

# bmemLavaan: An R Package for Estimating and Testing Mediation Effect in Mediation Models

Shuigen Ming and Hong Zhang  
University of Science and Technology of China

Zhiyong Zhang and Lijuan Wang  
University of Notre Dame

---

## Abstract

An R package **bmemLavaan** for estimating and testing mediation effect in mediation models is developed. The package uses the popular R package **lavaan** to specify mediation models. The package is especially useful to handle complex real data such as missing data and non-normal data, with four methods to deal with different missing data mechanisms and a robust method to deal with non-normal data. Three methods are included to conduct significance testing for mediation effects, two of which can efficiently deal with non-normal data. The package can also be used for sample size planning. This paper explains the aims and principles of the package and provides six examples of different complexities on how to use the package to conduct different types of mediation analysis.

*Keywords:* mediation models, mediation effect, missing data, non-normal data, R.

---

## 1. Introduction

Mediation analysis is widely used in social, behavioral, and education research (e.g., [MacKinnon, Fairchild, and Fritz 2007](#); [Preacher 2015](#)) as well as in health and medical studies (e.g., [Lange, Hansen, Sørensen, and Galatius 2017](#); [VanderWeele 2016](#)). Mediation analysis can help researchers understand the mechanism why one variable is related to another variable. For example, [Salthouse \(1993\)](#) found that processing speed mediated the relationship between age and cognition, which led to a theory to explain the adult age differences in cognition. Although simple mediation analysis can be conducted in the regression framework, more complex mediation analysis is typically done in the structural equation modeling (SEM) framework (e.g., [Iacobucci 2008](#); [MacKinnon 2008](#)).

There are two paradigms for studying mediation, the traditional approach and the causal mediation approach. The traditional approach was first systematically discussed by [Baron and Kenny \(1986\)](#). The method simply decomposes the total effect of an input variable  $X$  on an outcome variable  $Y$  to the direct effect from  $X$  and  $Y$  and the indirect effect through a mediator  $M$ . The method is still the dominated method in mediation analysis although it heavily relies on the assumption of no unmeasured confounding effects. The second approach, now often called causal mediation method, was typically built under the potential outcomes perspective (e.g., [Robins and Greenland 1992](#)) and was made popular recently through the

work by Imai and colleagues (e.g., Imai, Keele, and Tingley 2010; Tingley, Yamamoto, Hirose, Keele, and Imai 2014). Currently, the traditional approach is still more widely used in social sciences because of its flexibility to handle complex multivariate and longitudinal models as well as different types of data distributions (MacKinnon and Fairchild 2009; Tofghi and Thoemmes 2014; Preacher 2015).

Many commercial and free software programs can be used to conduct mediation analysis. For example, the SEM software **Mplus** can be used to conduct both traditional and causal mediation analysis (Muthén and Muthén 1998-2021). Other popular statistical programs such as **SAS** and **Stata** can also be used to conduct mediation analysis. Several R packages are available for each type of mediation analysis. For example, the packages **mediation** (Tingley *et al.* 2014) and **mma** (Yu and Li 2017) can be used to conduct casual mediation analysis with one or multiple mediators. The packages **RMediation** (Tofghi and MacKinnon 2011) and **bmem** (Zhang and Wang 2013b) can be used to conduct traditional mediation analysis. Particularly, we initially developed the R package **bmem** (Zhang and Wang 2013b) to expand the ability of the widely used R package **sem** (Fox 2006) for advanced mediation analysis. It eases the use of the bootstrap method for testing the mediation effects, and it also allows the handling of missing data and non-normal data (Zhang and Wang 2013a). Nowadays, the package **lavaan** (Rosseel 2012) has become the most popular R package for SEM. **lavaan** is also under active development with more functionalities than **sem**. At the same time, **bmem** is not actively maintained any more, and also lacks a document to explain how it can be used. Therefore, we decide to rewrite **bmem** to develop the package **bmemLavaan** to take advantage of both **lavaan** and **bmem** for mediation analysis. The package **bmemLavaan** provides the following useful features.

- **bmemLavaan** uses the same syntax as **lavaan** to specify a mediation model. This can benefit the existing large user base of **lavaan** and make **bmemLavaan** easy to adopt.
- In addition to the basic bootstrap method available in **lavaan**, **bmemLavaan** offers three types of bootstrap confidence intervals including the percentile, bias-corrected, and bias-corrected and accelerated intervals.
- **bmemLavaan** can handle missing data through listwise deletion, pairwise deletion, full information maximum likelihood, and two-stage methods as well as multiple imputation. **bmemLavaan** also expands the two-stage method for easy inclusion of auxiliary variables to handle certain types of missing not at random data.
- **bmemLavaan** implements an M-estimator to handle non-normal data using the robust procedure developed by Yuan and Zhang (2012). The method can also handle non-normal data and missing data simultaneously.
- **bmemLavaan** provides functions for statistical power analysis for testing mediation effects (Zhang and Wang 2013a).

The rest of the paper is organized as follows. In Section 2, we provide an overview of the methods behind the development of **bmemLavaan**. In Section 3, we discuss the implementation and main functions of **bmemLavaan**. In Section 4, we provide several examples to illustrate the use of **bmemLavaan**. In Section 5, we discuss the limitations and future directions of the development of **bmemLavaan**.

## 2. Methods

This section reviews the methods implemented in **bmemLavaan** based on a general SEM model. The model can be specified as a multiple-group **LISREL** (Linear Structural RELationship) model as follows:

$$\begin{aligned} \mathbf{x}_g &= \mathbf{\Lambda}_{xg}\boldsymbol{\xi}_g + \boldsymbol{\delta}_g, \\ \mathbf{y}_g &= \mathbf{\Lambda}_{yg}\boldsymbol{\eta}_g + \boldsymbol{\epsilon}_g, \\ \boldsymbol{\eta}_g &= \mathbf{B}_g\boldsymbol{\eta}_g + \mathbf{\Gamma}_g\boldsymbol{\xi}_g + \boldsymbol{\zeta}_g, \end{aligned} \quad (1)$$

where the subscript  $g$  represents the  $g^{th}$  group. Note that the parameters in model (1) are group specific. For the sake of simplicity, the subscript  $g$  is dropped in the following discussion. The first two equations in model (1) are called measurement equations. In this model,  $\boldsymbol{\eta}$  is a vector of latent endogenous variables and  $\boldsymbol{\xi}$  is a vector of latent exogenous variables.  $\mathbf{x}$  and  $\mathbf{y}$  are vectors of observed indicators of  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$ , respectively,  $\boldsymbol{\delta}$  and  $\boldsymbol{\epsilon}$  are measurement errors associated with  $\mathbf{x}$  and  $\mathbf{y}$ , respectively, and  $\mathbf{\Lambda}_x$  and  $\mathbf{\Lambda}_y$  are matrices of factor loadings. The third equation in the model is called a structural equation, where  $\mathbf{B}$  is a matrix of coefficients determining the relationship among  $\boldsymbol{\eta}$  and  $\mathbf{\Gamma}$  is a matrix of coefficients specifying the relationship between  $\boldsymbol{\eta}$  and  $\boldsymbol{\xi}$ . From a mediation analysis point of view, the coefficients in  $\mathbf{B}$  and  $\mathbf{\Gamma}$  represent the direct relationships, while the mediation effects (indirect effects)  $\theta$  are functions of  $\mathbf{B}$  and  $\mathbf{\Gamma}$ , which is denoted as follows in the paper:

$$\theta = I(\mathbf{B}, \mathbf{\Gamma}). \quad (2)$$

### 2.1. Parameter estimation in mediation models

Generally speaking, to calculate a mediation effect, one first estimates the direct coefficients in  $\mathbf{B}$  and  $\mathbf{\Gamma}$  in the SEM framework and then obtains a specific mediation effect estimate by replacing  $\mathbf{B}$  and  $\mathbf{\Gamma}$  with their estimates in Equation (2). For example, a simple mediation model is shown in Figure 1. In this model,  $X$ ,  $Y$ , and  $M$  are input, output, and mediator, respectively, while the coefficients  $a$  and  $b$  are direct effect coefficients.

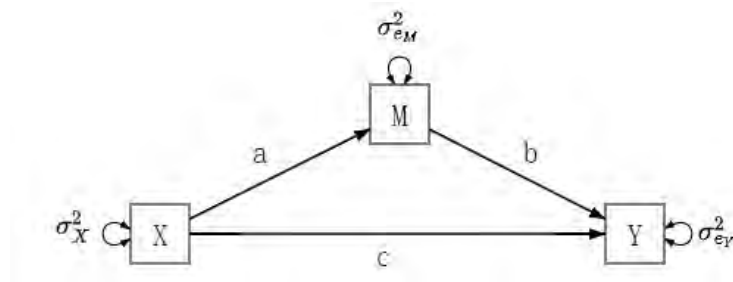


Figure 1: A simple example of mediation model (Zhang and Wang 2013a).

The simple mediation model can be specified using two regression equations:

$$\begin{aligned} M &= aX + e_M, \\ Y &= bM + cX + e_Y. \end{aligned} \quad (3)$$

For this simple mediation model, the mediation effect (indirect effect) is  $\theta = I(a, b) = a \times b$ . The coefficients  $a$  and  $b$  in model (3) can be estimated through SEM by rewriting the model as a LISREL model, which takes the following form:

$$\begin{aligned}
\mathbf{x} &= \mathbf{\Lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta}, \\
X &= 1 \xi + 0, \\
\mathbf{y} &= \mathbf{\Lambda}_y \boldsymbol{\eta} + \boldsymbol{\epsilon}, \\
\begin{bmatrix} M \\ Y \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
\boldsymbol{\eta} &= \mathbf{B} \boldsymbol{\eta} + \boldsymbol{\Gamma} \boldsymbol{\xi} + \boldsymbol{\zeta}, \\
\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ b & 0 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} + \begin{bmatrix} a \\ c \end{bmatrix} \xi + \begin{bmatrix} e_M \\ e_Y \end{bmatrix}.
\end{aligned} \tag{4}$$

Once the estimates  $\hat{a}$  and  $\hat{b}$  are obtained, the mediation effect can be naturally estimated as  $\hat{a} \times \hat{b}$ .

Theoretically, with the point estimates of the direct coefficients  $\mathbf{B}$  and  $\boldsymbol{\Gamma}$ , hypothesis testing of the mediation effect can be conducted through the delta method (Sobel 1982). Computationally, in the case of complete data, the R package **bmemLavaan** first estimates the direct coefficients using least squares methods or maximum likelihood methods assuming normality implemented in the R package **Lavaan**, and then estimates the mediation effects based on Equation (2). The standard errors of these parameters including the mediation effect can also be estimated. Based on these estimates, hypothesis testing and power analysis can further be conducted. In addition, the R package **bmemLavaan** implements several methods to handling missing data and non-normal data problems, which are detailed in Section 2.3 and Section 2.4.

## 2.2. Hypothesis testing and confidence interval

Once the point estimates of the mediation effects are obtained, statistical inference of mediation effects can be conducted (e.g. MacKinnon 2008). The Sobel test (Sobel 1982) has been widely used to conduct significance tests in the literature. However, MacKinnon, Lockwood, Hoffman, West, and Sheets (2002) found through simulation study that the Sobel test is overly conservative and thus underpowered. Therefore, tests based on bootstrap methods, which are more powerful, are recommended for testing mediation effects. A bootstrap method can obtain empirical distributions of unknown parameters, and confidence intervals can be constructed for significance testing of mediation effects.

**bmemLavaan** implements four types of bootstrapping based CI methods, namely the normal bootstrap CI, percentile bootstrap CI, bias-corrected (BC) bootstrap CI, and bias-corrected and accelerated (BCa) bootstrap CI (MacKinnon, Lockwood, and Williams 2004) (refer to Section 2.4 for details). One can also construct other types of test statistics and CIs with the estimated parameters and corresponding bootstrap samples.

## 2.3. Parameter estimation methods in the presence of missing data

In many studies, especially for observational studies in behavioral and social sciences, missing data are common. In this subsection, we review four methods that are included in the **bmemLavaan** package for handling missing data. One easy and common strategy is to ignore the observations with missing value. Listwise deletion (LD) and pairwise deletion (PD) are two types of commonly used methods based on the strategy. Because the lack of efficiency of LD and PD, multiple imputation (MI) is now widely used to handle missing data. Our

package also implements a two-stage maximum likelihood (TS-ML) based on the expectation-maximization (EM) algorithm.

### *Listwise deletion (LD) and pairwise deletion (PD)*

In LD, a case is removed if the value of any single model variable from the case is missing, and the mediation model is then estimated based on cases with complete observations. In PD, the covariance between any two variables is estimated using the complete data from both variables, and an estimated variance-covariance matrix for all model variables are consequently constructed. Based on the PD method, the number of cases could vary from covariance to covariance, and the estimated variance-covariance matrix may not be positive definite (Little and Rubin 2019). The direct effects in a mediation model can typically be estimated by minimizing the discrepancy function

$$F = \text{tr}(\mathbf{S}\boldsymbol{\Sigma}^{-1}) - \log|\mathbf{S}\boldsymbol{\Sigma}^{-1}| - p, \quad (5)$$

where  $\boldsymbol{\Sigma}$  is a variance-covariance matrix derived from the model under investigation,  $\mathbf{S}$  is the sample variance-covariance matrix, and  $p$  is the total number of observed variables in the model. The mediation effects can be estimated using the plug-in strategy based on Equation (2). Although the idea of the LD and PD methods is simple, they cannot utilize all the information in the data and can be inefficient.

### *Multiple imputation (MI)*

Multiple imputation (Little and Rubin 2019; Schafer 1997) is a procedure of filling each missing value with a set of plausible values of the true value. The multiple imputed datasets are then analyzed using the standard procedures for complete data, and the results from these analyses are combined to obtain point estimates of model parameters. Multiple imputation has been implemented in software such as SAS and R. For **bememLavaan**, the following multiple imputation algorithm for mediation analysis is implemented.

1. Let  $(X_1, Y_1), (X_2, Y_2), \dots, (X_p, Y_p)$  denote  $(\mathbf{x}, \mathbf{y})$  in the **LISREL** model (1), and let  $A_1, \dots, A_q$  denote auxiliary variables. Assume that

$$\mathbf{Z} := (X_1, Y_1, X_2, Y_2, \dots, X_p, Y_p, A_1, \dots, A_q)', \quad (6)$$

follows a multivariate normal distribution and denote its complete-data parameters, which is the corresponding mean and variance-covariance matrix, as  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ , respectively. Using the EM algorithm with a bootstrap (EMB) approach, we take draws from the posterior of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  based on the observed data (Honaker and King 2010).

2. Generate  $K$  draws from the posterior of the complete-data parameters. Then make imputations by drawing missing values from its distribution conditional on observed data and the draws of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ . Combine the generated values with the observed data to obtain  $K$  sets of complete data for  $(X_1, Y_1, X_2, Y_2, \dots, X_p, Y_p)$ .
3. For the  $k^{\text{th}}$  complete dataset ( $k = 1, \dots, K$ ), apply the complete data mediation analysis to obtain an estimate of the mediation effect, denoted by  $I(\hat{\mathbf{B}}_k, \hat{\boldsymbol{\Gamma}}_k)$ .

4. A combined estimate (the final point estimate from MI) for the mediation effect is defined as the average of the  $K$  complete data mediation effect estimates:

$$I(\hat{\mathbf{B}}, \hat{\mathbf{\Gamma}}) = \frac{1}{K} \sum_{k=1}^K I(\hat{\mathbf{B}}_k, \hat{\mathbf{\Gamma}}_k). \quad (7)$$

### *Two-stage maximum likelihood method (TS-ML)*

The EM algorithm is a popular technique for handling missing data (Little and Rubin 2019; Schafer 1997). It is an iterative method consisting of the expectation step (E-step) and maximization step (M-step). It starts with a guess of unknown parameter values. For example, parameter estimates from LD can be used as starting values. In the E-step, the conditional expectation of complete data log-likelihood function with respect to missing data given the observed data are calculated under data distribution with current parameter values. In the M-step, new parameter estimates are obtained by maximizing the resulting conditional expectation with respect to unknown parameters. These two steps are repeated until changes in the parameter estimates are small enough.

A two-stage maximum likelihood (TS-ML) method using the EM algorithm is adopted in mediation analysis in the presence of missing data. In the first stage, the EM algorithm is applied to estimate the saturated mean and variance-covariance matrix of  $Z$  given in Equation (6) without assuming any mediation model. In the second stage, model parameters are estimated using the saturated variance-covariance matrix obtained from the first stage. The two-stage strategy has been applied in the literature to deal with missing data under different situations (Yuan, Chan, and Bentler 2000; Enders 2003). There are at least two reasons for adopting the TS-ML method instead of the full information maximum likelihood (FIML) method. First, the TS-ML method performs at least comparably well with the FIML method (Savalei and Bentler 2009; Savalei and Falk 2014). Second, it is straightforward to include auxiliary variables in the TS-ML method, but this is not case in the FIML method. To include auxiliary variables, one only needs to augment them with the model variables directly as shown below.

For sample  $i$ , let  $z_{i,\text{obs}}$  and  $z_{i,\text{miss}}$  denote the observed data and missing data, respectively. Furthermore, let  $U$  denote the mean vector, and  $S$  denote the variance-covariance matrix of the augmented data  $Z$  (i.e.  $(X_1, Y_1, \dots, X_p, Y_p, A_1, \dots, A_q)'$  as mentioned in Equation (6)). The TS-ML algorithm is detailed as follows.

1. Let  $e$  denote the desired degree of accuracy, such as  $10^{-6}$ , which is also the iteration stopping criterion.  $U^{(0)}$  and  $S^{(0)}$  are the initial estimates from the LD method, which are the sample mean and variance-covariance matrix for the complete observed data.
2. Let  $U^{(t)}$  and  $S^{(t)}$  denote the estimates of  $U$  and  $S$  at the  $t$ th iteration. For iteration  $t+1$ , the EM algorithm is implemented in two steps. In the E-step, calculate the conditional expectation  $z_{i,\text{miss}}^{(t)}$  of  $z_{i,\text{miss}}$  given  $z_{i,\text{obs}}$ ,  $U = U^{(t)}$ , and  $S = S^{(t)}$  (refer to Appendix A for details). In the M-step, replace  $z_{i,\text{miss}}$  with the conditional expectation  $z_{i,\text{miss}}^{(t)}$  and then calculate the resulting sample mean  $U^{(t+1)}$  and sample variance  $S^{(t+1)}$ .

3. Calculate the maximum absolute error (or absolute relative difference) between  $(U^{(t)}, S^{(t)})$  and  $(U^{(t+1)}, S^{(t+1)})$ :

$$d = \max_{j,k} \left[ \left| u_j^{(t+1)} - u_j^{(t)} \right|, \left| s_{jk}^{(t+1)} - s_{jk}^{(t)} \right| \right],$$

where  $u_j$  is the  $j$ -th element of  $U$  and  $s_{jk}$  is the  $(j, k)$ -th element of  $S$ . If  $d < e$ , stop the iteration and go to step 4. Otherwise, go to step 2.

4. Let  $\hat{U}$  and  $\hat{S}$  denote the estimates of  $U$  and  $S$  from the EM algorithm after convergence. Let  $\hat{U}_M$  and  $\hat{S}_M$  denote the estimated mean vector and variance-covariance matrix of  $(X_1, Y_1, \dots, X_p, Y_p)$  in the mediation model. Then, the mediation model parameters can be estimated by minimizing Equation (5) with  $S$  being replaced with  $\hat{S}_M$ . Denote the resulting estimates of  $\mathbf{B}$  and  $\mathbf{\Gamma}$  by  $\hat{\mathbf{B}}$  and  $\hat{\mathbf{\Gamma}}$ , respectively. The final estimator of mediation effect is  $I(\hat{\mathbf{B}}, \hat{\mathbf{\Gamma}})$ .

Both MI and TS-ML methods used in **bmemLavaan** package can easily handle auxiliary variables for better missing data processing by augmenting the auxiliary variables with the model variables.

## 2.4. Parameter estimation methods in the presence of non-normal data

In practice, the observed data are often non-normal (Cain, Zhang, and Yuan 2017). With non-normal data, the estimated standard errors of the parameter estimates based on the normality assumption are biased, and the corresponding normal CI of a mediation effect is distorted. Our package **bmemLavaan** allows the handling of non-normal data through bootstrap, robust sandwich-type standard error, and robust Huber-type M-estimator.

### *Bootstrap method*

The first solution is the bootstrap method. The normal CI is based on the asymptotic theory and it might not perform well in finite sample situations (MacKinnon *et al.* 2004; Zu and Yuan 2010). In the literature, CIs constructed using the bootstrap method have been shown to perform better under many studied conditions (Cheung 2007; Fritz and MacKinnon 2007; Hayes and Scharkow 2013; MacKinnon *et al.* 2004; Preacher and Hayes 2004; Shrout and Bolger 2002). The algorithm for the bootstrap method is straightforward.

1. Randomly draw a sample of  $n$  individuals with replacement from the original dataset of sample size  $n$ . The resulting sample is called a bootstrap sample.
2. With the bootstrap sample, estimate model parameters and compute mediation effects.
3. Repeat steps 1 and 2 for  $B$  times to get  $B$  sets of estimates, and bootstrap CIs of model parameters and mediation effects are constructed based on these estimates.

Several types of bootstrap CIs have been adopted in the literature of mediation analysis (Cheung 2007; Fritz and MacKinnon 2007; Hayes and Scharkow 2013; MacKinnon *et al.* 2004), of which the percentile bootstrap CI and BC bootstrap CI are widely used in practice. The percentile bootstrap CI is the default in the package **bmemLavaan** due to its simplicity and

good performance. Let  $\theta$  denote a population mediation effect,  $\hat{\theta}$  and  $\hat{\theta}_i$  its estimators from the original data and the  $i^{\text{th}}$  bootstrap sample ( $i = 1, \dots, B$ ), respectively. The  $100(1 - \alpha)\%$  percentile bootstrap CI can be constructed by  $[\hat{\theta}_i(\alpha/2), \hat{\theta}_i(1 - \alpha/2)]$  with  $\hat{\theta}_i(a)$  denoting the  $100a^{\text{th}}$  percentile of the  $B$  bootstrap estimates. Alternatively, one can use the BC bootstrap CI, which has a better coverage probability in general. The BC bootstrap CI takes the form  $[\hat{\theta}_i(\tilde{\alpha}_l), \hat{\theta}_i(\tilde{\alpha}_u)]$ , where  $\tilde{\alpha}_l$  and  $\tilde{\alpha}_u$  are quantiles defined by

$$\tilde{\alpha}_l = \Phi[2z_0 + \Phi^{-1}(\alpha/2)] \text{ and } \tilde{\alpha}_u = \Phi[2z_0 + \Phi^{-1}(1 - \alpha/2)],$$

with

$$z_0 = \Phi^{-1} \left[ \frac{\#\{i : \hat{\theta}_i < \hat{\theta}; i = 1, \dots, B\}}{B} \right].$$

Take the mediation effect  $\theta = ab$  in the simple mediation model (4) for example. With the  $i^{\text{th}}$  ( $i = 1, \dots, B$ ) bootstrapping sample, one first estimates  $ab$  as  $\hat{\theta}_i := \hat{a}_i \hat{b}_i$ , where  $\hat{a}_i$  and  $\hat{b}_i$  are the estimates of  $a$  and  $b$ , respectively. One can first construct a  $z$ -test type of statistic that takes the form  $z = \hat{\theta}/\widehat{\text{se}}(\hat{\theta})$ , where  $\hat{\theta}$  is the parameter estimates from the original data and  $\widehat{\text{se}}(\hat{\theta}) = \sqrt{B^{-1} \sum_{i=1}^B (\hat{\theta}_i - \bar{\theta})^2}$  with  $\bar{\theta} = B^{-1} \sum_{i=1}^B \hat{\theta}_i$ . The corresponding  $(1 - \alpha)$  normal approximation bootstrap CI of  $\theta$  takes the form

$$[\hat{\theta} + \Phi^{-1}(\alpha/2) \times \widehat{\text{se}}(\hat{\theta}), \hat{\theta} + \Phi^{-1}(1 - \alpha/2) \times \widehat{\text{se}}(\hat{\theta})], \quad (8)$$

where  $\Phi$  is the standard normal cumulative distribution function. This method assumes that the bootstrap distribution can be well approximated by a normal distribution. The percentile and BC bootstrap CIs can be constructed as discussed above for  $\theta = ab$ .

### *Robust sandwich-type method*

The second type of method is to deal with non-normal data by constructing a sandwich-type standard error for parameter estimates (Zu and Yuan 2010). A robust CI of a mediation effect is then obtained by replacing the normal-based standard error with the sandwich-type standard error in Equation (8).

Recall that the simple mediation model in Figure 1 consists of two regression equations shown in Equation (3). Denote the parameters of the model as  $\boldsymbol{\theta} = (a, b, c, \sigma_x, \sigma_{\epsilon_m}, \sigma_{\epsilon_y})$ , where  $\sigma_x, \sigma_{\epsilon_m}$ , and  $\sigma_{\epsilon_y}$  are the variances of  $X, e_M$ , and  $e_Y$ , respectively. In what follows, we describe the estimation procedure of  $\boldsymbol{\theta}$ . Denote  $\mathbf{Z} = (X, M, Y)$  and  $\mathbf{Z} \sim N_3(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , where  $\boldsymbol{\Sigma}$  is the variance-covariance matrix of the simple model (3). Let  $\mathbf{z}_i, i = 1, \dots, n$  be a random sample of size  $n$  from  $\mathbf{Z}$ . We estimate  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  with the sample mean  $\bar{\mathbf{z}}$  and sample variance-covariance matrix  $\mathbf{S}$ , respectively. If we denote the estimator of  $\boldsymbol{\theta}$  as  $\hat{\boldsymbol{\theta}} = (\hat{a}, \hat{b}, \hat{c}, \hat{\sigma}_x, \hat{\sigma}_{\epsilon_m}, \hat{\sigma}_{\epsilon_y})$ , then  $\hat{a}, \hat{b}$ , and  $\hat{c}$  can be expressed as functions of  $\mathbf{S}$ . Specifically, let

$$\mathbf{S} = \begin{pmatrix} s_{xx} & s_{mx} & s_{yx} \\ s_{mx} & s_{mm} & s_{ym} \\ s_{yx} & s_{ym} & s_{yy} \end{pmatrix}, \quad (9)$$

then,

$$\hat{a} = (s_{xx})^{-1} s_{mx},$$



$$\begin{pmatrix} \hat{c} \\ \hat{b} \end{pmatrix} = \begin{pmatrix} s_{xx} & s_{mx} \\ s_{mx} & s_{mm} \end{pmatrix}^{-1} \begin{pmatrix} s_{yx} \\ s_{ym} \end{pmatrix}.$$

The estimators  $\hat{\sigma}_x$ ,  $\hat{\sigma}_{\epsilon_m}$ , and  $\hat{\sigma}_{\epsilon_y}$  can be obtained naturally using residual sum of squares based on  $\hat{a}$ ,  $\hat{b}$ , and  $\hat{c}$ .

Now we estimate the variance-covariance matrix of  $\hat{\boldsymbol{\theta}}$ . To this end, one can first estimate the variance-covariance matrix of  $\mathbf{S}$ , then apply the delta method (Casella and Berger 2002). Let  $\boldsymbol{\eta} = (\boldsymbol{\mu}', \boldsymbol{\sigma}')'$ , where  $\boldsymbol{\sigma} = \text{vec}(\boldsymbol{\Sigma})$  is a vector that stacks the columns of  $\boldsymbol{\Sigma}$  while omitting the elements above the diagonal. For the simple mediation model (3),  $\boldsymbol{\sigma} = (\sigma_{xx}, \sigma_{mx}, \sigma_{yx}, \sigma_{mm}, \sigma_{ym}, \sigma_{yy})'$ . Denote  $\hat{\boldsymbol{\eta}} = (\hat{\boldsymbol{z}}', \text{vec}(\hat{\mathbf{S}})')$ . Under the mild assumption that  $\mathbf{Z}$  has finite fourth-order moments, Zu and Yuan (2010) established the following asymptotic normality for  $\hat{\boldsymbol{\eta}}$ :

$$\sqrt{n}(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}) \rightarrow N(\mathbf{0}, \boldsymbol{\Omega}_{SW}) \text{ in distribution as } n \rightarrow \infty, \quad (10)$$

where  $\boldsymbol{\Omega}_{SW}$  is a sandwich-type variance-covariance matrix:

$$\boldsymbol{\Omega}_{SW} := \mathbf{A}^{-1} \mathbf{B} (\mathbf{A}')^{-1} \text{ with } \mathbf{A} = -\ddot{L}(\boldsymbol{\eta}) \text{ and } \mathbf{B} = \dot{L}'(\boldsymbol{\eta}) \dot{L}(\boldsymbol{\eta}).$$

Here  $L(\boldsymbol{\eta})$  is the log-likelihood function, and  $\dot{L}$  and  $\ddot{L}$  are the first derivative vector and second derivative matrix of  $L$ , respectively.  $\boldsymbol{\Omega}_{SW}$  can be consistently estimated by

$$\hat{\boldsymbol{\Omega}}_{SW} = \hat{\mathbf{A}}^{-1} \hat{\mathbf{B}} (\hat{\mathbf{A}}')^{-1} \text{ with } \hat{\mathbf{A}} = -\frac{1}{n} \sum_{i=1}^n \ddot{L}_i(\hat{\boldsymbol{\eta}}) \text{ and } \hat{\mathbf{B}} = \frac{1}{n} \sum_{i=1}^n \dot{L}'_i(\hat{\boldsymbol{\eta}}) \dot{L}_i(\hat{\boldsymbol{\eta}}), \quad (11)$$

where  $L_i$ ,  $\dot{L}_i$ , and  $\ddot{L}_i$  are the analogues of  $L$ ,  $\dot{L}$ , and  $\ddot{L}$ , respectively, but for the  $i^{\text{th}}$  individual. The second step is to estimate the variance-covariance of  $\hat{\boldsymbol{\theta}}$ . Since  $\hat{\boldsymbol{\theta}}$  is a function of  $\mathbf{s} = \text{vec}(\hat{\mathbf{S}})$ , we may write it as

$$\hat{\boldsymbol{\theta}} = h(\mathbf{S}).$$

If  $\boldsymbol{\Omega}_{SW}$  is partitioned into blocks involving  $\boldsymbol{\mu}$  and  $\boldsymbol{\sigma}$ , that is,

$$\boldsymbol{\Omega}_{SW} = \begin{pmatrix} \boldsymbol{\Omega}_{\boldsymbol{\mu}\boldsymbol{\mu}'} & \boldsymbol{\Omega}_{\boldsymbol{\mu}\boldsymbol{\sigma}'} \\ \boldsymbol{\Omega}_{\boldsymbol{\sigma}\boldsymbol{\mu}'} & \boldsymbol{\Omega}_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \end{pmatrix},$$

then the delta method gives that

$$\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \rightarrow N(\mathbf{0}, \boldsymbol{\Omega}_{\theta}) \text{ in distribution as } n \rightarrow \infty,$$

where  $\boldsymbol{\Omega}_{\theta}$  can be estimated by

$$\hat{\boldsymbol{\Omega}}_{\theta} = \dot{h}(\mathbf{s}) \hat{\boldsymbol{\Omega}}_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \dot{h}'(\mathbf{s}). \quad (12)$$

Here  $\hat{\boldsymbol{\Omega}}_{\boldsymbol{\sigma}\boldsymbol{\sigma}'}$  is an appropriate estimator of  $\boldsymbol{\Omega}_{\boldsymbol{\sigma}\boldsymbol{\sigma}'}$ . Furthermore, the standard error of the mediation effect can be estimated using the delta method since the mediation effect is a smooth function of  $\boldsymbol{\theta}$ .

### *Robust Huber-type method*

**bmemLavaan** also implements a robust M-estimators that can be used in situations where outliers are present or the data distributions have long tails in real data (Yuan and Zhang

2012). The key idea is to assign a weight to each observation according to its position in the data so that extreme cases have relatively smaller weights in estimating the mediation models.

For the robust sandwich-type standard error based on MLEs, each case is assigned a weight of 1.0. In this robust procedure, the position of each case is measured by the Mahalanobis distance defined as

$$d^2(\mathbf{z}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\mathbf{z}_i - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{z}_i - \boldsymbol{\mu}).$$

Let  $u_1(d)$  and  $u_2(d)$  be non-negative weight functions, which are monotonically decreasing in  $d$ . In Maronna (1976), robust M-estimators of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  are defined as

$$\hat{\boldsymbol{\mu}} = \frac{\sum_{i=1}^n u_1\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})\} \mathbf{z}_i}{\sum_{i=1}^n u_1\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})\}} \quad (13)$$

and

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^n u_2\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})\} (\mathbf{z}_i - \hat{\boldsymbol{\mu}})(\mathbf{z}_i - \hat{\boldsymbol{\mu}})', \quad (14)$$

respectively, where  $u_1\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})\}$  is the weight for estimating  $\boldsymbol{\mu}$  and  $u_2\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})\}$  is the weight for estimating  $\boldsymbol{\Sigma}$ . In the package **bmemLavaan**, we consider using the well-known Huber-type weight, which is widely used in the robust literature (Huber 1981; Tyler 1983; Wilcox 2011; Yuan and Bentler 1998; Yuan *et al.* 2000). The Huber-type weight is given as

$$u_1(d) = \begin{cases} 1 & \text{if } d \leq r, \\ r/d & \text{if } d > r, \end{cases} \quad \text{and} \quad u_2(d) = u_1^2(d)/\tau,$$

where  $r^2$  is the upper  $\kappa$ -quantile of the  $\chi^2$  distribution with  $p$  degrees of freedom,  $p$  is the total number of observed random variables (e.g.,  $p = 3$  in the simple mediation model (3)),  $\kappa$  is a prespecified proportion of cases one wants to downweight, and  $\tau$  is a constant determined by  $\kappa$  such that  $\hat{\boldsymbol{\Sigma}}$  is unbiased for  $\boldsymbol{\Sigma}$ .

The explicit expressions (13) and (14) provide an iterative algorithm for approximating  $\hat{\boldsymbol{\mu}}$  and  $\hat{\boldsymbol{\Sigma}}$ . That is, after the  $t^{\text{th}}$  iteration,  $\hat{\boldsymbol{\mu}}$  and  $\hat{\boldsymbol{\Sigma}}$  can be updated as

$$\hat{\boldsymbol{\mu}}^{(t+1)} = \frac{\sum_{i=1}^n u_1\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}^{(t)}, \hat{\boldsymbol{\Sigma}}^{(t)})\} \mathbf{z}_i}{\sum_{i=1}^n u_1\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}^{(t)}, \hat{\boldsymbol{\Sigma}}^{(t)})\}}$$

and

$$\hat{\boldsymbol{\Sigma}}^{(t+1)} = \frac{\sum_{i=1}^n u_2\{d(\mathbf{z}_i, \hat{\boldsymbol{\mu}}^{(t+1)}, \hat{\boldsymbol{\Sigma}}^{(t)})\} (\mathbf{z}_i - \hat{\boldsymbol{\mu}}^{(t+1)})(\mathbf{z}_i - \hat{\boldsymbol{\mu}}^{(t+1)})'}{n},$$

respectively. The algorithm is stopped if the change of  $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  between two consecutive iterations is small enough or a given maximum number of iterations is reached.

Among the three methods for handling non-normal data, the bootstrap method focuses on drawing samples from the empirical data. The estimations of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  are just sample mean and sample variance-covariance matrix from the original data sample, respectively. However, bootstrap method is a nonparametric method, which has no distribution assumptions, so robust estimations of the parameters' standard errors are achieved. Then hypothesis testing can be done based on the bootstrap robust distribution. The robust sandwich-type method focuses on the better estimation of parameters' standard errors. The estimations of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$

are also the same but the standard errors of model parameters have robust estimations. Then  $z$ -test is adopted to conduct hypothesis testing. The robust Huber-type method focus on the re-weighting of samples. Thus,  $\mu$  and  $\Sigma$  as well as the standard errors of model parameters all have robust estimations.

## 2.5. Statistical power analysis

The package **bmemLavaan** can also be used for statistical power analysis based on Monte Carlo simulation. In the power analysis, multivariate data are generated according to a prespecified model and a given sample size by default. The empirical power of a mediation effect is calculated based on a large number of generated datasets and a specified significance level.

The procedure of power analysis is as follows. First, multivariate data are generated according to the prespecified mediation model and the population parameter values given the sample size  $n$ . Second, model parameters are estimated and CI are constructed based on the methods described in Section 2.4. Three types of CIs, namely, normal CI, robust Huber-type CI, and bootstrap CI can be constructed with the chosen significance level  $\alpha$ . Third, significance tests are conducted based on the constructed CIs. Specifically, take the mediation effect as an example, if its CI contains 0, then the mediation effect is not significant in this simulation, otherwise, significant. Fourth, repeat the above three steps  $R$  times. Assume there are  $r$  times that the mediation effect is significant. Then the power for detecting the mediation effect is estimated by  $r/R$ . And the estimate of the standard error for the power is  $\sqrt{r(R-r)/R^3}$ . This is a classic Monte Carlo simulation process. The bias of parameter estimation and the empirical coverage probability can be calculated during the above process, despite they are the focus of power analysis. Specifically, the empirical coverage probability is calculated as the rate that the constructed CI covers the population value. Generally speaking, If the bias of parameter estimation is small and the empirical coverage probability is close to the confidence level  $1 - \alpha$ , then the calculated power can be trusted. Particularly, the package allows the generation of multivariate non-normal data through the [Vale and Maurelli \(1983\)](#) method to evaluate statistical power when non-normal data exist. Technical details for power analysis used in the package can be found in ([Zhang 2014](#)).

## 3. Implementation

We implemented the methods discussed in Section 2 in our R package **bmemLavaan**, which uses the R package **lavaan** ([Rosseel 2012](#)) for model specification. **bmemLavaan** can handle missing data and non-normal data in mediation analysis, as detailed in Section 2.3 and 2.4. The package consists of two main functions, namely **bmem()** for parameter estimation and hypothesis testing and **power.bmem()** for power analysis. We illustrate how to use the syntax of the package in this section using the simple mediation model here shown in Figure 2 and more complex examples will be provided in Section 4.

### 3.1. Data used for illustration

Data used in the example were from 76 families on mothers' education level (ME), home environment (HE), children's mathematical achievement (MATH) and reading recognition ability,

a subset of data from the National Longitudinal Survey of Youth study (Center for Human Resource Research 2009). In this example, the purpose is to investigate whether HE mediates the relationship between ME and MATH. The diagram of the model is given in Figure 2.

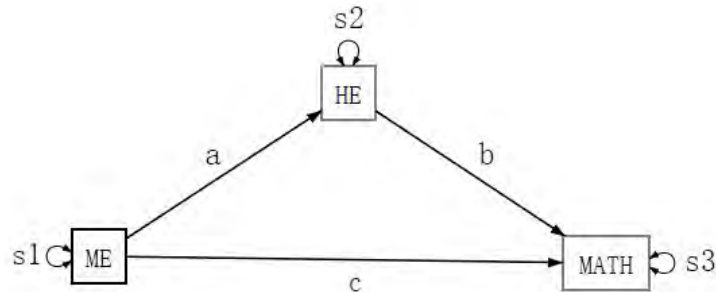


Figure 2: A simple mediation model example. HE, home environment; ME, mother's education; MATH, children's mathematical achievement (Zhang and Wang 2013a).

Here we start with loading the **bmemLavaan** package:

```
R> library("bmemLavaan")
```

### 3.2. Function `bmem()` for model fitting

The function `bmem()` is used to estimate model parameters and mediation effects. The basic arguments of `bmem()` are summarized in Table 1. The main arguments of `bmem()` include `data`, `model`, `method`, and `ci`. The R code for the mediation model depicted in Figure 2 is as follows:

```
R> fit <- bmem(data = data_1, model = model1, v = 1:3, method = "list",
+   ci = "bc", boot = 1000, parallel = TRUE, ncore = 8)
```

Here, `data_1` is the R object for the dataset described in Section 3.1. The argument `model` specifies the mediation model to be estimated, which can be constructed using the syntax of R package **lavaan**. For example, the R code for the simple mediation model in Figure 2 is given below.

```
R> model1 <- '
+ HE ~ a * ME
+ MATH ~ b * HE + c * ME
+ ME ~~ s1 * ME
+ HE ~~ s2 * HE
+ MATH ~~ s3 * MATH
+ ind := a * b'
```

Here, the regression models, variance and covariance, and mediation effect are specified using “~”, “~~”, and “:=”, respectively. The argument `method` specifies the method for handling

| Parameter             | Description                               | Canonical representation                | Remark   |
|-----------------------|---|---|--|
| <code>data</code>     | dataset                                   | A dataframe                             | Should be specified  |
| <code>model</code>    | Mediation model                           | <code>Lavaan</code> model specification | Should be specified by users   |
| <code>v</code>        | column IDs of data                        | Integer[nvar]                           | Default: 1:ncol(data)  |
| <code>method</code>   | Specify a method for missing data problem | Character                               | Default: <code>"tsm1"</code>   |
| <code>ci</code>       | Type of CIs                               | Character                               | Default: <code>"bc"</code>   |
| <code>cl</code>       | Confidence level of CI                    | Double[1]                               | Default: 0.95  |
| <code>boot</code>     | Number of bootstrap replicates            | Integer[1]                              | Default: 1000  |
| <code>robust</code>   | Whether use robust Huber-type method      | Logical                                 | Default: <code>FALSE</code> , otherwise: <code>TRUE</code>                       |
| <code>varphi</code>   | Percent of data to be downweighted        | Double[1]                               | Default: 0.1   |
| <code>parallel</code> | Whether use parallel computing            | Logical                                 | Default: <code>FALSE</code> , otherwise: <code>TRUE</code>                       |
| <code>ncore</code>    | Number of cores used                      | Integer[1]                              | Set <code>ncore &gt; 1</code> if <code>parallel</code> is not <code>FALSE</code> |

Table 1: Main arguments for the function `bmem()`. Here `nvar` is the number of variables.

missing data with the options including `"list"` for listwise deletion, `"pair"` for pairwise deletion, `"mi"` for multiple imputation, or `"tsm1"` for two-stage ML as described in Section 2.3),

The argument `ci` specifies the bootstrap method for constructing CIs of model parameters, (i.e., `"norm"` for normal bootstrap CI, `"perc"` for percentile bootstrap CI, `"bc"` for bias-corrected bootstrap CI, or `"bca"` for bias-corrected and accelerated bootstrap CI as described in Section 2.2). In `bmem()`, parallel computing is allowed by setting the argument `parallel` as `TRUE` and `ncore` as an integer value greater than 1.

The default value of the argument `robust` is `FALSE`, which results in parameter estimates and their standard errors based on normal distribution assumption. Alternatively, one can specify `robust` as `TRUE`, which is highly recommended for non-normal data, as it returns robust parameter estimates and their standard errors based on the Huber-type method. When the robust method is used, one can change the downweighting rate using the argument `varphi` which has a default value 0.1.

The results from an analysis can be summarized in a table using the function `summary()`, which is a generic function. The first part of the output includes the basic information for model fitting, such as missing data handling method (`Estimate method`), sample size, the number of bootstraps and the number of successfully converged bootstraps, and the type of confidence interval. The second part includes values of fitting statistics for mediation model that can be used to evaluate the model fit. The third part includes estimation results of model parameters, i.e., point estimates, standard errors, and CIs. For example, the results from the previous analysis results are summarized as follows: the dataset has 76 subjects and the listwise deletion method was used to deal with the missing data problem. There are

1000 requested bootstraps in this mediation analysis and they all are successful. As for the estimation results, from the CIs we can see, the mediation effect is significant and positive, which means home environment indeed mediates the effect of mothers' education level on children's mathematical achievement.

```
R> summary(fit)
```

```
Estimate method:                listwise deletion
Sample size:                    76
Number of request bootstrap draws: 1000
Number of successful bootstrap draws: 1000
Type of confidence interval:    bc
```

```
Values of statistics:
```

|       | Value    | SE     | 2.5%     | 97.5%    |
|-------|----------|--------|----------|----------|
| chisq | 0.000    | 0.000  | 0.000    | 0.000    |
| GFI   | 1.000    | 0.000  | 1.000    | 1.000    |
| AGFI  | 1.000    | 0.000  | 1.000    | 1.000    |
| RMSEA | 0.000    | 0.000  | 0.000    | 0.000    |
| NFI   | 1.000    | 0.000  | 1.000    | 1.000    |
| NNFI  | 1.000    | 0.000  | 1.000    | 1.000    |
| CFI   | 1.000    | 0.000  | 1.000    | 1.000    |
| BIC   | 1055.358 | 24.963 | 1015.474 | 1106.884 |
| SRMR  | 0.000    | 0.000  | 0.000    | 0.000    |

```
Estimation of parameters:
```

|              |     | Estimate | SE    | 2.5%  | 97.5% |
|--------------|-----|----------|-------|-------|-------|
| Regressions: |     |          |       |       |       |
| HE ~         |     |          |       |       |       |
| ME           | (a) | 0.312    | 0.106 | 0.128 | 0.546 |
| MATH ~       |     |          |       |       |       |
| HE           | (b) | 0.629    | 0.310 | 0.094 | 1.265 |
| ME           | (c) | 0.888    | 0.318 | 0.222 | 1.473 |

```
Variances:
```

|      |      |        |       |        |        |
|------|------|--------|-------|--------|--------|
| ME   | (s1) | 3.195  | 0.720 | 2.120  | 5.000  |
| HE   | (s2) | 2.436  | 0.456 | 1.658  | 3.351  |
| MATH | (s3) | 19.665 | 3.913 | 13.204 | 28.514 |

```
Defined parameters:
```

|     |       |       |       |       |       |
|-----|-------|-------|-------|-------|-------|
| a*b | (ind) | 0.196 | 0.116 | 0.034 | 0.509 |
|-----|-------|-------|-------|-------|-------|

### 3.3. Function `power.bmem()` for power analysis

Power analysis can be conducted via the function `power.bmem()` based on various CI types (i.e., normal, bootstrap, and robust Huber-type CIs). The basic arguments of `power.bmem()` are summarized in Table 2.

| Parameter             | Description                        | Canonical representation   | Remark  |
|-----------------------|------------------------------------|----------------------------|---|
| <code>model</code>    | Mediation model                    | Lavaan model specification | Should be specified by users  |
| <code>method</code>   | Specify a CI method                | Character                  | Default: "normal"   |
| <code>nobs</code>     | Sample size                        | Integer[1]                 | Default: 100  |
| <code>nrep</code>     | Number of Monte Carlo replications | Integer[1]                 | Default: 1000   |
| <code>nboot</code>    | Number of bootstrap replicates     | Integer[1]                 | Default: 1000   |
| <code>ci</code>       | Obtain bootstrap CIs' method       | Character                  | Default: "perc"   |
| <code>parallel</code> | Whether use parallel computing     | Logical                    | Default: FALSE, otherwise: TRUE   |
| <code>ncore</code>    | Number of cores used               | Integer[1]                 | Set <code>ncore &gt; 1</code> if <code>parallel</code> is not FALSE           |
| <code>skewness</code> | Skewness of observed variables     | Double[nvar]               | Default: NULL   |
| <code>kurtosis</code> | Kurtosis of observed variables     | Double[nvar]               | Default: NULL   |
| <code>ovnames</code>  | Observed variable names            | Character[nvar]            | Match the order of <code>skewness</code> and <code>kurtosis</code> parameters |

Table 2: Main arguments for the function `power.bmem()`. Here `nvar` is the number of variables.

Here is the power analysis R code for the model depicted in Figure 2 based on the norm CI.

```
R> model2 <- '
+ MATH ~ c * ME + start(0) * ME + b * HE + start(0.39) * HE
+ HE ~ a * ME + start(0.39) * ME
+ ME ~~ s1 * ME + start(1) * ME
+ HE ~~ s2 * HE + start(1) * HE
+ MATH ~~ s3 * MATH + start(1) * MATH
+ ind:= a * b'
R> power.result = power.bmem(model = model2, method = "normal", nobs = 76,
+ nrep = 1000, parallel = TRUE, skewness = c(0, 0, 1.3),
+ kurtosis = c(0, 0, 10), ovnames = c("ME", "HE", "MATH"), ncore = 8)
```

In the model, the population parameters are known *a priori* and should be incorporated in the model specification. For example,  $a$  is known to be 0.39, and is specified using “`HE ~ a * ME + start(0.39) * ME`”. The population parameter values can be determined from pilot

studies or the previous literature. The given population parameter values are be used for data simulation and also serve as starting values for model estimation.

The default value of the argument `method` is "normal", which corresponds to the normal CI, while the other two candidates are "boot" and "robust" corresponding to the bootstrap CI and the robust sandwich-type CI, respectively. If bootstrap CI is used (`method="boot"`), two additional arguments `nboot` for the number of bootstraps and `ci` for the type of CI can be specified. The argument `ci` specifies the type of bootstrap CI. The default value of `ci` returns percentile bootstrap CI (`ci="perc"`) while the alternative value returns BC bootstrap CI (`ci="bc"`). Power analysis can be conducted to take into consideration of non-normal data where the skewness and kurtosis of the observed variables can be specified by arguments `skewness` and `kurtosis`. If the observed variable names `ovnames` is provided, they should be provided to match the order of the skewness and kurtosis statistics.

Power analysis results can be summarized in a table using the function `summary()`. The main body of the result table includes the following. The first column `True` lists the true population parameter values in the argument `model`. The second column `Estimate` presents the average parameter estimates across all simulation replications. The third column `MSE` is the average bootstrap standard error. The fourth column `SD` is the standard deviation of the parameter estimates across all replications. The fifth column `Power` gives the power for testing whether a parameter is significantly nonzero. The sixth column `Power.se` provides the standard error of the estimated statistical power. The last column `Coverage` presents the empirical coverage probability of the CIs across all replications. The power for testing the mediation effect is listed at the bottom of the table entitled `Indirect/Mediation Effects`, which is about 0.866 in the present example.

```
R> summary(power.result)
```

Basic information:

|                                   |          |
|-----------------------------------|----------|
| Estimation method                 | ML       |
| Standard error                    | standard |
| Number of requested replications  | 1000     |
| Number of successful replications | 1000     |

|              |     | True  | Estimate | MSE   | SD    | Power | Power.se | Coverage |
|--------------|-----|-------|----------|-------|-------|-------|----------|----------|
| Regressions: |     |       |          |       |       |       |          |          |
| MATH ~       |     |       |          |       |       |       |          |          |
| ME           | (c) | 0.000 | -0.000   | 0.121 | 0.121 | 0.050 | 0.007    | 0.950    |
| HE           | (b) | 0.390 | 0.393    | 0.113 | 0.133 | 0.927 | 0.008    | 0.906    |
| HE ~         |     |       |          |       |       |       |          |          |
| ME           | (a) | 0.390 | 0.387    | 0.115 | 0.117 | 0.912 | 0.009    | 0.944    |
| Variances:   |     |       |          |       |       |       |          |          |
| ME           |     | 1.000 | 0.980    | 0.159 | 0.152 | 1.000 | 0.000    | 0.924    |
| HE           |     | 1.000 | 0.969    | 0.157 | 0.169 | 1.000 | 0.000    | 0.882    |
| MATH         |     | 1.000 | 0.952    | 0.154 | 0.373 | 1.000 | 0.000    | 0.595    |



Indirect/Mediation effects:

|    |       |       |       |       |       |       |       |
|----|-------|-------|-------|-------|-------|-------|-------|
| ab | 0.152 | 0.152 | 0.064 | 0.071 | 0.717 | 0.014 | 0.901 |
|----|-------|-------|-------|-------|-------|-------|-------|

Moreover, the function `power.curve()` can graphically display the power trend against the sample size, which can be used to determine the sample size required for a given power through interpolation. This function has the same arguments as the function `power.bmem()`, but the argument `nobs` should be a vector of sample sizes corresponding to the x-axis of the power curve. An example of the power curve can be found in Section 4.5.

## 4. Applications

We now use several examples to further demonstrate the use of the `bmemLavann` package.

### 4.1. A simple mediation model

We first analyze the **Pollack** dataset (Pollack, Vanepps, and Hayes 2012) using a simple mediation model. The raw data for this example are available from the R package `processR`. The **Pollack** dataset consists of 262 observations on 7 variables. We want to investigate whether the effect of economic stress (`estress`) on entrepreneurs' withdrawal intentions (`withdraw`) is mediated through the entrepreneurs' depressed affect (`affect`) (see Figure 3).

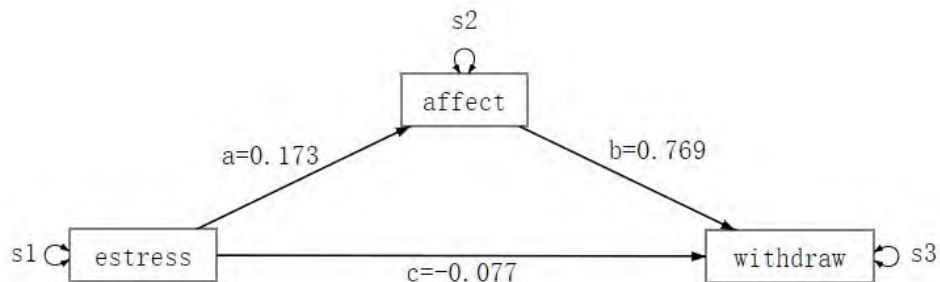


Figure 3: The simple mediation model of Pollack dataset.

Using this example, we compare the default method in function `bmem()` (the normal distribution based method) and the robust Huber-type method with the percentile bootstrap CI. The R code for the analysis given below and the estimation results are presented in Table 3.

```
R> model_s <- '
+   affect ~ a * estress
+   withdraw ~ b * affect + c * estress
+   estress ~~ s1 * estress
+   affect ~~ s2 * affect
+   withdraw ~~ s3 * withdraw
+   ind := a * b'
R> library("processR")
R> data("estress") # estress is raw data of Pollack
R> fit_n <- bmem(data = estress, model = model_s, v = c(2,3,4), ci = 'perc',
```

```

+ boot = 1000, parallel = TRUE, ncore = 8) # default method
R> summary(fit_n)
R> fit_r <- bmem(data = estress, model = model_s, v = c(2,3,4), ci='perc',
+ boot = 1000, robust = TRUE, varphi = 0.1, parallel = TRUE, ncore = 8)
R> # robust method
R> summary(fit_r)

```

| Parameter    | Default |       |                 | Robust |       |                 |
|--------------|---------|-------|-----------------|--------|-------|-----------------|
|              | EST     | SD    | 95% CI          | EST    | SD    | 95% CI          |
| <i>a</i>     | 0.173   | 0.041 | (0.093, 0.250)  | 0.162  | 0.028 | (0.111, 0.218)  |
| <i>b</i>     | 0.769   | 0.138 | (0.486, 1.016)  | 0.899  | 0.135 | (0.659, 1.176)  |
| <i>c</i>     | -0.077  | 0.056 | (-0.183, 0.034) | -0.084 | 0.054 | (-0.199, 0.014) |
| <i>s1</i>    | 2.019   | 0.154 | (1.695, 2.307)  | 1.968  | 0.170 | (1.629, 2.279)  |
| <i>s2</i>    | 0.461   | 0.081 | (0.316, 0.626)  | 0.297  | 0.046 | (0.212, 0.393)  |
| <i>s3</i>    | 1.269   | 0.098 | (1.061, 1.439)  | 1.216  | 0.102 | (1.004, 1.392)  |
| <i>a × b</i> | 0.133   | 0.033 | (0.070, 0.202)  | 0.146  | 0.032 | (0.089, 0.215)  |

Table 3: Estimates (percentile bootstrap CIs) of parameters in Figure 3. Default, the default method in function `bmem()`; Robust, robust Huber-type method. The percentile bootstrap CIs were constructed based on 1000 bootstraps.

There are some differences in the point estimates and CIs of these two methods. Take the variance of `affect` ( $s_2$ ) as an example, the estimate of the default method is clearly larger than the estimate of the robust method, and the corresponding CIs have little overlap. However, there are no difference in terms of the significance of the results. The results of both methods show that the mediation effect ( $a \times b$ ) is statistically significant and positive, demonstrating that the entrepreneurs' depressed affect mediated the effect of economic stress on entrepreneurs' withdrawal intentions.

## 4.2. A multiple-mediator model

To show the capacity of the package `bmemLavaan` for dealing with complex mediation models and missing data problems, we fit a multiple-mediator mediation model to the `Tal_Or` dataset analyzed in Tal-Or, Cohen, Tsfati, and Gunther (2010). The `Tal_Or` data are available from the R package `psych`, which consist of 123 observations on 6 variables. We are interested in whether the effect of a salience manipulation (`cond`) on the intention to buy a product (`reaction`) is mediated through the presumed media influence (`pmi`) and the importance of the message (`import`). The remaining variables `age` and `gender` are used as covariates. The mediation model is depicted in Figure 4.

Since the `Tal_Or` dataset has no missing data, we generate missing data with a MAR mechanism. The missing data probabilities of mediators (`pmi` and `import`) and outcome (`reaction`) depend only on the covariate `age`. Specifically, `pmi` and `import` are chosen to be missing independently with probability 0.5 if and only if `age` is larger than its 70% quantile, and `reaction` was chosen to be missing with probability 0.5 if and only if `age` is smaller than its 20% quantile. As a result, the missing rates for the variables `pmi`, `import`, and `reaction` are about 16%, 12%, and 8%, respectively.

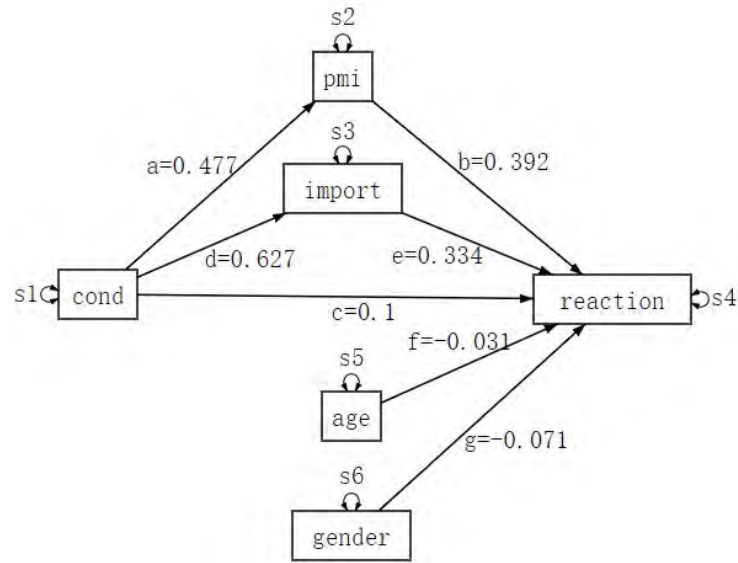


Figure 4: The multiple-mediator model of Tal\_Or dataset.

We apply the four missing data handling methods to the resulting “missing” version of **Tal\_Or** data (denoted by **Tal\_Or\_miss**) to demonstrate the capacity of the package **bmemLavaan** for dealing with missing data. The analysis results of the complete **Tal\_Or** data are used as a baseline for comparison. The R code for the analysis is given below. All results based on 1000 bootstraps are presented in Table 4.

```
R> library("psych")
R> model_m <- '
+ pmi ~ a * cond
+ import ~ d * cond
+ reaction ~ b * pmi + e * import + c * cond + f * age + g * gender
+ cond ~~ s1 * cond
+ pmi ~~ s2 * pmi
+ import ~~ s3 * import
+ reaction ~~ s4 * reaction
+ age ~~ s5 * age
+ gender ~~ s6 * gender
+ ind1 := a * b
+ ind2 := d * e
+ total := a * b + d * e'
R> data("Tal_Or")
R> dataset2 <- Tal_Or_miss # the missing version of Tal_Or dataset
R> fit_2 <- bmem(data = Tal_Or, model = model_m, ci = 'bc',
+ boot = 1000, parallel = TRUE, ncore = 8) # baseline
R> summary(fit_2)
R> fit_2_1 <- bmem(data = dataset2, model = model_m, method = 'list',
+ ci = 'bc', boot = 1000, parallel = TRUE, ncore = 8) # LD method
R> summary(fit_2_1)
```

```

R> fit_2_p <- bmem(data = dataset2, model = model_m, method = 'pair',
+   ci = 'bc', boot = 1000, parallel = TRUE, ncore = 8) # PD method
R> summary(fit_2_p)
R> fit_2_m <- bmem(data = dataset2, model = model1_m, m = 40,
+   method = 'mi', ci = 'bc', boot = 1000, parallel = TRUE, ncore = 8)
R> # MI method with 40 imputations
R> summary(fit_2_m)
R> fit_2_t <- bmem(data = dataset2, model = model_m, method = 'tsml',
+   ci = 'bc', boot = 1000, parallel = TRUE, ncore = 8) # TS-ML method
R> summary(fit_2_t)

```

The results from the four missing data methods share some similarities but also show differences. We focused on the comparisons on the results of parameters of interest, which are the direct effect ( $c$ ) and mediation effects ( $a \times b$ ,  $d \times e$ , and  $a \times b + d \times e$ ). Firstly, in terms of the parameter estimates, the biases, difference from the complete data, of MI and TS-ML methods are much smaller than the biases of LD and PD methods. Specifically, the estimation biases of the direct effect are 0.161, 0.175, 0.028, and 0.026 by LD, PD, MI, and TS-ML methods, respectively. The estimation biases of the total mediation effect are 0.046, 0.073, 0.019, 0.022 by LD, PD, MI, and TS-ML methods, respectively. Secondly, in terms of 95% CIs, because of the impact of estimation biases, the 95% CIs of MI and TS-ML methods is closer to the results of baseline than the 95% CIs of LD and PD methods in most cases. There are some exceptions where LD or PD method outperforms the other methods. For example, in terms of the mediation effect `ind2` ( $d \times e$ ), the 95% CI of PD method is closer to the 95% CI of baseline than the other three methods because the 95% CI of PD method include the 95% CI of baseline while has the smallest length (0.639, 0.527, 0.560, 0.540 by LD, PD, MI, and TS-ML methods, respectively). Thirdly, in terms of significance testing, we can conclude that the direct effect is not significant and the total mediation effect is significant for all four methods, which is the same as in the baseline analysis. But the mediation effect through `pmi` and the mediation effect through `import` are not significant almost in all four methods, which are contrary to the conclusions of baseline. In general, MI and TS-ML methods outperform LD and PD methods in mediation analysis on this missing data problem. However, the computing time of the MI method (91.56s) was about five times that of the other three methods (about 20s), as about  $K = 40$  multiple imputations were used in the MI method.

### 4.3. A latent mediation model

In this example, we show how to model a latent mediator. The **PoliticalDemocracy** dataset (Bollen 1989) used here is available from the R package **Lavaan**. The dataset contains 75 observations on 11 variables that are measures of political democracy and industrialization in developing countries. By design, `x1`, `x2` and `x3` represent the gross national product (GNP) per capital, the inanimate energy consumption per capital, and the percentage of the labor force in industry in 1960, respectively. They measure the degree of industrialization. `y1`, `y2`, `y3` and `y4` represent the expert ratings of the freedom of the press, the freedom of political opposition, the fairness of elections, and the effectiveness of the elected legislature in 1960, respectively. They measure the political democracy in 1960. `y5`, `y6`, `y7` and `y8` represent the same index as `y1`, `y2`, `y3` and `y4` but in 1965. They measure the political democracy in 1965. We are interested in whether the effect of the degree of industrialization in 1960 (`ind60`) on

| Para.                                     | Comp.                 | LD                     | PD                    | MI                    | TS-ML                 |
|---|-----------------------|------------------------|-----------------------|-----------------------|-----------------------|
| <i>a</i>                                  | 0.477 (-0.025, 0.907) | 0.154 (-0.396, 0.643)  | 0.272 (-0.261, 0.748) | 0.339(-0.159, 0.821)  | 0.338(-0.132, 0.859)  |
| <i>b</i>                                  | 0.392(0.239, 0.547)   | 0.378 (0.143, 0.601)   | 0.358 (0.159, 0.562)  | 0.368 (0.154, 0.549)  | 0.370(0.156,0.551)    |
| <i>c</i>                                  | 0.100 (-0.344,0.540)  | -0.061 (-0.596, 0.550) | 0.275 (-0.260, 0.736) | 0.128 (-0.373, 0.645) | 0.126 (-0.385, 0.695) |
| <i>d</i>                                  | 0.627(0.056, 1.189)   | 0.696 (0.002, 1.430)   | 0.478 (-0.174, 1.115) | 0.521(-0.201, 1.108)  | 0.553(-0.078, 1.201)  |
| <i>e</i>                                  | 0.334 (0.181, 0.460)  | 0.343(0.138, 0.532)    | 0.359(0.194, 0.535)   | 0.362(0.168, 0.547)   | 0.364(0.180, 0.550)   |
| <i>f</i>                                  | -0.031(-0.068,0.006)  | 0.039 (-0.080, 0.151)  | 0.044(-0.060, 0.145)  | 0.044(-0.067, 0.166)  | 0.044 (-0.076, 0.149) |
| <i>g</i>                                  | -0.071(-0.547,0.401)  | 0.572 (-0.347, 1.542)  | 0.395(-0.427, 1.338)  | 0.342(-0.492, 1.315)  | 0.344(-0.508, 1.249)  |
| <i>a</i> × <i>b</i>                       | 0.187(0.005,0.420)    | 0.058(-0.152, 0.295)   | 0.097 (-0.068, 0.327) | 0.126(-0.043, 0.362)  | 0.125(-0.040, 0.389)  |
| <i>d</i> × <i>e</i>                       | 0.209(0.024,0.467)    | 0.239 (0.019, 0.658)   | 0.172(-0.055, 0.472)  | 0.188(-0.051, 0.509)  | 0.202(-0.016, 0.524)  |
| <i>a</i> × <i>b</i> + <i>d</i> × <i>e</i> | 0.396(0.063, 0.735)   | 0.350(0.000, 0.817)    | 0.323(-0.010, 0.747)  | 0.415(0.063, 0.835)   | 0.418(0.047, 0.841)   |
| Time                                      | 16.22s                | 16.42s                 | 15.69s                | 91.56s                | 26.18s                |

Table 4: Estimates (BCa confidence intervals) of the parameters in Figure 4 with various missing data handling methods. Para., parameters; Comp., analysis results using complete data; LD, listwise deletion; PD, pairwise deletion; MI, Multiple imputation; TS-ML, two-stage maximum likelihood. The BCa confidence intervals are constructed based on 1000 bootstraps. The last row shows the computing time in seconds.

the political democracy in 1965 (`dem65`) is mediated through the political democracy in 1965

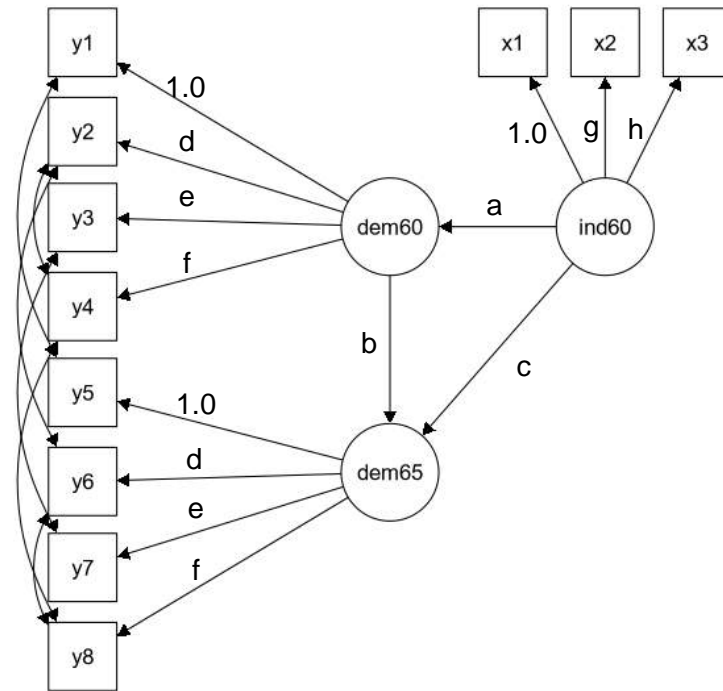


Figure 5: The latent mediator model of PoliticalDemocracy dataset.

We consider the perceptual bootstrap CI with 1000 bootstraps in this example. For comparison, we also conduct the mediation analysis without bootstrap by using `sem()` function in **Lavaan** package, which by default the uses the delta method to estimate the standard error of the indirect effect and utilizes the  $z$ -test for hypothesis testing (also known as Sobel test (Sobel 1982, model 298)). Note that **lavaan** can also conduct bootstrap analysis directly. The R code used here is given below and the results are presented in Table 5.

```
# measurement model
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3 + y4
dem65 =~ y5 + y6 + y7 + y8

# regressions
dem60 ~ a * ind60 + dem60
dem65 ~ c * ind60 + b * dem60

# residual correlations
y1 ~~ y5
y2 ~~ y4 + y6
y3 ~~ y7
y4 ~~ y8
y6 ~~ y8
```

In this example, we use three different formula types: latent

```

+   y6 ~~ y8
+   ind := a * b'
R> fit_b <- bmem(data = PoliticalDemocracy, model = model_1, ci = 'perc',
+   boot = 1000, parallel = TRUE, ncore = 8) # analysis with bootstrap
R> summary(fit_b)
R> fit_s <- sem(model_1, data = PoliticalDemocracy)
P> # analysis without bootstrap
R> summary(fit_s, fit.measures = TRUE)

```

| Parameter           | Bootstrap |       |                | Without bootstrap |       |                                  |
|---------------------|-----------|-------|----------------|-------------------|-------|----------------------------------|
|                     | EST       | SE    | 95% CI         | EST               | SE    | <i>z</i> -value/ <i>p</i> -value |
| <i>a</i>            | 1.471     | 0.373 | (0.734, 2.200) | 1.471             | 0.392 | 3.750/0.000                      |
| <i>b</i>            | 0.865     | 0.069 | (0.738, 1.033) | 0.865             | 0.075 | 11.554/0.000                     |
| <i>c</i>            | 0.600     | 0.242 | (0.150, 1.118) | 0.600             | 0.226 | 2.661/0.008                      |
| <i>d</i>            | 1.191     | 0.149 | (0.928, 1.493) | 1.191             | 0.139 | 8.551/0.000                      |
| <i>e</i>            | 1.175     | 0.130 | (0.972, 1.473) | 1.175             | 0.120 | 9.775/0.000                      |
| <i>f</i>            | 1.251     | 0.143 | (1.007, 1.591) | 1.251             | 0.117 | 10.712/0.000                     |
| <i>g</i>            | 2.180     | 0.149 | (1.924, 2.520) | 2.180             | 0.138 | 15.751/0.000                     |
| <i>h</i>            | 1.818     | 0.146 | (1.546, 2.105) | 1.818             | 0.152 | 11.971/0.000                     |
| <i>a</i> × <i>b</i> | 1.273     | 0.331 | (0.660, 1.958) | 1.273             | 0.358 | 3.559/0.000                      |

Table 5: Estimates results of parameters in Figure 5. Bootstrap, mediation analysis with bootstrap method using `bmem()` function; Without bootstrap, mediation analysis with bootstrap method using `sem()` function. The percentile bootstrap CIs were constructed based on 1000 bootstrap samples.

The 95% CI in the output of `bmem()` function and the *p*-value in the output of the `sem()` function all show that the indirect effect (*a* × *b*) is significant, demonstrating that `dem60` indeed mediated the effect of `ind60` on `dem65`. However, in terms of parameters of interest (*a*, *b*, *a* × *b*), the estimated standard errors of analysis with bootstrap method are smaller than the analysis without bootstrap.

#### 4.4. Power analysis for a simple mediation model

`bmemLavaan` can also be used to conduct power analysis for sample size planning for mediation analysis. To demonstrate its usage, we conducted a power analysis for the model described in Section 4.1. Power analysis can be conducted based on a selected CI (i.e., normal CI, robust CI, and bootstrap CI) that corresponds to a testing procedure for the mediation effect (i.e., *a* × *b*). In the following example code, we obtain the power when the sample size is 262, the same as in the real data. The population parameter values are also set the same as in the results of the real data analysis. For the bootstrap CI, 1,000 bootstraps are used. The power is calculated based on 1000 simulated datasets.

```

R> model_s_p <- '
+   affect ~ a * estress + start(0.173) * estress
+   withdraw ~ b * affect + start(0.769) * affect + c * estress +
+   start(-0.077) * estress

```

```

+   estress ~~ s1 * estress + start(2.019) * estress
+   affect  ~~ s2 * affect  + start(0.461) * affect
+   withdraw ~~ s3 * withdraw + start(1.269) * withdraw
+   ind := a * b'
R> power_n_1 <- power.bmem(model = model_s_p, method = "normal", nobs = 262,
+   nrep = 1000, parallel = TRUE, ncore = 8)
R> summary(power_n_1)
R> power_r_1 <- power.bmem(model = model_s_p, method = "robust", nobs = 262,
+   nrep = 1000, parallel = TRUE, ncore = 8)
R> summary(power_r_1)
R> power_b_1 <- power.bmem(model = model_s_p, method = "boot", nobs = 262,
+   nrep = 1000, nboot = 1000, ci = "perc", parallel = TRUE, ncore = 8)
R> summary(power_b_1)

```

The power analysis results are reported in Table 4.4. Overall, the bootstrap CI based method is evidently more powerful than the robust CI based method, which, in turn, is more powerful than the normal CI based method. For example, the powers for testing direct effect of **estress** on **withdraw** (*c*) are 0.155, 0.184, and 0.189 for the normal, robust, and bootstrap CI based methods, respectively. However, the bootstrap CI method is much more time consuming than the other two methods (15.97, 16.77, and 811.19 seconds, respectively).

| Parameter           | True   | Normal CI |       | Robust CI |       | Bootstrap CI |       |
|---------------------|--------|-----------|-------|-----------|-------|--------------|-------|
|                     |        | Bias      | Power | Bias      | Power | Bias         | Power |
| <i>a</i>            | 0.173  | 0.000     | 0.941 | 0.000     | 0.945 | 0.000        | 0.942 |
| <i>b</i>            | 0.769  | 0.003     | 0.976 | 0.005     | 0.993 | 0.009        | 0.994 |
| <i>c</i>            | -0.077 | 0.002     | 0.155 | 0.005     | 0.184 | 0.000        | 0.189 |
| <i>a</i> × <i>b</i> | 0.133  | 0.000     | 0.890 | 0.001     | 0.885 | 0.002        | 0.958 |

Table 6: Statistical power analysis for the simple mediation model in Figure 3.

To further understand the time consumed by power analysis, we conduct power analyses with different numbers of Monte Carlo replications (**nrep**) and bootstraps (**nboot**) based on the simple mediation model in Figure 3. Table 4.4 shows the time on a computer with 16 CPU cores and 16 GB memory. Clearly, one would expect to use more computing time with a larger number of replications and bootstraps. In practice, we recommend at least 1000 replications and 1,000 bootstraps.

| nrep | Normal | Robust | Bootstrap |         |         |
|------|--------|--------|-----------|---------|---------|
|      |        |        | 100       | 500     | 1000    |
| 100  | 14.73  | 14.15  | 47.03     | 186.28  | 345.08  |
| 500  | 16.15  | 16.96  | 184.81    | 834.43  | 1633.43 |
| 1000 | 18.87  | 19.52  | 352.11    | 1625.31 | 3304.32 |

Table 7: Time consumed to conduct power analyses for the model in Figure 3. The computing time here is presented in seconds. The first column shows different Monte Carlo replications (**nrep** argument). For each value of **nrep**, power analysis is conducted using 3 different numbers of bootstraps (**nboot** argument).



#### 4.5. Power analysis for a multiple-mediator model

Power analysis can be conducted for more complex mediation models such as the multiple-mediator model. The following R code calculates the power based on the population model in 4 when the sample size is 123. The power analysis results are reported in Table 4.5. The power for testing mediation effects of pmi is 0.415, 0.411, 0.590 from the normal, robust, and bootstrap CI based methods, respectively.

```
R> model_m_p <- '
+ pmi ~ a * cond + start(0.477) * cond
+ import ~ d * cond + start(0.627) * cond
+ reaction ~ b * pmi + start(0.392) * pmi + e * import +
+ start(0.334) * import + c * cond + start(0.10) * cond + f * age +
+ start(-0.031) * age + g * gender + start(-0.071) * gender
+ cond ~~ s1 * cond + start(0.249) * cond
+ pmi ~~ s2 * pmi + start(1.675) * pmi
+ import ~~ s3 * import + start(2.893) * import
+ reaction ~~ s4 * reaction + start(1.579) * reaction
+ age ~~ s5 * age + start(33.377) * age
+ gender ~~ s6 * gender + start(0.227) * gender
+ ind1 := a * b
+ ind2 := d * e
+ total := a * b + d * e'
R> power_n_2 <- power.bmem(model = model_m_p, method = "normal", nobs = 123,
+ nrep = 1000, parallel = TRUE, ncore = 8)
R> summary(power_n_2)
R> power_r_2 <- power.bmem(model = model_m_p, method = "robust", nobs = 123,
+ nrep = 1000, parallel = TRUE, ncore = 8)
R> summary(power_r_2)
R> power_b_2 <- power.bmem(model = model_m_p, method = "boot", nobs = 123,
+ nrep = 1000, nboot = 1000, ci = "bc", parallel = TRUE, ncore = 8)
R> summary(power_b_2)
```

A power curve can be generated with the function `power.curve()`. We illustrate its usage by analyzing the mediation effect `ind1` ( $a \times b$ ). The R code for the power curve based on the bootstrap CI is given below. In the function, `seq(100, 1000, by=100)` specifies a set of sample sizes from 100 to 1,000 with an interval 100. The generated power curve for `ind1` is shown in Figure 6. Notice that a power curve can be used to obtain an approximate sample size required for a given power through interpolation. For example, a sample size of about 400 is needed to get a power of 0.95. Note that calculating a power curve can take a significant amount of time. In this example, it took 42905.00 seconds, about 12 hours in total.

```
R> power.curve(model = model_m_p, method = "boot", nobs = seq(100, 1000,
+ by=100), nrep = 1000, nboot = 1000, ci = "bc", parallel = TRUE, ncore = 8)
```

#### 4.6. Power analysis for a latent mediation model

We can also conduct the post-hoc power analysis for the latent mediation model in Figure 5.

| Parameter                 | True   | Normal CI |       | Robust CI |       | Bootstrap CI |       |
|---------------------------|--------|-----------|-------|-----------|-------|--------------|-------|
|                           |        | Bias      | Power | Bias      | Power | Bias         | Power |
| <i>a</i>                  | 0.477  | 0.002     | 0.530 | 0.013     | 0.513 | 0.009        | 0.535 |
| <i>b</i>                  | 0.392  | 0.002     | 0.989 | 0.001     | 0.994 | 0.004        | 0.990 |
| <i>c</i>                  | 0.100  | 0.022     | 0.069 | 0.006     | 0.089 | 0.013        | 0.085 |
| <i>d</i>                  | 0.627  | 0.004     | 0.543 | 0.016     | 0.514 | 0.003        | 0.534 |
| <i>e</i>                  | 0.334  | 0.002     | 0.997 | 0.001     | 0.998 | 0.002        | 0.999 |
| <i>f</i>                  | -0.031 | 0.001     | 0.403 | 0.000     | 0.365 | 0.000        | 0.355 |
| <i>g</i>                  | -0.071 | 0.001     | 0.076 | 0.004     | 0.083 | 0.007        | 0.065 |
| $a \times b$              | 0.187  | 0.001     | 0.415 | 0.006     | 0.411 | 0.001        | 0.590 |
| $d \times e$              | 0.209  | 0.001     | 0.452 | 0.004     | 0.449 | 0.002        | 0.580 |
| $a \times b + d \times e$ | 0.396  | 0.002     | 0.766 | 0.010     | 0.750 | 0.002        | 0.804 |
| Time                      |        | 18.28     |       | 20.28     |       | 4295.00      |       |

Table 8: Statistical power analysis for the multiple-mediator mediation model in Figure 4. The last row shows the corresponding methods' computing time in seconds.

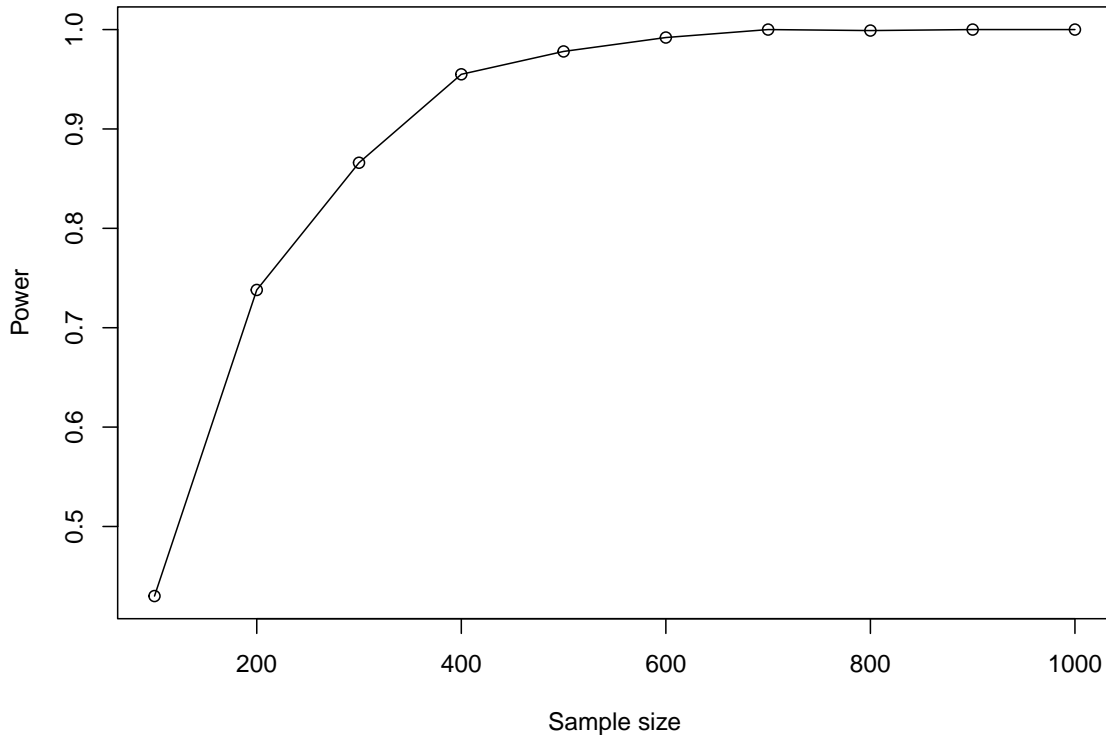


Figure 6: Power curve for testing the mediation effect `ind1` in Example 2. The x-axis is the sample size and y-axis is the power for testing the mediation effect  $a \times b$ .

The corresponding R code is presented as follows. The power analysis results are reported in

Table 4.6. The power for all model parameters are almost 1.0 except for the direct effect  $c$ .

```

R> model_l_p <- '
+   ind60 =~ x1 + g * x2 + start(2.180) * x2 + h * x3 + start(1.818) * x3
+   dem60 =~ y1 + d * y2 + start(1.191) * y2 + e * y3 +
+   start(1.175) * y3 + f * y4 + start(1.251) * y4
+   dem65 =~ y5 + d * y6 + start(1.191) * y6 + e * y7 +
+   start(1.175) * y7 + f * y8 + start(1.251) * y8
+   dem60 ~ a * ind60 + start(1.471) * ind60
+   dem65 ~ c * ind60 + start(0.6) * ind60 + b * dem60 + start(0.865) * dem60
+   y1 ~~ 0.583 * y5
+   y2 ~~ 1.440 * y4 + 2.183 * y6
+   y3 ~~ 0.712 * y7
+   y4 ~~ 0.363 * y8
+   y6 ~~ 1.372 * y8
+   x1 ~~ 0.081 * x1
+   x2 ~~ 0.120 * x2
+   x3 ~~ 0.467 * x3
+   y1 ~~ 1.855 * y1
+   y2 ~~ 7.581 * y2
+   y3 ~~ 4.956 * y3
+   y4 ~~ 3.225 * y4
+   y5 ~~ 2.313 * y5
+   y6 ~~ 4.968 * y6
+   y7 ~~ 3.560 * y7
+   y8 ~~ 3.308 * y8
+   ind := a * b'
R> power_n_3 <- power.bmem(model = model_l_p, method = "normal", nobs = 75,
+   nrep = 1000, parallel = TRUE, ncore = 8)
R> summary(power_n_3)
R> power_r_3 <- power.bmem(model = model_l_p, method = "robust", nobs = 75,
+   nrep = 1000, parallel = TRUE, ncore = 8)
R> summary(power_r_3)
R> power_b_3 <- power.bmem(model = model_l_p, method = "boot", nobs = 75,
+   nrep = 1000, nboot = 1000, ci = "bc", parallel = TRUE, ncore = 8)
R> summary(power_b_3)

```

## 5. Summary and Discussion

A new R package **bmemLavaan** is developed for estimating and testing mediation effects. The package can be used to obtain point estimates and the association confidence intervals based on different methods. Furthermore, it can handle missing data and non-normal data. Specifically, four methods are implemented for handling missing data (i.e., LD, PD, MI, and TS-ML) and three methods are implemented for handling non-normal data (i.e., bootstrap, robust method, and robust Huber-type method). It can also conduct statistical power analysis

| Parameter                | True  | Normal CI |       | Robust CI |       | Bootstrap CI |       |
|--------------------------|-------|-----------|-------|-----------|-------|--------------|-------|
|                          |       | Bias      | Power | Bias      | Power | Bias         | Power |
| <i>a</i>                 | 1.471 | 0.008     | 1.000 | 0.001     | 1.000 | 0.004        | 1.000 |
| <i>b</i>                 | 0.865 | 0.045     | 0.948 | 0.43      | 0.960 | 0.041        | 0.951 |
| <i>c</i>                 | 0.600 | 0.081     | 0.342 | 0.080     | 0.342 | 0.062        | 0.198 |
| <i>d</i>                 | 1.191 | 0.002     | 1.000 | 0.004     | 1.000 | 0.007        | 1.000 |
| <i>e</i>                 | 1.175 | 0.001     | 1.000 | 0.003     | 1.000 | 0.008        | 1.000 |
| <i>f</i>                 | 1.251 | 0.006     | 1.000 | 0.004     | 1.000 | 0.009        | 1.000 |
| <i>g</i>                 | 2.180 | 0.001     | 1.000 | 0.003     | 1.000 | 0.004        | 1.000 |
| <i>h</i>                 | 1.818 | 0.001     | 1.000 | 0.000     | 1.000 | 0.005        | 1.000 |
| $a \times b(\text{ind})$ | 1.272 | 0.076     | 0.942 | 0.083     | 0.953 | 0.063        | 0.951 |
| Time                     |       | 19.86     |       | 22.42     |       | 5409.53      |       |

Table 9: Estimate biases and significance test powers of parameters in Figure 5. The last row shows the corresponding methods’ computing time in seconds.

for mediation analysis. The input and output of **bmemeLavaan** are based on the widely R package **lavaan**, which can greatly reduce the learning curve of the new package.

Although our package **bmemeLavaan** covers a wide range of statistical methods for mediation analyses, it can still be improved in the future. First, to handle missing not at random (MNAR) data, our **bmemeLavaan** package can transform MNAR data into MAR data with appropriate auxiliary variables using the MI or TS-ML method. But it assumes that the missingness of data can be explained by the auxiliary variables. When there exists no such auxiliary variables, estimation biases in mediation analysis can be large (Zhang and Wang 2013a). Selection models have been developed to deal with MNAR data in certain circumstances (e.g., Best, Spiegelhalter, Thomas, and Brayne 1996). Although it is not straightforward to extend those selection models to mediation analyses, it deserves investigation in the future. Second, the methods requiring bootstrap can be very time consuming. In the future, we will extend time-efficient robust methods (e.g., von Oertzen and Brick 2014) to mediation analyses, and evaluate their performance. Third, the robust Huber-type method is now only developed for single group analysis. How to extend the robust Huber-type method to mediation analysis for multiple group data deserves further investigation. Finally, extensions can be made to handle other types of data such as categorical data and survival data.

## Acknowledgment

The work of SM and HZ was partially supported by the Natural Science Foundation of China (No. 12171451, 72091212, and 11771096). The work of ZZ and LW was supported by a grant from the US Department of Education (R305D210023).

## References

- Baron RM, Kenny DA (1986). “The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations.” *Journal of Personality and Social Psychology*, **51**, 1173–1182.

- Best N, Spiegelhalter D, Thomas A, Brayne C (1996). “Bayesian Analysis of Realistically Complex Models.” *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **159**(2), 323–342.
- Bollen KA (1989). *Structural Equations with Latent Variables*, volume 210. John Wiley & Sons.
- Cain MK, Zhang Z, Yuan KH (2017). “Univariate and Multivariate Skewness and Kurtosis for Measuring Nonnormality: Prevalence, Influence and Estimation.” *Behavior Research Methods*, **49**(5), 1716–1735.
- Casella G, Berger RL (2002). *Statistical Inference, 2nd edition*. Pacific Grove, CA: Duxbury Press.
- Center for Human Resource Research (2009). *NLSY79 Child & Young Adult Data Users Guide: A Guide to the 1986–2006 Child Data, 1994–2006 Young Adult Data*. Columbus, Ohio: CHRR.
- Cheung MW (2007). “Comparison of Approaches to Constructing Confidence Intervals for Mediating Effects Using Structural Equation Models.” *Structural Equation Modeling: A Multidisciplinary Journal*, **14**(2), 227–246.
- Enders CK (2003). “Using the Expectation Maximization Algorithm to Estimate Coefficient Alpha for Scales with Item-Level Missing Data.” *Psychological Methods*, **8**(3), 322–337.
- Fox J (2006). “Structural Equation Modeling with the **sem** Package in R.” *Structural Equation Modeling*, **13**(3), 465–486.
- Fritz MS, MacKinnon DP (2007). “Required Sample Size to Detect the Mediated Effect.” *Psychological Science*, **18**(3), 233–239.
- Hayes AF, Scharkow M (2013). “The Relative Trustworthiness of Inferential Tests of the Indirect Effect in Statistical Mediation Analysis: Does Method Really Matter?” *Psychological Science*, **24**(10), 1918–1927.
- Honaker J, King G (2010). “What to Do about Missing Values in Time-Series Cross-Section Data.” *American Journal of Political Science*, **54**(2), 561–581.
- Huber PJ (1981). *Robust Statistics*. New York: Wiley.
- Iacobucci D (2008). *Mediation analysis*. 156. Sage.
- Imai K, Keele L, Tingley D (2010). “A General Approach to Causal Mediation Analysis.” *Psychological Methods*, **15**(4), 309–331.
- Lange T, Hansen KW, Sørensen R, Galatius S (2017). “Applied Mediation Analyses: A Review and Tutorial.” *Epidemiology and Health*, **39**(0), 1–13.
- Little RJ, Rubin DB (2019). *Statistical Analysis with Missing Data, 3rd edition*. Hoboken, NJ: John Wiley & Sons.
- MacKinnon DP (2008). *Introduction to Statistical Mediation Analysis*. London: Taylor & Francis.

- MacKinnon DP, Fairchild AJ (2009). “Current Directions in Mediation Analysis.” *Current Directions in Psychological Science*, **18**(1), 16–20.
- MacKinnon DP, Fairchild AJ, Fritz MS (2007). “Mediation Analysis.” *Annu. Rev. Psychol.*, **58**, 593–614.
- MacKinnon DP, Lockwood CM, Hoffman JM, West SG, Sheets V (2002). “A Comparison of Methods to Test Mediation and Other Intervening Variable Effects.” *Psychological Methods*, **7**(1), 83–104.
- MacKinnon DP, Lockwood CM, Williams J (2004). “Confidence Limits for the Indirect Effect: Distribution of the Product and Resampling Methods.” *Multivariate Behavioral Research*, **39**(1), 99–128.
- Maronna RA (1976). “Robust M-Estimators of Multivariate Location and Scatter.” *The Annals of Statistics*, **4**, 51–67.
- Muthén LK, Muthén BO (1998-2021). *Mplus User’s Guide*. Sixth edition. Muthén and Muthén, Los Angeles, CA. URL <http://www.statmodel.com>.
- Pollack JM, Vanepps EM, Hayes AF (2012). “The Moderating Role of Social Ties on Entrepreneurs’ Depressed Affect and Withdrawal Intentions in Response to Economic Stress.” *Journal of Organizational Behavior*, **33**(6), 789–810.
- Preacher KJ (2015). “Advances in Mediation Analysis: A Survey and Synthesis of New Developments.” *Annual Review of Psychology*, **66**(1), 825–852.
- Preacher KJ, Hayes AF (2004). “SPSS and SAS Procedures for Estimating Indirect Effects in Simple Mediation Models.” *Behavior Research Methods, Instruments, & Computers*, **36**(4), 717–731.
- Robins JM, Greenland S (1992). “Identifiability and Exchangeability for Direct and Indirect Effects.” *Epidemiology*, pp. 143–155.
- 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.
- Salthouse TA (1993). “Speed Mediation of Adult Age Differences in Cognition.” *Developmental Psychology*, **29**(4), 722–742.
- Savalei V, Bentler PM (2009). “A Two-Stage Approach to Missing Data: Theory and Application to Auxiliary Variables.” *Structural Equation Modeling: A Multidisciplinary Journal*, **16**(3), 477–497.
- Savalei V, Falk CF (2014). “Robust Two-Stage Approach Outperforms Robust Full Information Maximum Likelihood with Incomplete Nonnormal Data.” *Structural Equation Modeling: A Multidisciplinary Journal*, **21**(2), 280–302.
- Schafer JL (1997). *Analysis of Incomplete Multivariate Data*. London: Chapman & Hall.
- Shrout PE, Bolger N (2002). “Mediation in Experimental and Nonexperimental Studies: New Procedures and Recommendations.” *Psychological Methods*, **7**(4), 422–445.

- Sobel ME (1982). “Asymptotic Confidence Intervals for Indirect Effects in Structural Equation Models.” *Sociological Methodology*, **13**, 290–312.
- Tal-Or N, Cohen J, Tsfati Y, Gunther AC (2010). “Testing Causal Direction in the Influence of Presumed Media Influence.” *Communication Research*, **37**(6), 801–824.
- Tingley D, Yamamoto T, Hirose K, Keele L, Imai K (2014). “**mediation**: R Package for Causal Mediation Analysis.” *Journal of Statistical Software*, **59**(5), 1–38.
- Tofighi D, MacKinnon DP (2011). “**RM**ediation: An R Package for Mediation Analysis Confidence Intervals.” *Behavior Research Methods*, **43**, 692–700.
- Tofighi D, Thoemmes F (2014). “Single-Level and Multilevel Mediation Analysis.” *The Journal of Early Adolescence*, **34**(1), 93–119.
- Tyler DE (1983). “Robustness and Efficiency Properties of Scatter Matrices.” *Biometrika*, **70**(2), 411–420.
- Vale CD, Maurelli VA (1983). “Simulating Multivariate Nonnormal Distributions.” *Psychometrika*, **48**(3), 465–471.
- VanderWeele TJ (2016). “Mediation Analysis: A Practitioner’s Guide.” *Annual Review of Public Health*, **37**, 17–32.
- von Oertzen T, Brick TR (2014). “Efficient Hessian Computation Using Sparse Matrix Derivatives in RAM Notation.” *Behavior Research Methods*, **46**(2), 385–395.
- Wilcox RR (2011). *Introduction to Robust Estimation and Hypothesis Testing*. San Diego, CA: Academic Press.
- Yu Q, Li B (2017). **mma**: *Multiple Mediation Analysis*. R package version 10.5-1, URL <https://CRAN.R-project.org/package=mma>.
- Yuan KH, Bentler PM (1998). “Structural Equation Modeling with Robust Covariances.” *Sociological Methodology*, **28**(1), 363–396.
- Yuan KH, Chan W, Bentler PM (2000). “Robust Transformation with Applications to Structural Equation Modelling.” *British Journal of Mathematical and Statistical Psychology*, **53**(1), 31–50.
- Yuan KH, Zhang Z (2012). “Robust Structural Equation Modeling with Missing Data and Auxiliary Variables.” *Psychometrika*, **77**(4), 803–826.
- Zhang Z (2014). “Monte Carlo Based Statistical Power Analysis for Mediation Models: Methods and Software.” *Behavior Research Methods*, **46**(4), 1184–1198.
- Zhang Z, Wang L (2013a). “Methods for Mediation Analysis with Missing Data.” *Psychometrika*, **78**(1), 154–184.
- Zhang Z, Wang L (2013b). **bm**em: *Mediation Analysis with Missing Data Using Bootstrap*. R package version 1.5, URL <http://CRAN.R-project.org/package=bmem>.
- Zu J, Yuan KH (2010). “Local Influence and Robust Procedures for Mediation Analysis.” *Multivariate Behavioral Research*, **45**(1), 1–44.

## A. E-step for the EM Algorithm

The E-step of the EM algorithm is to fill in the missing data using their expectations

$$E(z_{ij}|z_{\text{obs}}, U^{(t)}, S^{(t)}) = z_{ij}^{(t)}; i = 1, \dots, N, j = 1, \dots, p + 3, \quad (15)$$

and

$$E(z_{ij}z_{ik}|z_{\text{obs}}, U^{(t)}, S^{(t)}) = z_{ij}^{(t)}z_{ik}^{(t)} + c_{ijk}^{(t)}, \quad (16)$$

where

$$z_{ij}^{(t)} = \begin{cases} z_{ij}, & \text{if } z_{ij} \text{ is observed} \\ E(z_{ij}|z_{\text{obs}}, U^{(t)}, S^{(t)}), & \text{if } z_{ij} \text{ is missing,} \end{cases} \quad (17)$$

and

$$c_{ijk}^{(t)} = \begin{cases} \text{cov}(z_{ij}, z_{ik}|z_{\text{obs}}, U^{(t)}, S^{(t)}), & \text{if both } z_{ij} \text{ and } z_{ik} \text{ are missing} \\ 0, & \text{otherwise,} \end{cases} \quad (18)$$

with  $j, k = 1, 2, \dots, p+3$  and  $z_{\text{obs}}$  denoting the observed data. The expectation  $E(z_{ij}|z_{\text{obs}}, U^{(t)}, S^{(t)})$  and covariance  $\text{cov}(z_{ij}, z_{ik}|z_{\text{obs}}, U^{(t)}, S^{(t)})$  are readily available from the conditional distribution of the multivariate normal distribution with mean  $U^{(t)}$  and covariance  $S^{(t)}$ .

### Affiliation:

Shuigen Ming  
 Department of Statistics and Finance  
 University of Science and Technology of China  
 E-mail: [msg0@mail.ustc.edu.cn](mailto:msg0@mail.ustc.edu.cn)

Hong Zhang  
 Department of Statistics and Finance  
 University of Science and Technology of China  
 E-mail: [zhangh@ustc.edu.cn](mailto:zhangh@ustc.edu.cn)

Zhiyong Zhang  
 Department of Psychology  
 University of Notre Dame  
 E-mail: [zzhang4@nd.edu](mailto:zzhang4@nd.edu)

Lijuan Wang  
 Department of Psychology  
 University of Notre Dame  
 E-mail: [lwang4@nd.edu](mailto:lwang4@nd.edu)