Huang and Keller
Large-scale Assessments in Education (2025) 13:13
https://doi.org/10.1186/s40536-025-00248-9

Large-scale Assessments in Education

METHODOLOGY Open Access

Working with missing data in large-scale assessments

Francis Huang[1*] and Brian Keller[1]

*Correspondence:
huangf@missouri.edu

[1] University of Missouri, Columbia, MO, USA

Abstract

Missing data are common with large scale assessments (LSAs). A typical approach to handling missing data with LSAs is the use of listwise deletion, despite decades of research showing that approach can be a suboptimal strategy resulting in biased estimates. In order to help researchers account for missing data, we provide a tutorial using R and the freely available Blimp program to impute and analyze multiply imputed datasets.

Keywords: Missing data, Multiple imputation, Large scale assessments, Blimp

Participants in (international) large-scale assessments (LSAs)[1] do not always respond to every survey item resulting in missing data. The clustered nature of the data (e.g., students within schools) adds an additional complication that the missing items can be at different levels in the dataset as well. Furthermore, the complex sampling designs used (e.g. stratified two-stage cluster sampling) necessitate the use of weights in order for results to generalize to the population of interest. Although articles have discussed missingness with regard to measurement and plausible values used in LSAs (Grund et al., 2021; Weirich et al., 2014), papers that discuss how to handle missing data in LSAs are not common.[2]

Given that missing data are ubiquitous in LSAs, a de facto (and seemingly accepted) approach to handling missing data is the use of listwise deletion (LD) and this can be seen in recent applied articles in *Large-scale Assessments in Education* published in 2024. Articles will generally indicate the original sample size and then the reduced sample size. For example:

– Using data from the International Civic and Citizenship Education Study (ICCS) of the International Association for the Evaluation of Educational Achievement (IEA), missing data were reported for 18 countries that ranged from 2 to 26% (Beyer & Brese, 2024).

[1] Although international LSAs are popular, we use the more inclusive term LSA as the methods described also apply to other assessments that are not international in nature.

[2] One study discusses how imputation works using TIMSS data as a case study but focuses only on student-level data using commercial software (Bouhlila & Sellaouti, 2013). Since then, open-source software such as R (R Core Team, 2022) have gained much popularity and other freely available software (that can perform multilevel multiple imputation) such as Blimp (Keller & Enders, 2023) have been developed.

- In a study of five countries using PISA 2018 data, the author reported that the analytic sample consisted of "82%, 78%, 83%, 90%, and 79%" of the original sample because they did not "possess the required data for estimating the statistical models" (Rodríguez De Luque, 2024, p. 6).
- Using TIMSS 2019 data, authors wanted to include a group-level socioeconomic status (SES) variable but 26% of study participants were missing student-level SES, and authors instead opted not to use the aggregated variable (Ye et al., 2024), despite being an important predictor (e.g., see Raudenbush & Bryk, 2002; Singer, 1998).

These are not isolated cases (and other studies do not even mention how they handled missing data) but these examples are only meant to show that even in 2024, LD (or complete case analysis or excluding variables with missing data) is a common approach to handling missing data.

If data are missing completely at random (MCAR; i.e., the missingness is not related to any other observed or unobserved variable), LD would be an acceptable approach, providing unbiased estimates (though with less power to detect effects due to the reduced sample size). However, in reality, the MCAR assumption is not likely to be met and often, missingness may be related to some other observed variable otherwise referred to as missing at random (MAR; e.g., low performing students may be less likely to provide responses).[3] Decades of research (Acock, 2005; Enders, 2010; King et al., 1998) have indicated that LD may not be an optimal strategy for handling missing data (e.g., results can be biased) though this continues to be routinely used when analyzing LSAs, despite being referred to (together with pairwise deletion) as "among the worst methods available for practical applications" (Wilkinson & Task Force on Statistical Inference, 1999, p. 598).

However, a commonly accepted approach for handling missing data is the use of multiple imputation (MI) (Little & Rubin, 2019; Rubin, 1987; Schafer & Graham, 2002) and MI has been extended to work as well with multilevel data (such as LSA datasets). In 2016, research on multilevel MI was described as being in its infancy (Enders et al., 2016). More recently, Grund et al. (2024) indicated that we "have seen the emergence of new and extremely flexible methods for handling missing data that are especially useful for multilevel data" (p. 289) and today, researchers can choose from a growing number of options for handling missing data.

The current manuscript is a software tutorial that provides applied researchers a means to account for missing data using freely available software. We first provide a high-level introduction to MI. Several articles (e.g., Acock, 2005; White et al., 2011) and books (Enders, 2010; Rubin, 1987) have provided detailed explanations of MI over the years and we describe the general workflow of working with imputed datasets. We then provide a tutorial on imputing and analyzing LSA data with missing values with free software using R (R Core Team, 2022) together with Blimp (Keller & Enders, 2023) software. Note that an additional challenge with the use of LSAs is that if ability measures

[3] A third type of missingness is referred to as missing not at random (MNAR) and refers to the situation where the value of the missing variable is related to the variable itself. MNAR is also referred to as nonignorable missingness and in this situation, the missingness is due to the unobserved value.

(e.g., math, reading scores) are to be used in models (as predictors or outcomes), these are usually found in LSA datasets as five to ten plausible values[4] which are already a form of imputed data (Jewsbury et al., 2024) and will then require a set of imputations per plausible value to handle accordingly. In addition, LSA developers routinely prescribe the use of weights in analytic models (Fishbein et al., 2021) which suggests that weights should also be used in some manner in the imputation process. To promote transparency and replicability, syntax and data are available from the first author's Github repository at https://github.com/flh3.

Brief background on multiple imputation

MI involves generating multiple (m) copies of a dataset with imputed values in place of the missing values. The imputed, "filled in" values are sampled (typically) from a conditional distribution of the missing data given the observed data and values will differ in each of the m datasets. With each of the imputed datasets, the analytic model of interest is fit producing m sets of results with different regression coefficient estimates with their standard errors. When fitting a regression model, the multiple m results are then combined into one set of results using Rubin's (1987, p. 76) rules where the regression coefficients are averaged across imputations and the standard errors are combined to account for variability within and between imputations.

Any one of the m complete datasets will have filled in variables which will differ from imputed dataset to imputed dataset. Random (or stochastic) variability is added to each imputed value so "that the data will mimic the uncertainty in relations among variables present in the nonmissing values" (Widaman, 2006, p. 52). Using results from only one complete dataset will treat the specific imputed values as if they were completely observed (i.e., not missing) and will disregard the uncertainty in the fact that any one imputed dataset only contains possible (or plausible) values for the missing data. The pooling process accounts for this uncertainty (or error) in the imputed values and is reflected in the standard errors.[5]

Pooling estimates and standard errors

Combining or pooling results from m regression results use Rubin's (2004) rules to account for imputation variability. The pooled point estimates (e.g., the regression coefficient b) is merely the average of the coefficients ($\overline{b}$) from the m analyses. The uncertainty in the coefficients is represented in the standard errors and pooling the standard errors requires additional work to obtain.

Based on formulas from Schafer and Olsen (1998, p. 557), the pooled standard errors are composed of the within ($\overline{U_b}$) and between imputation (B_b) variance for each b coefficient. The average of the squared standard errors over the m sets of analyses,

[4] The National Assessment for Educational Progress (NAEP) uses 20 plausible values (National Center for Education Statistics, n.d.).

[5] For single-level data, a commonly accepted approach to account for missing data is to use full information maximum likelihood (FIML) and has been referred to as implicit imputation as no imputation is actually performed (Widaman, 2006). Readers can refer to Enders (2010) for a detailed discussion on its use. However, recent studies have warned also against the use of FIML with multilevel data (Grund et al., 2018, pp. 134–135) as is implemented in software such as Mplus. As indicated as well by Enders (2017, p. 5), "a regression analysis with mixtures of categorical and continuous variables is a very simple, yet common, scenario where maximum likelihood estimation is not optimal".

$\overline{U}_b = \frac{\Sigma SE_b^2}{m}$, is the within imputation variance of a regression coefficient. The variance of the regression coefficients over the m sets of analysis is the between imputation variance $B_b = \frac{\left(b-\bar{b}\right)^2}{m-1}$. Combining the between and within imputation variance is done using $T_B = \overline{U}_b + \left(1 + \frac{1}{m}\right)B_b$ and the pooled standard error is $SE_b = \sqrt{T_b}$. If the between group variability is ignored, the standard errors will be underestimated (Schafer, 1999). The estimate ($\bar{b}$) can then be divided by its corresponding standard error that results in the t-statistic used when evaluating statistical significance.

How many datasets to impute?

As to how many (m) datasets to impute, Grund et al. (2018) indicated that generating 20 imputations may be sufficient for most applications in which the goal is to estimate model parameters but as many as 100 or more imputations can be useful for more elaborate hypotheses. The quality of estimates can generally be improved by including variables that can help predict the missing variables as well as increasing the number of imputations used (Grund et al., 2018). White et al. (2011, p. 387) offer a (rough and not universally accepted) rule of thumb that the number of imputations can be based on the percent of incomplete data. For example, if overall, 80% of the data are complete, then a minimum of 20 imputed datasets should be generated (i.e., $[1-0.80]\times 100$). Although decades ago, the suggestion that 5–10 imputations (if the amount of missing data is modest) would suffice (e.g., Rubin, 1987; Schafer, 1999), newer studies emphasize that more imputations are better, especially if researchers are interested in the efficiency and replicability of standard errors (von Hippel, 2018).

Specifying an imputation model with weights

In the imputation phase, the researcher must specify an imputation model that is referred to as congenial with the substantive (or analytic) model (Meng, 1994). In other words, the imputation model should match the analytic model such that all relevant associations among the variables are included. An example of this is that if the substantive model is focused on testing interactions, nonlinear terms, and/or random slopes, these features should be included when imputing the data and not doing so may result in biased estimates (Enders et al., 2020). Imputation approaches that fill in values that do not require a model to be specified may be limited in the types of imputations that can be generated or models analyzed (i.e., only random intercept models).

A common feature when analyzing LSA data is the use of weights to avoid biased estimates (Meinck, 2015; Rabe-Hesketh & Skrondal, 2006). As substantive models involving LSAs use weights, a congenial imputation model should account for this in some manner. One approach is to include the weight as a covariate in the imputation model (Carpenter et al., 2023, p. 279).[6] Other approaches are also possible (e.g., using weights to form strata) though in their analysis of different approaches using an applied example, differences in results between approaches did not vary as much compared to their simulation results (Quartagno et al., 2020). In practice though, when using MI with complex

[6] This may also necessitate the use of interactions with the weight and the observed variables but this may also cause model fitting problems due to the complexity introduced (Carpenter et al., 2023, p. 275).

datasets, weights are often ignored in the imputation phase for simplicity (De Silva et al., 2021). In the succeeding section, we provide a tutorial on how to carry out MI with both weights and plausible values (and the Online Appendix provides an imputation example if plausible values are not necessary).

Software tutorial

For the applied example, we used the Belgian PISA 2018 student and school datasets. The original datasets can be downloaded from https://www.oecd.org/en/data/datasets/pisa-2018-database.html and the merged, reduced dataset can be downloaded from https://github.com/flh3/pubdata/raw/refs/heads/main/miscdata/belgium.rds. The sample consists of 8,475 observations from 288 schools (or around 29.4 students per school). The main R packages used in this example were: the dplyr (Wickham et al., 2020) and tidyr (Wickham, 2021) packages for data management; rblimp (Keller, 2024) together with the Blimp (Keller & Enders, 2023) program for multiple imputation; the WeMix (Bailey et al., 2023) and MLMusingR (Huang, 2025) packages for fitting models and combining results with plausible values, respectively. Other R functions from different packages were also used. Syntax for performing the current analysis are shown and can be modified by readers to suit their own analysis.

Blimp, which is a standalone program, must be downloaded and installed from https://www.appliedmissingdata.com/blimp#blimpdownload (available for Windows, Mac, and Linux systems) and rblimp must be installed afterwards which will allow R to access Blimp functions from within R.[7] rblimp is not currently on CRAN and can be installed using:

```
> remotes::install_github('blimp-stats/rblimp')
```

We focused on investigating how level-1 (student) and level-2 (school; `cntschid`) variables were associated with student math abilities (see Table 1). We use a mix of continuous and categorical predictors and with PISA 2018, mathematics scores used 10 plausible values (pvs).[8] The reason that pvs are provided is that no one student who participates in an LSA completes the entire battery of assessments, which would take too long, but only completes a portion (Huang, 2024). Math abilities were predicted by gender, socio-economic status, student-immigration status, student behavior at the school that hinders learning, and lack of staff at the school hinders instruction.

When comparing model results using secondary datasets (with and without missing data), it is unclear what the results should be since the true values are unknown (unlike with simulated data). Instead, in this example, we selected a variable with minimal missing data, the index of socio, economic, and cultural (`escs`; 2% missing) and removed an additional 10% of the values in such a manner that missingness was related to the first pv in mathematics (missing at random; the dataset can be downloaded from https://github.com/flh3/pubdata/raw/refs/heads/main/miscdata/belgiumwmiss.rds). In the original combined dataset, 90% of cases were complete and, in the instance where additional `escs` values were removed, 82% of cases were complete. We then imputed and

[7] More information about Blimp can be found at: https://github.com/blimp-stats.

[8] To see an example of syntax when pvs are not used, see the online Appendix. The reason the code is slightly different is that imputations per pv are not needed.

Table 1 Descriptive Statistics for the PISA Belgium 2018 Dataset

Variable	Description	*M (SD)*	*n (%)*	% missing
Student-level (n = 8,457)				
`pv1math` to `pv10math`	Ten plausible variables for mathematics	~510 (~95)		0
gender (`st004d01t`)			Male = 4204 (49.6) Female = 4271 (50.4)	0
`escs`	Index of economic, social, and cultural status	0.10 (0.92)		2% / 12%*
immig2 (`immig`)	Immigration status		Native = 6782 (80) 2nd gen = 812 (10) 1st gen = 661 (8)	3%
School level (n = 288)				
lackstaff (`sc017q01na`)	School's instruction hindered by a lack of: teaching staff		Not at all = 42 (15) Very little = 112 (39) To some extent = 95 (33) A lot = 22 (8)	6%
`stubeha`	Student behavior hindering learning	0.29 (0.71)		3%

Unweighted statistics shown. *Additional cases removed

fit multiple models using the dataset with 82% complete data. We compared results of the original dataset, the dataset with additional data removed, and finally, the combined results using multiple imputed datasets (which we refer to as results A, B, and C, respectively). We hypothesized that results from A and C would be most similar and B most dissimilar.

Two-level random intercept multilevel models were fit with maximum likelihood using both student (`w_fstuwt`) and school-level (`w_schgrnrabwt`) weights using robust standard errors. Categorical variables are factors which in R are dummy coded automatically using the first reference group specified.

Fitting multilevel models with the data

The original combined dataset is saved in the `comb` object and the data with the additional missing data is saved in the `wmiss` object. We use the :: notation to specifically call functions in a particular package (e.g., rio [Chan et al., 2018]) without having to load the library first; the necessary package must still be first installed using the `install.packages` function:

```
# load in the datasets
> comb <- rio::import("https://github.com/flh3/pubdata/raw/refs/heads/main/miscdata/belgium.rds",
       trust = TRUE) #combined original dataset
> wmiss <-
rio::import("https://github.com/flh3/pubdata/raw/refs/heads/main/miscdata/belgiumwmiss.rds",
       trust = TRUE) #dataset with additional missing data
```

The models were fit using the `mixPV` function in the MLMusingR package (Huang, 2025) that can be loaded by running:

```
> library(MLMusingR)
```

The `mixPV` function makes use of the `mix` function in WeMix but simplifies fitting several models with plausible values and can automatically pool results appropriately. The `mixPV` and `mix` functions assume that data are nested and follow multilevel

model syntax commonly used in lme4 (Bates et al., 2015). Weighted multilevel models were fit using both datasets:

```
> library(WeMix)
> l1a <- mixPV(pv1math + pv2math + pv3math + pv4math +
  pv5math + pv6math + pv7math + pv7math +
  pv9math + pv10math ~ gender + escs + immig2 +
  stubeha + lackstaff + (1|cntschid),
  weights = c('w_fstuwt', 'w_schgrnrabwt'),
  data = comb, mc = TRUE)
> l1b <- mixPV(pv1math + pv2math + pv3math + pv4math +
  pv5math + pv6math + pv7math + pv7math +
  pv9math + pv10math ~ gender + escs + immig2 +
  stubeha + lackstaff + (1|cntschid),
  weights = c('w_fstuwt', 'w_schgrnrabwt'),
  data = wmiss, mc = TRUE)
```

Using the `mixPV` function, the pvs are specified separately and weights are specified for each level. Weights are specified in ascending order from the lowest to highest level (e.g., student weights and then school weights). The `mc = TRUE` option allows for faster processing which turns on the multiple core processing option. The results of both models can be seen in Table 2.

Table 2 Comparison of fixed effect model results with differing levels of missing data and using 200 multiply imputed datasets

	90% complete	**82% complete**	**Imputed**
(Intercept)	500.191***	521.194***	493.822***
	(10.178)	(9.072)	(11.541)
Student level			
Female	−20.727***	−18.631***	−17.183***
	(2.401)	(2.415)	(2.906)
escs	17.780***	17.114***	16.008***
	(1.509)	(1.603)	(2.184)
Second-Generation[1]	−22.277***	−20.868***	−26.045***
	(3.697)	(3.719)	(4.605)
First-Generation[1]	−26.651***	−24.650***	−26.376***
	(4.894)	(4.928)	(7.242)
School level			
Student behavior	−36.187***	−32.480***	−36.610***
	(5.458)	(4.909)	(5.863)
Very little[2]	27.253*	11.987	27.341 +
	(13.077)	(12.014)	(14.846)
To some extent[2]	8.999	−2.408	8.791
	(13.492)	(11.362)	(15.158)
A lot[2]	22.648	8.357	21.990
	(15.232)	(13.701)	(18.353)
n	7,676	6,911	8,475
No of plausible values/imputed datasets	10	10	200

[1] Immigration status. Native is the reference group. [2]School instruction hindered by lack of staff. Reference group is not at all. Robust standard errors in parenthesis. + $p < 0.10$. * $p < 0.05$. *** $p < 0.001$

Imputing datasets with plausible values

We inspect the dataset to get a better understanding of what variables have missing data. We can use the `mice::md.pattern` (van Buuren & Groothuis-Oudshoom, 2011) or the `MLMusingR::nmiss` (Huang, 2025) functions to identify the variables.

```
> MLMusingR::nmiss(wmiss) #inspect missing data
Percent missing per variable:
      pv1math       pv2math       pv3math       pv4math       pv5math       pv6math
   0.00000000    0.00000000    0.00000000    0.00000000    0.00000000    0.00000000
      pv7math       pv8math       pv9math      pv10math       stubeha     teachbeha
   0.00000000    0.00000000    0.00000000    0.00000000    0.03445428    0.03445428
       gender          escs        immig2     lackstaff      w_fstuwt w_schgrnrabwt
   0.00000000    0.12235988    0.02595870    0.05699115    0.00000000    0.00000000
     cntschid
   0.00000000

Percent complete cases: 0.8154572
(Minimum) number to impute: 18
```

Of the predictors, only `gender` had complete data and the range of missingness was from 2.6% for `immig2` to 12.2% for `escs`. Of the variables, `immig2`, `gender`, and `lackstaff` are factor variables. There is no missing data for the outcomes of `pv1math` to `pv10math`. As pvs are themselves a form of imputed data (Jewsbury et al., 2024), m datasets can be imputed per pv. The total number of imputed datasets and models fit would then be m x number of pvs. For this tutorial, we impute 20 datasets per pv for a total of 200 datasets. Note: for users starting out, users may want to complete a test case first where a few datasets are imputed (e.g., $m = 2$) as both imputation and pooling take time and allow users to debug and isolate issues if any are found.

Imputing the datasets

In order to impute the datasets when pvs are to be used (either as a predictor or an outcome), the `wmiss` dataset, which is currently in a wide format (i.e., the pvs are listed as unique variables), needs to be converted into a tall (long) format where there is one variable that has the pvs. In order to do that, the `pivot_longer` function in the `tidyr` package can be used:

```
> tall <- pivot_longer(wmiss, pv1math:pv10math, values_to = 'math')
```

As the pvs are contiguous variables in the dataset, we can specify `pv1math:pv10math` which indicates that the `pv1math` to the `pv10math` variable can be transposed from wide to tall. A new variable called `name` (the default variable name if the option is unspecified) is created which has values of `pv1math` to `pv10math`. The variable name for the outcome will be named `math` since `values_to='math'` was specified. The tall dataset now consists of 84,750 observations (i.e., 10 pvs × 8,475 observations). We also run the following syntax which creates several new variables used in the imputation, data management, and model fitting processes:

```
> m <- 20 #number of datasets to impute
> ns <- nrow(wmiss) #count how many observations there are
> tall$.pv <- as.numeric(gsub("[^0-9]", "", tall$name)) #extract the numeric value
> nopv <- length(table(as.character(tall$.pv))) #the number of plausible values
```

The `tall$.pv` variable contains the extracted numeric value from the pv1math to pv10math values stored in the `name` variable. A limitation currently of Blimp is that all variables included must be in numeric format. As there are three factor variables (and `name` is a character variable), we need to first run some syntax (using dplyr) to recode all nonnumeric factors into numeric variables.

```
> tall_numeric <- mutate(tall,
  across(everything(), as.numeric)
)
```

In the process, the text values in the `name` variable are converted into NA (or missing). The output can be saved into a new dataset (here, a data.frame named `tall_numeric`). Plain numeric variables end up losing the descriptive factor labels (which are useful when reading the output); in a later step, we restore the original factor labels to the numeric variables.

Prior to imputation, we need to know which variables are going to be entered into the model as dummy-coded variables, which variables are the weight and clustering variables, and which variable(s) have complete data (if any). In addition, the analytic model must also be known as we need to impute the data using this substantive model. A weight variable will also be included as a predictor. The function that performs the imputation is the Blimp program which is called using the `rblimp` function:

```
> mymodel <- rblimp(
  data = tall_numeric,
  nominal = 'gender immig2 lackstaff',
  # ordinal = '',
  clusterid = 'cntschid',
  fixed = 'gender w_fstuwt',
  model = 'math ~ gender escs immig2
           stubeha lackstaff w_fstuwt',
  options = 'labels',
  seed = 1234,
  nimps = m
) |> by_group('.pv')
```

Several arguments are specified:

- data: the name of the dataset with missing data.
- nominal: the variables in the model which will be entered as dummy codes.
- ordinal: the variables in the model which have values which represent ordered categories (e.g., such as Likert scales that may have values ranging from 1 = strongly disagree to 4 = strongly agree). This is not used in the current example. Although the `lackstaff` variable may be considered ordinal, in the analytic model, this is entered as a dummy coded variable with the first variable label representing the

reference group so this is included in the nominal argument instead. Ordinal variables also take longer to impute compared to nominal variables.

- clusterid: represents the name of the clustering or grouping variable. For three-level models, the second-level id comes before the third-level id (i.e., `clusterid='level2id level3id'`). Currently, Blimp cannot model cross-classified data structures.
- fixed: indicates the predictor variables with no missing data.
- model: represents the substantive model used in the imputation process. Note that the model statement represents how the analytic model will be fit with the addition that the weight variable is included as a predictor (here we use the student-level weight). The model formula is entered except that a space is used to separate variables (instead of the + used in models in R). If interactions are to be tested in the analytic model, they should be included and simply entered as `variable1 * variable2`. If a random slope model is to be fit, a pipe operator can be added at the end of the model statement followed by the variables which are expected to randomly vary by group. For example, if `escs` is expected to randomly vary, add "`| escs`" at the end of the model statement. As the current model does not include that, a random intercept model was fit.
- The `option='labels'` argument is included which can help assess if there are variables which may require some consideration in the imputation process (explained later).
- seed: a specified seed number is included to be able to reproduce results. As every imputation uses some random noise to vary imputation results from imputation to imputation, a seed is required to obtain identical results in the future (or if the imputation is done by another researcher, the same imputed datasets will be created). The seed can be any numeric value with less than 10 digits and is required.
- nimp: specifies the number of datasets to impute. This can be hardcoded (i.e., `nimp=20`) but in this case, we created an *m* variable (i.e., `m<-20`) earlier which will be used in succeeding code as well.

After all the options are specified in the `rblimp` function, we add `|> by_group('.pv')` which indicates that the *m* imputations are performed for each pv separately. The `|>` operator is known as the pipe operator which was introduced in R 4.1.0. Without the `by_group` option specified, only *m* imputations are performed (regardless of the number of pvs). The output for the `rblimp` function is saved in the `mymodel` object. This process can take a while depending on how many imputations are requested. Using a Windows PC with an AMD Ryzen 9 processor with 16 gb of RAM required 35 min to complete. As indicated, for a trial run, users can set $m = 2$ first to test if everything works properly.

Post imputation, we can check the quality of imputations. As the imputation process uses what is referred to as Bayesian sampling, the Potential Scale Reduction (PSR) factor, also known as the r-hat ($\widehat{R}$) value (Gelman & Rubin, 1992), can be inspected. PSR values close to 1 are desirable and high PSR values may signal that more imputations or iterations are needed. For each parameter estimated, a maximum PSR of < 1.1

suggests that the sampler has "converged satisfactorily" (Carpenter et al., 2023, p. 229). To check the maximum PSR per imputation, we enter:

```
> mymodel |> sapply(\(x) tail(x@psr, 1) |> max(na.rm = TRUE))

     1      2      3      4      5      6      7      8      9     10
1.0462 1.0307 1.0361 1.0392 1.0603 1.0540 1.0407 1.0645 1.0301 1.0451
```

Results suggest that imputations are fine as the maximum PSR out of all the estimates per plausible value are < 1.1. If, however, there are PSRs > 1.1, figuring out the variables with high PSRs can be inspected by using:

```
> lapply(mymodel, psr)
```

The output is long and scrolling through the results (not shown) can indicate which variables are responsible for contributing to the highest PSRs. The `options='labels'` statement allows the researcher to view the variables related to the PSR values.

Fitting the analytic model of interest

The final steps in the process involve getting the imputed data in a format amenable for the analysis of interest, fitting the models, pooling the results, and summarizing the output. All the 200 imputed datasets are saved in the `mymodel` object and to extract them all in one large dataset, we use:

```
> impdat <- lapply(mymodel, as.mitml) |>
  unlist(recursive = FALSE) |>
  do.call(rbind, args = _)
```

The object `impdat` contains 1.7 million observations and consists of the datasets stacked on top of each other in a single data frame. This will not be used yet in the analysis but allows users to perform some more basic data management prior to the analysis (e.g., combining certain variables to form a scale, recoding certain variables). We also need to number the imputations in the data frame using:

```
> impdat$.implist <- rep(1:(m * nopv), each = ns)
```

Using `table(impdat$.implist)` will show that there are 8,475 observations in each of the 200 iterations (output not shown). At this point, we can also convert the numeric variables back to factors with the factor labels. As it currently is, we only see a list of numeric values for our original factors—for example, using the `lackstaff` variable, we see only categories of 1 through 4:

```
> table(impdat$lackstaff)

     1      2      3      4
250958 720793 576658 146591
```

To copy the factor attributes from the original `comb` dataset, we can use:

```
> attributes(impdat$lackstaff) <- attributes(comb$lackstaff)
> table(impdat$lackstaff)

    Not at all    Very little To some extent          A lot
        251676         720205         578231         144888
```

We convert the other two numeric variables to factors as they were originally:

```
> attributes(impdat$gender) <- attributes(comb$gender)
> attributes(impdat$immig2) <- attributes(comb$immig2)
```

Converting the variables back to factors is necessary (in order for R to enter them as dummy codes in the model) and helps when viewing regression results. As a final step before performing the analysis, we need to convert the data frame back into a list which R can use. A list in R can be a container of different types of data and can consist of multiple datasets which can be analyzed one at a time. To create a list called `alldat`, we split the `impdat` data by the imputation number which is comprised of both the pv and the imputation number.

```
> alldat <- split(impdat, impdat$.implist)
```

Once the data are in a list format, we can apply a function to all the elements of a list and save the results of each model fit. We use the `lapply` function ("list apply") in R which requires the list to use (`alldat`) and the function (`FUN =`) to perform. We name the function `modfit` (model fit). Output is saved in the `allres` object. When researchers are using their own data, the `modfit` function should be modified as necessary.[9]

```
> modfit <- function(x) {
  mix(math ~ gender + escs + immig2 + stubeha +
      lackstaff + (1|cntschid),
  weights = c('w_fstuwt', 'w_schgrnrabwt'),
  data = x)
}
> allres <- lapply(alldat, FUN = modfit)
```

The `function(x)` signifies that the following lines are part of a function and the `x` acts as a placeholder for the 200 elements (in this case the 200 datasets) in the `impdat` list. Although the model will be fit as specified, this procedure is not generally recommended as models are fit to each element of the list in a serial fashion, one at a time which can take a large amount of time. Using a Windows PC with an AMD Ryzen 9 processor with 16 gb of RAM required 48 min to complete. However, with the availability of multiple processing cores in modern computers, the models can be fit in parallel where several models are fit at the same time. Using parallel computing, only 4.2 min

[9] For more information on using the mix function, see https://cran.r-project.org/web/packages/WeMix/vignettes/Introduction_to_Mixed_Effects_Models_With_WeMix.pdf.

were required using the following syntax (the parallel package is already installed with R by default).

```
> library(parallel)
> cl <- makeCluster(detectCores() - 1)
> registerDoParallel(cl)
> clusterExport(cl, list('modfit', 'mix'))
> allres <- parLapply(cl, alld2, fun = modfit)
> stopCluster(cl)
```

After the 200 models are fit, results need to be combined using Rubin's (1987) rules. The `mixPV` summary function can pool the results from the 200 models. The saved output though must first be classified as a `mixPV` object for the `summary` function to work.

```
> class(allres) <- 'mixPV'
> summary(allres)
```

Comparing fixed effect model results for models fit using the dataset with 90% complete, 82% complete, and the imputed datasets (or models A, B, and C, respectively) show that the imputed (A) and 90% complete results (C) are closest to each other (see Table 2). However, with the 82% complete dataset (B), larger differences were seen especially with the level-2 coefficients (even though the only additional missing data was added to a level-1 variable). In addition, the intercept also differed to a larger extent when 82% of the data were complete. In terms of the random effects, ICCs were lower for the dataset with 82% complete data (see Table 3).

Conclusions

Although missing data are commonly found in LSAs, a common approach to handling missing data is through the use of listwise deletion which has been shown, over several decades of research, to be a suboptimal strategy. Using freely available software, we provide a step-by-step tutorial for researchers using R and the Blimp program that can be used to impute missing data in a multilevel setting. Particular attention should be paid to clustered, weighted data that use plausible values, such as the data structures found in LSAs (a slightly simpler approach when pvs are not used is shown in the Appendix). Interested readers can use the provided syntax and modify the code as necessary to suit their analytic models and there should be fewer reasons to resort to listwise deletion.

Table 3 Comparison of Random Effects Estimates with Differing Levels of Missing Data and Using Multiply Imputed Datasets

	90% complete	82% complete	Imputed
School intercept -τ_{00}	2675.176***	2099.088***	3282.989***
	(368.453)	(368.062)	(424.464)
Residual -σ^2	4397.184***	3895.583***	4150.606***
	(180.682)	(168.531)	(220.188)
ICC	.38	.35	.44

***$p < .001$. ICC = intraclass correlation coefficient. Robust standard errors in parenthesis

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40536-025-00248-9.

Supplementary material 1.

Acknowledgements
Blimp was developed with funding from the Institute of Education Sciences awards R305D150056 and R305D190002.

Author contributions
F.H. and B.K. wrote the main manuscript text. F.H. developed the workflow and tutorial. B.K. developed the Blimp software. All authors reviewed the manuscript.

Data availability
The data for the tutorial can be accessed at: —https://github.com/flh3/pubdata/raw/refs/heads/main/miscdata/belgium.rds—https://github.com/flh3/pubdata/raw/refs/heads/main/miscdata/belgiumwmiss.rds. The syntax used in this manuscript can be downloaded from https://francish.net.

Declarations

Competing interests
The authors declare no competing interests.

Received: 8 November 2024 Accepted: 27 March 2025
Published online: 23 April 2025

References

Acock, A. C. (2005). Working with missing values. *Journal of Marriage and Family, 67*(4), 1012–1028.
Bailey, P., Kelley, C., Nguyen, T., & Huo, H. (2023). *WeMix: Weighted mixed-effects models using multilevel pseudo maximum likelihood estimation*. https://CRAN.R-project.org/package=WeMix
Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, 67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01
Beyer, C., & Brese, F. (2024). Secondary school students' attitudes of tolerance towards minorities. *Large-Scale Assessments in Education*. https://doi.org/10.1186/s40536-024-00217-8
Bouhlila, D. S., & Sellaouti, F. (2013). Multiple imputation using chained equations for missing data in TIMSS: A case study. *Large-Scale Assessments in Education*. https://doi.org/10.1186/2196-0739-1-4
Carpenter, J. R., Kenward, M. G., Bartlett, J. W., Morris, T. P., Quartagno, M., & Wood, A. M. (2023). *Multiple Imputation and its Application 2e* (1st ed.). Wiley. https://doi.org/10.1002/9781119756118
Chan, C., Chan, G. C., Leeper, T. J., & Becker, J. (2018). *rio: A Swiss-army knife for data file I/O*.
De Silva, A. P., De Livera, A. M., Lee, K. J., Moreno-Betancur, M., & Simpson, J. A. (2021). Multiple imputation methods for handling missing values in longitudinal studies with sampling weights: Comparison of methods implemented in Stata. *Biometrical Journal, 63*(2), 354–371. https://doi.org/10.1002/bimj.201900360
Enders, C. K. (2010). *Applied missing data analysis*. Guilford Publications.
Enders, C. K. (2017). Multiple imputation as a flexible tool for missing data handling in clinical research. *Behaviour Research and Therapy, 98*, 4–18. https://doi.org/10.1016/j.brat.2016.11.008
Enders, C. K., Du, H., & Keller, B. T. (2020). A model-based imputation procedure for multilevel regression models with random coefficients, interaction effects, and nonlinear terms. *Psychological Methods, 25*(1), 88–112. https://doi.org/10.1037/met0000228
Enders, C. K., Mistler, S. A., & Keller, B. T. (2016). Multilevel multiple imputation: A review and evaluation of joint modeling and chained equations imputation. *Psychological Methods, 21*(2), 222.
Fishbein, B., Foy, P., & Yin, L. (2021). *TIMSS 2019 user guide for the international database* (2nd edition). TIMSS & PIRLS International Study Center. https://timss2019.org/international-database/downloads/TIMSS-2019-User-Guide-for-the-International-Database-2nd-Ed.pdf
Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science, 7*(4), 457–472. https://doi.org/10.1214/ss/1177011136
Grund, S., Lüdtke, O., & Robitzsch, A. (2018). Multiple imputation of missing data for multilevel models: Simulations and recommendations. *Organizational Research Methods, 21*(1), 111–149. https://doi.org/10.1177/1094428117703686
Grund, S., Lüdtke, O., & Robitzsch, A. (2021). On the treatment of missing data in background questionnaires in educational large-scale assessments: An evaluation of different procedures. *Journal of Educational and Behavioral Statistics, 46*(4), 430–465. https://doi.org/10.3102/1076998620959058
Grund, S., Lüdtke, O., & Robitzsch, A. (2024). Missing data in the analysis of multilevel and dependent data. In M. Stemmler, W. Wiedermann, & F. L. Huang (Eds.), *Dependent data in social sciences research: Forms, issues, and methods of analysis* (pp. 289–324). Springer International Publishing. https://doi.org/10.1007/978-3-031-56318-8_12
Huang, F. (2025). *MLMusingR: Practical Multilevel Modeling* (p. 0.4.0) . https://doi.org/10.32614/CRAN.package.MLMusingR
Huang, F. L. (2024). Using plausible values when fitting multilevel models with large-scale assessment data using R. *Large-Scale Assessments in Education, 12*(1), 7. https://doi.org/10.1186/s40536-024-00192-0

Jewsbury, P. A., Jia, Y., & Gonzalez, E. J. (2024). Considerations for the use of plausible values in large-scale assessments. *Large-Scale Assessments in Education, 12*(1), 24. https://doi.org/10.1186/s40536-024-00213-y
Keller, B. (2024). *rblimp: Integration of Blimp Software into R*. https://github.com/blimp-stats/rblimp
Keller, B., & Enders, C. (2023). *Blimp user's guide (Version 3)* (Version 3) [Computer software]. www.appliedmissingdata.com/blimp
King, G., Honaker, J., Joseph, A., & Scheve, K. (1998). *Listwise deletion is evil: What to do about missing data in political science*. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=813e3040f8a0988bfd3b385f49cbbabb802afa0c
Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). Wiley.
Meinck, S. (2015). Computing sampling weights in large-scale assessments in education. *Survey Methods: Insights from the Field*, 1–13.
Meng, X.-L. (1994). Multiple-imputation inferences with uncongenial sources of input. *Statistical Science*. https://doi.org/10.1214/ss/1177010269
National Center for Education Statistics. (n.d.). *NAEP Nations Report Card -*. National Center for Education Statistics. Retrieved March 5, 2025, from https://nces.ed.gov/nationsreportcard/glossary.aspx
Quartagno, M., Carpenter, J. R., & Goldstein, H. (2020). Multiple imputation with survey weights: A multilevel approach. *Journal of Survey Statistics and Methodology, 8*(5), 965–989.
R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/
Rabe-Hesketh, S., & Skrondal, A. (2006). Multilevel modelling of complex survey data. *Journal of the Royal Statistical Society Series a: Statistics in Society, 169*(4), 805–827.
Raudenbush, S., & Bryk, A. (2002). *Hierarchical linear models: Applications and data analysis methods* (2nd ed.). Sage.
Rodríguez De Luque, J. J. (2024). Examining the associations between high achievement in reading and school climate: Evidence from five South American countries. *Large-Scale Assessments in Education, 12*(1), 32. https://doi.org/10.1186/s40536-024-00220-z
Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys* (Vol. 81). Wiley.
Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys* (Vol. 81). Wiley.
Schafer, J. L. (1999). Multiple imputation: A primer. *Statistical Methods in Medical Research, 8*(1), 3–15. https://doi.org/10.1177/096228029900800102
Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods, 7*(2), 147–177. https://doi.org/10.1037/1082-989X.7.2.147
Schafer, J. L., & Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate Behavioral Research, 33*(4), 545–571. https://doi.org/10.1207/s15327906mbr3304_5
Singer, J. D. (1998). Using SAS PROC MIXED to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational and Behavioral Statistics, 23*(4), 323–355. https://doi.org/10.2307/1165280
van Buuren, S., & Groothuis-Oudshoom, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software, 45*, 1–67.
von Hippel, P. T. (2018). How many imputations do you need? A two-stage calculation using a quadratic rule. *Sociological Methods and Research*. https://doi.org/10.1177/0049124117747303
Weirich, S., Haag, N., Hecht, M., Böhme, K., Siegle, T., & Lüdtke, O. (2014). Nested multiple imputation in large-scale assessments. *Large-Scale Assessments in Education, 2*(1), 1–18. https://doi.org/10.1186/s40536-014-0009-0
White, I. R., Royston, P., & Wood, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine, 30*(4), 377–399. https://doi.org/10.1002/sim.4067
Wickham, H. (2021). *tidyr: Tidy messy data*. https://CRAN.R-project.org/package=tidyr
Wickham, H., François, R., Henry, L., & Müller, K. (2020). *dplyr: A grammar of data manipulation*. https://CRAN.R-project.org/package=dplyr
Widaman, K. F. (2006). Best practices in quantitative methods for developmentalists: III. Missing data: What to do with or without them. *Monographs of the Society for Research in Child Development, 71*(3), 42–64. https://doi.org/10.1111/j.1540-5834.2006.00404.x
Wilkinson, L. (1999). Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist, 54*(8), 594–604. https://doi.org/10.1037/0003-066X.54.8.594
Ye, W., Scherer, R., & Blömeke, S. (2024). Teachers' and principals' perceptions of school emphasis on academic success: Measurement invariance, agreement, and relations to student achievement. *Large-Scale Assessments in Education*. https://doi.org/10.1186/s40536-024-00207-w

Publisher's Note