Journal of Statistical Software
MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

Imputation of Incomplete Multilevel Data with mice

Hanne Oberman
Utrecht University

Johanna Munoz Avila
University Medical Center Utrecht

Valentijn de Jong
University Medical Center Utrecht

Gerko Vink
Utrecht University

Thomas Debray
University Medical Center Utrecht

Abstract

This is a tutorial paper on imputing incomplete multilevel data with **mice**. Footnotes in the current version show work in progress/under construction. The last section is not part of the manuscript, but purely for reminders. We aim to submit at JSS, so there is no word count limit ("There is no page limit, nor a limit on the number of figures or tables"). [Just adding some text to get a better guess of what the actura abstract will look like: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.]

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

1. Introduction

In many contemporary data analysis efforts, some form of hierarchical or clustered data structures are recorded. In the simplest case, such a structure entails the nesting of units within clusters (e.g., students within school classes). More complex clustered structures may occur when there are multiple hierarchical levels (e.g., patients within hospitals within regions or countries), or when the clustering is non-nested (e.g., electronic health record data from

Table 1: Concepts in multilevel methods

Concept	**Details**
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.
Random effect	Multilevel models typically accommodate for variability by including a separate group mean for each cluster. 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]. [TODO: add stratification.]

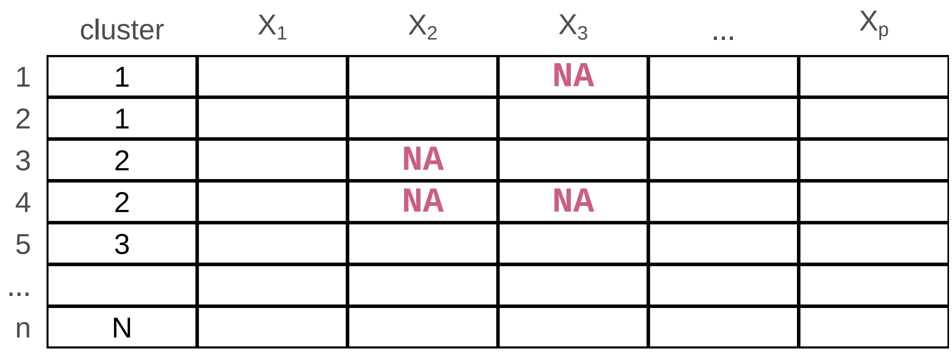


Figure 1: Missingness in multilevel data

diverse settings and populations within large databases). The clustered structure of multilevel data should be taken into account when developing analysis models: 1) for the simple reason that groups of observations share some common variance, and 2) because ignoring multilevel structures can be harmful to the statistical inferences and introduce bias in estimators (Hox, Moerbeek, and van de Schoot 2017). There are many names for models that take clustering into account. Some popular examples are 'multilevel models', 'hierarchical models', 'mixed effect models' and 'random effect models'. Table 1 provides an overview of some key concepts in multilevel modeling.

1.1. Missingness in multilevel data

The process of analyzing multilevel data is further complicated when not all data entries are observed. Just as with single level data, missingness may occur at the unit level. But with multiple levels of data comes the potential for clustered missingness. Therefore, incomplete multilevel data can be categorized into two general patterns: systematic missingness and sporadic missingness (Resche-Rigon, White, Bartlett, Peters, and Thompson 2013). Systematic missingness implies that one or more variables are never observed in a certain cluster. With sporadic missingness there may be observed data for some but not all units in a cluster (Van Buuren 2018; Jolani 2018). We have visualized this difference in Figure 1, which shows an $n \times p$ set $\mathbf{X} = X_1, \ldots, X_p$, with n units distributed over N clusters and p variables.

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
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 [@rubi76; @meng94]. [TODO: add congeniality, but maybe in-text?]

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.

A key characteristic of the missing data to take into account in analyses is the mechanism behind the missingness. We distinguish between MCAR, MAR and MNAR in theory (see Table 2), but in practice this distinction is less clear. Since the essence of the true non-response mechanism may not be known, it is generally inferred or assumed to be ignorable (i.e., MCAR or MAR). [TODO: add that this assumption may not always be valid, especially with modern types of big data sources.] Depending on the actual missingness-generating mechanism, missing data handling strategies may be more or less suitable, see e.g., Yucel (2008) and Hox, van Buuren, and Jolani (2015).

Since excluding observations is not a desirable workflow, the missingness in multilevel data should be accommodated before or within the analysis of scientific interest. In this paper, we focus on the former approach: imputing (i.e., filling in) the missing data with plausible values, whereafter the completed data may be analyzed as if it were completely observed. Imputation separates the missing data problem from the scientific problem, which makes the missing data strategy very generic and popular. If each missing value is replaced multiple times, the resulting inferences may validly convey the uncertainty due to missingness (c.f. Rubin 1976). The R package **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). In this paper, we will discuss how to use **mice** in the context of multilevel data.

[TODO: clarify why clustering is relevant during imputation, and why this exposes the need

Table 3: Notation

Concept	Details
	[TODO: explain **lme4** notation here]

for specialized imputation methods and more attention during their implementation ("thou shall not simply run `mice()` on any incomplete dataset").] [TODO: Add that the more the random effects are of interest, the more you need multilevel imputation models.] [TODO: Add an overview of all possible predictor matrix values in manuscript or `ggmice` legend.]

1.2. Aim of this paper

This papers serves as a tutorial for imputing incomplete multilevel data with **mice** in R. 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 (e.g., **jomo** and **mdmb**) are outside the scope of this tutorial. Assumed knowledge includes basic familiarity with multilevel imputation (see e.g. Audigier, White, Jolani, Debray, Quartagno, Carpenter, van Buuren, and Resche-Rigon 2018, and Grund, Lüdtke, and Robitzsch (2018)) and the **lme4** notation for multilevel models (see Table 3).

We illustrate how to impute incomplete multilevel data by means of three case studies:

- `popmis` from the **mice** package (simulated data on perceived popularity, $n = 2,000$ pupils across $N = 100$ schools, van Buuren and Groothuis-Oudshoorn 2021);
- `hiv` from the **GJRM** package (simulated data on HIV diagnoses, $n = 6,416$ patients across $N = 9$ regions, Radice 2021);
- `impact` from the **metamisc** package (empirical data on traumatic brain injuries, $n = 11,022$ patients across $N = 15$ studies, Debray and de Jong 2021).

[TODO: add where novice vs more experienced readers should start reading.] For each of these datasets, we will 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. With the `popmis` data, we show how (and how not) to develop an imputation model. With the `hiv` data we focus on extending the imputation model to include Heckman-type selection-inclusion methods. With the `impact` data we provide an example of multivariate missingness in real-world data. Together, this should give enough scaffolding for applied researchers who are faced with incomplete multilevel data.

1.3. Setup

[TODO: Add environment info, seed and version number(s) somewhere.] Set up the R environment and load the necessary packages:

```
R> set.seed(2022)
R> library(mice)          # for imputation
R> library(ggmice)        # for visualization
R> library(ggplot2)       # for visualization
```

```
R> library(dplyr)        # for data wrangling
R> library(lme4)         # for multilevel modeling
R> library(mitml)        # for multilevel pooling
```

2. Case study I: popularity data

In this section we'll go over the different steps involved with imputing incomplete multilevel data. The data we're using is the `popmis` dataset from the `mice` package. This is a simulated dataset with pupils ($n = 2000$) clustered within schools ($N = 100$). In this tutorial we'll use the following variables:

- `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, random slopes and a cross-level interaction. The outcome variable is `popular`, which is predicted from the unit-level variable `sex` and the cluster-level variable `texp`:

```
R> mod <- popular ~ 1 + sex + texp + sex:texp + (1 + sex | school)
```

Load the data into the environment and select the relevant variables:

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

Plot the missing data pattern:

```
R> plot_pattern(popmis)
```

The missingness is univariate and sporadic, which is illustrated in the missing data pattern in Figure 2.

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(popmis)
```

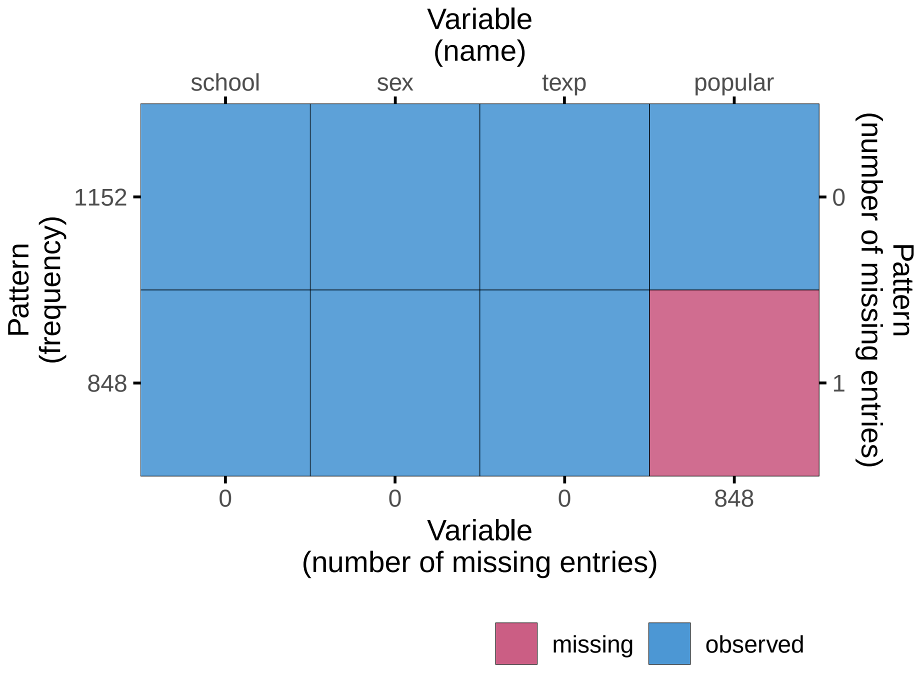
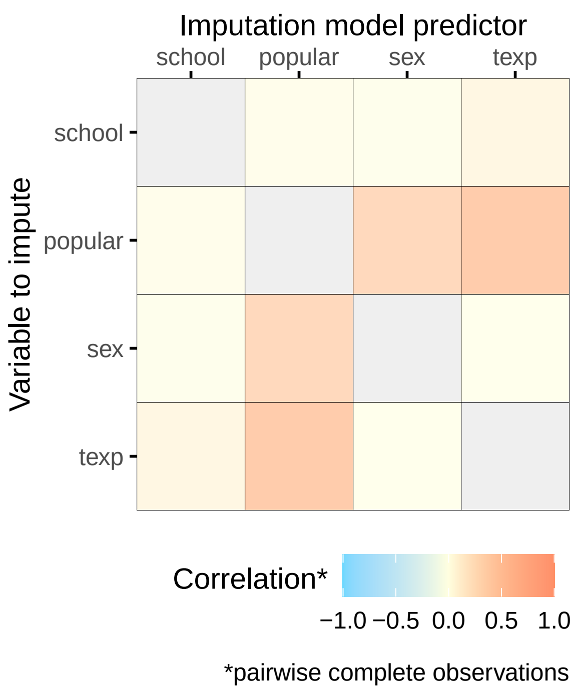


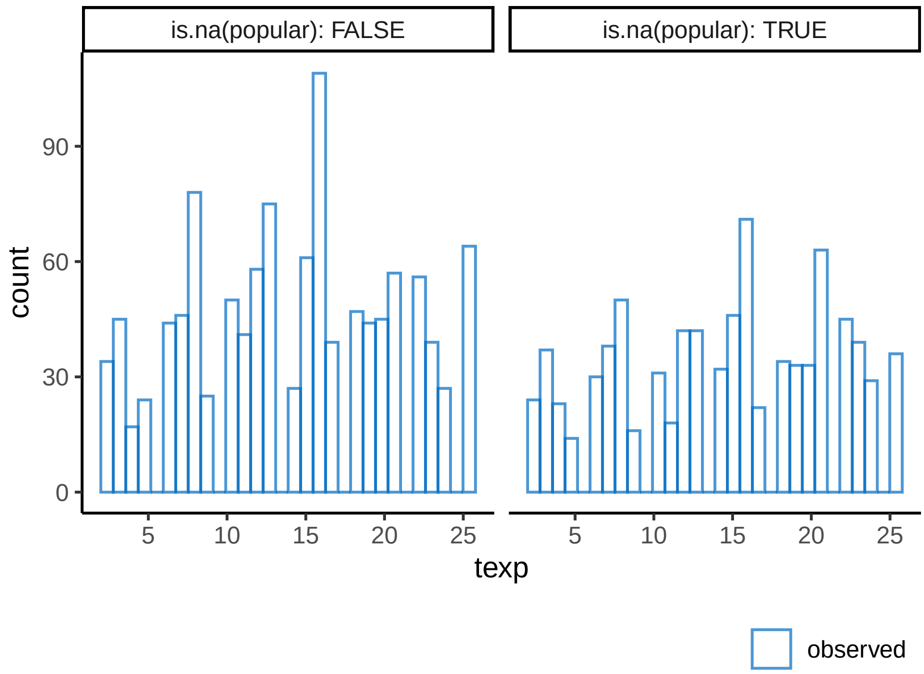
Figure 2: Missing data pattern in the popularity data



This shows us that both `sex` and `texp` may be useful imputation model predictors. Moreover, the missingness in `popular` may depend on the observed values of other variables. We'll highlight one other variable to illustrate, but ideally one would inspect all relations. The questions we'll ask are: 'Does the missing data of pupil popularity (`popular`) depend on observed teacher popularity (`texp`)?'. This can be evaluated statistically, but visual inspection usually suffices. We'll make a histogram of `texp` separately for the pupils with known popularity and missing popularity.

Plot the histogram for teacher experience conditional on the missingness indicator of `popular`:

```
R> ggmice(popmis, aes(texp)) +
+   geom_histogram(fill = "white") +
+   facet_grid(. ~ is.na(popular), scales = "free", labeller = label_both)
```



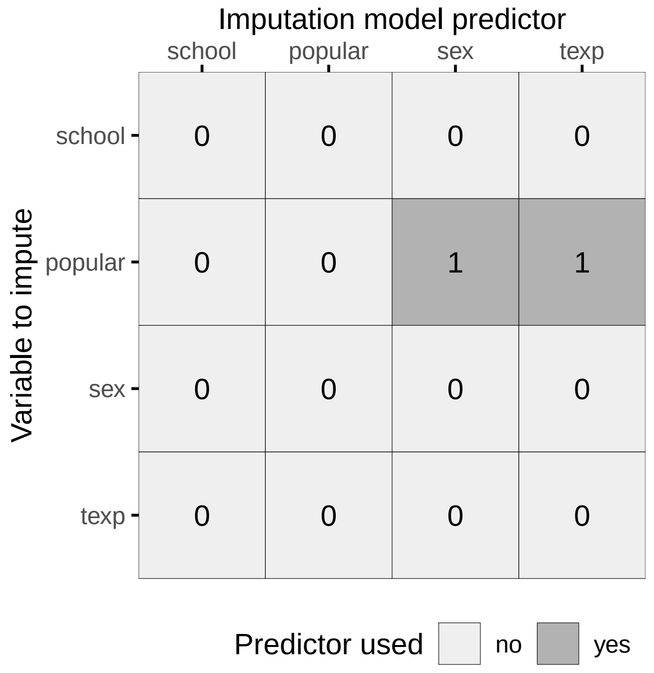
This shows us that there are no apparent differences in the distribution of `texp` depending on the missingness indicator of `popular` (t = -0.873, p = 0.383). [TODO: think about what is a meaningful rule of thumb to signal that the user should be worried?]

Imputation ignoring the cluster variable (not recommended)

The first imputation model that we'll use is likely to be invalid. We do <u>not</u> use the cluster identifier `school` as imputation model predictor. With this model, we ignore the multilevel structure of the data, despite the high ICC. This assumes exchangeability between units. We include it purely to illustrate the effects of ignoring the clustering in our imputation effort.

Create a methods vector and predictor matrix for `popular`, and make sure `school` is not included as predictor:

```
R> meth <- make.method(popmis) # methods vector
R> pred <- quickpred(popmis)   # predictor matrix
R> plot_pred(pred)
```



Impute the data, ignoring the cluster structure:

```
R> imp <- mice(popmis, pred = pred, print = FALSE)
```

Analyze the imputations:

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

Print the estimates:

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


Call:

testEstimates(model = as.mitml.result(fit), extra.pars = TRUE)

Final parameter estimates and inferences obtained from 5 imputed data sets.

             Estimate Std.Error   t.value       df   P(>|t|)       RIV       FMI
(Intercept)     3.445     0.153    22.473   20.745     0.000     0.783     0.486
sex             1.175     0.205     5.728   13.364     0.000     1.208     0.602
texp            0.103     0.010     9.991   15.982     0.000     1.001     0.553
sex:texp       -0.020     0.014    -1.460   11.610     0.171     1.421     0.644
```

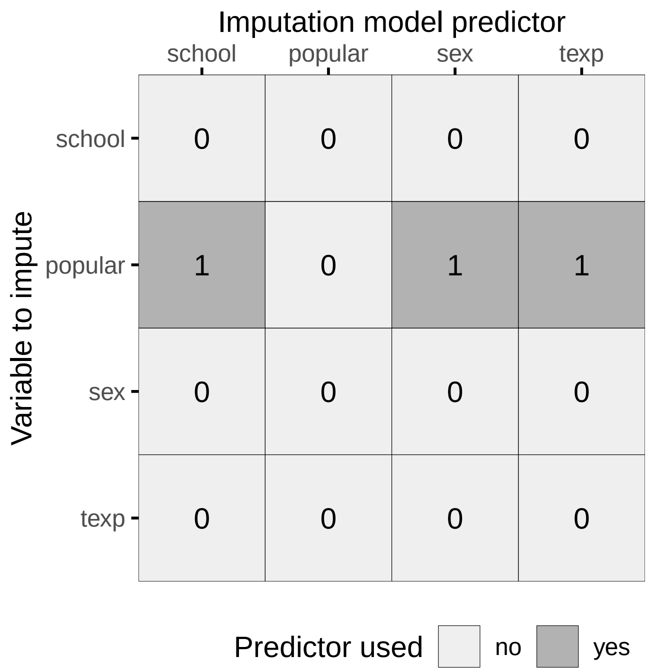
```
                              Estimate
Intercept~~Intercept|school      0.158
sex~~sex|school                  0.186
Intercept~~sex|school           -0.025
Residual~~Residual               0.704
ICC|school                       0.183

Unadjusted hypothesis test as appropriate in larger samples.
```

Imputation with the cluster variable as predictor (not recommended)

We'll now use `school` as a predictor to impute all other variables. This is still not recommended practice, since it only works under certain circumstances and results may be biased (Drechsler 2015; Enders, Mistler, and Keller 2016). But at least, it includes some multilevel aspect. This method is also called 'fixed cluster imputation', and uses N-1 indicator variables representing allocation of N clusters as a fixed factor in the model (Reiter, Raghunathan, and Kinney 2006; Enders et al. 2016). Colloquially, this is 'multilevel imputation for dummies'.

[TODO: Add that it doesn't work with systematic missingness (only with sporadic). There's some pros and cons, and it may not even differ much if the number of clusters is low.]

```
R> # adjust the predictor matrix
R> pred["popular", "school"] <- 1
R> plot_pred(pred)
```




```
R> # impute the data, cluster as predictor
R> imp <- mice(popmis, pred = pred, print = FALSE)
```

Analyze the imputations:

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

Print the estimates:

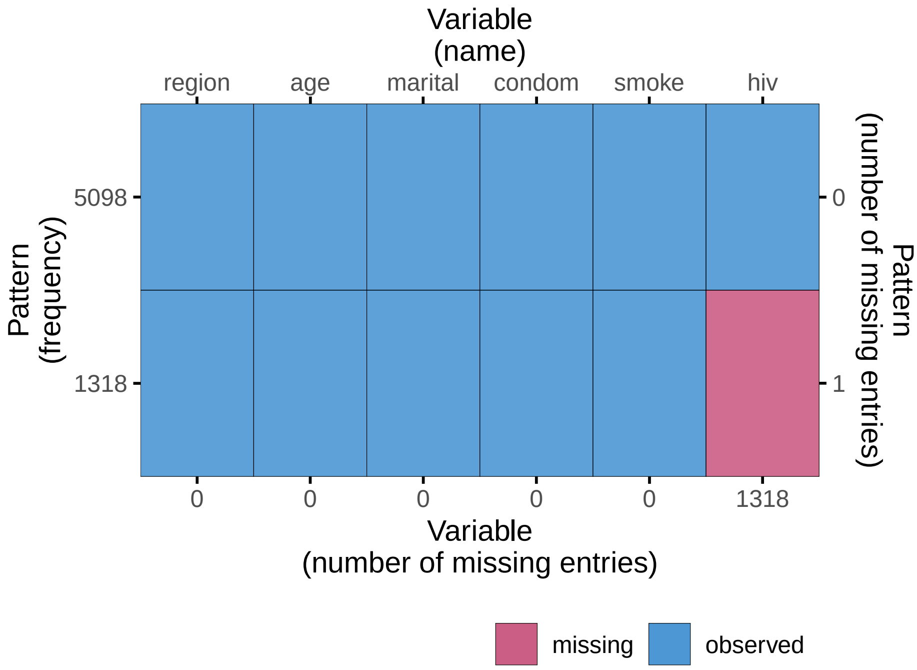
```
R> testEstimates(as.mitml.result(fit), extra.pars = TRUE)


Call:

testEstimates(model = as.mitml.result(fit), extra.pars = TRUE)

Final parameter estimates and inferences obtained from 5 imputed data sets.

             Estimate Std.Error   t.value        df   P(>|t|)       RIV       FMI
(Intercept)     3.341     0.179    18.648    14.134     0.000     1.137     0.587
sex             1.267     0.216     5.866    12.916     0.000     1.255     0.612
texp            0.108     0.012     9.061    12.109     0.000     1.351     0.631
sex:texp       -0.030     0.014    -2.058    11.313     0.063     1.467     0.651

                            Estimate
Intercept~~Intercept|school    0.198
sex~~sex|school                0.231
Intercept~~sex|school         -0.042
Residual~~Residual             0.622
ICC|school                     0.242

Unadjusted hypothesis test as appropriate in larger samples.
```

3. Case study II: HIV data

Data are simulated and included in the `GJRM` package. We will use the following variables:

- `region` Cluster variable,
- `hiv` HIV diagnosis (0=no, 1=yes),
- `age` Age of the patient,
- `marital` Marital status,
- `condom` Condom use during last intercourse,
- `smoke` Smoker (levels; inclusion restriction variable).

The imputation of these date is based on the toy example from IPDMA Heckman Github repo.



4. Case study III: IMPACT data

We illustrate how to impute incomplete multilevel data by means of a case study: `impact` from the **metamisc** package (empirical data on traumatic brain injuries, $n = 11,022$ units across $N = 15$ clusters, Debray and de Jong 2021). [TODO: add more info about the complete data.] 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).

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 unfortunate outcomes. The estimand is the adjusted odds ratio for `ct`, after including `type`, `age` `motor_score` and `pupil` into the analysis model:

```
R> mod <- mort ~ 1 + type + age + motor_score + pupil + ct + (1 | name)
```

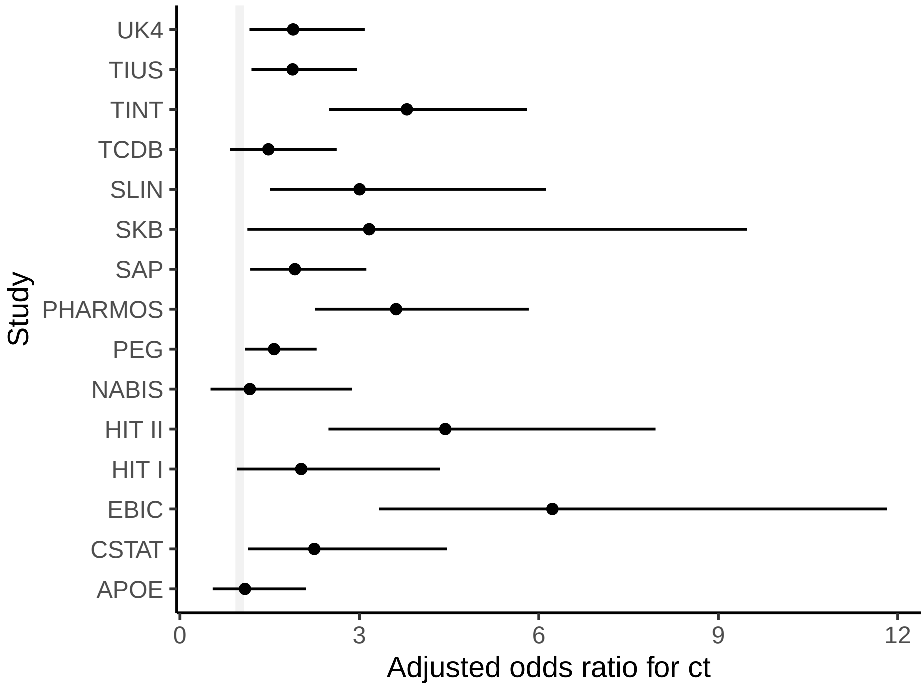
Note that variables `hypots`, `hypox`, `tsah` and `edh` are not part of the analysis model, and may thus serve as auxiliary variables for imputation.

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 complete and incomplete data into the R workspace:

```
R> data("impact", package = "metamisc")      # complete data
R> dat <- read.table("link/to/the/data.txt") # incomplete data
```

The estimated effects in the complete data are visualized in Figure **??**.




```
R> # fit <- glmer(mod, family = "binomial", data = impact) # fit the model
R> # tidy(fit, conf.int = TRUE, exponentiate = TRUE)       # print estimates
```

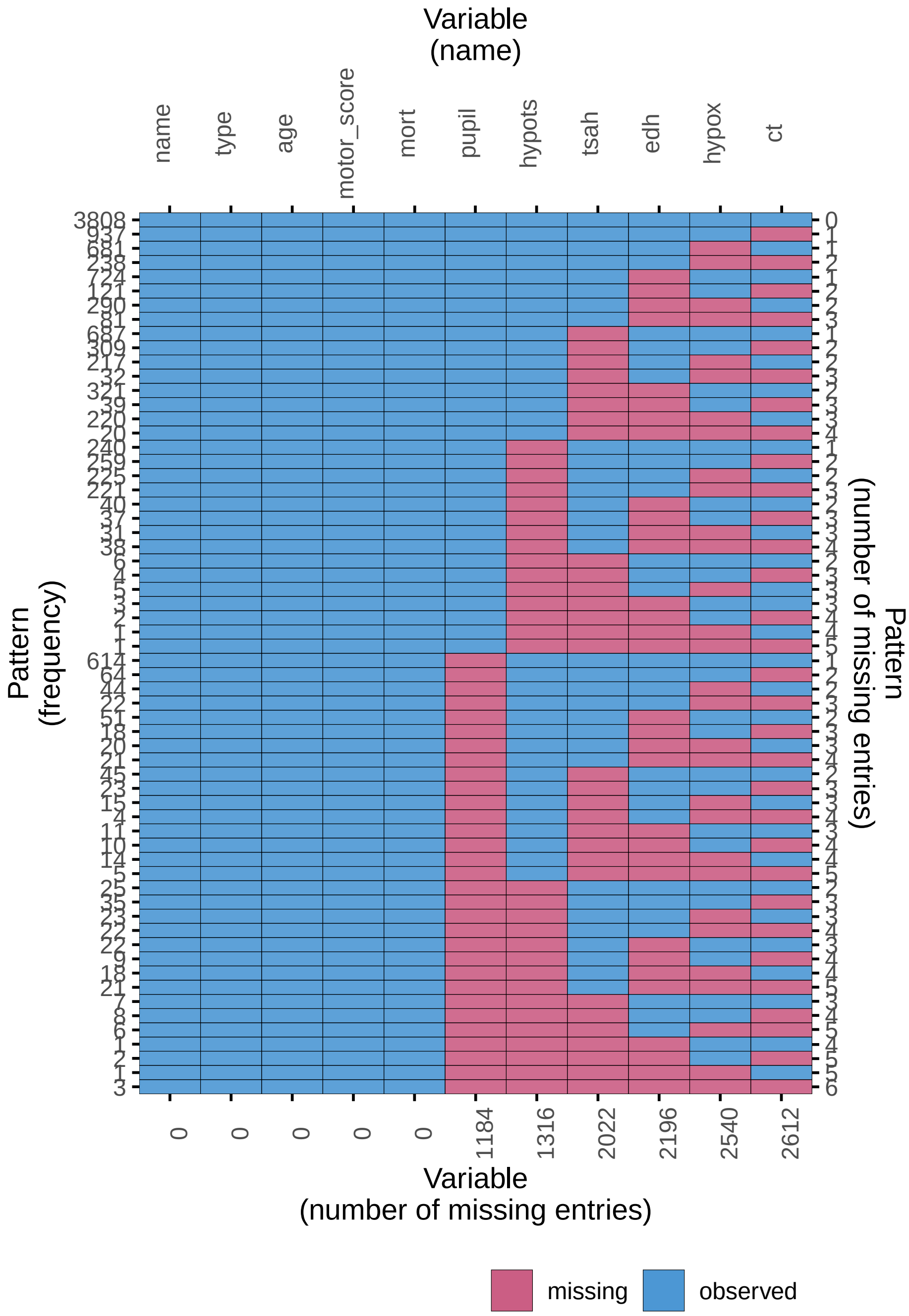
[TODO: add ICC before/after imputation and interpret: This tells us that the multilevel structure of the data should probably be taken into account. If we don't, we'll may end up with incorrect imputations, biasing the effect of the clusters towards zero.]

[TODO: add descriptive statistics of the complete and incomplete data.]

4.1. Missingness

To explore the missingness, it is wise to look at the missing data pattern:

```
R> plot_pattern(dat, rotate = TRUE)  # plot missingness pattern
```

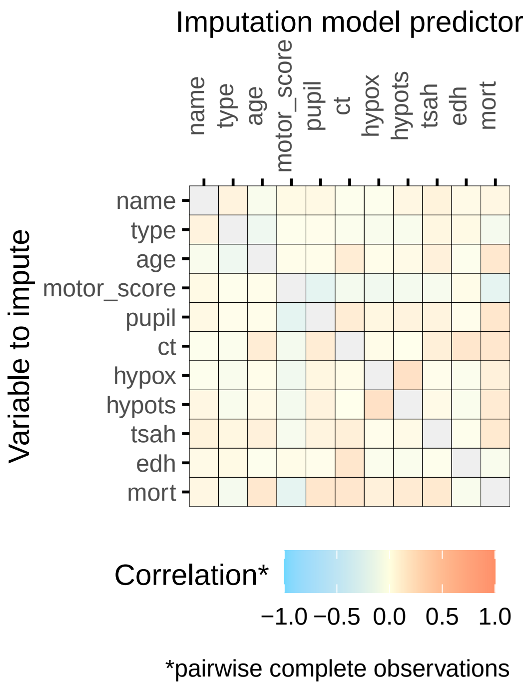


This shows... [TODO: fill in that we need to impute `ct` and `pupil`.]

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... [TODO: fill in that 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) # plot correlations
```



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)  # remove variable
```

4.2. Complete case analysis

As previously stated, complete case analysis lowers statistical power and may bias results. The complete case analysis estimates are:

```
R> fit <- glmer(mod, family = "binomial", data = na.omit(dat)) # fit the model
R> tidy(fit, conf.int = TRUE, exponentiate = TRUE)             # print estimates
```

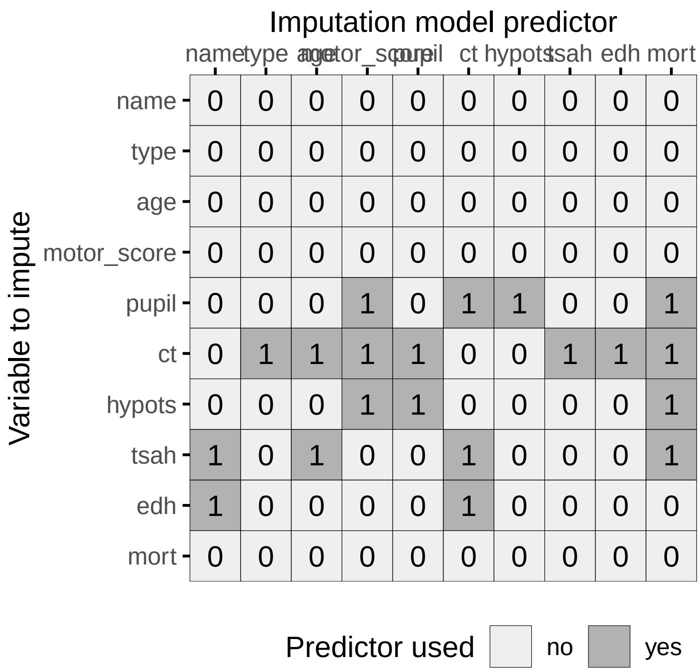
```
# A tibble: 11 x 9
   effect  group term  estimate std.error statistic   p.value conf.low conf.high
   <chr>   <chr> <chr>    <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
 1 fixed   <NA>  (Int~   0.0863    0.0182     -11.6  2.99e-31   0.0571     0.130
 2 fixed   <NA>  type~   0.757     0.137       -1.54 1.22e- 1   0.531      1.08
 3 fixed   <NA>  age     1.03      0.00265     12.9  7.40e-38   1.03       1.04
 4 fixed   <NA>  moto~   0.651     0.0732      -3.82 1.34e- 4   0.522      0.811
 5 fixed   <NA>  moto~   0.489     0.0555      -6.30 2.97e-10   0.391      0.611
 6 fixed   <NA>  moto~   0.274     0.0321     -11.0  2.28e-28   0.218      0.345
 7 fixed   <NA>  pupi~   3.20      0.317       11.7  8.18e-32   2.63       3.88
 8 fixed   <NA>  pupi~   1.75      0.195        5.06 4.27e- 7   1.41       2.18
 9 fixed   <NA>  ctIII   2.41      0.268        7.89 3.05e-15   1.94       2.99
10 fixed   <NA>  ctIV~   2.30      0.214        8.95 3.55e-19   1.92       2.76
11 ran_pa~ name  sd__~   0.230   NA            NA    NA         NA         NA
```

As we can see... [TODO: fill in.]

4.3. Imputation model

Create a methods vector and predictor matrix, and make sure `name` is not included as predictor:

```
R> meth <- make.method(dat) # methods vector
R> pred <- quickpred(dat)   # predictor matrix
R> plot_pred(pred)
```



[TODO: mutate data to get the right data types for imputation (e.g. integer for clustering variable).]

5. Discussion

- JOMO in **mice** -> on the side for now
- Additional levels of clustering
- More complex data types: timeseries and polynomial relationship in the clustering.

6. Think about

- Adding some kind of help function to mice that suggests a suitable predictor matrix to the user, given a certain analysis model.
- Adding a `multilevel_ampute()` wrapper function in mice.
- Exporting `mids` objects to other packages like `lme4` or `coxme`?
- Adding a ICC=0 dataset to show that even if there is no clustering it doesn't hurt.
- Show use case for deductive imputation for cluster level variables?
- env dump in repo

References

Audigier V, White IR, Jolani S, Debray TPA, Quartagno M, Carpenter J, van Buuren S, Resche-Rigon M (2018). "Multiple Imputation for Multilevel Data with Continuous and Binary Variables." Statistical Science, **33**(2), 160–183. ISSN 0883-4237, 2168-8745. doi:10.1214/18-STS646. 1702.00971.

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

Drechsler J (2015). "Multiple Imputation of Multilevel Missing Data—Rigor Versus Simplicity." Journal of Educational and Behavioral Statistics, **40**(1), 69–95. ISSN 1076-9986. doi:10.3102/1076998614563393.

Enders CK, Mistler SA, Keller BT (2016). "Multilevel Multiple Imputation: A Review and Evaluation of Joint Modeling and Chained Equations Imputation." Psychological Methods, **21**(2), 222–240. ISSN 1939-1463. doi:10.1037/met0000063.

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.

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.

Hox JJ, Moerbeek M, van de Schoot R (2017). Multilevel Analysis: Techniques and Applications, Third Edition. Routledge. ISBN 978-1-317-30868-3.

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.

Radice GMaR (2021). "GJRM: Generalised Joint Regression Modelling."

Reiter JP, Raghunathan T, Kinney SK (2006). "The Importance of Modeling the Sampling Design in Multiple Imputation for Missing Data." undefined.

Resche-Rigon M, White IR, Bartlett JW, Peters SAE, Thompson SG (2013). "Multiple Imputation for Handling Systematically Missing Confounders in Meta-Analysis of Individual Participant Data." Statistics in medicine, **32**(28), 4890–4905. ISSN 1097-0258 0277-6715. doi:10.1002/sim.5894.

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

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.

Affiliation:

Hanne Oberman
Utrecht University
Padualaan 14
3584 CH Utrecht
E-mail: h.i.oberman@uu.nl
URL: https://hanneoberman.github.io/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd