0297
## dL                0.1049         0.0625 0.0479 0.0398
## dML               0.4605         0.1532 0.1220 0.1061
##
##
## Decision:
##
##                               Significance level
## Distance measure   99%   95%   90%
## dG                 reject reject reject
## SRMR              reject reject reject
## dL                reject reject reject
## dML               reject reject reject
##
## Additional information:
##
## Out of 499 bootstrap replications 467 are admissible.
## See ?verify() for what constitutes an inadmissible result.
##
## The seed used was: 2311
## -----

```

In this case, the null hypothesis must be rejected on all conventional levels, casting doubt on any causal claims that may be derived from the postulated model.

### 5.5.7 testMICOM()

```

testMICOM(
  .object           = NULL,
  .alpha            = 0.05,
  .approach_p_adjust = "none",
  .handle_inadmissibles = c("drop", "ignore", "replace"),
  .R                = 499,

```

```
.seed          = NULL,  
.verbose      = TRUE  
)
```

The function performs the permutation-based test for measurement invariance of composites across groups proposed by [Henseler et al. \(2016b\)](#). According to the authors, measurement invariance in composite models can be assessed by a three-step procedure. The first two steps involve an assessment of configural and compositional invariance ([Henseler et al., 2016b](#), p.413 and p.414f). The third step involves mean and variance comparisons across groups ([Henseler et al., 2016b](#), p.415f). Assessment of configural invariance is qualitative in nature; hence, only step 2 and 3 are tested by the `testMICOM()` function.

As `testMICOM()` requires at least two groups, `.object` must be of class `cSEMResults_multi`. It is possible to compare more than two groups, however, multiple-testing issues arise in this case. To adjust p-values, several p-value adjustments are available via the `approach_p_adjust` argument.

Similar to the `testOMF()` function, there are arguments to set the number of permutation runs to conduct (`.R`), the random number seed (`.seed`), instructions how inadmissible results are to be handled (`handle_inadmissibles`), and whether the function should be verbose in a sense that progress is printed to the console.

The resulting object is a list of class `cSEMTtestMICOM` containing the following list elements:

**Step2** A list containing the results of the test for compositional invariance (Step 2).

**Step3** A list containing the results of the test for mean and variance equality (Step 3).

**Information** Additional test information such as the number of permutation runs, the number of admissible results, the number of total runs, and the seed used.

### Example

For illustration, I use the model setup described in [Henseler et al. \(2016b, p.317\)](#). To generate random data for the model, I use the `cSEM.DGP` package ([Rademaker and](#)

Schamberger, 2020). `cSEM.DGP` is a package I wrote together with Tamara Chamberger. The package provides the `generateData()` function that enables random data generation based on a population structural equation model. For details, see the [the `cSEM.DGP` package website](#).<sup>14</sup>

First, two data generating processes (DGPs) are created; both consist of a single-indicator common factor  $Y$  and a two-indicator composite  $X$ . The common factor is explained by  $X$  with a coefficient of 0.6. The DGPs only differ in how the composite  $X$  is build. In group 1 the first weight is 0.4 and the second is 0.8. In group 2, weights switch position. Hence, the two DGPs (groups) are not compositionally invariant but the mean and the variances of the composite are equal across groups:

```
dgp1 <- "  
# Structural model  
Y ~ 0.6*X  
  
# Reflective measurement model  
Y =~ 1*y1  
# Composite measurement model  
X <~ 0.4*x1 + 0.8*x2  
  
# Indicator correlation  
x1 ~~ 0.3125*x2  
"  
  
dgp2 <- "  
# Structural model  
Y ~ 0.6*X  
  
# Reflective measurement model  
Y =~ 1*y1  
# Composite measurement model  
X <~ 0.8*x1 + 0.4*x2  
  
# Indicator correlation  
x1 ~~ 0.3125*x2  
"
```

Based upon the indicator correlation matrix implied by the DGPs, sample data of size 399 (for group 1) and 200 (for group 2) is drawn from a standard normal distribution using the `generateData()` function of the `cSEM.DGP` package:

---

<sup>14</sup>See: <https://m-e-rademaker.github.io/cSEM.DGP/>; last accessed: 19.04.2020



```
require(cSEM.DGP)
g1 <- generateData(dgp1, .N = 399) # requires cSEM.DGP
g2 <- generateData(dgp2, .N = 200) # requires cSEM.DGP
```

The model to estimate is identical for both groups; however, the sample data stem from different DGPs. Estimation should thus yield different weight estimates for the composite weights of  $X$ . Moreover, the test for measurement invariance should find strong evidence of compositional invariance but no evidence of mean and variance differences across groups:

```
# Specify model
model <- "
# Structural model
Y ~ X

# Measurement model
Y =~ y1
X <- x1 + x2
"

# Estimate
csem_results <- csem(.data = list("group1" = g1, "group2" = g2), model)

# Compare weight estimates
wg1 <- summarize(csem_results)$group1$Estimates$Weight_estimates[, c("Name", "Estimate")]
wg2 <- summarize(csem_results)$group2$Estimates$Weight_estimates[, c("Name", "Estimate")]
data.frame(rbind(wg1, wg2), "Group" = rep(c("group1", "group2"), each = 3))

##      Name Estimate Group
## 1 X <- x1 0.3833547 group1
## 2 X <- x2 0.8310082 group1
## 3 Y <- y1 1.0000000 group1
## 4 X <- x1 0.8855056 group2
## 5 X <- x2 0.2488690 group2
## 6 Y <- y1 1.0000000 group2
```

```
testMICOM(csem_results, .seed = 1987, .verbose=FALSE)

## -----
## ----- Test for measurement invariance based on Henseler et al (2016) -----
## ===== Step 1 - Configural invariance =====
##
## Configural invariance is a precondition for step 2 and 3.
## Do not proceed to interpret results unless
## configural invariance has been established.
```

```

##
## ===== Step 2 - Compositional invariance =====
##
## Null hypothesis:
##
##      +-----+
##      |
##      |  H0: Compositional measurement invariance of the constructs.  |
##      |
##      +-----+
##
## Test statistic and p-value:
##
##   Compared groups: group1_group2
##
##                               p-value by adjustment
## Construct Test statistic      none
## X           0.7863            0.0000
## Y           1.0000            1.0000
##
##
## ===== Step 3 - Equality of the means and variances =====
##
## Null hypothesis:
##
##      +-----+
##      |
##      |  1. H0: Difference between group means is zero                |
##      |  2. H0: Log of the ratio of the group variances is zero      |
##      |
##      +-----+
##
## Test statistic and critical values:
##
##   Compared groups: group1_group2
##
## Mean
##
##                               p-value by adjustment
## Construct Test statistic      none
## X           0.0289            0.7295
## Y           -0.0962           0.2485
##
## Var
##
##                               p-value by adjustment
## Construct Test statistic      none
## X           -0.0876           0.4549
## Y           0.0753            0.5671
##

```

```
##
## Additional information:
##
## Out of 499 permutation runs, 499 where admissible.
## See ?verify() for what constitutes an inadmissible result.
##
## The seed used was: 1987
##
## Number of observations per group:
##
## Group      No. observations
## group1    399
## group2    200
## -----
```

As expected, the weights differ and the null hypothesis of compositional invariance of the composite  $X$  is clearly rejected. Likewise, as expected, all other null hypothesis are not rejected.

### 5.5.8 testMGD()

```
testMGD(
  .object          = NULL,
  .alpha          = 0.05,
  .approach_p_adjust = "none",
  .approach_mgd   = c("all", "Klesel", "Chin", "Sarstedt",
                    "Keil", "Nitzl", "Henseler", "CI_para",
                    "CI_overlap"),
  .parameters_to_compare = NULL,
  .handle_inadmissibles = c("replace", "drop", "ignore"),
  .R_permutation    = 499,
  .R_bootstrap      = 499,
  .saturated        = FALSE,
  .seed             = NULL,
  .type_ci          = "CI_percentile",
  .type_vcv         = c("indicator", "construct"),
  .verbose          = TRUE
)
```

The function performs various group comparison tests, commonly called “tests of multigroup differences” – hence the function name. As of version 0.2.0 of the package, eight different test approaches are available via the `.approach_mgd` argument. By default, all approaches are computed (`.approach_mgd = "all"`). I

briefly describe the tests here; for details, however, readers are referred to the original sources.

**Approach suggested by Klesel et al. (2019); `.approach_mgd = "Klesel"`** The model-implied variance-covariance matrix – either indicator (`.type_vcv = "indicator"`) or construct (`.type_vcv = "construct"`) – is compared across groups. If the model-implied indicator or construct correlation matrix based on a saturated structural model should be compared, set `.saturated = TRUE`. To measure the distance between the model-implied variance-covariance matrices, the geodesic distance ( $dG$ ) and the squared Euclidean distance ( $dL$ ) are used. If more than two groups are compared, the average distance over all groups is used.

**Approach suggested by Sarstedt et al. (2011); `.approach_mgd = "Sarstedt"`** Groups are compared in terms of parameter differences across groups. Sarstedt et al. (2011) propose to test if parameter  $k$  is equal across all groups. If several parameters are tested simultaneously it is recommended to adjust the significance level or the p-values for decision making. By default no multiple testing correction is done, however, several common adjustments are available via `.approach_p_adjust`. Note: the test has severe conceptual shortcomings. It is only implemented for comparison to other tests. Use is discouraged.

**Approach suggested by Chin and Dibbern (2010); `.approach_mgd = "Chin"`** Similar to Sarstedt et al. (2011)'s approach, groups are compared in terms of parameter differences across groups. However, Chin and Dibbern (2010) suggest to test if parameter  $k$  is equal between two groups. If more than two groups are tested for equality, parameter  $k$  is compared between all pairs of groups. In this case, it is also recommended to adjust the significance level or the p-values since this is essentially a multiple testing setup. If several parameters are tested simultaneously, correction is by group and number of parameters. By default no multiple testing correction is done, however, several common adjustments are available via `.approach_p_adjust`.

**Approach suggested by Keil et al. (2000); `.approach_mgd = "Keil"`** The test is similar to those proposed by Chin and Dibbern (2010) and Sarstedt et al.

(2011). Groups are compared in terms of parameter differences across groups. As in [Chin and Dibbern \(2010\)](#), [Keil et al. \(2000\)](#) tests if parameter  $k$  is equal between two groups. It is assumed, that the standard errors of the coefficients are equal across groups. The calculation of the standard error of the parameter difference is adjusted as proposed by [Henseler et al. \(2009\)](#). If more than two groups are tested for equality, parameter  $k$  is compared between all pairs of groups. In this case, it is recommended to adjust the the p-values since this is essentially a multiple testing setup. If several parameters are tested simultaneously, correction is by group and number of parameters. By default no multiple testing correction is done, however, several common adjustments are available via `.approach_p_adjust`.

**Approach suggested by [Nitzl \(2010\)](#); `.approach_mgd = "Nitzl"`** Again, groups are compared in terms of parameter differences across groups. Similarly to [Keil et al. \(2000\)](#), a single parameter  $k$  is tested for equality between two groups. However, in contrast, it is assumed, that the standard errors of the coefficients are unequal across groups. If more than two groups are tested for equality, parameter  $k$  is compared between all pairs of groups. In this case, it is recommended to adjust the p-values since this is essentially a multiple testing setup. If several parameters are tested simultaneously, correction is by group and number of parameters. By default no multiple testing correction is done, however, several common adjustments are available via `.approach_p_adjust`.

**Approach suggested by [Henseler \(2007\)](#); `.approach_mgd = "Henseler"`**

Groups are compared in terms of parameter differences across groups. In doing so, the bootstrap estimates of one parameter are compared across groups. In the literature, this approach is also known as PLS-MGA. This test is an one-sided test, therefore we perform a left-sided and a right-sided test to investigate whether a parameter differs across two groups. Consequently, `.approach_p_adjust` is ignored. Moreover, no overall decision is returned. For a more detailed description, see also [Henseler et al. \(2009\)](#).

**Approach mentioned in [Sarstedt et al. \(2011\)](#); `.approach_mgd = "CI_param"`**

The approach is based on the confidence intervals (CIs) constructed around the parameter estimates for two groups. If the parameter of one group is covered by

the CI of the other group and/or vice versa, it can be concluded that there is no group difference. Since it is based on the CIs `.approach_p_adjust` is ignored.

**Approach mentioned in Sarstedt et al. (2011); `.approach_mgd = "CI_overlap"`**

The approach is also based on the confidence intervals (CIs) constructed around the parameter estimates of the two groups. In this case, if the two CIs overlap, it can be concluded that there is no group difference. Since it is based on the confidence intervals `.approach_p_adjust` is ignored.

By default, approaches based on parameter differences across groups compare all parameters (`.parameters_to_compare = NULL`). To compare only a subset of parameters, the user can provide the parameters in lavaan model syntax similar to how the main model to estimate would be specified. Assume, for example, the following postulated model:

```
model_to_estimate <- "  
Structural model  
eta2 ~ eta1  
eta3 ~ eta1 + eta2  
  
# Reflective Measurement  
eta1 =~ y11 + y12 + y13  
eta2 =~ y21 + y22 + y23  
eta3 =~ y31 + y32 + y33  
"
```

If only the path from `eta1` to `eta3` and the loadings of `eta1` are to be compared across groups, write:

```
to_compare <- "  
Structural parameters to compare  
eta3 ~ eta1  
  
# Loadings to compare  
eta1 =~ y11 + y12 + y13  
"
```

and then use `.parameters_to_compare = to_compare`. Note that the “model” provided to `.parameters_to_compare` does not need to be an estimable model! In contrast to all other functions in cSEM that have the `.handle_inadmissibles` argument, the argument’s default is *“replace”* instead of *“ignore”*. The reason is

that the approach suggested by [Sarstedt et al. \(2011\)](#) requires an equal number of resamples.

Argument `.R_permutation` is ignored for the *"Nitzl"* and the *"Keil"* approach as these tests do not require permutation. `.R_bootstrap` is ignored if `.object` already contains resamples, i.e., has class `cSEMResults_resampled` or if only the *"Klesel"* or the *"Chin"* approach are used.

The argument `.saturated` is used by *"Klesel"* only. If `.saturated = TRUE` the original structural model is ignored and replaced by a saturated model, i.e., a model in which all constructs are allowed to correlate freely. This is useful to test differences in the measurement models between groups in isolation.

The resulting object is a list of class `cSEMTestMGD` containing a list element for each test that was chosen via `.approach_mgd` as well as an element `Information` with additional information such as the seed, number of permutation runs etc.

## Example

For illustration, I randomly split the `Switching` data set into two groups. In this case, groups should, sampling error aside, not differ systematically:

```
set.seed(1224)
s <- sample(1:nrow(Switching))

switch1 <- Switching[s[1:383], ]
switch2 <- Switching[s[384:767], ]

model <- "
# Measurement models
INV =~ INV1 + INV2 + INV3 + INV4
SAT =~ SAT1 + SAT2 + SAT3
INT =~ INT1 + INT2

# Structural model
INT ~ INV + SAT
"

## Estimation
out_switching <- csem(
  .data = list(switch1, switch2),
  .model = model,
  .resample_method = "bootstrap",
  .handle_inadmissibles = "replace",
```

```
.seed = 170818
)
```

```
testMGD(out_switching, .verbose = FALSE)
```

```
## -----
## ----- Overview -----
##
## Total permutation runs = 623
## Admissible permutation results = 499
## Permutation seed = -1572774555
##
## Total bootstrap runs = 499
## Admissible bootstrap results:
##
## Group      Admissibles
## Data_1      499
## Data_2      499
##
## Bootstrap seed:
##
## Group      Seed
## Data_1      170818
## Data_2      170818
##
## Number of observations per group:
##
## Group      No. Obs.
## Data_1      383
## Data_2      384
##
## Overall decision (based on alpha = 5%):
##
##           p_adjust = 'none'
## Sarstedt      reject
## Chin          Do not reject
## Keil          Do not reject
## Nitzl         Do not reject
##
## For details on a particular approach type:
##
## - `print(<object-name>, .approach_mgd = '<approach-name>')`
##
## -----
```



As stated in the output, to get details on e.g., the test proposed by [Klesel et al. \(2019\)](#), it must be chosen explicitly:

```
testMGD(out_switching, .approach_mgd = "Klesel", .verbose = FALSE)

## -----
## ----- Overview -----
##
## Total permutation runs          = 655
## Admissible permutation results  = 499
## Permutation seed                = -1554328016
##
## Total bootstrap runs            = NA
## Admissible bootstrap results:
##
## Group          Admissibles
## Data_1         NA
##
## Bootstrap seed:
##
## Group          Seed
## Data_1         NA
##
## Number of observations per group:
##
## Group          No. Obs.
## Data_1         383
## Data_2         384
## -----
## ----- Test for multigroup differences based on Klesel et al. (2019) -----
##
## Null hypothesis:
##
## +-----+
## |                                             |
## | H0: Model-implied indicator covariance matrix is equal across groups. |
## |                                             |
## +-----+
##
## Test statistic and p-value:
##
## Distance measure  Test statistic    p-value    Decision
## dG                0.0158           0.9399    Do not reject
## dL                0.0521           0.9739    Do not reject
## -----
```

As expected, the null hypothesis is not rejected on all conventional  $\alpha$ -levels.

## 5.5.9 testHausman()

```
testHausman(  
  .object           = NULL,  
  .eval_plan       = c("sequential", "multiprocess"),  
  .handle_inadmissibles = c("drop", "ignore", "replace"),  
  .R               = 499,  
  .resample_method  = c("bootstrap", "jackknife"),  
  .seed            = NULL  
)
```

The function calculates the regression-based Hausman test for endogeneity. Practically, the test is used to compare OLS to 2SLS estimates (e.g., [Wooldridge, 2010](#), p.131). The function is currently in an experimental stage, poorly documented and may undergo changes in the future. The principal intend is to allow users to supply a *cSEMResults* object whose path model has been estimated by two-stage least squares (2SLS; using `.approach_path = "2SLS"` in `csem()`). The instruments required for the computation are extracted from the object by `testHausman()` to perform the regression based test for endogeneity using bootstrap or jackknife standard errors.

### Example

To illustrate the test, I use an example from [Dijkstra and Henseler \(2015a\)](#). The values can be found on pages 15 and 16 of the paper:

```
# Preparation  
Lambda <- t(kronecker(diag(6), c(0.7, 0.7, 0.7)))  
Phi <- matrix(c(1.0000, 0.5000, 0.5000, 0.5000, 0.0500, 0.4000,  
              0.5000, 1.0000, 0.5000, 0.5000, 0.5071, 0.6286,  
              0.5000, 0.5000, 1.0000, 0.5000, 0.2929, 0.7714,  
              0.5000, 0.5000, 0.5000, 1.0000, 0.2571, 0.6286,  
              0.0500, 0.5071, 0.2929, 0.2571, 1.0000, sqrt(0.5),  
              0.4000, 0.6286, 0.7714, 0.6286, sqrt(0.5), 1.0000),  
            ncol = 6)  
  
## Create population indicator covariance matrix  
Sigma <- t(Lambda) %*% Phi %*% Lambda  
diag(Sigma) <- 1  
dimnames(Sigma) <- list(paste0("x", rep(1:6, each = 3), 1:3),  
                       paste0("x", rep(1:6, each = 3), 1:3))
```

```

## Generate data
dat <- MASS::mvrnorm(n = 500, mu = rep(0, 18), Sigma = Sigma, empirical = TRUE)
# empirical = TRUE to show that 2SLS is in fact able to recover the true population
# parameters.

```

The model to be estimated is nonrecursive; there are feedback loops and, therefore, endogeneity issues.

```

model <- "
## Structural model (nonrecursive)
eta5 ~ eta6 + eta1 + eta2
eta6 ~ eta5 + eta3 + eta4

## Measurement model
eta1 =~ x11 + x12 + x13
eta2 =~ x21 + x22 + x23
eta3 =~ x31 + x32 + x33
eta4 =~ x41 + x42 + x43

eta5 =~ x51 + x52 + x53
eta6 =~ x61 + x62 + x63
"

## Estimation
res_2sls <- csem(dat, .model = model, .approach_paths = "2SLS",
                 .instruments = list("eta5" = c('eta1','eta2','eta3','eta4'),
                                     "eta6" = c('eta1','eta2','eta3','eta4')))

## Test for endogeneity
test_ha <- testHausman(res_2sls)
test_ha

## -----
## ----- Regression-based Hausman test -----
##
## Null hypothesis:
##
## +-----+
## |                                             |
## |  H0: Variable(s) suspected to be endogenous are uncorrelated with the |
## |  error term (no endogeneity).          |
## |                                             |
## +-----+
##
## Regression output:
##
##
## Dependent construct: 'eta5'

```

```

##
## Independent construct      Estimate  Std. error  t-stat.  p-value
## eta1                      -0.3000    0.1082    -2.7731  0.0056
## eta2                       0.4999    0.0881     5.6755  0.0000
## eta6                       0.2501    0.1251     1.9995  0.0456
## Resid_eta6                0.9784    0.2901     3.3728  0.0007
##
## Dependent construct: 'eta6'
##
## Independent construct      Estimate  Std. error  t-stat.  p-value
## eta3                       0.4999    0.0559     8.9412  0.0000
## eta4                       0.2501    0.0518     4.8243  0.0000
## eta5                       0.5001    0.1052     4.7536  0.0000
## Resid_eta5                -0.0056    0.1435    -0.0389  0.9690
## -----

```

The residuals term `Resid_eta6` obtained from a regression of  $\eta_6$  on all instruments is statistically significant in a regression of `eta5` on `eta1`, `eta2`, `eta6`, and `Resid_eta6`. This is statistical indication that  $\eta_6$  is endogenous. `Resid_eta5` on the other hand is statistically insignificant. Hence, the test does not indicate any statistical evidence of endogeneity which is likely wrong, in this case.

## 5.6 Technical details

The package is developed in R using the RStudio IDE ([RStudio Team, 2015](#)). Changes to the package are tracked using the powerful version control system Git (e.g., [Chacon and Straub, 2014](#)). Git is primarily a command-line tool, although some of the typical actions one would perform in Git are integrated into the RStudio IDE. However, to fully access Gits capabilities, use of the terminal is indispensable. In short, given a project folder, called a repository in Git parlance, Git essentially distinguishes between unmodified files, files staged for commit – i.e., files that have been modified, are ready to be committed, but have not yet been “officially recorded” by Git –, and committed files. A commit in Git is like a snapshot or a record of the current state of the repository. These snapshots are stored in a highly space-efficient manner and can be, if necessary, accessed or modified later in the project development cycle. In fact, unless the repository is deleted completely, commits are essentially impossible to destroy by the user. Hence, no changes are lost and potential bugs introduced by e.g., a modification to existing code can quickly and safely be reverted

at any later stage in the project development cycle. Moreover, given that each commit was labeled in a meaningful way, commits provide a natural change history. In addition, Git offers a flexible branching system that corresponds well to the demands of a collaborative software project like an R package. After setting up Git, every change to a repository happens by default on the so-called “master branch”. Branches are best understood as separate instances of a repository. If a branch is created it is always created based on a parent branch – typically the master branch. At the point of creation, the new branch is an exact copy of its parent that lives independently of the repository it has been copied from. Branches are primarily used to implement new features or to test a modification to existing code while leaving the parent branch unaffected. Branches can be modified and extended and, if necessary, branched again. Once work on the copy-branch is finished – because a feature has been implemented and sufficient stability tests have been made –, it can be merged back into the parent branch. If, in the meantime, the parent had been modified, Git will detect all changes and inform of any conflicts introduced by changes of the same line of code in both the parent and the copy-branch.

Since Git tracks changes on the local machine, a remote copy of the package repository is hosted and maintained in a [GitHub](#) repository.<sup>15</sup> GitHub is a platform for collaboration on code projects. Everyone contributing to the package may obtain a local clone of the repository, implement modifications and push – if granted permission by the repository owner – the changes back to the remote where they are now accessible to all other users connected to the project. Git and GitHub offer a variety of tools to synchronize the remote and local copies of a project allowing me, the maintainer, to track, check, comment, and organize changes to different files by different people at different times.

One of the more tedious tasks in package development is testing the code. To make the process as hassle-free as possible, I use two helpful resources. First the actual tests are embedded in the [testthat](#) (Wickham, 2011) framework. `testthat` is an R package which offers a number of functions to organize and structure tests of different parts of code. Second, instead of running each test manually whenever the package is modified, I make use of the continuous integration (CI) service [Travis CI](#)<sup>16</sup> to perform

---

<sup>15</sup>See: <https://github.com/M-E-Rademacher/cSEM>; last accessed: 19.04.2020

<sup>16</sup>See: <https://travis-ci.org/>; last accessed: 19.04.2020

this task automatically. Once set up correctly, Travis CI automatically tracks changes to `cSEM`'s remote package repository on GitHub. Whenever a change to the package is detected – typically, because a change was pushed from the local repository to the remote repository – Travis CI executes the following steps:

1. Create a brand new virtual machine on one of Travis CI's servers.
2. Install an operating system on the machine.<sup>17</sup>
3. Install R and download the required packages as indicated in the `DESCRIPTION` file of the package.
4. Clone the package repository and install the package using `install.packages("cSEM")`.
5. Run R `CMD check`, a standardized list of checks run by CRAN.
6. Run `testthat::test_check("cSEM")` which runs all user written tests organized within the `testthat` framework.

If any of the installations, checks, and tests exit with an error or a warning, Travis-CI indicates where the error or warning occurred. These can then be fixed locally; upon pushing the changes back to the remote, Travis CI picks up a change and reexecutes steps 1-6. This process is repeated until no issues are encountered.

The package website <https://m-e-rademaker.github.io/cSEM/> is build using the `pkgdown` package (Wickham and Hesselberth, 2020). `pkgdown` creates a static website almost automatically from package meta data files like `DESCRIPTION` or `README` and other files such as the `.Rd` files.

Currently, the package website is hosted on [GitHub pages](#).<sup>18</sup> The actual content of the website, which is required by GitHub pages to build the website, is stored in a branch called "gh-pages" in the [GitHub repository](#). Instead of updating the content of the website manually, I use Travis CI for automatic deployment. Whenever a change

---

<sup>17</sup>To make sure the package passes the checks and tests on every common operating system, I set up Travis CI to actually create three virtual machines. Each machine then uses a different operating system. Currently, I use Ubuntu xenial 16.04 (and 18.04), mac OS 10.13, and windows x86\_64-w64-mingw32/x64. For Windows, I actually use a different CI service called [appveyor](https://www.appveyor.com/) (<https://www.appveyor.com/>). However, the steps described here for Travis CI are virtually identical to those from appveyor.

<sup>18</sup><https://pages.github.com/>; last accessed: 13.03.2020

to the remote repository is made, Travis CI will go through the steps described above. When on the master branch, once testing is finished, an additional deployment script is ran that calls the central `pkgdown` command `pkgdown::build_site()`. The command triggers building the website. If building the site was successful, the script pushes the results into the `gh-pages` branch, thus updating all files required for building the website. GitHub pages then rebuilds the website based on the new files and a couple of seconds later the changes will be visible on the website.

---

# Bibliography

---

- Agarwal, R. and Karahanna, E. (2000), “Time flies when you’re having fun: Cognitive absorption and beliefs about information technology usage”, *MIS Quarterly*, Vol. 24(4), pp. 665–694.
- Aguirre-Urreta, M. I., Rönkkö, M., and Marakas, G. M. (2016), “Omission of causal indicators: Consequences and implications for measurement”, *Measurement: Interdisciplinary Research and Perspectives*, Vol. 14(3), pp. 75–97.
- Ainin, S., Parveen, F., Moghavvemi, S., Jaafar, N. I., and Shuib, N. L. M. (2015), “Factors influencing the use of social media by SMEs and its performance outcomes”, *Industrial Management & Data Systems*, Vol. 115(3), pp. 570–588.
- Barrett, P. (2007), “Structural equation modelling: Adjudging model fit”, *Personality and Individual Differences*, Vol. 42(5), pp. 815–824.
- Bartlett, M. S. (1937), “The statistical conception of mental factors”, *British Journal of Psychology. General Section*, Vol. 28(1), pp. 97–104.
- Becker, J.-M., Klein, K., and Wetzels, M. (2012), “Hierarchical latent variable models in PLS-SEM: Guidelines for using reflective-formative type models”, *Long Range Planning*, Vol. 45(5), pp. 359–394.
- Bengtsson, H. (2018), *future: Unified Parallel and Distributed Processing in R for Everyone*, r package version 1.10.0.  
**URL:** <https://CRAN.R-project.org/package=future>



- Benitez, J., Castillo, A., Llorens, J., and Braojos, J. (2018a), “IT-enabled knowledge ambidexterity and innovation performance in small U.S. firms: The moderator role of social media capability”, *Information & Management*, Vol. 55(1), pp. 131–143.
- Benitez, J., Henseler, J., Castillo, A., and Schuberth, F. (2020), “How to perform and report an impactful analysis using partial least squares: Guidelines for confirmatory and explanatory IS research”, *Information & Management*, Vol. 57(2), pp. 1–16.
- Benitez, J., Llorens, J., and Braojos, J. (2018b), “How information technology influences opportunity exploration and exploitation firm’s capabilities”, *Information & Management*, Vol. 55(4), pp. 508–523.
- Benitez, J., Ray, G., and Henseler, J. (2018c), “Impact of information technology infrastructure flexibility on mergers and acquisitions”, *MIS Quarterly*, Vol. 42(1), pp. 25–43.
- Bentler, P. M. (1990), “Comparative fit indexes in structural models”, *Psychological Bulletin*, Vol. 107(2), pp. 238–246.
- Bentler, P. M. (1995), *EQS Structural Equations Program Manual*, Vol. 6, Multivariate Software, Inc., Encino, CA.
- Bentler, P. M. (2006), *EQS 6 Structural Equations Program Manual*, Multivariate Software, Inc., Encino, CA.
- Bentler, P. M. and Bonett, D. G. (1980), “Significance tests and goodness of fit in the analysis of covariance structures”, *Psychological Bulletin*, Vol. 88(3), pp. 588–606.
- Beran, R. and Srivastava, M. S. (1985), “Bootstrap tests and confidence regions for functions of a covariance matrix”, *The Annals of Statistics*, Vol. 13(1), pp. 95–115.
- Bollen, K. (1993), *Testing Structural Equation Models*, Sage Publications, Newbury Park.
- Bollen, K. A. (1986), “Sample size and Bentler and Bonett’s nonnormed fit index”, *Psychometrika*, Vol. 51(3), pp. 375–377.
- Bollen, K. A. (1987), “Total, direct, and indirect effects in structural equation models”, *Sociological Methodology*, Vol. 17, pp. 37–69.

- Bollen, K. A. (1989a), “A new incremental fit index for general structural equation models”, *Sociological Methods & Research*, Vol. 17(3), pp. 303–316.
- Bollen, K. A. (1989b), *Structural Equations with Latent Variables*, John Wiley & Sons Inc, New York.
- Bollen, K. A. and Bauldry, S. (2011), “Three Cs in measurement models: Causal indicators, composite indicators, and covariates.”, *Psychological Methods*, Vol. 16(3), pp. 265–284.
- Bollen, K. A. and Diamantopoulos, A. (2017), “In defense of causal-formative indicators: A minority report.”, *Psychological Methods*.
- Bollen, K. A. and Stine, R. A. (1992), “Bootstrapping goodness-of-fit measures in structural equation models”, *Sociological Methods & Research*, Vol. 21(2), pp. 205–229.
- Box, G. E. P. (1976), “Science and statistics”, *Journal of the American Statistical Association*, Vol. 71(356), pp. 791–799.
- Braojos, J., Benitez, J. e., Llorens, J., and Ruiz, L. (forthcoming), “Impact of IT integration on the firm’s knowledge absorption and desorption”, *Information & Management*, p. 103290.
- Brock, J. K.-U. and Zhou, Y. J. (2005), “Organizational use of the Internet: Scale development and validation”, *Internet Research*, Vol. 15(1), pp. 67–87.
- Brown, T. A. (2015), *Confirmatory Factor Analysis for Applied Research*, Guilford Publications, New York, 2nd ed.
- Browne, M. W. (1984), “Asymptotically distribution-free methods for the analysis of covariance structures”, *British Journal of Mathematical and Statistical Psychology*, Vol. 37(1), pp. 62–83.
- Browne, M. W. and Cudeck, R. (1992), “Alternative ways of assessing model fit”, *Sociological Methods & Research*, Vol. 21(2), pp. 230–258.
- Carrión, G. C., Henseler, J., Ringle, C. M., and Roldán, J. L. (2016), “Prediction-oriented modeling in business research by means of PLS path

- modeling: Introduction to a JBR special section”, *Journal of Business Research*, Vol. 69(10), pp. 4545–4551.
- Chacon, S. and Straub, B. (2014), *Pro Git*, Apress, 2 ed.
- Charles, E. P. (2005), “The correction for attenuation due to measurement error: Clarifying concepts and creating confidence sets”, *Psychological Methods*, Vol. 10(2), pp. 206–226.
- Cheah, J.-H., Sarstedt, M., Ringle, C. M., Ramayah, T., and Ting, H. (2018), “Convergent validity assessment of formatively measured constructs in PLS-SEM”, *International Journal of Contemporary Hospitality Management*, Vol. 30(11), pp. 3192–3210.
- Chiang, H.-S. and Hsiao, K.-L. (2015), “YouTube stickiness: The needs, personal, and environmental perspective”, *Internet Research*, Vol. 25(1), pp. 85–106.
- Chin, W. W. and Dibbern, J. (2010), “An introduction to a permutation based procedure for multi-group PLS analysis: Results of tests of differences on simulated data and a cross cultural analysis of the sourcing of information system services between germany and the USA”, in “Handbook of Partial Least Squares”, Springer Berlin Heidelberg, pp. 171–193.
- Chin, W. W., Marcolin, B. L., and Newsted, P. R. (2003), “A partial least squares latent variable modeling approach for measuring interaction effects: Results from a monte carlo simulation study and an electronic-mail emotion/adoption study”, *Information Systems Research*, Vol. 14(2), pp. 189–217.
- Chin, W. W. and Todd, P. A. (1995), “On the use, usefulness, and ease of use of structural equation modeling in MIS research: A note of caution”, *MIS Quarterly*, Vol. 19(2), pp. 237–246.
- Cho, E. (2016), “Making reliability reliable”, *Organizational Research Methods*, Vol. 19(4), pp. 651–682.
- Cohen, J. (1988), *Statistical Power Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, Hillsdale.
- Cohen, J. (1992), “A power primer”, *Psychological Bulletin*, Vol. 112(1), pp. 155–159.

- Cohen, J., Cohen, P., West, S. G., and Aiken, L. S. (2003), *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, Mahwah.
- Cronbach, L. J. (1951), “Coefficient alpha and the internal structure of tests”, *Psychometrika*, Vol. 16(3), pp. 297–334.
- Croon, M. (2002), “Using predicted latent scores in general latent structure models”, in Marcoulides, G. and Moustaki, I., (Eds.), “Latent Variable and Latent Structure Models”, Lawrence Erlbaum, pp. 195–224.
- Cudeck, R. and Henly, S. J. (2003), “A realistic perspective on pattern representation in growth data: Comment on bauer and curran (2003)”, *Psychological Methods*, Vol. 8(3), pp. 378–383.
- Davison, A. C. (1997), *Bootstrap Methods and Their Application*, Cambridge University Press.
- Devlieger, I. and Rosseel, Y. (2017), “Factor score path analysis”, *Methodology*, Vol. 13(Supplement 1), pp. 31–38.
- Diamantopoulos, A. (2008), “Formative indicators: Introduction to the special issue”, *Journal of Business Research*, Vol. 61(12), pp. 1201–1202.
- Dijkstra, T. K. (1981), *Latent Variables in Linear Stochastic Models*, Ph.D. thesis, Rijksuniversiteit te Groningen.
- Dijkstra, T. K. (2010), “Latent variables and indices: Herman Wold’s basic design and partial least squares”, in “Handbook of Partial Least Squares”, Springer Berlin Heidelberg, pp. 23–46.
- Dijkstra, T. K. (2013a), “A note on how to make partial least squares consistent”, Unpublished.
- Dijkstra, T. K. (2013b), “The simplest possible factor model estimator”, *Working Paper*.
- Dijkstra, T. K. (2016), “High-dimensional concepts and substance driven ways to make them operational, some tools”, in “Cluj Economics and Business Seminar”, Cluj-Napoca, Babes- Bolyia University, pp. 1–38.

- Dijkstra, T. K. (2017), “A perfect match between a model and a mode”, *Working Paper*.
- Dijkstra, T. K. (2018), “A suggested quasi empirical bayes approach for handling ‘Heywood’-cases, very preliminary”, *Working Paper*.
- Dijkstra, T. K. and Henseler, J. (2011), “Linear indices in nonlinear structural equation models: Best fitting proper indices and other composites”, *Quality & Quantity*, Vol. 45(6), pp. 1505–1518.
- Dijkstra, T. K. and Henseler, J. (2015a), “Consistent and asymptotically normal PLS estimators for linear structural equations”, *Computational Statistics & Data Analysis*, Vol. 81, pp. 10–23.
- Dijkstra, T. K. and Henseler, J. (2015b), “Consistent partial least squares path modeling”, *MIS Quarterly*, Vol. 39(2), pp. 29–316.
- Dijkstra, T. K. and Schermelleh-Engel, K. (2014), “Consistent partial least squares for nonlinear structural equation models”, *Psychometrika*, Vol. 79(4), pp. 585–604.
- Duarte, P. and Amaro, S. (2018), “Methods for modelling reflective-formative second order constructs in PLS”, *Journal of Hospitality and Tourism Technology*.
- Edwards, J. R. (2001), “Multidimensional constructs in organizational behavior research: An integrative analytical framework”, *Organizational Research Methods*, Vol. 4(2), pp. 144–192.
- Edwards, J. R. (2011), “The fallacy of formative measurement”, *Organizational Research Methods*, Vol. 14(2), pp. 370–388.
- Efron, B. and Hastie, T. (2016), *Computer Age Statistical Inference*, Cambridge University Press.
- Fornell, C. and Bookstein, F. L. (1982), “Two structural equation models: LISREL and PLS applied to consumer exit-voice theory”, *Journal of Marketing Research*, Vol. 19(4), pp. 440–452.
- Fornell, C. and Larcker, D. F. (1981), “Evaluating structural equation models with unobservable variables and measurement error”, *Journal of Marketing Research*, Vol. 18(1), pp. 39–50.

- Gerbing, D. W. and Anderson, J. C. (1984), “On the meaning of within-factor correlated measurement errors”, *Journal of Consumer Research*, Vol. 11(1), pp. 572–580.
- Grace, J. B. and Bollen, K. A. (2008), “Representing general theoretical concepts in structural equation models: The role of composite variables”, *Environmental and Ecological Statistics*, Vol. 15(2), pp. 191–213.
- Green, K. W. and Inman, R. A. (2007), “The impact of JIT-II-selling on organizational performance”, *Industrial Management & Data Systems*, Vol. 107(7), pp. 1018–1035.
- Greiff, S. and Heene, M. (2017), “Why psychological assessment needs to start worrying about model fit”, *European Journal of Psychological Assessment*, Vol. 33(5), pp. 313–317.
- Gu, H., Wen, Z., and Fan, X. (2017), “Examining and controlling for wording effect in a self-report measure: A Monte Carlo simulation study”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 24(4), pp. 545–555.
- Hair, J., Hollingsworth, C., Randolph, A., and Chong, A. (2017a), “An updated and expanded assessment of PLS-SEM in information systems research”, *Industrial Management & Data Systems*, Vol. 117(3), pp. 442–458.
- Hair, J. F., Howard, M. C., and Nitzl, C. (2020), “Assessing measurement model quality in PLS-SEM using confirmatory composite analysis”, *Journal of Business Research*, Vol. 109, pp. 101–110.
- Hair, J. F., Hult, G. T. M., Ringle, C. M., and Sarstedt, M. (2017b), *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*, Sage Publications Ltd.
- Hair, J. F., Risher, J. J., Sarstedt, M., and Ringle, C. M. (2019a), “When to use and how to report the results of PLS-SEM”, *European Business Review*, Vol. 31(1), pp. 2–24.
- Hair, J. F. and Sarstedt, M. (2019), “Factors versus composites: Guidelines for choosing the right structural equation modeling method”, *Project Management Journal*, Vol. 50(6), pp. 619–624.

- Hair, J. F., Sarstedt, M., Pieper, T. M., and Ringle, C. M. (2012a), “The use of partial least squares structural equation modeling in strategic management research: A review of past practices and recommendations for future applications”, *Long Range Planning*, Vol. 45(5-6), pp. 320–340.
- Hair, J. F., Sarstedt, M., and Ringle, C. M. (2019b), “Rethinking some of the rethinking of partial least squares”, *European Journal of Marketing*, Vol. 53(4), pp. 566–584.
- Hair, J. F., Sarstedt, M., Ringle, C. M., and Mena, J. A. (2012b), “An assessment of the use of partial least squares structural equation modeling in marketing research”, *Journal of the Academy of Marketing Science*, Vol. 40(3), pp. 414–433.
- Hajiheydari, N. and Ashkani, M. (2018), “Mobile application user behavior in the developing countries: A survey in Iran”, *Information Systems*, Vol. 77, pp. 22–33.
- Hayduk, L. A. (2014), “Shame for disrespecting evidence: The personal consequences of insufficient respect for structural equation model testing”, *BMC Medical Research Methodology*, Vol. 14(1).
- Henseler, J. (2007), “A new and simple approach to multi-group analysis in partial least squares path modeling”, in Martens, H. and Næs, T., (Eds.), “Proceedings of PLS’07 - The 5th International Symposium on PLS and Related Methods”, PLS, Norway: Matforsk, As, pp. 104–107.
- Henseler, J. (2017), “Bridging design and behavioral research with variance-based structural equation modeling”, *Journal of Advertising*, Vol. 46(1), pp. 178–192.
- Henseler, J. (2018), “Partial least squares path modeling: Quo vadis?”, *Quality & Quantity*, Vol. 52(1), pp. 1–8.
- Henseler, J. (2019), *ADANCO 2.0.1*, Composite Modeling GmbH & Co., Kleve, Germany.
- Henseler, J. (forthcoming), *Composite-Based Structural Equation Modeling: An Introduction to Partial Least Squares & Co. Using ADANCO*, Guilford Press.
- Henseler, J. and Chin, W. W. (2010), “A comparison of approaches for the analysis of interaction effects between latent variables using partial least squares path

- modeling”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 17(1), pp. 82–109.
- Henseler, J., Dijkstra, T. K., Sarstedt, M., Ringle, C. M., Diamantopoulos, A., Straub, D. W., Ketchen, D. J., Hair, J. F., Hult, G. T. M., and Calantone, R. J. (2014), “Common beliefs and reality about PLS: Comments on Rönkkö and Evermann (2013)”, *Organizational Research Methods*, Vol. 17(2), pp. 182–209.
- Henseler, J., Fassott, G., Dijkstra, T. K., and Wilson, B. (2012), “Analysing quadratic effects of formative constructs by means of variance-based structural equation modelling”, *European Journal of Information Systems*, Vol. 21(1), pp. 99–112.
- Henseler, J., Hubona, G., and Ray, P. A. (2016a), “Using PLS path modeling in new technology research: Updated guidelines”, *Industrial Management & Data Systems*, Vol. 116(1), pp. 2–20.
- Henseler, J., Ringle, C. M., and Sarstedt, M. (2015), “A new criterion for assessing discriminant validity in variance-based structural equation modeling”, *Journal of the Academy of Marketing Science*, Vol. 43(1), pp. 115–135.
- Henseler, J., Ringle, C. M., and Sarstedt, M. (2016b), “Testing measurement invariance of composites using partial least squares”, *International Marketing Review*, Vol. 33(3), pp. 405–431.
- Henseler, J., Ringle, C. M., and Sinkovics, R. R. (2009), “The use of partial least squares path modeling in international marketing”, *Advances in International Marketing*, Vol. 20, pp. 277–320.
- Henseler, J., Wilson, B., Götz, O., and Hautvast, C. (2007), “Investigating the moderating role of fit on sports sponsorship and brand equity”, *International Journal of Sports Marketing and Sponsorship*, Vol. 8(4), pp. 34–42.
- Hermida, R. (2015), “The problem of allowing correlated errors in structural equation modeling: Concerns and considerations”, *Computational Methods in Social Sciences*, Vol. 3(1), pp. 5–17.
- Hesterberg, T. C. (2015), “What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum”, *The American Statistician*, Vol. 69(4), pp. 371–386.



- Hew, J.-J., Lee, V.-H., Ooi, K.-B., and Wei, J. (2015), “What catalyses mobile apps usage intention: An empirical analysis”, *Industrial Management & Data Systems*, Vol. 115(7), pp. 1269–1291.
- Hoyle, R. (2012), *Handbook of Structural Equation Modeling*, Guilford Press, New York.
- Hsieh, C.-T., Lai, F., and Shi, W. (2006), “Information orientation and its impacts on information asymmetry and E-business adoption”, *Industrial Management & Data Systems*, Vol. 106(6), pp. 825–840.
- Hsu, L.-C. (2017), “Investigating community members’ purchase intention on Facebook fan page”, *Industrial Management & Data Systems*, Vol. 117(5), pp. 766–800.
- Hu, L. and Bentler, P. M. (1999), “Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 6(1), pp. 1–55.
- Hwang, H., Kim, S., Lee, S., and Park, T. (2017a), *gesca: Generalized Structured Component Analysis (GSCA)*, R package version 1.0.4.  
**URL:** <https://CRAN.R-project.org/package=gesca>
- Hwang, H., Malhotra, N. K., Kim, Y., Tomiuk, M. A., and Hong, S. (2010), “A comparative study on parameter recovery of three approaches to structural equation modeling”, *Journal of Marketing Research*, Vol. 47(4), pp. 699–712.
- Hwang, H. and Takane, Y. (2004), “Generalized structured component analysis”, *Psychometrika*, Vol. 69(1), pp. 81–99.
- Hwang, H. and Takane, Y. (2014), *Generalized Structured Component Analysis*, CRC Press.
- Hwang, H., Takane, Y., and Jung, K. (2017b), “Generalized structured component analysis with uniqueness terms for accommodating measurement error”, *Frontiers in Psychology*, Vol. 8.
- Johnson, R. E., Rosen, C. C., Chang, C.-H. D., Djurdjevic, E., and Taing, M. U. (2012), “Recommendations for improving the construct clarity of higher-order

- multidimensional constructs”, *Human Resource Management Review*, Vol. 22(2), pp. 62–72.
- Jöreskog, K. G. (1967), “Some contributions to maximum likelihood factor analysis”, *Psychometrika*, Vol. 32(4), pp. 443–482.
- Jöreskog, K. G. (1969), “A general approach to confirmatory maximum likelihood factor analysis”, *Psychometrika*, Vol. 34(2), pp. 183–202.
- Jöreskog, K. G. (1970a), “A general method for analysis of covariance structures”, *Biometrika*, Vol. 57(2), pp. 239–251.
- Jöreskog, K. G. (1970b), “A general method for estimating a linear structural equation system”, *ETS Research Bulletin Series*, Vol. 1970(2), pp. i–41.
- Jöreskog, K. G. (1971), “Statistical analysis of sets of congeneric tests”, *Psychometrika*, Vol. 36(2), pp. 109–133.
- Jöreskog, K. G. (1978), “Structural analysis of covariance and correlation matrices”, *Psychometrika*, Vol. 43(4), pp. 443–477.
- Jöreskog, K. G. and Sörbom, D. (1982), “Recent developments in structural equation modeling”, *Journal of Marketing Research*, Vol. 19(4), pp. 404–416.
- Jöreskog, K. G. and Sörbom, D. (1993), *LISREL 8 : Structural Equation Modeling with the SIMPLIS Command Language*, Scientific Software International Lawrence Erlbaum, Chicago, Ill. Hillsdale, N.J.
- Keil, M., Tan, B. C., Wei, K.-K., Saarinen, T., Tuunainen, V., and Wassenaar, A. (2000), “A cross-cultural study on escalation of commitment behavior in software projects”, *MIS Quarterly*, Vol. 24(2), pp. 299–325.
- Kettenring, J. R. (1971), “Canonical analysis of several sets of variables”, *Biometrika*, Vol. 58(3), pp. 433–451.
- Khan, G. F., Sarstedt, M., Shiau, W.-L., Hair, J. F., Ringle, C. M., and Fritze, M. (2019), “Methodological research on partial least squares structural equation modeling (PLS-SEM): An analysis based on social network approaches”, *Internet Research*, Vol. 29(3), pp. 398–406.

- Klein, A. and Moosbrugger, H. (2000), “Maximum likelihood estimation of latent interaction effects with the LMS method”, *Psychometrika*, Vol. 65(4), pp. 457–474.
- Klesel, M., Schuberth, F., Henseler, J., and Niehaves, B. (2019), “A test for multigroup comparison using partial least squares path modeling”, *Internet Research*, Vol. 29(3), pp. 464–477.
- Kline, R. B. (2015), *Principles and Practice of Structural Equation Modeling*, Taylor & Francis Ltd.
- Lai, K. (2019), “Confidence interval for RMSEA or CFI difference between nonnested models”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 27(1), pp. 16–32.
- Landis, R. S., Edwards, B. D., and Cortina, J. M. (2009), *Statistical and Methodological Myths and Urban Legends: Doctrine, Verity, and Fable in the Organizational and Social Sciences*, chap. On the Practice of Allowing Correlated Residuals among Indicators in Structural Equation Models, Routledge/Taylor & Francis Group, New York, pp. 193–214.
- Law, K. S. and Wong, C.-S. (1999), “Multidimensional constructs in structural equation analysis: An illustration using the job perception and job satisfaction constructs”, *Journal of Management*, Vol. 25(2), pp. 143–160.
- Leek, J. T. and Peng, R. D. (2015), “What is the question?”, *Science*, Vol. 347(6228), pp. 1314–1315.
- Little, T. D., Cunningham, W. A., Shahar, G., and Widaman, K. F. (2002), “To parcel or not to parcel: Exploring the question, weighing the merits”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 9(2), pp. 151–173.
- Little, T. D., Rhemtulla, M., Gibson, K., and Schoemann, A. M. (2013), “Why the items versus parcels controversy needn’t be one”, *Psychological Methods*, Vol. 18(3), pp. 285–300.
- Lohmöller, J.-B. (1989), *Latent Variable Path Modeling with Partial Least Squares*, Physica-Verlag, Heidelberg.

- MacCallum, R. C. (2003), “2001 presidential address: Working with imperfect models”, *Multivariate Behavioral Research*, Vol. 38(1), pp. 113–139.
- MacKenzie, S. B. and Podsakoff, P. M. (2012), “Common method bias in marketing: Causes, mechanisms, and procedural remedies”, *Journal of Retailing*, Vol. 88(4), pp. 542–555.
- Marsh, H. W. and Hau, K.-T. (1996), “Assessing goodness of fit: Is parsimony always desirable?”, *The Journal of Experimental Education*, Vol. 64(4), pp. 364–390.
- Marsh, H. W., Hau, K.-T., and Wen, Z. (2004), “In search of golden rules: Comment on hypothesis-testing approaches to setting cutoff values for fit indexes and dangers in overgeneralizing hu and bentler’s (1999) findings”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 11(3), pp. 320–341.
- McDonald, R. P. and Ho, M.-H. R. (2002), “Principles and practice in reporting structural equation analyses”, *Psychological Methods*, Vol. 7(1), p. 64.
- McNeish, D. (2018), “Thanks coefficient alpha, we’ll take it from here.”, *Psychological Methods*, Vol. 23(3), pp. 412–433.
- Mertens, W. and Recker, J. (2019), “New guidelines for null hypothesis significance testing in hypothetico-deductive is research”, *Journal of the Association for Information Systems*.
- Monecke, A. and Leisch, F. (2012), “semPLS: Structural equation modeling using partial least squares”, *Journal of Statistical Software*, Vol. 48(3).
- Mulaik, S. A., James, L. R., Alstine, J. V., Bennett, N., Lind, S., and Stilwell, C. D. (1989), “Evaluation of goodness-of-fit indices for structural equation models”, *Psychological Bulletin*, Vol. 105(3), pp. 430–445.
- Müller, K. and Wickham, H. (2020), *tibble: Simple Data Frames*, R package version 3.0.0.  
**URL:** <https://CRAN.R-project.org/package=tibble>
- Müller, T., Schuberth, F., and Henseler, J. (2018), “PLS path modeling - a confirmatory approach to study tourism technology and tourist behavior”, *Journal of Hospitality and Tourism Technology*, Vol. 9(3), pp. 249–266.

- Muthén, B. O. (2002), “Beyond SEM: General latent variable modeling”, *Behaviormetrika*, Vol. 29(1), pp. 81–117.
- Niemand, T. and Mai, R. (2018), “Flexible cutoff values for fit indices in the evaluation of structural equation models”, *Journal of the Academy of Marketing Science*, Vol. 46(6), pp. 1148–1172.
- Nitzl, C. (2010), “Eine anwenderorientierte Einfuehrung in die Partial Least Square (PLS)-Methode”, in “Arbeitspapier”, 21, Universitaet Hamburg, Institut fuer Industrielles Management, Hamburg.
- Noonan, R. and Wold, H. (1982), “PLS path modeling with indirectly observed variables: A comparison of alternative estimates for the latent variable”, in Jöreskog, K. G. and Wold, H., (Eds.), “Systems Under Indirect Observation: Causality, Structure, Prediction Part II”, North-Holland, Amsterdam, pp. 75–94.
- Nunkoo, R., Teeroovengadam, V., Thomas, P., and Leonard, L. (2017), “Integrating service quality as a second-order factor in a customer satisfaction and loyalty model”, *International Journal of Contemporary Hospitality Management*, Vol. 29(12), pp. 2978–3005.
- Padilla, M. A. and Vepriusky, A. (2012), “Correlation attenuation due to measurement error”, *Educational and Psychological Measurement*, Vol. 72(5), pp. 827–846.
- Park, I., , Sharman, R., Rao, H. R., and and (2015), “Disaster experience and hospital information systems: An examination of perceived information assurance, risk, resilience, and HIS usefulness”, *MIS Quarterly*, Vol. 39(2), pp. 317–344.
- Paxton, P., Curran, P. J., Bollen, K. A., Kirby, J., and Chen, F. (2001), “Monte Carlo experiments: Design and implementation”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 8(2), pp. 287–312.
- Podsakoff, P. M., MacKenzie, S. B., and Podsakoff, N. P. (2012), “Sources of method bias in social science research and recommendations on how to control it”, *Annual Review of Psychology*, Vol. 63, pp. 539–569.

- Polites, G. L., Roberts, N., and Thatcher, J. (2012), “Conceptualizing models using multidimensional constructs: A review and guidelines for their use”, *European Journal of Information Systems*, Vol. 21(1), pp. 22–48.
- R Core Team (2017), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
- R Core Team (2020), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
- Rademaker, M. and Schamberger, T. (2020), *cSEM.DGP: Generate data for structural equation models*, R package version 0.1.0.  
**URL:** <https://github.com/M-E-Rademaker/cSEM.DGP>
- Rademaker, M. E. and Schuberth, F. (2020), *cSEM: Composite-Based Structural Equation Modeling*, R package version 0.2.0.  
**URL:** <https://github.com/M-E-Rademaker/cSEM>
- Rademaker, M. E., Schuberth, F., and Dijkstra, T. K. (2019), “Measurement error correlation within blocks of indicators in consistent partial least squares”, *Internet Research*, Vol. 29(3), pp. 448–463.
- Ramayah, T., Cheah, J., Chuah, F., Ting, H., and Memon, M. A. (2016), *Partial Least Squares Structural Equation Modeling (PLS-SEM) using SmartPLS 3.0: An Updated Guide and Practical Guide to Statistical Analysis*, Pearson, Singapore, 2nd ed.
- Ray, S., Danks, N. P., and Estrada, J. M. V. (2019), *seminr: Domain-Specific Language for Building PLS Structural Equation Models*, R package version 1.0.1.  
**URL:** <https://CRAN.R-project.org/package=seminr>
- Raykov, T., Marcoulides, G. A., and Patelis, T. (2014), “The importance of the assumption of uncorrelated errors in psychometric theory”, *Educational and Psychological Measurement*, Vol. 75(4), pp. 634–647.
- Rhemtulla, M., van Bork, R., and Borsboom, D. (2020), “Worse than measurement error: Consequences of inappropriate latent variable measurement models”, *Psychological Methods*, Vol. 25(1), pp. 30–45.

- Rigdon, E. E. (2016), “Choosing PLS path modeling as analytical method in European management research: A realist perspective”, *European Management Journal*, Vol. 34(6), pp. 598–605.
- Rigdon, E. E., Becker, J.-M., and Sarstedt, M. (2019), “Factor indeterminacy as metrological uncertainty: Implications for advancing psychological measurement”, *Multivariate Behavioral Research*, pp. 1–15.
- Rigdon, E. E., Sarstedt, M., and Ringle, C. M. (2017), “On comparing results from CB-SEM and PLS-SEM: Five perspectives and five recommendations”, *Marketing ZFP*, Vol. 39(3), pp. 4–16.
- Ringle, C. M., Sarstedt, M., and Straub, D. (2012), “A critical look at the use of PLS-SEM in MIS quarterly”, *MIS Quarterly*, Vol. 36(1), pp. iii–xiv.
- Ringle, C. M., Wende, S., and Becker, J.-M. (2015), *SmartPLS 3*, <http://www.smartpls.com>, Bönningstedt.
- Rönkkö, M. (2014), “The effects of chance correlations on partial least squares path modeling”, *Organizational Research Methods*, Vol. 17(2), pp. 164–181.
- Rönkkö, M. (2017), *matrixpls: Matrix-based Partial Least Squares Estimation*, R package version 1.0.5.  
**URL:** <https://github.com/mronkko/matrixpls>
- Rönkkö, M., McIntosh, C. N., Antonakis, J., and Edwards, J. R. (2016), “Partial least squares path modeling: Time for some serious second thoughts”, *Journal of Operations Management*, Vol. 47-48, pp. 9–27.
- Rosseel, Y. (2012), “lavaan: An R package for structural equation modeling”, *Journal of Statistical Software*, Vol. 48(2), pp. 1–36.  
**URL:** <http://www.jstatsoft.org/v48/i02/>
- Rossiter, J. R. (2002), “The C-OAR-SE procedure for scale development in marketing”, *International Journal of Research in Marketing*, Vol. 19(4), pp. 305–335.

- RStudio Team (2015), *RStudio: Integrated Development Environment for R*, RStudio, Inc., Boston, MA.  
**URL:** <http://www.rstudio.com/>
- Rubio, D. M. and Gillespie, D. F. (1995), “Problems with error in structural equation models”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 2(4), pp. 367–378.
- Rueda, L., Benitez, J., and Braojos, J. (2017), “From traditional education technologies to student satisfaction in management education: A theory of the role of social media applications”, *Information & Management*, Vol. 54(8), pp. 1059–1071.
- Sanchez, G., Trinchera, L., and Russolillo, G. (2017), *plspm: Tools for Partial Least Squares Path Modeling (PLS-PM)*, R package version 0.4.9.  
**URL:** <https://CRAN.R-project.org/package=plspm>
- Saris, W. E. and Aalberts, C. (2003), “Different explanations for correlated disturbance terms in MTMM studies”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 10(2), pp. 193–213.
- Sarstedt, M., Hair, J. F., Cheah, J.-H., Becker, J.-M., and Ringle, C. M. (2019), “How to specify, estimate, and validate higher-order constructs in PLS-SEM”, *Australasian Marketing Journal*, Vol. 27(3), pp. 197–211.
- Sarstedt, M., Hair, J. F., Ringle, C. M., Thiele, K. O., and Gudergan, S. P. (2016), “Estimation issues with PLS and CBSEM: Where the bias lies!”, *Journal of Business Research*, Vol. 69(10), pp. 3998–4010.
- Sarstedt, M., Henseler, J., and Ringle, C. M. (2011), “Multigroup analysis in partial least squares (PLS) path modeling: Alternative methods and empirical results”, in “Advances in International Marketing”, Emerald Group Publishing Limited, pp. 195–218.
- Satorra, A. and Bentler, P. M. (2001), “A scaled difference chi-square test statistic for moment structure analysis”, *Psychometrika*, Vol. 66(4), pp. 507–514.



- Saylors, R. and Trafimow, D. (2020), “Why the increasing use of complex causal models is a problem: On the danger sophisticated theoretical narratives pose to truth”, *Organizational Research Methods*.
- Schamberger, T., Schubert, F., Henseler, J., and Dijkstra, T. K. (2020), “Robust partial least squares path modeling”, *Behaviormetrika*, Vol. 47, pp. 307–334.
- Schneeweiss, H. (1993), “Consistency at large in models with latent variables”, in Haagen, K., Bartholomew, D. J., and Deistler, M., (Eds.), “Statistical Modelling and Latent Variables”, North-Holland, Amsterdam, pp. 299–322.
- Schubert, F., Henseler, J., and Dijkstra, T. K. (2018a), “Confirmatory composite analysis”, *Frontiers in Psychology*, Vol. 9.
- Schubert, F., Henseler, J., and Dijkstra, T. K. (2018b), “Partial least squares path modeling using ordinal categorical indicators”, *Quality & Quantity*, Vol. 52(1), pp. 9–35.
- Schubert, F., Chamberger, T., and Dijkstra, T. K. (2017), *MoMpoly: Non-Iterative Method of Moments for Polynomial Factor Models*, R package version 0.1.4.
- Shmueli, G. (2010), “To explain or to predict?”, *Statistical Science*, Vol. 25(3), pp. 289–310.
- Shmueli, G., Ray, S., Estrada, J. M. V., and Chatla, S. B. (2016), “The elephant in the room: Predictive performance of PLS models”, *Journal of Business Research*, Vol. 69(10), pp. 4552–4564.
- Shmueli, G., Sarstedt, M., Hair, J. F., Cheah, J.-H., Ting, H., Vaithilingam, S., and Ringle, C. M. (2019), “Predictive model assessment in PLS-SEM: Guidelines for using PLSpredict”, *European Journal of Marketing*, Vol. 53(11), pp. 2322–2347.
- Simon, H. (1969), *The Sciences of the Artificial*, Cambridge, MA.
- Sivo, S. A., Fan, X., Witta, E. L., and Willse, J. T. (2006), “The search for "optimal" cutoff properties: Fit index criteria in structural equation modeling”, *The Journal of Experimental Education*, Vol. 74(3), pp. 267–288.

- Steiger, J. H. (2007), “Understanding the limitations of global fit assessment in structural equation modeling”, *Personality and Individual Differences*, Vol. 42(5), pp. 893–898.
- Steiger, J. H. and Lind, J. C. (1980), “Statistically-based tests for the number of common factors”, *Paper presented at the Psychometrika Society Meeting, Iowa City, IA*.
- Tenenhaus, M., Amanto, S., and Vinzi, V. E. (2004), “A global goodness-of-fit index for PLS structural equation modelling”, in “Proceedings of the XLII SIS Scientific Meeting”, pp. 739–742.
- Tenenhaus, M., Vinzi, V. E., Chatelin, Y.-M., and Lauro, C. (2005), “PLS path modeling”, *Computational Statistics & Data Analysis*, Vol. 48(1), pp. 159–205.
- Thomson, G. H. (1934), “The meaning of "i" in the estimate of "g"”, *British Journal of Psychology. General Section*, Vol. 25(1), pp. 92–99.
- Tucker, L. R. and Lewis, C. (1973), “A reliability coefficient for maximum likelihood factor analysis”, *Psychometrika*, Vol. 38(1), pp. 1–10.
- van Riel, A. C., Henseler, J., Kemény, I., and Sasovova, Z. (2017), “Estimating hierarchical constructs using consistent partial least squares”, *Industrial Management & Data Systems*, Vol. 117(3), pp. 459–477.
- Varadhan, R. (2015), *alabama: Constrained Nonlinear Optimization*, R package version 2015.3-1.  
**URL:** <https://CRAN.R-project.org/package=alabama>
- Venables, W. N. and Ripley, B. D. (2002), *Modern Applied Statistics with S*, Springer, New York, 4th ed.
- Venkatesh, V., Chan, F. K., and Thong, J. Y. (2012), “Designing e-government services: Key service attributes and citizens’ preference structures”, *Journal of Operations Management*, Vol. 30(1-2), pp. 116–133.
- Wasserstein, R. L., Schirm, A. L., and Lazar, N. A. (2019), “Moving to a world beyond " $p \leq 0.05$ "”, *The American Statistician*, Vol. 73, pp. 1–19.

- Westfall, P. H., Henning, K. S., and Howell, R. D. (2012), “The effect of error correlation on interfactor correlation in psychometric measurement”, *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 19(1), pp. 99–117.
- Wickham, H. (2011), “testthat: Get started with testing”, *The R Journal*, Vol. 3, pp. 5–10.
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Mueller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019), “Welcome to the tidyverse”, *Journal of Open Source Software*, Vol. 4(43), p. 1686.
- Wickham, H., François, R., Henry, L., and Müller, K. (2020), *dplyr: A Grammar of Data Manipulation*, R package version 0.8.4.  
**URL:** <https://CRAN.R-project.org/package=dplyr>
- Wickham, H. and Henry, L. (2020), *tidyr: Tidy Messy Data*, R package version 1.0.2.  
**URL:** <https://CRAN.R-project.org/package=tidyr>
- Wickham, H. and Hesselberth, J. (2020), *pkgdown: Make Static HTML Documentation for a Package*, R package version 1.5.0.  
**URL:** <https://CRAN.R-project.org/package=pkgdown>
- Wilks, S. S. (1938), “The large-sample distribution of the likelihood ratio for testing composite hypotheses”, *The Annals of Mathematical Statistics*, Vol. 9(1), pp. 60–62.
- Wilson, B. (2010), “Using PLS to investigate interaction effects between higher order branding constructs”, in Esposito Vinzi, V., Chin, W. W., Henseler, J., and Wang, H., (Eds.), “Handbook of Partial Least Squares: Concepts, Methods and Applications”, Springer, Berlin, pp. 621–652.
- Wilson, B. and Henseler, J. (2007), “Modeling reflective higher-order constructs using three approaches with PLS path modeling: A Monte Carlo comparison”, in Thyne, M. and Deans, K. R., (Eds.), “Conference Proceedings ANZMAC 2007”, pp. 791–800.
- Wold, H. (1975), *Path Models With Latent Variables: The NIPALS Approach*, Academic Press, New York.

- Wold, H. (1982), “Soft modeling: The basic design and some extensions”, in Jöreskog, K. G. and Wold, H., (Eds.), “Systems under Indirect Observation: Causality, Structure, Prediction Part II”, North-Holland, Amsterdam, pp. 1–54.
- Wooldridge, J. M. (2010), *Econometric Analysis of Cross Section and Panel Data*, MIT Press, 2 ed.
- Wu, Y.-L. and Li, E. Y. (2018), “Marketing mix, customer value, and customer loyalty in social commerce”, *Internet Research*, Vol. 28(1), pp. 74–104.
- Xie, Y. (2020), *knitr: A General-Purpose Package for Dynamic Report Generation in R*, R package version 1.28.  
**URL:** <https://yihui.org/knitr/>
- Yan, Y., Zhang, X., Zha, X., Jiang, T., Qin, L., and Li, Z. (2017), “Decision quality and satisfaction: The effects of online information sources and self-efficacy”, *Internet Research*, Vol. 27(4), pp. 885–904.
- Yang, H., Lee, H., and Zo, H. (2017), “User acceptance of smart home services: An extension of the theory of planned behavior”, *Industrial Management & Data Systems*, Vol. 117(1), pp. 68–89.
- Yim, B. and Leem, B. (2013), “The effect of the supply chain social capital”, *Industrial Management & Data Systems*, Vol. 113(3), pp. 324–349.
- Yuan, K.-H. (2005), “Fit indices versus test statistics”, *Multivariate Behavioral Research*, Vol. 40(1), pp. 115–148.
- Zimmerman, D. W. (2007), “Correction for attenuation with biased reliability estimates and correlated errors in populations and samples”, *Educational and Psychological Measurement*, Vol. 67(6), pp. 920–939.

---

# Eidesstattliche Erklärung

---

Hiermit erkläre ich, dass ich diese Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass diese Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde.

Weiterhin erkläre ich, dass das Promotionsverfahren zu keinem früheren Zeitpunkt an einer anderen Hochschule oder bei einem anderen Fachbereich beantragt wurde.

Würzburg, Juni 2020

# Curriculum Vitae

## MANUEL RADEMAKER (NÉ STEINER)

(born March 13th 1987, German citizen)

---

University of Wuerzburg, Department of Economics  
Sanderring 2, 97070 Wuerzburg, Germany  
manuel.rademaker@uni-wuerzburg.de, Phone: +49 931 31 86952

- EDUCATION**      Doctoral Studies of Econometrics, University of Wuerzburg, 2014 – 2020  
M.Sc. in Economics (Grade 1.6), University of Wuerzburg, 2012 – 2014  
B.Sc. in Economics (Grade 1.9), University of Fribourg, Switzerland, 2008 – 2012
- Abitur, Waldorfschule Vaihingen a.d. Enz, 2006
- EMPLOYMENT**    Lecturer in Introductory Statistics, Cooperative State University Bad Mergentheim, 2018, 2019, 2020  
Research Assistant, Chair of Econometrics, University of Wuerzburg, 2014 – present  
Student Research Assistant, Chair of Econometrics, University of Wuerzburg, 2013 – 2014
- RESEARCH INTERESTS**    Structural equation modeling (SEM), composite-based SEM, the partial least squares (PLS) algorithm, fit assessment and fit indices in composite-based SEM, static and dynamic linear and non-linear panel data, instrumental variable estimation, effective data visualization
- PUBLICATIONS & WORKING PAPERS**    Schubert, F., Rademaker, M.E., and Henseler, J. (forthcoming), *Estimating and assessing second-order constructs using PLS-PM: the case of composites of composites*, Industrial Management & Data Systems.
- Rademaker, M. E., Schubert, F., Dijkstra, T. K. (2019) *Measurement error correlation within blocks of indicators in consistent partial least squares: Issues and remedies*. Internet Research, Vol. 29(3), pp. 448-463.
- Hoehn, B., Schubert, F., Steiner, M. (2014) Dealing with Heteroskedasticity, Autocorrelation and Endogeneity in German Audit Fee Panel Data - Comparing Approaches (Working paper)
- CONFERENCE PRESENTATIONS**    Rademaker, Manuel; Schubert, Florian (2019) cSEM: A package for composite-based SEM. Meeting of the Working Group Structural Equation Modeling, Tübingen, Germany. February/March 27 – 1
- SEMINARS**            Bavarian Graduate Program in Economics  
*Advanced Econometrics*, (Prof. Jeffrey Wooldridge), Muggendorf, 2016
- Bavarian Graduate Program in Economics  
*Advanced Econometrics*, (Prof. Arthur van Soest), Muggendorf, 2015

**TEACHING EXPERIENCE** Introductory Statistics I, Introductory Statistics II, Computer Lab in Regression Analysis, Econometrics I, Econometrics II, Econometrics III

**SOFTWARE** Professional: R, R package development  
Advanced:  $\text{\LaTeX}$ , MS Excel, Git/GitHub  
Basic: MS Word, Stata, Gretl, Mathematica, Matlab, EViews, HTML, Python, Java, C, C++

Creator and maintainer of the R packages `cSEM` and `cSEM.DGP`

**LANGUAGES** German (Mother tongue), English (fluent), French (basic)

**REFERENCES** Professor Dr. Martin Kukuk  
Chair for Econometrics  
University of Wuerzburg  
martin.kukuk@uni-wuerzburg.de

Wuerzburg, November 7, 2020