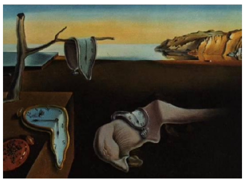


Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Imputation of Incomplete Multilevel Data with R

Hanne I. Oberman
Utrecht University

Johanna Muñoz
University Medical Center Utrecht

Valentijn M.T. de Jong
University Medical Center Utrecht

Gerko Vink
University Medical Center Utrecht

Thomas P.A. Debray
University Medical Center Utrecht

Abstract

This tutorial illustrates the imputation of incomplete multilevel data with the R packackage **mice**. Our scope is only simple multilevel models, to show how imputation can yield less biased estimates from incomplete clustered data. More complex models can be accomodated, but are outside the scope of this paper. Incomplete multilevel data requires careful consideration of the missing data problem and analysis strategy. In this tutorial, we focus on a popular strategy for accommodating missingness in multilevel data: replacing the missing data with one or more plausible values, i.e., imputation.Imputation separates the missing data problem from the main analysis and the completed data can be analyzed as if it has been fully observed. This tutorial illustrates the imputation of incomplete multilevel data with the statistical programming language R. We aim to show how imputation can yield less biased estimates from incomplete clustered data. We provide practical guidelines and code snippets for different missing data situations, including non-ignorable missingness mechanisms. For brevity, we focus on multilevel imputation using chained equations with the R mice package and its adjacent packages.

Keywords: missing data, multilevel, clustering, **mice**, R.

1. Introduction: Clustering and incomplete data

1. missing data occur often in data with human subjects

2. missing data may be resolved, but need to be handled in accordance with the analysis of scientific interest
3. in human-subjects research, there is often clustering, which may be captured with multilevel modeling techniques
4. if the analysis of scientific interest is a multilevel model, the missing data handling method should accommodate the multilevel structure of the data
5. both missingness and multilevel structures require advanced statistical techniques
6. this tutorial sets out to facilitate empirical researchers in accommodating both multilevel structures as well as missing data.
7. we illustrate the use of the software by means of three case studies from the social and biomedical sciences.

In hierarchical datasets, clustering is a concern because the homoscedasticity in the error terms cannot be assumed across clusters and the relationship among variables may vary at different hierarchical levels. When multiple imputation is used to deal with missing data, as the imputation and analysis process is performed separately, it is necessary that imputation model being congenial with the main analysis model (Meng, 1994), e.g. if the main model accounts for the hierarchical structure also imputation model should do it (Audigier, 2021). Not including clustering into the imputation process may lead to effect estimates with smaller standard errors and inflated type I error.

1.1. overview of software

The popular **mice** package in R **?** TODO: **?**(Is this section relevant?)

1.2. Scope

This papers serves as a tutorial for imputing incomplete multilevel data with **mice** in R. **mice** has become the de-facto standard for imputation by chained equations, which iteratively solves the missingness on a variable-by-variable basis. **mice** is known to yield valid inferences under many different missing data circumstances (Van Buuren 2018).

We provide practical guidelines and code snippets for different missing data situations, including non-ignorable mechanisms. For reasons of brevity, we focus on multilevel imputation by chained equations with **mice** exclusively; other imputation methods and packages (see e.g. **?**, and Grund, Lüdtke, and Robitzsch (2018)) are outside the scope of this tutorial. Assumed knowledge includes basic familiarity with the **lme4** notation for multilevel models (see Table **??**).

We illustrate imputation of incomplete multilevel data using three case studies:

- `popmis` from the **mice** package (simulated data on perceived popularity, $n = 2,000$ pupils across $N = 100$ schools with data that are MAR, van Buuren and Groothuis-Oudshoorn 2021);
- `impact` from the **metamisc** package (empirical data on traumatic brain injuries, $n = 11,022$ patients across $N = 15$ studies with data that are MAR, Debray and de Jong 2021);

- `obesity` from the **micemd** package [simulated data on obesity, $n = 2,111$ patients across $N = 5$ regions with data that are MNAR].

For each of these datasets, we discuss the nature of the missingness, choose one or more imputation models and evaluate the imputed data, but we will also highlight one specific aspect of the imputation workflow.

This tutorial is dedicated to readers who are unfamiliar with multiple imputation. More experienced readers can skip the background section and introduction (case study 1) and directly head to practical applications of multilevel imputation under MAR conditions (case study 2, Section 4.3) or under MNAR conditions (case study 3).

2. Background

2.1. Concepts in multilevel data

Many datasets include individuals that are clustered together, for example in geographic regions, or even different studies. In the simplest case, individuals (e.g., students) are nested within a single cluster (e.g., school classes). More complex clustered structures may occur when there are multiple hierarchical levels (e.g., students in different schools or patients within hospitals within regions across countries), or when the clustering is non-nested (e.g., electronic health record data from diverse settings and populations within large databases). With clustered data we generally assume that individuals from the same cluster tend to be more similar than individuals from other clusters. In statistical terms, this implies that observations from the same cluster are not independent and may in fact be correlated. If this correlation is left unaddressed, estimates of p values, confidence intervals even model parameters are prone to bias (Localio, Berlin, Ten Have, and Kimmel 2001). Statistical methods for clustered data typically adopt hierarchical models that explicitly describe the grouping of observations. These models are also known as 'multilevel models', 'hierarchical models', 'mixed effect models', 'random effect models', and in the context of time-to-event data as 'frailty models'. Table **??** provides an overview of some key concepts in multilevel modeling.

In R, multilevel models may be fitted using package such as **lme4** or **nlme**. For linear mixed-effects models, the function

```
lmer(formula, data, ...)
```

2.2. Concepts in missing data

missing data mechanisms etc.

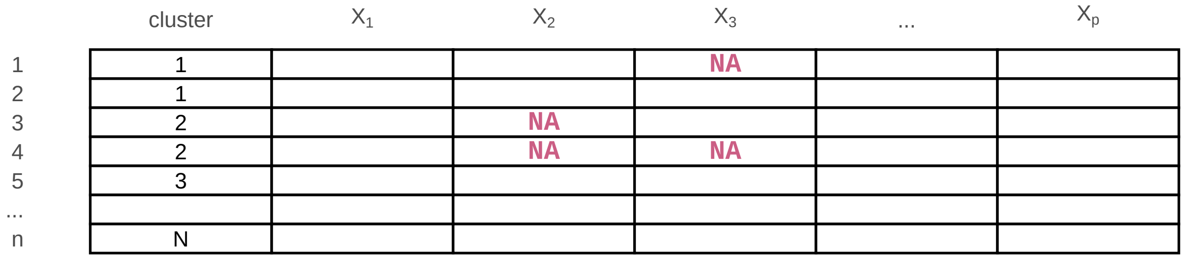
As with any other dataset, clustered datasets may be impacted by missingness in much the same way. Several strategies can be used to handle missing data, including complete case analysis and imputation. We focus on the latter approach and discuss statistical methods for replacing the missing data with one or more plausible values. Imputation separates the

Table 1: Concepts in multilevel methods

Concept	Details
Sample units	Units of the population from which measurements are taken in a sample, e.g., students.
Cluster	Variable that specify the cluster or agruppation, e.g., Classroom
Hierarchical data	Data are grouped into clusters at different levels, observations belonging to the same cluster are expected to share certain characteristics.
Level-1	Variable that varies within a cluster, eg. Test score
Level-2	Variable that does not vary within a cluster but between, e.g. teacher experience.
Hierarchical model	Model accounting for dependant observations relying on certain parameters (within cluster) which in turn depend on other parameters (between cluster)
Fixed effect	Effects that are constant across all sample units, e.g. something that researchers control for and can repeat, such as a teaching strategy (tutoring after class)
Random effect	Effects that are a source of random variation in the data, and whose levels are not fully sampled. e.g. test score tendency during academic year between students due to no controlled factors such as genetic,family history
Mixed effect	Includes fixed and random effects, e.g. the fixed effect would be the treatment effect of a drug and the random effect would be the ID of the hospital where the patient is treated. Multilevel models typically accommodate for variability by including a separate group mean for each cluster e.g random intercept on hospitals. In addition to random intercepts, multilevel models can also include random coefficients and heterogeneous residual error variances across clusters (see e.g. @gelm06, @hox17 and @jong21).
ICC	The variability due to clustering is often measured by means of the intraclass coefficient (ICC). The ICC can be seen as the percentage of variance that can be attributed to the cluster-level, where a high ICC would indicate that a lot of variability is due to the cluster structure.
Stratified intercept	

missing data problem from the analysis and the completed data can be analyzed as if it were completely observed. It is generally recommended to impute the missing values more than once to preserve uncertainty due to missingness and to allow for valid inferences (c.f. Rubin 1976).

With incomplete clustered datasets we can distinguish between two types of missing data: sporadic missingness and systematic missingness (**?**). Sporadic missingness arises when variables are missing for some but not all of the units in a cluster (Van Buuren 2018; Jolani 2018). For example, it is possible that test results are missing for several students in one or more classes. When all observations are missing within one or more clusters, data are said to be systematically missing. Sporadic missingness is visualized in Figure XYZ.



	cluster	X_1	X_2	X_3	...	X_p
1	1			NA		
2	1					
3	2		NA			
4	2		NA	NA		
5	3					
...						
n	N					

Column X_1 in Figure 1 is completely observed, column X_2 is systematically missing in cluster 2, and column X_3 is sporadically missing. To analyze these incomplete data, we have to take the nature of the missingness and the cluster structure into account. For example, the sporadic missingness in X_3 could be easily amended if this would be a cluster-level variable (and thus constant within clusters). We could then just extrapolate the true (but missing) value of X_3 for unit 1 from unit 2, and the value for unit 4 from unit 3. If X_3 would instead be a unit-level variable (which may vary within clusters), we could not just recover the unobserved 'truth', but would need to use some kind of missing data method, or discard the incomplete units altogether (i.e., complete case analysis). Complete case analysis can however introduce bias in statistical inferences and lowers statistical power. Further, with the systematic missingness in X_2, it would be impossible to fit a multilevel model without accommodating the missingness in some way. Complete case analysis in that case would mean excluding the entire cluster from the analyses. The wrong choice of missing data handling method can thus be extremely harmful to the inferences.

Imputation of missing data requires consideration of the mechanism behind the missingness. Rubin proposed to distinguish between data that are missing completely at random (MCAR), data that are missing at random (MAR) and data that are missing not at random (MNAR; see Table **??**). For each of these three missingness generating mechanisms, different imputation strategies are warranted (Yucel (2008) and Hox, van Buuren, and Jolani (2015)). We here consider the general case that data are MAR, and expand on certain MNAR situations.

Table 2: Concepts in missing data methods

Concept	**Details**
MCAR	Missing Completely At Random, where the probability to be missing is equal across all data entries

Concept	**Details**
MAR	Missing At Random, where the probability to be missing depends on observed information
MNAR	Missing Not At Random (MNAR), where the probability to be missing depends on unrecorded information, making the missingness non-ignorable (Rubin 1976; **?**).

2.3. Imputation with mice

The R package **mice** provides a framework for imputing incomplete data on a variable-by-variable basis. The `mice()` function allows users to flexibly specify how many times and under what model the missing data should be imputed. This is reflected in the first four function arguments

```
mice(data, m, method, predictorMatrix, ...)
```

where `data` refers to the incomplete dataset, `m` determines the number of imputations, `method` denotes the functional form of the imputation model and `predictorMatrix` specifies the interrelational dependencies between variables and imputation models (i.e., the set of predictors to be used for imputing each incomplete variable).

Box 2. The `methods`.

TODO: @ Hanne i added a wide explanation about hte prediction matrix in the Model Specification subsection, so you can check it and consider to modify your box.. because i dont think is usefull to repeat information across the paper…
Box 3. The predictor matrix. The entries corresponding to the level-1 predictors are coded with a 3, indicating that both the original values as well as the cluster means of the predictor are included into the imputation model. The entry of 4 in the predictor matrix adds three variables to the imputation model for the imputation model predictor: the value of the predictor, the cluster means of the predictor and the random slopes of the predictor. - -2 = cluster variable - 1 = overall effect - 3 = overall + group-level effect
- 4 = individual-level (random) and group-level (fixed) effect

3. Multilevel imputation workflow

There are different strategies that can be adopted in the imputation process that account for clustering: inclusion of cluster indicator variable, performing a separate imputation process for each cluster, or performing a simultaneous imputation process by using an imputation method that accounts for clustering.zot

The selection of each strategy depends mainly on the assumptions in the main analysis and also on the restriction of the analyzed data.

Regarding the restrictions imposed by the nature of the data, i.e., the sample size, the use of cluster indicator variables is restricted in datasets where there are not many clusters and many observations per cluster (Graham, 2009). The last restriction is also required when imputations are performed on each cluster separately. When this restriction cannot be achieved, one can use an imputation model that simultaneously imputes all clusters using a hierarchical model (Allison 2002).

Under this hierarchical imputation model, observations within clusters are correlated and this correlation is modelled by a random effect so the hierarchical model can be estimated even when there are few observations per cluster. However, this strategy is best suited for balanced data (Grund, 2017) and when the number of clusters is adequate sufficient for the random effects model (Austin,2018).

Here it is important to evaluate the assumptions imposed by the main model, for instance the cluster indicator strategy bias the estimator when the model is based on a hierarchical model (Taaljard,2008). Although an imputation strategy congenial with the main model is preferred, it is important to consider whether a complex imputation model is appropriate for the data, as a less complex imputation strategies may also lead to unbiased estimates in certain scenarios (Bailey 2020). Below we provide a imputation workflow that can be used in general to impute cluster data.

3.1. Exploration of Available Data

First, explore the data, preferably using visual tools. This will help you choose an imputation model that suits your data. In addition, the intraclass correlation coefficient (ICC) can be examined to assess cluster differences, aiding in the choice between methods for dependent (such as clustering) or independent data.

Next, explore the missingness. Look at the **proportion of missing values** in the dataset variables. Variables for which most data is missing may contribute little in the analysis, and may be removed from the analysis model. Removing them also from the imputation model can lower the risk of multicollinearity or computational issues, especially with certain parametric imputation methods. You can also identify predictors for the imputation model by using inflow criteria to see connections between missing data in one variable and observed variables, and outflow criteria to identify connections between observed values in one variable and missing data in others.

Check the missing patterns at cluster level, this can help you to select the most appropriate imputation approach in terms of computational efficiency (e.g., simpler regression imputation versus FCS in univariate patterns).

3.2. Assess Estimation Procedure Robustness to Missing Data

Before diving into imputation, make sure your main model can handle missing data. Sometimes, simpler methods like complete case analysis might be suffice, especially if your missingness is low (usually <5%). There are scenarios in which specific Maximum Likelihood (ML) estimation methods outperform Multiple Imputation (MI) methods, for instance when the response variable is the sole incomplete variable, mixed models demonstrate robustness to missing data under the Missing at Random (MAR) assumption and with a correct variance-covariance specification.Molenberghs and Kenward (2007).

3.3. Pre-imputation

Clean up your dataset before the imputation process. For imputation, include the variables necessary for your analysis, depending on your analysis this may be the exposure or intervention, outcome, covariates, confounders, stratification factors, and the cluster variable. Think about adding other useful variables, even if they're not in the main model, such as instrumental or auxiliary variables which might improve the accuracy of the imputation model, especially if they're associated to the probability of missingness for some incomplete variables.

Figure out if you can directly impute incomplete variables by using deductive imputation. This involves inferring missing values based on logical connections between variables. It's especially useful for variables that depend on each other, like calculating BMI from weight and height.

Deductive imputation is also useful for getting values for level-1 variables from level-2 ones, like in Individual Participant Data (IPD). For example, deductive imputation can be employed to infer missing test values for patients associated with a center unit that lacks a test procedure or machine, utilizing accessible metadata or protocol details.

3.4. Setting Imputation Model

Clustering Inclusion

When it comes to handling clusters during the imputation process, you've got a few options. You can use a cluster indicator variable, run separate imputation for each cluster, or go for a simultaneous imputation method that takes clustering into account Eddings and Marchenko.

Which strategy you choose depends on the assumptions in your main analysis and the limitations of your data. If your analysis do not use a hierarchical model (like a descriptive approach) and you have a small number of clusters with lots of observations in each, using a cluster indicator or separate imputation might be the way to go Graham (2009). Conversely, if you have more clusters or fewer observations per cluster, you might want to try a simultaneous hierarchical imputation model **?**.

In a hierarchical imputation model, random intercepts and random effects model the correlations between observations within clusters, making it possible to estimate even with a small number of observations per cluster. There are various proposed multiple imputation models based on hierarchical models, each with its own set of assumptions Audigier, White, Jolani, Debray, Quartagno, and Carpenter.

If your analysis uses a hierarchical model, make sure the assumptions of your imputation model match up. For example, using a cluster indicator approach may lead to bias estimates if your model is based on a hierarchical structure Taljaard, Donner, and Klar (2008); Speidel, Drechsler, and Sakshaug (2018). Even if you prefer an imputation strategy that aligns with your main model, check if it suits your data; sometimes simpler strategies can give unbiased estimates in certain scenarios Bailey, Andridge, and Shoben (2020).

Choice of Individual Imputation Methods

Start by choosing the imputation model for each incomplete variable in your dataset. The mice package suggests methods based on variable types for non-clustered variables, and for the clustered ones you can use the micemd package's find.defaultMethod() function. This function selects from different 2l imputation methods based on cluster size and the proportion of missing data in each cluster.

Besides the package-defined imputation methods, you can specify custom methods using the "I formula". This lets you calculate deterministic variables during the imputation or tweak imputation methods based on specific conditions, like conditioning the imputation model to the level of an incomplete covariate (e.g., a pregnancy test for females).

Model Specification

The imputation model must be congenial with the main model Meng (1994). Congeniality issues arise when the imputation model and the main model make different assumptions, often due to the omission of a polynomial or interaction term or the use of transformed variables.

The imputation model can incorporate additional terms compared to the main model without causing compatibility issues. For instance, it's recommended to include the outcome variable in the imputation model for prediction variables **?**. In cases where the outcome is time-to-event, the Nelson-Aalen estimate of the time to the event should be added as a covariate in the imputation model, as well as the event indicator [REF: White IR, Royston P. Imputing missing covariate values for the Cox model. Stat Med. 2009 Jul 10;28(15):1982-98. Available from: doi.org/10.1002/sim.3618]. Additionally, including auxiliary variables, even if not part of the main model, can be linked to the probability of missingness, improving the likelihood of meeting the Missing at Random (MAR) assumption and enhancing estimation efficiency **?**.

Imputation models are specified on a variable basis, either using a prediction matrix in the pred parameter or through a list of formulas in the formula parameter. In the prediction matrix option, the type of each predictor variable is specified for each incomplete variable (see table).

$$
\begin{aligned}
y_{ij} =& (\beta_0 + b_{0_i}) + \beta_1 x_{1_{ij}} + \beta_2 x_{2_{i.}} \\
&+ (\beta_3 + b_{3_i}) x_{3_{ij}} \\
&+ \beta_4 x_{4_{ij}} + \beta_{m4} \overline{x_{4_{i.}}} \\
&+ (\beta_5 + b_{5_i}) x_{5_{ij}} + \beta_{m5} \overline{x_{5_{i.}}}
\end{aligned}
\tag{1}
$$

Type	Definition
1	Fixed variable, e.g., level-1 $x_{1_{ij}}$ or level-2 $x_{2_{i.}}$
2	Random variable, e.g., $x_{3_{ij}}$
3	*Fixed variable with cluster mean $x_{4_{ij}}$
4	*Random variable with cluster mean $x_{5_{ij}}$
-2	Cluster variable, in this case the one defined by j index
-3.	Random variable only included on selection model (Heckman model)
-4.	Random variable only included on main model (Heckman model)

- It has been advised to use variables of type 3 and 4, as the inclusion of the means of the cluster is beneficial on FCS Mistler and Enders (2017).

Recipes have been proposed for imputing incomplete level-1 and level-2 variables for hierarchical models **?** (§ 7.10), which can be convenient to follow when dealing with numerous variables (interaction terms at different levels) and models with many random effects that are prone to convergency problems due to overspecification in the imputation model. These rules were designed to ensure compatibility among the conditionally specified imputation models and congeniality between the imputation and the main model. However, they may not be applicable to all 2l imputation methods. For instance, the 2l.2stage imputation methods of the micemd package only allow the inclusion of random predictor variables (2).

On the other hand, the formula option is useful in specifying complex imputation models with polynomial terms or interactions and compared with the prediction matrix method do not requires the inclusion of additional terms as Just Another Variable (JAV) Van Buuren (2018)(§ 6.4).

For some interaction terms, including treatment interaction effects, it has been suggested to conduct separate imputation by treatment group Zhang, Dashti, Carlin, Lee, and Moreno-Betancur (2023). Additionally, one might apply random forest or deep learning methods, as these can handle interaction and non-linear terms without requiring the explicit specification of an imputation model.

3.5. Post-Imputation

During the imputation process, certain issues may arise that halt the process. In hierarchical model imputations, many issues are related to overfitting the imputation model. To troubleshoot, it is recommended to inspect the imputation log file for variables causing problems. One approach to address this is to reduce the number of predictors, by applying the previous referred recepies step by step or by using functions like quickpred. Another option is to consider variable transformations, such as scaling when the model is invariant to linear transformations e.g., random intercept models. Also, adjusting the level of the hierarchical model (e.g., using a homogeneous variance imputation method or a 1l model) can also be beneficial.

It is crucial to check the range of imputed variables, as excessively large imputed values for one predictor may trigger convergence issues in other variables. To tackle this, including post-processing specifications on problematic variables or utilizing imputation models like Predicted Mean Matching (PMM) can ensure that imputed values align with observable values.

In some situations, adopting a separate imputation strategy might be worth considering. For example, in analyses involving multiple endpoints, conducting distinct imputation processes for each endpoint could be more effective than a unified imputation approach.

3.6. Convergence and Sensitivity Analysis

Before starting the analysis of each imputed dataset, it is crucial to validate the convergence of the imputation process. This is commonly accomplished through trace plots that depict the mean and variance of the incomplete variables across iterations. These plots serve to uncover potential circular issues or the need for additional iterations. Additionally, it is also important to verify that imputed values fall within a plausible range and also to check the distribution of imputed variables, ensuring that the imputed variable distribution aligns with the distribution of observed values (under the MAR assumption). An alternative approach involves assessing the prediction accuracy of the imputation method Cai, Van Buuren, and Vink (2023).

While the majority of Multiple Imputation by Chained Equations (MICE) methods are based on Missing at Random (MAR) assumptions, field expert input may suggest that the Missing Not at Random (MNAR) mechanism could be plausible for certain variables. An MNAR variable is one in which the probability of missingness depends on an unobservable variable. This can occur when missingness is associated with the incomplete value itself (self-marking) or when there is an unobserved variable linked to both the value and the probability of missingness of the incomplete value (indirectly non-informative). Specifically, for the indirectly non-informative case in hierarchical datasets, imputation methods based on the Heckman method can be considered.Hammon and Zinn (2020); Hammon (2022); Muñoz, Efthimiou, Audigier, De Jong, and Debray (2023a)

4. Illustrations

In this section, we demonstrate the workflow using three case studies.

4.1. Setup

```
R> set.seed(123)          # for reproducibility
R> library(lme4)          # for multilevel modeling
R> library(nlme)          # for multilevel modeling
R> library(mice)          # for imputation
R> library(miceadds)      # for multilevel imputation methods
R> library(micemd)        # for selection-model imputation methods
```

```
R> library(mitml)        # for multilevel parameter pooling
R> library(broom)        # for clean model estimates
R> library(broom.mixed)  # for multilevel model estimates
R> library(dplyr)        # for data wrangling
R> library(ggmice)       # for visualization
R> library(ggplot2)      # for visualization
```

4.2. Popularity data

In this section we will go over the different steps involved with imputing incomplete multilevel data with the R package mice. We consider the simulated `popmis` dataset, which included pupils ($n = 2000$) clustered within schools ($N = 100$). The following variables are of primary interest:

- `school`, school identification number (clustering variable);
- `popular`, pupil popularity (self-rating between 0 and 10; unit-level);
- `sex`, pupil sex (0 = boy, 1 = girl; unit-level);
- `texp`, teacher experience (in years; cluster-level).

The analysis model corresponding to this dataset is multilevel regression with random intercepts for the different schools. We will estimate the association between the pupils' sex and their popularity score. This model can be expressed in `lme4` code as:

```
popular ~ 1 + sex + (1 | school)
```

Given the j-th student belonging to the i-th school, the main model can be formulated as:

$$popular_{ij} = (\beta_0 + b_i) + \beta_1 sex_{ij} + \beta_2 texp_{ij} + \epsilon_{ij}$$

where ϵ_{ij} corresponds to the error term.

We load the data into the environment with

```
R> data("popmis", package = "mice")
```

and select the relevant variables

```
R> dat <- popmis[, c("school", "popular", "texp", "sex")]
```

which results in the following data structure.

```
R> head(dat)
```

```
  school popular texp sex
1      1      NA   24   1
2      1      NA   24   0
3      1       7   24   1
4      1      NA   24   1
5      1      NA   24   1
6      1       7   24   0
```

The association of interest can be visualized with `ggmice`,

```
ggmice(dat, aes(sex, popular)) +
  geom_jitter()
```

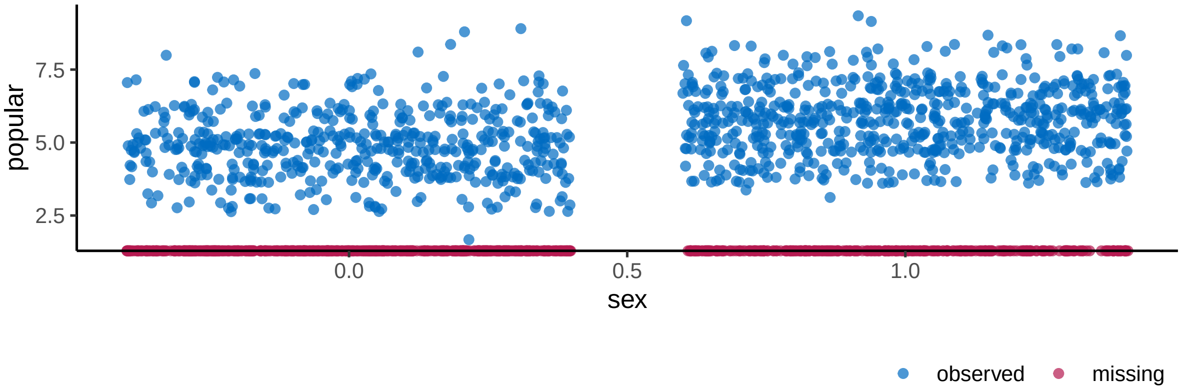


Figure 1: Scatterplot of student popularity by sex

where missing datapoints in the `popular` variable are represented by red points on the X-axis of the figure.

With the `ggmice` function `plot_pattern` we can visualize the missing data pattern

```
R> plot_pattern(dat)
```

which shows us that the missingness is univariate and sporadic.

To develop the best imputation model for the incomplete variable `popular`, we need to know whether the observed values of `popular` are related to observed values of other variables. Plot the pair-wise complete correlations in the incomplete data

```
R> plot_corr(dat)
```

This shows us that not just the analysis-model variable `sex`, but also the cluster-level covariate teacher experience, `texp`, may be a useful as an imputation model predictor. Moreover, the missingness in `popular` may depend on the observed values of other variables. With `ggmice()` we can visualize the distribution of the teacher experience for cases where `popular` is observed and cases where `popular` is missing.

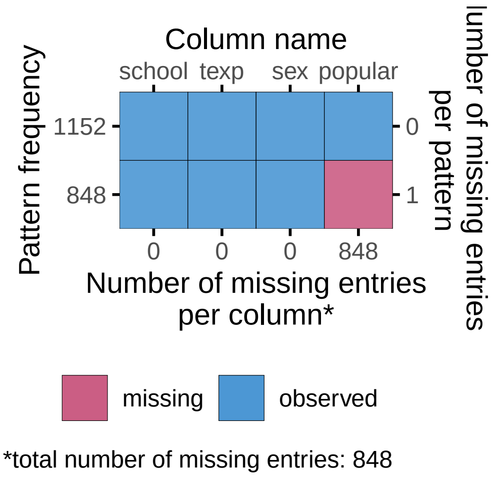


Figure 2: Missing data pattern.

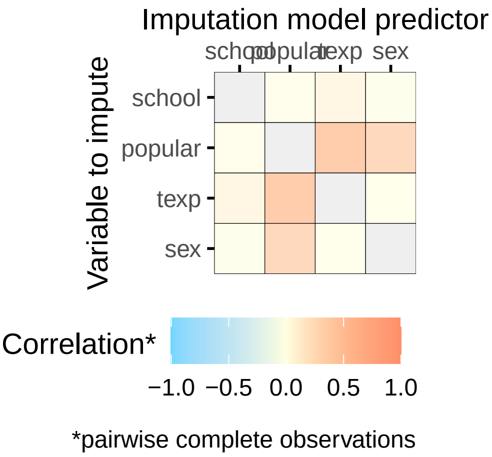
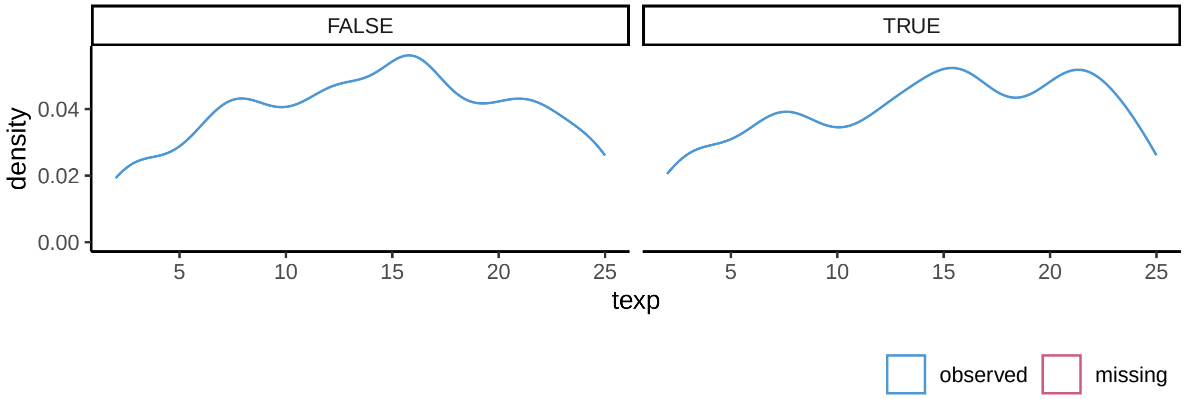


Figure 3: Pair-wise correlations.

```
R> ggmice(dat, aes(texp)) +
+   geom_density() +
+   facet_wrap(~is.na(popular))
```

It appears that students with a missing value for `popular` are in clusters with a slightly higher `texp` value.

```
t.test(dat$texp ~ is.na(dat$popular)) |>
  tidy() |>
  kable()
```



estimate	estimate1	estimate2	statistic	p.value	parameter	conf.low	conf.high	method	alternative
- 0.2599581	14.15278	14.41274	- 0.8731291	0.3827095	1796.115	- 0.8438947	0.3239786	Welch Two Sample t-test	two.sided

Although there are no significant differences in the distribution of `texp` depending on the missingness indicator of `popular`, this variable can serve as auxiliary variable in the imputation of `popular`.

```
R> meth <- make.method(dat)
R> meth
```

```
 school popular    texp     sex 
     ""   "pmm"      ""      "" 
```

```
R> pred <- quickpred(dat)
R> pred
```

```
        school popular texp sex
school       0       0    0   0
popular      0       0    1   1
texp         0       0    0   0
sex          0       0    0   0
```

Adjust the methods vector.

```
R> meth["popular"] <- "2l.pmm"
```

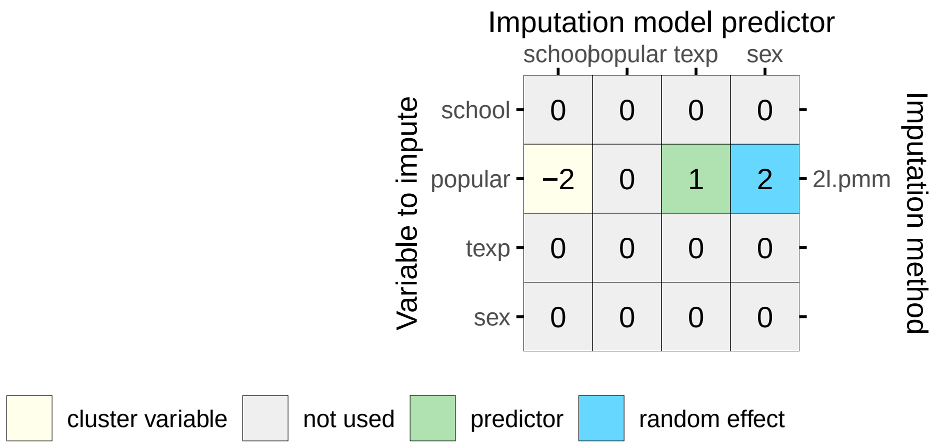
The `pmm` method is better (more efficient) because it will still look for donors (maybe outside of cluster) based on predictive distance, even for very small clusters.

Adjust the predictor matrix.

```
R> pred["popular", "school"] <- -2
R> pred["popular", "sex"] <- 2
```

Visualize the imputation methods and predictors.

```
plot_pred(pred, method = meth)
```



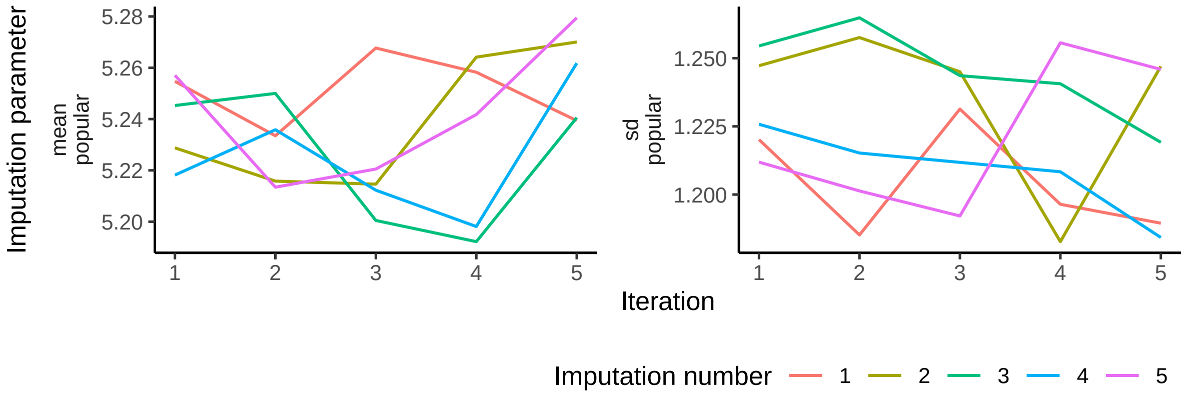
	school	popular	texp	sex
school	0	0	0	0
popular	−2	0	1	2
texp	0	0	0	0
sex	0	0	0	0

Impute the data.

```
R> imp <- mice(
+  data = dat,
+  method = meth,
+  predictorMatrix = pred,
+  printFlag = FALSE
+)
```

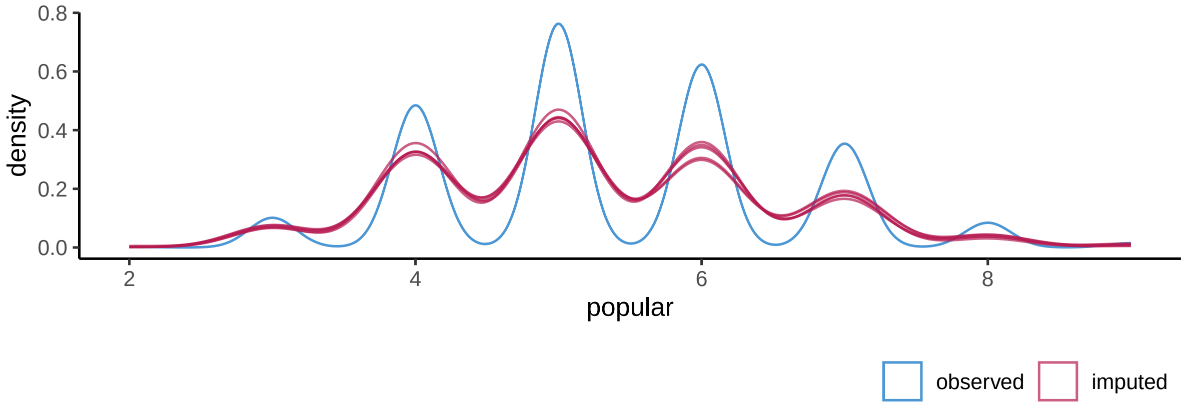
Evaluate the convergence.

```
R> plot_trace(imp)
```



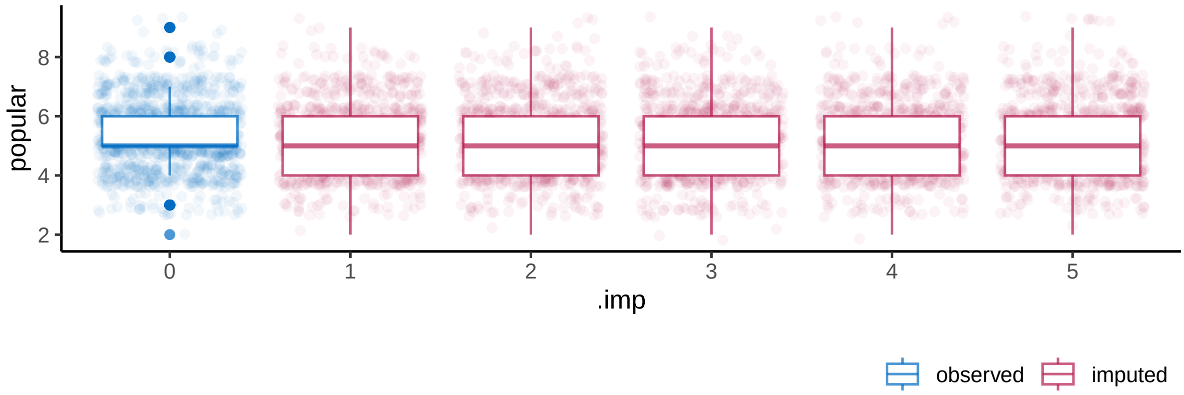
Troubleshoot non-convergence of imputation model. Evaluate the distribution of imputed values.

```
R> ggmice(imp, aes(popular, group = .imp)) +
+   geom_density()
```



Evaluate the distribution of imputed values.

```
R> ggmice(imp, aes(.imp, popular)) +
+   geom_jitter(alpha = 0.05) +
+     geom_boxplot()
```

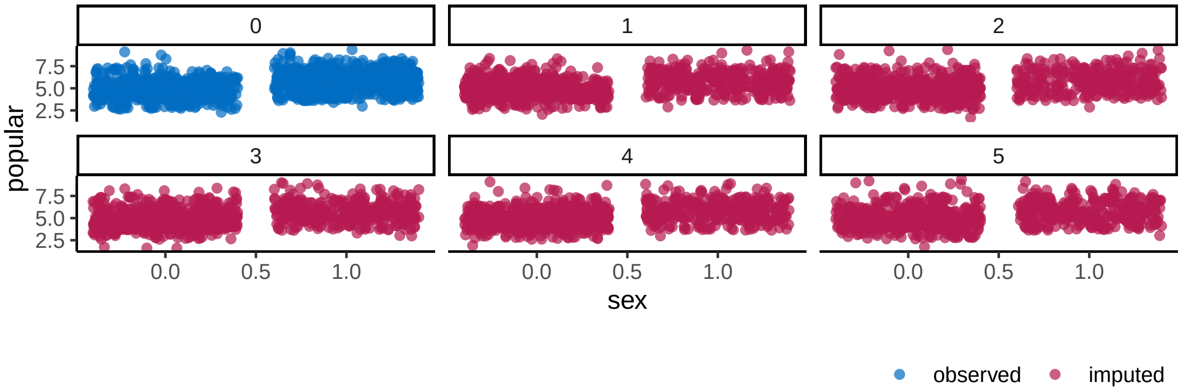


```
R> ggmice(imp, aes(sex, popular)) +
+   geom_jitter() +
+   facet_wrap(~ .imp)
```

Analyze the imputed data.

```
R> fit <- with(
+   imp,
+   lmer(popular ~ texp + sex + (1 | school))
+)
```

Pooling the estimates does not provide estimates of the variance components.



```
R> pool(fit)
```

```
Class: mipo    m = 5
         term m   estimate         ubar            b            t dfcom
1 (Intercept) 5 3.59902513 0.0284760238 3.461284e-03 0.0326295651  1995
2        texp 5 0.09225928 0.0001140591 2.273382e-05 0.0001413397  1995
3         sex 5 0.85187018 0.0009628869 4.946825e-04 0.0015565060  1995
        df       riv    lambda       fmi
1 216.1756 0.1458610 0.1272938 0.1352573
2 100.6502 0.2391794 0.1930144 0.2085857
3  26.9008 0.6164992 0.3813792 0.4227574
```

Therefore, `mitml` is used.

```
R> est <- testEstimates(as.mitml.result(fit), extra.pars = TRUE)
```

Display results in table.

```
R> est$estimates |>
+   round(3) |>
+   kable()
```

	Estimate	Std.Error	t.value	df	P(>|t|)	RIV	FMI
(Intercept)	3.599	0.181	19.924	246.857	0	0.146	0.134
texp	0.092	0.012	7.760	107.369	0	0.239	0.208
sex	0.852	0.039	21.592	27.501	0	0.616	0.422

```
R> est$extra.pars |>
+   round(3) |>
+   kable()
```

	Estimate
Intercept~~Intercept|school	0.470
Residual~~Residual	0.462
ICC|school	0.504

4.3. IMPACT data

The second case study is the `impact` data from the **metamisc** package (empirical data on traumatic brain injuries, $n = 11,022$ units across $N = 15$ clusters, Debray and de Jong 2021).

The `impact` data set contains traumatic brain injury data on $n = 11022$ patients clustered in $N = 15$ studies with the following 11 variables:

- `name` Name of the study,
- `type` Type of study (RCT: randomized controlled trial, OBS: observational cohort),
- `age` Age of the patient,
- `motor_score` Glasgow Coma Scale motor score,
- `pupil` Pupillary reactivity,
- `ct` Marshall Computerized Tomography classification,
- `hypox` Hypoxia (0=no, 1=yes),
- `hypots` Hypotension (0=no, 1=yes),
- `tsah` Traumatic subarachnoid hemorrhage (0=no, 1=yes),
- `edh` Epidural hematoma (0=no, 1=yes),
- `mort` 6-month mortality (0=alive, 1=dead).

Check if there is systematic missingness in this dataset. For illustration purposes, we made Marshall Computerized Tomography classification (ct) systematically missing.

The analysis model for this dataset is a prediction model with `mort` as the outcome. In this tutorial we'll estimate the adjusted prognostic effect of `ct` on mortality outcomes. The estimand is the adjusted odds ratio for `ct`, after including `type`, `age` `motor_score` and `pupil` into the analysis model:

```
mort ~ type + age + motor_score + pupil + ct + (1 | name)
```

In this dataset all the variables are level-1 (patient), except by the level-2 (study) type variable. The analysis model for this dataset is a prediction model with `mort` as the outcome. In this tutorial we'll estimate the adjusted prognostic effect of `ct` on mortality outcomes. The estimand is the adjusted odds ratio for `ct`, after including `type`, `age` `motor_score` and `pupil`. Therefore the main model for the i-th patient from the j-th study can be described as:

$$mort_{ij} = (\beta_0 + b_i) + \beta_1 type_{.j} + \beta_2 age_{ij} + \beta_3 motorscore_{ij} + \beta_4 pupil_{ij} + \beta_5 ct_{ij} + \epsilon_{ij}$$

where ϵ_{ij} corresponds to the error term.

Note that variables `hypots`, `hypox`, `tsah` and `edh` are not part of the analysis model, and may thus serve as auxiliary variables for imputation.

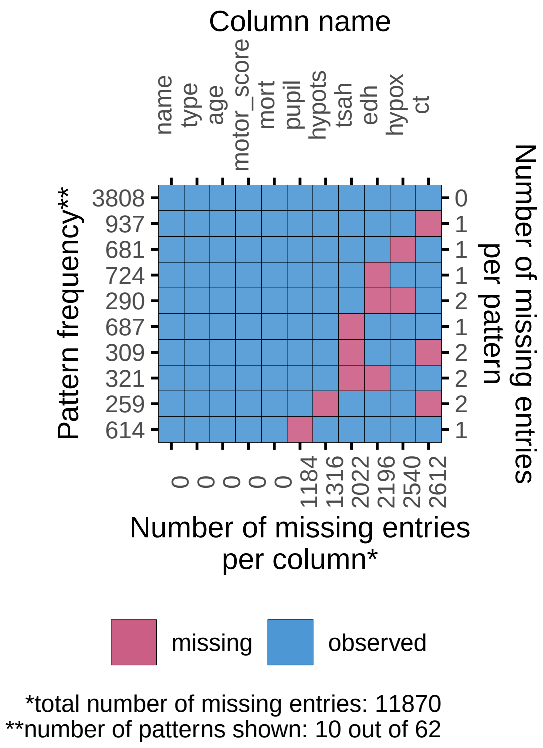
The `impact` data included in the **metamisc** package is a complete data set. The original data has already been imputed once (Steyerberg et al, 2008). For the purpose of this tutorial we have induced missingness (mimicking the missing data in the original data set before imputation). The resulting incomplete data can be accessed from zenodo link to be created.

Load the incomplete data into the R workspace:

```
R> dat <- read.table("link/to/the/data.txt")
```

To explore the missingness, we should look at the missing data pattern. The ten most frequent missingness patterns are shown with

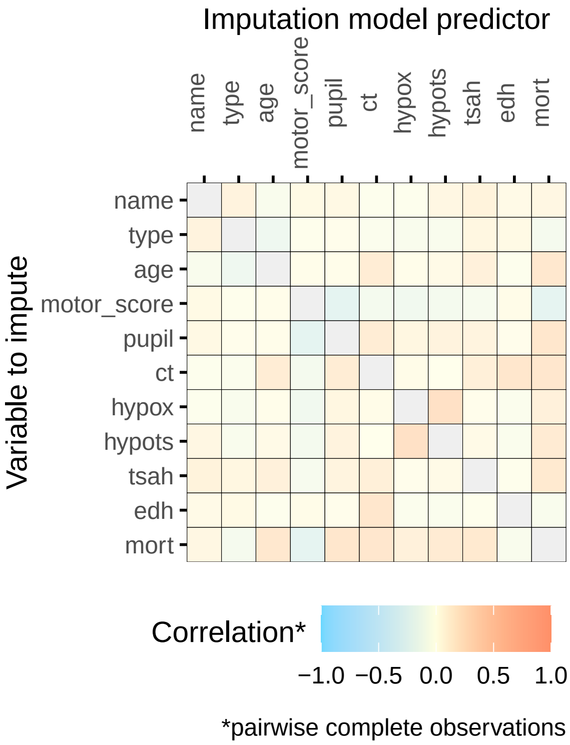
```
R> plot_pattern(dat, rotate = TRUE, npat = 10L)
```



This shows that we need to impute `ct` and `pupil`. To develop the best imputation model, we need to investigate the relations between the observed values of the incomplete variables and the observed values of other variables, and the relation between the missingness indicators of the incomplete variables and the observed values of the other variables. To see whether the missingness depends on the observed values of other variables, we can test this statistically or use visual inspection (e.g. a histogram faceted by the missingness indicator).

We should impute the variables `ct` and `pupil` and any auxiliary variables we might want to use to impute these incomplete analysis model variables. We can evaluate which variables may be useful auxiliaries by plotting the pairwise complete correlations

```
R> plot_corr(dat, rotate = TRUE)
```



This shows us that `hypox` and `hypot` would not be useful auxiliary variables for imputing `ct`. Depending on the minimum required correlation, `tsah` could be useful, while `edh` has the strongest correlation with `ct` out of all the variables in the data and should definitely be included in the imputation model. For the imputation of `pupil`, none of the potential auxiliary variables has a very strong relation, but `hypots` could be used. We conclude that we can exclude `hypox` from the data, since this is neither an analysis model variable nor an auxiliary variable for imputation

```
R> dat <- select(dat, !hypox)
```

Mutate data to get the right data types for imputation (e.g. integer for clustering variable).

```
R> # dat <- mutate(
R> #   dat,
R> #   name = as.integer(name))
```

This is necessary because otherwise PMM cannot be used for these factor variables.

```
R> dat <- mutate(
+  dat,
+  across(everything(), as.numeric))
```

Create an initial methods vector for the incomplete variables

```
R> meth <- make.method(dat)
R> meth
```

```
       name        type         age motor_score       pupil          ct 
         ""          ""          ""          ""       "pmm"       "pmm" 
     hypots        tsah         edh        mort 
      "pmm"       "pmm"       "pmm"          "" 
```

which should be adjusted to the appropriate `2l` methods.

```
R> # meth[c("pupil", "ct")] <- "2l.pmm"
R> # meth[c("hypots", "tsah", "edh")] <- "2l.pmm"
R> meth[meth == "pmm"] <- "2l.pmm"
```

Create an initial predictor matrix

```
R> pred <- quickpred(dat)
```

This predictor matrix is too large to display inline. A visualization of the adapted predictor matrix is presented in Figure XYZ.

We should make sure `name` is used as clustering variable

```
R> pred[, "name"] <- -2
```

and the analysis-model outcome should be used as a predictor in all imputation models

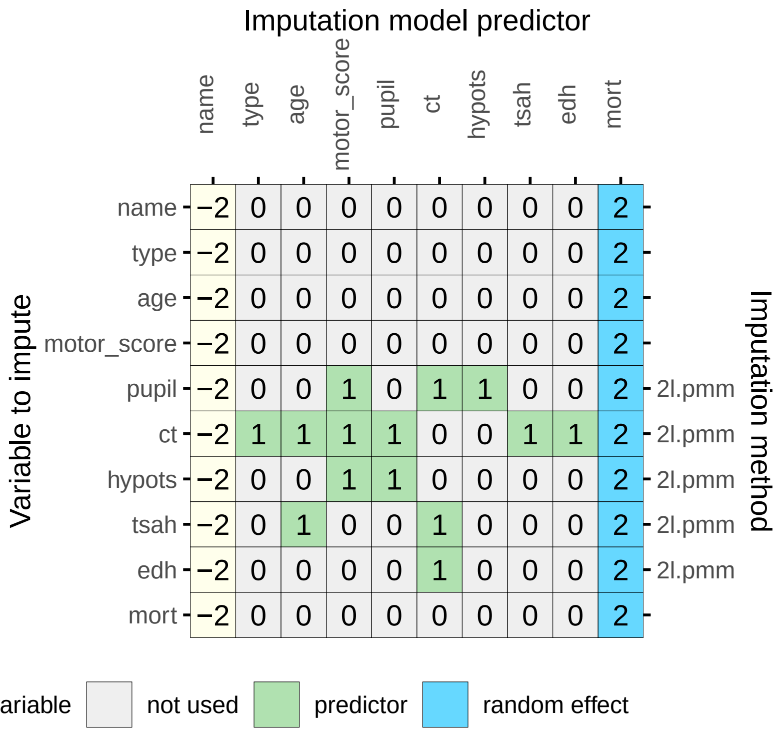
```
R> pred[, "mort"] <- 2
R> # pred[pred == 1] <- 2
```

the resulting predictor matrix is visualized with

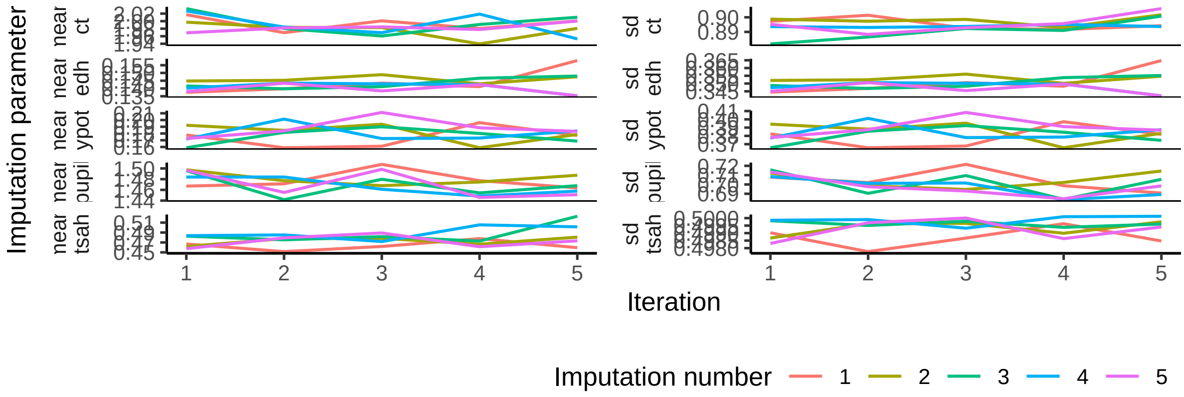
```
R> plot_pred(pred, method = meth, rotate = TRUE)
```

Impute the incomplete data with

```
R> imp <- mice(
+  dat,
+  method = meth,
+  predictorMatrix = pred,
+  printFlag = FALSE
+)
```



	name	type	age	motor_score	pupil	ct	hypots	tsah	edh	mort	
name	−2	0	0	0	0	0	0	0	0	2	
type	−2	0	0	0	0	0	0	0	0	2	
age	−2	0	0	0	0	0	0	0	0	2	
motor_score	−2	0	0	0	0	0	0	0	0	2	
pupil	−2	0	0	1	0	1	1	0	0	2	2l.pmm
ct	−2	1	1	1	1	0	0	1	1	2	2l.pmm
hypots	−2	0	0	1	1	0	0	0	0	2	2l.pmm
tsah	−2	0	1	0	0	1	0	0	0	2	2l.pmm
edh	−2	0	0	0	0	1	0	0	0	2	2l.pmm
mort	−2	0	0	0	0	0	0	0	0	2	



Evaluate the convergence of the algorithm

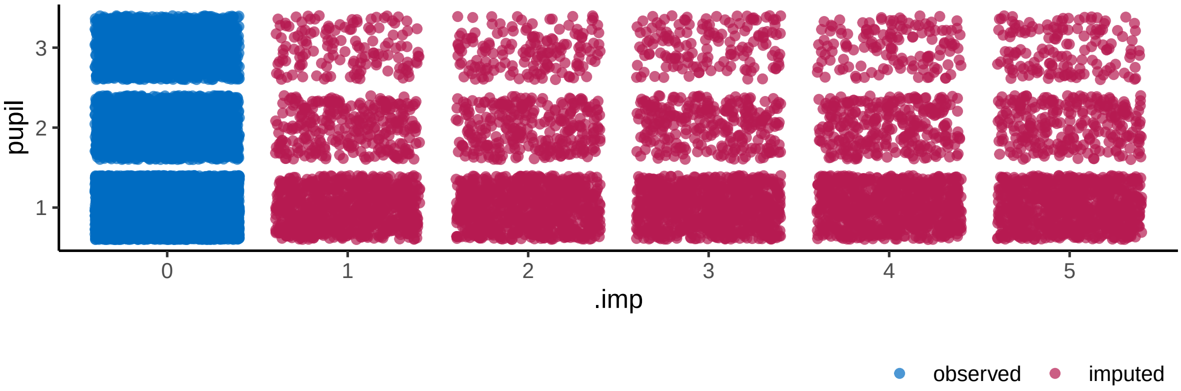
```
R> plot_trace(imp)
```

Evaluate the imputed values.

```
R> ggmice(imp, aes(.imp, pupil)) +
+  geom_jitter()
```

Convert the data back to factors.

```
long <- complete(imp, "long", include = TRUE)
long <- mutate(long, across(c("motor_score", "pupil", "ct"), as.factor))
imp <- as.mids(long)
```



Analyze the imputed data:

```
R> fit <- imp %>%
+  with(glmer(
+    mort ~ type + age + motor_score + pupil + ct + (1 | name),
+    family = "binomial"
+    ))
```

The estimated effects after imputation are presented in Table XYZ.

```
R> est <- testEstimates(as.mitml.result(fit), extra.pars = TRUE)
```

Display results in table.

```
R> est$estimates |>
+  round(3) |>
+  kable()
```

	Estimate	Std.Error	t.value	df	P(>|t|)	RIV	FMI
(Intercept)	-1.910	0.335	-5.700	9617.768	0.000	0.021	0.021
type	-0.347	0.179	-1.941	256541.695	0.052	0.004	0.004
age	0.032	0.002	19.550	1965.628	0.000	0.047	0.046
motor_score2	-0.576	0.070	-8.195	14898.703	0.000	0.017	0.017
motor_score3	-0.892	0.072	-12.390	14760.294	0.000	0.017	0.017
motor_score4	-1.272	0.074	-17.203	10235.157	0.000	0.020	0.020
pupil2	1.311	0.074	17.758	41.455	0.000	0.451	0.342
pupil3	0.661	0.077	8.533	165.867	0.000	0.184	0.165
ct2	0.783	0.075	10.461	264.558	0.000	0.140	0.130
ct3	0.786	0.095	8.280	10.443	0.000	1.624	0.676

```
R> est$extra.pars |>
+  round(3) |>
+  kable()
```

	Estimate
Intercept~~Intercept|name	0.083

4.4. Obesity data

In this example, we demonstrate a multilevel imputation of random intercept and random slope model with a continuous response. We utilize the obesity dataset included in the `micemd`@ package, a simulated dataset that emulates an electronic survey in which individuals are asked to provide information about their weight and consumption habits in different countries. We use the following variables from the dataset:

- **Cluster:** Region of the patients' healthcare provider (Cluster variable),
- **Gender:** Subjects' Gender (0=male, 1=female),
- **Age:** Subjects' age,
- **Height:** Subjects' height in metres,
- **Weight:** Subjects' weight in kilograms,
- **BMI:** Subjects' body mass index,
- **FamOb:** Family obesity history (yes or no),
- **Time:** Response time in minutes (exclusion-restriction variable).

In this dataset, Age and FamOb are MAR, while the weight variable is affected by selection bias, attributed to an indirect MNAR mechanism. This MNAR mechanism typically arises when an unobserved or omitted variable influences both the value of the incomplete variable (in this case, Weight) and its likelihood of being missing (denoted as R).

In the primary analysis model, BMI serves as the dependent variable, with Age, Gender, and FamOb as predictors. Because of the clustered nature of the data, which is quantified with the Intraclass Correlation Coefficient (ICC) below, we include random intercepts, as well as a random slope for the Age variable. The model is represented as:

$$BMI_{ij} = (\beta_o + b_{oj}) + (\beta_1 + b_{oj}) * Age_{ij} + \beta_2 * FamOb_{ij} + \beta_3 Gender_{ij} + \epsilon_{ij} \tag{2}$$

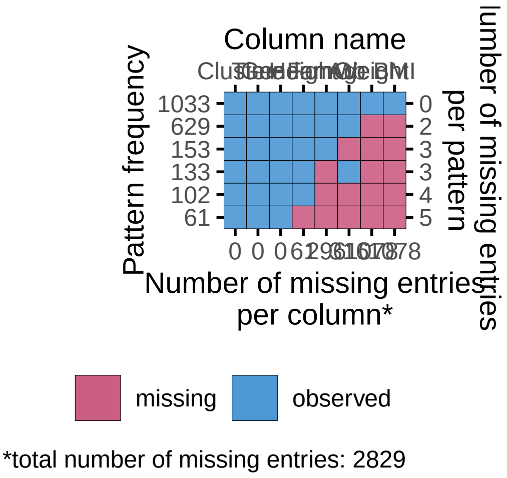
We start by loading the data:

```
R> data(Obesity, package = "micemd")
```

Now, let's begin by examining the missing patterns in the data by cluster:

```
R> plot_pattern(Obesity)
```

```
R> # Obesity |>
R> #   split(~Cluster) |>
R> #   lapply(plot_pattern)
```



We observe that the missing pattern is non-monotonic and quite similar across the clusters. However, regarding the weight variable, we notice that it is systematically missing in cluster 3. In order to evaluate if we require a imputation method that accounts for clustering we assess the Intraclass Correlation

```
R> Nulmodel <- lme4::lmer(BMI ~ 1 + (1|Cluster), data = Obesity)
R> performance::icc(Nulmodel)
```

```
# Intraclass Correlation Coefficient

    Adjusted ICC: 0.362
  Unadjusted ICC: 0.362
```

As the ICC is greater than 0.1 and as we will use a mixed model for the analysis, we decide to use two-level (2l) imputation methods. In this imputation process, we include all predictor variables from equation 2 in the main model. However, since BMI is a composite of weight and height, we use deterministic imputation for these, which is described below.

We use the **find.defaultMethodfunction** provided in the **micemd** package, which suggests an appropriate method for MAR variables based on the type of variable, number of observations in the cluster, and number of clusters.

It suggests using '2l.2stage.bin' for the FamOb variable and '2l.2stage.norm' for the age variable. However, after inspecting the age density plot, we consider modifying its method to '2l.2stage.pmm'. For the BMI variable, we employ deterministic imputation.

```
meth_mar <-
  find.defaultMethod(
    Obesity,
    ind.clust = 1,
    I.small = 7,
    ni.small = 100,
    prop.small = 0.4
  )
meth_mar["BMI"] <- "~ I(Weight / (Height)^2)"
meth_mar["Age"] <- "2l.2stage.pmm"
meth_mar["Weight"] <- "2l.2stage.pmm"
```

For these imputation models, it is necessary to specify the prediction matrix, with the cluster variable labelled as -2 and the predictor variable measured within clusters labelled as 2, encompassing all variables. We need to supress the variable Time as this variable is not specified in the main model. We also modify the relationship between BMI, weight and height in the prediction matrix to avoid circular predictions. Then we proceed to run the imputation model.

```
pred_mar <- quickpred(Obesity)
pred_mar["Weight",] <- 2
pred_mar["Weight","Weight"] <- 0
pred_mar[, "Cluster"] <- -2
pred_mar[, "Time"] <- 0
pred_mar[pred_mar == 1] <- 2

pred_mar[c("Height", "Weight"), "BMI"] <- 0
#plot_pred(pred_mar)
imp_mar <-
  mice(
    data = Obesity,
    method = meth_mar,
    predictorMatrix = pred_mar,
    printFlag = FALSE
  )
```

After confirming convergence, we proceed to save the results for future use.

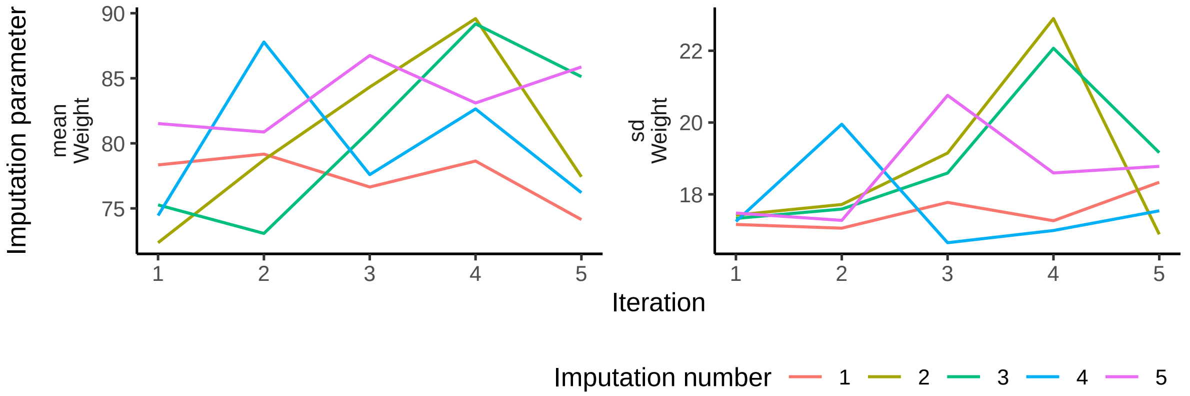
```
summary(complete(imp_mar, "long")$Weight)
```

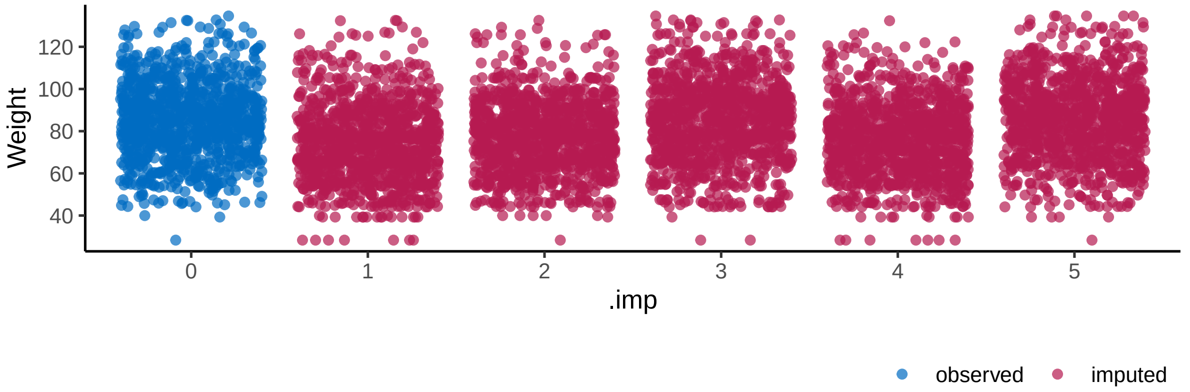
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  28.35   69.38   81.96   82.36   94.46  134.61 
```

```
plot_trace(imp_mar, "Weight")
```

```
ggmice(imp_mar, aes(.imp, Weight)) +
  geom_jitter()
```

We consider the possibility that patients may not have been selected randomly, which would then have led to a distribution for weight that does not reflect the weight in the population.





It's likely that an omitted variable, like self-esteem, could influence this selection. For instance, individuals with lower self-esteem might have higher weight values, impacting their willingness to provide honest information due to embarrassment.

To address this situation where data are Missing Not At Random (MNAR), one approach is to apply the Heckman selection model. This method has recently been extended to allow for variations in intercepts and exposure effects (random intercept and slope) Galimard, Chevret, Curis, and Resche-Rigon (2018); Muñoz, Hufstedler, Gustafson, Bärnighausen, De Jong, and Debray (2023b). This method involves specifying two equations: one for the outcome, describing the incomplete variable in terms of partially observed predictors (in this case, all variables from the main model), and the other for the selection model, explaining the probability of being observed based (R) on certain variables.

To apply the **2l.2stage.heckman** method, the weight variable should be specified as '2l.2stage.heckman' found in the micemd package. Additionally, the prediction matrix needs modification, as this model includes one submodel for the individuals' measured outcome, and one for the selection of individuals. For the outcome equation we consider the same imputation model that we used for the MAR case (main model).

$$Weight_{ij} = \beta_o^O + \beta_1^O Age_{ij} + \beta_2^O FamOb_{ij} + \beta_3^O Gender_{ij} + \epsilon_{ij}^O$$

Regarding the selection equation, we include the same predictors as those in the main model, as well as a time variable. Here the time variable serves as a restriction exclusion variable specifically explaining the probability of being observed but not affecting the incomplete value (Weight). In this context, we assume that the time a user spends completing the survey serves as a proxy for the barriers they may encounter in survey completion, such as familiarity with the survey content or internet speed. These factors may lead the user to skip specific questions or even the entire survey. Also, we assume the time does not have any influence on the subject's weight.

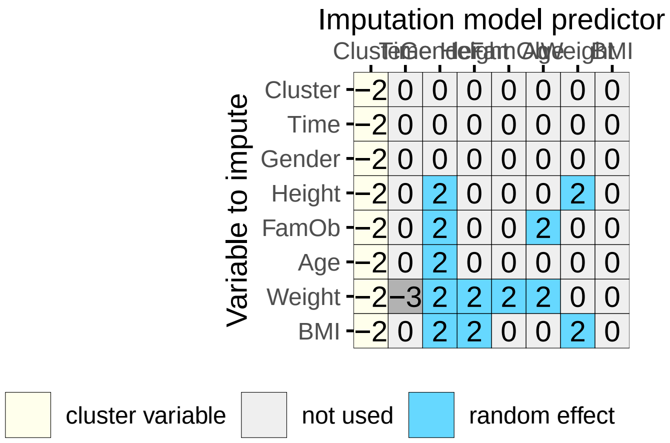
$$R_{ij} = \beta_o^S + \beta_1^S Age_{ij} + \beta_2^S FamOb_{ij} + \beta_3^S Gender_{ij} + \beta_4^S Time_{ij} + \epsilon_{ij}^S$$

These two equations are jointly estimated under the assumption that the error terms are interconnected with a bivariate normal distribution. For a more comprehensive understanding of the model and the exclusion restriction, see Muñoz *et al.* (2023b).

To use information from both equations, we must adjust the prediction matrix. The cluster variable remains specified as before (-2). In this imputation method, all the variables present in both the selection and outcome equations are included with a random effect.

However, it is essential to distinguish which of these variables appear in each equation. In this framework, when a variable is shared between both equations, it is denoted as (2). Predictors exclusive to the outcome equation are indicated as (-4), while those exclusive to the selection equation are labelled as (-3). Consequently, the only alteration needed in the predictor matrix pertains to the variable 'Time'.

```
pred_mnar <- pred_mar
pred_mnar["Weight","Time"] <- -3
plot_pred(pred_mnar)
```



	Cluster	Time	Gender	Height	FamOb	Age	Weight	BMI
Cluster	−2	0	0	0	0	0	0	0
Time	−2	0	0	0	0	0	0	0
Gender	−2	0	0	0	0	0	0	0
Height	−2	0	2	0	0	0	2	0
FamOb	−2	0	2	0	0	2	0	0
Age	−2	0	2	0	0	0	0	0
Weight	−2	−3	2	2	2	2	0	0
BMI	−2	0	2	2	0	0	2	0

We also need to modify the method of the weight variable.

```
meth_mnar <- meth_mar
meth_mnar["Weight"] <- "2l.2stage.heckman"
```

As before we use the 'pmm', option but this time for the Heckman imputation, this approach ensures that the imputed values remain within the range of observable values. We then run the MNAR imputation model by setting the pmm parameter to true.

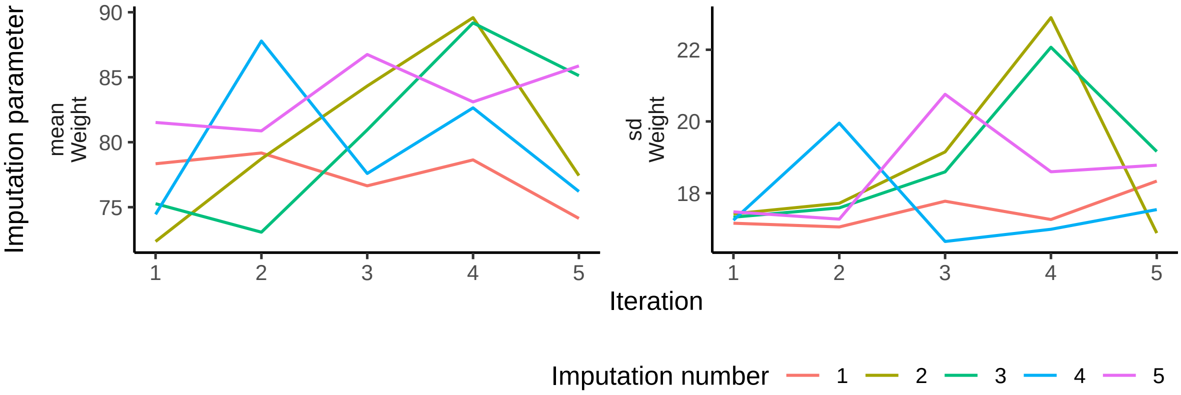
```
imp_mnar <-
  mice(
    data = Obesity,
    method = meth_mnar,
    predictorMatrix = pred_mnar,
    pmm = T,
    printFlag = FALSE
  )
```

We check the convergence of the results

```
summary(complete(imp_mnar, "long")$Weight)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  28.35   67.39   82.57   81.71   96.04  134.61 
```

```
plot_trace(imp_mar, "Weight")
```



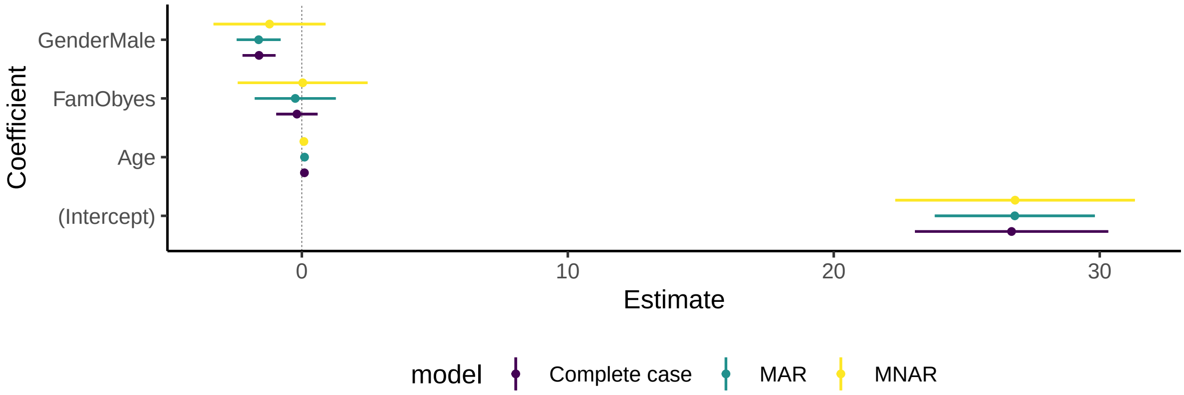
We proceed to compare the effects on the model. We run the analysis model on each of the completed datasets as well as the dataset where the incomplete values are removed (Complete Case analysis, CC).

We proceed to fit the mixed model by using the complete case database and the imputed datasets (MAR,MNAR), and we compare the estimates of each dataset by using a plot (the plot function can be found in the appendix).

```
# Save all datasets in a list
list_data <- list(Obesity |> filter(complete.cases(Obesity)), # complete case
                  imp_mar, # imputed under mar
                  imp_mnar) # imputed under mnar

# Run mixed models on each dataset
list_models <- lapply(list_data,
                   FUN = function(x)
```

```
                        with(x,nlme::lme(BMI ~ Age + FamOb + Gender,
                                         random =  ~ 1 + Age | Cluster)))
# Plot coefficients
plot_models(list_models,
            mod_name = c("Complete case", "MAR", "MNAR"))
```



We note that there is minimal disparity in the age effect, FamObs, or Gender across the various imputation models under consideration. An analysis of the intercept reveals that, under the MNAR assumption, a higher average BMI is anticipated compared to the MAR assumption. Nonetheless, with respect to precision of the estimates, we notice that in general assuming MNAR leads to wider confidence intervals. In this case it does not have any influence on the final result but there could be cases where variation in the assumed missing mechanism could lead also to differences on the significance of a statistical test and therefore lead to different conclusions.

5. Conclusion

This paper is dedicated to exploring the imputation process for incomplete datasets, with a primary focus on utilizing a hierarchical model for analysis. Initially, users are encouraged to consider the main analysis within the context of the incomplete dataset, along with insights provided by domain experts, to gain a better understanding of variable relationships. Employing clear data visualization is instrumental in comprehending the missing data patterns, establishing a missing mechanism, and aiding in the selection of suitable imputation methods.

The “Mice” and “Mice”-based R packages offer a range of imputation methods tailored for hierarchical data, easily adaptable to the dataset’s structure. Before proceeding with the analysis of the imputed dataset, it is essential to assess the convergence of the imputation method. This evaluation can reveal issues such as circular problems, the need for additional iterations, or challenges associated with the chosen imputation method.

6. Summary and discussion

What is missing from this manuscript...

Computational details

The results in this paper were obtained using R~4.3.0. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at [https://CRAN.R-project.org/].

7. Disclaimer

The views expressed in this paper are the personal views of the authors and may not be understood or quoted as being made on behalf of or reflecting the position of the regulatory agency/agencies or organizations with which the authors are employed/affiliated.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under ReCoDID grant agreement No 825746.

References

More technical details

Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*).
For more technical style details, please check out JSS's style FAQ at [https://www.jstatsoft.org/pages/view/style#frequently-asked-questions] which includes the following topics:

- Title vs. sentence case.
- Graphics formatting.
- Naming conventions.
- Turning JSS manuscripts into R package vignettes.
- Trouble shooting.
- Many other potentially helpful details…

Using BibTeX

References need to be provided in a BibTeX file (`.bib`). All references should be made with `@cite` syntax. This commands yield different formats of author-year citations and allow to include additional details (e.g.,pages, chapters, …) in brackets. In case you are not familiar with these commands see the JSS style FAQ for details.
Cleaning up BibTeX files is a somewhat tedious task – especially when acquiring the entries automatically from mixed online sources. However, it is important that informations are complete and presented in a consistent style to avoid confusions. JSS requires the following format.

- item JSS-specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- item Titles should be in title case.
- item Journal titles should not be abbreviated and in title case.
- item DOIs should be included where available.
- item Software should be properly cited as well. For R packages `citation("pkgname")` typically provides a good starting point.

References

(????). "FAQ: Accounting for Clustering with Mi Impute | Stata." https://www.stata.com/support/faqs/statistics/clustering-and-mi-impute/.

Audigier V, White I, Jolani S, Debray T, Quartagno M, Carpenter J (????). "Comparison of Multiple Imputation Methods for Systematically and Sporadically Missing Multilevel Data." p. 20.

Bailey BE, Andridge R, Shoben AB (2020). "Multiple Imputation by Predictive Mean Matching in Cluster-Randomized Trials." *BMC Medical Research Methodology*, **20**(1), 72. ISSN 1471-2288. doi:10.1186/s12874-020-00948-6.

Cai M, Van Buuren S, Vink G (2023). "Graphical and Numerical Diagnostic Tools to Assess Multiple Imputation Models by Posterior Predictive Checking." *Heliyon*, **9**(6), e17077. ISSN 24058440. doi:10.1016/j.heliyon.2023.e17077.

Debray T, de Jong V (2021). "Metamisc: Meta-Analysis of Diagnosis and Prognosis Research Studies."

Eddings W, Marchenko Y (????). "FAQ: Accounting for Clustering with Mi Impute | Stata." https://www.stata.com/support/faqs/statistics/clustering-and-mi-impute/.

Galimard JE, Chevret S, Curis E, Resche-Rigon M (2018). "Heckman Imputation Models for Binary or Continuous MNAR Outcomes and MAR Predictors." *BMC Medical Research Methodology*, **18**(1), 90. ISSN 1471-2288. doi:10.1186/s12874-018-0547-1.

Graham JW (2009). "Missing Data Analysis: Making It Work in the Real World." *Annual Review of Psychology*, **60**(1), 549–576. ISSN 0066-4308, 1545-2085. doi:10.1146/annurev.psych.58.110405.085530.

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. ISSN 1094-4281. doi:10.1177/1094428117703686.

Hammon A (2022). "Multiple Imputation of Ordinal Missing Not at Random Data." *AStA Advances in Statistical Analysis*. ISSN 1863-818X. doi:10.1007/s10182-022-00461-9.

Hammon A, Zinn S (2020). "Multiple Imputation of Binary Multilevel Missing Not at Random Data." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **69**(3), 547–564. ISSN 0035-9254, 1467-9876. doi:10.1111/rssc.12401.

Hox J, van Buuren S, Jolani S (2015). "Incomplete Multilevel Data: Problems and Solutions." In J Harring, L Staplecton, S Beretvas (eds.), *Advances in Multilevel Modeling for Educational Research: Addressing Practical Issues Found in Real-World Applications*, CILVR Series on Latent Variable Methodology, pp. 39–62. Information Age Publishing Inc., Charlotte, NC. ISBN 978-1-68123-328-4.

Jolani S (2018). "Hierarchical Imputation of Systematically and Sporadically Missing Data: An Approximate Bayesian Approach Using Chained Equations." *Biometrical Journal. Biometrische Zeitschrift*, **60**(2), 333–351. ISSN 1521-4036. doi:10.1002/bimj.201600220.

Localio AR, Berlin JA, Ten Have TR, Kimmel SE (2001). "Adjustments for Center in Multicenter Studies: An Overview." *Annals of Internal Medicine*, **135**(2), 112–123. ISSN 0003-4819. doi:10.7326/0003-4819-135-2-200107170-00012.

Meng XL (1994). "Multiple-Imputation Inferences with Uncongenial Sources of Input." *Statistical Science*, **9**(4). ISSN 0883-4237. doi:10.1214/ss/1177010269.

Mistler SA, Enders CK (2017). "A Comparison of Joint Model and Fully Conditional Specification Imputation for Multilevel Missing Data." *Journal of Educational and Behavioral Statistics*, **42**(4), 432–466. ISSN 1076-9986, 1935-1054. doi:10.3102/1076998617690869.

Molenberghs G, Kenward MG (2007). *Missing Data in Clinical Studies*. 1 edition. Wiley. ISBN 978-0-470-84981-1 978-0-470-51044-5. doi:10.1002/9780470510445.

Muñoz J, Efthimiou O, Audigier V, De Jong VMT, Debray TPA (2023a). "Multiple Imputation of Incomplete Multilevel Data Using Heckman Selection Models." *Statistics in Medicine*, p. sim.9965. ISSN 0277-6715, 1097-0258. doi:10.1002/sim.9965.

Muñoz J, Hufstedler H, Gustafson P, Bärnighausen T, De Jong VMT, Debray TPA (2023b). "Dealing with Missing Data Using the Heckman Selection Model: Methods Primer for Epidemiologists." *International Journal of Epidemiology*, **52**(1), 5–13. ISSN 0300-5771, 1464-3685. doi:10.1093/ije/dyac237.

Rubin DB (1976). "Inference and Missing Data." *Biometrika*, **63**(3), 581–592. doi:10.2307/2335739.

Speidel M, Drechsler J, Sakshaug JW (2018). "Biases in Multilevel Analyses Caused by Cluster-Specific Fixed-Effects Imputation." *Behavior Research Methods*, **50**(5), 1824–1840. ISSN 1554-3528. doi:10.3758/s13428-017-0951-1.

Taljaard M, Donner A, Klar N (2008). "Imputation Strategies for Missing Continuous Outcomes in Cluster Randomized Trials." *Biometrical Journal*, **50**(3), 329–345. ISSN 1521-4036. doi:10.1002/bimj.200710423.

Van Buuren S (2018). *Flexible Imputation of Missing Data*. Chapman and Hall/CRC.

van Buuren S, Groothuis-Oudshoorn K (2021). "Mice: Multivariate Imputation by Chained Equations."

Yucel RM (2008). "Multiple Imputation Inference for Multivariate Multilevel Continuous Data with Ignorable Non-Response." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **366**(1874), 2389–2403. doi:10.1098/rsta.2008.0038.

Zhang J, Dashti SG, Carlin JB, Lee KJ, Moreno-Betancur M (2023). "Should Multiple Imputation Be Stratified by Exposure Group When Estimating Causal Effects via Outcome Regression in Observational Studies?" *BMC Medical Research Methodology*, **23**(1), 42. ISSN 1471-2288. doi:10.1186/s12874-023-01843-6.

Affiliation:

Hanne I. Oberman
Methodology and Statistics
Padualaan 14
Utrecht The Netherlands
E-mail: `h.i.oberman@uu.nl`
URL: https://www.hanneoberman.github.io

Johanna Muñoz
Julius Center for Health Sciences and Primary Care
Universiteitsweg 100
Utrecht The Netherlands

Valentijn M.T. de Jong
Julius Center for Health Sciences and Primary Care
Utrecht The Netherlands

Gerko Vink
Julius Center for Health Sciences and Primary Care
Universiteitsweg 100
Utrecht The Netherlands

Thomas P.A. Debray
Julius Center for Health Sciences and Primary Care
Universiteitsweg 100
Utrecht The Netherlands

Journal of Statistical Software
published by the Foundation for Open Access Statistics
MMMMMM YYYY, Volume VV, Issue II
`doi:10.18637/jss.v000.i00`

http://www.jstatsoft.org/
http://www.foastat.org/
Submitted: yyyy-mm-dd
Accepted: yyyy-mm-dd