A Model-Based Imputation Procedure for Multilevel Regression Models With Random Coefficients, Interaction Effects, and Nonlinear Terms

Craig K. Enders, Han Du, and Brian T. Keller
University of California, Los Angeles

Abstract

Despite the broad appeal of missing data handling approaches that assume a missing at random (MAR) mechanism (e.g., multiple imputation and maximum likelihood estimation), some very common analysis models in the behavioral science literature are known to cause bias-inducing problems for these approaches. Regression models with incomplete interactive or polynomial effects are a particularly important example because they are among the most common analyses in behavioral science research applications. In the context of single-level regression, fully Bayesian (model-based) imputation approaches have shown great promise with these popular analysis models. The purpose of this article is to extend model-based imputation to multilevel models with up to 3 levels, including functionality for mixtures of categorical and continuous variables. Computer simulation results suggest that this new approach can be quite effective when applied to multilevel models with random coefficients and interaction effects. In most scenarios that we examined, imputation-based parameter estimates were quite accurate and tracked closely with those of the complete data. The new procedure is available in the Blimp software application for macOS, Windows, and Linux, and the article includes a data analysis example illustrating its use.

Translational Abstract

Multiple imputation is a missing data handling technique that creates several copies of the incomplete data, each with different estimates of the missing values. The researcher analyzes each data set, and the resulting estimates and standard errors are averaged into a single set of results. The primary goal of this paper was to outline a novel multiple imputation approach to multilevel analyses with interactive effects. Multilevel data are exceedingly common throughout psychology and the behavioral sciences, examples of such nested data structures include children within classrooms, individuals within families, employees within workgroups, and repeated measurements within individuals, to name a few. Interactive effects are equally common and occur when the magnitude of an association between two variables is modified by a third variable. Most popular current approaches to handling multilevel missing data produced biased estimates of interactive effects, and our approach addresses this important practical problem. The study used computer simulation to create many artificial data sets with missing values, after which it imputed each data set and examined the accuracy of the resulting estimates. The computer simulation results indicated that the proposed procedure works quite well, with trivial biases in most cases. We provide a software program for MacOS and Windows that implements the imputation strategy, and the paper illustrates its use.

Keywords: missing data, multiple imputation, Bayesian estimation

Supplemental materials: http://dx.doi.org/10.1037/met0000228.supp

A good deal of methodological literature supports missing data handling methods that assume a missing at random (MAR) mechanism whereby the probability of missingness is unrelated to an incomplete variable’s scores after conditioning on the observed data (Little & Rubin, 2002; Rubin, 1976). Full information maximum likelihood estimation and multiple imputation are MAR-based methods that enjoy widespread use in behavioral science applications. When missing values are restricted to the outcome variable, maximum likelihood solutions abound in popular software packages (e.g., mixed modeling packages in SPSS, Stata, R, etc.) and are probably preferable because valid estimates are obtained by simply fitting the analysis model to the observed data (Little, 1992; von Hippel, 2007). For additive analysis models with incomplete explanatory variables (e.g., multiple regression, multilevel models with random intercepts), classic multiple imputation routines are similarly plentiful and effective (Schafer, 1997; Scha-
The structure of the article is as follows. First, we discuss the issue of compatibility, as this helps clarify why conventional imputation approaches fail when applied to analysis models with interactive or nonlinear effects. Second, we provide an overview of model-based imputation in the context of a single-level moderated regression analysis. Third, we outline an extension of model-based imputation that accommodates data structures with up to three levels, and we then show how to accommodate categorical variables in the procedure. Fourth, we report the results from three simulation studies that evaluate the proposed procedure. Finally, we use the Blimp application to apply model-based imputation to a real data analysis.

Compatibility of the Analysis and Imputation Models

An important concern with multiple imputation is whether the distributions implied by an imputation procedure match those induced by the analysis model. This issue, known as compatibility, has roots in mathematical statistics (Arnold, Castillo, & Sarabia, 1999, 2001; Arnold & Press, 1989) and is a topic of recent interest in the missing data literature (Bartlett et al., 2015; Carpenter & Kenward, 2013; Hughes et al., 2014; Liu et al., 2014).

To illustrate the concept of compatibility, consider a linear regression analysis model with an incomplete predictor $X_i$.

$$y_i = \beta_0 + \beta_i(x_i) + \epsilon_i$$

A typical application of multiple imputation uses a reverse linear regression that to define a distribution of missing values, given the outcome variable.

$$x_i = \gamma_0 + \gamma_i(y_i) + \epsilon_i$$

Together, analysis and imputation form a set of conditional models, and these conditional models imply a set of conditional distributions—the analysis model assumes that $Y$ is normal given $X$, and the imputation model assumes that $X$ is normal given $Y$. Compatibility is concerned with whether a set of conditional models and their corresponding distributions relate to one another in a coherent way.

The formal definition of compatibility given by Arnold and colleagues (Arnold et al., 1999, 2001; Arnold & Press, 1989) and more recently by Liu, Gelman, Hill, Su, and Kropko (2014) and Bartlett, Seaman, White, and Carpenter (2015) is complex, so

---

1. Model-based imputation for single-level regression models is available in the R packages “simcfe” (Bartlett & Keogh, 2018) and “mdmb” (Robitzsch & Lüdtke, 2018) and in dedicated Bayesian analysis packages such as OpenBUGS (Zhang & Wang, 2017) and JAGS, among others.

2. The two-level imputation procedure from Goldstein et al. (2014) is available in the REALCOM (Carpenter, Goldstein, & Kenward, 2011) software and the R packages “jomo” (Quartagno & Carpenter, 2018) and “mdmb” (Robitzsch & Lüdtke, 2018).

3. Definition 1 of Liu et al. (2014, p. 160) states the following: “A set of conditional models $g_j(x, | x_{-j}, 0); \theta_j \in \Theta, j = 1, \ldots, p$ is said to be compatible if there exists a joint model $f(x | 0) ; \theta \in \Theta$ and a collection of surjective maps $f_j : \Theta \rightarrow \Theta_j$; $j = 1, \ldots, p$ such that for each $j$, $\theta_j \in \Theta_j$ and $0 \in f_j^{-1}(\theta_j) = \{0; f_j(0) = \theta_j\}$, we have $g_j(x, | x_{-j}, \theta_j) = f_j(x | x_{-j}, 0)$. Otherwise, $g_j ; j = 1, \ldots, p$ is said to be incompatible.”
we sketch the basic ideas here and refer interested readers to these sources for additional details. Distilled at a basic level, the definition of compatibility says that a joint distribution exists, and the conditional distributions induced by this joint distribution are exactly the same as those we specify in our analyses. Returning to the previous example, the bivariate normal distribution has the property that its corresponding conditional distributions are also normal with constant variance. Thus, analysis and imputation models from Equations 1 and 2 are compatible because they are identical to those induced by a bivariate normal joint distribution. The practical implication of compatibility is that the imputation model should generate appropriate imputations for the analysis because the two models are functionally linked to a common joint distribution.

Next, consider a moderated regression model (Aiken & West, 1991; Cohen, Cohen, West, & Aiken, 2002), examples of which abound in the applied literature.

\[ y_i = \beta_0 + \beta_1(x_{1i}) + \beta_2(x_{2i}) + \beta_3(x_{1i})(x_{2i}) + \epsilon_i \]

Further, assume that \( X_1 \) and the product are incomplete. The moderated regression analysis assumes the outcome variable is conditionally normal, given the covariates and their interaction. However, the interactive effect in the analysis model precludes the possibility that \( X_1 \) is normal when conditioning on \( Y \) (Arnold et al., 1999, 2001; Arnold & Press, 1989; Bartlett et al., 2015; Liu et al., 2014; Sarabia, Castillo, & Arnold, 2001; Seaman et al., 2012). As such, an imputation model based on reverse linear regression is incompatible with the moderated regression because it specifies a distribution of missing values that is implausible given the interaction term in the analysis model. This incompatibility is the source of the biases noted in the literature (Bartlett et al., 2015; Enders et al., 2014; Kim et al., 2015; Seaman et al., 2012; Zhang & Wang, 2017). Model-based imputation attempts to remedy this problem by sampling imputations from a set of compatible models.

**Model-Based Imputation for Single-Level Regression**

So that readers can better understand our multilevel imputation scheme, this section summarizes model-based imputation for a single-level moderated regression analysis such as that in Equation 3. The methodology we describe here is closely related to substantive model-compatible imputation and the sequential Bayesian approach from the literature (Bartlett et al., 2015; Goldstein et al., 2014; Grund et al., 2018; Ibrahim et al., 2002; Kim et al., 2018; Kim et al., 2015; Lüdtke, Robitzsch, & West, 2019; Zhang & Wang, 2017).\(^4\) Note that imputation relies on a set of regression models, the parameters of which are obtained via an iterative Bayesian estimation algorithm (the Gibbs sampler). We discuss the estimation steps later in the article, but for now assume that the necessary quantities have been estimated.

A variety of older imputation methods can be applied to the moderated regression analysis (Grund et al., 2018; Kim et al., 2018; Kim et al., 2015; Seaman et al., 2012; van Buuren, 2012; van Buuren et al., 2018; Vink & van Buuren, 2013; von Hippel, 2009), including passive imputation (e.g., impute \( X_1 \) conditional on \( Y \) and \( X_2 \), then compute the product term deterministically) and just-another-variable imputation (e.g., treat the product of \( X_1 \) and \( X_2 \) as variable to be imputed). The limitations of these approaches are well documented, so we refer interested readers to the literature for additional information (e.g., recent work by Kim and colleagues provides a comprehensive evaluation of several imputation strategies; Kim et al., 2018; Kim et al., 2015).

The idea behind model-based imputation is to parameterize the imputation problem as a set of compatible univariate distributions, one of which aligns with the analysis model. To frame the procedure that we adopt throughout the article, consider the analysis model from Equation 3. We motivate our procedure by applying the conditional probability rule to factor the joint distribution of the analysis variables as

\[ p(Y, X_1, X_2) = p(Y | X_1, X_2) \times p(X_1, X_2) \]  

where \( p(Y, X_1, X_2) \) is the joint distribution, \( p(Y | X_1, X_2) \) is the distribution of \( Y \) induced by the analysis model (i.e., a normal distribution, conditional on the covariates and their interaction), and \( p(X_1, X_2) \) is the joint distribution of the covariates. The above expression readily generalizes to a scenario with \( R \) covariates, in which case the factorization is \( p(Y, X_1, \ldots, X_R) = p(Y | X_1, \ldots, X_R) \times p(X_1, \ldots, X_R) \).

Although it is not the only way to do so, we assure that imputation models arising from Equation 4 are mutually compatible by specifying a multivariate normal distribution for the explanatory variables. That is, we do not allow nonlinear relations among covariates. Although conceptually similar to what we are doing, the so-called “sequential” parameterization of the joint distribution (Erler et al., 2017; Erler et al., 2016; Ibrahim et al., 2002; Lüdtke et al., 2019) is somewhat more flexible in that it can accommodate nonlinear relations among covariates. In our view, specifying linear relations among the covariates is not a substantial practical limitation because most substantive researchers would not have a theoretical basis for specifying nonlinear relations among variables that would otherwise have been treated as fixed in a complete-data analysis.

To impute \( X_i \), we must derive its conditional distribution given the other analysis variables. Applying Bayes’ theorem gives the expression from Bartlett et al. (2015) and Kim, Sugar, and Belin (2015), among others.

\[ p(X_i | Y, X_1, X_2) \propto p(Y | X_1, X_2) \times p(X_i | X_1, X_2) \]  

A benefit of assuming multivariate normality for the covariates is that the joint distribution of dependent variable and covariates must exist (a critical component of compatibility) when the analysis model is specified as a linear regression with normal errors and constant variance, as follows (Arnold et al., 1999, 2001; Arnold & Press, 1989; Liu et al., 2014).

\[ x_{1i} = \gamma_0 + \gamma_1 x_{2i} + e_i \]

\[ e_i \sim N(0, \sigma^2_e) \]  

Because \( X_1 \) appears in both terms on the right side of Equation 5—once as a covariate and once as an outcome—the posterior distribution of missing values has a complex form that depends on the product of two normal distributions.

\(^4\) The key differences among these procedures are how they partition the covariate distribution and whether they treat complete covariates as random variables. Our approach specifies all predictor variables as normally distributed variables, regardless of whether they are incomplete.
\[
p(X_i \mid Y, X_j) \propto N(\beta_0 + \beta_1(x_{ij}) + \beta_2(x_{ij}) + \beta_3(x_{ij})_0, \sigma^2_2) \\
\times N(\gamma_0 + \gamma_1(x_{ij}), \sigma^2_3)
\]  
(7)

We can derive the exact distribution of missing values by comparing the product of the two normal kernels to the form of a univariate normal distribution and matching powers of \(X_i\), which gives an expression algebraically equivalent to the one in Kim et al. (2015, p. 1878).

Having illustrated model-based imputation in the context of a familiar single-level regression analysis, we now extend the procedure to multilevel regression models with missing values at any level. To keep the ensuing discussion as simple as possible, we describe the procedure for a two-level analysis, but the extension to three levels is straightforward. To provide an analytic context, consider a two-level random coefficient analysis with a pair of Level 1 covariates and a single Level 2 explanatory variable

\[
y_{ij} = \beta_0 + \beta_1(x_{ij}) + \beta_2(x_{ij}) + \beta_3(x_{ij})_0 + b_{0j} + b_{1j}(x_{ij}) + e_{ij}
\]

(9)

where the \(\beta\)'s are fixed effects, \(b_{0j}\) and \(b_{1j}\) are random intercept and random slope residuals, respectively, for Level 2 cluster \(j\), and \(e_{ij}\) is a within-cluster residual for observation \(i\) in cluster \(j\). For simplicity, we do not center predictor variables in the analysis model, but the resulting imputations can be grand or group mean centered in the subsequent analysis (Enders & Tofghi, 2007; Krefet, de Leeuw, & Aiken, 1995).

This model, which features an interaction between a manifest variable and a random effect (i.e., latent slope variable), has been the focus of recent missing data research (Enders et al., 2019; Enders, Keller, & Levy, 2018; Grund, Lüdtke, & Robitzsch, 2016; Grund et al., 2018; Kunkel & Kaisar, 2017; Lüdtke, Robitzsch, & Grund, 2017). A standard method for imputing \(X_i\) is to specify a “reverse random coefficient model” that features \(Y\) as a random slope predictor of \(X_i\).

\[
x_{ij} = \gamma_0 + \gamma_1(y_{ij}) + \gamma_2(x_{ij}) + \gamma_3(x_{ij}) + u_{0j} + u_{1j}(y_{ij}) + e_{ij}
\]

(10)

Consistent with the moderated regression analysis, this reverse regression model is incompatible with the analysis model because it incorrectly assumes that the conditional distribution of \(X_i\) given \(Y\) is normal. As such, it gives imputations that are implausible given the random coefficient in the analysis model. Later in this section we show that the correct (compatible) conditional distribution is a complex nonlinear function similar to that in Equation 8.

To begin, we specify a multivariate normal distribution for the explanatory variables because this ensures that we can derive a set of mutually compatible imputation models using pairs of regression models, for example, \(p(Y \mid X_j, \ldots, x_k) \propto p(X_j \mid X_{j-1}, \ldots, x_k)\). Further, we divide the \(R\) covariates into a set of \(P\) predictors at Level 1 and a second set of \(Q\) covariates at Level 2. Note that auxiliary variables enter both conditional models in the same way as the substantive covariates, so we do not differentiate substantively motivated predictor variables from auxiliary variables. The joint distribution for a two-level analysis model is

\[
x^{(1)}_i \sim MN(\mu_j, \Sigma_j) \quad x^{(2)}_i \sim MN(\mu, \Sigma_2)
\]

(11)

where \(x^{(i)}_i\) is a \(P\)-element vector of Level 1 scores for observation \(i\), \(\mu_j\) is the corresponding \(P\)-element vector of latent cluster means (i.e., random intercepts) for group \(j\), \(x^{(2)}_i\) is an \(R\)-element between-cluster score vector that includes the \(P\) latent group means in \(\mu\) and the \(Q\) Level 2 covariate scores, \(\mu\) contains the \(R\) grand means, \(\Sigma_j\) is a \(P\) by \(P\) within-cluster covariance matrix, and \(\Sigma_2\) is an \(R\) by \(R\) between-cluster covariance matrix. For simplicity, we focus on two-level analyses, but the extension to three levels is straightforward. In this case, the covariate distribution becomes

\[
x^{(1)}_i \sim MN(\mu_j, \Sigma_j) \quad x^{(2)}_i \sim MN(\mu, \Sigma_2) \quad x^{(3)}_i \sim MN(\mu, \Sigma_3)
\]

(12)

where \(k\) indexes the third level. The online supplemental materials gives a description of the three-level procedure. Note that Equations 11 and 12 are slightly different from recent related work that treats complete covariates as fixed (Erlr et al., 2017; Erlr et al., 2016).

Returning to the random coefficient analysis from Equation 9, the predictors follow a multivariate normal distribution
\[
\begin{align*}
(x_{1ij}^{(1)}, x_{2ij}^{(2)}) & \sim MN(\mu_j, \Sigma_j) \\
(x_{1ij}^{(1)}, x_{2ij}^{(2)}) & \sim MN(\mu, \Sigma_2)
\end{align*}
\]  

where \(x_{1ij}^{(1)} = (x_{1ij}, x_{2ij}), \mu_j = (\mu_{1j}, \mu_{2j}), x_{2ij}^{(2)} = (\mu_{1j}, \mu_{2j}, x_{3ij}), \mu = (\mu_1, \mu_2, \mu_3), \Sigma_j = 2 \times 2 \text{ within-cluster covariance matrix, and} \Sigma_2 = a 3 \times 3 \text{ between-cluster covariance matrix. Importantly, the cluster means in } \mu_j \text{ can be represented as arithmetic averages of the Level 1 scores, or they can be modeled as normally distributed latent variables (Lüdtke, Marsh, Robitzsch, & Trautwein, 2011; Lüdtke et al., 2008; Shi & Raudenbush, 2010). In this context, latent group means are just random intercepts from a multilevel model and should not be confused with latent means from a factor analysis model. Following the recommendation of Grund, Lüdtke, and Robitzsch (2017), we use latent cluster means because these quantities readily accommodate unequal group sizes, whereas taking arithmetic average of Level 1 scores assumes balanced data (Carpenter & Kenward, 2013; Grund, Lüdtke, & Robitzsch, 2017; Resche-Rigon & White, 2018). The online supplemental materials give the full conditional distribution of the latent cluster means for our two-level example.

Next, we must derive the conditional distribution of \(X_i\) given all other predictors, \(X_{-i} = (X_1, ..., X_i, ..., X_n)\). Applying Bayes’ theorem gives \(p(X_i|Y, X_{-i}) = p(Y|X_i, X_{-i})p(X_i|X_{-i})\), which is a more general expression for Equation 5. The multivariate normal assumption for the covariates again ensures that specifying each \(p(X_i|X_{-i})\) as a linear regression with normal errors and constant variance yields mutually compatible imputation models (Arnold et al., 1999, 2001; Arnold & Press, 1989; Liu et al., 2014). Returning to the random coefficient analysis example, the within-cluster associations in \(\Sigma_j\) can equivalently be expressed as a pair of regression models

\[
\begin{align*}
x_{1ij} & = \mu_{1j} + \eta_{1i}(x_{2ij} - \mu_{2j}) + e_{1ij} \\
x_{2ij} & = \mu_{2j} + \eta_{2i}(x_{1ij} - \mu_{1j}) + e_{2ij}
\end{align*}
\]

where the leading subscript on the slope coefficient indexes the outcome variable. Three points are worth noting. First, because predictor variables are centered at their latent group means, \(\eta_{1i}\) and \(\eta_{2i}\) are “pure” within-cluster regression slopes (Kreft et al., 1995; Raudenbush & Bryk, 2002). Second, the group means, \(\mu_{1j}\) and \(\mu_{2j}\), are normally distributed latent variables that are identical to random intercepts (i.e., random effects). Finally, although neither equation appears to condition on \(X_i\), this variable does in fact influence \(X_j\) and \(X_k\) indirectly via their cluster means, \(\mu_{1j}\) and \(\mu_{2j}\). The equations below illustrate this point.

Next, the between-cluster variances and covariances in \(\Sigma_2\) can be modeled with the following set of Level 2 regression models,

\[
\begin{align*}
\mu_{ij} & = \mu_1 + \eta_{1i}(\mu_{2j} - \mu_2) + \eta_{2j}(x_{3ij} - \mu_3) + \xi_{ij} \\
\xi_{ij} & \sim N(0, \sigma^2_{\xi_{ij}})
\end{align*}
\]

\[
\begin{align*}
\mu_{2j} & = \mu_2 + \eta_{2i}(\mu_{1j} - \mu_1) + \eta_{3j}(x_{2ij} - \mu_2) + \xi_{2j} \\
\xi_{2j} & \sim N(0, \sigma^2_{\xi_{2j}})
\end{align*}
\]

\[
\begin{align*}
\mu_3 & = \mu_3 + \eta_{3i}(\mu_{1j} - \mu_1) + \eta_{2j}(\mu_{2j} - \mu_2) + \xi_{3j} \\
\xi_{3j} & \sim N(0, \sigma^2_{\xi_{3j}})
\end{align*}
\]

Two points are worth noting. First, we include regressions for the latent group means (i.e., random intercepts) because these quantities are integral to Level 1 and Level 2 imputation and must be estimated at every iteration of estimation. Second, the explanatory variables in each equation are again centered, such that the grand means function as fixed intercepts. Here again, our procedure is different from previous work (Erler et al., 2017; Erler et al., 2016; Goldstein et al., 2014) because we explicitly model the Level 1 and Level 2 parts of all predictors, treating the latter terms as normally distributed latent variables.

Having defined the necessary regressions, we can now derive the posterior distribution of the missing values in the same manner as we did for a single-level analysis. To illustrate, consider the random slope predictor \(X_i\). As before, \(X_i\)’s distribution has a complex form that depends on the product of two normal distributions (and two sets of model parameters).

\[
\begin{align*}
 p(X_i | Y, X_j, X_k) & \propto p(Y | X_i, X_j, X_k) \times p(X_i | X_j, X_k) \\
 & \propto N(\beta_1 + b_{1ij}, \beta_2 + b_{2ij} + b_{3ij}) + \beta_3(x_{3ij}) \times N(\mu_{ij} + \gamma_{ij}(x_{2ij} - x_{2j}), \sigma^2_{\epsilon_{ij}})
\end{align*}
\]

(19)

Because \(x_{1ij}\) is conditionally normal given the other predictors, we can derive the exact distribution of the missing values by comparing the product of the two normal kernels to the form of a univariate normal distribution and matching powers of \(X_1\), as follows.

\[
x_{1ij\text{miss}} = N\left( \begin{pmatrix} \alpha^2_{\epsilon_{ij}}(\beta_1 + b_{1ij})(x_{3ij} - \beta_0 - b_{3ij}) \end{pmatrix} + \frac{\sigma^2_{\epsilon_{ij}}(\beta_1 + b_{1ij})^2 + \sigma^2_{\epsilon_{ij}}}{\alpha^2_{\epsilon_{ij}}}, \sigma^2_{\epsilon_{ij}} \right)
\]

(20)

Examining the correct conditional distribution highlights the fact that the reverse random coefficient imputation model from Equation 10 is incompatible with the analysis model (e.g., because it assumes constant variance, whereas the correct variance is a nonlinear function). Also, the previous expression highlights that missing values condition on all variables in the analysis, not just the covariates. For this reason, we would expect the procedure to accommodate a range of MAR processes that depend on the analysis variables (e.g., missingness induced by the outcome).

Analogous distributions can be derived for the other explanatory variables, although the form of each distribution will generally depend on the level at which a covariate is measured as well as its role in the analysis. For example, consider the Level 2 predictor

---

5 The Blimp application described later can implement latent or manifest group means, with latent cluster means set by default. In our extensive test simulations, we have rarely observed meaningful differences between the two methods.

6 The generic probability notation may not convey the fact that \(X_i\) conditions on \(X_i\) via its latent group mean in the between-cluster part of the model (see Equation 16). We could instead write the conditional distribution as \(p(X_i | Y, X_j, X_k, X_{1\text{sub}}, X_{2\text{sub}}) \propto p(Y | X_i, X_j, X_k, X_{1\text{sub}}, X_{2\text{sub}}) \times p(X_i | X_j, X_k, X_{1\text{sub}}, X_{2\text{sub}}),\) where \(X_{1\text{sub}}\) and \(X_{2\text{sub}}\) are the between-cluster parts of the covariate.
Because this variable is constant for all observations in Level 2 cluster $j$, its conditional distribution features a product over all observations in that group.

\[ p(X_j | Y, X_1, X_2) = p(Y | X_1, X_2, X_3) \times p(X_1 | X_1, X_2) \]

\[
\prod_{i=1}^{n} N(\beta_0 + b_{ij}, (\beta_0 + b_{ij})_x, X_1, X_2) \\
\times N(\mu_3 + \eta_1(\mu_{ij} - \mu_3), \eta_2(\mu_{ij} - \mu_2), \sigma_2^2)
\]

(21)

In practice, it is more straightforward to use a Metropolis sampling step (Hastings, 1970) to draw imputations from $p(Y | X_1, X_2, X_3)$ because this approach can approximate a target distribution such as Equation 20 by simply evaluating candidate imputations in both normal likelihood functions. This eliminates the need to derive exact distributions for every unique application. As described in the next section, the Metropolis algorithm is one computational step in the Gibbs sampler that generates parameter values, random effects, and latent group means. Interested readers can find the technical details for the Metropolis sampler in the online supplemental materials.

A brief sidebar about centering is warranted given its important role in multilevel analyses. We use latent group mean centering in the covariate models to partition explanatory variables into within- and between-cluster components, and we do not impose any structure on their covariance matrices (e.g., the association between $x_{ij}$ and $x_{ij} - \mu_1$ need not be the same as that between $\mu_1$ and $\mu_2$). For simplicity, we did not center predictors in the analysis model. If the substantive goal seeks to disentangle within- and between-cluster influences of a Level 1 covariate, latent group means can also be added to the $Y$ part of the imputation model (e.g., $\mu_1$ and $\mu_2$ could be specified as Level 2 predictors of $Y$) and the resulting imputations can then be centered at their grand or group means (Enders & Tofighi, 2007; Kreft et al., 1995).

Gibbs Sampler Algorithm for Bayesian Estimation

Model-based imputation requires parameter values and random effect estimates for the substantive model and a set of parameter values and latent group means (i.e., random intercepts) for each explanatory variable. We use an iterative Bayesian estimation algorithm known as the Gibbs sampler to generate these quantities at each computational cycle, and a final Metropolis within Gibbs step draws incomplete covariate scores from their target distributions. The Bayesian paradigm views the model parameters, random effects, latent group means, and missing values as random variables that have a joint distribution, and the Gibbs sampler estimates one quantity at a time, drawing values from a probability distribution that conditions on a prior distribution and the current values of all other variables. In the interest of space, we sketch the major algorithmic steps here and refer interested readers to the online supplemental materials for a technical exposition of full conditional distributions (e.g., the form and parameters of each distribution, default and user-specifiable priors).

The full cadre of estimation steps for a two-level model with continuous variables is given below. The first five steps generate estimates for the substantive analysis model, and the remaining steps target explanatory variables. To simplify the presentation, we index the entire set of covariates as $r = 1, \ldots, R$, such that $r \leq P$ corresponds to either a Level 1 observation or its corresponding Level 2 group mean, and $r > P$ refers to a manifest Level 2 variable. The computational steps for a single iteration $t$ are as follows.

1. Draw regression coefficients in $B^{(0)}$ from $p(B | \cdot)$.
2. Draw the residual variance $\sigma^2_{(0)}$ from $p(\sigma_{(0)}^2 | \cdot)$.
3. Draw random effects $b_j^{(0)}$ from $p(b_j | \cdot)$.
4. Draw the random effect covariance matrix $\Sigma_b^{(0)}$ from $p(\Sigma_b | \cdot)$.
5. Draw missing outcome scores $y_{r(i)}^{(0)}$ from $p(y_{r(i)} | X_r, X_{-r})$.
6. Draw latent cluster means $\mu_r^{(0)}$ from $p(\mu_r | \cdot)$ for $r = 1$ to $P$.
7. Draw the grand mean $\mu_0^{(0)}$ from $p(\mu_0 | \cdot)$ for $r = 1$ to $R$.
8. Draw within-cluster regression coefficients in $\gamma_0^{(0)}$ from $p(\gamma_0 | \cdot)$ for $r = 1$ to $P$.
9. Draw the within-cluster residual variance $\sigma^2_{(0)}$ from $p(\sigma^2_{(0)} | \cdot)$ for $r = 1$ to $P$.
10. Draw between-cluster regression coefficients $\theta_0^{(0)}$ from $p(\theta_0 | \cdot)$ for $r = 1$ to $R$.
11. Draw the between-cluster residual variance $\sigma^2_{(0)}$ from $p(\sigma^2_{(0)} | \cdot)$ for $r = 1$ to $R$.
12. Using a Metropolis sampler, draw missing covariates $X_{r(i)}^{(0)}$ from $p(Y | X_r, X_{-r}) \times p(X_r | X_{-r})$ for $r = 1$ to $R$.

The dot after the vertical pipe conveys the idea that the entire set of variables being conditioned on are fixed at their current values (e.g., the first step draws the substantive model’s regression coefficients from a distribution that conditions on random effects, variance components, and imputed data from the previous iteration). Finally, note that a given model or variable may only require a subset of these steps. For example, if the outcome variable is complete, Steps 1 through 4 are needed for the posterior distributions of the covariates, but imputation Step 5 is omitted.

Categorical Variables

Thus far we have focused on continuous explanatory variables, but model-based imputation readily accommodates incomplete ordinal and nominal variables. We use a cumulative probit model for ordinal variables and a multinomial probit model for nominal responses (Agresti, 2012; Albert & Chib, 1993; Carpenter & Kenward, 2013; Jiao & van Dyk, 2015; Johnson & Albert, 1999; McCulloch & Rossi, 1994). In the interest of space, we describe the procedure for binary and ordered responses here and take up the multinomial probit model in a separate work. Additional details on probit imputation schemes (also known as latent variable imputation) are widely available in the literature (Asparouhov & Muthén, 2010; Carpenter & Kenward, 2013; Enders et al., 2018; Goldstein, Bonnet, & Rocher, 2007; Goldstein, Carpenter, Kenward, & Levin, 2009).
Probit regression imagines discrete responses arising from an underlying normal latent variable distribution (Agresti, 2012; Albert & Chib, 1993; Johnson & Albert, 1999). We denote the discrete and latent versions of covariate \( r \) as \( X_r \) and \( X'_r \), respectively. The probit model defines the underlying latent variable as a z-score, with the linear predictor from a regression model defining the center of the distribution and the variance fixed to establish a scale. For a binary covariate, the model additionally incorporates a threshold parameter, \( \kappa \), that divides the latent distribution into two segments, such that \( X'_r \) is below the threshold when \( X_r \) equals zero and above the threshold when \( X_r \) equals one. This threshold parameter, which is typically fixed at zero to avoid redundancy with the fixed regression intercept, can be viewed as the z-score cutoff above which the discrete score changes from zero to one. The probit model for ordinal variables has an identical formulation but incorporates additional threshold parameters. For example, an ordered categorical variable with \( c = 1, \ldots, C \) response options requires \( C - 1 \) threshold parameters, such that \( X_r = c \) if \( \kappa_{c-1} < X'_r \leq \kappa_c \). In this situation, the first threshold is still fixed at zero, but the remaining thresholds are updated at each iteration of the Gibbs sampler. We use the Metropolis-Hastings step described by Cowles (1996) for this purpose because it converges more quickly than other algorithms (Albert & Chib, 1993).

To illustrate imputation for categorical explanatory variables, reconsider the random coefficient analysis from Equation 9, this time treating the Level 1 covariate \( X_r \) as binary. We previously motivated model-based imputation by assuming a multivariate normal distribution for the explanatory variables because this ensures that we can derive mutually compatible imputation models. For categorical covariates, the normality assumption applies to the underlying latent scores. This necessitates a change to the Level 1 normal distribution, where diagonal elements corresponding to the categorical variables are fixed at unity, as follows:

\[
\begin{bmatrix}
    x_{1ij} \\
    x_{2ij}
\end{bmatrix} \sim \text{MN}(\mathbf{\mu}_i, \Sigma_i) \quad \Sigma_i = \begin{pmatrix}
    \sigma_{X_1i}^2 & \sigma_{X_1X_2i} \\
    \sigma_{X_1X_2i} & 1
\end{pmatrix} \quad (22)
\]

Introducing latent variables changes the within-cluster regressions, which now model associations on the latent metric.

\[
x_{1ij} = \mu_{1ij} + \gamma_1(x_{2ij} - \mu_{2ij}) + e_{1ij} \\
\quad e_{1ij} \sim N(0, \sigma_{e_{1ij}}^2)
\]

\[
x_{2ij} = \mu_{2ij} + \gamma_2(x_{1ij} - \mu_{1ij}) + e_{2ij} \\
\quad e_{2ij} \sim N(0, 1 - \gamma_2^2\sigma_{X_2i}^2)
\]

Importantly, the residual variance in \( X'_2 \)’s equation is no longer a free parameter but is a deterministic subtraction of explained variance from the total within-cluster variance,\(^7\) which is fixed at unity in the joint distribution of the covariates (Equation 22). An alternate parameterization fixes the residual variance in Equation 24 to unity, which then defines total variance as \( 1 + \gamma_2^2\sigma_{X_2i}^2 \). No changes are needed for the between-cluster part of the model, so Equations 16 to 18 also apply to this example.

Given a full sample of latent variable scores, standard Bayesian estimation steps generate the parameter values, random effects, and latent group means required for imputation. The Gibbs sampler algorithm outlined in the previous section is augmented with an additional step that draws latent variable scores for each case, after which it performs the estimation steps treating the synthetic scores as real data. For cases with observed data, latent variable scores are drawn from a truncated normal distribution, such that observed scores of zero and one have latent scores below and above the threshold, respectively (e.g., \( x_{2ij} < \kappa \) if \( x_{2ij} = 0 \), and \( x_{2ij} \geq \kappa \) when \( x_{2ij} = 1 \). Robert (1995) describes an efficient approach for sampling from tails of a truncated normal distribution, but synthetic values can also be generated by repeatedly drawing values from a normal distribution until obtaining a score in the desired range.

For cases with missing data, the final Metropolis sampling step generates latent variable imputations from an unrestricted normal distribution, and it subsequently creates discrete imputes for the analysis model by comparing the continuous values to the threshold(s). As such, the posterior distribution of the missing values has a complex form that now depends on the product of two normal distributions and a function that reflects this categorization process. To illustrate, the posterior distribution of \( X_2 \) is

\[
p(X_2 | X_1, X_3) \propto p(Y | X_1, X_2, X_3) \times p(X'_2 | X_1, X_3) \times p(X_2 | X'_2)
\]

\[
\times N(\beta_0 + b_0j, \gamma_2(x_{1ij} - \mu_{1ij}), 1 - \gamma_2^2\sigma_{X_2i}^2)
\]

\[
\times (I(x_{2ij} = 0)I(x_{2ij} = 1) + I(x_{2ij} < 0)I(x_{2ij} = 0)) \quad (25)
\]

where \( p(X'_2 | X_1, X_3) \) is the distribution induced by the probit model from Equation 24, and the indicator functions that comprise this model ensure that the final term reflect the link between the latent and discrete imputes (i.e., \( X_2 = 1 \) if \( X'_2 \geq \kappa \) and \( X_2 = 0 \) if \( X'_2 < \kappa \)). Consistent with the procedure for continuous variables, we use a Metropolis sampler to draw candidate pairs of latent and discrete imputations, retaining those that are likely to originate from the distribution in Equation 25. The technical details for the Metropolis step are given in the online supplemental materials.

**Simulation Study 1**

**Two-Level Random Coefficient Analysis**

This section describes the first of three Monte Carlo simulation studies used to evaluate model-based imputation. For this simulation, a random coefficient model with a single covariate at each level served as the population model.

\[
y_{ij} = \beta_0 + \beta_1(x_{1ij}) + \beta_2(x_{2ij}) + b_0j + b_1(x_{1ij}) + \epsilon_{ij} \quad (26)
\]

We generated 1,000 artificial data sets within each cell of a design that varied five between-subjects factors: the intraclass correlation of the Level 1 variables (ICC = .10 and .50), number

\(^7\) In Equation 28, \( \sigma_{X_2i}^2 \) is the within-cluster variance of \( X_2 \) because latent group mean centering removes all between-cluster variance from the Level 1 predictors. In the case of a Level 2 or Level 3 categorical variable, \( \sigma_{X_2i}^2 \) would reflect the total between-cluster variation at that level. With two or more predictors in a covariate model, \( \gamma_2^2\sigma_{X_2i}^2 \) is replaced by the analogous matrix expression, \( \boldsymbol{\gamma}' \Sigma_r \boldsymbol{\gamma} \), where \( \Sigma_r \) is the relevant within- or between-cluster covariance matrix.
of Level 2 clusters \((J = 30 \text{ and } 100)\), within-cluster sample size \((n_i = 10 \text{ and } 30)\), missing data rate (15% or 25% missing data on each predictor), and distribution shape of the Level 2 predictor (normally distributed vs. a single-degree of freedom chi-square). These conditions routinely arise in behavioral science research (e.g., the low and high ICCs are typical of cross-sectional and longitudinal data, respectively; the sample size combinations are distributed around the 30/30 rule-of-thumb from the literature), and the missing data rates are high enough to reveal biases for a reverse random coefficient imputation strategy (Enders et al., 2019; Enders et al., 2018; Grund et al., 2016).

We derived population parameters following variance decompositions given in Snijders and Bosker (2012, pp. 116–117) and Rights and Sterba (2018). In particular, Rights and Sterba (2018) define variance-explained effect size measures that we used to derive meaningful parameter values for the simulation. In line with our treatment of explanatory variables (e.g., the normal distributions from Equation 11), these authors treat covariates as random variables that have within-cluster and between-cluster (i.e., Level 1 and Level 2) covariance matrices. This is convenient for modifying the intraclass correlations and defining variance explained effect sizes for the fixed and random parts of the model at each level. To establish a metric for the covariates, we constrained the within-cluster variance of \(X_1\) at one and solved for the between-cluster variance that gives the desired ICC. The total variance of the Level 2 predictor \(X_2\) was also set to one, and its correlation with the between-cluster part of \(X_1\) was \(r = .30\). Finally, we set the total variance of \(Y\) to 100 and solved for the within- and between-cluster variances that gave the same ICC as \(X_1\). These arbitrary constraints on the variances allowed us to specify effect sizes and solve for the corresponding model parameters.

Applying expressions from Rights and Sterba (2018), we chose a value for the slope variance that explained 10% of the within-cluster variance in the outcome, and we derived the fixed Level 1 regression slope that accounted for an additional 10% of this variance. Given these parameters, the residual within-cluster variance is fully determined. Moving to the Level 2 parameters, the slope variance and between-cluster variance of \(X_1\) determine a portion of the between-cluster variance (this is a consequence of grand mean centering). Similarly, the \(\beta_1\) coefficient and the between-cluster variance of \(X_1\) determine part the variance attributable to the fixed effects. Next, we solved for the \(\beta_2\) coefficient that explained an additional 10% of the Level 2 variance, and we computed the residual (intercept) variance by subtracting out explained variance due to the fixed and random effects. Finally, we set the correlation between the random intercepts and slopes at \(r = .30\). Our goal for the simulation was to implement effect sizes that are meaningful to substantive researchers and large enough to expose potential problems with the imputation procedures. Equations 27 and 28 give the resulting population parameters for the ICC = .10 and .50 conditions, respectively.

\[
y_{ij} = 50 + 3.162(x_{1ij}) + .744(x_{2ij}) + b_{0j} + b_{1j}(x_{1ij}) + \varepsilon_{ij}
\]

\[
\begin{pmatrix}
  b_{0j} \\
  b_{1j}
\end{pmatrix} \sim \text{MN}(0, \begin{pmatrix} 7.000 & 2.510 \\ 2.510 & 10.000 \end{pmatrix})
\]

\[
\varepsilon_{ij} \sim N(0, 72)
\]

(27)

All variables or terms on the right side of the population regression equation were first generated by sampling deviation scores from univariate or multivariate normal distributions with the desired variances and covariances. To simulate nonnormal data, we created a chi-square variate by squaring \(X_2\) and rescaling it to have a zero mean and unit variance (on average) prior to inducing its between-cluster correlation with \(X_1\). This resulted in a variable with skewness and excess kurtosis values of approximately 2.0 and 9.0, respectively. After generating the predictor variables and residual terms, the outcome variable was computed as the weighted sum in Equation 27 or 28.

We imposed a 15% or 25% missing data rate on both explanatory variables, such that missing values on \(X_1\) and \(X_2\) were generated as a function of \(Y\) and the \(Y\) group means, respectively.\(^8\) We used a logistic regression equation to link missingness probabilities to the outcome variable. Using a latent variable formulation for logistic regression (Agresti, 2012; Johnson & Albert, 1999), we derived intercept and slope coefficients that produced the desired missing data rate and a pseudo \(R^2\) (McKelvey & Zavoina, 1975) value equal to .50 between the cause of missingness and the latent propensities. Finally, we sampled a missing data indicator for each observation (0 = observed, 1 = missing) from a binomial distribution with success rate equal to that observation’s missingness probability from the logistic regression model, and we deleted scores with indicator values of one.

We used the Blimp 2.0 application (Enders et al., 2018; Keller & Enders, 2019) to apply reverse random coefficient imputation (i.e., conventional fully conditional specification) and model-based imputation.\(^9\) The reverse random coefficient approach uses a model similar to that in Equation 10 to impute \(X_1\), and it uses the \(Y\) and \(X_1\) cluster means (computed as arithmetic averages) as predictors of the missing \(X_2\) scores. The algorithmic steps for this version of fully conditional specification (Blimp offers others) are identical to invoking the mice.impute.2l.pan and mice.impute.2lonly .norm functions in the R package MICE (van Buuren, 2011; van Buuren et al., 2018). The model-based approach is identical to the procedure described earlier except that the simulation model includes a single Level 1 covariate rather than two. Similar model-based procedures for this particular random coefficient model can be implemented in specialized Bayesian analysis programs such as JAGS (Erler et al., 2017; Erler et al., 2016; Grund et al., 2018; Plummer, 2016) and in the R packages

---

\(^8\) In a second set of simulations not reported here, we created Level 1 and Level 2 auxiliary variables, \(A_1\) and \(A_2\), that were responsible for missingness on \(X_1\) and \(X_2\), respectively. We set the correlation between each covariate-auxiliary pair at approximately .50. The model-based imputation results were similar to those presented here, although listwise deletion performed much better because the MAR selection mechanism was much weaker.

\(^9\) We implemented the PRIOR2 options for the substantive analysis and the XPRIOR3 option for the covariate models. The technical Appendix in the online supplemental materials describes these priors in detail.
After examining potential scale reduction factors (Gelman et al., 2014; Gelman & Rubin, 1992) from several artificial data sets, we generated 10 imputations from a Gibbs sampler algorithm with 1,000 burn-in and thinning iterations (i.e., imputed data sets were saved at 1000-iteration increments). We used the complete-data maximum likelihood estimator in Mplus 8 (Muthén & Muthén, 1998–2017) to fit the random slope model to the multiply imputed data sets, and we wrote a custom R program to pool estimates and standard errors. To provide additional comparisons, we also report the complete-data (pre-deletion) and listwise deletion results. The online supplemental materials also present limited simulation results for the full information maximum likelihood estimator for missing data in Mplus, which can accommodate incomplete covariates with numerical integration. Computational tasks were executed on UCLA’s Hoffman2 supercomputer, and we used a Linux shell script to coordinate simulation tasks. All simulation code is available upon request.

Results

Although not part of our main simulation design, we began by examining the large-sample behavior of the imputation methods in data sets with 1,000 Level 2 clusters and 50 observations per cluster. Figure 1 gives trellis plots displaying relative bias values (the difference between an average estimate and its true value expressed as a percent of the true value) for each combination of intraclass correlation, distribution shape, and missing data rate. As a rough heuristic, published simulations often define bias values less than 10% in absolute value as acceptable (Finch, West, & MacKinnon, 1997; Kaplan, 1988), so the figures display these thresholds as dashed lines. As seen in the figure, model-based imputation estimates were effectively indistinguishable from those of the complete data. Perhaps somewhat surprisingly, violating normality by including a skewed Level 2 predictor had no material impact on parameter recovery. In contrast, there was no situation where fully conditional specification (reverse random coefficient imputation) produced uniformly accurate estimates, as the random slope variance was consistently underestimated by 10% to 20% (the covariance was also biased). Previous studies have also noted this bias (Enders et al., 2018; Grund et al., 2016), which is a consequence of incompatibility. Finally, listwise deletion estimates were uniformly and substantially biased. Because there is no reason to expect deletion to improve with smaller samples, we omit this approach from further discussion.

Turning to the full simulation design, Figures 2 and 3 give trellis plots displaying relative bias values with normally distributed predictors and 15% and 25% missing data, respectively. The online supplemental materials give a full set of graphical and tabular displays of the model-based estimates and their bias values. Considered as a whole, model-based imputation estimates tracked closely with those of the complete data, and the procedure was clearly preferable to reverse random coefficient imputation (fully conditional specification). The combination of ICC = .10, small sample size (30 clusters with 10 observations per group), and 25% missing data produced the largest bias values, but performance was still quite good for most parameters. For comparison, Figure 4 displays the relative bias values from the simulation conditions with a skewed Level 2 predictor and 25% missing data rate (see the online supplemental materials for a few set of graphical displays). Consistent with the large-sample simulation, violating normality at Level 2 had little to no impact on parameter recovery. Because the results did not materially differ from those in Figures 2 and 3, no further discussion is warranted. Although imputation is our main focus, we also applied the full information maximum likelihood estimator in Mplus to the normally distributed simulation data because this option would be widely available to substantive researchers. The maximum likelihood estimates were prone to large biases, particularly in the ICC = .50 conditions. These results, which appear in the online supplement, underscore that problems related to nonlinear terms are not restricted to multiple imputation.

The trellis plot in Figure 5 displays frequentist confidence interval coverage values for the 25% missing data rate condition. Because the other conditions were quite similar, we give the full set of plots in the online supplemental materials. Coverage is the proportion of estimates where the 95% symmetric confidence interval included the true parameter, and the dashed lines at .925 and .975 correspond to Bradley’s (1978) so-called liberal criterion. Coverage values lower than the nominal 95% rate reflect Type I error inflation (e.g., a coverage value of 90% suggests a twofold increase in Type I errors), whereas values greater than 95% reflect conservative inference. We restrict our attention to the fixed effects because the literature argues that symmetric confidence intervals are inappropriate for variance estimates (Maas & Hox, 2005; Snijders & Bosker, 2012). As seen in the figure, coverage values for the Level 2 slope coefficient were often too low (about 90%) in conditions with only 30 clusters, but the complete-data coverage rates exhibited the same pattern.

Simulation Study 2

Two-Level Random Coefficient Analysis With a Categorical Predictor

The second simulation study evaluated model-based imputation with a categorical explanatory variable. The random coefficient analysis from Equation 26 again served as the population model, but the Level 2 covariate X2 was a dichotomous variable with equal category proportions, on average. Creating the binary variable required two small changes to the data-generating process, but the simulation design and procedures were otherwise identical. First, to derive the true population parameters, we set the variance of the underlying continuous X2* scores at .25, which is the same value we would expect from the binary variable. Second, after deriving the true parameter values, we generated continuous data and subsequently dichotomized X2 by splitting the underlying continuous distribution at zero (the population mean). We again used Blimp 2 to implement conventional fully conditional specification and model-based imputation, respectively, and we used listwise deletion as an additional comparison.
Figure 1. Average relative bias values from the large-sample simulation featuring a random coefficient model with either a normal or skewed Level 2 predictor. The dashes represent bias values of ±0.10. FCS = fully conditional specification ("reverse random coefficient" imputation); MBI = model-based imputation; LWD = listwise deletion.
Figure 2. Average relative bias values from the simulation featuring a random coefficient model with normally distributed predictors and 15% missing data. The dashes represent bias values of ±0.10. FCS = fully conditional specification ("reverse random coefficient" imputation); MBI = model-based imputation.
Figure 3. Average relative bias values from the simulation featuring a random coefficient model with normally distributed predictors and 25% missing data. The dashes represent bias values of ±0.10. FCS = fully conditional specification ("reverse random coefficient" imputation); MBI = model-based imputation.
Figure 4. Average relative bias values from the simulation featuring a random coefficient model with a skewed Level 2 predictor and 25% missing data. The dashes represent bias values of ±0.10. FCS = fully conditional specification ("reverse random coefficient" imputation); MBI = model-based imputation.
Figure 5. Confidence interval coverage for the fixed effects from a random coefficient model with 25% missing data. The dashes at .925 and .975 represent Bradley’s (1978) so-called liberal criterion. FCS = fully conditional specification ("reverse random coefficient" imputation); MBI = model-based imputation.
Results

As a precursor to the full simulation study, we again examined the large-sample behavior of the imputation methods in data sets with 1,000 Level 2 clusters and 50 observations per cluster. The trellis plots in Figure 6 show that model-based imputation estimates were virtually unbiased, whereas reverse random coefficient imputation (fully conditional specification) consistently underestimated the slope variance by 10% to 20% of its true value. Because the full simulation produced results that were virtually identical to those from the first simulation study, we give the graphical summaries in the online supplement. The results can be summarized as follows: (a) model-based estimates largely tracked with those of the complete data; (b) the combination of small within-cluster sample size and low intraclass correlation produced the largest bias values (similar to those from Figures 2 and 3); and (c) parameter recovery for fully conditional specification was meaningfully worse. Finally, coverage values were comparable to those from Figure 5 (e.g., complete-data and imputation-based estimates of the Level 2 slope coefficient were often too low).

Simulation Study 3

Three-Level Analysis With a Cross-Level Interaction

Thus far we have considered random coefficient models because they have been the focus of recent multilevel imputation literature. However, model-based imputation can accommodate a much broader range of interactive and nonlinear effects. To illustrate its performance in a different context, the final simulation examined a three-level random coefficient analysis with a covariate at each level and a cross-level interaction involving a Level 1 and Level 3 predictor.

\[ y_{ijk} = \beta_0 + \beta_1(x_{ijk}) + \beta_2(x_{2ij}) + \beta_3(x_{3i}) + \beta_4(x_{4ijk}) + b_{0i} + b_{1i}(x_{ijk}) + b_{2i}(x_{2ij}) + b_{3i}(x_{3i}) + b_{4i} + \epsilon_{ijk} \]  

For this analysis, the imputation procedure applies the multivariate normal distribution from Equation 12 to the covariates, which again induces a set of linear regression models at each level. The analysis model functions in the same way as it did before (i.e., defining the distribution of Y given the predictors and interaction). In the interest of space, the online supplemental materials gives the posterior distributions of the missing data for this problem.

The simulation generated 1,000 artificial data sets within each cell of a design that varied four between-subjects factors: the distribution of variance across levels (20% and 50% of the variation distributed across Level 2 and Level 3), number of Level 3 clusters (K = 30 and 100), Level 1 within-cluster sample size (n_j = 10 and 50), and missing data rate for the covariates (15% or 25%). The two variability configurations featured 80% or 50% of Y and X_1’s variance at Level 1, with between-cluster variability distributed evenly across the two higher levels (we refer to these as the ICC = .20 and .50 conditions, respectively). In these same conditions, 80% or 50% of X_2’s variance was assigned to Level 2 with the rest at Level 3. The number of Level 2 clusters within each Level 3 cluster was held constant at five, resulting in a range of sample sizes between 1,500 (30 Level 3 clusters and 10 Level 1 observations per Level 2 group) and 25,000 (100 Level 3 clusters and 50 Level 1 observations per Level 2 group). As before, missingness on the covariates was imposed as a function of the within- and between-cluster parts of the outcome variable. Because this simulation is meant as a proof of concept under ideal circumstances, we do not investigate the impact of nonnormality. It was somewhat surprising that imputing a skewed predictor did not impact parameter recovery, but we would be hesitant to assume that the same holds true when the nonnormal variable is part of an interaction. Keller (2019) provides a thorough investigation of model-based imputation for multilevel interactive effects, and his work examines this issue.

The effect size measures from Rights and Sterba (2018) readily extend to three-level models, so our data-generating process largely mimicked the previous simulations. As before, we set the covariate correlations to r = .30, and we specified fixed effects hierarchically, such that each coefficient incremented the explained variance at a particular level by 10%. Because interaction effects tend to be smaller in magnitude (Chaplin, 1991), we set the cross-level interaction coefficient to explain an additional 5% of the within-cluster variance. Finally, we identified random slope variances that explained 10% of the outcome’s variance at Level 1, and we determined the residual variance at each level by subtracting out the explained portions of variance due to the fixed and random effects. Equations 30 and 31 give the resulting population parameters for the ICC = .20 and .50 conditions, respectively.

\[ y_{ijk} = 49.836 + 3.098(x_{ijk}) + .724(x_{2ij}) + .654(x_{3i}) + 1.549(x_{4ijk})(x_{ijk}) + b_{0i} + b_{1i}(x_{ijk}) + b_{2i}(x_{2ij}) + b_{3i}(x_{3i}) + b_{4i} + \epsilon_{ijk} \]  

\[ (b_{0i}) \sim MN(0, 1.549 \times 1.990) \]  

\[ (b_{1i}) \sim MN(0, 1.549 \times 1.990) \]  

\[ e_{ijk} \sim N(0, 52.000) \]  

(30)

\[ y_{ijk} = 49.740 + 2.449(x_{ijk}) + 1.445(x_{2ij}) + .938(x_{3i}) + 1.225(x_{4ijk})(x_{ijk}) + b_{0i} + b_{1i}(x_{ijk}) + b_{2i}(x_{2ij}) + b_{3i}(x_{3i}) + b_{4i} + \epsilon_{ijk} \]  

\[ (b_{0i}) \sim MN(0, 1.183 \times 2.243) \]  

\[ (b_{1i}) \sim MN(0, 1.183 \times 2.243) \]  

\[ e_{ijk} \sim N(0, 32.500) \]  

(31)

We again used Blimp 2 to apply just-another-variable imputation (i.e., conventional fully conditional specification that treats the cross-level product as a variable to be imputed) and model-based imputation. We are unaware of other software packages that apply these approaches to three-level data. After examining potential scale reduction factors (Gelman et al., 2014; Gelman & Rubin, 1992) from several artificial data sets, we generated 10 imputations from a Gibbs sampler algorithm with 1,000 burn-in and thinning iterations (i.e., imputed data sets were saved at 1,000-iteration increments). As before, we used Mplus’ complete-data maximum likelihood estimator to fit the analysis model to the multiply imputed data sets, and we wrote a custom R program to pool estimates and standard errors. Computational tasks were executed on UCLA’s Shared Hoffman2 Cluster, and we used a Linux shell script to coordinate simulation tasks. All simulation code is available upon request.
Figure 6. Average relative bias values from the large-sample simulation featuring a random coefficient model with an incomplete binary Level 2 predictor. The dashes represent bias values of $\pm 0.10$. FCS = fully conditional specification ("reverse random coefficient" imputation); MBI = model-based imputation; LWD = listwise deletion.
Results

Figures 7 and 8 display average relative bias values for the fixed effects and variance components, respectively. In the interest of space, we focus on the 25% missing data rate and point readers to the online supplemental materials for a full set of graphical displays. For clarity we omit listwise deletion from Figures 7 and 8, but these results are in the online supplemental materials. As seen in the figures, imputing the incomplete product term with fully conditional specification (i.e., just-another-variable imputation) introduced substantial biases that did not dissipate as sample size increased (e.g., the interaction and Level 3 slope coefficients were 20% to 40% lower than their true values, and a number of variance estimates were also distorted). In contrast, model-based imputation estimates were generally quite accurate, with most bias values well below 10%. Estimating this model with only 30 Level 3 units gave negatively biased estimates of the Level 3 slope and the interaction coefficient, but increasing the number of groups to 100 effectively eliminated this issue. Variance estimates were generally quite accurate, and increasing either the within-cluster sample size at Level 1 (e.g., from 10 to 50) or the number of Level 3 groups improved parameter recovery. Not surprisingly, listwise deletion estimates were uniformly and severely biased (see the online supplemental materials).

Finally, Figure 9 displays frequentist confidence interval coverage values for the 25% missing data rate condition, and the online supplemental materials give a full set of plots. Model-based coverage values were almost always within Bradley’s liberal bounds. The complete-data coverage values for the Level 3 slope coefficient and the interaction coefficient were lower (worse) in some cases (e.g., with 30 Level 3 groups), which presumably occurred because missing data uncertainty widened confidence intervals and counteracted the complete-data estimator’s natural tendency for undercoverage.

Real Data Example

This section illustrates model-based imputation in the context of a cluster-randomized trial of a novel math problem-solving intervention (Montague, Krawec, Enders, & Dietz, 2014). The data set, which is available on the Blimp website, features three levels: 6,874 repeated measurements at Level 1 nested in 982 students at Level 2, and students nested in 29 schools at Level 3. Schools were randomly assigned to an intervention (novel curriculum) or control (standard curriculum) condition, such that all students within a given school received the same treatment. To keep the example relatively simple, we ignore nesting at the school level and focus on a two-level model (in fact, there was a relatively small proportion of variation at the school level). The analysis is a two-level growth curve model with a cross-level interaction involving condition and months (i.e., the group-by-time interaction). In addition to an intervention dummy code and its interaction with the temporal predictor, the substantive analysis model features a time-varying math self-efficacy rating scale and a lunch assistance dummy code as covariates.

\[
\text{probsolv}_{i,t} = \beta_0 + \beta_1(\text{matheff}_{i,t}) + \beta_2(\text{month0}_{i,t}) + \beta_3(\text{frlunch}_{i,t}) + \beta_4(\text{condition}_{i,t}) + \beta_5(\text{month0}_{i,t})(\text{condition}_{i,t}) + b_{0i} + b_{1i}(\text{matheff}_{i,t}) + b_{2i}(\text{month0}_{i,t}) + e_{i,t}
\]  

(32)

The percentage of missing observations for each incomplete variable were as follows: problem-solving (11.5%), self-efficacy (11.45%), and lunch assistance status (4.7%). A substantial proportion of problem-solving and self-efficacy scores resulted from planned missingness where the control group was assessed bimonthly instead of monthly. Regardless of mechanism, time-varying variables were imputed by fixing the corresponding temporal predictor at the planned assessment dates.

The Appendix gives the Blimp 2 script for model-based imputation. To add the third level of nesting, one would simply list the school-level identifier on the CLUSTERID line (to our knowledge, Blimp is the only application that can apply this procedure to three-level data structures). The Blimp 2 user guide (Keller & Enders, 2019) provides a detailed description of the scripting language, including a number of new conventions and commands that differ from its predecessor. A byproduct of our procedure is that the software also gives Bayesian estimates (e.g., posterior means and standard deviations) as optional output, so we provide these results as a comparison to illustrate a simple sensitivity analysis. We also used Blimp to implement a second model-based imputation procedure that incorporates gender (complete), standardized math scores (4.7% missing), and the school-level percentage of nonnative English speakers (complete) as auxiliary variables (these additional variables are added to the MODEL line, and gender must be declared as ORDINAL). Finally, although theoretical and computer simulation results generally argue against it, we also included fully conditional specification. The procedure should achieve near-optimal performance in this example because the interacting variables are complete. The software is available as a free download for macOS, Windows, and Linux at www.appliedmissingdata.com/multilevel-imputation, and the full set of analysis scripts and the data are available from the same URL.

As a starting point, it is important to recall that Blimp specifies a multivariate normal joint distribution for the explanatory variables (or their latent scores, in the case of discrete predictors), and the software does not allow users to specify nonlinear relations (e.g., quadratic terms or random coefficients) among pairs of covariates. The so-called sequential version of model-based imputation outlined by Ibrahim, Chen, and Lipsitz (2002) and more recently by Erler and colleagues (Erler et al., 2017; Erler et al., 2016) offers this flexibility, but this approach is not yet available for three-level data structures. Assuming normality for explanatory variables simplifies imputation because the user need only specify the substantive analysis on the MODEL line, and the software automatically constructs the appropriate covariate models. The cross-level product term in the analysis model is specified by joining the interacting variables with an asterisk (e.g., \text{month0\text{condition}}), and the random coefficients are specified by listing the random predictors to the right of the vertical pipe (e.g., MODEL: ... | matheff month0). In a three-level model, Blimp would automatically include random coefficients at Level 2 and Level 3.

Executing the script in the Appendix generates imputations for the lower-order variables that are consistent with the specified interaction effect, but the product term is not added to the filled-in data sets. Rather, Blimp saves uncentered lower-order variables so that users can apply group mean or grand mean centering (or leave variables uncentered) prior to computing interactions. This is in contrast to fully conditional specification, which treats the inter-
Figure 7. Average relative bias values for the fixed effects from the three-level simulation featuring random coefficients, a cross-level interaction, and 25% missing data. The dashes represent bias values of ±0.10. FCS = fully conditional specification (“just another variable” imputation); MBI = model-based imputation.
Figure 8. Average relative bias values for the variance components from the three-level simulation featuring random coefficients, a cross-level interaction, and 25% missing data. The dashes represent bias values of ±0.10. FCS = fully conditional specification (“just another variable” imputation); MBI = model-based imputation.
Figure 9. Confidence interval coverage for the fixed effects from the three-level simulation featuring random coefficients, a cross-level interaction, and 25% missing data. The dashes at .925 and .975 represent Bradley’s (1978) so-called liberal criterion. FCS = fully conditional specification (“just another variable” imputation); MBI = model-based imputation.
action as a variable to be imputed. In this framework, it may be necessary to work with uncentered variables then rescale the imputed product term to approximate a centered solution (Enders et al., 2014). In addition to its theoretical and empirical benefits, model-based imputation is highly convenient because it allows researchers to apply familiar procedures for probing interaction effects (Aiken & West, 1991; Bauer & Curran, 2005).

Blimp can print a table of potential scale reduction factor diagnostics (Gelman & Rubin, 1992), and it optionally saves parameter values that can readily be converted to trace plots in other software (e.g., an R plotting script is provided with the other files for this example). The potential scale reduction factors suggest that the Gibbs sampler converges in approximately 2,000 iterations (i.e., across all models and all parameters, the highest potential scale reduction factor is approximately 1.05). Based on this information, we requested 20 imputations from a Gibbs sampler with 4,000 burn-in and 2,000 thinning iterations (i.e., we saved the first data set after 4,000 computational cycles and saved additional data sets every 2,000 iterations thereafter). The job takes approximately 1 min on a 2018 10-core iMac Pro and about 2 min on a 2017 two-core MacBook Pro. The resulting imputations are compatible with all major analysis packages (theSAVE command outputs imputations in stacked format or as separate files), and the set of files for this example includes analysis scripts for Mplus, R, SPSS, and Stata.

Table 1 gives the parameter estimates from the analysis, with random effect covariances omitted from the table for brevity. The key finding is that the intervention-by-time interaction coefficient is positive and significant, meaning that students in the intervention schools exhibited more rapid problem-solving gains than students in control schools. The Bayesian slope variance estimates are slightly larger than those of the imputation procedures (these quantities can be viewed as estimates taken across more than 30,000 imputations), but differences were generally slight. In particular, fully conditional specification estimates were not dramatically different from those of model-based imputation. As noted previously, fully conditional specification should achieve its optimal performance in this example because the random predictor (months since baseline) and the interacting variables are complete. Theoretical and simulation results suggest that this would not be true in general.

### Discussion

Despite the broad appeal of multiple imputation and other MAR-based approaches, a broad class of regression models featuring interactive effects, polynomial terms, or random coefficients are known to cause bias-inducing problems for popular missing data handling procedures (Bartlett et al., 2015; Enders et al., 2014; Seaman et al., 2012; Zhang & Wang, 2017). A growing body of recent missing data research has focused on fully Bayesian multiple imputation methods that are appropriate for interactive and nonlinear effects (Bartlett et al., 2015; Erler et al., 2017; Erler et al., 2016; Goldstein et al., 2014; Kim et al., 2018; Kim et al., 2015; Zhang & Wang, 2017). Building on these recent developments, this article outlined a model-based multiple imputation methodology designed to handle a wide range of interactive and nonlinear effects in single-level and multilevel regression models with up to three levels. This procedure offers a number of compelling advantages: It (a) has a strong theoretical foundation in the Bayesian framework; (b) readily extends to three-level data structures; (c) uses latent variables (i.e., random effects) to model between-cluster variation and covariation; (d) readily accommodates categorical variables; and (e) produces Bayesian analysis results (e.g., posterior means and standard deviations) as a byproduct of estimation. The primary downside of the procedure is that covariates can exert nonlinear or random influences on the outcome but not each other. The so-called sequential approach to fully Bayesian imputation (Erler et al., 2017; Erler et al., 2016; Ibrahim et al., 2002) can accommodate certain patterns of nonlinearities, but this approach has not been extended to the range of applications that we consider here (e.g., three-level models, categorical variables).

Computer simulation results suggest that model-based imputation is quite effective when applied to multilevel models with random coefficients and interaction effects. In most scenarios that we examined, estimates tracked closely with those of a complete

### Table 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bayes Mean</th>
<th>Bayes SD</th>
<th>MBI Est.</th>
<th>MBI SE</th>
<th>MBI + AV Est.</th>
<th>MBI + AV SE</th>
<th>FCS Est.</th>
<th>FCS SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>46.468</td>
<td>.594</td>
<td>46.516</td>
<td>.577</td>
<td>46.541</td>
<td>.587</td>
<td>46.613</td>
<td>.571</td>
</tr>
<tr>
<td>Self-efficacy slope</td>
<td>.511</td>
<td>.055</td>
<td>.508</td>
<td>.054</td>
<td>.499</td>
<td>.056</td>
<td>.498</td>
<td>.053</td>
</tr>
<tr>
<td>Monthly change slope</td>
<td>.406</td>
<td>.043</td>
<td>.410</td>
<td>.039</td>
<td>.405</td>
<td>.039</td>
<td>.406</td>
<td>.042</td>
</tr>
<tr>
<td>Lunch assistance slope</td>
<td>−.916</td>
<td>.306</td>
<td>−.933</td>
<td>.285</td>
<td>−.899</td>
<td>.290</td>
<td>−.925</td>
<td>.290</td>
</tr>
<tr>
<td>Condition slope</td>
<td>−.467</td>
<td>.270</td>
<td>−.474</td>
<td>.267</td>
<td>−.441</td>
<td>.266</td>
<td>−.487</td>
<td>.272</td>
</tr>
<tr>
<td>Months by condition slope</td>
<td>.330</td>
<td>.053</td>
<td>.326</td>
<td>.051</td>
<td>.331</td>
<td>.051</td>
<td>.331</td>
<td>.054</td>
</tr>
</tbody>
</table>

#### Level 2 (Student-Level) variance components

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bayes Mean</th>
<th>Bayes SD</th>
<th>MBI Est.</th>
<th>MBI SE</th>
<th>MBI + AV Est.</th>
<th>MBI + AV SE</th>
<th>FCS Est.</th>
<th>FCS SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept variance</td>
<td>33.877</td>
<td>8.268</td>
<td>30.225</td>
<td>7.853</td>
<td>31.993</td>
<td>8.120</td>
<td>30.862</td>
<td>8.034</td>
</tr>
<tr>
<td>Self-efficacy slope variance</td>
<td>.447</td>
<td>.105</td>
<td>.400</td>
<td>.098</td>
<td>.408</td>
<td>.100</td>
<td>.394</td>
<td>.100</td>
</tr>
<tr>
<td>Months slope variance</td>
<td>.137</td>
<td>.032</td>
<td>.129</td>
<td>.033</td>
<td>.128</td>
<td>.032</td>
<td>.128</td>
<td>.032</td>
</tr>
</tbody>
</table>

Note.  Bayes = Bayesian posterior summary; MBI = model-based imputation; MBI + AV = model-based imputation with auxiliary variables; FCS = fully conditional specification.
data analysis and either reduced or eliminated biases associated with conventional approaches such as fully conditional specification imputation (and maximum likelihood estimation, although our investigation of this procedure was quite limited). These improvements were particularly salient for a model with a cross-level interaction term.

Importantly, model-based imputation in Blimp assumes that covariates are multivariate normal. It was somewhat surprising that imputing a skewed predictor did not impact parameter recovery, but we are hesitant to assume that the same holds true when the nonnormal variable is part of an interaction. Further, an obvious avenue for future research is to examine the impact of nonnormal Level 1 covariates. Keller (2019) provides a thorough investigation of model-based imputation for multilevel interactive effects, and his work examines these cases. Second, the normality assumption implies that covariates are linearly related, and we did not investigate scenarios where covariates are nonlinearly related. At least for two-level models, it is possible to implement a comparable sequential decomposition of the covariate distribution (Ibrahim et al., 2002) in dedicated Bayesian analysis software such as JAGS (Erler et al., 2017; Erler et al., 2016; Grund et al., 2018). Because that approach allows some covariates to be nonlinear functions of others, it could be more robust to normality violations than our method (Lüdtke et al., 2019). In practice, we suspect that researchers would rarely have the information needed to correctly specify such nonlinearities, but this alternative is important to consider.

This article was an initial foray and is necessarily limited in scope and generalizability. First, we investigated a small subset of nonlinear effects that are possible. The framework can readily accommodate three-way and higher interactions, polynomial effects, and combinations of interactive and polynomial terms. Virtually nothing is known about the application of model-based imputation to these models, and a great deal of research is needed to clarify the procedure’s limitations. Second, our article offered a very limited glimpse into categorical variable imputation. Simulations conducted while developing Blimp suggest that latent variable imputation can work well in a wide range of situations, but the procedure is almost certainly sensitive to the number and type of categorical variables, the distribution of response options, and sample size, to name a few. Finally, the simulation conditions we examined were necessarily limited in scope, and multilevel models offer myriad possibilities for manipulating sample sizes, intraclass correlations, effect sizes, and number of random effects.

In sum, our article outlined a new imputation approach for multilevel models with interactive or nonlinear effects. Limited computer simulations suggest that model-based imputation can offer substantial improvement over conventional imputation methods and maximum likelihood estimation. The Blimp application offers a user-friendly environment for implementing model-based imputation, and the software’s website has a number of resource materials, including analysis scripts for all major software platforms.

References

ENDERS, DU, AND KELLER
Appendix

Blimp 2 Syntax for Real Data Example

Blimp syntax for applying bayesian estimation and model-based imputation with the real data example

DATA: ~/desktop/example.dat;
VARIABLES: school student wave condition eslpct
  ethnic male frlunch achgroup stanmath month0
  month7 probsolv matheff condbymonth;
ORDINAL: frlunch condition;
CLUSTERID: student;
MISSING: 999;
MODEL: probsolv ~ matheff month0 frlunch condition
  month0*condition | matheff month0;
SEED: 90291;
NIMPS: 20;
BURN: 4000;
THIN: 2000;
CHAINS: 10 processors 10;
OPTIONS: estimates latent psr;
SAVE: stacked = ~/desktop/imps.csv;