Conducting Monte Carlo simulations with PLS-PM and other variance-based estimators for structural equation models: a tutorial using the R package cSEM

Received 26 July 2022
Revised 31 December 2022
Accepted 9 February 2023

Tamara Schamberger
Department of Design, Production and Management, University of Twente, Enschede, The Netherlands and
Faculty of Business Management and Economics, Julius-Maximilians-Universität Würzburg, Würzburg, Germany

Abstract

Purpose – Structural equation modeling (SEM) is a well-established and frequently applied method in various disciplines. New methods in the context of SEM are being introduced in an ongoing manner. Since formal proof of statistical properties is difficult or impossible, new methods are frequently justified using Monte Carlo simulations. For SEM with covariance-based estimators, several tools are available to perform Monte Carlo simulations. Moreover, several guidelines on how to conduct a Monte Carlo simulation for SEM with these tools have been introduced. In contrast, software to estimate structural equation models with variance-based estimators such as partial least squares path modeling (PLS-PM) is limited.

Design/methodology/approach – As a remedy, the R package *cSEM* which allows researchers to estimate structural equation models and to perform Monte Carlo simulations for SEM with variance-based estimators has been introduced. This manuscript provides guidelines on how to conduct a Monte Carlo simulation for SEM with variance-based estimators using the R packages *cSEM* and *cSEM.DGP*.

Findings – The author introduces and recommends a six-step procedure to be followed in conducting each Monte Carlo simulation.

Originality/value – For each of the steps, common design patterns are given. Moreover, these guidelines are illustrated by an example Monte Carlo simulation with ready-to-use R code showing that PLS-PM needs the constructs to be embedded in a nomological net to yield valuable results.

Keywords Monte Carlo simulation, Composites, Structural equation modelling, Guidelines, cSEM, R

Paper type Technical paper

1. Introduction

Structural equation modeling (SEM) is a popular method in social and behavioral sciences such as marketing research (Steenkamp and Baumgartner, 2000), psychology (Fassinger, 1987; MacCallum and Austin, 2000; Higgins, 2002), business management (Hult *et al.*, 2006; Mak and Sockel, 2001) and information systems research (Urbach *et al.*, 2010). Traditionally, SEM uses covariance-based estimators such as a maximum likelihood (ML, Jöreskog, 1970) estimator or a generalized least squares (GLS, Jöreskog and Goldberger, 1972) estimator to obtain parameter estimates. Besides covariance-based estimators, variance-based estimators such as partial least squares path modeling (PLS-PM, Wold, 1975), consistent partial least

An earlier version of this article was published in the following PhD thesis: Schamberger, T. (2022) Methodological Advances in Composite-based Structural Equation Modeling. University of Würzburg/ University of Twente, https://doi.org/10.3990/1.9789036553759.

Industrial Management & Data Systems
Vol. 123 No. 6, 2023
pp. 1789-1813

0263-5577
DOI 10.1108/IMDS-07-2022-0418

squares (PLSc, Dijkstra and Henseler, 2015a) or generalized structured component analysis (GSCA, Hwang and Takane, 2004) have been introduced to estimate structural equation models.

SEM in general and variance-based estimators for structural equation models in particular are constantly being refined and new methods are constantly being introduced, such as a new criterion to assess discriminant validity (Roemer *et al.*, 2021), a combination of ML and PLS-PM to obtain parameter estimates (Ghasemy *et al.*, 2021), or an approach to estimate second-order constructs (Schuberth *et al.*, 2020). These need methodological and theoretical justification, e.g. statistical properties such as the bias or the standard errors need to be evaluated. Instead of formally proving statistical properties, a common practice is to provide evidence for these by using Monte Carlo simulations. For this purpose, a series of samples is drawn from a given population and analyzed using the method of interest. To evaluate the estimates, they can be compared to their known population counterparts. The Monte Carlo method was introduced in 1949 by Metropolis and Ulam in physics. With increasing computational power, the Monte Carlo method gained increasing attention and Monte Carlo simulations are now frequently applied in various research fields, including physics (Landau and Binder, 2021), econometrics (Hendry, 1984), biology (Manly, 2018) and medicine (Mode, 2011).

In general, Monte Carlo simulations can be performed in any statistical environment. Nevertheless, R (R Core Team, 2020) is popular among researchers because of its open-source nature. Several guidelines are available for Monte Carlo simulations for SEM with covariance-based estimators in R (Lee, 2015; Rosseel, 2014). However, these functions cannot be used for structural equation models with variance-based estimators. For that purpose, the R package *cSEM* (Rademaker and Schuberth, 2020) can be used. *cSEM* provides researchers with a tool to estimate structural equation models with PLS-PM, PLSc, GSCA and other variance-based estimators. Moreover, *cSEM* is accompanied by the R package *cSEM.DGP* to simulate data for predetermined structural equation models (Rademaker and Schamberger, 2020). Thus, it provides users with all the necessary tools to perform Monte Carlo simulations for SEM with variance-based estimators in R. However, as yet, no tutorial is available on how to conduct Monte Carlo simulations for SEM using *cSEM*.

To address this gap, this paper provides guidelines for performing Monte Carlo simulations for SEM with PLS-PM and other variance-based estimators using the open-source software *cSEM*. To demonstrate the guidelines, I conduct an exemplary Monte Carlo simulation to investigate PLS-PM's and PLSc's finite sample behavior, particularly regarding the consequences of sample correlations among measurement errors on statistical inference.

The remainder of the paper is structured as follows. Section 2 gives an overview of Monte Carlo simulations for SEM. In addition, this section provides an overview of existing software tools for performing Monte Carlo simulations for SEM with PLS-PM and other variance-based estimators. This demonstrates the need to provide guidelines for Monte Carlo simulations for SEM with *cSEM*. Section 3 gives step-by-step guidelines on how to conduct Monte Carlo simulations for SEM using the R package *cSEM*. Section 4 illustrates the guidelines by an exemplary Monte Carlo simulation. The paper closes with a conclusion in Section 5.

2. The need for further guidelines on Monte Carlo simulations for SEM

"Monte Carlo is the confluence of deterministic, stochastic, and computational methods with computer generated random numbers an important component" (Hurd, 1985). Although it is assumed that the first application of Monte Carlo methods was to estimate the number π, the first published Monte Carlo method was as an approach to answer questions in physics (Metropolis and Ulam, 1949). Originally, the method was introduced as an approach to obtain probabilities that could not be obtained analytically. It is based on the idea of the law of large numbers and other asymptotic theorems of statistics (Metropolis and Ulam, 1949). Monte Carlo simulations are often applied to estimate parameters of interest by using a large number of simulated samples. These are often generated using inverse sampling (Johansen, 2010). Inverse sampling proceeds to generate values according to the quantile function of the considered distribution, such that these follow the considered distribution. For each of the simulated samples, the parameters of interest are estimated. Naturally, "the estimate will never be confined within given limits with certainty, but only - if the number of trials is great - with great probability" (Metropolis and Ulam, 1949). Consequently, the parameter estimates can be evaluated, e.g. in terms of bias, consistency and efficiency. With increasing computational power, Monte Carlo simulations have gained increasing attention in various research fields – also in the context of SEM.

Monte Carlo simulations for SEM are used for different purposes. First, they are used to evaluate the performance of methodological enhancements. On the one hand, Monte Carlo simulations are used to evaluate the bias and standard errors of new approaches to estimate model parameters such as PLSc (Dijkstra and Henseler, 2015b), PLSe1 (Huang, 2013) and PLSe2 (Ghasemy *et al.*, 2021) in finite samples. On the other hand, they are used to evaluate the performance of enhancements for model assessment such as goodness-of-fit indices, tests for the overall model fit (Moshagen, 2012), or bootstrap-based techniques for inference (Jung *et al.*, 2019). These kinds of Monte Carlo simulation are often conducted to demonstrate the performance of an enhancement in the specific situation for which it was developed. Second, Monte Carlo simulations for SEM are often conducted to demonstrate the limitations of SEM approaches in specific situations (Rönkkö and Evermann, 2013). Third, Monte Carlo simulations for SEM are used to compare the performance of different estimators (Reinartz *et al.*, 2009; Hwang *et al.*, 2017).

Since Monte Carlo simulations for SEM are frequently applied, various tools have been proposed to perform such simulations. For example, the commercial software Mplus (Muthén and Muthén, 1998-2017) or LISREL (Jöreskog and Sörbom, 1993) provide tools for data generation and model estimation using covariance-based approaches such as ML or GLS. Besides commercial software, open-source software such as the R packages *lavaan* (Rosseel, 2012) and *simsem* (Pornprasertmanit *et al.*, 2021) are available for Monte Carlo simulations for SEM with covariance-based estimators. The R package *lavaan* is probably the most popular R package for covariance-based SEM. It comprises most of the available approaches for covariance-based SEM and interacts with the R package *simsem* to simulate data. Consequently, it provides users with all necessary tools to perform Monte Carlo simulations with covariance-based SEM. Nevertheless, these tools cannot be used to perform Monte Carlo simulations with PLS-PM and other variance-based estimators for structural equation models.

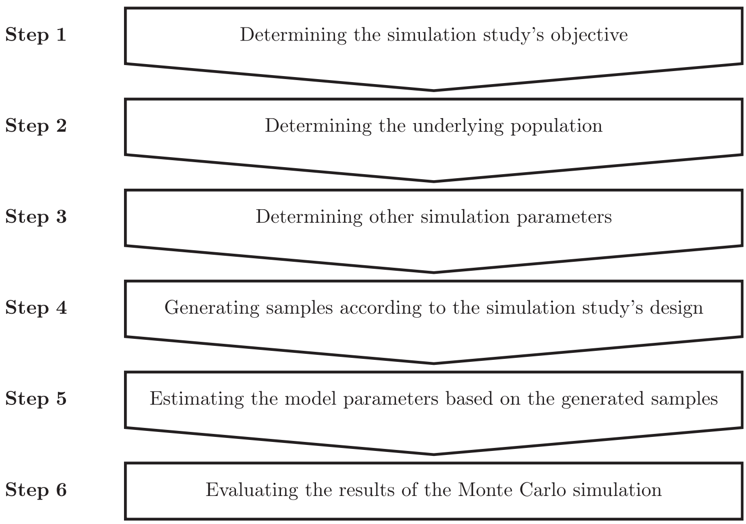
Variance-based estimators for structural equation models such as PLS-PM or GSCA gained traction over the last two decades. Thus, Monte Carlo simulations with PLS-PM and other variance-based estimators for structural equation models increasingly gained attention. Nevertheless, compared to covariance-based estimators, existing software tools for this type of SEM are still limited. Consequently, one rarely comes across guidelines for this type of Monte Carlo simulation. Available commercial software such as ADANCO (Henseler, 2019) or SmartPLS (Ringle *et al.*, 2015) do not provide tools for generating data and thus cannot directly be applied to conduct a Monte Carlo simulation. In contrast, the R package *matrixpls* (Rönkkö, 2017) can be used to obtain parameter estimates with variance-based estimators such as PLS-PM, PLSc and GSCA. The package interacts with the R package *simsem* to simulate data and can thus also be used for Monte Carlo simulations. Nevertheless, *matrixpls* does not provide users with the possibility of obtaining model parameters with approaches such as Kettenring's (1971) approaches for generalized canonical correlation analysis.

Additionally, higher-order constructs (Schuberth *et al.*, 2020) or nonlinear structural equation models containing higher-order terms (Dijkstra and Schermelleh-Engel, 2014) cannot be estimated using *matrixpls*. While most available software for SEM uses the data as input to obtain parameter estimates, *matrixpls* uses a variance-covariance matrix as input to obtain parameter estimates, which makes it less flexible in terms of evaluating a model's prediction performance.

As an alternative, the R package *cSEM* was introduced. This package comprises a majority of variance-based estimators for SEM and is well established (Rademaker and Schuberth, 2020). To elaborate, researchers using *cSEM* are provided with several possibilities to determine standard errors of the parameter estimates, such as jackknife (Tukey, 1958) or bootstrap (Efron, 1979). Moreover, nonlinear structural equation models and models containing higher-order constructs can be estimated. Besides model estimation, *cSEM* provides users with the possibility of identifying inadmissible solutions, performing out-of-sample predictions and assessing the models. Finally, it is accompanied by the R package *cSEM.DGP* (Rademaker and Schamberger, 2020) which was designed to simulate data for predefined structural equation models. Consequently, *cSEM* provides users with all the necessary tools to perform Monte Carlo simulations with PLS-PM and other variance-based estimators for structural equation models in R, which is why I provide guidelines for applying these tools in the following section.

3. Monte Carlo simulations for SEM using the R package cSEM

To support researchers in conducting Monte Carlo simulations for SEM with variance-based estimators using the R package *cSEM*, I present guidelines in this section by explaining the different steps that need to be performed, namely determining the simulation study's objective, determining the underlying population, determining other simulation parameters, generating samples according to the simulation study's design, estimating the model parameters based on the generated samples and evaluating the results of the Monte Carlo simulation as displayed in Figure 1.



Source(s): Author's own creation

Figure 1.
Steps to perform a Monte Carlo simulation with variance-based estimators using the R package *cSEM*

3.1 Determining the objective of the Monte Carlo simulation

The starting point of a Monte Carlo simulation is to determine its objective. Considering PLS-PM and other variance-based estimators, most Monte Carlo simulations have one of the following objectives: (1) to demonstrate the performance of a new methodology (e.g. Dijkstra and Henseler, 2015a; Henseler *et al.*, 2012; Henseler and Sarstedt, 2013; Henseler *et al.*, 2015, 2016; Klesel *et al.*, 2019); (2) to compare the results of different estimators (e.g. Dijkstra and Henseler, 2015a; Henseler and Chin, 2010; Henseler and Sarstedt, 2013; Reinartz *et al.*, 2009) and (3) to evaluate the performance of an existing approach in a specific situation (e.g. Henseler, 2010; Henseler *et al.*, 2012, 2015; Rönkkö and Evermann, 2013).

3.2 Determining the underlying population

To determine the underlying population, the population model has to be specified in the first step. Structural equation models consist of two parts: (1) equations describing the relations between the constructs and (2) equations describing the relations between the constructs and their related observed variables. In the following explications, I will follow the common notation for structural equation models (Bollen, 1989). The relations between the constructs can be written as follows:

$$\boldsymbol{\eta} = \boldsymbol{B}\boldsymbol{\eta} + \boldsymbol{\zeta} \tag{1}$$

where $\boldsymbol{\eta}$ is a vector of the constructs η_j, $\boldsymbol{B}$ contains the corresponding path coefficients and $\boldsymbol{\zeta}$ is a vector of structural error terms.

Besides the relations between the constructs, two different types of relations between the constructs and their related observed variables can be distinguished. First, if the theoretical concepts are modeled as common factors which explain the variance-covariance structure of their related observed variables, the relations between the construct η_j and its observed variables $\boldsymbol{x}_j$ are represented in terms of loadings (Jöreskog, 1969):

$$\boldsymbol{x}_j = \boldsymbol{\lambda}_j \eta_j + \boldsymbol{\varepsilon}_j \tag{2}$$

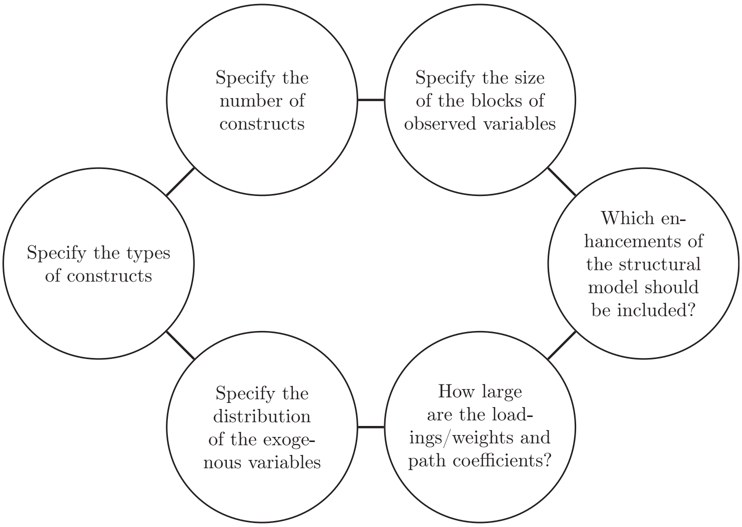
The observed variables related to the $j-$ th construct are stored in a block of observed variables $\boldsymbol{x}_j$, with $\boldsymbol{\lambda}_j$ as a vector of loadings and ε_j as a vector of measurement errors. Second, the theoretical concept can be modeled as a composite which emerges from its observed variables and conveys all information between its observed variables and the other variables in the model (Henseler, 2021). Composites are linear combinations of observed variables:

$$\eta_j = \boldsymbol{w}_j' \boldsymbol{x}_j \tag{3}$$

The vector $\boldsymbol{x}_j$ represents the block of observed variables related to the composite η_j, and $\boldsymbol{w}_j$ is a vector of the corresponding weights.

Figure 2 shows the relevant steps to determine the underlying population. The concrete population is directly influenced by the objective of the Monte Carlo simulation, and all the population model's parameters should be determined accordingly. In addition, the choice of population model may be limited by the estimation methods used, since, for example, not every estimator can handle nonlinear structural models. Consequently, it is not possible to provide general guidelines on how to specify a population model. Nevertheless, Table 1 provides an overview of common design patterns for Monte Carlo simulations using PLS-PM. To evaluate the influence of the population model on the simulation results, researchers can consider different population models for their simulation study.

Once the model has been determined, it needs to be specified in lavaan syntax for later estimation. To specify a model in lavaan syntax, it is specified as a string in which the



Source(s): Author's own creation

Figure 2.
How to specify a population

Table 1.
Common design patterns for Monte Carlo simulations with PLS-PM

Design pattern	Commonly used	References
Number of constructs	2	Henseler *et al.* (2015) and Rönkkö and Evermann (2013)
	3	Henseler *et al.* (2012) and Henseler (2010)
	4	Henseler and Sarstedt (2013) and Klesel *et al.* (2019)
	6	Reinartz *et al.* (2009)
Size of the blocks of observed variables	2	Henseler (2010) and Reinartz *et al.* (2009)
	3	Rönkkö and Evermann (2013), Henseler and Sarstedt (2013), Henseler *et al.* (2015) and Klesel *et al.* (2019)
	4	Reinartz *et al.* (2009)
Enhancements of the structural model	Interaction terms	Henseler and Chin (2010)
	Nonlinear structural relations	Dijkstra and Schermelleh-Engel (2014)
	Non-recursive structural model	Klesel *et al.* (2022)

Source(s): Author's own creation

structural relations are determined using ~, correlations are determined using ~~, relations between common factors and their observed variables are determined using =~, and relations between composites and their observed variables are determined using <~.

As Figure 2 shows, the distribution of the exogenous variables needs to be determined. Using this distribution and the population relations between the variables, the distribution of the observed variables can be determined. For simplicity, it is customary to assume a multivariate standard normal distribution for the exogenous variables (e.g. Jannoo *et al.*, 2014; Jung *et al.*, 2019; Rademaker *et al.*, 2019), therefore, samples can be generated by using the population variance-covariance matrix.

As an alternative to calculating the population variance-covariance matrix by hand, the population correlation matrix can be obtained using the R package *cSEM.DGP* (Rademaker and Schamberger, 2020). In *cSEM.DGP*, several assumptions about the error terms are imposed. First, the structural error terms ζ_k are assumed to be uncorrelated with those constructs η_j that solely occur in an exogenous position and thus are not explained through other constructs. Second, the measurement errors ε_j are assumed to be uncorrelated among one another, with the measurement errors of other constructs and the structural error terms. To obtain the population correlation matrix, in the first step, the model has to be specified in lavaan syntax including the population parameters. This population model has to be used as input for the function generateData() to obtain the population correlation matrix. Further, the argument .return_type of the generateData() function needs to be set to "cor" and the argument .empirical needs to be set to "TRUE" to obtain the considered matrix.

3.3 Determining other simulation parameters

Besides determining the population for the Monte Carlo simulation, other simulation parameters have to be specified. The simulation parameters that need to be considered depend on the simulation study's objective. Nevertheless, some parameters need to be considered for all simulation studies:

(1) Sample size:

For a Monte Carlo simulation, samples need to be drawn from the above-described population. To provide empirical evidence for statistical properties such as consistency or asymptotic efficiency, large samples should be considered. Moreover, Monte Carlo simulations are often applied to evaluate the estimator's finite sample behavior. Thus, sample sizes of 100 (Dijkstra and Schermelleh-Engel, 2014; Hair *et al.*, 2017; Henseler and Chin, 2010; Klesel *et al.*, 2019; Roemer *et al.*, 2021), 200 (Dijkstra and Schermelleh-Engel, 2014; Henseler and Chin, 2010) and 500 (Hair *et al.*, 2017; Henseler and Chin, 2010; Klesel *et al.*, 2019; Roemer *et al.*, 2021) observations are commonly used for Monte Carlo simulations with PLS-PM and other variance-based estimators. To evaluate the effect of the sample size on the performance of an approach, different sample sizes should be considered.

(2) Number of draws:

As the Monte Carlo principle relies on the law of large numbers, several samples need to be drawn and used to estimate the model parameters. A high number of draws implies estimates that are more precise (Metropolis and Ulam, 1949). However, a high number of draws implies a longer computational time. For simulation studies with variance-based estimators for structural equation models, it is common to set the number of draws to 500 (Hwang *et al.*, 2010; Goodhue *et al.*, 2012; Jung *et al.*, 2019) or 1,000 (Aguirre-Urreta and Rönkkö, 2015; Dijkstra and Henseler, 2015b).

3.4 Generating samples according to the simulation study's design

Once the design of the Monte Carlo simulation has been specified, samples need to be generated according to the simulation design, i.e. the population and the other simulation parameters. Using *cSEM*, the easiest way to simulate data according to the simulation design is to use the generateData() function of the package *cSEM.DGP*. In doing so, samples from a multivariate normal distribution and also from a distribution with predefined values for the skewness and kurtosis can be drawn:

```
generateData(.model, .N, .skewness, .kurtosis, .return_type)
```

To obtain nonnormally distributed samples with predefined values for skewness and kurtosis, the Fleishman–Vale–Maurelli procedure (Fleishman, 1978; Vale and Maurelli, 1983) is used. The argument .model contains the population model in lavaan syntax including the population parameters as described above. The argument .*N* equals the corresponding sample size and .skewness and .kurtosis can be used if a distribution of the observed variables different to the multivariate normal distribution is considered. As default, .skewness and .kurtosis are set to the values of the normal distribution, i.e. to a skewness of 0 and a kurtosis of 3. The argument .return_type determines the requested output format. By default, it is set to "data.frame", thus, it does not need to be adjusted to obtain a simulated sample. Note that *cSEM.DGP* is still limited in terms of model complexity, i.e. considering the number of concepts that can be taken into account. If the defined population model is not supported by *cSEM.DGP*, samples according to the simulation design can be drawn by using other well-developed R packages such as *simsem*. For more flexibility considering data generation, the R package *covsim* (Grønneberg and Foldnes, 2017), for example, could be used. As an alternative, samples can be drawn by using the corresponding quantile function of the considered distribution.

3.5 Estimating the model parameters based on the generated samples

Once the samples have been drawn according to the simulation design, the model parameters need to be estimated. Using the R package *cSEM*, model estimation is done using the function csem(), which has a variety of possible arguments. In its simplest form, csem() only requires a sample (.data) and a model in lavaan syntax (.model):

```
csem(.data, .model)
```

Added to this, csem() has a variety of optional arguments. The optional arguments all have default values and thus only need further definition if different options should be used. For a detailed explanation of all optional arguments and their corresponding default values, please refer to the manual of the *cSEM* package (Rademaker and Schuberth, 2020).

The csem() result is a list with the parameter estimates and further information about the estimation. Moreover, several post-estimation functions like assess(), infer(), predict(), summarize() and verify() can be applied. The post-estimation function verify() checks for inadmissible results. Inadmissible results are results which did not converge, where at least one standardized loading is larger than 1, where the construct correlation matrix is not positive semi-definite, at least one reliable estimate is larger than 1, or where the observed variables' population variance-covariance matrix is not positive definite (Rademaker and Schuberth, 2020). An inadmissible result indicates a problem with the model for the available sample. Consequently, depending on the simulation study's objective, inadmissible results could be removed from the simulation and replaced with admissible counterparts. However, in terms of good research practice, the share of inadmissible results generated through the simulation study should be reported. A huge share of inadmissible results in a Monte Carlo simulation indicates that the corresponding estimator is problematic in the considered research situation. The post-estimation function summarize() gives a summary of the estimation results, including the parameter estimates and their standard errors – if resampling was applied. For an explanation of the other post-estimation functions, please refer to Rademaker and Schuberth (2020).

3.6 Evaluating the results of the Monte Carlo simulation

As a last step of a Monte Carlo simulation with PLS-PM and other variance-based estimators, the results need to be evaluated. The type of evaluation depends highly on the Monte Carlo simulation's objective. Therefore, general guidelines cannot be provided. Nevertheless, some possibilities are given in the following exposition.

If an approach to obtain parameter estimates is evaluated, statistical properties, such as bias, consistency and efficiency should often be evaluated. The bias can be estimated by comparing the average parameter estimates to their population values: $\frac{1}{n_{draws}}\sum_{i=1}^{n_{draws}}\widehat{\theta}_i - \theta_i$. To provide evidence for the consistency of parameter estimates, one often evaluates whether the bias decreases for increasing sample sizes and whether estimations based on very large samples – for example, 100,000 observations – show any bias. Further, the efficiency of parameter estimates obtained by different approaches is evaluated by comparing the standard errors for different sample sizes. To evaluate both the bias and the variance jointly, measures such as the root mean squared error or the mean absolute error are applied. Besides presenting the results in the form of a table (Dijkstra and Henseler, 2015a), visualizing the results by using barplots (Schuberth *et al.*, 2018), density plots (Schamberger *et al.*, 2020), boxplots (Jannoo *et al.*, 2014) or similar can improve the comprehensibility of the results.

3.7 Providing information about the Monte Carlo simulation

The previous subsections provide detailed guidelines on how to conduct a Monte Carlo simulation for PLS-PM and other variance-based estimators using cSEM. Besides designing and running a Monte Carlo simulation, and presenting the respective results, a detailed explanation of the procedure used for the Monte Carlo simulation should be provided in the interest of good research practice. In the following, I provide concise guidelines on what information should be reported.

First, the specific procedure that the researchers followed in their Monte Carlo simulation needs to be presented. This includes (1) the objective of the Monte Carlo simulation, (2) the population model including all population parameters and (3) all other simulation parameters including sample size, number of draws and any other parameters chosen. Second, the software must be given, including its version. For R packages, the functions must be specified with all arguments, e.g. PLS modes or weighting schemes. Third, data storage and availability is an important issue. Researchers should always ensure that their results are reproducible. Consequently, the data should either be made publicly available or researchers need to ensure that the data can be reproduced by using a seed during the simulation study. Using a seed ensures that although the data is randomly drawn, the same random samples are drawn when the code is rerun. If a seed was used during the simulation, this should be mentioned in the simulation design. In summary, researchers should provide all necessary information to ensure that their results can be replicated by others.

4. Example: The consequences of sample correlations among measurement errors on statistical inference concerning the PLS-PM results

To illustrate the guidelines for conducting Monte Carlo simulations with PLS-PM and other variance-based estimators for structural equation models using the R package *cSEM*, I conducted an exemplary Monte Carlo simulation.

4.1 Determining the objective

The exemplary Monte Carlo simulation investigates PLS-PM's and PLSc's finite sample behavior, particularly regarding the consequences of sample correlations among measurement errors on statistical inference. Since correlations of zero are hardly ever found in empirical research, PLS-PM's assumption of uncorrelated error terms can yield a bimodal distribution of the parameter estimates in finite samples. This effect has previously been discussed in the literature (Rönkkö, 2014). The illustrative Monte Carlo simulation presented here evaluates the effect of this assumption in empirical research on an extreme case with uncorrelated constructs. Moreover, the effect of disattenuation, i.e. using PLSc instead of PLS-PM, is examined. Specifically, the study evaluates whether a t-test regarding a path coefficient estimate which is zero in the population is able to hold the level of significance if the parameter estimates are obtained with PLS-PM and PLSc if one construct is modeled as a composite and one construct is modeled as a common factor. Additionally, we evaluate the guidelines that propose that PLS-PM and PLSc need a nomological net – and thus the constructs should not be isolated, i.e. need nonzero relations to other variables than their related observed variables – to provide valuable estimates (Henseler *et al.*, 2014). Consequently, we evaluate whether the bimodal distribution of the parameter estimates vanishes and whether the t-test is able to hold its significance level if the model is embedded in more contexts.

The illustrative Monte Carlo simulation's objective can be classified using the classes described in Section 3.1. In this example, two approaches are compared and the performance of the two approaches is evaluated in a specific situation. Consequently, the exemplary Monte Carlo simulation can be classified in the second group (comparison of estimators) and third group (evaluation of the performance of existing approaches) of Monte Carlo simulations for SEM.

4.2 Underlying population

Some parts of the underlying population are directly motivated by the Monte Carlo simulation's objective. Since we consider an extreme case of two uncorrelated constructs in which one construct is modeled as a common factor and another is modeled as a composite, the types of constructs are already specified. Further, the exogenous construct is modeled as a common factor and the endogenous construct is modeled as a composite. The population model should be non-recursive, and exactly two constructs are included. Also, the two constructs are related to three observed variables. Consequently, the following structural relation is considered:

$$\eta_2 = 0.0 \cdot \eta_1 + \zeta_2 \tag{4}$$

We use the following relations between the observed variables and their associated constructs:

$$x_{11} = 0.9 \cdot \eta_1 + \varepsilon_{11} \tag{5}$$

$$x_{12} = 0.8 \cdot \eta_1 + \varepsilon_{12} \tag{6}$$

$$x_{13} = 0.7 \cdot \eta_1 + \varepsilon_{13} \tag{7}$$

$$\eta_2 = 0.6 \cdot x_{21} + 0.4 \cdot x_{22} + 0.2 \cdot x_{23} \tag{8}$$

The random measurement errors are assumed to be uncorrelated and ζ_2 is assumed to be uncorrelated with both η_1 and the measurement errors. The variances of the measurement errors of the first block are set to 0.19, 0.36 and 0.51, respectively. The correlations between the observed variables of the composite are set to 0.5 each, such that the composite η_2 has unit variance. For simplicity and following other Monte Carlo simulations with variance-based estimators (e.g. Jannoo *et al.*, 2014), the observed variables are assumed to be multivariate normally distributed. Figure 3 displays the population model.

Once the population model has been determined, the model has to be specified in lavaan syntax as input for the later simulation:

```
model <- '
# Relations between the constructs and the observed variables
eta1 =~ x11 + x12 + x13
eta2 <~ x21 + x22 + x23

# Structural relations
eta2 ~ eta1'
```

Code I

The observed variables' population variance-covariance matrix depends on the loadings, the intra-block correlations of the composite's observed variables, the measurement errors' variances and the structural relations. Note that *cSEM* as well as *cSEM.DGP* rely on correlations and not on variances and covariances. Due to the zero path between the constructs η_1 and η_2, the correlations between the observed variables associated with η_1 and η_2 are zero. The correlations between the observed variables x_{ji} and x_{jk} of a common factor η_j can be obtained as follows:

$$\text{Cov}(x_{ji}, x_{jk}) = \lambda_{ji} \cdot \lambda_{jk}$$

The correlations between the observed variables of a composite are given in the model definition. Consequently, the observed variables' population correlation matrix is given as follows:

$$\Sigma = \begin{array}{c} \begin{array}{cccccc} \underline{x_{11}} & \underline{x_{12}} & \underline{x_{13}} & \underline{x_{21}} & \underline{x_{22}} & \underline{x_{23}} \end{array} \\ \begin{pmatrix} 1.00 & & & & & \\ 0.72 & 1.00 & & & & \\ 0.63 & 0.56 & 1.00 & & & \\ 0.00 & 0.00 & 0.00 & 1.00 & & \\ 0.00 & 0.00 & 0.00 & 0.50 & 1.00 & \\ 0.00 & 0.00 & 0.00 & 0.50 & 0.50 & 1.00 \end{pmatrix} \end{array} \tag{9}$$

As explained above, the observed variables' population correlation matrix can be obtained using the R package *cSEM.DGP* by using the model including the population parameters as input for the generateData() function:

```
library(cSEM.DGP)
model_dgp <- '
# Relations between the constructs and the observed variables
eta1 =~ 0.9*x11 + 0.8*x12 + 0.7*x13
eta2 <~ 0.6*x21 + 0.4*x22 + 0.2*x23

# Structural relations
eta2 ~ 0.0*eta1

# Intra-block correlations
x21 ~~ 0.5*x22 + 0.5*x23
x22 ~~ 0.5*x23'

Sigma <- generateData(.model = model_dgp, .return_type = "cor", .empirical = TRUE)
```

Code II

To evaluate the influence of the population model on the simulation results, as well as the guidelines that state PLS-PM and PLSc need the constructs to be embedded in a nomological net and not to be isolated, we considered a second population model. The relation between η_1 and η_2, as well as how they relate to their observed variables remain the same as for the first population model. In addition, both constructs are connected to one composite each with a

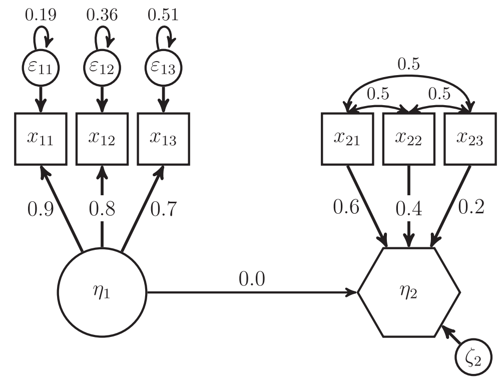


Figure 3.
Population model with two constructs

Source(s): Author's own creation

nonzero path. The following relations between the constructs η_1 and η_2 and the new composites η_3 and η_4 are considered:

$$\eta_2 = 0.0 \cdot \eta_1 + 0.3 \cdot \eta_3 + \zeta_2 \tag{10}$$

$$\eta_1 = 0.3 \cdot \eta_4 + \zeta_1 \tag{11}$$

Both composites η_3 and η_4 are assumed to be composed of three indicators each. The population model is displayed in Figure 4. The observed variables are assumed to be multivariate normally distributed. Note that the population correlation matrix as well as the R code for simulating the model with four constructs are given in the Appendix.

4.3 Determining other simulation parameters

Considering the rest of the simulation design, the following parameters are chosen: First, two sample sizes, namely 200 and 500 observations, are considered to evaluate the estimators' finite sample behavior. Second, the number of draws is set to 500. Consequently, in total two estimators, two population models with two sample sizes each, and one number of draws, i.e. eight conditions, are considered.

4.4 Data generation

Using the *cSEM.DGP* package, we generated samples with 200 observations by first determining the population model including the model parameters in lavaan syntax, and afterward using the generateData() function:

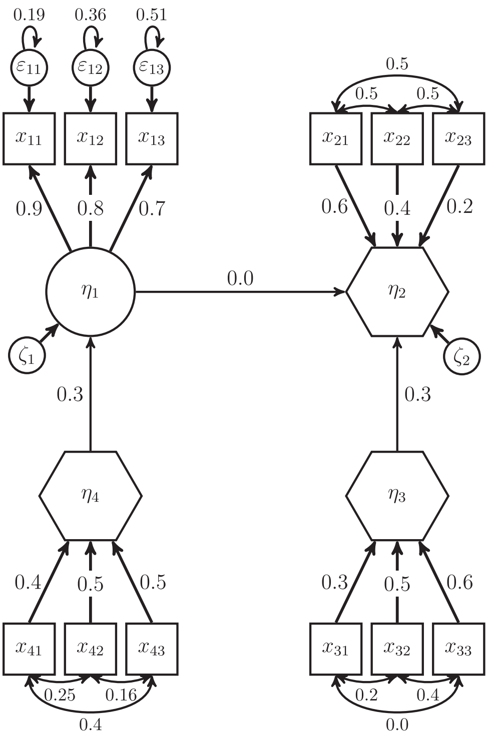
```
library(cSEM.DGP)
model_dgp <- '
# Relations between the constructs and the observed variables
eta1 =~ 0.9*x11 + 0.8*x12 + 0.7*x13
eta2 <~ 0.6*x21 + 0.4*x22 + 0.2*x23

# Structural relations
eta2 ~ 0.0*eta1

# Intra-block correlations
x21 ~~ 0.5*x22 + 0.5*x23
x22 ~~ 0.5*x23'

data <- generateData(.model = model_dgp, .N = 200)
```

Code III



Source(s): Author's own creation

Figure 4.
Population model with four constructs

Since the observed variables are assumed to be multivariate normally distributed, the other arguments of generateData() do not have to be adjusted from their defaults.

Note that according to the simulation design, several samples need to be simulated. These can either all be simulated first and estimated afterward or each sample can be estimated before simulating a new sample. The samples with 500 observations are simulated similarly by setting the argument .N to 500.

4.5 Model estimation

To investigate PLS-PM's and PLSc's finite sample behavior, particularly regarding the consequences of sample correlations among measurement errors on statistical inference, the model described above in lavaan syntax and the simulated samples need to be used as input for the csem() function. Additionally, the standard errors of the path coefficient estimates need to be obtained. Consequently, the argument.resample_method needs to be adjusted from its default. For example, the standard errors can be obtained using a bootstrap approach:

```
csem(.data, .model, .resample_method = "bootstrap")
```

To ensure that all results rely on the same number of bootstrap samples and that all bootstrap results taken into account are in fact admissible, the argument .handle_inadmissibles needs to be set to "replace". Also, csem() uses a disattenuation for common factors as default. To obtain PLS-PM estimates instead of PLSc estimates in the case of common factors, the argument .disattenuate needs to be set to "FALSE". The argument .seed is set to "123" to ensure that the bootstrap results can be reproduced. The other optional arguments are not changed from their defaults. Consequently, the following two commands provided the requested estimates:

```
# Estimation of the model based on the simulated sample
res_PLSc <- csem(.model = model, .data = data, .resample_method = "bootstrap",
                 .handle_inadmissibles = "replace", .seed = 123)

# Estimation of the model based on the simulated sample without disattenuation
res_PLS <- csem(.model = model, .data = data, .resample_method = "bootstrap",
                .disattenuate = FALSE, .handle_inadmissibles = "replace",
                .seed = 123)
```

Code IV

In the illustrative Monte Carlo simulation, for each sample size, 500 samples needed to be simulated and estimated using PLSc and PLS-PM using the simulation design. To do so, we applied the following code:

```
library(cSEM)
library(cSEM.DGP)
# Define model for the data generation
model_dgp <- '
# Relations between the constructs and the observed variables
eta1 =~ 0.9*x11 + 0.8*x12 + 0.7*x13
eta2 <~ 0.6*x21 + 0.4*x22 + 0.2*x23

# Structural relations
eta2 ~ 0.0*eta1

# Intra-block correlations
x21 ~~ 0.5*x22 + 0.5*x23
x22 ~~ 0.5*x23'

# Define model for the parameter estimation
model <- '
# Relations between the constructs and the observed variables
eta1 =~ x11 + x12 + x13
eta2 <~ x21 + x22 + x23

# Structural relations
eta2 ~ eta1'

# Define lists to store the simulation results
res_PLS <- list()
res_PLSc <- list()
i <- 1
j <- 0
set.seed(123)
while(i < 501){
  data <- generateData(.model = model_dgp, .N = 200)
  res_PLSc_temp <- csem(.model = model, .data = data)
  res_PLS_temp <- csem(.model = model, .data = data, .disattenuate = FALSE)

  if(sum(verify(res_PLSc_temp)) == 0 && sum(verify(res_PLS_temp)) == 0){
    res_PLSc[[i]] <- csem(.model = model, .data = data, .resample_method = "bootstrap",
                          .handle_inadmissibles = "replace", .seed = 123)
    res_PLS[[i]] <- csem(.model = model, .data = data, .resample_method = "bootstrap",
                         .disattenuate = FALSE, .handle_inadmissibles = "replace",
                         .seed = 123)
    i <- i+1
  }else{
    j <- j +1
  }
}
```

Code V

Note that we included only admissible solutions because inadmissible solutions can significantly destroy the results. Moreover, in practice, when an inadmissible solution occurs, researchers are advised to reconsider their model. Therefore, it would not be fair to consider solutions in the Monte Carlo simulation that would be ignored in practice. If estimating one of the generated samples yields an inadmissible result, this is replaced by an admissible one. Consequently, all results are based on 500 admissible solutions which are based on 499 admissible bootstrap results each. To obtain the simulation results for the population model with two constructs and a sample size of 500, the argument N has to be set to 500. I give the code that needs to be run to obtain the results for the population model with four constructs in the Appendix.

4.6 Evaluating the simulation results

Considering the situation outlined above, we expected that the t-test would not hold the significance level for the path coefficient in the case of the population model with two constructs. Further, in line with previous findings in the literature where similar population models were used, we expected that the parameter estimates would show a bimodal distribution in this case (Rönkkö, 2014). For the population model with four constructs, we expected that the bimodal form of the distribution would vanish for the path coefficient between η_1 and η_2. Further, we assumed that the t-test is able to hold its significance level in this case, which would also be in line with previous findings in the literature that showed that PLS-PM and PLSc need the constructs to be embedded in a nomological net (Henseler *et al.*, 2014). Considering the two sample sizes, namely 200 and 500 observations, we expected that the results would become more accurate in the sense that the bias as well as the standard errors would decrease.

To evaluate the results, in the first step, the share of significant path coefficient estimates has to be determined for the various sample sizes and population models. Note that for the larger model with four constructs, only the path coefficient between the common factor η_1 and the composite η_2 is considered to be able to evaluate the population model's effect on the test's performance regarding significance for this path coefficient. The other parameters are not considered here.

Using bootstrap, standard errors $\hat{\sigma}_{\hat{b}_{21}}$ were obtained for all path coefficient estimates. These estimated standard errors can be used to test the null hypothesis: $H_0 : b_{21} = 0$ for each estimation, achieved by using the t-test statistic

$$t = \frac{\hat{b}_{21}}{\hat{\sigma}_{\hat{b}_{21}}} \qquad (12)$$

which is also provided in the output of the csem() function. The test statistic is asymptotically standard normally distributed. The significance level α is set to 0.01, 0.05 and 0.1, respectively. To obtain the p-value, the asymptotic distribution of the test statistic was used. The p-value of the test statistic is compared to the level of significance to decide whether the structural coefficient is significant at the given level of significance. If the share of significant coefficients is smaller or equal to the level of significance, the test was able to hold the level of significance. Besides evaluating the share of significant path coefficients in terms of a t-test, we could also use bootstrap confidence intervals. Since the share of significant path coefficients was similar if confidence intervals were used, we do not report the results here. Nevertheless, I give a table with the corresponding results in the Appendix.

The results of the illustrative Monte Carlo simulations have been stored in lists where each list element contains the results for one simulated sample. As explained earlier, every result of the csem() function is a list, and several post-estimation functions can be applied. The post-estimation function summarize() yields an overview of the estimation, and the parameter estimates including their estimated standard errors, t-statistics and p values:

```
summarize(res_PLS[[6]])
```

```
## --------------------------------------------------------------------------------
## ----------------------------------- Overview -----------------------------------
##
##  General information:
##  ------------------------
##  Estimation status                  = Ok
##  Number of observations             = 200
##  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                      = No
##
##  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                        = 123
##
##  Construct details:
##  ------------------
##  Name  Modeled as     Order         Mode
##
##  eta2  Common factor  First order   "modeA"
##  eta1  Composite      First order   "modeB"
##
## ----------------------------------- Estimates ----------------------------------
##
## Estimated path coefficients:
## ============================
##                                                               CI_percentile
##   Path           Estimate  Std. error   t-stat.   p-value         95%
##   eta1 ~ eta2      0.1830      0.0859    2.1303    0.0331 [-0.1254; 0.3349 ]
##
```

Code VI

```
## Estimated loadings:
## ===================
##                                                              CI_percentile
##   Loading         Estimate  Std. error   t-stat.   p-value        95%
##   eta2 =~ x21       0.9029      0.1481    6.0974    0.0000 [ 0.7069; 0.9489 ]
##   eta2 =~ x22       0.7613      0.1467    5.1909    0.0000 [ 0.5050; 0.8906 ]
##   eta2 =~ x23       0.8292      0.1550    5.3480    0.0000 [ 0.6359; 0.9166 ]
##   eta1 =~ x11       0.3576      0.3214    1.1127    0.2659 [-0.3561; 0.8706 ]
##   eta1 =~ x12       0.4583      0.2967    1.5450    0.1223 [-0.1798; 0.9182 ]
##   eta1 =~ x13       0.9894      0.2438    4.0583    0.0000 [ 0.0130; 0.9970 ]
##
## Estimated weights:
## ==================
##                                                              CI_percentile
##   Weight          Estimate  Std. error   t-stat.   p-value        95%
##   eta2 <~ x21       0.5301      0.1369    3.8734    0.0001 [ 0.2315; 0.7165 ]
##   eta2 <~ x22       0.2337      0.1491    1.5668    0.1172 [-0.0790; 0.4995 ]
##   eta2 <~ x23       0.4142      0.1635    2.5337    0.0113 [ 0.0882; 0.6824 ]
##   eta1 <~ x11      -0.1713      0.4426   -0.3871    0.6987 [-0.8943; 0.8710 ]
##   eta1 <~ x12       0.0202      0.4422    0.0456    0.9636 [-0.7712; 0.9690 ]
##   eta1 <~ x13       1.0633      0.3871    2.7470    0.0060 [-0.3044; 1.2160 ]
##
## Estimated indicator correlations:
## =================================
##                                                              CI_percentile
##   Correlation     Estimate  Std. error   t-stat.   p-value        95%
##   x11 ~~ x12        0.4641      0.0593    7.8234    0.0000 [ 0.3310; 0.5701 ]
##   x11 ~~ x13        0.4886      0.0527    9.2698    0.0000 [ 0.3806; 0.5949 ]
##   x12 ~~ x13        0.4869      0.0647    7.5254    0.0000 [ 0.3500; 0.5951 ]
##
## ----------------------------------- Effects ----------------------------------
##
## Estimated total effects:
## ========================
##                                                               CI_percentile
##   Total effect     Estimate  Std. error   t-stat.   p-value        95%
##   eta1 ~ eta2        0.1830      0.0859    2.1303    0.0331 [-0.1254; 0.3349 ]
## ------------------------------------------------------------------------------
```

Code VII

Consequently, the p-value corresponding to the t-statistic of Equation (12) can be extracted by using "summarize(res_PLS[[i]])$Estimates$path_estimates$p_value". The complete results are displayed in Figure 5 using a barplot. Also, the concrete results in terms of numbers are given in the Appendix. Further, besides the results in terms of test statistics, I give the results using confidence intervals to detect significant path coefficients in the Appendix.

The results show that for the model with two constructs, the t-test for the path coefficient b_{21} is not able to hold the level of significance. Only for a sample size of 500 and $\alpha = 10\%$, the PLS-PM results are close to the level of significance. For all other sample sizes and levels of significance, we detected higher shares of significant path coefficients for models estimated with both PLS-PM and PLSc. Consequently, significant relations between the two constructs are detected too often, which can lead to false conclusions. Moreover, disattenuation of the estimates does not increase the estimators' ability to correctly identify nonsignificant relations. If the model with four constructs, i.e. with a higher context, is considered, the significance test for the path coefficient b_{21} is able to hold the level of significance for all values of α and all sample sizes. The results for models estimated with PLS-PM and PLSc only hardly differ at all.

Eventually, Figure 6 shows the density for the path coefficient estimates between η_1 and η_2 for the different simulation conditions. The results demonstrate that for the model with two

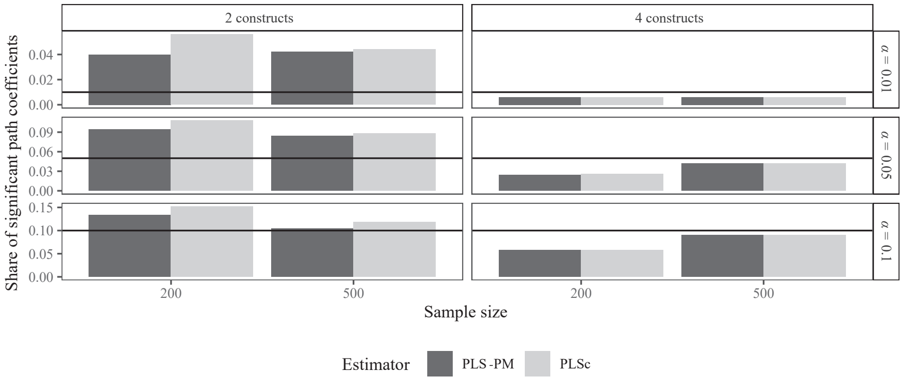


Figure 5. Share of significant path coefficients between η_1 and η_2

Source(s): Author's own estimation

constructs, the path coefficient estimates show the bimodal distribution as expected. In contrast, for the model with four constructs, the results do not have the bimodal distribution. This is in line with previous findings in the literature which state that PLS-PM and PLSc need models with more context to yield valuable results (Henseler *et al.*, 2014).

As stated before, I ensured that all simulation results only consider admissible results. Therefore, in the analysis, I removed all inadmissible results. Nevertheless, the shares of inadmissible solutions are displayed in Table 2 [1].

This shows that for the model with two constructs which are connected through a zero path, more than 50% of the simulated samples result in an inadmissible solution. Moreover, for the model with four constructs, the share of inadmissible solutions is much lower and also

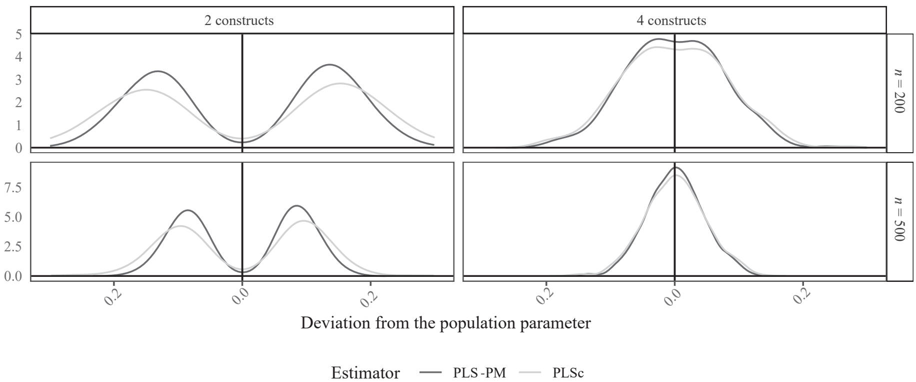


Figure 6. Density for the path coefficient estimates

Source(s): Author's own estimation

Table 2. Share of inadmissible solutions

Sample size	Number of constructs	Share of inadmissible solutions
200	2	0.5292
500	2	0.5136
200	4	0.1379
500	4	0.0512

Source(s): Author's own estimation

decreases with increasing sample size. High shares of inadmissible results indicate a problem with the considered model and thus we find support for the hypothesis that PLS-PM and PLSc need the constructs to be embedded in a nomological net. The results of the Monte Carlo simulation are completely in line with the expectations that were described above, and thus are also in line with previous findings in the literature.

5. Conclusion

This paper has presented guidelines for performing Monte Carlo simulations with PLS-PM and other variance-based estimators for structural equation models using the R package *cSEM*. First, the need for further guidelines on Monte Carlo simulations for SEM is explained. Second, step-by-step guidelines for how to conduct a Monte-Carlo simulation with PLS-PM and other variance-based estimators for structural equation models are given. The guidelines are accompanied by an illustrative example and the corresponding ready-to-use R code. Moreover, suggestions for evaluating the results and the general design for this type of Monte Carlo simulation are provided. The latter is based on commonly used design patterns for Monte Carlo simulations with variance-based estimators given in the literature. Finally, guidelines on information which should be provided about the Monte Carlo simulation are given.

Future research should provide guidelines for Monte Carlo simulations with PLS-PM and other variance-based estimators for structural equation models using the R package *cSEM* for more specific situations, e.g. in using the post-estimation function predict(). Moreover, future research should provide guidelines for larger Monte Carlo simulations for which using high-performance clusters could be helpful. Nevertheless, the interested reader is referred to Rademaker and Schuberth (2020) for a more detailed explanation of the R package *cSEM* and the optional arguments.

Note

1. Note that these only refer to the original estimation and do not consider the inadmissible solutions generated during the resampling to obtain standard errors.

References

Aguirre-Urreta, M. and Rönkkö, M. (2015), "Between-group equivalence in comparisons using PLS: results from three simulation studies", *Communications of the Association for Information Systems*, Vol. 36, pp. 33-51.

Bollen, K.A. (1989), *Structural Equations with Latent Variables*, Wiley, New York.

Dijkstra, T.K. and Henseler, J. (2015a), "Consistent and asymptotically normal PLS estimators for linear structural equations", *Computational Statistics and Data Analysis*, Vol. 81, pp. 10-23, doi: 10.1016/j.csda.2014.07.008.

Dijkstra, T.K. and Henseler, J. (2015b), "Consistent partial least squares path modeling", *MIS Quarterly*, Vol. 39 No. 2, pp. 29-316, doi: 10.25300/MISQ/2015/39.2.02.

Dijkstra, T.K. and Schermelleh-Engel, K. (2014), "Consistent partial least squares for nonlinear structural equation models", *Psychometrika*, Vol. 79 No. 4, pp. 585-604, doi: 10.1007/s11336-013-9370-0.

Efron, B. (1979), "Bootstrap methods: another look at the jackknife", *The Annals of Statistics*, Vol. 7 No. 1, pp. 1-26, doi: 10.1214/aos/1176344552.

Fassinger, R.E. (1987), "Use of structural equation modeling in counseling psychology research", *Journal of Counseling Psychology*, Vol. 34 No. 4, pp. 425-436, doi: 10.1037/0022-0167.34.4.425.

Fleishman, A.I. (1978), "A method for simulating non-normal distributions", *Psychometrika*, Vol. 43 No. 4, pp. 521-532, doi: 10.1007/BF02293811.

Ghasemy, M., Jamil, H. and Gaskin, J.E. (2021), "Have your cake and eat it too: PLSe2 = ML + PLS", *Quality and Quantity*, Vol. 55, pp. 497-541, doi: 10.1007/s11135-020-01013-6.

Goodhue, D.L., Lewis, W. and Thompson, R. (2012), "Does PLS have advantages for small sample size or non-normal data?", *MIS Quarterly*, Vol. 36 No. 3, p. 981, doi: 10.2307/41703490.

Grønneberg, S. and Foldnes, N. (2017), "Covariance model simulation using regular vines", *Psychometrika*, Vol. 82 No. 4, pp. 1035-1051, doi: 10.1007/s11336-017-9569-6.

Hair, J.F., Hult, G.T.M., Ringle, C.M., Sarstedt, M. and Thiele, K.O. (2017), "Mirror, mirror on the wall: a comparative evaluation of composite-based structural equation modeling methods", *Journal of the Academy of Marketing Science*, Vol. 45 No. 5, pp. 616-632, doi: 10.1007/s11747-017-0517-x.

Hendry, D.F. (1984), "Chapter 16 Monte Carlo experimentation in econometrics", in *Handbook of Econometrics*, Elsevier, pp. 937-976, doi: 10.1016/s1573-4412(84)02008-0.

Henseler, J. (2010), "On the convergence of the partial least squares path modeling algorithm", *Computational Statistics*, Vol. 25 No. 1, pp. 107-120, doi: 10.1007/s00180-009-0164-x.

Henseler, J. (2019), *ADANCO 2.0.1*, Composite Modeling GmbH, Kleve.

Henseler, J. (2021), *Composite-Based Structural Equation Modeling: Analyzing Latent and Emergent Variables*, Guilford Press, New York, NY, available at: https://www.ebook.de/de/product/39418152/joerg_henseler_composite_based_structural_equation_modeling_analyzing_latent_and_emergent_variables.html

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 No. 1, pp. 82-109, doi: 10.1080/10705510903439003.

Henseler, J. and Sarstedt, M. (2013), "Goodness-of-fit indices for partial least squares path modeling", *Computational Statistics*, Vol. 28 No. 2, pp. 565-580, doi: 10.1007/s00180-012-0317-1.

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 No. 1, pp. 99-112, doi: 10.1057/ejis.2011.36.

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 No. 2, pp. 182-209, doi: 10.1177/1094428114526928.

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 No. 1, pp. 115-135, doi: 10.1007/s11747-014-0403-8.

Henseler, J., Ringle, C.M. and Sarstedt, M. (2016), "Testing measurement invariance of composites using partial least squares", *International Marketing Review*, Vol. 33 No. 3, pp. 405-431, doi: 10.1108/imr-09-2014-0304.

Higgins, G.E. (2002), "General theory of crime and deviance: a structural equation modeling approach", *Journal of Crime and Justice*, Vol. 25 No. 2, pp. 71-95, doi: 10.1080/0735648x.2002.9721158.

Huang, W. (2013), *Plse: Efficient Estimators and Tests for Partial Least Square*, PhD thesis, University of California, Los Angeles.

Hult, G.T.M., Ketchen, D.J., Cui, A.S., Prud'homme, A.M., Seggie, S.H., Stanko, M.A., Xu, A.S. and Cavusgil, S.T. (2006), "An assessment of the use of structural equation modeling in international business research", *Research Methodology in Strategy and Management*, Emerald (MCB UP), pp. 385-415.

Hurd, C.C. (1985), "A note on early Monte Carlo computations and scientific meetings", *Annals of the History of Computing*, Vol. 7 No. 2, pp. 141-155, doi: 10.1109/MAHC.1985.10019.

Hwang, H. and Takane, Y. (2004), "Generalized structured component analysis", *Psychometrika*, Vol. 69 No. 1, pp. 81-99, doi: 10.1007/bf02295841.

Hwang, H., Ho, M.H.R. and Lee, J. (2010), "Generalized structured component analysis with latent interactions", *Psychometrika*, Vol. 75 No. 2, pp. 228-242, doi: 10.1007/s11336-010-9157-5.

Hwang, H., Takane, Y. and Jung, K. (2017), "Generalized structured component analysis with uniqueness terms for accommodating measurement error", *Frontiers in Psychology*, Vol. 8, p. 2137, doi: 10.3389/fpsyg.2017.02137.

Jannoo, Z., Yap, B., Auchoybur, N. and Lazim, M.A. (2014), "The effect of nonnormality on CB-SEM and PLS-SEM path estimates", *International Journal of Mathematical, Computational, Physical and Quantum Engineering*, Vol. 8 No. 2, pp. 285-291.

Jöreskog, K.G. (1969), "A general approach to confirmatory maximum likelihood factor analysis", *Psychometrika*, Vol. 34 No. 2, pp. 183-202, doi: 10.1007/bf02289343.

Jöreskog, K.G. (1970), "A general method for estimating a linear structural equation system", *ETS Research Bulletin Series*, Vol. 1970 No. 2, pp. i-41, doi: 10.1002/j.2333-8504.1970.tb00783.x.

Jöreskog, K.G. and Goldberger, A.S. (1972), "Factor analysis by generalized least squares", *Psychometrika*, Vol. 37 No. 3, pp. 243-260, doi: 10.1007/bf02306782.

Jöreskog, K.G. and Sörbom, D. (1993), *LISREL 8 : Structural Equation Modeling with the SIMPLIS Command Language*, Scientific Software International Lawrence Erlbaum, Chicago, IL. Hillsdale, N.J.

Johansen, A. (2010), "Monte Carlo methods", in Penelope Peterson, E.B. and McGaw, B. (Eds), *International Encyclopedia of Education*, 3rd ed., Elsevier, pp. 296-303, doi: 10.1016/b978-0-08-044894-7.01543-8.

Jung, K., Lee, J., Gupta, V. and Cho, G. (2019), "Comparison of bootstrap confidence interval methods for GSCA using a Monte Carlo simulation", *Frontiers in Psychology*, Vol. 10, doi: 10.3389/fpsyg.2019.02215.

Kettenring, J.R. (1971), "Canonical analysis of several sets of variables", *Biometrika*, Vol. 58 No. 3, pp. 433-451, doi: 10.1093/biomet/58.3.433.

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 No. 3, pp. 464-477, doi: 10.1108/intr-11-2017-0418.

Klesel, M., Schuberth, F., Niehaves, B. and Henseler, J. (2022), "Multigroup analysis in information systems research using PLS-PM: a systematic investigation of approaches", *The Data Base for Advances in Information Systems*, Vol. 53 No. 3, pp. 26-48, doi: 10.1145/3551783.3551787.

Landau, D. and Binder, K. (2021), *A Guide to Monte Carlo Simulations in Statistical Physics*, 5th ed., Cambridge University Press, Cambridge.

Lee, S. (2015), "Implementing a simulation study using multiple software packages for structural equation modeling", *SAGE Open*, Vol. 5 No. 3, p. 182, 215824401559, doi: 10.1177/2158244015591823.

MacCallum, R.C. and Austin, J.T. (2000), "Applications of structural equation modeling in psychological research", *Annual Review of Psychology*, Vol. 51 No. 1, pp. 201-226, doi: 10.1146/annurev.psych.51.1.201.

Mak, B.L. and Sockel, H. (2001), "A confirmatory factor analysis of IS employee motivation and retention", *Information and Management*, Vol. 38 No. 5, pp. 265-276, doi: 10.1016/s0378-7206(00)00055-0.

Manly, B.F. (2018), *Randomization, Bootstrap and Monte Carlo Methods in Biology*, Chapman and Hall/CRC, doi: 10.1201/9781315273075.

Metropolis, N. and Ulam, S. (1949), "The Monte Carlo method", *Journal of the American Statistical Association*, Vol. 44 No. 247, pp. 335-341.

Mode, C.J. (Ed.) (2011), *Applications of Monte Carlo Methods in Biology, Medicine and Other Fields of Science*, IntechOpen, London.

Moshagen, M. (2012), "The model size effect in SEM: inflated goodness-of-fit statistics are due to the size of the covariance matrix", *Structural Equation Modeling: A Multidisciplinary Journal*, Vol. 19 No. 1, pp. 86-98, doi: 10.1080/10705511.2012.634724.

Muthén, L.K. and Muthén, B.O. (1998-2017), *Mplus*, 8th ed., Muthén & Muthén, Los Angeles, CA.

Pornprasertmanit, S., Miller, P., Schoemann, A. and Jorgensen T.D. (2021), "Simsem: SIMulated structural equation modeling", available at: https://CRAN.R-project.org/package=simsem (R packag eversion 0.5-16).

R Core Team (2020), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna.

Rademaker, M.E. and Schamberger, T. (2020), "cSEM.DGP: generate data for structural equation models", available at: https://github.com/M-E-Rademaker/cSEM.DGP (R package version 0.0.0.9000).

Rademaker, M.E. and Schuberth, F. (2020), "cSEM: composite-based structural equation modeling", available at: https://github.com/M-E-Rademaker/cSEM (R package version 0.5.0).

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 No. 3, pp. 448-463, doi: 10.1108/intr-12-2017-0525.

Reinartz, W., Haenlein, M. and Henseler, J. (2009), "An empirical comparison of the efficacy of covariance-based and variance-based SEM", *International Journal of Research in Marketing*, Vol. 26 No. 4, pp. 332-344, doi: 10.1016/j.ijresmar.2009.08.001.

Ringle, C.M., Wende, S. and Becker, J.M. (2015), "SmartPLS 3", Bönningstedt, available at: http://www.smartpls.com

Rönkkö, M. (2014), "The effects of chance correlations on partial least squares path modeling", *Organizational Research Methods*, Vol. 17 No. 2, pp. 164-181, doi: 10.1177/1094428114525667.

Rönkkö, M. and Evermann, J. (2013), "A critical examination of common beliefs about partial least squares path modeling", *Organizational Research Methods*, Vol. 16 No. 3, pp. 425-448, doi: 10.1177/1094428112474693.

Roemer, E., Schuberth, F. and Henseler, J. (2021), "HTMT2–an improved criterion for assessing discriminant validity in structural equation modeling", *Industrial Management and Data Systems*, Vol. 121 No. 12, pp. 2637-2650, doi: 10.1108/imds-02-2021-0082.

Rönkkö, M. (2017), "Matrixpls: matrix-based partial least squares estimation", R package version 1.0.5.

Rosseel, Y. (2012), "Lavaan: an R package for structural equation modeling", *Journal of Statistical Software*, Vol. 48 No. 2, pp. 1-36, doi: 10.18637/jss.v048.i02.

Rosseel, Y. (2014), "The lavaan tutorial", Ghent University, available at: https://lavaan.ugent.be/tutorial/tutorial.pdf.

Schamberger, T., Schuberth, F., Henseler, J. and Dijkstra, T.K. (2020), "Robust partial least squares path modeling", *Behaviormetrika*, Vol. 47, pp. 307-334, doi: 10.1007/s41237-019-00088-2.

Schuberth, F., Henseler, J. and Dijkstra, T.K. (2018), "Partial least squares path modeling using ordinal categorical indicators", *Quality and Quantity*, Vol. 52 No. 1, pp. 9-35, doi: 10.1007/s11135-016-0401-7.

Schuberth, F., Rademaker, M.E. and Henseler, J. (2020), "Estimating and assessing second-order constructs using PLS-PM: the case of composites of composites", *Industrial Management and Data Systems*, Vol. 120 No. 12, pp. 2211-2241, doi: 10.1108/imds-12-2019-0642.

Steenkamp, J.B.E. and Baumgartner, H. (2000), "On the use of structural equation models for marketing modeling", *International Journal of Research in Marketing*, Vol. 17 Nos 2-3, pp. 195-202, doi: 10.1016/s0167-8116(00)00016-1.

Tukey, J. (1958), "Bias and confidence in not quite large samples", *Annals of Mathematical Statistics*, Vol. 29, p. 614, doi: 10.1214/aoms/1177706647.

Urbach, N. and Ahlemann, F. (2010), "Structural equation modeling in information systems research using partial least squares", *Journal of Information Technology Theory and Application*, Vol. 11 No. 2, pp. 5-40.

Vale, C.D. and Maurelli, V.A. (1983), "Simulating multivariate nonnormal distributions", *Psychometrika*, Vol. 48 No. 3, pp. 465-471, doi: 10.1007/bf02293687.

Wold, H. (1975), *Path Models with Latent Variables: The NIPALS Approach*, Academic Press, New York.

Appendix

Table A1. Observed variables" population correlations for the population model with 4 constructs

x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	x_{31}	x_{32}	x_{33}	x_{41}	x_{42}	x_{43}
1.000											
0.720	1.000										
0.630	0.560	1.000									
0.000	0.000	0.000	1.000								
0.000	0.000	0.000	0.500	1.000							
0.000	0.000	0.000	0.500	0.500	1.000						
0.000	0.000	0.000	0.108	0.096	0.084	1.000					
0.000	0.000	0.000	0.216	0.192	0.168	0.200	1.000				
0.000	0.000	0.000	0.216	0.192	0.168	0.000	0.400	1.000			
0.196	0.174	0.152	0.000	0.000	0.000	0.000	0.000	0.000	1.000		
0.184	0.163	0.143	0.000	0.000	0.000	0.000	0.000	0.000	0.250	1.000	
0.200	0.178	0.155	0.000	0.000	0.000	0.000	0.000	0.000	0.400	0.160	1.000

Source(s): Author's own estimation

Simulation code for the population model with four constructs

```
library(cSEM)
library(cSEM.DGP)
# Model definition in lavaan syntax for the csem function
model <- '
# Relations between the constructs and the observed variables
eta1 =~ x11 + x12 + x13
eta2 <~ x21 + x22 + x23
eta3 <~ x31 + x32 + x33
eta4 <~ x41 + x42 + x43

# Relations between the constructs
eta2 ~ eta1 +  eta3
eta1 ~ eta4'

# Population model with the population values in lavaan syntax
model_dgp <- '
# Relations between the constructs and the observed variables
eta1 =~ 0.9*x11 + 0.8*x12 + 0.7*x13
eta2 <~ 0.6*x21 + 0.4*x22 + 0.2*x23
eta3 <~ 0.3*x31 + 0.5*x32 + 0.6*x33
eta4 <~ 0.4*x41 + 0.5*x42 + 0.5*x43

# Intra block correlations of the observed variables
x21 ~~ 0.5*x22 + 0.5*x23
x22 ~~ 0.5*x23

x31 ~~ 0.2*x32 + 0.0*x33
x32 ~~ 0.4*x33

x41 ~~ 0.25*x42 + 0.4*x43
x42 ~~ 0.16*x43

# Relations between the constructs
eta2 ~ 0.0*eta1 + 0.3*eta3
eta1 ~ 0.3*eta4'
```

Code VIII

```r
# Lists for the simulation results
res_PLS <- list()
res_PLSc <- list()
i <- 1
j <- 0
set.seed(123)
while(i < 501){
  data <- generateData(.model = model_dgp, .N = 200)
  res_PLSc_temp <- csem(.model = model, .data = data)
  res_PLS_temp <- csem(.model = model, .data = data, .disattenuate = FALSE)

  if(sum(verify(res_PLSc_temp)) == 0 && sum(verify(res_PLS_temp)) == 0){
    res_PLSc[[i]] <- csem(.model = model, .data = data, .resample_method = "bootstrap",
                          .handle_inadmissibles = "replace", .seed = 123)
    res_PLS[[i]] <- csem(.model = model, .data = data, .resample_method = "bootstrap",
                         .disattenuate = FALSE, .handle_inadmissibles = "replace",
                         .seed = 123)
    i <- i+1
  }else{
    j <- j +1
  }
}
```

Code IX

Simulation results

number of constructs	sample size	share of significant path coefficients PLS-PM	PLSc
2	200	0.080	0.102
	500	0.062	0.088
4	200	0.028	0.028
	500	0.038	0.038

Source(s): Author's own estimation

Table A2. Share of significant path coefficients for the 95% confidence interval

Number of constructs	n	Estimator	Share of significant path coefficients $\alpha = 1\%$	$\alpha = 5\%$	$\alpha = 10\%$
2	200	PLS-PM	0.040	0.094	0.134
		PLSc	0.056	0.108	0.152
	500	PLS-PM	0.042	0.084	0.104
		PLSc	0.044	0.088	0.118
4	200	PLS-PM	0.006	0.024	0.058
		PLSc	0.006	0.026	0.058
	500	PLS-PM	0.006	0.042	0.090
		PLSc	0.006	0.042	0.090

Source(s): Author's own estimation

Table A3. Share of significant path coefficients for the t-tests

Corresponding author
Tamara Schamberger can be contacted at: t.s.schamberger@utwente.nl