Journal of Statistical Software

November 2021, Volume 100, Issue 6. *doi: 10.18637/jss.v100.i06*

Efficient Bayesian Structural Equation Modeling in Stan

Edgar C. Merkle
University of Missouri

Ellen Fitzsimmons
University of Missouri

James Uanhoro
Ohio State University

Ben Goodrich
Columbia University

Abstract

Structural equation models comprise a large class of popular statistical models, including factor analysis models, certain mixed models, and extensions thereof. Model estimation is complicated by the fact that we typically have multiple interdependent response variables and multiple latent variables (which may also be called random effects or hidden variables), often leading to slow and inefficient posterior sampling. In this paper, we describe and illustrate a general, efficient approach to Bayesian SEM estimation in Stan, contrasting it with previous implementations in R package **blavaan** (Merkle and Rosseel 2018). After describing the approaches in detail, we conduct a practical comparison under multiple scenarios. The comparisons show that the new approach is clearly better. We also discuss ways that the approach may be extended to other models that are of interest to psychometricians.

Keywords: Bayesian SEM, **blavaan**, **JAGS**, MCMC, structural equation model, Stan.

1. Introduction

Structural equation models (SEMs) are commonly used in the social sciences, where it is customary to (attempt to) measure unobservable traits such as cognitive abilities, attitudes, and proficiencies. Such models provide a formal way of connecting these unobservable traits to related, observed variables (e.g., test scores, Likert responses, etc), which has led to longstanding popularity of SEMs. SEMs are also related to research on causality and directed acyclic graphs (e.g., Pearl 2013), to generalized linear mixed models (e.g., Bates, Mächler, Bolker, and Walker 2015; Gelman, Carlin, Stern, Dunson, Vehtari, and Rubin 2013; Stroup

2013), and to time series models (e.g., Driver, Oud, and Voelkle 2017), illustrating the models' broad applicability across disciplines.

A defining feature of the SEM framework is the ability to instantiate regressions on latent variables, as opposed to observed variables. This framework is more general than the traditional mixed modeling framework, allowing for products between latent variables and other free parameters, as might be seen in factor analysis (e.g., Bollen 1989; Merkle and Wang 2018). The generality of SEM implies that the estimation methods are relatively complex, which has historically led researchers to rely on closed-source implementations of optimization methods via software like **Mplus** (Muthén and Muthén 2017), **LISREL** (Jöreskog and Sörbom 1997), and **EQS** (Bentler 2008). A small number of more recent R (R Core Team 2021) packages, including **sem** (Fox, Nie, and Byrnes 2021), **OpenMx** (Boker *et al.* 2011), and **lavaan** (Rosseel 2012), provide open source SEM functionality that utilize classical estimation methods including maximum likelihood or least squares.

While maximum likelihood and least squares methods are most popular, Bayesian approaches to SEM and related models have received increased recent attention (e.g., Depaoli and van de Schoot 2017; Fox 2010; Jackman 2009; Kaplan 2014; Kruschke 2011; MacCallum, Edwards, and Cai 2012; Merkle and Wang 2018; Muthén and Asparouhov 2012; Van Erp, Mulder, and Oberski 2018). Researchers have specifically found the methods to be useful for estimation of complex SEMs (including, e.g., latent variable interactions; Lee, Song, and Tang 2007), for automatically handling uncertainty associated with latent variable estimation, and for scaling to high-dimensional datasets.

Despite the increased popularity of Bayesian latent variable models, coding the models via **JAGS** (Plummer 2003) or Stan (Carpenter *et al.* 2017) syntax can be difficult, and the resulting sampling can be time-consuming and inefficient. These issues have been partially addressed by R package **blavaan** (Merkle and Rosseel 2018), which uses **lavaan** model specification syntax and originally relied on **JAGS** for model estimation (via package **runjags**, which provides an R interface to **JAGS**; see Denwood 2016). Package **blavaan** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=blavaan`. Other R packages have addressed these issues for related models, including **brms** (Bürkner 2017) for mixed and multivariate models, **rstanarm** (Gabry and Goodrich 2020) for regression models, **ctsem** (Driver *et al.* 2017) for time series models, **edstan** (Furr 2017) for item response models, and **pcFactorStan** (Pritikin 2021) for pairwise comparison factor models.

The original **blavaan** approach was similar to the **brms** approach for generalized linear mixed (and related) models, where **JAGS** code was generated at runtime from the user-specified model syntax. However, this approach became very slow for some models, forcing the user to wait hours or more for enough samples to make inferences. This reduced the viability of **blavaan** for applied data analysis and for further development, leading us to implement Stan functionality in **blavaan**. The original Stan implementation was similar to the **JAGS** implementation, generating Stan syntax for a user-specified model and relying on package **rstan** (Stan Development Team 2021) for Markov chain Monte Carlo (MCMC). This Stan implementation has not been formally described, which represents one contribution of the current paper. In general, though, the Stan implementation was not much faster or more efficient than the **JAGS** approach.

The primary contribution of this paper is to describe and illustrate a new approach to Stan SEM estimation, which greatly improves the speed and efficiency of model estimation. The approach can be flexibly applied to models in the traditional SEM framework, with general (possibly non-conjugate) prior distributions. It has been implemented in **blavaan** alongside the previous **JAGS** and Stan implementations, allowing for easy comparison across implementations.

In the sections below, we first formally define the models under consideration. We then describe the three MCMC approaches that are now implemented in **blavaan**: the original **JAGS** approach described in Merkle and Rosseel (2018), the original Stan approach that is being formally described in this paper for the first time, and the new Stan approach that is the primary focus of this paper. After describing the approaches, we explicitly discuss some problematic issues associated with estimation of SEMs via MCMC. These are issues that are often overlooked in the literature, but they are necessary to have fully functional Bayesian SEM software. Finally, we compare the three approaches via three examples, highlighting the advantages of the new Stan approach.

2. Model definition

The **blavaan** package generally relies on the **lavaan** representation of a structural equation model, which is generally based on the **LISREL** "all-y" representation (e.g., Jöreskog and Sörbom 1997).

Let $\boldsymbol{y}_i$ be the p (continuous) observed variables associated with observation i. Then a structural equation model with m latent variables may be represented by the equations

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\eta}_i + \boldsymbol{\epsilon}_i \\ \boldsymbol{\eta}_i &= \boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{\eta}_i + \boldsymbol{\zeta}_i, \end{aligned} \tag{1}$$

where $\boldsymbol{\eta}_i$ is an $m \times 1$ vector containing the latent variables; $\boldsymbol{\epsilon}_i$ is a $p \times 1$ vector of measurement errors; and $\boldsymbol{\zeta}_i$ is an $m \times 1$ vector of structural errors. The vectors $\boldsymbol{\nu}$ and $\boldsymbol{\alpha}$ contain intercept parameters for the manifest and latent variables, respectively; $\boldsymbol{\Lambda}$ is a matrix of factor loadings; and $\boldsymbol{B}$ contains parameters that reflect regression relationships between latent variables.

The residuals $\boldsymbol{\epsilon}_i$ and $\boldsymbol{\zeta}_i$ are assumed to be multivariate normal:

$$\begin{aligned} \boldsymbol{\epsilon}_i &\sim N_p(\boldsymbol{0}, \boldsymbol{\Theta}) \\ \boldsymbol{\zeta}_i &\sim N_m(\boldsymbol{0}, \boldsymbol{\Psi}), \end{aligned}$$

where the associated covariance matrices are often diagonal. These assumptions imply that the marginal distribution of $\boldsymbol{y}$ (integrating out the latent variables) is multivariate normal with parameters

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\alpha} \\ \boldsymbol{\Sigma} &= \boldsymbol{\Lambda}(\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Psi}(\boldsymbol{I} - \boldsymbol{B}^\top)^{-1}\boldsymbol{\Lambda}^\top + \boldsymbol{\Theta}, \end{aligned}$$

which requires that $(\boldsymbol{I} - \boldsymbol{B})$ be invertible. The traditional **LISREL** framework includes additional matrices for exogenous observed variables, but these are not often utilized in **lavaan**. Instead, an exogenous observed variable is more commonly "upgraded" to latent variable status, where the latent variable accounts for all of the observed variable's variance (and the associated variance parameter in $\boldsymbol{\Theta}$ is fixed to 0).

Many Bayesian approaches to SEM estimation rely on sampling the $\boldsymbol{\eta}_i$ in tandem with other model parameters. This is advantageous because observed variables are often independent conditioned on the $\boldsymbol{\eta}_i$, so that the conditional distribution of each observed variable is a univariate normal. However, as we will see later, the sampling of the $\boldsymbol{\eta}_i$ can have a major impact on the speed and efficiency of MCMC estimation.

3. MCMC approaches

In the sections below, we briefly describe the three MCMC approaches implemented in **blavaan**: the original **JAGS** approach, the original Stan approach, and the new Stan approach that is the focus of this paper.

3.1. Parameter expansion in JAGS

In previous work (Merkle and Rosseel 2018), we developed a parameter expansion approach that can be applied to SEMs for continuous data (also see Palomo, Dunson, and Bollen 2007). The method allows us to often use univariate normal distributions in place of multivariate normal distributions, and it can be generally implemented in **JAGS**.

The approach involves converting the model of interest (the "inferential model") to an overparameterized, equivalent model (the "working model") from which it is easier to sample. The conversion focuses on a model's covariance parameters, converting each covariance to a "phantom" latent variable. This conversion makes observed variables conditionally independent of one another (conditioned on latent variables), meaning that our likelihood involves a series of univariate distributions instead of a single multivariate distribution. Such a conversion can speed up sampling in **JAGS**, where computations involving the multivariate normal distribution are very slow. The full details underlying these procedures can be found in Merkle and Rosseel (2018).

3.2. Likelihood simplification in Stan

The phantom latent variable approach is not essential in Stan and, in testing, we found that the approach did not lead to gains in sampling speed or efficiency. However, we did make initial progress in Stan by capitalizing on the structure of the SEM latent variable covariance matrix $\boldsymbol{\Psi}$. This capitalization was inspired by related work on estimating multivariate autoregressive models in Stan (Joseph 2016).

We provide an overview of this approach here. For traditional SEMs, the distribution of latent variables can typically be expressed as shown in Equation 1:

$$\boldsymbol{\eta} \sim N((\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\alpha}, (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Psi}(\boldsymbol{I} - \boldsymbol{B}^{\top})^{-1}).$$

Evaluation of this multivariate normal log-likelihood is time-consuming in Stan because we need to compute the inverse and determinant of the covariance matrix. However, for many models, the structure of the covariance matrix leads to simplifications. For example, the matrix $\boldsymbol{B}$ is often triangular with zeros along its diagonal (leading to what some call a "recursive" model), and the matrix $\boldsymbol{\Psi}$ is often diagonal. When both of these properties are fulfilled, we can use standard matrix properties (e.g., Petersen and Pedersen 2012) to write the determinant as a product of scalar values:

$$\begin{aligned}\det((\boldsymbol{I}-\boldsymbol{B})^{-1}\boldsymbol{\Psi}(\boldsymbol{I}-\boldsymbol{B}^\top)^{-1}) &= (\det(\boldsymbol{I}-\boldsymbol{B}))^{-1}\det(\boldsymbol{\Psi})(\det(\boldsymbol{I}-\boldsymbol{B}))^{-1}\\ &= 1\cdot\prod_{i=1}^{m}\psi_{ii}\cdot 1.\end{aligned}$$

Relatedly, the inverse of $\boldsymbol{\Sigma}$ is simplified as

$$((\boldsymbol{I}-\boldsymbol{B})^{-1}\boldsymbol{\Psi}(\boldsymbol{I}-\boldsymbol{B}^\top)^{-1})^{-1} = (\boldsymbol{I}-\boldsymbol{B}^\top)\boldsymbol{\Psi}^{-1}(\boldsymbol{I}-\boldsymbol{B}),$$

which completely removes the need to compute matrix inversions when $\boldsymbol{\Psi}$ is diagonal. When either $\boldsymbol{B}$ is triangular or $\boldsymbol{\Psi}$ is diagonal (but not both), we can use a subset of the above simplifications to improve sampling efficiency as much as possible. Given a specific model, **blavaan** automatically determines which simplifications are available and uses them for Stan estimation. The simplifications are implemented in Stan as a custom log-probability density function. This approach is available in **blavaan** via the argument `target = "stanclassic"`.

3.3. New Stan approach

Both methods mentioned above exploit the fact that the latent variables in the model can be sampled along with other model parameters. This generally simplifies likelihood computations and allows us to immediately extend the methods to situations where observed variables have non-normal distributions. Most Bayesian approaches to SEM, and to other models with "random" parameters, sample the latent variables.

However, the sampling of latent variables greatly increases the dimension of the parameter space, which can reduce sampling speed and efficiency. Thus, the key to fast sampling in Stan is to work with a model likelihood that is marginal over latent variables, a result that was also recently discussed by Hecht, Gische, Vogel, and Zitzmann (2020). This is somewhat unintuitive, because previous researchers have focused on the simplifications that we can gain from sampling the latent variables. One concern related to using the marginal likelihood involves our inability to make inferences about the latent variables (because they are integrated out of the likelihood). But this concern is addressed by **blavaan** because, conditional on the other model parameters, the latent variable posterior distribution is tractable. Thus, the latent variables can be sampled in a "generated quantities" block of the Stan syntax, even though they do not directly play a role in the MCMC sampling.

The new **blavaan** approach utilizes the marginal likelihood, and we have written a single Stan program that can estimate the majority of multivariate normal SEMs that a user could specify. This file is compiled once during (or before) package installation. Then, once the user specifies a model, many pieces of information about the data and about the model are passed to the compiled model, with sampling occurring immediately. To complement the Stan file, we have new R code that serves as a pipeline from **blavaan** to the Stan model and back. The **blavaan** user will not notice many differences, because the commands for model specification and estimation are the same as before. However, the model is now sent to the pre-compiled Stan model by default, whereas the previous approaches wrote **JAGS** or Stan code at runtime.

It is worth noting that our Stan SEM file stands on its own, so that users of languages beyond R (e.g., Python) could also utilize the file if they can pass all the required data in to the Stan model. This is more challenging than it may sound due to the many pieces of data that are required, including the dimensions of all SEM matrices, the free entries of SEM matrices, equality constraints on free parameters, prior distribution parameters, and so on.

Because our Stan model is precompiled, the possible models that can be estimated are restricted in two ways. First, there is some inflexibility in choice of prior distributions. For most types of model parameters, the form of each parameter's prior cannot be changed (though the specific prior hyperparameters can). For example, regression parameters ($\boldsymbol{B}$) in **blavaan** currently have $\text{N}(0, 10)$ priors by default, where the normal distribution is parameterized by standard deviation. Users can change the mean or standard deviation of this normal prior, but they cannot change the fact that the prior is normal. However, for scale parameters, users have the option to place priors on variances, standard deviations, or precisions.

The second restriction of the precompiled Stan model involves equality constraints. While our code currently allows for equality constraints within a class of parameters (e.g., loadings can be constrained equal to one another or intercepts can be constrained equal to one another), it does not allow for constraints across classes of parameters. Additionally, if users wish to set one parameter equal to a function of other parameters, that is not currently possible. However, if users desire features that are not included in our current implementation, they can export our Stan file (via argument `mcmcfile=TRUE`), make the desired changes, and recompile the model. Alternatively, they could use the other MCMC methods available in **blavaan**, which provide more flexibility because they are not precompiled.

4. Challenging issues

While the above methods can be readily applied to "vanilla" models such as confirmatory factor analysis with uncorrelated factors, the SEM framework includes many model and data characteristics that require further attention for estimation. Below, we highlight three characteristics that are not always addressed in texts on Bayesian SEM.

4.1. Covariance parameters

The general SEM presented earlier includes two covariance matrices with free parameters: $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$. These matrices can sometimes include some fixed values and some free values, so prior distributions for these matrices are not straightforward. That is, there are some models for which we cannot simply place an inverse Wishart prior on the covariance matrix, nor an LKJ prior (Lewandowski, Kurowicka, and Joe 2009). Those priors were meant for unrestricted covariance/correlation matrices, not for matrices with some fixed values and some free values.

In the Stan approaches, we consequently decompose the covariance matrices into standard deviations and correlations. For example, $\boldsymbol{\Theta}$ is written as

$$\boldsymbol{\Theta} = \boldsymbol{D}_{\Theta}\boldsymbol{R}_{\Theta}\boldsymbol{D}_{\Theta},$$

where $\boldsymbol{D}_{\Theta}$ is a diagonal matrix of standard deviations and $\boldsymbol{R}_{\Theta}$ is a correlation matrix. Prior distributions are then placed individually on the free standard deviation parameters and on the free correlation parameters within the two matrices. This approach is similar to that of Barnard, McCulloch, and Meng (2000), and Liu, Zhang, and Grimm (2016) provide a comparison of this approach to the use of inverse Wisharts in the context of growth curve models.

The use of an independent prior on each free parameter can sometimes lead to a non-positive definite covariance matrix during MCMC sampling. Stan is able to reject such a covariance matrix and continue sampling, whereas **JAGS** will terminate. This is why we developed the parameter expansion method in **JAGS**: the parameter-expanded model involves diagonal covariance matrices that cannot become non-positive definite. The non-positive definite covariance matrices have implications for model calibration, however, and we detail this in the simulation-based calibration study later.

We are aware of a variety of other prior distributions proposed for covariance matrices (e.g., Chung, Gelman, Rabe-Hesketh, Liu, and Dorie 2015; Consonni and Veronese 2003; Mulder and Pericchi 2018; Spezia 2019). The strategy implemented in **blavaan** is worthwhile because it is relatively easy to specify informative prior distributions for individual standard deviation and correlation parameters in the model. In contrast, many of the other prior distributions are proposed for convenience or due to the fact that they maintain positive definiteness, and they have less-intuitive interpretations as compared to our approach. But we plan to further consider these alternative priors in the future.

4.2. Missing data

While it is often useful and desirable to directly model the missing values with the rest of the model (e.g., Merkle 2011; O'Muircheartaigh and Moustaki 1999), **blavaan** employs a "missing at random" approach to missing data that differs across **JAGS** and Stan. In **JAGS**, one can include NA values in the data, and **JAGS** will sample these missing values as if they were extra model parameters. In contrast, Stan does not allow NA values in the data, so that one must handle the missing data manually. We utilize a "full information" likelihood (e.g., Wothke 2000) in our Stan models, which is the same likelihood that is used to handle missing data in **lavaan** and other software that performs maximum likelihood SEM estimation. This requires some additional overhead in preparing the data to be sent to Stan, because each case's observed values must be indexed, and cases are sorted by missing data pattern to speed up computations. Missing values could also be directly sampled ("imputed") in Stan, though this functionality is not currently available.

4.3. Latent variable scaling

Structural equation models typically require some parameter constraints to achieve parameter identification, where we must "set the scale" of each latent variable. The two most popular ways to do this involve (i) fixing each latent variable's variance to one, or (ii) fixing a single loading (parameter in $\boldsymbol{\Lambda}$) to one for each latent variable. Of these two, the latter method is most straightforward to implement in a Bayesian setting.

The former method (of fixing each latent variance to one) is more challenging. This is because, as described by Peeters (2012), one loading per latent variable must be sign constrained to achieve global parameter identification. Otherwise, the sign of each loading may flip back and forth, with a model's regression parameters and covariance parameters potentially flipping along with the loadings. One solution to this issue involves the placement of a truncated normal prior (truncated from below at 0) on one loading per latent variable, preventing the sign changes. This solution is adopted in **blavaan**'s **JAGS** approach to model estimation.

A different solution is implemented in **blavaan**'s Stan approaches. In those approaches, the sign flipping is allowed to occur during MCMC sampling. The issue is then handled after sampling, in the "generated quantities" block. In this block, one loading per latent variable is transformed to always be positive, and the signs of associated parameters (loadings, regressions, and covariance parameters) are flipped every time the sampled value of the focal loading is negative. This approach can improve sampling efficiency because no boundary constraints are introduced in the parameter space. Further information about this approach can be found in a thread on the Stan Discourse site (`https://discourse.mc-stan.org/t/latent-factor-loadings/1483`).

5. Applications

As stated earlier, the MCMC approaches described here are all implemented in package **blavaan** for general SEM estimation. These include the original **JAGS** approach (obtained via argument `target = "jags"`), the original Stan approach (`target = "stanclassic"`), and the new Stan approach (`target = "stan"`). Common model estimation functions in **blavaan** include `bcfa()` and `bsem()`, both of which call the more general `blavaan()` function with prespecified argument settings. The commands and settings mimic those of **lavaan**, except that the model is estimated via Bayesian methods instead of frequentist methods.

As a concrete example, the following code specifies a model for the well-known "political democracy" data (Bollen 1989) and estimates it via each of the three approaches described earlier. This dataset includes 75 countries measured on 11 attributes, seven of which were measured in 1960 and four of which were measured in 1965. The intent of the model is to study relationships between countries' levels of industrialization and democracy over time.

```
R> model <- '
+    # measurement model
+      ind60 =~ x1 + x2 + x3
+      dem60 =~ y1 + y2 + y3 + y4
+      dem65 =~ y5 + y6 + y7 + y8
+    # regressions
+      dem60 ~ ind60
+      dem65 ~ ind60 + dem60
+    # residual correlations
+      y1 ~~ y5
+      y2 ~~ y4 + y6
+      y3 ~~ y7
+      y4 ~~ y8
+      y6 ~~ y8
+  '
R> fit1 <- bsem(model, data = PoliticalDemocracy, target = "jags")
R> fit2 <- bsem(model, data = PoliticalDemocracy, target = "stanclassic")
R> fit3 <- bsem(model, data = PoliticalDemocracy, target = "stan")
```

The above commands use the default number of burnin/warmup and sampling iterations, as well as the package's default prior distribution for each type of model parameter. The default prior distributions have generally been chosen to be weakly informative for a variety of SEMs typically encountered in practice, but we encourage users to consider their own prior distributions for their own applications. Some further discussion of prior distributions appears later in the simulation-based calibration section.

Following model estimation, convergence diagnostics such as `Rhat` and effective sample size are immediately available via the `summary()` method and `blavInspect()` function, and many types of plots are available via the `plot()` method, which relies on package **bayesplot** (Gabry and Mahr 2021). Further examples of **blavaan** syntax and functionality can be found in Merkle and Rosseel (2018), noting that `target="jags"` was the default at the time that paper was written, while `target="stan"` is now the default.

In the following sections, we compare the three MCMC approaches on speed and sampling efficiency via three example models. These comparisons are platform dependent, require some arbitrary decisions to be carried out, and exhibit variability from run to run. But the comparisons uniformly show that the new Stan approach is best, and we encourage users to try out other models to see for themselves. After reporting on these comparisons, we study the extent to which the new Stan approach is calibrated.

5.1. Performance comparison

All comparisons were carried out on a Dell desktop with a large amount of RAM, running Ubuntu Linux. We define sampling efficiency as "effective sample size per second" (ESS/s), where effective sample sizes are computed via the **rstan** `monitor()` function and sampling time is measured after Stan model compilation. The warmup time for Stan models was fixed to 300 iterations, whereas the burn-in time for **JAGS** models was fixed to 1000 iterations. There is clear arbitrariness in the warmup and burn-in choices here, so that the ESS/s metric is somewhat crude.

We examine the MCMC methods' speed and efficiency on three models, two of which are popular models often used to illustrate SEM methods. The third is a more complex model that is known to pose difficulties for the original **blavaan** approach. We do *not* conduct a full Monte Carlo study here so, as mentioned earlier, our results are subject to noise. But the results are generally consistent across the models presented here as well as many others not presented, so we think they can be taken as general evidence for the approaches' relative performance. The results are also consistent with those of Yackulic, Dodrill, Dzul, Sanderlin, and Reid (2020), who study marginalization of discrete latent variables in ecological models.

Political democracy

For our first comparison, we continue with the Bollen (1989) political democracy model. The **blavaan** code to fit the model was shown earlier. The **JAGS** method was fastest here, averaging 0.55 seconds per 100 iterations. Next fastest was the new (marginal) Stan method, averaging 1.44 seconds per 100 iterations, followed by the old Stan method at 7.36 seconds per 100 iterations. But it is more important to examine the methods' sampling efficiencies (effective sample size per second), which are shown in Figure 1. A separate metric is shown for each parameter, with the parameters being numbered along the x-axis. Parameters are ordered on the x-axis by parameter type, with the details being shown in the figure caption. The figure shows that the speed of **JAGS** is offset by the effective sample size, so that the new Stan method is best in terms of sampling efficiency. The old Stan method exhibits efficiency similar to that of **JAGS**, though the old Stan method was also prone to some divergent transitions.

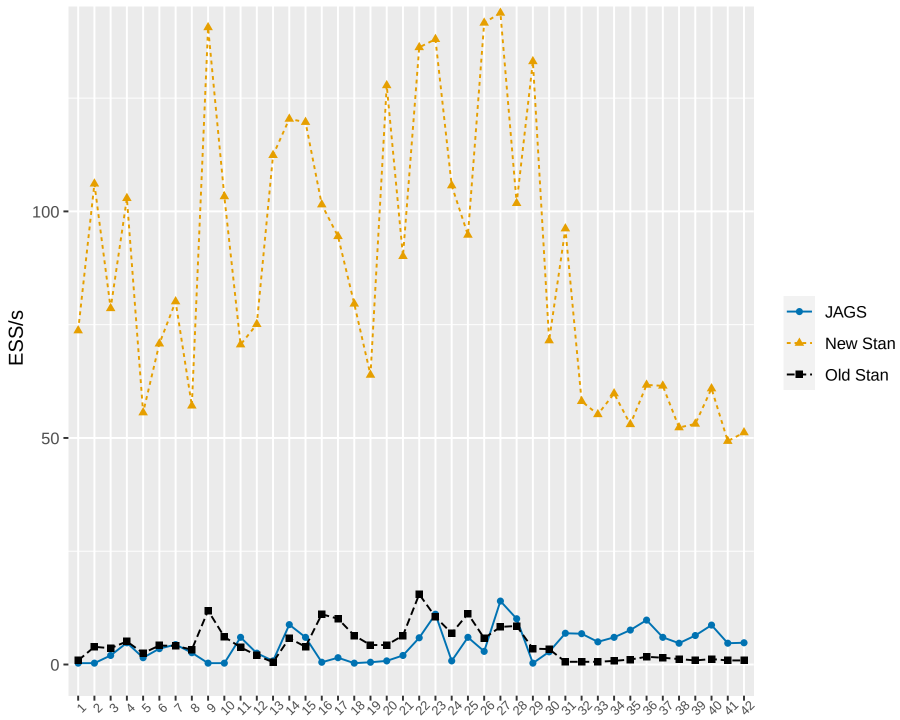

Figure 1: Sampling efficiency of the MCMC procedures for the Bollen example. Parameter numbers correspond to different types of parameters: loadings are 1–8; regressions are 9–11; observed variances are 12–28; latent variances are 29–31; intercepts are 32–42.

Holzinger and Swineford

Our second example involves a confirmatory factor analysis of the Holzinger and Swineford (1939) data. This is the version of the dataset included in package **lavaan**, which has 301 individuals measured on nine cognitive scales. The confirmatory factor model fit to the data includes three latent variables, each of which is associated with three observed variables. The **blavaan** code to specify and fit the model is

```
R> HS.model <- ' visual  =~ x1 + x2 + x3
+                textual =~ x4 + x5 + x6
+                speed   =~ x7 + x8 + x9 '
R> fit <- bcfa(HS.model, data = HolzingerSwineford1939)
```

where additional arguments would typically be used to specify the number of sampling iterations, to specify priors, to specify the MCMC sampler, and so on.

In terms of speed, the **JAGS** method is again fastest, averaging 0.91 seconds per 100 iterations. This was followed by the marginal Stan method at 2.83 seconds per 100 iterations, then the old Stan method at 9.55 seconds per 100 iterations. The ESS/s metrics for this model are visualized in Figure 2. The graph is similar to that of the previous section, with fewer parameters in this model as compared to the last model. We see that the gold line, representing the new Stan method, is the highest for all the model parameters, being at least twice as large as the other methods' efficiencies. The **JAGS** and old Stan methods are again similar to one another for this example, with **JAGS** being better for the majority of parameters.

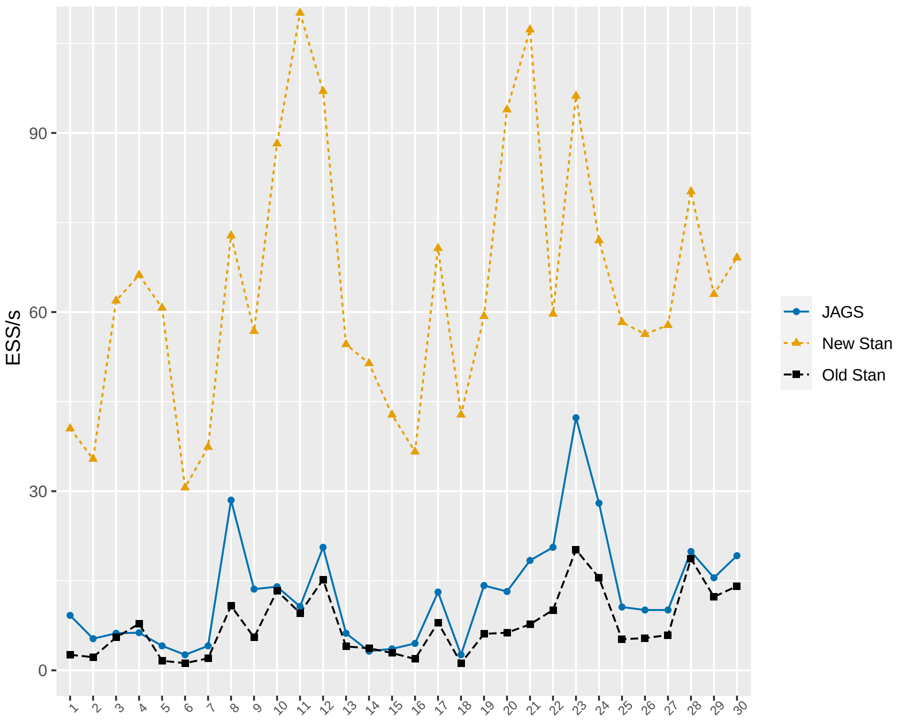

Figure 2: Sampling efficiency of the MCMC procedures for the Holzinger and Swineford example. Parameter numbers correspond to different types of parameters: loadings are 1–6; observed variable variances are 7–15; latent variable variances are 16–18; latent variable covariances are 19–21; intercepts are 22–30.

Growth model

For our final comparison, we use a "multiple indicator univariate latent change score" model presented in Kievit *et al.* (2018). The **blavaan** code to fit this model, as specified by Kievit *et al.* (2018), is

```
R> MILCS <- '
+    COG_T1 =~ 1*T1X1 + T1X2 + T1X3
+    COG_T2 =~ 1*T2X1 + equal("COG_T1 =~ T1X2")*T2X2 +
+      equal("COG_T1 =~ T1X3")*T2X3
+
+    COG_T2 ~ 1*COG_T1
+    dCOG1 =~ 1*COG_T2
+    COG_T2 ~ 0*1
+    COG_T2 ~~ 0*COG_T2
+
+    dCOG1 ~ 1
+    COG_T1 ~ 1
+    dCOG1 ~~ dCOG1
+    COG_T1 ~~ COG_T1
+    dCOG1 ~ COG_T1
+
```

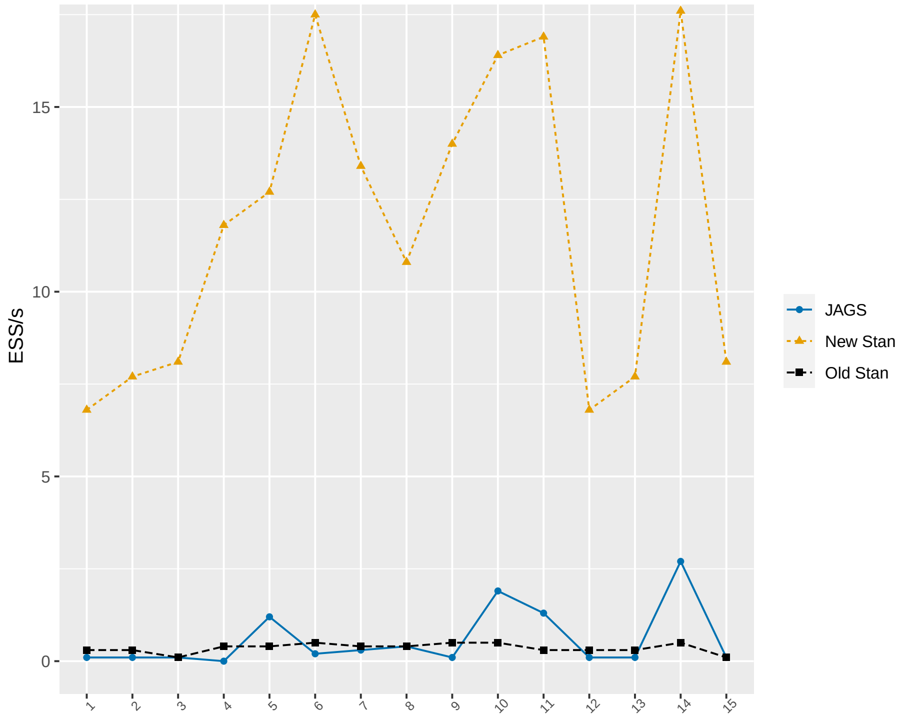

Figure 3: Sampling efficiency of the MCMC procedures for the growth model example. Parameter numbers correspond to different types of parameters: loadings are 1–2; regressions are 3; observed variances are 4–6; observed variable covariances are 7–9; latent variances are 10–11; intercepts are 12–13; latent means are 14–15.

```
+    T1X1 ~~ T2X1
+    T1X2 ~~ T2X2
+    T1X3 ~~ T2X3
+
+    T1X1 ~~ T1X1
+    T1X2 ~~ T1X2
+    T1X3 ~~ T1X3
+
+    T2X1 ~~ equal("T1X1 ~~ T1X1")*T2X1
+    T2X2 ~~ equal("T1X2 ~~ T1X2")*T2X2
+    T2X3 ~~ equal("T1X3 ~~ T1X3")*T2X3
+
+    T1X1 ~ 0*1
+    T1X2 ~ 1
+    T1X3 ~ 1
+    T2X1 ~ 0*1
+    T2X2 ~ equal("T1X2 ~ 1")*1
+    T2X3 ~ equal("T1X3 ~ 1")*1
+   '
R> fit <- blavaan(MILCS, data = simdatMILCS, fixed.x = FALSE)
```

where we fit the model to 500 artificial observations. Further information about this model and its specification can be found in Kievit *et al.* (2018). This model, and others described in Kievit *et al.* (2018), have been especially difficult to fit in **blavaan**, requiring long run times and exhibiting high autocorrelation among parameter draws. We ended up thinning the **JAGS** samples by 20 in our analyses here, because it was the only way that we could consistently obtain an `Rhat` value below 1.05.

The sampling speed is now reversed for this example, with the marginal Stan method at 23.4 seconds per 100 iterations, the **JAGS** method at 28.4 seconds per 100 iterations, and the old Stan method at 577.27 seconds per 100 iterations. If we instead compute the **JAGS** speed while accounting for thinning (i.e., counting only each twentieth iteration in the computations), then the **JAGS** speed is at 567.98 seconds per 100 iterations.

The approaches' sampling efficiencies are shown in Figure 3, where parameter ordering is again described in the figure caption. The **JAGS** and old Stan methods are very low for this model, with the new Stan method displaying much better efficiency and yielding useful results in a matter of minutes, as opposed to hours or days. This and related examples (not shown) convinced us to make the new Stan method the default in **blavaan**, replacing the original default of **JAGS**. The new Stan method reliably produces fast, efficient samples for a large number of models, whereas the other methods exhibit more variability in their speeds and efficiencies, and are seldom clearly better than the new Stan method.

5.2. Verification and simulation-based calibration

The posterior estimates resulting from **blavaan** have been verified in a few manners. Initially, we treated **lavaan** as a gold standard, comparing posterior means and standard deviations under weak priors to the maximum likelihood estimates and standard errors from **lavaan**. For all three MCMC methods, the posterior means obtained under this approach typically agree with **lavaan** estimates up to the tenth digit. The posterior standard deviations tend to be close to, but slightly larger than, the maximum likelihood standard errors. But now that there are multiple MCMC methods implemented in **blavaan**, we have also been able to compare MCMC methods to one another in order to verify that they were producing similar posterior distributions.

Here, we use the simulation-based calibration method proposed by Talts, Betancourt, Simpson, Vehtari, and Gelman (2018) to study the calibration of **blavaan**'s new (marginal) Stan implementation. This involves repeatedly generating data from the model's prior distribution, fitting the model to the generated data, and examining the ranks of the posterior MCMC samples relative to samples from the prior distribution. If the MCMC algorithm is calibrated, then these ranks should be approximately uniformly distributed. Deviations from uniformity are taken as miscalibration.

Method

Our simulation-based calibration study utilized the political democracy model presented earlier. We generated 500 datasets of size 75 from the prior predictive distribution and fit the model to each generated dataset via MCMC. The study involved two conditions that differed by the prior distributions that were used. First, we used the default prior distributions from **blavaan**, which are intended to be weakly informative in many situations encountered in practice. Second, we used a set of more informative prior distributions to contrast with the noninformative priors. Both sets of prior distributions are shown in Table 1.

	ν	λ	β	θ	ψ	ρ
Set 1 (default)	N(0,32)	N(0,10)	N(0,10)	Gamma(1,.5)	Gamma(1,.5)	Beta(1,1)
Set 2 (informative)	N(0,32)	N(1.25,.25)	N(1.5,.25)	Gamma(10,10)	Gamma(10,10)	Beta(5,5)

Table 1: Prior distributions used in the simulation-based calibration study. Normal priors are parameterized using standard deviations. Gamma priors are placed on standard deviations associated with θ and ψ parameters, as opposed to variances or precisions.

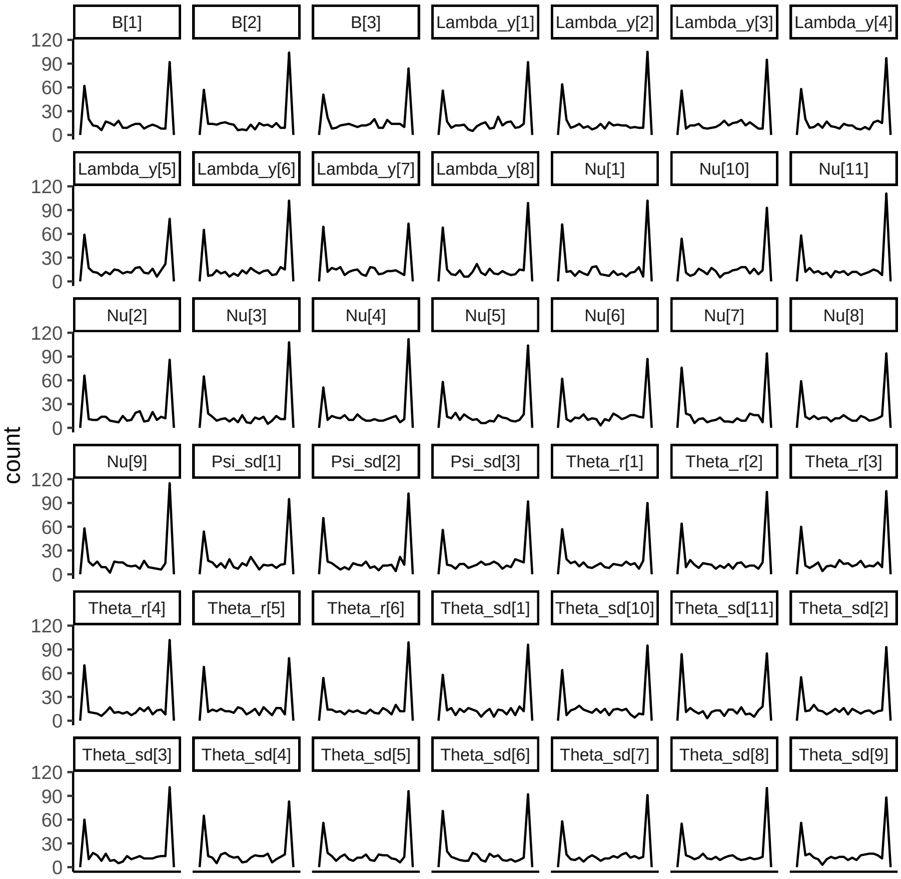

Figure 4: Simulation-based calibration rank frequencies, default **blavaan** priors.

Results

Rank frequencies for the **blavaan** default priors are shown in Figure 4. Perhaps surprisingly, the distributions are far from uniform, with peaks generally occurring near zero and one. These peaks represent posterior distributions that exhibit less variability than they should, given the non-informative priors from which we started. Clearly, the results are far from the uniformity that would be expected from a calibrated algorithm.

The non-uniformity occurs because our model has a large number of parameters with independent prior distributions, with many parameters playing a role in a model covariance matrix that has some covariances fixed to zero and other covariances that are free. For the model considered here, there are many combinations of parameters that lead to a non-positive definite covariance matrix here, and the MCMC sampler will avoid these combinations of pa-

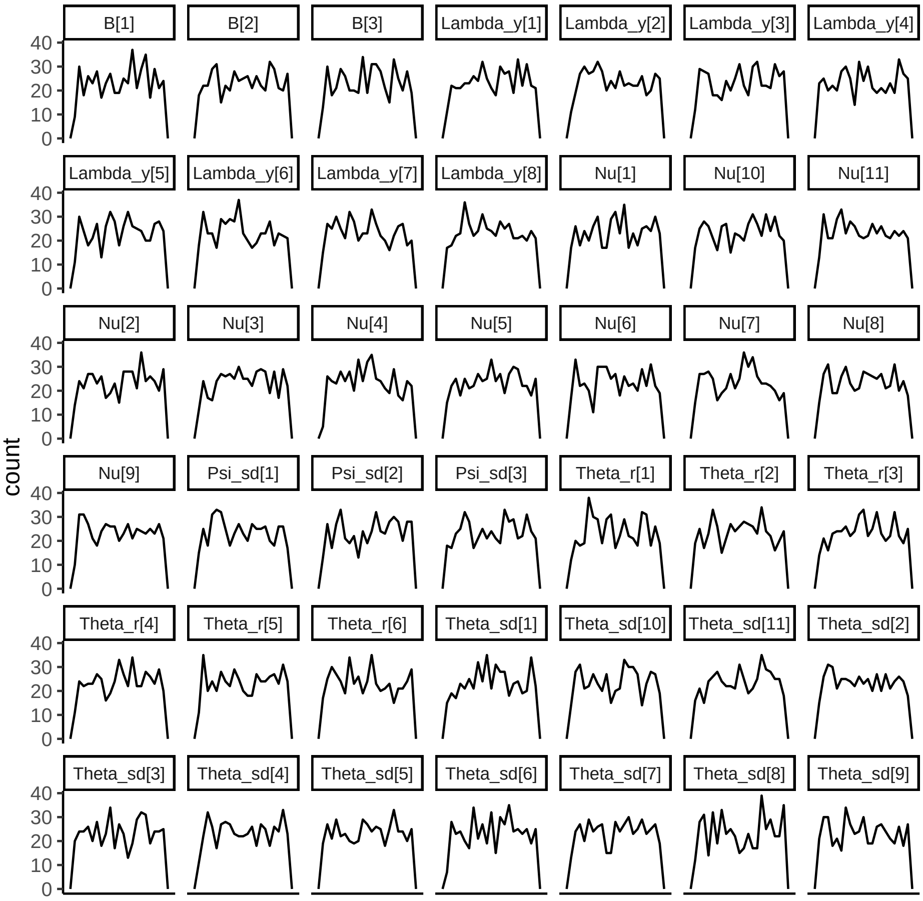

Figure 5: Simulation-based calibration rank frequencies, informative priors.

rameters during sampling. In the simulation-based calibration study, this leads to posterior samples that are not calibrated with respect to the independent priors. Instead, we might say that the posteriors are calibrated with respect to regions of the prior distribution that are positive definite.

To provide evidence that the MCMC algorithm is indeed calibrated with respect to priors that maintain positive definiteness, we show the results of the informative priors in Figure 5. These frequencies are now much closer to uniform, because the information in the prior distributions now generally leads to positive definite model covariance matrices. This interplay between the informativeness of prior distributions and posterior calibration is worthy of further attention, because existing MCMC algorithms for SEM use a series of independent priors on parameters that each play a role in the model-implied covariance matrix. A researcher's priors can be more informative than expected, based solely on the fact that the model-implied covariance matrix must remain positive definite during MCMC sampling. Further, depending on the model likelihood used (marginal vs. conditional), the degree of information present in uninformative priors may vary. These results highlight the utility of **blavaan** for conducting detailed study of MCMC algorithms, as well as the fact that there is room to improve the default **blavaan** priors in the future.

6. Conclusion

The results in this paper show that we can improve sampling efficiency by integrating the latent variables out of the model likelihood, which is the opposite of most popular approaches to Bayesian SEM estimation (where the popular approaches are largely based on results summarized by, e.g., Lee 2007; Song and Lee 2012). We can expect the marginal sampling efficiency to be even more advantageous as sample sizes increase, because the sample size has no impact on the dimension of the parameter space here. In contrast, the dimension of the parameter space increases with sample size under conditional approaches, where latent variables count as parameters.

While the marginal approach is promising, use of the marginal likelihood leads us back to problems that frequentists often encounter in SEM. These problems include the fact that the marginal likelihood does not have a closed form when we have non-normal observed variables (e.g., ordinal variables) or when we have latent variable interactions. We think that some progress can be made here by employing other Bayesian methods, including data augmentation (e.g., Chib and Greenberg 1998) in the ordinal case. The use of data augmentation for psychometric models has been described by Fox (2010) and Fox, Mulder, and Sinharay (2017), and such methods may be implemented in future versions of **blavaan**.

For situations where the marginal likelihood does not exist in closed form, it is also possible to move back to the original **blavaan** approaches that sample the latent variables. However, in our experience, the original approaches are even slower and less efficient in those situations (as compared to the models considered here), making them questionable for applied work. Further, even if those methods did exhibit reasonable efficiency, the marginal likelihood is generally necessary for obtaining suitable information criteria such as DIC (Spiegelhalter, Best, Carlin, and Linde 2002) or WAIC (Watanabe 2010). Merkle, Furr, and Rabe-Hesketh (2019) discuss why the marginal likelihood is preferable here, and Zhang, Tao, Wang, and Shi (2019) discuss related applications of DIC to multilevel item response models. Thus, we think that use of the new Stan approach, paired with new tricks for handling non-normal observed variables, is the most promising approach for applications to the non-normal modeling situations typically encountered in practice.

We should note that the three methods here are not the only ones that can be conceptualized in **JAGS** or in Stan. For example, the pre-compiled marginal Stan approach can be modified so that the latent variables are part of the model likelihood. This leads to a method that is similar to the "old Stan" method, except that it simplifies computation of the model likelihood in different ways. In limited testing, we found that this approach was somewhat closer to the **JAGS** approach in sampling efficiency, but still similar to the "old Stan" approach studied in this paper. Alternatively, it is possible to define a marginal approach in **JAGS**, but our limited testing there indicates that use of the multivariate normal distribution leads to decreases in **JAGS** sampling efficiency. For some models, it is also possible to fit the model to the sample covariance matrix instead of to the raw data (e.g., Choi and Levy 2017). This can improve sampling efficiency because we only have to compute a single likelihood for the covariance matrix, whereas we must compute a separate likelihood for each individual in the raw data. However, SEM includes many models where the sample covariance matrix is not a sufficient statistic, so this approach cannot always be applied.

Finally, our analyses here have included no use of parallelization. While between-chain parallelization is immediately available in **blavaan**, there have been recent advances in within-chain parallelization in Stan. These advances are promising for further improvement of Stan sampling efficiency, and we plan to include this in future versions of **blavaan**.

Computational details

The results in this paper were obtained with R 4.1.2, **g++** compiler version 9.3.0 for C++, **rstan** 2.21.2 with **StanHeaders** 2.21.0-7 on an Ubuntu 20.04.3 system. The replication code with random seeds is provided, but exact replication requires the same operating system, same compiler, same compiler version, and same hardware (for timings). We have run the code on other systems and obtained similar results that lead to the same qualitative conclusions as were reported in the paper.

Acknowledgments

This work was partially supported by a research leave to the first author, which was provided by the University of Missouri.

References

Barnard J, McCulloch R, Meng XL (2000). "Modeling Covariance Matrices in Terms of Standard Deviations and Correlations, with Application to Shrinkage." *Statistica Sinica*, **10**, 1281–1311.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using **lme4**." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Bentler PM (2008). ***EQS** 6 Structural Equations Program Manual.* Multivariate Software, Inc.

Boker S, Neale M, Maes H, Wilde M, Spiegel M, Brick T, Spies J, Estabrook R, Kenny S, Bates T, Mehta P, Fox J (2011). "OpenMx: An Open Source Extended Structural Equation Modeling Framework." *Psychometrika*, **76**, 306–317. doi:10.1007/s11336-010-9200-6.

Bollen KA (1989). *Structural Equations with Latent Variables.* John Wiley & Sons, New York.

Bürkner PC (2017). "**brms**: An R Package for Bayesian Multilevel Models Using Stan." *Journal of Statistical Software*, **80**(1), 1–28. doi:10.18637/jss.v080.i01.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*, **76**(1), 1–32. doi:10.18637/jss.v076.i01.

Chib S, Greenberg E (1998). "Analysis of Multivariate Probit Models." *Biometrika*, **85**, 347–361. doi:10.1093/biomet/85.2.347.

Choi J, Levy R (2017). "Markov Chain Monte Carlo Estimation Methods for Structural Equation Modeling: A Comparison of Subject-level Data and Moment-level Data Approaches." *Biometrics & Biostatistics International Journal*, **6**, 463–474. doi:10.15406/bbij.2017.06.00182.

Chung Y, Gelman A, Rabe-Hesketh S, Liu J, Dorie V (2015). "Weakly Informative Prior for Point Estimation of Covariance Matrices in Hierarchical Models." *Journal of Educational and Behavioral Statistics*, **40**(2), 136–157. doi:10.3102/1076998615570945.

Consonni G, Veronese P (2003). "Enriched Conjugate and Reference Priors for the Wishart Family on Symmetric Cones." *The Annals of Statistics*, **31**(5), 1491–1516. doi:10.1214/aos/1065705116.

Denwood MJ (2016). "**runjags**: An R Package Providing Interface Utilities, Model Templates, Parallel Computing Methods and Additional Distributions for MCMC Models in **JAGS**." *Journal of Statistical Software*, **71**, 1–25. doi:10.18637/jss.v071.i09.

Depaoli S, van de Schoot R (2017). "Improving Transparency and Replication in Bayesian Statistics: The WAMBS-Checklist." *Psychological Methods*, **22**, 240–261. doi:10.1037/met0000065.

Driver CC, Oud JHL, Voelkle MC (2017). "Continuous Time Structural Equation Modeling with R Package **ctsem**." *Journal of Statistical Software*, **77**(5), 1–35. doi:10.18637/jss.v077.i05.

Fox J, Nie Z, Byrnes J (2021). ***sem**: Structural Equation Models*. R package version 3.1-13, URL https://CRAN.R-project.org/package=sem.

Fox JP (2010). *Bayesian Item Response Modeling: Theory and Applications*. Springer-Verlag, New York.

Fox JP, Mulder J, Sinharay S (2017). "Bayes Factor Covariance Testing in Item Response Models." *Psychometrika*, **82**(4), 979–1006. doi:10.1007/s11336-017-9577-6.

Furr DC (2017). ***edstan**: Stan Models for Item Response Theory*. R package version 1.0.6, URL https://CRAN.R-project.org/package=edstan.

Gabry J, Goodrich B (2020). ***rstanarm**: Bayesian Applied Regression Modeling via Stan*. R package version 2.21.1, URL https://CRAN.R-project.org/package=rstanarm.

Gabry J, Mahr T (2021). ***bayesplot**: Plotting for Bayesian Models*. R package version 1.8.1, URL https://CRAN.R-project.org/package=bayesplot.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013). *Bayesian Data Analysis*. 3rd edition. Chapman & Hall/CRC.

Hecht M, Gische C, Vogel D, Zitzmann S (2020). "Integrating Out Nuisance Parameters for Computationally More Efficient Bayesian Estimation – An Illustration and Tutorial." *Structural Equation Modeling: A Multidisciplinary Journal*, **27**, 483–493. doi:10.1080/10705511.2019.1647432.

Holzinger KJ, Swineford FA (1939). *A Study of Factor Analysis: The Stability of a Bi-factor Solution.* Number 48 in Supplementary Educational Monograph. University of Chicago Press, Chicago.

Jackman S (2009). *Bayesian Analysis for the Social Sciences.* John Wiley & Sons.

Jöreskog KG, Sörbom D (1997). ***LISREL** 8 User's Reference Guide.* Scientific Software International.

Joseph M (2016). "Exact Sparse CAR Models in Stan." *Technical Report 3*, Stan Case Studies. URL https://mc-stan.org/users/documentation/case-studies/mbjoseph-CARStan.html.

Kaplan D (2014). *Bayesian Statistics for the Social Sciences.* The Guilford Press, New York.

Kievit RA, Brandmaier AM, Ziegler G, van Harmelen AL, de Mooij SM, Moutoussis M, Goodyer IM, Bullmore E, Jones PB, Fonagy P, Lindenberger U, Dolan RJ (2018). "Developmental Cognitive Neuroscience Using Latent Change Score Models: A Tutorial and Applications." *Developmental Cognitive Neuroscience*, **33**, 99–117. doi:https://doi.org/10.1016/j.dcn.2017.11.007.

Kruschke JK (2011). *Doing Bayesian Data Analysis: A Tutorial with R and BUGS.* Academic Press, Burlington.

Lee SY (2007). *Structural Equation Modeling: A Bayesian Approach.* John Wiley & Sons, Chichester.

Lee SY, Song XY, Tang NS (2007). "Bayesian Methods for Analyzing Structural Equation Models with Covariates, Interaction, and Quadratic Latent Variables." *Structural Equation Modeling*, **14**, 404–434. doi:10.1080/10705510701301511.

Lewandowski D, Kurowicka D, Joe H (2009). "Generating Random Correlation Matrices Based on Vines and Extended Onion Method." *Journal of Multivariate Analysis*, **100**, 1989–2001. doi:10.1016/j.jmva.2009.04.008.

Liu H, Zhang Z, Grimm KJ (2016). "Comparison of Inverse Wishart and Separation-Strategy Priors for Bayesian Estimation of Covariance Parameter Matrix in Growth Curve Analysis." *Structural Equation Modeling*, **23**, 353–367. doi:10.1080/10705511.2015.1057285.

MacCallum RC, Edwards MC, Cai L (2012). "Hopes and Cautions in Implementing Bayesian Structural Equation Modeling." *Psychological Methods*, **17**, 340–345. doi:10.1037/a0027131.

Merkle EC (2011). "A Comparison of Imputation Methods for Bayesian Factor Analysis Models." *Journal of Educational and Behavioral Statistics*, **36**, 257–276. doi:10.3102/1076998610375833.

Merkle EC, Furr D, Rabe-Hesketh S (2019). "Bayesian Model Assessment: Use of Conditional vs Marginal Likelihoods." *Psychometrika*, **84**, 802–829. doi:10.1007/s11336-019-09679-0.

Merkle EC, Rosseel Y (2018). "**blavaan**: Bayesian Structural Equation Models via Parameter Expansion." *Journal of Statistical Software*, **85**(4), 1–30. doi:10.18637/jss.v085.i04.

Merkle EC, Wang T (2018). "Bayesian Latent Variable Models for the Analysis of Experimental Psychology Data." *Psychonomic Bulletin & Review*, **25**, 256–270. doi:10.3758/s13423-016-1016-7.

Mulder J, Pericchi LR (2018). "The Matrix-F Prior for Estimating and Testing Covariance Matrices." *Bayesian Analysis*, **13**, 1193–1214. doi:10.1214/17-BA1092.

Muthén B, Asparouhov T (2012). "Bayesian Structural Equation Modeling: A More Flexible Representation of Substantive Theory." *Psychological Methods*, **17**, 313–335. doi:10.1037/a0026802.

Muthén LK, Muthén B (2017). ***Mplus** User's Guide*. 8th edition. Muthén & Muthén, Los Angeles.

O'Muircheartaigh C, Moustaki I (1999). "Symmetric Pattern Models: A Latent Variable Approach to Item Non-Response in Attitude Scales." *Journal of the Royal Statistical Society A*, **162**, 177–194. doi:10.1111/1467-985X.00129.

Palomo J, Dunson DB, Bollen K (2007). "Bayesian Structural Equation Modeling." In SY Lee (ed.), *Handbook of Latent Variable and Related Models*, pp. 163–188. Elsevier. doi:10.1016/B978-044452044-9/50011-2.

Pearl J (2013). "Linear Models: A Useful "Microscope" for Causal Analysis." *Journal of Causal Inference*, **1**, 155–170. doi:10.1515/jci-2013-0003.

Peeters CFW (2012). "Rotational Uniqueness Conditions under Oblique Factor Correlation Metric." *Psychometrika*, **77**, 288–292. doi:10.1007/S11336-012-9259-3.

Petersen KB, Pedersen MS (2012). "The Matrix Cookbook." Version 20121115, URL http://www2.imm.dtu.dk/pubdb/p.php?3274.

Plummer M (2003). "**JAGS**: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. Technische Universität Wien, Vienna, Austria. URL https://www.R-project.org/conferences/DSC-2003/Proceedings/Plummer.pdf.

Pritikin JN (2021). ***pcFactorStan**: Stan Models for the Pairwise Comparison Factor Model*. R package version 1.5.3, URL https://CRAN.R-project.org/package=pcFactorStan.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rosseel Y (2012). "**lavaan**: An R Package for Structural Equation Modeling." *Journal of Statistical Software*, **48**(2), 1–36. doi:10.18637/jss.v048.i02.

Song XY, Lee SY (2012). *Basic and Advanced Bayesian Structural Equation Modeling: With Applications in the Medical and Behavioral Sciences*. John Wiley & Sons, Chichester.

Spezia L (2019). "Modeling Covariance Matrices by the Trigonometric Separation Strategy with Application to Hidden Markov Models." *TEST*, **28**, 399–422. doi:10.1007/s11749-018-0580-8.

Spiegelhalter DJ, Best NG, Carlin BP, Linde AVD (2002). "Bayesian Measures of Model Complexity and Fit." *Journal of the Royal Statistical Society B*, **64**, 583–639. doi:10.1111/1467-9868.00353.

Stan Development Team (2021). "**rstan**: The R Interface to Stan." R package version 2.21.2, URL https://CRAN.R-project.org/package=rstan.

Stroup WW (2013). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. Chapman & Hall/CRC, Boca Raton.

Talts S, Betancourt M, Simpson D, Vehtari A, Gelman A (2018). "Validating Bayesian Inference Algorithms with Simulation-Based Calibration." arXiv:1804.06788 [stat.ME], URL http://arxiv.org/abs/1804.06788.

Van Erp S, Mulder J, Oberski DL (2018). "Prior Sensitivity Analysis in Default Bayesian Structural Equation Modeling." *Psychological Methods*, **23**, 363–388. doi:10.1037/met0000162.

Watanabe S (2010). "Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory." *Journal of Machine Learning Research*, **11**, 3571–3594.

Wothke W (2000). "Longitudinal and Multi-Group Modeling with Missing Data." In TD Little, KU Schnabel, J Baumert (eds.), *Modeling Longitudinal and Multilevel Data: Practical Issues, Applied Approaches, and Specific Examples*. Lawrence Erlbaum Associates, Mahwah.

Yackulic CB, Dodrill M, Dzul M, Sanderlin JS, Reid JA (2020). "A Need for Speed in Bayesian Population Models: A Practical Guide to Marginalizing and Recovering Discrete Latent States." *Ecological Applications*, **30**(5), e02112. doi:10.1002/eap.2112.

Zhang X, Tao J, Wang C, Shi NZ (2019). "Bayesian Model Selection Methods for Multilevel IRT Models: A Comparison of Five DIC-based Indices." *Journal of Educational Measurement*, **56**, 3–27. doi:10.1111/jedm.12197.

Affiliation:

Edgar C. Merkle
Department of Psychological Sciences
University of Missouri
28A McAlester Hall
Columbia, MO, 65211, United States of America
E-mail: merklee@missouri.edu
URL: https://ecmerkle.github.io/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2021, Volume 100, Issue 6 *Submitted:* 2019-06-29
doi:10.18637/jss.v100.i06 *Accepted:* 2021-02-26