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 tutorial illustrates the imputation of incomplete multilevel data with the R package **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

TODO: send complete manuscript by september 30th!

Many datasets include individuals from multiple settings, geographic regions, or even different studies. In the simplest case, individuals (e.g., students) are nested within so-called clusters (e.g., school classes). More complex clustered structures may occur when there are multiple

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.]

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 diverse settings and populations within large databases). In general, 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 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). [TODO: make a link to imputation methods, which require adequate handling and propagation of variance; we are not recommending the adoption of multilevel models for data analysis here, but rather for imputation.] Statistical methods for clustered data typically adopt hierarchical models that explicitly describe the grouping of observations. These models are also know as '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

Like any other dataset, clustered datasets are prone to missing data. 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. Afterwards, the completed data can be analyzed as if they were completely observed. In contrast to single imputation (where missing data are only replaced once), multiple imputation allows to preserve uncertainty due to missingness and is therefore recommended (c.f. Rubin 1976).

When clustered datasets are affected by missing values, we can distinguish between two types of missing data: sporadic missingness and systematic missingness (Resche-Rigon, White, Bartlett, Peters, and Thompson 2013). 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. [TODO: Provide an example for one of the case studies below.] When all observations are missing within one or more clusters, data are systematically missing. [TODO: Refer to Figure 1 and put interpretation in the figure caption.]

Imputation of missing data requires to consider the mechanism behind the missingness. Rubin

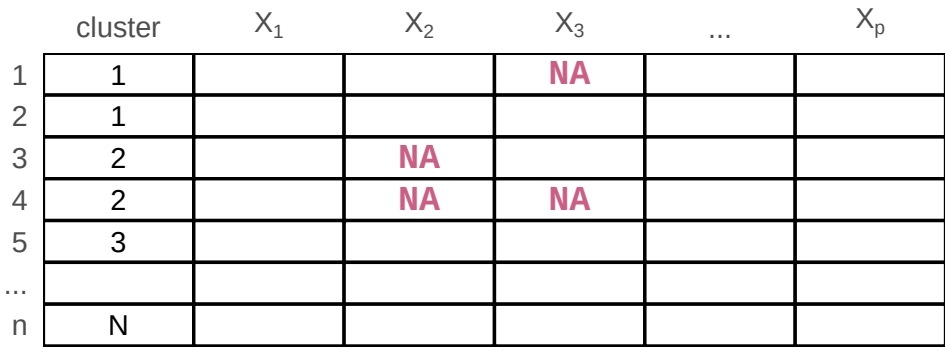


Figure 1: Missingness in multilevel data

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?]

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 2). 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 special MNAR situations.

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). However, commonly used imputation methods were not designed for use in clustered data and usually generate observations that are independent. For this reason, we discuss how the R package **mice** can be used to impute multilevel data.

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

Table 3: Notation

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

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 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);
- `hiv` from the **GJRM** package (simulated data on HIV diagnoses, $n = 6,416$ patients across $N = 9$ regions with data that are MNAR, Radice 2021).

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 imptuation. More experienced readers can skip the introduction (case study 1) and directly head to practical applications of multilevel imputation under MAR conditions (case study IMPACT) or under MNAR conditions (case study HIV).

TODO: explicit statement about not going into workings of the methods. Galimer 2l methods.

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(miceadds)      # 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

[TODO: explain case study]

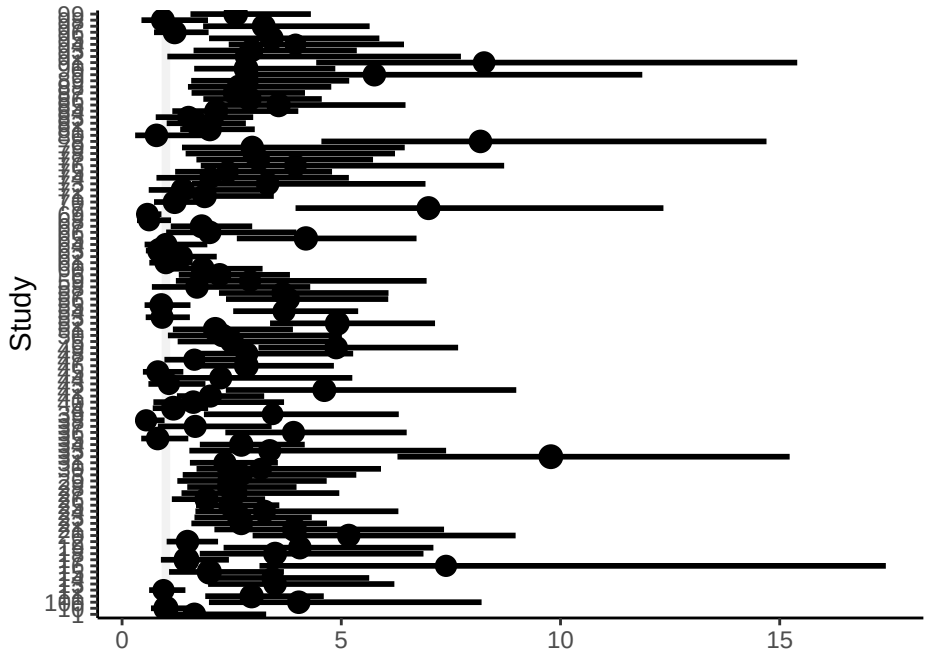
In this section we'll go over the different steps involved with imputing incomplete multilevel data with the R package mice. We consider the simulated `popmis` dataset, which included pupils ($n = 2000$) clustered within schools ($N = 100$). The following variables are of primary interest:

- `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 research objective of the popmis dataset is to predict the pupils' popularity based on their gender and the experience of the teacher. 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 + (1 | school)
```

The true effect is:



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

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

Plot the missing data pattern:

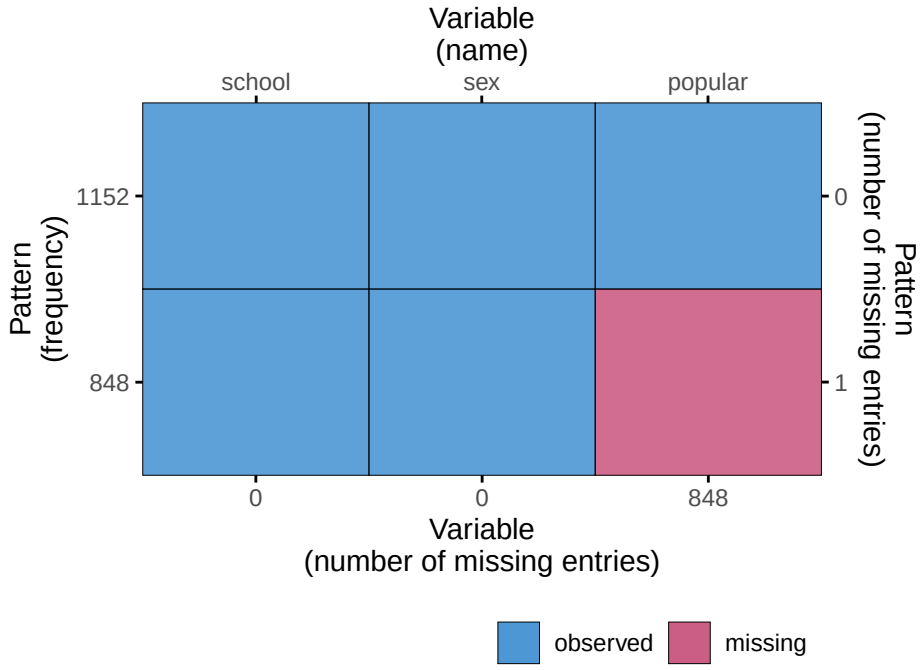


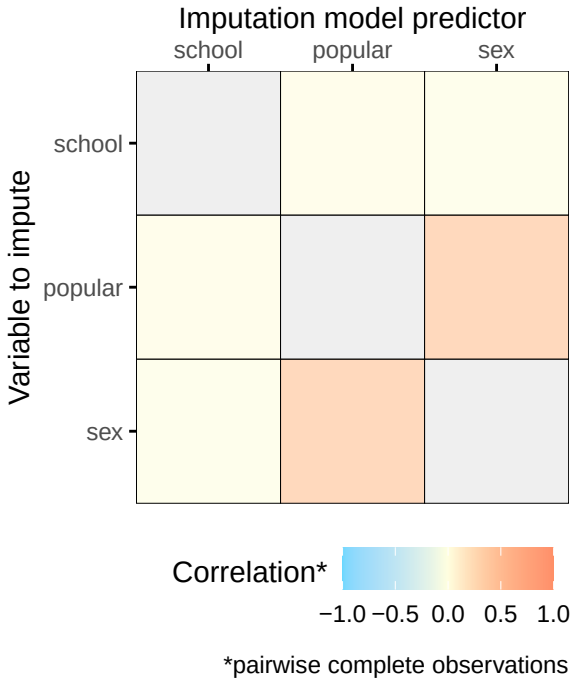
Figure 2: Missing data pattern in the popularity data

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

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

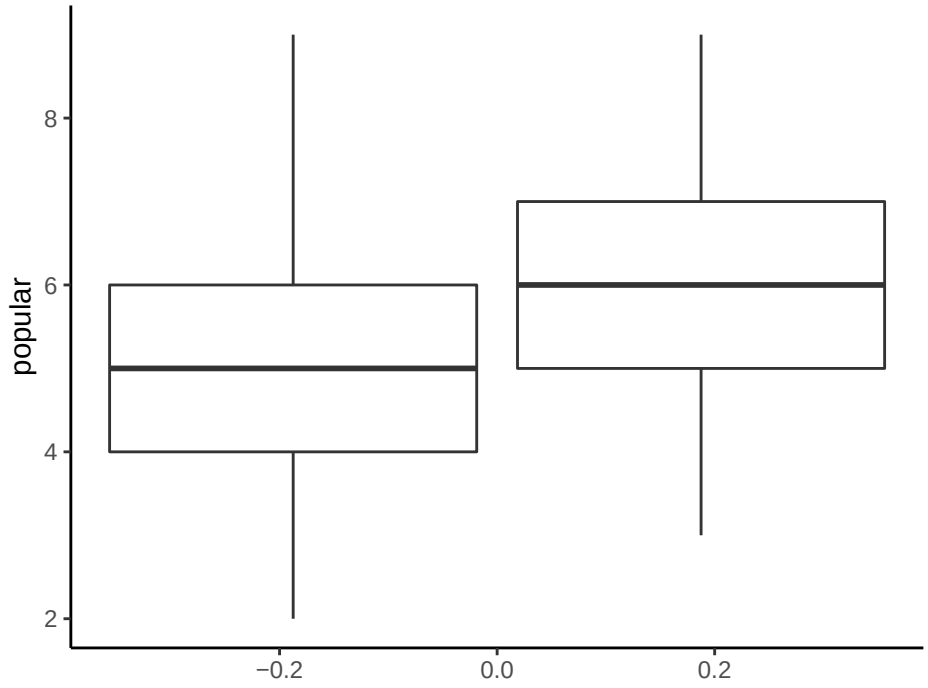
To develop the best imputation model for the incomplete variable `popular`, we need to know whether the observed values of `popular` are related to observed values of other variables. Plot the pair-wise complete correlations in the incomplete data:

```
R> plot_corr(popmis)
```



This shows us that `sex` may be a useful imputation model predictor. Moreover, the missingness in `popular` may depend on the observed values of other variables.

```
R> # ggmice(popmis, aes(sex)) +
R> #   geom_histogram(fill = "white") +
R> #   facet_grid(. ~ is.na(popular), scales = "free", labeller = label_both)
R>
R> ggplot(popmis, aes(y = popular, group = sex)) +
+   geom_boxplot() +
+   theme_classic()
```

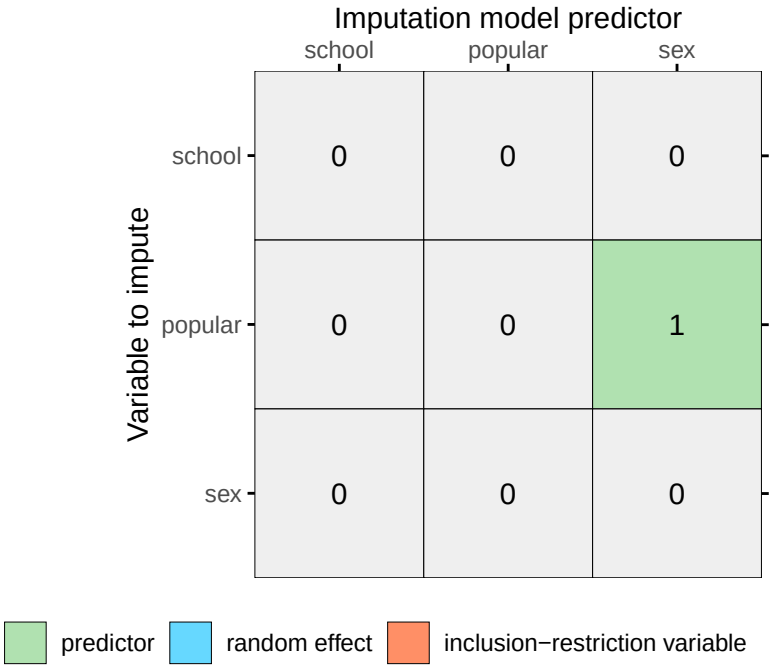


Imputation ignoring the cluster variable (not recommended)

The first imputation model that we'll use is likely to be invalid. We do not 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  + (1 | 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)     5.010     0.316    15.841     4.324     0.000    25.159     0.972
sex             0.748     0.413     1.812     4.112     0.142    72.195     0.990

                           Estimate
Intercept~~Intercept|school    0.271
```

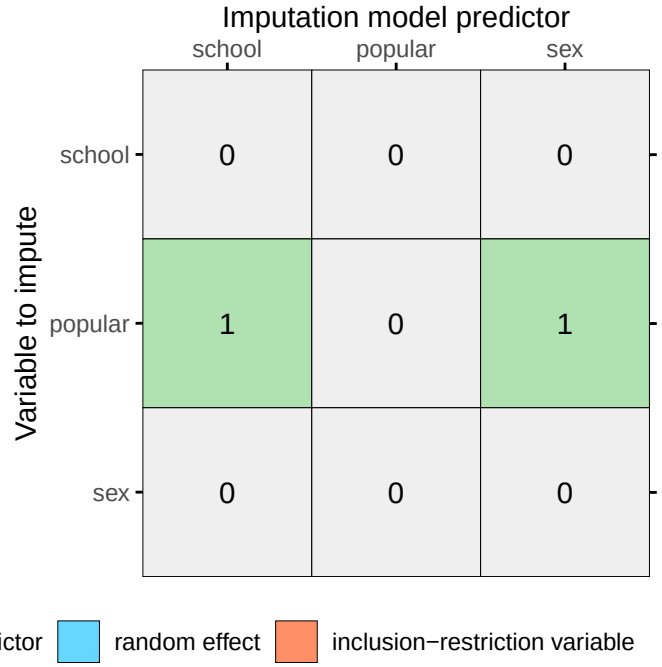
```
Residual~~Residual              1.125
ICC|school                      0.213

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 + (1 | school)))
```

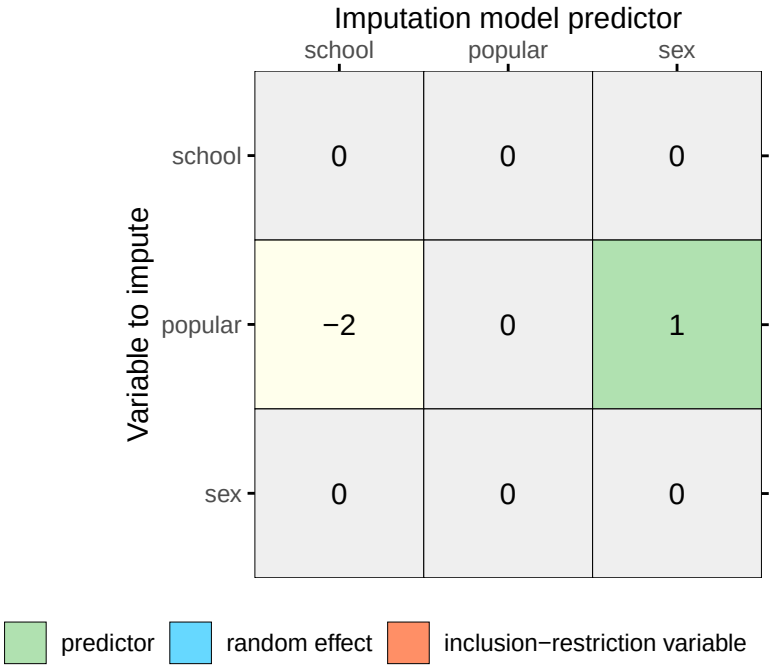
Print the estimates:

```
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)     5.043     0.293    17.187     4.469     0.000    17.558     0.961
sex             0.799     0.329     2.428     4.206     0.069    39.298     0.982

                           Estimate
Intercept~~Intercept|school    0.335
Residual~~Residual             1.296
ICC|school                     0.208

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

Imputation with multilevel model

```
R> # adjust the predictor matrix
R> pred["popular", "school"] <- -2
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 + (1 | 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)     5.051     0.259    19.541     4.645     0.000    12.876     0.947
sex             0.820     0.235     3.484     4.426     0.021    19.256     0.964

                           Estimate
Intercept~~Intercept|school    0.350
Residual~~Residual             1.318
```

```
ICC|school                    0.217

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

3. Case study II: IMPACT data (syst missingness, pred matrix)

[TODO: check if there is systematic missingness in this dataset, if not make Marshall Computerized Tomography classification (ct) systematically missing.]

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, [TODO: make this one var? also shows that you don't always need random effects everywhere?]
- `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)
```

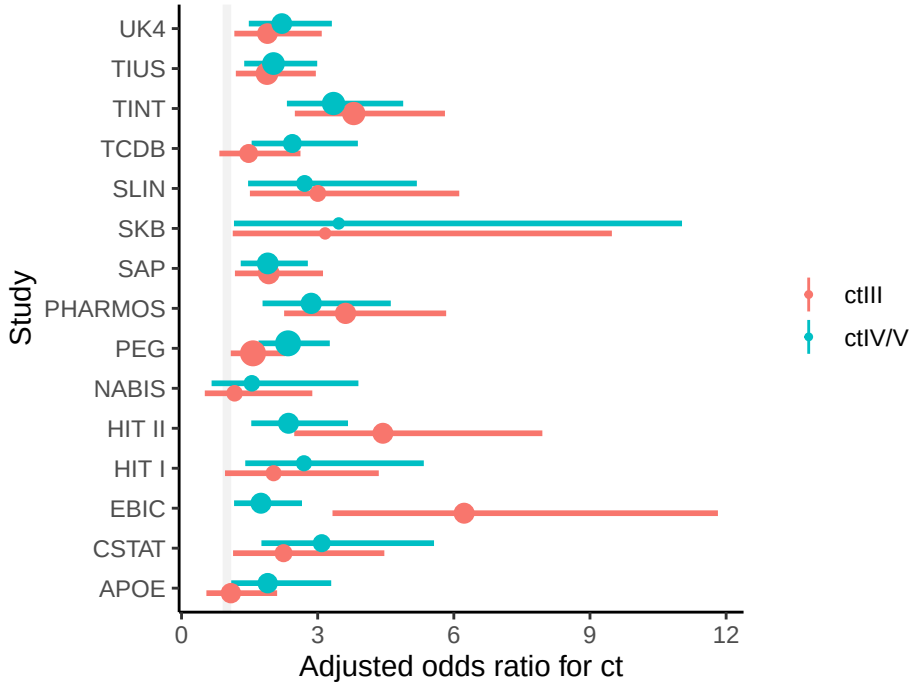
Note that variables `hypots`, `hypox`, `tsah` and `edh` are not part of the analysis model, and may thus serve as auxiliary variables for imputation.

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: show how much variance there is after different methods]

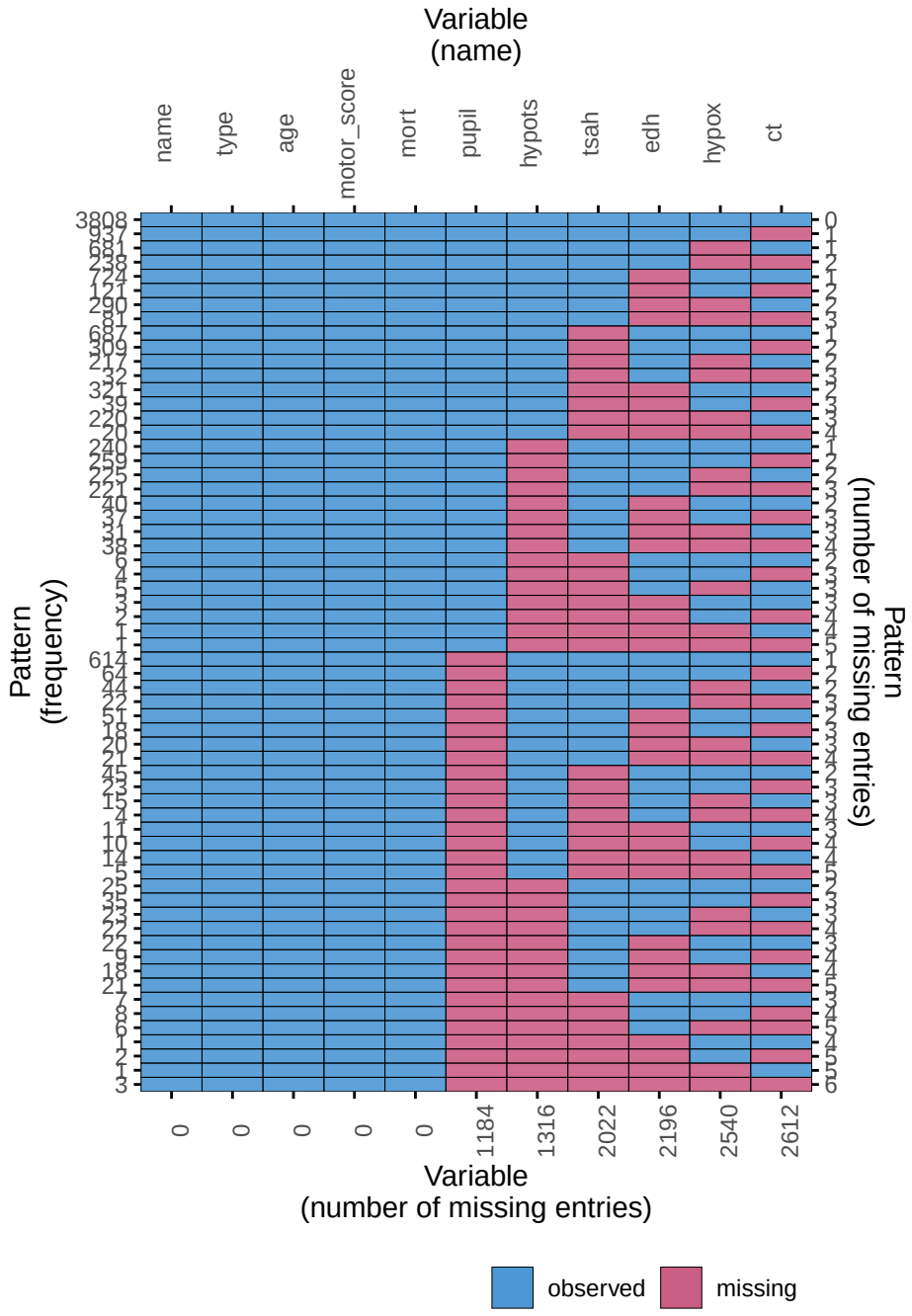
[TODO: add ICC before/after imputation and interpret: This tells us that the multilevel structure of the data should probably be taken into account. If we don't, we'll may end up with incorrect imputations, biasing the effect of the clusters towards zero.]

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

3.1. Missingness

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

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

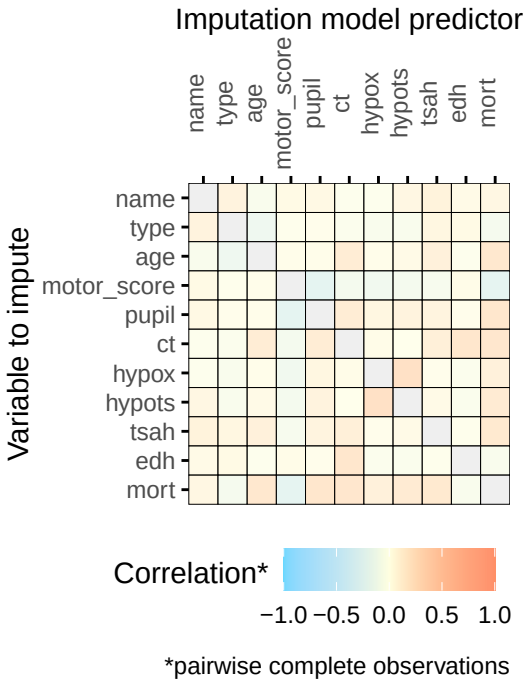


This shows... [TODO: fill in that we need to impute `ct` and `pupil`.]

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
```

3.2. Complete case analysis [TODO: remove this?]

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        estim~1 std.er~2 stati~3   p.value conf.~4 conf.~5
```

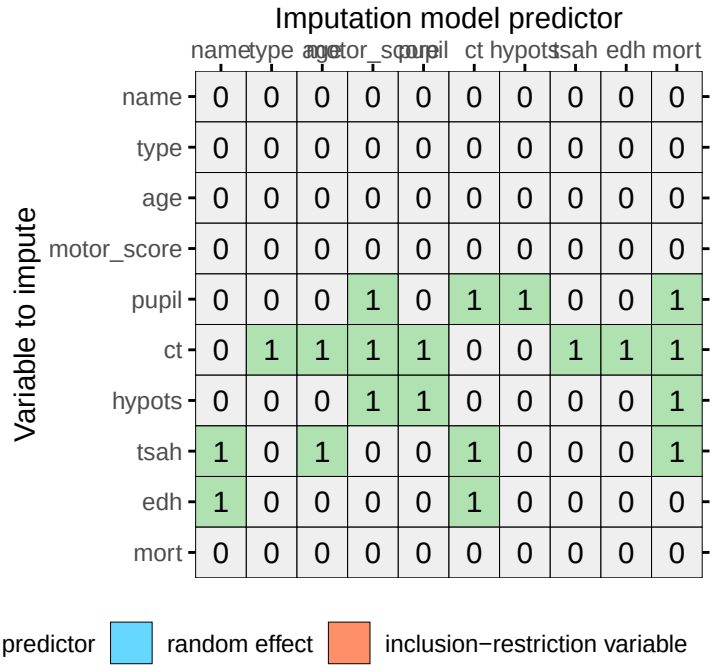
```
   <chr>    <chr> <chr>          <dbl>    <dbl>   <dbl>     <dbl>   <dbl>   <dbl>
 1 fixed    <NA>  (Intercept)  0.0863  0.0182   -11.6   2.99e-31  0.0571   0.130
 2 fixed    <NA>  typeRCT      0.757   0.137     -1.54  1.22e- 1  0.531    1.08
 3 fixed    <NA>  age          1.03    0.00265   12.9   7.40e-38  1.03     1.04
 4 fixed    <NA>  motor_scor~  0.651   0.0732    -3.82  1.34e- 4  0.522    0.811
 5 fixed    <NA>  motor_scor~  0.489   0.0555    -6.30  2.97e-10  0.391    0.611
 6 fixed    <NA>  motor_scor~  0.274   0.0321   -11.0   2.28e-28  0.218    0.345
 7 fixed    <NA>  pupilNone    3.20    0.317     11.7   8.18e-32  2.63     3.88
 8 fixed    <NA>  pupilOne     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/V       2.30    0.214      8.95  3.55e-19  1.92     2.76
11 ran_pars name  sd__(Inter~  0.230  NA         NA    NA        NA       NA
# ... with abbreviated variable names 1: estimate, 2: std.error, 3: statistic,
#   4: conf.low, 5: conf.high
```

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

3.3. Imputation model

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

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



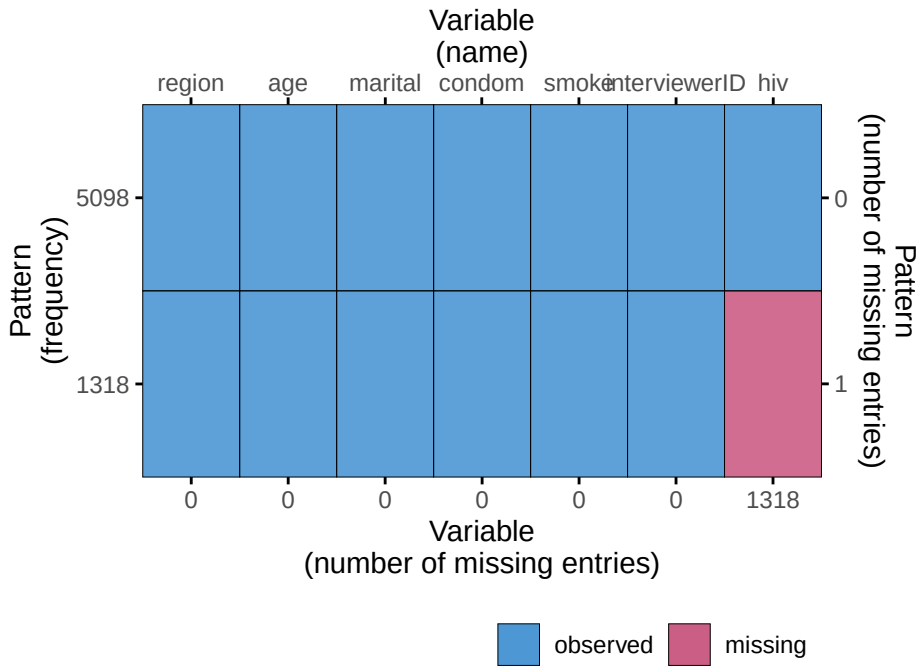
[TODO: mutate data to get the right data types for imputation (e.g. integer for clustering variable).]

4. Case study III: 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.



R=region+language+(1|InterviewID) model with with interviewer as random effects, because the observations are not independent. Interviews are not allocated randomly. In theory we expect the inclusion-restriction variable to be randomly assigned, that's why we're adding region and language to compensate for non-random allocation.

```
R> data("hiv",  package = "GJRM")
R> # We select  5 predictor variables over 9 regions
R> colnames(hiv)
```

```
 [1] "hivconsent"      "hiv"             "age"             "education"
 [5] "wealth"          "region"          "marital"         "std"
 [9] "age1sex_cat"     "highhiv"         "partner"         "condom"
[13] "aidscare"        "knowsdiedofaids" "evertestedHIV"   "smoke"
[17] "religion"        "ethnicity"       "language"        "interviewerID"
[21] "sw"

R> hivdata <- hiv[,c("hiv","hivconsent","age","marital","condom","highhiv","interviewerID"
R> # Study/group variable has to be recoded as integer
R> hivdata$region<-as.integer(hivdata$region)
R>
R> # Categorical variables have to be recoded as factor
R> # to use the 2l.binary imputation method, it is required that the level names should no
R>
R> hivdata$hiv <- as.factor(hivdata$hiv) #to use the 2l.heckman method, it is required tha
R> #the missing variable is stored as a factor in the dataset, otherwise the method will
R> #apply the imputation correction for a  missing continuous variable instead for
R> #a missing binary variable,which is in this case the binary response of the hiv test.
R>
R> hivdata$marital <- as.factor(hivdata$marital)
R> levels(hivdata$marital)<-c("never_married","currently_married","formerly_married")
R> hivdata$condom <- as.factor(hivdata$condom)
R> levels(hivdata$condom)<-c("No_Condom_Last_Intercourse","Condom_Last_Intercourse")
R> hivdata$highhiv <- as.factor(hivdata$highhiv)
R> levels(hivdata$highhiv)<-c("Not_High_Risk_of_HIV","High_Risk_of_HIV")
R> hivdata$interviewerID <- as.factor(hivdata$interviewerID)
R>
R> hivdata$interviewerID <-  as.factor(as.character(hivdata$interviewerID))
R> interv<-as.data.frame(table(hivdata$interviewerID,hivdata$region))
R>
R> summary(hivdata)

  hiv         hivconsent          age                    marital
 0   :4457   Min.   :0.0000   Min.   :15.00   never_married    :2513
 1   : 641   1st Qu.:1.0000   1st Qu.:20.00   currently_married:3582
 NA's:1318   Median :1.0000   Median :28.00   formerly_married : 321
             Mean   :0.7946   Mean   :30.12
             3rd Qu.:1.0000   3rd Qu.:37.00
             Max.   :1.0000   Max.   :59.00

                        condom                      highhiv     interviewerID
 No_Condom_Last_Intercourse:5319   Not_High_Risk_of_HIV:4655   404    : 306
 Condom_Last_Intercourse   :1097   High_Risk_of_HIV    :1761   303    : 248
                                                               807    : 247
                                                               403    : 246
                                                               903    : 241
                                                               305    : 232
```

```
                                                                (Other):4896
    region                   language
 Min.   :1.000   Language: bemba  :2347
 1st Qu.:3.000   Language: nyanja :1927
 Median :5.000   Language: other  : 606
 Mean   :4.946   Language: lozi   : 604
 3rd Qu.:7.000   Language: tonga  : 476
 Max.   :9.000   Language: english: 456
                 (Other)          :   0
```

```
R> #  We obtain here the random effects for each interviewer, this is an approximation
R> #of the interviewer's skill which will be used as an exclusion constraint.
R> #Here, since the location of the interviewer was not randomly assigned to the
R> #subjects, the assignment was corrected for region and language.
R>
R> hivdata$hivconsent <- as.factor(hivdata$hivconsent)
R> ID_mixed <- lme4::glmer(hivconsent ~ region + language+(1 | interviewerID), data = hivd
R> reffect  <- ranef(ID_mixed)$interviewerID
R> reffect$interviewerID <- levels(hivdata$interviewerID)
R> colnames(reffect)     <- c("IDreffect","interviewerID")
R> hivdata  <-  merge(hivdata, reffect, by="interviewerID", all.x=TRUE)
R>
R> hivdata$interviewerID<-NULL
R> hivdata$hivconsent<-NULL
R> hivdata$language<-NULL
R> summary(hivdata)
```

```
  hiv            age                      marital
 0   :4457   Min.   :15.00   never_married    :2513
 1   : 641   1st Qu.:20.00   currently_married:3582
 NA's:1318   Median :28.00   formerly_married : 321
             Mean   :30.12
             3rd Qu.:37.00
             Max.   :59.00
                         condom                     highhiv         region
 No_Condom_Last_Intercourse:5319   Not_High_Risk_of_HIV:4655   Min.   :1.000
 Condom_Last_Intercourse   :1097   High_Risk_of_HIV    :1761   1st Qu.:3.000
                                                               Median :5.000
                                                               Mean   :4.946
                                                               3rd Qu.:7.000
                                                               Max.   :9.000

  IDreffect
 Min.   :-0.682850
 1st Qu.:-0.195361
 Median :-0.080531
 Mean   :-0.007813
```

```
 3rd Qu.: 0.257165
 Max.   : 0.667147

R> #Set prediction matrix and methods
R> ini <- mice(hivdata, maxit = 0)
R> meth<-ini$method
R> meth["hiv"]<-"2l.binary"
R> pred <- ini$pred
R> pred[,"region"] <- 0
R> pred["region",] <- 0
R> pred["hiv","region"]<- -2
R> pred["hiv","IDreffect"]<- 0
R>
R> # MAR imputation model, we used the 2l.binary from the miceadds package
R> data_imp_mar <- mice( data= hivdata, # dataset with missing values
+                       m = 10,   # number of imputations
+                       meth = meth, #imputation method vector
+                       pred = pred, #imputation predictors matrix
+                       maxit=1)


 iter imp variable
  1   1  hiv
  1   2  hiv
  1   3  hiv
  1   4  hiv
  1   5  hiv
  1   6  hiv
  1   7  hiv
  1   8  hiv
  1   9  hiv
  1   10  hiv

R> # source("mice.impute.2l.heckman.R")
R> # sum(as.numeric(data_imp_mar[["imp"]][["hiv"]][["1"]]))
R> # sum(as.numeric(data_imp_heckp[["imp"]][["hiv"]][["1"]]))
R> # sum(as.numeric(data_imp_mar[["imp"]][["hiv"]][["2"]]))
R> # sum(as.numeric(data_imp_heckp[["imp"]][["hiv"]][["2"]]))
R>
R> # Heckman model
R> pred["hiv","IDreffect"] <- -3
R> meth<-ini$method
R> meth["hiv"]<-"2l.heckman"
R>
R> data_imp_heckp <- mice( hivdata, # dataset with missing values
+                         m = 10,   # number of imputations
+                         seed = 1234, #seed attached to the dataID
```

```
+                         meth = meth, #imputation method vector
+                         pred = pred, #imputation predictors matrix
+                         print = T,
+                         meta_method="reml",
+                         pmm=FALSE)
```

```
 iter imp variable
  1   1  hiv
  1   2  hiv
  1   3  hiv
  1   4  hiv
  1   5  hiv
  1   6  hiv
  1   7  hiv
  1   8  hiv
  1   9  hiv
  1   10  hiv
  2   1  hiv
  2   2  hiv
  2   3  hiv
  2   4  hiv
  2   5  hiv
  2   6  hiv
  2   7  hiv
  2   8  hiv
  2   9  hiv
  2   10  hiv
  3   1  hiv
  3   2  hiv
  3   3  hiv
  3   4  hiv
  3   5  hiv
  3   6  hiv
  3   7  hiv
  3   8  hiv
  3   9  hiv
  3   10  hiv
  4   1  hiv
  4   2  hiv
  4   3  hiv
  4   4  hiv
  4   5  hiv
  4   6  hiv
  4   7  hiv
  4   8  hiv
  4   9  hiv
```

```
4   10  hiv
5   1  hiv
5   2  hiv
5   3  hiv
5   4  hiv
5   5  hiv
5   6  hiv
5   7  hiv
5   8  hiv
5   9  hiv
5   10  hiv
```

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.

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.

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.

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