Computational Options for Standard Errors and Test Statistics with Incomplete Normal and Nonnormal Data in SEM

Victoria Savalei & Yves Rosseel

To cite this article: Victoria Savalei & Yves Rosseel (2021): Computational Options for Standard Errors and Test Statistics with Incomplete Normal and Nonnormal Data in SEM, Structural Equation Modeling: A Multidisciplinary Journal, DOI: 10.1080/10705511.2021.1877548

To link to this article: https://doi.org/10.1080/10705511.2021.1877548

© 2021 The Author(s). Published with license by Taylor & Francis Group, LLC.

Published online: 08 Oct 2021.

Submit your article to this journal

Article views: 852

View related articles

View Crossmark data
Computational Options for Standard Errors and Test Statistics with Incomplete Normal and Nonnormal Data in SEM

Victoria Savalei* and Yves Rosseel**

*University of British Columbia; **Ghent University

ABSTRACT
This article provides an overview of different computational options for inference following normal theory maximum likelihood (ML) estimation in structural equation modeling (SEM) with incomplete normal and nonnormal data. Complete data are covered as a special case. These computational options include whether the information matrix is observed or expected, whether the observed information matrix is estimated numerically or using an analytic asymptotic approximation, and whether the information matrix and the outer product matrix of the score vector are evaluated at the saturated or at the structured estimates. A variety of different standard errors and robust test statistics become possible by varying these options. We review the asymptotic properties of these computational variations, and we show how to obtain them using lavaan in R. We hope that this article will encourage methodologists to study the impact of the available computational options on the performance of standard errors and test statistics in SEM.

Introduction

The focus of this article is on the methods for drawing inferences from linear structural equation models (SEMs) with incomplete normal and nonnormal data. We cover complete data as a special case. We focus on the normal theory maximum likelihood (ML) estimator, which is the default for continuous data in all SEM software, including the lavaan package in R (R Core Team, 2020; Rosseel, 2012). With incomplete data, the ML estimator as implemented in SEM software is commonly referred to as the full information maximum likelihood (FIML) estimator (Arbuckle, 1996; Schafer & Graham, 2002). Because the FIML estimator is just the ML estimator, in this article we will refer to it as ML.

For incomplete data, we assume that the missing mechanism is either MCAR (missing completely at random) or MAR (missing at random) (Rubin, 1976). MCAR and MAR missing data mechanisms generally result in ignorable missing data (Little & Rubin, 2020), which means that inference based on the observed data log-likelihood remains valid. When data are normally distributed, the ML estimator is consistent with MCAR and MAR missing data, meaning that the ML parameter estimates approach the true values when the model is correct. When data are nonnormal but complete, the ML estimator is consistent as long as the data distribution has finite fourth order moments (Arminger & Schoenberg, 1989; Shapiro, 1983). When data are nonnormal and MCAR, the ML estimator remains consistent (Yuan & Bentler, 2000). The most complicated case is when data are nonnormal and MAR. Despite some sources claiming consistency in the general case of MAR data (Arminger & Sobel, 1990), counterexamples can be constructed to show this is not always the case (Yuan, 2009; Yuan & Savalei, 2014; Q. Zhang et al., 2019). Yuan (2009) showed that the ML estimator is consistent with nonnormal MAR data for several common types of MAR mechanisms. Yuan and Bentler (2010) further showed that consistency is retained for all MAR mechanisms as long as the variables with missing values are linearly related to the observed variables in the model. In this article, we will assume that the ML estimator is consistent with MAR data when the model is correctly specified. When referring to the results that hold for MAR data, we assume either the type of missing mechanism or the model are such that the property of consistency is retained.

Once ML parameter estimates have been obtained, ML standard errors for each estimated parameter and the likelihood ratio (LR) chi-square to assess the overall model fit can be computed. In addition to the ML standard errors, so-called robust or sandwich standard errors can also be computed (Arminger & Schoenberg, 1989; Arminger & Sobel, 1990; Satorra & Bentler, 1994; Yuan & Bentler, 2000; Yuan & Hayashi, 2006), which can protect against nonnormality of the data, model misspecification, or both. In this article, we use the terms robust and sandwich interchangeably. In addition to the LR chi-square, two so-called robust test statistics are popular with nonnormal data: a scaled (mean-corrected) chi-square and an adjusted (mean-and-variance-corrected) chi-square (Asparouhov & Muthén, 2010; Satorra & Bentler, 1994; Yuan & Bentler, 2000). These robust statistics, which are rescalings of the LR chi-square, can protect against distortion of Type I error rates due to nonnormality, and they have also been recommended for use even with normal data when the sample size is small (Maydeu-Olivares, 2017).

The goal of this article is to describe different computational versions of ML and robust standard errors, and of the scaled and adjusted chi-square statistics, with incomplete normal and nonnormal data in SEM. These computational options include whether the information matrix is observed or expected, whether the observed information matrix is obtained numerically or using...
an analytic asymptotic (i.e., increasingly accurate in large samples) approximation, and whether the information matrix and the outer product matrix of the score vector are evaluated at the saturated or at the structured estimates. A variety of different computational versions of standard errors and test statistics can be obtained by varying these options. We list the properties of the resulting standard errors and test statistics under different missing data mechanisms, for normal and nonnormal data, and for correctly specified versus misspecified models.

While ML computations tend to be implemented similarly across software, different software packages have different computational defaults for robust computations, which are often hidden from the user. Further, methodologists who custom-program their own robust computations do not always clearly state which types of estimates were used (e.g., Lai, 2019; Yuan & Bentler, 2000). Yet, these choices may have a substantial effect on the performance of standard errors and test statistics in simulation studies. If the computational choices made in a particular simulation are not reported, knowledge aggregation across studies is difficult. There are very few simulation studies explicitly comparing the impact of the computational options described in this article on performance of standard errors and/or test statistics (Maydeu-Olivares, 2017; Xia et al., 2016), and only one that we know of with incomplete data (Savalei, 2010a). In this article, we illustrate how to obtain all computational options in lavaan 0.6–8 (Rosseel, 2012). This software package has the advantage of allowing a greater variety of types of standard errors and robust test statistics relative to any other SEM package; it is also free and open source. In addition, custom implementation of computational options that are not yet directly available in lavaan is straightforward. In this article, we illustrate how to obtain all computational options for standard errors and test statistics using lavaan.

This article is organized as follows. We first review ML estimation with incomplete data and provide expressions for the relevant quantities based on the log-likelihood: the first and second derivatives of the log-likelihood, and the outer product matrix of the score vector. Next, we review the asymptotic distribution of the ML estimator under different assumptions and describe different computational options for ML and robust standard errors. We then review the asymptotic distribution of the LR test statistic and describe different computational options for the scaling corrections leading to robust statistics. These two sections also include an overview of the computational options available in lavaan, and how to obtain estimates that are not directly available. We conclude with a discussion, which includes an overview of existing limited literature studying the impact of these computational options and a summary of future research directions.

**Derivatives of the log-likelihood**

**Overview of ML estimation**

Let \( Y \) be a \( p \)-dimensional random variable with a continuous multivariate distribution with mean vector \( \mu^0 \), covariance matrix \( \Sigma^0 \), and finite 4th order moments. Under a general mean-and-covariance structure model, \( \mu^0 = \mu(\theta^0) \) and \( \Sigma^0 = \Sigma(\theta^0) \), where \( \theta^0 \) is a \( q \times 1 \) vector of model parameters. Letting \( \theta^i = ((\text{vech}\Sigma^0_i), \mu^0_i) \), i.e., a \( (p^i + p) \times 1 \) vector that contains the non-redundant elements of \( \Sigma^0 \) (Magnus & Neudecker, 1999) and \( \mu^0 \), we can also represent the general mean-and-covariance structure model more succinctly as \( \theta^i = \beta(\theta^0) \). We will denote the hypothesized, or structured, model by \( H_0 \) and we will denote the saturated, or unstructured, model by \( H_1 \). The traditional saturated model of SEM has the parameters \( \theta^0 = \beta^0 \), i.e., it does not impose any restrictions on the means or the covariance matrix.

Let \( y_1, \ldots, y_n \) be the observed values for a random sample of size \( n \) from the distribution of \( Y \). Each \( y_i \) has the dimensions \( p_i \times 1 \), where \( 1 \leq p_i \leq p \) for \( i = 1, \ldots, n \). The dimensions of the individual observations vary due to missing data. We estimate the model parameters by maximum likelihood under the assumption of a multivariate normal distribution for \( Y \). The observed log-likelihood for an observation \( y_i \) is given by

\[
l_i(\theta) = \frac{1}{2} p_i \log (2\pi) - \frac{1}{2} \log |\Sigma_i(\theta)| - \frac{1}{2} (y_i - \mu_i(\theta)) \Sigma_i(\theta)^{-1} (y_i - \mu_i(\theta)),
\]

where \( \mu_i \) is a \( p_i \times 1 \) sub-vector of \( \mu \), omitting elements corresponding to variables that are missing from \( y_i \), and \( \Sigma_i \) is the \( p_i \times p_i \) sub-matrix of \( \Sigma \), omitting rows and columns corresponding to the variables missing from \( y_i \). To construct these matrices more formally, let \( \tau_i = \frac{\partial \Sigma_i}{\partial \theta_i} \), a \( 0 \times 1 \) matrix of size \( p_i \times p \) (Yuan & Bentler, 2000); then, \( \mu_i = \tau_i \mu \) and \( \Sigma_i = \tau_i \Sigma \tau_i \).

The observed log-likelihood for the entire sample is obtained by summing over the individual log-likelihoods:

\[
l(\theta) = \sum_{i=1}^{n} l_i(\theta).
\]

We denote the structured model parameter estimates by \( \hat{\theta} \) and the maximized log-likelihood by \( l(\hat{\theta}) \). We will write \( \hat{\mu} = \mu(\hat{\theta}) \), \( \hat{\Sigma} = \Sigma(\hat{\theta}) \), as well as \( \hat{\beta} = \beta(\hat{\theta}) \). We denote the saturated model parameter estimates by \( \hat{\theta} \) and the maximized log-likelihood by \( l(\hat{\beta}) \). We will also write \( \hat{\mu} = \mu(\hat{\beta}) \) and \( \hat{\Sigma} = \Sigma(\hat{\beta}) \). We assume that the missing data mechanism is such that \( \hat{\beta} \) is consistent for \( \beta^0 \); it follows that, when \( H_0 \) is true, \( \hat{\theta} \) is consistent for \( \theta^0 \) (Shapiro, 1985). When \( H_0 \) is false, we will denote the limiting value of \( \hat{\theta} \) by \( \theta^* \), which we will refer to as pseudo-parameters, i.e., the population values that the ML method consistently estimates when the model is wrong (Armitage & Schoenberg, 1989; Armitage & Sobel, 1990; Yuan & Hayashi, 2006).\(^1\) We will write \( \mu^* = \mu(\theta^*) \), \( \Sigma^* = \Sigma(\theta^*) \), and \( \beta^* = \beta(\theta^*) \). When \( H_0 \) is false, \( \beta^0 \neq \beta^* \).

\(^1\) It should be noted that \( \theta^* \) will be different depending on whether the data are complete or incomplete, and for different types of incomplete data; see X. Zhang and Savalei (2020). Thus, we cannot say that with incomplete data, the ML estimator retains consistency for the complete data pseudo-parameters when the model is wrong.
The first derivative of the log-likelihood

**Saturated model \((H_1)\)**

We denote the first derivative, or the score vector, of Equation 1 with respect to \( \beta = (\{\text{vech} \Sigma\}' , \mu)' \) by

\[
d'_{\beta,i}(\beta) = \frac{\partial l_i(\beta)}{\partial \beta'.}
\]

An explicit expression is given in Appendix (part B). The first derivative of the log-likelihood for the sample is the sum of the individual derivatives. It is more useful to define the average derivative and its limiting expression as the sample size goes to infinity:

\[
d_{\beta,n} = \frac{1}{n} \sum_{i=1}^{n} d_{\beta,i}
\]

\[
d_\beta = \lim_{n \to \infty} d_{\beta,n}
\]

Note that \(d_{\beta,n} = d_{\beta,n}(\hat{\beta}) = 0\) by construction, and \(d_{\beta} = d_\beta(\beta^0) = 0\).

**Structured model \((H_0)\)**

By the chain rule, the first derivative of Equation 1 with respect to \( \theta \) is given by

\[
d'_{\theta,i}(\theta) = \frac{\partial l_i(\theta)}{\partial \theta'} = \frac{\partial l_i(\beta)}{\partial \beta'} \frac{\partial \beta(\theta)}{\partial \theta'} = d'_{\beta,i}(\beta) \Delta
\]

where \( \Delta = \Delta(\theta) = \frac{\partial \beta(\theta)}{\partial \theta} \) is a \((p' + p) \times q\) matrix of model derivatives. As well, \(d'_{\theta,n} = d_{\theta,n} \Delta\) and \(d'_{\theta} = d_{\theta} \Delta\). Note that \(d'_{\theta,n} = d_{\theta,n} \Delta = 0\) by construction, where \( \Delta = \Delta(\hat{\theta}) \). Under \(H_0\), \(d_{\theta} = d_{\theta}(\theta^0) = 0\). However, when \(H_0\) is false, it is also the case that \(d_{\theta} = d_{\theta}(\theta^0) = 0\); the pseudo-parameters \(\theta^*\) are the values that make the first derivative vanish in the limit.

The second derivative of the log-likelihood

**Saturated model \((H_1)\)**

The negative of the second derivative (or the Hessian) of the individual log-likelihood in Equation 1 with respect to \( \beta \) is

\[
A_{\beta,i}(\beta) = -\frac{\partial^2 l_i(\beta)}{\partial \beta \partial \beta'}
\]

An explicit expression is given in Appendix (part A). The sample average and its limiting expression are as follows:

\[
A_{\beta,n} = -\frac{1}{n} \sum_{i=1}^{n} A_{\beta,i}
\]

\[
A_{\beta} = \lim_{n \to \infty} A_{\beta,n}
\]

where \(j = 1, \ldots, J\) enumerates the missing data patterns, \(q_j\) is the probability of pattern \(j\) in the population, \(A_j = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} A_{\beta,i}\) and \(n_j\) is the number of observations in the \(j\)th pattern. The matrices \(A_{\beta,n}\) and \(A_{\beta}\) can be described as the observed and the expected information matrices for the saturated model. In particular, \(A_{j}\) can be thought of as the conditional expected value of the second derivative given pattern \(j\).

Under a general MAR mechanism, observations within each pattern are not randomly sampled from the population, and the limiting values of pattern-specific means and covariance matrices are not subcomponents of the overall population vector of means \(\mu^0\) and the population covariance matrix \(\Sigma^0\). Denote the limiting values of these pattern-specific means and covariance matrices by \(\zeta_j\) and \(\Pi_j\), respectively, \(j = 1, \ldots, J\). An explicit expression for \(A_j\) is given by:

\[
A_j(\beta|\zeta_j, \Pi_j) = \left( \kappa_j^j(\Sigma_j^{-1} \otimes (\Sigma_j^{-1} \Pi_j \Sigma_j^{-1} + h_j h_j' - 5 \Sigma_j^{-1})) \kappa_{\beta_j}^j (h_j \otimes \Sigma_j^{-1} \tau_j) \right)
\]

where \(\tau_j\) and \(k_j\) is \(\tau_j \otimes \tau_j)D_p\) add rows and columns of zeros corresponding to missing variables in pattern \(j\), \(D_p\) is the duplication matrix of order \(p\) (Magnus & Neudecker, 1999), \(\mu_j = \tau_j \mu\), \(\Sigma_j = \tau_j \Sigma \tau_j'\), \(h_j = \Sigma_j^{-1}(\zeta_j - \mu_j)\). The selection matrices \(\tau_j\) and \(\kappa_j\) ensure the dimensions of \(A_j\) are \(p^* \times p^*\) for all \(j\). In this expression, \(A_j\) is a function of \(\beta\) through \(\Sigma_j\) and \(\mu_j\). Because the pattern-specific population means and covariance matrices \(\zeta_j\) and \(\Pi_j\) are not known under a general MAR mechanism, in the sample we can estimate \(A_{j,n} = A_j(\hat{\beta}|\hat{y}_j, S_j)\) for each \(j = 1, \ldots, J\), where \(\hat{y}_j\) and \(S_j\) are the sample mean and the sample covariance matrix in pattern \(j\). These lead to the so-called observed information estimates, because \(A_{\beta,n} = \frac{1}{J} \sum_{j=1}^{J} A_{j,n}\). We will use the following two observed information estimates of \(A_{\beta}\):

\[
A_{\beta} = A_{\beta,n} = \frac{1}{J} \sum_{j=1}^{J} q_j A_{j,n}(\hat{\beta}|\hat{y}_j, S_j)
\]

\[
A_{\beta} = A_{\beta,n} = \frac{1}{J} \sum_{j=1}^{J} q_j A_{j,n}(\hat{\beta}|\hat{y}_j, S_j),
\]

where the second estimate assumes a structured model has also been fit to data.

In the special case of MCAR data, observations falling within each pattern can be considered a random sample from the population, and the pattern-specific population means and covariances are subsets of the general vector of means and covariance matrix: \(\zeta_j = \mu_j^0\) and \(\Pi_j = \Sigma_j^0\) for \(j = 1, \ldots, J\). This consequence of MCAR is known as the homogeneity of means and covariance matrices (Kim & Bentler, 2002).\(^3\) Note that

\(^2\)To simplify notation, we assume that \(J\) enumerates all possible patterns that exist in the population (i.e., we implicitly set \(n_j = 0\) for those patterns missing from a particular sample).

\(^3\)The converse is not true, however: it is possible to construct examples where homogeneity of means and covariance matrices holds while the mechanism is not MCAR (Yuan et al., 2018).
Then, the general form of $A_\beta$ in Equation 4 under MCAR is given by (Yuan & Bentler, 2000):

$$A_\beta = \frac{1}{J} \sum_{j=1}^{J} q_j W_j = W,$$

and two additional estimates are possible:

$$\hat{A}_{\beta, E} = W(\hat{\beta}) = \hat{W}$$

$$\hat{A}_{\beta, E} = W(\hat{\beta}) = W$$

Because these estimates replace $\bar{y}_j$ and $S_j$ with their estimated expected values under a particular model, they are known as expected information estimates. They also necessarily assume the model under which the expectation is taken is true. Because complete data can be viewed as a special case of MCAR data, these estimates are valid for complete data as well. Table 1 provides parallel expressions to Equations 5 and 8 in the special case of complete data. Of note, observed and expected information estimates are actually the same with complete data when evaluated at the saturated model estimates: $\hat{A}_{\beta, E} = \hat{A}_\beta = W_c$, where $W_c$ is the complete data simplification of $W$.

**Structured model ($H_0$)**

The second derivative of Equation 1 with respect to $\theta$ can be obtained from $A_{\beta,i}$ via the multivariate version of the chain rule for the second derivative (e.g., Magnus & Neudecker, 1999, p. 110, Theorem 9)4

$$A_{\theta,i}(\theta) = -\frac{\partial^2 l_i(\theta)}{\partial \theta \partial \theta^T} = \Delta_{A_{\beta,i} \Delta} - \left( d_{\beta,i} \otimes I_q \right) H,$$

where $I_q$ is a $q \times q$ identity matrix and $H = \frac{\partial \text{vec}(A)}{\partial \theta^T}$ is the $q(p^* + p) \times q$ matrix of second derivatives of $\beta(\theta)$.5 We define the sample average and its limiting expression as follows:

$$A_{\theta,n} = \frac{1}{n} \sum_{i=1}^{n} A_{\theta,i}$$

$$A_\theta = \lim_{n \rightarrow \infty} A_{\theta,n} = \Delta_{A_{\beta} \Delta} - \left( d_{\beta} \otimes I_q \right) H$$

When $H_0$ is true, $d_{\beta} = 0$, and therefore:

$$A_0^0 = \Delta_{A_{\beta}^0 \Delta^0}.$$

The following two estimates can be constructed based on the estimate $A_\beta$ from Equation 7:

$$\hat{A}_\theta = \hat{\Delta}_{A_{\beta} \Delta} - \left( \hat{d}_{\beta} \otimes I_q \right) \hat{H}$$

(15)

$$\hat{A}_{\theta,h1} = \hat{\Delta}_{A_{\beta} \Delta}$$

(16)

where $\hat{H} = H(\hat{\theta})$.6 The estimate in Equation 15 is typically obtained via numeric differentiation of the analytic first derivative of the log-likelihood (i.e., $\hat{d}_{\beta,n}$), because $\hat{H}$ is too cumbersome to obtain analytically; for this reason, $\hat{A}_\theta$ is commonly referred to as the numeric Hessian. The second term in Equation 15 will tend to be small, and Equation 16 constitutes a valid asymptotic approximation when $H_0$ is true, by Equation 14. An additional estimate can be constructed based on the estimate $\hat{A}_\theta$ from Equation 6:

$$\hat{A}_\theta = \hat{\Delta}_{A_{\beta} \Delta} - \left( \hat{d}_{\beta} \otimes I_q \right) \hat{H} = \hat{\Delta}_{A_{\beta} \Delta}$$

(17)

where the second term drops out because $\hat{d}_{\beta} = 0$. All three estimates in Equations 15–17 can be considered observed information estimates, although most commonly this term refers to the estimate in Equation 15.

In the special case of MCAR data, two additional estimates based on expected information are also possible:

$$\hat{A}_{\theta, E} = \hat{\Delta}_{\hat{W} \hat{\Delta}}$$

(18)

$$\hat{A}_{\theta, E} = \hat{\Delta}_{\hat{W} \hat{\Delta}}$$

(19)

based on the Equations 11 and 10. With complete data, equations 17 and 19 yield identical estimates, i.e., $\hat{A}_\theta = \hat{A}_{\theta, E}$. Other than this equivalence, complete data versions of these equations do not provide any additional insight and are omitted (see Browne & Arminger, 1995; Lee, 2007; Yuan & Hayashi, 2006).

**Outer product matrix of the score vector**

**Saturated model ($H_1$)**

The outer product of the score vector in Equation 2 with itself will be denoted by

$$B_{\beta,i}(\beta) = \frac{\partial l_i(\beta)}{\partial \beta^T} \frac{\partial l_i(\beta)}{\partial \beta} = d_{\beta,i} \otimes d_{\beta,i}.$$

The average outer product and its limiting expression are given by:

$$B_{\beta,n} = \frac{1}{n} \sum_{i=1}^{n} B_{\beta,i} = \frac{1}{J} \sum_{j=1}^{J} \frac{n_j}{n} W_j I_{1,n} W_j$$

(20)

---

4Yuan and Bentler (2000) and Savalei and Rosseel (2010a) both inaccurately stated a simplified version of this equation that omits the second term, but this simplification only holds asymptotically and when $H_0$ is true; see Equation 13.

5The matrix $H = H(\theta)$ is formed by stacking vertically $(p^* + p)$ number of $q \times q$ matrices of the form: $H_i = \frac{\partial^2 l_i(\theta)}{\partial \beta \partial \beta^T} = \frac{\partial^2 l_i(\theta)}{\partial \beta \partial \beta^T}$, where $\delta_i$ is the transpose of the rth row of $\Delta$, $r = 1, \ldots, (p^* + p)$. The following identity also holds: $(d_{\beta} \otimes I_q) H = \sum_{r=1}^{p^*+p} \frac{\partial^2 l_i(\theta)}{\partial \beta \partial \beta^T}$. The subscript “$h1$” in Equation 16 corresponds to the syntax used to call this estimate in lavaan. It is meant to capture that the difference between the two structured estimates of observed information, $\hat{A}_\theta$ and $\hat{A}_{\theta,h1}$, is that the latter only uses the first term based on the information matrix for the $H_1$ model.
Note that Equation 26 does use the $\hat{\theta}$ to estimate $\hat{\Delta}$, so the estimate $\hat{B}_{\theta}$ is a kind of mixed estimate.

In the special case of complete data,

$$B_{\theta} = \Delta W_{\hat{\theta}} \Gamma W_{\hat{\theta}},$$

(27)

where $\Gamma$ is defined in Table 1. Based on this expression, several additional estimates of $B_{\theta}$ that mix structured and saturated estimates can be constructed. Two such estimates are as follows:

$$B_{\theta} = \hat{\Delta} W_{\hat{\theta}} \hat{\Gamma} W_{\hat{\theta}},$$

and

$$\hat{B}_{\theta} = \Delta \hat{\Delta} W_{\hat{\theta}} \Gamma \hat{\Delta} W_{\hat{\theta}} \hat{\Delta},$$

(28)

where $\hat{\Gamma}$ is computed using Equation 24. The estimate $\hat{B}_{\theta}$ makes use of the fact that with complete data $W_{\hat{\theta}}^0 = A_{\hat{\theta}}^0$, and observed information estimate $\hat{\Delta}$ is a consistent estimate of $A_{\hat{\theta}}^0$. When the mean structure is saturated (i.e., $\mu = \hat{\mu}$), it can be shown that $\hat{B}_{\theta} = \hat{B}_{\theta}$ (see Appendix, part D, also Yuan & Hayashi, 2006).

**Standard errors for $\hat{\theta}$**

**Asymptotic distributions of estimators**

**Saturated model ($H_1$)**

Under the assumption of multivariate normality of the data, and assuming the missing mechanism is such that $\hat{\theta}$ is consistent for $\theta^0$, the asymptotic distribution of $\hat{\theta}$ is given by (e.g., Little & Rubin, 2020):

$$\sqrt{n}(\hat{\theta} - \theta^0) \rightarrow N(0, (A_{\theta}^0)^{-1}),$$

(29)

where $A_{\theta}^0 = A_{\theta}(\theta^0)$ is defined in Equation 4. With MCAR data, $A_{\theta}^0 = W_0^0 = W(\theta^0)$ where $W$ is defined in Equation 9. With complete data, $\hat{\theta} = (vechS')$, and $A_{\theta}^0 = W_0^0$, where $W_0$ is defined in Table 1.

When multivariate normality cannot be assumed, the asymptotic distribution of $\hat{\theta}$ is given by (Armingr & Schoenberg, 1989; Arminger & Sobel, 1990; White, 1982; Yuan, 2009; Yuan & Bentler, 2000):

$$\sqrt{n}(\hat{\theta} - \theta^0) \rightarrow N(0, \Omega^0) = N(0, (A_{\theta}^0)^{-1} B_{\theta,\theta} (A_{\theta}^0)^{-1}),$$

(30)

where $B_{\theta,\theta} = B_{\theta}(\theta^0)$ is defined in Equation 21. The covariance matrix in Equation 30 is a type of sandwich covariance matrix, where the bread is the inverse of the information matrix and the meat is the outer product matrix of the score vector. It follows from Equations 29 and 30 that when data are multivariate normal, $B_{\theta}^0 = A_{\theta}^0$. For MCAR data, this result can be shown algebraically (see Appendix, part C). With complete data, $\Omega^0 = (W_0^0)^{-1} B_{\theta}^0 (W_0^0)^{-1}$, where $\Gamma$ is defined in Table 1.

**Structured model ($H_0$)**

When $H_0$ is true and data are multivariate normal, the asymptotic distribution of $\hat{\theta}$ is given by:

$$\sqrt{n}(\hat{\theta} - \theta_0) \rightarrow N(0, (A_{\theta}^0)^{-1}),$$

(31)
where $A_0^2$ is defined in Equation 14. It is also the case that $A_0^2 = B_0^2$.

When $H_0$ is true but normality cannot be assumed, the asymptotic distribution of $\hat{\theta}$ is given by (Yuan & Bentler, 2000):

$$\sqrt{n} (\hat{\theta} - B_0^2) \to N \left(0, (A_0^2)^{-1} B_0^2 (A_0^2)^{-1} \right).$$  \hfill (32)

When $H_0$ is false, the asymptotic distribution of $\hat{\theta}$ is given by:

$$\sqrt{n} (\hat{\theta} - B_0^2) \to N \left(0, (A_0^2)^{-1} B_0^2 (A_0^2)^{-1} \right)$$  \hfill (33)

where $A_0^2 = A_0 (\theta')$ is defined in Equation 13 and $B_0^2 = B_0 (\theta')$ is defined in Equation 21. No simplification of the sandwich covariance matrix in Equation 33 occurs even for normal data because $A_0^2 \neq B_0^2$ in general.

The expressions for the asymptotic covariance matrix of $\hat{\theta}$ are summarized in Table 2, which also includes the special case of complete data. Of particular note, when data are complete and $H_0$ is true, the asymptotic covariance matrix in Equation 32 is commonly written as (Satorra & Bentler, 1994):

$$(\Delta \hat{\theta}^2 W^2 \partial^2 \Delta) - \Delta \hat{\theta}^2 W^2 \Pi^2 W^2 \Delta (\Delta \hat{\theta}^2 W^2 \Delta)^{-1},$$  \hfill (34)

which follows by Equations 9, 14, and 27, and the famous Satorra-Bentler standard errors are estimated from this equation.\footnote{The covariance matrix in Equation 32 can also be expressed as $(\Delta \hat{\theta}^2 W^2 \partial^2 \Delta) - \Delta \hat{\theta}^2 W^2 \Pi^2 W^2 \Delta (\Delta \hat{\theta}^2 W^2 \Delta)^{-1}$, which parallels Equation 34. However, this expression is not used with incomplete data because there is no direct way to estimate $\hat{\Omega}^2$. With complete data, $\hat{\Omega}^2 = \Gamma^2$, and an estimator of $\hat{\rho}^2$ is available (see Equation 24).}

### Estimates of standard errors

Table 3 provides a summary of sample estimates of the asymptotic covariance matrix of $\sqrt{n} \hat{\theta}$ in Table 2 that remain consistent under various conditions. Standard errors for $\hat{\theta}$ are obtained by taking the square-root of the diagonal elements of the estimated asymptotic covariance matrix and dividing by the square-root of the sample size.

When the data are normal and $H_0$ is true (first column of Table 2), there are five different types of standard errors that remain consistent with MAR data, three based on estimates of the information matrix $A_0^2$ (Equations 15–17) and two based on estimates of the outer product matrix $B_0^2$ (Equations 25–26). Estimates based on the outer product matrix are rarely used in practice. With incomplete data, most software packages implement standard errors based on the numeric Hessian by default, which is equivalent to $A_\theta$. However, the approximation $A_{\theta \text{ML}}$ is a viable analytic alternative that is also based on structured estimates. The saturated estimate $A_\theta$ is also an interesting alternative. It should be noted that while $A_\theta$ is always consistent for $A_0^2$, it is not actually consistent for the asymptotic covariance matrix of $\hat{\theta}$ when $H_0$ is false (given by Equation 33). Thus, simply using observed information does not protect against model misspecification, and there is no theoretical advantage of using $A_\theta$ over $A_{\theta \text{ML}}$ or $A_\theta$.

Table 3 also lists two additional types of standard errors that are consistent for MCAR or complete data only (Equations 18–19) as they are based on expected information. It also lists two mixed estimates (Equation 28), which are only used with complete data. When data are complete, most software packages default to standard errors based on a structured expected information estimate, i.e., those based on $A_{\theta \text{E}}$ in Equation 18.

When the data are not normal but $H_0$ is true, many different sandwich-type standard error estimates can be obtained by crossing different computational versions of the “bread” estimate with those of the “meat” estimate (see the second column of Table 3). The properties of the resulting standard errors are inherited from these components; for example, if a component of the sandwich assumes MCAR data, then the resulting standard errors are only consistent under MCAR data. The most likely combinations to be used in practice are those that use the same types of estimates (e.g., both structured or both saturated), though there is no theoretical reason why mixed estimates should not be studied or used. The combination $A_{\theta \text{E}} B_0 A_{\theta \text{E}}$ yields the classic Satorra-Bentler standard errors based on estimating Equation 34. These standard errors are based on $\Gamma$ (via $B_0$, see Equation 28) and the estimate $\hat{W}_C$.

When $H_0$ is false, only one sandwich combination provides consistent estimates of the variability of $\hat{\theta}$ (see the last row of Table 3), regardless of whether the data are normal. This is because only $\hat{A}_\theta$ is consistent for $A_0^2$ and only $B_0$ is consistent.

### Table 2. Asymptotic covariance matrix of $\sqrt{n} \hat{\theta}$.

<table>
<thead>
<tr>
<th>$H_0$ is true</th>
<th>Normal Data</th>
<th>Nonnormal Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0^2 = B_0^2$</td>
<td>$A_0^2$ and $B_0^2$</td>
<td>$A_0^2$ and $B_0^2$</td>
</tr>
</tbody>
</table>

Note: $A_0^2 = B_0^2$.

<table>
<thead>
<tr>
<th>$H_0$ is false</th>
<th>Normal Data</th>
<th>Nonnormal Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0^2 = B_0^2$</td>
<td>$A_0^2$ and $B_0^2$</td>
<td>$A_0^2$ and $B_0^2$</td>
</tr>
</tbody>
</table>

Complete Data Special Case:

<table>
<thead>
<tr>
<th>$H_0$ is true</th>
<th>Normal Data</th>
<th>Nonnormal Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0^2 = B_0^2$</td>
<td>$A_0^2$, $B_0^2$, and $A_0^2 = B_0^2$</td>
<td>$A_0^2$, $B_0^2$, and $A_0^2 = B_0^2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$H_0$ is false</th>
<th>Normal Data</th>
<th>Nonnormal Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0^2 = B_0^2$</td>
<td>$A_0^2$, $B_0^2$, and $A_0^2 = B_0^2$</td>
<td>$A_0^2$, $B_0^2$, and $A_0^2 = B_0^2$</td>
</tr>
</tbody>
</table>

### Table 3. Some consistent estimates of the asymptotic covariance matrix of $\sqrt{n} \hat{\theta}$.

<table>
<thead>
<tr>
<th>$H_0$ is true</th>
<th>Normal Data</th>
<th>Nonnormal Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0^2$ (see Eq. 15)</td>
<td>$A_0^2$ and $B_0^2$</td>
<td>$A_0^2$ and $B_0^2$</td>
</tr>
</tbody>
</table>

| MAR, MCAR, | $A_0^2$ and $B_0^2$ | $A_0^2$ and $B_0^2$ |
| or complete data | $A_0^2$ and $B_0^2$ | $A_0^2$ and $B_0^2$ |

| $H_0$ is true; MAR or complete data only | $A_0^2$ and $B_0^2$ | $A_0^2$ and $B_0^2$ |

Note: Other mixed estimates are possible when $H_0$ is true and data are complete, such as $(\Delta \hat{\theta}^2 W^2 \Delta)^{-1}$ (see complete data and $(\Delta \hat{\theta}^2 W^2 \Delta)^{-1}$). Because such mixed estimates are somewhat odd, the table highlights only those that are used in software. With complete data and when the mean structure is saturated, $B_0 = B_0$ (see part C of the Appendix), and thus the combination $A_0^2 = B_0 A_0^2$ would also provide consistent estimates regardless of whether $H_0$ is true or false in this case.

<table>
<thead>
<tr>
<th>$H_0$ may be false</th>
<th>Normal Data</th>
<th>Nonnormal Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0^2$ and $B_0^2$</td>
<td>$A_0^2$ and $B_0^2$</td>
<td>$A_0^2$ and $B_0^2$</td>
</tr>
</tbody>
</table>
for $B_0$, in general. The resulting standard errors are often called Huber-White standard errors (Eicker, 1967; Huber, 1967; White, 1980). In the SEM literature, the default meaning of the term “sandwich standard errors” usually refers to these estimates. When the mean structure is saturated and the data are complete, $\hat{B}_0 = \bar{B}_0$ (see Appendix, part D), and using $\bar{B}_0$ as the meat would produce standard errors identical to Huber-White standard errors.

**Lavaan options for standard errors**

In lavaan 0.6–8, the default estimator is estima\texttt{tor = “ML”}, and the default treatment of missing data is listwise deletion. To appropriately treat missing data while performing ML estimation (i.e., to do FIML estimation), the user must add missing = “ML” to the syntax.\(^8\) The information options in lavaan are toggled via the following options: information, observed\texttt{.information}, and h1\texttt{.information}. Table 4 provides lavaan syntax for obtaining most of the standard error estimates in the first column of Table 3. These standard error estimates are available via se = “standard” (default with estimator = “ML”). To obtain estimates based on expected information, we set information = “expected”, and further specify whether the expected information matrix should be based on structured (h1\texttt{.information = “structured”}) or saturated estimates (h1\texttt{.information = “unstructured”}). To obtain estimates based on observed information, we set information = “observed”, which defaults to a secondary setting of observed\texttt{.information = “hessian”} to produce standard errors based on the numeric Hessian.\(^9\) We can change this setting to observed\texttt{.information = “h1”} to obtain the analytic estimates of observed information based on either structured or saturated estimates. Finally, we can set information = “first.order” to obtain standard error estimates based on the outer product matrix, which can also be structured (h1\texttt{.information = “structured”}) or unstructured (h1\texttt{.information = “unstructured”}).

Table 5 provides syntax for obtaining sandwich standard errors, which produce most of the options listed in the second column of Table 3, as well as the Huber-White standard errors shown in the last row of Table 3. For historic reasons, there are two types of standard error commands available in lavaan to obtain sandwich standard errors. The first is se = “robust\texttt{.huber\texttt{.white}}”\(^10\) (default with estimator = “MLR”, discussed shortly), which can be used with or without missing = “ML”. The second is se = “robust\texttt{.sem}” (default with estimator = “MLM”), which can only be used with complete data. The default computation of sandwich standard errors when se = “robust\texttt{.huber\texttt{.white}}” is to produce the Huber-White standard errors, i.e., shown in the last row of Table 3. As Table 5 shows, most other combinations in the second column of Table 3 can be obtained by toggling the same lavaan options as with normal data, plus an additional option, h1\texttt{.information\texttt{.meat}}, which permits mixed estimates by allowing the meat of the sandwich (i.e., the outer product matrix) to use different types of estimates (structured vs. unstructured) than what is specified for the bread on the sandwich in h1\texttt{.information}.

When se = “robust\texttt{.sem}”, the standard errors are computed by estimating Equation 34. By default, this command produces classic Satorra-Bentler standard errors, which use $\hat{\Gamma}$ and $\hat{W}_E$. Toggling options does not impact the estimate of $\hat{\Gamma}$, which is always $\hat{\Gamma}$, but only affects how $\hat{W}_E$ is estimated, leading to expressions that can be equivalently written in sandwich form with the mixed estimates $\hat{B}_0$ or $\hat{B}_0$ (see Equation 28 or Table 3) as the meat of the sandwich. Only some combinations of these mixed meat estimates with various information matrix (bread) estimates are possible in lavaan, because lavaan generally uses the same estimate of $\hat{W}_E$ in all parts of Equation 34.\(^11\)

as Table 5 shows, se = “robust\texttt{.huber\texttt{.white}}” and se = “robust\texttt{.sem}” will produce identical standard error estimates with complete data when information = “expected”, h1\texttt{.information = “unstructured”}. In this case, using saturated estimates $\hat{W}_S$ and $\hat{\Gamma}$ in Equation 34 is equivalent to estimating the sandwich using saturated estimates of both the bread and the meat. In addition, because $\hat{B}_0 = \hat{B}_0$ when the mean structure is saturated, se = “robust\texttt{.huber\texttt{.white}}” and se = “robust\texttt{.sem}” will produce identical

---

\(^8\)Aliases include missing = ”FIML” and missing = ”direct”.

\(^9\)When observed\texttt{.information = “hessian”}, lavaan currently does not respond to the h1\texttt{.information} option if it is also included; the numerically obtained Hessian is always structured.

\(^10\)se = “sandwich” is an alias.

\(^11\)An exception is when information = “observed”, observed\texttt{.information = “hessian”}. In this case, lavaan uses the numeric Hessian to estimate $\hat{\Delta}_1 \hat{W}_E \hat{\Delta}_0$ in Equation 34. This is lavaan’s best interpretation of what the user wants when observed\texttt{.information = “hessian”} is used alongside se = “robust\texttt{.sem}”.

---

**Table 4. Available options for ML standard errors in lavaan.**

<table>
<thead>
<tr>
<th>lavaan options with se = “standard”</th>
<th>Asymptotic Covariance Matrix Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>information = “expected”, h1\texttt{.information = “structured”}</td>
<td>$\hat{\Delta}_1 \hat{\Delta}_0$ (default without missing = “ML”)</td>
</tr>
<tr>
<td>information = “expected”, h1\texttt{.information = “unstructured”}</td>
<td>$\hat{\Delta}_1 \hat{\Delta}_0$ (equal to $\hat{\Delta}_1 \hat{\Delta}_0$ with complete data)</td>
</tr>
<tr>
<td>information = “observed”, observed\texttt{.information = “hessian”, h1\texttt{.information = “structured”}}</td>
<td>$\hat{\Delta}_1^{-1}$ (default with missing = “ML”)</td>
</tr>
<tr>
<td>information = “observed”, observed\texttt{.information = “h1”, h1\texttt{.information = “structured”}}</td>
<td>$\hat{\Delta}_1^{-1}$ (equal to $\hat{\Delta}_1^{-1}$ with complete data)</td>
</tr>
<tr>
<td>information = “first.order”, h1\texttt{.information = “structured”}</td>
<td>$\hat{B}_0^{-1}$</td>
</tr>
<tr>
<td>information = “first.order”, h1\texttt{.information = “unstructured”}</td>
<td>$\hat{B}_0^{-1}$</td>
</tr>
</tbody>
</table>

Current lavaan defaults with missing = “ML” are in italics. All estimates are consistent only when $H_0$ is true. Options based on $\hat{B}_0$ and $\hat{B}_0$ are unavailable.
Table 5. Available “bread” and “meat” options for sandwich-type standard errors in lavaan.

<table>
<thead>
<tr>
<th>( \hat{A}_{B1} )</th>
<th>( \hat{A}_{B2} )</th>
<th>( \hat{A}_{B3} )</th>
<th>( \hat{A}_{B4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>se = “robust.huber.white”, information = “expected”, h1.information = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
</tr>
<tr>
<td>se = “robust.huber.white”, information = “expected”, h1.information = “unstructured”, h1.information.meat = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
</tr>
<tr>
<td>se = “robust.huber.white”, information = “expected”, h1.information = “unstructured”, h1.information.meat = “unstructured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
<td>se = “robust.huber.white”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
</tr>
</tbody>
</table>

Complete data only:

<table>
<thead>
<tr>
<th>( \hat{A}_{B1} )</th>
<th>( \hat{A}_{B2} )</th>
<th>( \hat{A}_{B3} )</th>
<th>( \hat{A}_{B4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>se = “robust.sem”, information = “expected”, h1.information = “structured”</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
</tr>
<tr>
<td>Not available</td>
<td>Not available</td>
<td>se = “robust.sem”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
<td>se = “robust.sem”, information = “observed”, observed.information = “h1”, h1.information = “structured”</td>
</tr>
</tbody>
</table>

Sandwich-type standard errors are obtained from the diagonals of the asymptotic covariance matrix of the form \( A^{-1}BA^{-1} \), where the options for \( A^{-1} \) are given in columns and options for \( B \) are given in rows. Current lavaan defaults are in italics. With complete data, the second and third columns will produce identical estimates (\( \hat{A}_{B2} = \hat{A}_{B3} \)). Specifying \textit{information} = “observed”, \textit{observed.information} = “hessian”, h1. \textit{information} = “unstructured” is another way to obtain \( \hat{A}_{B1} = \hat{B}_{B1} = \hat{A}_{B2} \).
standard error estimates when \texttt{information} = “expected”, \texttt{h1.information} = “structured” but only for models with no mean structure (see Appendix, part D).

Rather than manually specifying desired standard error options to accompany ML estimates, most end users opt for lavaan’s prepackaged options to address nonnormality or model misspecification, available via \texttt{estimator} = “MLM” (complete data only) and \texttt{estimator} = “MLR” (complete or incomplete data). Table 10 summarizes the defaults for these choices. Because these names are also used in the popular commercial package Mplus (Muthen & Muthen, 2012), but with different defaults, the options for \texttt{estimator} = “MLM” to match Mplus are also listed in the table. When \texttt{estimator} = “MLR”, lavaan and Mplus use the same defaults. We note that the specification of these prepackaged options under \texttt{estimator} is somewhat misleading, because the estimator remains the same (ML) but rather the specifications MLM and MLR trigger the computations of robust standard errors and test statistics. The default computations under MLM and MLR can be modified in the same manner (i.e., using \texttt{information}, \texttt{observed.information}, and \texttt{h1.information} options).

**Scaling corrections to the test statistic**

**Asymptotic distribution of the test statistics**

The likelihood ratio (LR) test statistic $T$ and the two most popular robust test statistics, the scaled (or mean-corrected) chi-square $T_M$ and the adjusted (or mean-and-variance corrected) chi-square $T_{MV}$, are given by:

$$ T = 2(\ln(\hat{\theta}) - \ln(\tilde{\theta})) $$

$$ T_M = \frac{1}{c}T; T_{MV} = \frac{1}{a}T + b, $$

where $b = d(1 - c/a)$ and $d = p^* + p - q$ is the degrees of freedom for the model. In these equations, $c$ is an estimate of

$$ c_0 = \text{Tr}(U^0\Omega^0) / d, $$

$a$ is an estimate of

$$ a_0 = \sqrt{\text{Tr}((U^0\Omega^0)^2)} / d, $$

where $\Omega^0$ is defined in Equation 30 and

$$ U^0 = A^0 - A^0\Delta^0(\Delta^0A^0)^{-1}\Delta^0A^0. $$

The $(p^* + p) \times (p^* + p)$ matrix $U^0$ is called the residual weight matrix, and it has reduced rank $d$. \footnote{\(U^0 = A^0(I - P^0), \) where $I - P^0$ is an oblique projection matrix onto the column space of $(\Delta^0)^\top$; the orthogonal complement of $\Delta^0$, which has rank $d$. The meaning of $U^0$ is that its the inverse of the asymptotic covariance matrix of the model residuals, i.e., of $\sqrt{n}(\hat{\theta} - \tilde{\theta})$.} The constants $c_0$ and $a_0$ are commonly referred to as scaling factors or scaling corrections.

The statistic $T_M$ is the most popular type of robust test statistic for nonnormal data, also known as the Satorra-Bentler chi-square, or the Yuan-Bentler chi-square when extended to incomplete data (Satorra & Bentler, 1994; Yuan & Bentler, 2000). The version of $T_{MV}$ in Equation 36 was proposed by Asparouhov and Muthén (2010); it is an improvement on the older version of the mean-and-variance corrected chi-square that required an adjustment to the degrees of freedom (Satorra & Bentler, 1994). \footnote{This version is no longer in much use, but it is available in lavaan for complete data via test = "mean.var.adjusted".}

Under multivariate normality and when $H_0$ is true, $T$ is asymptotically distributed as a central chi-square with $d$ degrees of freedom, and thus its approximate expected value is $E(T) \approx d$. However, if the data are nonnormal, $T$ is asymptotically distributed as a mixture of $d$ independent chi-square variates, each with 1 degree of freedom; the weights of the mixture are given by the eigenvalues of the matrix product $U^0\Omega^0$. The mean-corrected chi-square $T_{MV}$, which corrects by the estimated average eigenvalue of $U^0\Omega^0$, is such that $E(T_M) \approx d$. The mean-and-variance adjusted chi-square $T_{MV}$ is such that $E(T_{MV}) \approx d$ and $\text{var}(T_{MV}) \approx 2d$. A chi-square reference distribution with $d$ degrees of freedom is used for both robust statistics, although neither is asymptotically chi-square distributed, in general. When the data are normal and $H_0$ is true, $a_0 = c_0 = 1$, and $b_0 = 0$, so that asymptotically the scaling corrections do nothing.

Different computational versions of the robust test statistics can be obtained by using different estimates of $a_0$ and $c_0$. Table 6 provides several estimates of $U^0$ and $\Omega^0$ that can be used to compute the scaling corrections. As described in the note to the table, some estimates of $U^0$ will always have the property of reduced rank (equal to $d$) by construction, whereas others will only have this property asymptotically (when $H_0$ is true). Thus, not all versions of the estimate of $c_0$ can be interpreted as correcting $T$ by the average eigenvalue of the estimate of $U^0\Omega^0$. It is unknown whether having this property in finite samples affects the performance of the robust statistics.

While different computational versions of $U^0$ and $\Omega^0$ in Table 6 are asymptotically equivalent when $H_0$ is true, they will differ in finite samples, and the resulting scaling corrections may produce robust statistics that vary in their ability to control Type I error rates. The estimates of each matrix in Table 6 are also no longer asymptotically equivalent when $H_0$ is false, and the resulting robust statistics may also vary in power. In fact, different computational versions of the robust test statistics may have different power even with normal data. This is because variants of $a$ and $c$ that depend on structured estimates of either $U^0$ or $\Omega^0$ will only approach 1 when $H_0$ is true.

Asparouhov and Muthén (2005) offered an alternative expression for $c_0$ in Equation 37:

$$ c_0 = \left( \text{Tr}(B^0_\theta(A^0_\theta)^{-1}) - \text{Tr}(B^0_\theta(A^0_\theta)^{-1}) \right) / d, $$

which follows from Equations 14 and 30, the identity $B^0_\theta = \Delta^0(A^0_\theta)^{-1}$, as well as properties of the trace operator. Additional estimates of $c_0$ are possible based on this equation. Most of these estimates will be equivalent to some estimate of Equation 37, but some will not be expressible in this original form, most notably the version used by default with the MLR.
Table 6. Some estimates of $U^0$ and $Ω^0$.

<table>
<thead>
<tr>
<th>MAR, MCAR, or complete data</th>
<th>Estimates of $U^0$</th>
<th>Estimates of $Ω^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_{U1} = A_b - A_b \hat{A} \hat{A}^{-1} \Delta A_b$</td>
<td>$Ω = \hat{A}^{-1} θ \hat{A}$</td>
<td></td>
</tr>
<tr>
<td>$U = A - A \hat{A} \hat{A}^{-1} \Delta A$</td>
<td>$Ω = \hat{A}^{-1} θ \hat{A}$</td>
<td></td>
</tr>
</tbody>
</table>

| MCAR or complete data only | $U_U = W - W \hat{A} \hat{A}^{-1} \Delta W$ | $Ω = W - W \hat{A} \hat{A}^{-1} \Delta W$ |
| Note: $U = U_U$ with complete data | Note: $U = U_Ω$ with complete data |

Not all estimates are listed; many mixed estimates are possible. All estimates of $U^0$ but $U$ and $U_{U1}$ have rank equal to $d$. The estimates $U$ and $U_{U1}$ approach this property only asymptotically (and only when $H_0$ is true). The scaling corrections are meant to control Type I error rates, which are defined when $H_0$ is true. All listed estimates are consistent for $U^0$ and $Ω^0$.

Table 7. Estimates of $U^0 Ω^0$ used in scaling corrections with complete data in lavaan.

<table>
<thead>
<tr>
<th>lavaan options with test = “satorra.bentler” or test = “scaled.shifted”</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>information = “expected”, hl.information = “structured”</td>
<td>$U_{H1}$</td>
</tr>
<tr>
<td>information = “expected”, hl.information = “unstructured” or information = “observed”, observed.information = “hl”, hl.information = “unstructured”</td>
<td>$U_{H1}$</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “hl”, hl.information = “structured”</td>
<td>$U_{H1}$</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “hessian”, hl.information = “structured”</td>
<td>$U_{H1}$</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “hessian”, hl.information = “unstructured”</td>
<td>$U_{H1}$</td>
</tr>
</tbody>
</table>

Current lavaan defaults are in italics. Note that the inclusion of se = “robust.huber.white” will override the default of expected information. When test = “satorra.bentler” (default with estimator = “MLM”), $T_M$ is computed; when test = “scaled.shifted” (default with estimator = “MLM”), $T_{H1}$ is computed (see Equation 30). These test options are not compatible with missing = “ML”. With both tests, lavaan options only affect the estimate of $U^0$; the saturated estimate $Ω = Ω$ of $Ω^0$ is always used. For different estimates of $Ω^0$, use test = “yuan.bentler” (see Table 9).

Lavaan options for the scaling corrections

The original scaled chi-square $T_M$ (Satorra & Bentler, 1994) was developed for complete data. In lavaan, this test is available via test = “satorra.bentler” (default with estimator = “MLM”), see Table 10). Available computational variations are shown in Table 7. Notably, the estimate of $Ω^0$ is always unstructured, in accordance with the original references (Satorra & Bentler, 1994), and only the computation of $U^0$ varies. The default estimate of $U^0$ is based on expected information evaluated at the structured estimates. The mean-and-variance adjusted chi-square $T_{MV}$ is available in lavaan via test = “scaled.shifted” (default with estimator = “MLM”); because it depends on the same two matrices, the computational options in Table 7 also apply to this test. Neither test = “satorra.bentler” nor test = “scaled.shifted” are compatible with missing = “ML”, so they must be applied to complete data (or listwise deletion will be performed by default).

For incomplete data, only $T_M$ is directly available in lavaan, but $T_{MV}$ can be obtained via custom computations, as we illustrate shortly. To obtain $T_M$ with missing data, the user can specify either test = “yuan.bentler.mplus” (default with estimator = “MLR”), (see Table 10), which estimates $c_q$ using Equation 40, or test = “yuan.bentler”, which estimates $c_0$ using Equation 37. Both these test options can be run with or without missing = “ML”, so they can be used with complete data as well, albeit with different defaults (expected rather than observed information).

Computational variations that are available in lavaan for these tests are shown in Tables 8 and 9, respectively. With test = “yuan.bentler.mplus”, lavaan computational options only affect the second trace term in Equation 40; the first trace term is always computed using saturated estimates. With test = “yuan.bentler”, choices of estimates of $U^0$ and $Ω^0$ can be completely crossed. As with test = “satorra.bentler”, the default computation with test = “yuan.bentler” uses an unstructured estimate of $Ω^0$ (i.e., $Ω$ when missing = “ML” is included) and a structured estimate of $U^0$, which also has the property of reduced rank (i.e., $U_{H1}$ when missing = “ML” is included). Because the default type of information for test = “yuan.bentler” changes to expected information when missing = “ML” is not included, test = “satorra.bentler” and test = “yuan.bentler” will yield an identical value for $T_M$ with complete data (but see footnote 15).

Custom computations

We now illustrate how to verify computations available in lavaan using its internal functions and how to obtain custom computations that are not available in lavaan 0.6–8. We will use the HolzingerSwineford1939 dataset available in lavaan for illustration. Only the first nine variables were used in the analysis; incomplete data were created by randomly

---

14The estimate in Table 10 can be obtained without computing the matrix $Δ$, by obtaining numeric Hessians and by taking the outer product of the individual score vectors (i.e., $δ_j$, $δ_k$).

15This default will be overwritten if se = “robust.huber.white” is also included, because these types of standard errors trigger the use of observed information, and by default the type of information is set to be the same for both standard errors and tests.

16See https://www.rdocumentation.org/packages/lavaan/versions/0.6-5/topics/HolzingerSwineford1939 for description.
Table 8. Scaling correction options based on Equation 40 in lavaan.

<table>
<thead>
<tr>
<th>lavaan options with test = “yuan.bentler.mplus”</th>
<th>Trace Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>information = “expected”, h1.information = “structured” (default without missing = “ML”)</td>
<td>Tr((\hat{\beta}<em>{h1}\frac{p}{l})) - Tr((\hat{\beta}</em>{h1}\frac{p}{k}))</td>
</tr>
<tr>
<td>information = “expected”, h1.information = “unstructured” [*]</td>
<td>Tr((\hat{\beta}<em>{h1}\frac{p}{l})) - Tr((\hat{\beta}</em>{h1}\frac{p}{k})) = Tr((\hat{\mu}_{l})).</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “hessian”, h1.information = “structured”</td>
<td>Tr((\hat{\beta}<em>{h1}\frac{p}{l})) - Tr((\hat{\beta}</em>{h1}\frac{p}{k}))</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “structured”</td>
<td>Tr((\hat{\beta}<em>{h1}\frac{p}{l})) - Tr((\hat{\beta}</em>{h1}\frac{p}{k}))</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “h1”, h1.information = “unstructured” [*]</td>
<td>Tr((\hat{\beta}<em>{h1}\frac{p}{l})) - Tr((\hat{\beta}</em>{h1}\frac{p}{k}))</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “structured”, h1.information = “structured”</td>
<td>Tr((\hat{\beta}<em>{h1}\frac{p}{l})) - Tr((\hat{\beta}</em>{h1}\frac{p}{k}))</td>
</tr>
</tbody>
</table>

The scaling correction c is obtained by dividing the corresponding trace expression by d. The italics show defaults in the current version of lavaan with missing = “ML”. The defaults without missing = “ML” are as shown unless se = “robust.huber.white”. The h1.information option does not affect the first term of the trace expression, which is always computed using saturated estimates. Estimates in cells marked with [*] can also be obtained via test = “yuan.bentler” (see Table 9) because the two mathematical forms of the scaling correction become equivalent. Not all possible options are shown; the h1.information.meat option can be used to specify a different type of estimate for \(\hat{\beta}_{h1}\) only.

Table 9. Scaling correction options based on Equation 37 in lavaan.

<table>
<thead>
<tr>
<th>lavaan options with test = “yuan.bentler”</th>
<th>Estimate of (\hat{u}^p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>information = “expected”, h1.information = “structured” (default without missing = “ML”)</td>
<td>(U_{l})</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “hessian”, h1.information = “structured”</td>
<td>(\hat{U}_{h1})</td>
</tr>
<tr>
<td>information = “observed”, observed.information = “structured”, h1.information = “structured”</td>
<td>(\hat{U}_{h1})</td>
</tr>
</tbody>
</table>

Additional lavaan options with test = “yuan.bentler”

<table>
<thead>
<tr>
<th>omega.information = “expected”, omega.h1.information = “structured”</th>
<th>Estimate of (\sqrt{\hat{\omega}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>omega.information = “unstructured”, omega.h1.information = “structured” (default without missing = “ML”)</td>
<td>(\hat{\Omega}_{l})</td>
</tr>
<tr>
<td>omega.information = “unstructured”, omega.h1.information = “unstructured”</td>
<td>(\hat{\Omega})</td>
</tr>
<tr>
<td>omega.information = “observed”, omega.h1.information = “structured”</td>
<td>(\hat{\Omega})</td>
</tr>
<tr>
<td>omega.information = “observed”, omega.h1.information = “unstructured”</td>
<td>(\hat{\Omega})</td>
</tr>
</tbody>
</table>

The defaults in the current version of lavaan with missing = “ML” are in italics. The defaults without missing = “ML” are as shown unless se = “robust.huber.white”. The options for estimates of \(\hat{u}^p\) and \(\sqrt{\hat{\omega}}\) can be completely crossed. If the additional options are omitted, the default computation of \(\sqrt{\hat{\omega}}\) is performed. Not all possible options are shown; the h1.information.meat option can be used to specify mixed estimates of \(\hat{u}^p\), and the omega.h1.information.meat option can be used to specify mixed estimates of \(\sqrt{\hat{\omega}}\).

The syntax below fits the model using MLR and shows how to verify the default settings for information options under this choice of estimator:

```r
library(lavaan)
set.seed(1)
HS.comp <- HolzingerSwineford1939[, part00="c", 1:9]
HS.missing <- as.data.frame(lapply(HS.comp, function(x) { x[ sample(1:length(x), 20) ] <- NA } ))
HS.model <- visual = x1 + x2 + x3
teststat = x0 + x5 + x6
speed = x7 + x8 + x9

# fit with MLR
fit.ml <- sem(HS.model, data = HS.missing, missing = "ML", estimator="MLR")
lavInspect(fit.ml, options = "test")
# [1] "yuan.bentler.mplus"
lavInspect(fit.ml, options = "information")
# [1] "observed" "observed"
lavInspect(fit.ml, options = "h1.observed information")
# [1] "hessian" "hessian"
lavInspect(fit.ml, options = "h1.hessian information")
# [1] "structured" "structured"

# fit with PML
fit.ml <- sem(HS.model, data = HS.missing, missing = "ML", estimator="PML")
lavInspect(fit.ml, options = "test")
# [1] "yuan.bentler.mplus"
lavInspect(fit.ml, options = "information")
# [1] "observed" "observed"
lavInspect(fit.ml, options = "h1.observed information")
# [1] "hessian" "hessian"
lavInspect(fit.ml, options = "h1.hessian information")
# [1] "structured" "structured"

# fit with robust MLR
fit.ml <- sem(HS.model, data = HS.missing, missing = "ML", estimator="MLR")
lavInspect(fit.ml, options = "test")
# [1] "yuan.bentler.mplus"
lavInspect(fit.ml, options = "information")
# [1] "observed" "observed"
lavInspect(fit.ml, options = "h1.observed information")
# [1] "hessian" "hessian"
lavInspect(fit.ml, options = "h1.hessian information")
# [1] "structured" "structured"

# fit with robust PML
fit.ml <- sem(HS.model, data = HS.missing, missing = "ML", estimator="PML")
lavInspect(fit.ml, options = "test")
# [1] "yuan.bentler.mplus"
lavInspect(fit.ml, options = "information")
# [1] "observed" "observed"
lavInspect(fit.ml, options = "h1.observed information")
# [1] "hessian" "hessian"
lavInspect(fit.ml, options = "h1.hessian information")
# [1] "structured" "structured"
```

This code illustrates that one can obtain the estimates of \(\hat{\beta}_{h1}\), \(\hat{\beta}_{h2}\), \(\hat{\beta}_{h3}\) using the lavInspect function. Options for \(\hat{\beta}_{h1}\) include “h1.information.observed” and “h1.information.expected”. Similarly, options for \(\hat{\beta}_{h2}\) include “information.observed” (which produces the.

---

17 For advanced users, we note that lavaan organizes H matrises corresponding to the order (\(\mu_\cdot (\text{vech}\Sigma)\)); e.g., see row.names(Abats.unstr).
Table 10. “MLM” and “MLR” definitions in lavaan.

<table>
<thead>
<tr>
<th>Equivalent Specification</th>
<th>estimator = “MLM”</th>
<th>estimator = “MLL”, mimic = “Mplus”</th>
<th>estimator = “MLR”</th>
</tr>
</thead>
<tbody>
<tr>
<td>se = “robust.sem”</td>
<td>( \hat{A}_g^{\hat{B}} \hat{A}_g )</td>
<td>( \hat{A}_g^{\hat{B}} \hat{A}_g )</td>
<td>( \hat{A}_g^{\hat{B}} \hat{A}_g )</td>
</tr>
<tr>
<td>test = “satorra.bentler”, h1.information = “structured”</td>
<td>( \hat{A}_g^{\hat{B}} \hat{A}_g )</td>
<td>( \hat{A}_g^{\hat{B}} \hat{A}_g )</td>
<td></td>
</tr>
</tbody>
</table>

Options triggered by default are in italics. Specifying mimic = “Mplus” leads lavaan to mimic the “MLM” method as implemented in Mplus, which uses unstructured estimates (in the “Equivalent Specification” syntax for this option, setting information = “observed” will produce identical results because \( \hat{A}_g^{\hat{B}} = \hat{A}_g \) with complete data). The “MLM” option is not compatible with missing = “ML”. Not included in the table is the “MLMV” option, which differs from “MLM” only in that it sets test = “scaled.shifted”, so that \( \hat{T}_M \) instead of \( \hat{T}_N \) is computed, with the same defaults. Under the default options for “MLM”, the resulting standard errors will correctly adjust for nonnormality, but not for model misspecification. Under the default options for “MLR” (i.e., using the numeric Hessian as the “bread”), the resulting standard errors will adjust for nonnormality and model misspecification.

The code in this section can be adapted to compute any other variation of the robust statistics or robust standard errors.

### Discussion

In this article, we have summarized the different computational versions of standard errors and test statistics that are used with incomplete normal and nonnormal data in SEM. Complete data equations were included as a special case. We have illustrated how to obtain the available options in the popular R package lavaan, and have shown how to obtain currently unavailable options via lavaan’s internal functions. We hope that this article encourages the extensive study of these computational options by methodologists. Scenarios of interest would be formed by the intersection of the following options: complete and different types of incomplete data, normal and different types of nonnormal data, and different types of correctly specified and misspecified models. We briefly review the existing literature, which is largely limited to complete data.

### Existing comparisons

With complete data, some research has been devoted to investigating whether saturated or structured estimates of the expected information matrix result in better performance of robust standard errors and test statistics. Noting that with complete data, EQS uses structured estimates in the computation of the expected information matrix whereas Mplus uses saturated estimates, Xia et al. (2016) compared the impact of this choice on the performance of robust standard errors and of \( \hat{T}_M \). They investigated data with varying degrees of nonnormality, and...
included both correct and incorrect models. They found that structured estimates were preferred in both computations. Foldnes and Olsson (2019) replicated this finding with models of varying size and degrees of freedom and with additional types of nonnormal data; these authors only studied correctly specified models.

The impact of using observed versus expected information estimates with complete data has been investigated by several authors. Dolan and Molenar (1991) found very minor differences between observed and expected information in ML standard errors with normal data (see also Maydeu-Olivares, 2017). Studying only correct models, Savalei (2010a) found that observed information outperformed expected information in both ML and robust standard errors, with both complete and incomplete data. Differences were more substantial with incomplete MAR data, as would be theoretically expected.

In the context of complete data, Maydeu-Olivares (2017) conducted a comprehensive investigation of ML and robust standard errors as implemented in Mplus. Models with varying degree of misspecification were studied. Notably, ML standard error estimates based on the outer product matrix were included, which exhibited significant bias even with normal data and under correct model specification. Three types of robust standard errors were investigated, which correspond to Mplus’s version of MLM (see Table 10), MLR, and ML with expected information. When the model was correctly specified, all three types of standard errors performed well with normal and nonnormal data. When the model was misspecified, the default MLR standard errors outperformed the other two, as would be theoretically expected (see also Lai, 2018). Finally, robust test statistics under the same three specifications were also investigated. Scaling corrections based on Equation 37, which are used with MLR, performed the worst, regardless of type of information. The MLM statistic performed only slightly better, although this may be because it is based on saturated estimates in Mplus (Xia et al., 2016). The statistic that performed best was $T_{MV}$ (estimator = “MLMV”), also based on the saturated estimates. Overall, it is hard to draw firm conclusions from this study because several options are confounded in the default implementation of options in Mplus.

**Future research**

In this section, we outline some interesting future research directions. When it comes to standard errors, given the good relative performance of Huber-White standard errors (default with MLR) in existing studies with complete data (Lai, 2018; Maydeu-Olivares, 2017), further studies of how these standard errors perform with different types of incomplete data are needed. These standard errors are robust to both nonnormality and model misspecification. Often in statistics, the price of such broad robustness is poorer performance relative to methods that make more assumptions, as long as those assumptions are correct. However, good performance of these standard errors with complete data established in prior research is encouraging in this regard. When the model is correct or approximately correct, many other comparisons become of interest. One is between the use of the numeric Hessian versus the analytic asymptotic approximations in both ML and robust standard errors; these two estimates have never been directly compared, though the latter version has been used in some studies of standard errors with incomplete data (Savalei & Bentler, 2009; Savalei & Falk, 2014). Another comparison of interest is between structured and saturated estimates in robust standard errors, to determine whether findings from complete data (e.g., Xia et al., 2016) generalize to incomplete data.

Another research direction would be to explore whether the property of consistency of Huber-White standard errors is meaningful when the model is misspecified. In existing studies with misspecified models, Huber-White standard errors have been evaluated based on whether they accurately approximate the actual empirical variability of parameter estimates across replications and whether they have good coverage of the pseudo-parameters. To understand the meaning of the pseudo-parameters, consider a population 1-factor model with five indicators, with unit variances, with loadings of $.7$, and a residual covariance of $.3$ between the first two indicators. Suppose a misspecified model with no residual covariance is fit to this population. The true loadings are $\lambda^0 = (.7, .7, .7, .7, .7)^T$, but the loadings obtained by fitting the 1-factor model without a residual covariance to the population covariance matrix are $\lambda = (.87, .87, .595, .595, .595)^T$. To complicate matters further, with incomplete data, these pseudo-parameters may vary with the characteristics of missing data (X. Zhang & Savalei, 2020). Consistency of Huber-White standard errors in this situation means that in large samples they will provide accurate estimates of the variability of $\lambda$ around the pseudo-parameters $\lambda^0$, not around the true loadings $\lambda^0$. Simulation studies with misspecified models should consider adding measures such as coverage defined around the true values (in our example, $\lambda^0$) rather than around the pseudo-values ($\lambda$). Huber-White standard errors are no longer guaranteed to have better performance for such measures.

When it comes to the test statistics, the impact of the computational options on the scaling corrections has been studied very little even with complete data, and with incomplete data, it is entirely unstudied. Studies with complete data have shown that structured estimates of $U^0$ are preferred when computing $T_M$ (Xia et al., 2016). Future research should explore whether this finding generalizes to incomplete data. With complete data, typically a saturated estimate of $\Omega^0$ is always used regardless of how $U^0$ is estimated. Future research should investigate the impact of using saturated versus structured estimates in the matrices involved in the scaling, with both complete and incomplete data. We should note that existing studies of robust with incomplete data statistics have not always made clear how these matrices were estimated.

Studies with complete data have also found that the variant of the scaling correction for $T_M$ based on Equation 40 (default with MLR) performs worse (Maydeu-Olivares, 2017), but similar comparisons with incomplete data have not been conducted. While some authors have argued that the estimate of...
Tr(U^\Omega U^\Omega) needs to have rank d in order for the scaling correction to make sense (Bentler & Yuan, 1999; Jiange & Yuan, 2017), in our view it is the finite sample performance that should be used to judge the quality of the various versions of the scaling corrections. Asparouhov and Muthén (2005) have also identified other benefits of estimates of Equation 37, such as ease of difference tests computations.

The statistic \( T_{MV} \) has been recommended with both complete and incomplete MCAR data based on its performance in limited simulation studies (Maydeu-Olivares, 2017; Savalei, 2010b). However, it is not often used with incomplete data because it is not readily available in software such as Mplus or lavaan. We have shown how to obtain it using custom computations in lavaan. The various computational options for it have never been studied. Aside from comparing the relative performance of the different computational versions of \( T_M \) and \( T_{MV} \) to find the best implementation of each, a thorough comparison, in terms of Type I error rates and power, of the best performing version of \( T_M \) and \( T_{MV} \) to each other would be needed with both complete and incomplete data.

In addition to the standard errors and test statistic, the options affecting the computation of the information matrix and the outer product matrix will impact other quantities, such as fit indices and estimates of fraction of missing information. When data are nonnormal, robust difference tests rely on the same scaling corrections, robust equations for RMSEA and CFI use the scaling correction \( c \) to improve finite sample performance (Brosseau-Liard & Savalei, 2014; Brosseau-Liard et al., 2012). Thus, they too will vary depending on how the scaling correction is computed. With complete data, Xia et al. (2016) found that robust RMSEA and CFI were affected very little by whether saturated or structured estimates were used in the information matrix to compute \( c \). Further, the Mplus variant of the scaling correction has not been studied in this context. The information matrix is also used to estimate the fraction of missing information for each parameter; we are currently investigating how the computational choices for the information matrix affect these estimates. Scaling corrections are also used when computing different versions of robust difference tests for nested models (Satorra & Bentler, 2001, 2010), and the impact of computational choices on the performance of these tests has not been studied. Finally, other estimation methods for incomplete nonnormal data, such as two-stage ML (Savalei & Falk, 2014; Yuan & Lu, 2008) are also affected by the same computational choices, and their impact needs to be studied.

Most simulation studies in the SEM literature generate nonnormal data using the Vale-Maurelli method (Vale & Maurelli, 1983), which is a generalization of the Fleishman method (Fleishman, 1978). This method permits the user to specify univariate skewness and kurtosis for each variable, and then subsequently rotates the set of variables to the pre-specified mean and covariance structure. This method has been criticized on several grounds (e.g., Astivia & Zumbo, 2019), but the most significant criticism is that it generates very limited kinds of multivariate nonnormal distributions, ones with multivariate properties very close to the Gaussian despite nonnormal marginals (Foldnes & Grenneberg, 2015). While many existing studies of test statistics took care to demonstrate that the LR chi-square statistic \( T \) did break down in the generated nonnormal data, it is not known whether the findings that robust statistics performed well in these studies will hold up for nonnormal data generated in other ways. For robust standard errors, it has been shown that the type of nonnormality does influence performance (Falk, 2018).

Many alternative methods of nonnormal data generation exist (e.g., Grenneberg & Foldnes, 2017; Mair et al., 2012; Qu et al., 2019), and the performance of the various computational versions of robust standard errors and test statistics should be evaluated on data with more varied types of multivariate nonnormality.

Alternative types of uncertainty estimates for nonnormal data exist. Lai (2019) proposed an improvement to MLR standard errors based on a second-order approximation, and Falk (2018) proposed robust likelihood-based confidence intervals. Extensions of these methods to incomplete data present interesting future research directions. Future research should also consider extending and studying variants of Huber-White standard errors for handling outlying observations, which have been developed in the regression context (e.g., MacKinnon & White, 1985), but extensions to SEM are limited (but see Yuan & Zhang, 2012). Finally, in this article, we have dealt with the most straightforward type of mean and covariance structure model. More complicated models would involve conditioning on exogenous covariates, the presence of constraints, or the presence of interaction terms. The equations in this article do not cover these extensions, and these extensions should be subject of future research.

**Conclusion**

We hope that the present article encourages methodologists to investigate the performance of the different computational versions of standard errors and robust test statistics in a systematic way and across a wide variety of types of incomplete data, model misspecification, and nonnormality. It would be important to conduct such studies without confounding the available computational variations by using software defaults. We have provided the lavaan code to disentangle the available options. We also hope that this article communicates the importance of reporting exactly how a particular method was implemented even in studies that do not make computational variations their focus; otherwise, future aggregation of results across studies will not be meaningful.

While we have outline types of standard error and scaling correction estimates that are consistent under various scenarios, we believe it is too early for applied researchers to use this information in order to select which options to use in practice. Theoretical expectations and practical performance do not always align. Consistency is not the only desirable property of an estimator, and small sample performance, measured by bias and efficiency, would be relevant. For example, while theoretically there is usually no good reason to use expected information in computations with incomplete data (exceptions would include a planned missingness design, Graham et al., 2006), in practice

---

18While Maydeu-Olivares (2017) included RMSEA in the study, the default Mplus implementation was studied, which does not correctly adjust for nonnormality; see Brosseau-Liard et al. (2012).
the expected information matrix may be more stable in small samples. We have found, for example, that its use in the scaling correction sometimes produces more interpretable results when the independence model is fit to data (e.g., for the purposes of computing CFI). Overall, the scarcity of research on the relative performance of standard errors, scaling corrections, and other statistics dependent on them, especially with incomplete data, means that it would be premature to make recommendations. The goal of the present article is to make the available options explicit and to encourage their study by methodologists.

**Funding**

This work was supported by the National Sciences and Engineering Research Council of Canada (NSERC) to Victoria Savalei.

**ORCID**

Yves Rosseel (http://orcid.org/0000-0002-4129-4477)

**References**


**Multiple Corresponding ORCIDs**

Yves Rosseel (http://orcid.org/0000-0002-4129-4477)


Appendix

A. Derivation of $A_{\beta,n}$ and $A_{\beta,n}$

Let $\hat{\beta}_j = (\text{vec}\Sigma_j', \mu_j')$. The first differential of the log-likelihood in Equation 1 is given by (for how to work with differentials, see Magnus & Neudecker, 1999):

$$dl(\hat{\beta}_j) = -\frac{1}{2} \text{Tr}(\Sigma_j^{-1} d\Sigma_j) + (y_i - \mu_j)' \Sigma_j^{-1} d\mu_i + \frac{1}{2} (y_i - \mu_j)' \Sigma_j^{-1} d\Sigma_j \Sigma_j^{-1} (y_i - \mu_j)$$

The first differential of the first term of $dl(\hat{\beta}_j)$ is given by:

$$-\frac{1}{2} d\text{Tr}(\Sigma_j^{-1} d\Sigma_j) = -\frac{1}{2} d\text{Tr}(d\Sigma_j^{-1} d\Sigma) = -\frac{1}{2} d\text{Tr}(d\Sigma_j^{-1} d\Sigma_j^{-1} d\Sigma) = \frac{1}{2} (d\text{vec}\Sigma_j)' (\Sigma_j^{-1} \otimes \Sigma_j^{-1}) d\text{vec}\Sigma_j$$

The first differential of the second term of $dl(\hat{\beta}_j)$ is given by:

$$d((y_i - \mu_j)' \Sigma_j^{-1} d\mu_i) = -d\mu_i' \Sigma_j^{-1} d\mu_i = (y_i - \mu_j)' \Sigma_j^{-1} d\Sigma_j \Sigma_j^{-1} d\mu_i = (d\text{vec}\Sigma_j)' (\Sigma_j^{-1} (y_i - \mu_j) \otimes \Sigma_j^{-1}) d\mu_i$$

The first differential of the third term of $dl(\hat{\beta}_j)$ is given by:

$$\frac{1}{2} d((y_i - \mu_j)' \Sigma_j^{-1} d\Sigma_j \Sigma_j^{-1} (y_i - \mu_j)) = -0.5 (y_i - \mu_j)' \Sigma_j^{-1} d\Sigma_j \Sigma_j^{-1} d\mu_i - 0.5 (y_i - \mu_j)' \Sigma_j^{-1} d\Sigma_j \Sigma_j^{-1} d\Sigma_j \Sigma_j^{-1} (y_i - \mu_j)$$

$$= - (d\text{vec}\Sigma_j)' (\Sigma_j^{-1} (y_i - \mu_j) \otimes \Sigma_j^{-1}) d\mu_i - (d\text{vec}\Sigma_j)' (\Sigma_j^{-1} \otimes \Sigma_j^{-1}) (y_i - \mu_j) (y_i - \mu_j)' \Sigma_j^{-1} (d\text{vec}\Sigma_j)$$

The second differential of $l(\hat{\beta}_j)$ is the sum of the three components above (components of the first differential of $dl(\hat{\beta}_j)$), and is given by:

$$d^2 l(\hat{\beta}_j) = 0.5 (d\text{vec}\Sigma_j)' (\Sigma_j^{-1} \otimes \Sigma_j^{-1}) d\text{vec}\Sigma_j - d\mu_i' \Sigma_j^{-1} d\mu_i - 2 (d\text{vec}\Sigma_j)' (\Sigma_j^{-1} (y_i - \mu_j) \otimes \Sigma_j^{-1}) d\mu_i - (d\text{vec}\Sigma_j)' (\Sigma_j^{-1} \otimes \Sigma_j^{-1}) (y_i - \mu_j) (y_i - \mu_j)' \Sigma_j^{-1} (d\text{vec}\Sigma_j)$$

The negative Hessian for the $i$th observation is (Magnus & Neudecker, 1999):

$$-\frac{\partial^2 l(\hat{\beta}_j)}{\partial \hat{\beta}_j \partial \hat{\beta}_j'} = \left( D_{\beta} (\Sigma_j^{-1} \otimes (\Sigma_j^{-1} (y_i - \mu_j) (y_i - \mu_j)' \Sigma_j^{-1} - 0.5 \Sigma_j^{-1})) D_{\beta}' \right)$$

$$\times \frac{0}{\Sigma_j^{-1}}$$

To obtain the Hessian with respect to $\beta = (\text{vec}\Sigma_j', \mu_j')$, we write $\hat{\beta}_j = ((\text{vec}\Sigma_j'), \mu_j') = (D_{\beta} \kappa_j, \text{vec}\Sigma_j', (\tau_j \mu))$, so that $\frac{\partial^2 l}{\partial \beta \partial \beta'} = \left( \begin{array}{c} D_{\beta} \kappa_j \\ 0 \\ \tau_j \end{array} \right)$. Using the identity $D_{\beta} D_{\beta}' \kappa_i = \kappa_i$, we obtain

$$A_{\beta,n} = -\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta'} = -\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta'} \frac{\partial^2 l(\beta)}{\partial \beta \partial \beta'}$$

$$= \left( \begin{array}{c} (\Sigma_j^{-1} \otimes (\Sigma_j^{-1} (y_i - \mu_j) (y_i - \mu_j)' \Sigma_j^{-1} - 0.5 \Sigma_j^{-1})) \kappa_j \\ (y_i - \mu_j)' \Sigma_j^{-1} \otimes \tau_j \Sigma_j^{-1} \kappa_j \\ \tau_j \Sigma_j^{-1} \tau_j \end{array} \right)$$

Sorting by missing data patterns and averaging across $n$, we obtain

$$A_{\beta,n} = \frac{1}{n} \sum_{j=1}^{J} h_j \frac{\partial^2 l(\beta)}{\partial \beta \partial \beta'} = \left( \begin{array}{c} (\Sigma_j^{-1} \otimes (\Sigma_j^{-1} \Sigma_j^{-1} + h_j \kappa_j)) - 0.5 \Sigma_j^{-1} \kappa_j \\ (h_j \Sigma_j^{-1} \otimes \tau_j \Sigma_j^{-1}) \kappa_j \\ \tau_j \Sigma_j^{-1} \tau_j \end{array} \right)$$

where $\bar{y}_j = \frac{1}{n} \sum_{i=1}^{n} y_i$, $h_{j,n} = \Sigma_j^{-1} (\bar{y}_j - \mu_j)$, and $S_j = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y}_j) (y_i - \bar{y}_j)'$ for $j = 1, \ldots, J$. See also (Magnus & Neudecker, 1999), p. 319, Exercise 2.

B. Derivation of $B_{\beta,n}$

Here we provide expressions for the components of $B_{\beta,n}$ in Equation 20. We write
where the blocks correspond to the partitioning of \( \beta = (\text{vech}\Sigma', \mu') \). An explicit expression for \( d_{\beta,i} \) in Equation 2 is

\[
d_{\beta,i} = \left( \frac{\partial l_i(\beta)}{\partial (\text{vech}\Sigma)}, \frac{\partial l_i(\beta)}{\partial \mu} \right)
\]

\[
= \left( 0.5 \text{vec} \left( \Sigma_j^{-1}(y_i - \mu_j)(y_i - \mu_j)' \Sigma_j^{-1} - \Sigma_j^{-1} \right) \kappa_j, (y_i - \mu_j)' \Sigma_j^{-1} \tau_j \right)'
\]

Let \( w_i = \text{vec}(y_i - \mu_j)(y_i - \mu_j)' \). For the \( i \)th observation belonging to the \( j \)th missing data pattern, we have:

\[
\frac{\partial l_i(\beta)}{\partial \mu} \frac{\partial l_i(\beta)}{\partial \mu'} = \tau_j' \Sigma_j^{-1}(y_i - \mu_j)(y_i - \mu_j)' \Sigma_j^{-1} \tau_j
\]

\[
= 0.5 \tau_j' \Sigma_j^{-1}(y_i - \mu_j)(w_i - \text{vec}\Sigma_j)(w_i - \text{vec}\Sigma_j)'(\Sigma_j^{-1} \otimes \Sigma_j^{-1}) \kappa_j
\]

\[
\frac{\partial l_i(\beta)}{\partial \text{vech}\Sigma} \frac{\partial l_i(\beta)}{\partial \text{vech}\Sigma} = \frac{1}{4} \kappa_j'(\Sigma_j^{-1} \otimes \Sigma_j^{-1})(w_i - \text{vec}\Sigma_j)(w_i - \text{vec}\Sigma_j)'(\Sigma_j^{-1} \otimes \Sigma_j^{-1}) \kappa_j.
\]

Averaging across \( n \) leads to:

\[
B_{22} = \sum_{j=1}^{J} \frac{n}{n} \tau_j' \Sigma_j^{-1} M_j \Sigma_j^{-1} \tau_j,
\]

\[
B_{21} = \sum_{j=1}^{J} \frac{n}{n} \tau_j' \Sigma_j^{-1} T_j (\Sigma_j^{-1} \otimes \Sigma_j^{-1}) \kappa_j,
\]

\[
B_{11} = \sum_{j=1}^{J} \frac{n}{n} \kappa_j'(\Sigma_j^{-1} \otimes \Sigma_j^{-1}) V_j (\Sigma_j^{-1} \otimes \Sigma_j^{-1}) \kappa_j,
\]

where \( M_j = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu_j)(y_i - \mu_j)' = S_j + (\bar{y}_j - \mu_j)(\bar{y}_j - \mu_j)' \), \( T_j = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu_j)(w_i - \text{vec}\Sigma_j)' \), and \( V_j = \frac{1}{n} \sum_{i=1}^{n} (w_i - \text{vec}\Sigma_j)(w_i - \text{vec}\Sigma_j)' \). Define the following matrix:

\[
\Gamma_{1,n} = \frac{1}{n} \sum_{i=1}^{n} t_i' \cdot \left( \frac{(D_{p}\beta)^{-1} \kappa_j' V_j (D_{p}\beta)^{-1}}{\tau_j' T_j (D_{p}\beta)^{-1}}, \frac{\tau_j' \kappa_j'}{\tau_j' M_j \tau_j} \right),
\]

where \( t_i = \left( (w_i - \text{vec}\Sigma_j)' D_{p}' \beta, (y_i - \mu_j)' \right)' \). The expression in Equation 20 follows by noting that \( d_{\beta,i} = W_i t_i \). The following identity has been used: \( \kappa_j'(\Sigma_j^{-1} \otimes \Sigma_j^{-1}) = \kappa_j'(\Sigma_j^{-1} \otimes \Sigma_j^{-1}) \kappa_j(D_{p}\beta)^{-1} \kappa_j' \). In the special case of complete data,

\[
\Gamma_n = \frac{1}{n} \sum_{i=1}^{n} t_i' \cdot \left( \frac{(D_{p}\beta)^{-1} \kappa_j' V (D_{p}\beta)^{-1}}{T D_{p}' S + (\bar{y} - \mu)'(\bar{y} - \mu)} \right),
\]

where \( t_i = \left( (w_i - \text{vec}\Sigma)' D_{p}' \beta, (y_i - \mu)' \right)' = \left( (\text{vech}(y_i - \mu)(y_i - \mu)' - \text{vech}\Sigma)', (y_i - \mu)' \right)' \), \( w_i = \text{vec}(y_i - \mu)(y_i - \mu)' \), \( T = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu)(w_i - \text{vec}\Sigma)' \), and \( V = \frac{1}{n} \sum_{i=1}^{n} (w_i - \text{vec}\Sigma)(w_i - \text{vec}\Sigma)' \).

C. Equivalence of \( B_{\beta}^\beta \) and \( A_{\beta}^\beta \) with MCAR

As shown in part B of this Appendix, components of \( B_{\beta}^\beta \) depend on third- and fourth-order sample moments of the data. In the limit these moments simplify with normally distributed MCAR data. Under MCAR, data within each pattern are a random sample from the same population, so that \( \bar{y}_j \rightarrow \mu_j^0 \) and \( S_j \rightarrow \Sigma_j^0 \) for \( j = 1, \ldots, J \). This implies that \( M_j^0 = S_j + (\bar{y}_j - \mu_j^0)(\bar{y}_j - \mu_j^0)' \rightarrow \Sigma_j^0 \) for all \( j \). Normality implies that the third-order moments about the true mean tend to zero, so that \( T_j^0 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu_j^0)' \left( (\text{vech}(y_i - \mu_j^0)(y_i - \mu_j^0)' - \text{vech}\Sigma_j^0)' \right) ' \rightarrow 0 \) for all \( j \). Under normality, fourth-order moments are
functions of second-order moments, \( V_p^0 = \frac{1}{2} \sum_{i=1}^{n} (y_i^2 - \text{vec}(\Sigma_\theta))^T (y_i^2 - \text{vec}(\Sigma_\theta)) \) \( \rightarrow 2D_p D_p' (\Sigma_\theta^0 \otimes \Sigma_\theta^0) \) for all \( j \). Then, \( B_0^0 \rightarrow W_0^0 \). Under MCAR, \( \zeta_j = \mu_0^0 \) for all \( j \), so that \( A_0^0 = W_0^0 \), and therefore \( B_0^0 = A_0^0 \). This result also holds for complete data by viewing it as a special case of MCAR data with \( J = 1 \).

D. Equivalence of \( B_0 \) and \( B_0 \) with complete data when \( \hat{\mu} = \hat{y} \)

We show that with complete data and when the mean structure is saturated, \( B_0 = B_0 \), where \( B_0 = \hat{\Delta}' \hat{W}_c \hat{W}_c \hat{\Delta}, B_0 = \hat{\Delta} \hat{W}_c \hat{W}_c \hat{\Delta} \), where \( \hat{\Delta} \) is defined in Equation 24 and \( \hat{\Gamma} = \frac{1}{n} \sum_{i=1}^{n} i_i \). With \( \hat{\mu} = \hat{\mu} = \hat{y} \),

\[ \hat{w}_i = \hat{w}_i = \text{vec}(\hat{y}_i - \hat{y})(\hat{y}_i - \hat{y})' \]

\[ \hat{T} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - \hat{y}) (\hat{w}_i - \text{vec}(\hat{\Sigma})) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - \hat{y}) (\hat{w}_i - \text{vec}(\hat{\Sigma}))' = \hat{T}, \]

and

\[ \hat{V} = \frac{1}{n} \sum_{i=1}^{n} (\hat{w}_i - \text{vec}(\hat{\Sigma})) (\hat{w}_i - \text{vec}(\hat{\Sigma}')) + (\text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})) (\text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})))' = \hat{V} = (\text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})) (\text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})))' \]}

Then,

\[ \hat{\Gamma} = \hat{\mu} + \begin{pmatrix} D_p' \text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})) (\text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})))' D_p \end{pmatrix} \]

\[ \hat{B}_0 = \hat{B}_0 + .25 \hat{\Lambda}_1 D_p' \text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})) (\text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})))' (\text{vec}(\hat{S} - \text{vec}(\hat{\Sigma})))' D_p \hat{\Lambda}_1, \]

where \( \hat{\Lambda}_1 = \frac{\partial \text{vec}(\hat{\Sigma})}{\partial \theta} \) is a \( p \times q \) submatrix of \( \hat{\Delta} \). For properties of duplication matrices, see Magnus and Neudecker (1999). With complete data,

\[ d_{\theta,0} = .5 \text{vec}(\Sigma_\theta^{-1} (\hat{y} - \mu)(\hat{y} - \mu)' \Sigma_\theta^{-1} - \Sigma_\theta^{-1} D_p, (\hat{y} - \mu)' \Sigma_\theta^{-1})' \]

At the solution, \( \hat{d}_{\theta,0} = \hat{d}_{\theta,0} \hat{\Delta} = 0 \). When \( \hat{y} = \hat{\mu} \), the expression for \( d_{\theta,0} \) simplifies. Setting it to zero yields, for the covariance structure part,

\[ \text{vec}(\Sigma_\theta^{-1} \Sigma_\theta^{-1} - \Sigma_\theta^{-1})' D_p \hat{\Delta}_1 = 0, \]