




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 package **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 accommodated, 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 [R Core Team \(2017\)](#)

1.2. scope

This paper 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 (?).

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 ?) 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, ?);
- **impact** from the **metamisc** package (empirical data on traumatic brain injuries, $n = 11,022$ patients across $N = 15$ studies with data that are MAR, ?);
- **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 introduction (case study 1) and directly head to practical applications of multilevel imputation under MAR conditions (case study 2) 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 (?). 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.

Box 1. The intraclass correlation coefficient.

In R, multilevel models may be fitted using the package **lme4**. 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 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 (??). 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 (? and ?). We here consider the general case that data are MAR, and expand on certain MNAR situations.

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**.

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. (Stata: <https://www.stata.com/support/faqs/statistics/clustering-and-mi-impute/>) TODO: replace ref.

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 data, for instance, 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 modeled 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 random effects model is appropriated, i.e. the number of clusters is adequate. (Austin, 2018).

Here it is important to evaluate the assumptions imposed by the main model, for instance by using the cluster indicator strategy may lead to bias estimates when the model is based on a hierarchical model (Taaljard, 2008). Even when an imputation strategy congenial with the main model is preferred, it is important to consider whether it is appropriate for the data as a less complex imputation strategies may also lead to unbiased estimates in certain scenarios (Bailey 2020). For instance, in causal effect analysis, separately imputation may

lead to smaller bias when the size of the smaller exposure cluster is large, compared with an imputation model that includes exposure-confounder interactions. (Zhang,2023).

Below we provide a imputation workflow that can be used in general to impute cluster data.

3.1. Focus on the Main Analysis

When dealing with incomplete clustered data, start by looking at your research questions and planned analysis. Pretend there are no missing data at first. This will help you understand your research hypotheses, the main statistical model, and give you insights into your data's structure (refer to the level table), variable types (e.g., confounders, auxiliary variables), and other considerations such as variable relationships, interactions or polynomial terms.

3.2. Exploration of Available Data

Now, explore what's in your data. Use simple tools like histograms and QQ plots to understand variable distributions, plausible range of values and identify errors or outliers. Check interactions between predictors with scatter plots, look for collinearity with VIF or correlations plots, this may also help you to identify non-considered relationships that may affect the main model. You might also consider to test assumptions related to your response variable such as variance homogeneity, independence between observations (eg. ACF, variograms). This will help you choose an imputation model that suits your data. In addition, the intra-class correlation coefficient (ICC) can be examined to assess cluster differences, aiding in the choice between the 2l and 1l methods for imputation.

Next, explore the missingness. Look at the **proportion of missing values** in the dataset variables. This helps find potential predictors and cut down on unnecessary variables in the imputation model. Doing this 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 appropriated imputation approach in terms of computational efficiency (e.g., simpler regression imputation versus FCS in univariate patterns).

3.3. Assess Estimation Procedure Robustness to Missing Data

Before diving into imputation, make sure your estimation model can handle missing data. Sometimes, simpler methods like complete case analysis might 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.?

3.4. Pre-imputation

Clean up your dataset before the imputation process. Keep initially the essential variables for your main model, but include extra variables if needed for specific procedures during analysis (e.g. confounders on balancing procedures). Think about adding other useful variables, even if they're not in the main model, such as instrumental or auxiliary variables which might boost your parameter estimates, especially if they're associated to the probability of missingness for some incomplete variables.

Figure out if you can directly impute incomplete variables by just use deductive imputation. This involves inferring missing values based on logical connections between variables. It's especially handy 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). In these situations, you can guess missing information from metadata or by using or through deduction based on time or protocols. For example, you could deduce missing test values for patients who have passed away.

3.5. 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 ?.

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 ?. 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 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 ?.

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 ??. 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 ?.

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 cluster ones, you can use the micemd package's `find.defaultMethod()` function. This function

selects from different 21 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 ?. 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 [REF]. 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
-2	Cluster variable, in this case the one defined by j index
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 $x_{5_{ij}}$
-3.	Random variable only included on selection model (Heckman model)
-4.	Random variable only included on main model (Heckman model)

- 3 and 4 type have been advised to used as the inclusion of the means of the cluster is beneficial on FCS ?

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 convergence 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) (§ 6.4).

For some interaction terms, it has been suggested, for instance, for treatment interaction effects, to conduct separate imputation by treatment group. Additionally, it can be used imputation models based on random forest or deep learning that can handle interaction and non-linear terms without requiring the explicit specification of an imputation model.

3.6. Post-Imputation

During the imputation process, certain issues may arise that halt the process. In hierarchical model imputations, many issues are related to over fitting 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 using step by step the previous referred recepies or by using functions like quickpred. Another option is to consider variable transformations, such as scaling when 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.

3.7. Convergence and Sensitivity Analysis

Before starting into 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.

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.???

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()
```

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)
```

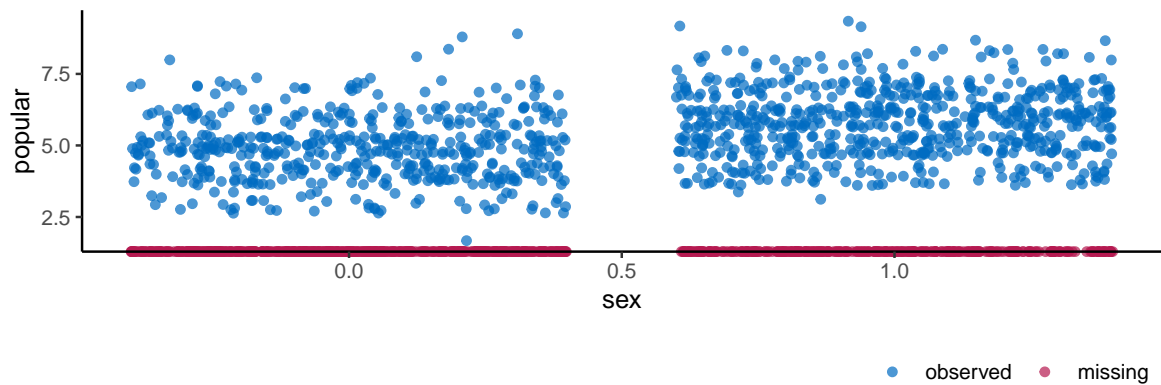


Figure 1: Scatterplot of student popularity by sex

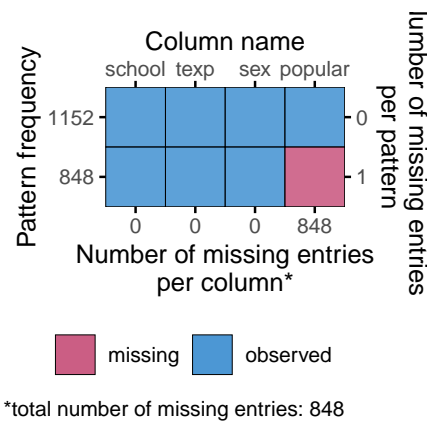


Figure 2: Missing data pattern.

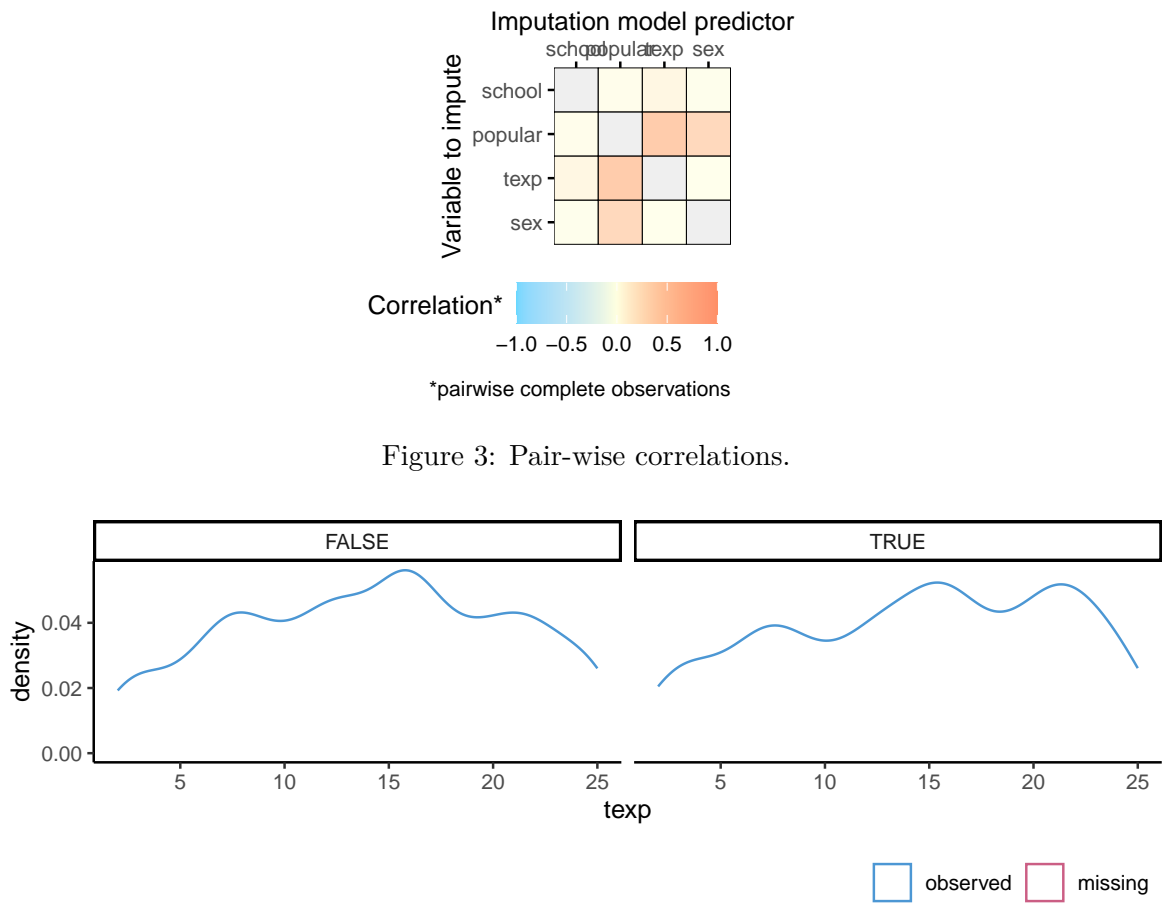
which shows us that the missingness is univariate and sporadic.

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.

```
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
- 14.152781	14.41274	- 0.382709	5796.115	- 0.323978	Welch Two	0.2599581	0.8438947	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

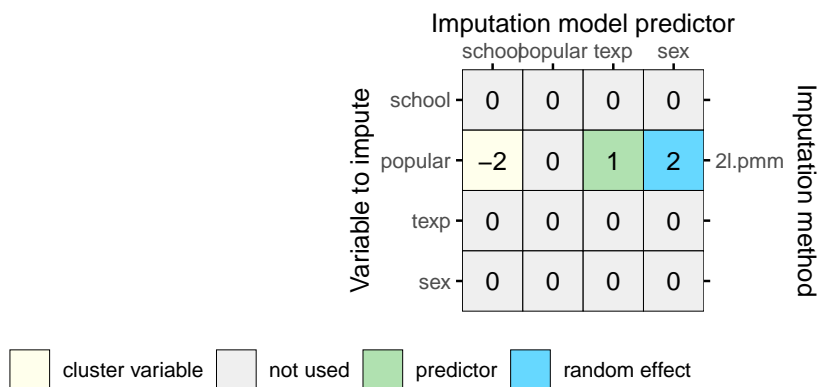
```

Visualize the imputation methods and predictors.

```

plot_pred(pred, method = meth)

```

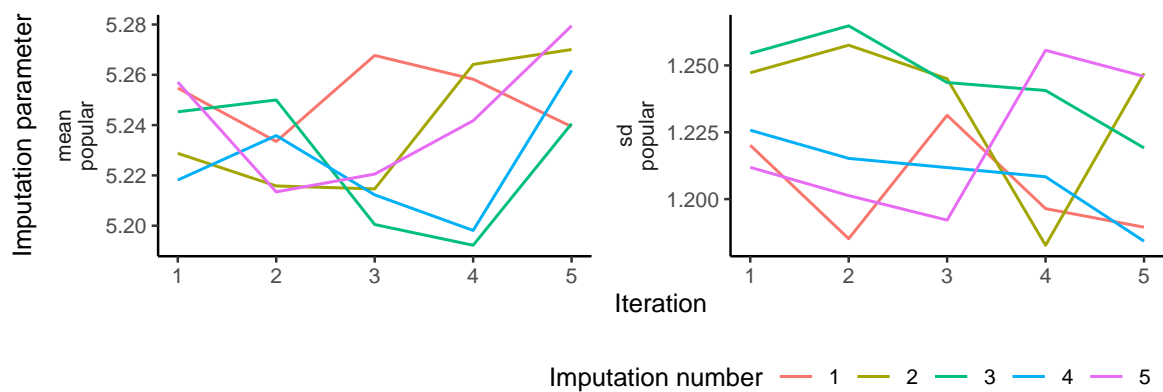


Impute the data.

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

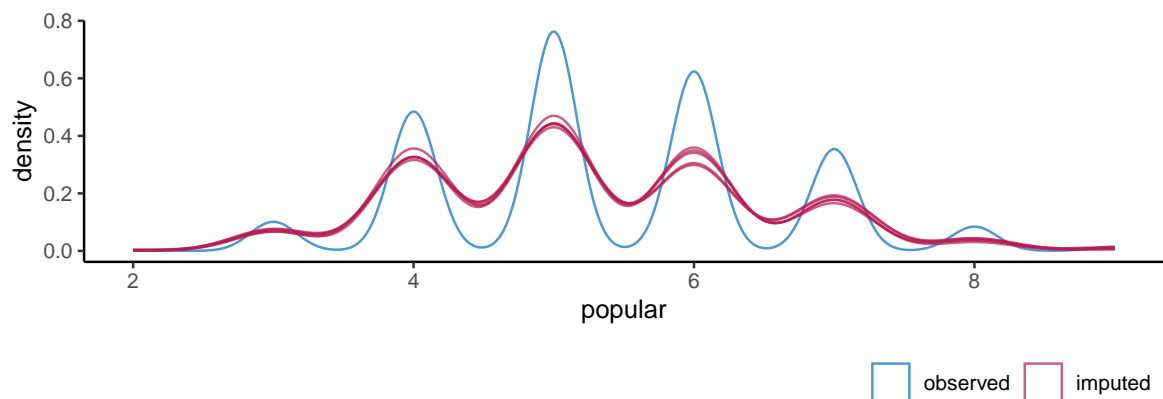
Evaluate the convergence.

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

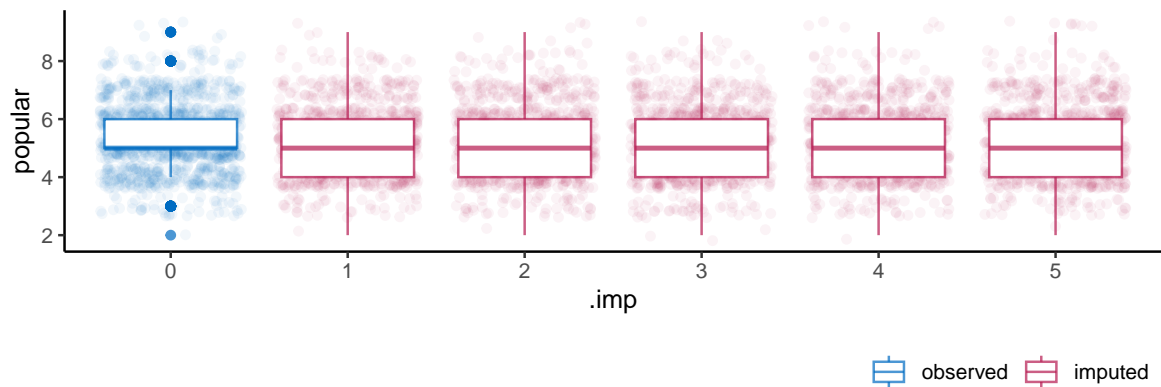


Troubleshoot non-convergence of imputation model. Evaluate the distribution of imputed values.

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

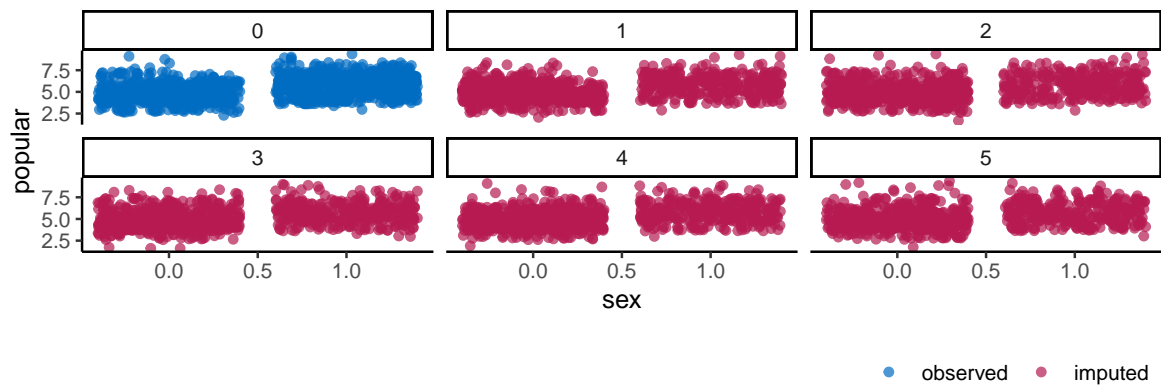


Evaluate the distribution of imputed values.



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

```
R> ggmmice(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, ?).

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.

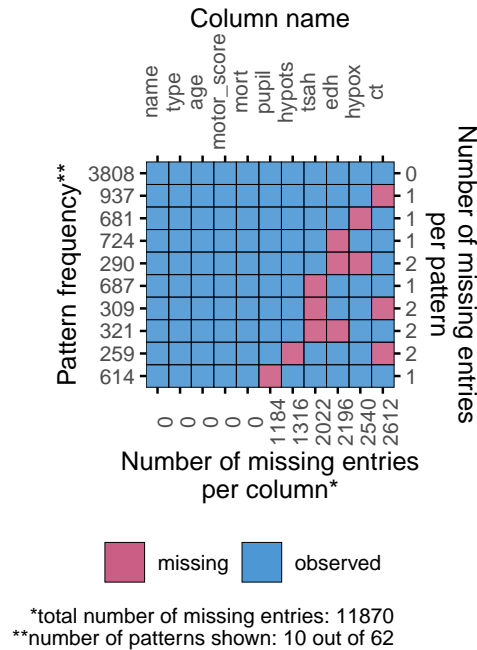
The **impact** data included in the **metamisc** package is a complete data set. The original data has already been imputed once (Steyerberg et al, 2008). For the purpose of this tutorial we have induced missingness (mimicking the missing data in the original data set before imputation). The resulting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

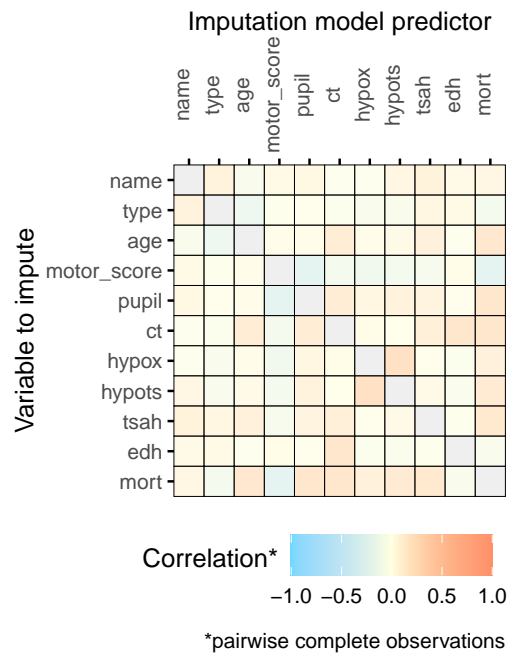
```
R> plot_corr(dat, rotate = TRUE)
```

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 21 methods.

```
R> # meth[c("pupil", "ct")] <- "21.pmm"
R> # meth[c("hypots", "tsah", "edh")] <- "21.pmm"
R> meth[meth == "pmm"] <- "21.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

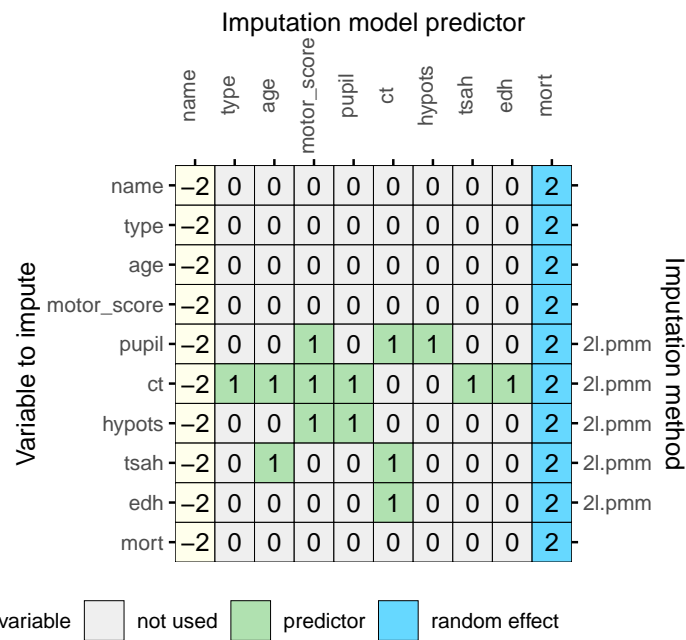
```
R> pred[, "name"] <- -2
```

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

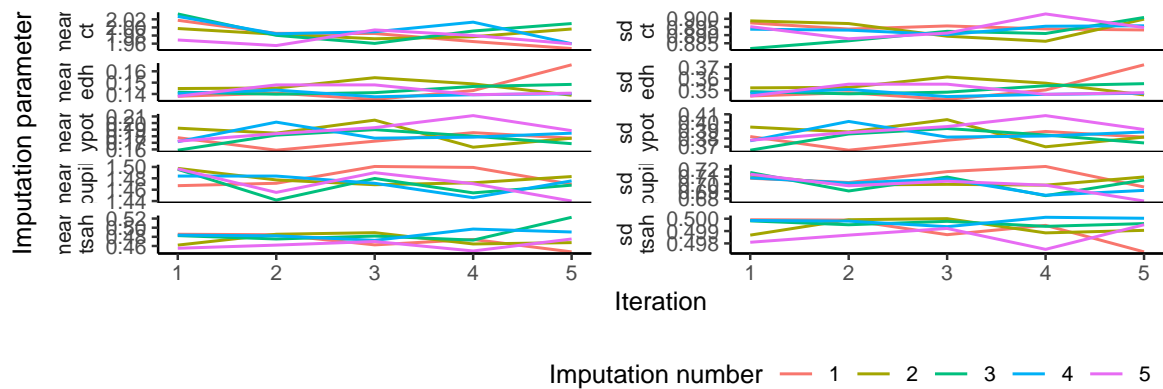
```
R> plot_pred(pred, method = meth, rotate = TRUE)
```



Impute the incomplete data with

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

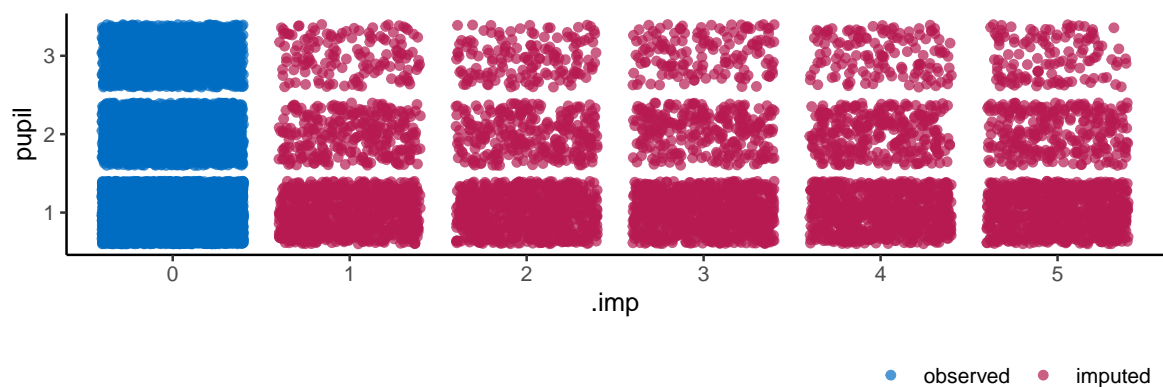
Evaluate the convergence of the algorithm



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

Evaluate the imputed values.

```
R> ggmmice(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)	-1.892	0.333	-5.682	11445.455	0.000	0.019	0.019
type	-0.346	0.178	-1.948	686356.768	0.051	0.002	0.002
age	0.032	0.002	19.564	2418.910	0.000	0.042	0.041
motor_score2	-0.575	0.070	-8.226	71295.018	0.000	0.008	0.008
motor_score3	-0.891	0.072	-12.440	66712.189	0.000	0.008	0.008
motor_score4	-1.276	0.074	-17.305	18426.087	0.000	0.015	0.015
pupil2	1.315	0.068	19.339	110.826	0.000	0.235	0.204
pupil3	0.657	0.075	8.745	423.463	0.000	0.108	0.101
ct2	0.747	0.084	8.884	41.781	0.000	0.448	0.340
ct3	0.766	0.090	8.535	12.103	0.000	1.352	0.631

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

	Estimate
Intercept~~Intercept name	0.082

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 simulate data for 5 clusters so that the true values are known. 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:

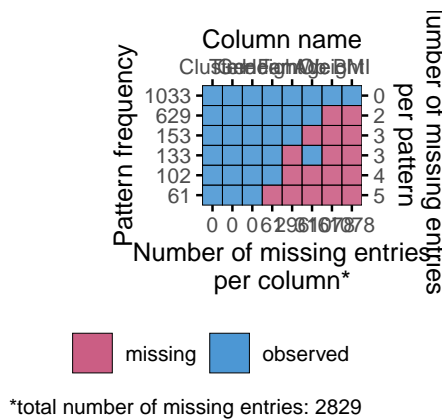
$$BMI_{ij} = (\beta_o + b_{oj}) + (\beta_1 + b_{oj}) * Age_{ij} + \beta_2 * FamOb_{ij} + \beta_3 Gender_{ij} + \epsilon_{ij} \quad (2)$$

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 an imputation method that accounts for clustering we assess the Intraclass Correlation


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

Since the ICC is above 0.1 and as the main analysis will be use a mixed model, 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 FAV 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"
```

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 supprime 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[, "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
  )
```

	Imputation model predictor						
	Cluster	Age	FamOb	Gender	Weight	BMI	
Variable to impute	Cluster	-2	0	0	0	0	0
	Time	-2	0	0	0	0	0
	Gender	-2	0	0	0	0	0
	Height	-2	0	2	0	0	2
	FamOb	-2	0	2	0	0	0
	Age	-2	0	2	0	0	0
	Weight	-2	0	2	2	0	0
	BMI	-2	0	2	0	2	0

cluster variable
 not used
 random effect

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

```

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 4.428   69.890   82.800   82.835   95.491  145.885

```

We are also contemplating the utilisation of the predictive mean matching (pmm) option, as the values imputed using a fully parametric method may be implausibly low for some patients.

```

meth_mar["Weight"] <- "2l.2stage.pmm"
imp_mar_pmm <- mice(
  data = Obesity,
  method = meth_mar,
  predictorMatrix = pred_mar,
  printFlag = FALSE
)

```

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

```

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 28.35   71.04   83.65   83.87   95.66  134.61

```

```
plot_trace(imp_mar_pmm, "Weight")
```

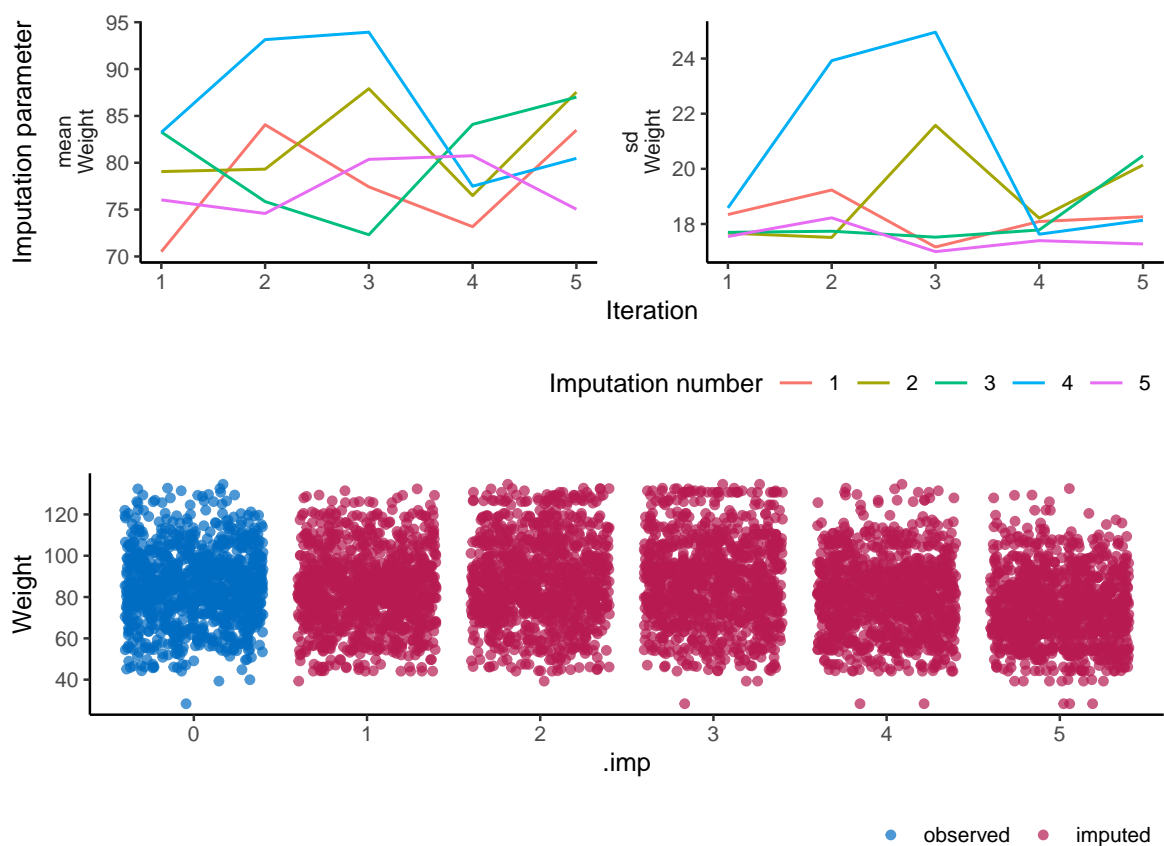
```
ggmice(imp_mar_pmm, aes(.imp, Weight)) +
  geom_jitter()
```

Analyze the imputed data.

```

R> # fit <- with(
R> #   imp_mar_pmm,
R> #   lme(BMI ~ Age + FamOb + Gender, random = ~ 1 + Age | Cluster)
R> # )
R> fit <- with(
+   imp_mar_pmm,
+   lmer(BMI ~ Age + FamOb + Gender + (1 + Age | Cluster))
+ )
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)	28.445	2.262	12.578	158.191	0.000	0.189	0.169
Age	0.041	0.032	1.312	383.212	0.190	0.114	0.107
FamObyes	0.000	0.316	-0.001	207.642	0.999	0.161	0.147
GenderMale	-1.361	0.411	-3.310	7.985	0.011	2.422	0.761

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

	Estimate
Intercept~~Intercept Cluster	19.862
Age~~Age Cluster	0.002
Intercept~~Age Cluster	-0.067
Residual~~Residual	26.010
ICC Cluster	0.395

After confirming convergence, we proceed to save the results for future use. 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, two approaches have been proposed for dealing with Missing Not at Random (MNAR) data: pattern-mixed models and selection models. Within pattern-mixed models, methods like the delta method and more advanced ones like NARFS have been suggested. The selection model approach includes methods such as the Heckman model, which can be particularly useful in this case. Several methods, including those by ?, and the recently a Heckman method designed for two-level data, allow for variations in intercepts and exposure effects (random intercept and slope) ?.

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 because 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. 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 ?.

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)
```

Imputation model predictor

	Cluster	Time	Gender	FamOb	Age	Weight	BMI
Variable to impute							
Cluster	-2	0	0	0	0	0	0
Time	-2	0	0	0	0	0	0
Gender	-2	0	0	0	0	0	0
Height	-2	0	2	0	0	0	2
FamOb	-2	0	2	0	0	2	0
Age	-2	0	2	0	0	0	0
Weight	-2	-3	2	2	0	0	0
BMI	-2	0	2	2	0	0	2

-restriction variable cluster variable not used random effect

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

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

Then we proceed to run the imputation model as before, after executing these imputation procedures, it is essential to assess convergence and the coherence of the imputed values.

```
imp_mnar <- mice(
  data = Obesity,
  method = meth_mnar,
  predictorMatrix = pred_mnar,
  printFlag = FALSE
)
summary(complete(imp_mnar, "long")$Weight)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
16.26	70.95	85.21	86.14	100.14	190.95

Upon examining the weight variable, we noticed that the imputed range falls outside the realm of plausible values (as weight should be positive).

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

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
16.26	70.95	85.21	86.14	100.14	190.95

Consequently, 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 imputation model but this time using the option of pmm, to assure that weight values are in the range of the observable data, this can be implemented by setting the pmm parameter to true.

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

We check the convergence of the results

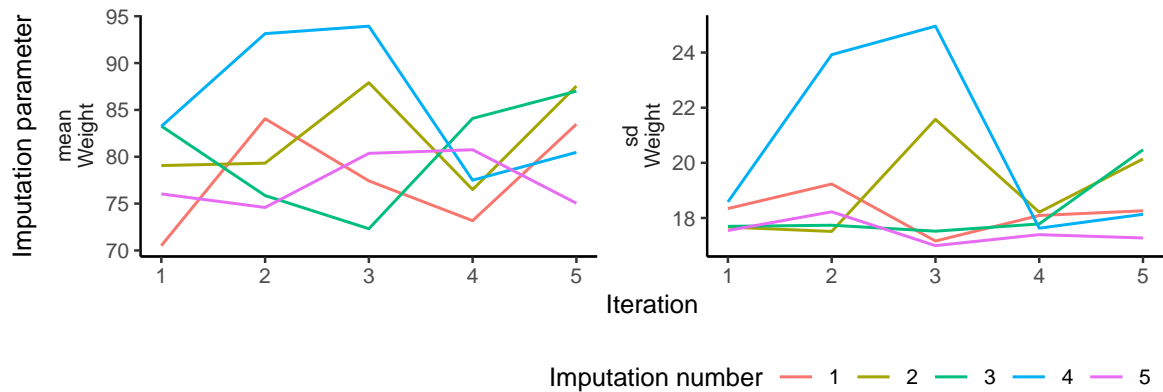
```
summary(complete(imp_mnar_pmm, "long")$Weight)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
28.35	71.51	85.07	85.35	98.90	134.61

```
plot_trace(imp_mar_pmm, "Weight")
```

After this modification 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).

```
# cc_rs <- with(setDT(Obesity)[complete.cases(Obesity), ],
#               lme(BMI ~ Age + FamOb + Gender, random = ~ 1 + Age |
#                   Cluster))
mar_rs <- with(
  imp_mar,
  lme(BMI ~ Age + FamOb + Gender, random = ~ 1 + Age | Cluster)
)
mar_pmm_rs <- with(
```



```

    imp_mar_pmm,
    lme(BMI ~ Age + FamOb + Gender, random = ~ 1 + Age | Cluster)
  )
mnar_rs <- with(
  imp_mnar,
  lme(BMI ~ Age + FamOb + Gender, random = ~ 1 + Age | Cluster)
)
mnar_pmm_rs <- with(
  imp_mnar_pmm,
  lme(BMI ~ Age + FamOb + Gender, random = ~ 1 + Age | Cluster)
)
# list_models <- list(cc_rs, mar_rs, mar_pmm_rs, mnar_rs, mnar_pmm_rs)
# plot_models(list_models,
#             mod_name = c("Complete case", "MAR", "MAR_pmm", "MNAR", "MNAR_pmm"))

est <- testEstimates(as.mitml.result(mnar_pmm_rs), extra.pars = TRUE)

```

Display results in table.

```

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

```

	Estimate	Std.Error	t.value	df	P(> t)	RIV	FMI
(Intercept)	28.880	2.391	12.078	56.876	0.000	0.361	0.290
Age	0.050	0.037	1.343	19.874	0.194	0.814	0.497
FamObyes	-0.027	0.490	-0.056	11.685	0.957	1.410	0.642
GenderMale	-1.484	0.804	-1.846	4.813	0.126	10.318	0.934

```

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

```

	Estimate
Intercept~~Intercept Cluster	19.111
Age~~Age Cluster	0.001
Intercept~~Age Cluster	-0.046
Residual~~Residual	30.071
ICC Cluster	0.372

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 estimates, we notice that in general MNAR imputation 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 significant test and therefore lead to contradictory 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. Funding

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

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.

7. 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/>].

Acknowledgments

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

References

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

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.

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 Centre for Health Sciences and Primary Care
Universiteitsweg 100
Utrecht The Netherlands

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

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

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