Journal of Statistical Software

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

Imputation of Incomplete Multilevel Data with mice

Hanne I. Oberman
Methodology and Statistics
Utrecht University

Johanna Muñoz
Julius Center for Health Sciences and Primary Care,
University Medical Center Utrecht, Utrecht University,
Utrecht, The Netherlands

Thomas P. A. Debray
Julius Center for Health Sciences and Primary Care,
University Medical Center Utrecht, Utrecht University,
Utrecht, The Netherlands

Gerko Vink
Methodology and Statistics
Utrecht University

Valentijn M. T. de Jong
Julius Center for Health Sciences and Primary Care,
University Medical Center Utrecht, Utrecht University,
Utrecht, The Netherlands
Data Analytics and Methods Task Force,
European Medicines Agency,
Amsterdam, The Netherlands

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 accomodated, but are outside the scope of this paper.

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

1. Introduction

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

Table 1: Concepts in multilevel methods

Concept	Details
Sample unit	Units of the population from which measurements are taken in a sample.
Hierarchical levels	Data are grouped into clusters at different levels, observations belonging to the same cluster are expected to share certain characteristics.
Fixed effect	Effects that are constant across all sample units, e.g. something that researchers control for and can repeat, such as the administration of a drug.
Random effect	Effects that are a source of random variation in the data, and whose levels are not fully sampled. e.g. individuals are drawn from a population of hospitals, here it is not possible to sample all hospitals but drug effects could vary between hospitals.
Mixed effect	Includes fixed and random effects, e.g. the fixed effect would be the treatment effect of a drug and the random effect would be the ID of the hospital where the patient is treated. Multilevel models typically accommodate for variability by including a separate group mean for each cluster e.g random intercept on hospitals. In addition to random intercepts, multilevel models can also include random coefficients and heterogeneous residual error variances across clusters (see e.g. @gelm06, @hox17 and @jong21).
ICC	The variability due to clustering is often measured by means of the intraclass coefficient (ICC). The ICC can be seen as the percentage of variance that can be attributed to the cluster-level, where a high ICC would indicate that a lot of variability is due to the cluster structure.
Stratified intercept	

within a single cluster (e.g., school classes). More complex clustered structures may occur when there are multiple hierarchical levels (e.g., students in different schools or patients within hospitals within regions across countries), or when the clustering is non-nested (e.g., electronic health record data from diverse settings and populations within large databases). With clustered data we generally assume that individuals from the same cluster tend to be more similar than individuals from other clusters. In statistical terms, this implies that observations from the same cluster are not independent and may in fact be correlated. If this correlation is left unaddressed, estimates of p values, confidence intervals even model parameters are prone to bias (Localio, Berlin, Ten Have, and Kimmel 2001). Statistical methods for clustered data typically adopt hierarchical models that explicitly describe the grouping of observations. These models are also known as 'multilevel models', 'hierarchical models', 'mixed effect models', 'random effect models', and in the context of time-to-event data as 'frailty models'. Table 1 provides an overview of some key concepts in multilevel modeling.

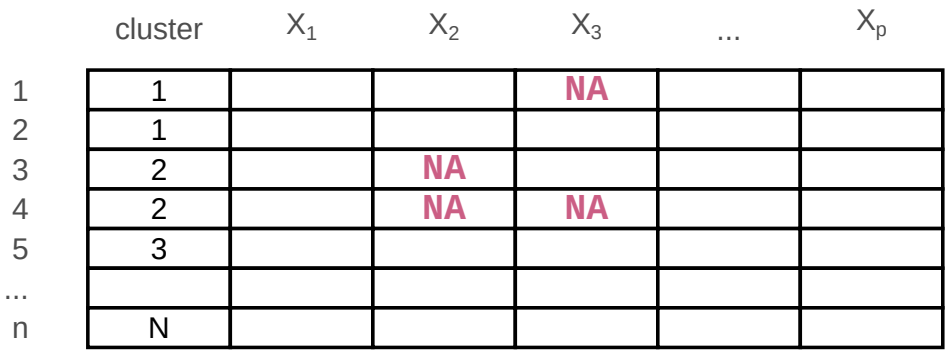


Figure 1: Sporadic missingness in multilevel data

1.1. Missingness in multilevel data

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 (Van Buuren 2018; Jolani 2018). For example, it is possible that test results are missing for several students in one or more classes. When all observations are missing within one or more clusters, data are said to be systematically missing.

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

Table 2: Concepts in missing data methods

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

Concept	Details
	(Rubin 1976; Meng 1994).

1.2. Aim of this paper

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

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

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

- `popmis` from the **mice** package (simulated data on perceived popularity, $n = 2,000$ pupils across $N = 100$ schools with data that are MAR, van Buuren and Groothuis-Oudshoorn 2021);
- `impact` from the **metamisc** package (empirical data on traumatic brain injuries, $n = 11,022$ patients across $N = 15$ studies with data that are MAR, Debray and de Jong 2021);
- `obesity` from the **micemd** package [simulated data on obesity, $n = 2,111$ patients across $N = 5$ regions with data that are MNAR].

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

This tutorial is dedicated to readers who are unfamiliar with multiple imputation. More experienced readers can skip the 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).

1.3. Setup

Install non-CRAN packages if necessary:

```
R> devtools::install_github("amices/ggmice")
```

Set up the R environment and load the necessary packages:

```
R> set.seed(2022)        # for reproducibility
R> library(mice)         # for imputation
```

```
R> library(miceadds)     # for additional imputation routines
R> library(ggmice)       # for incomplete/imputed data visualization
R> library(ggplot2)      # for visualization
R> library(dplyr)        # for data wrangling
R> library(lme4)         # for multilevel modeling
R> library(mitml)        # for multilevel parameter pooling
R> library(micemd)  # for imputation cf. heckman models
R> library(metamisc)     # for case study data
```

2. Case study I: 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 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)
```

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

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

First we plot the pattern of missing data within categories of the relevant variables. 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)
```

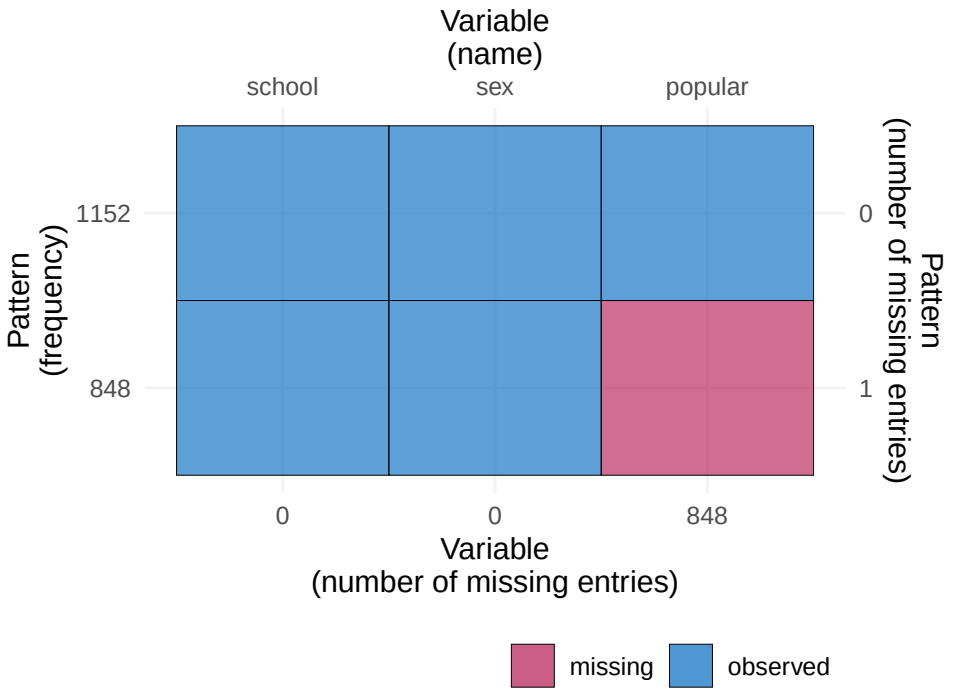
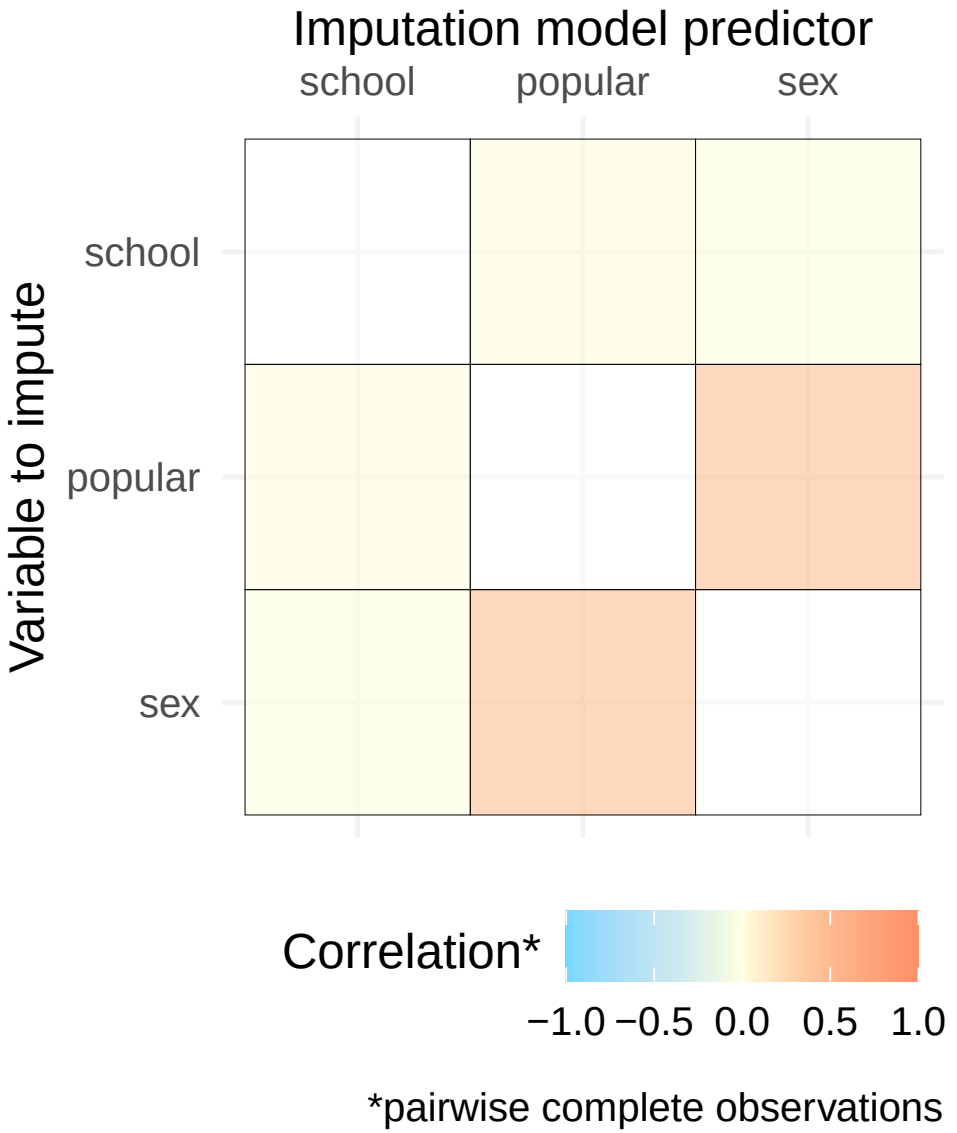
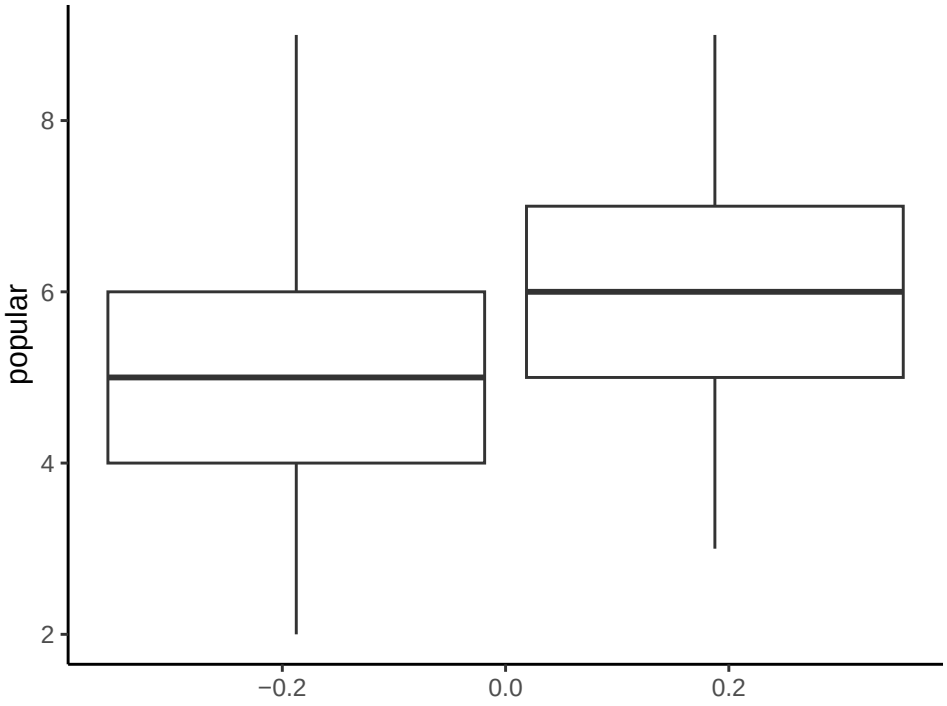


Figure 2: Missing data pattern in the popularity data



This shows us that `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()
```

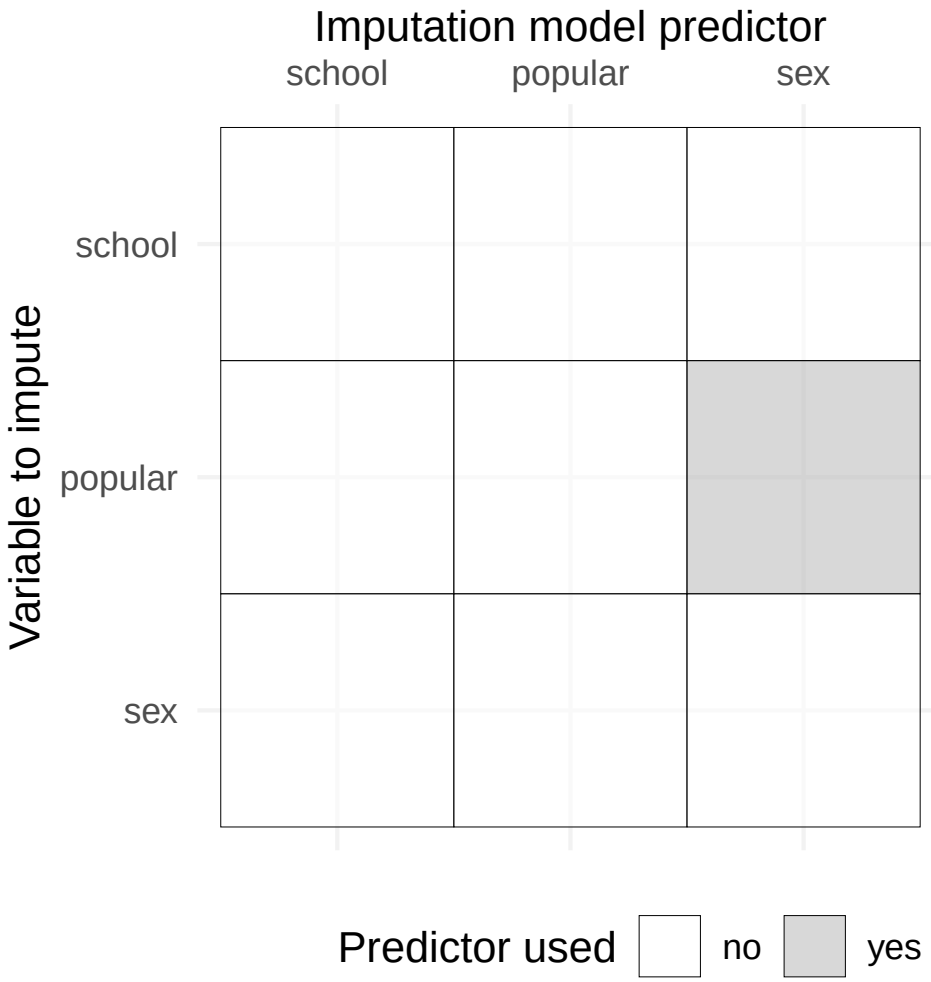


Imputation ignoring the cluster variable (not recommended)

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:

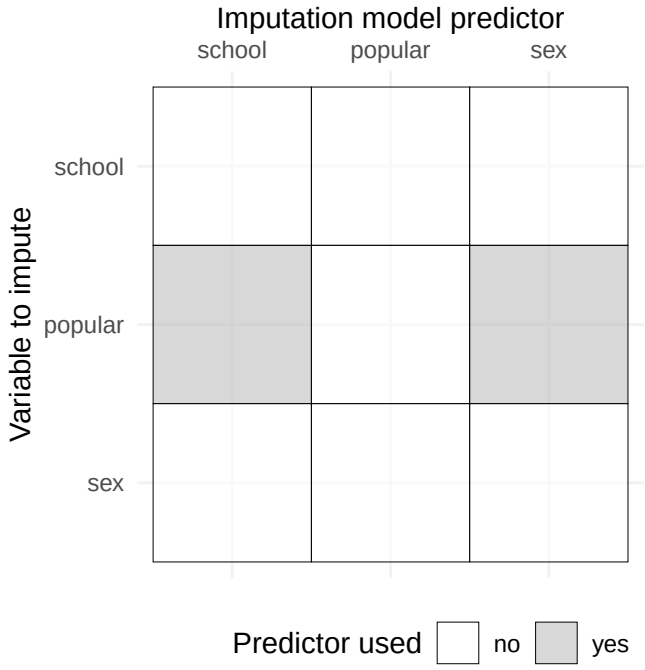
```
R> testEstimates(as.mitml.result(fit), extra.pars = TRUE)


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)     5.012     0.295    16.994     4.362     0.000    22.587     0.969
sex             0.695     0.251     2.768     4.287     0.047    28.390     0.975

                            Estimate
Intercept~~Intercept|school    0.266
Residual~~Residual             1.035
ICC|school                     0.208

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'.

```
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)))
```

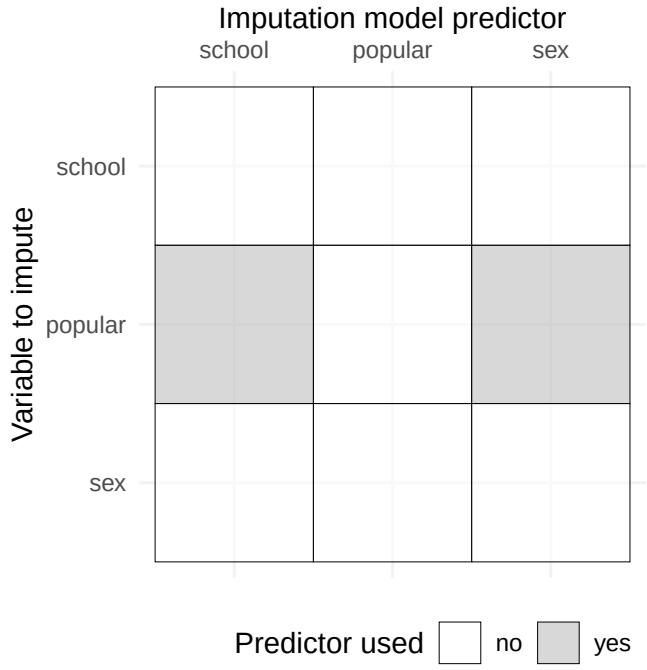
Print the estimates:

```
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)     4.915     0.217    22.642     4.926     0.000     9.110     0.926
sex             0.975     0.283     3.444     4.250     0.024    32.504     0.978

                           Estimate
Intercept~~Intercept|school    0.351
Residual~~Residual             1.153
ICC|school                     0.233

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)     5.011     0.410    12.222     4.226     0.000    35.955     0.980
sex             0.928     0.381     2.434     4.168     0.069    48.221     0.985

                           Estimate
Intercept~~Intercept|school    0.313
Residual~~Residual             1.428
```

```
ICC|school                    0.188

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

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

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). 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 mortality 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)
```

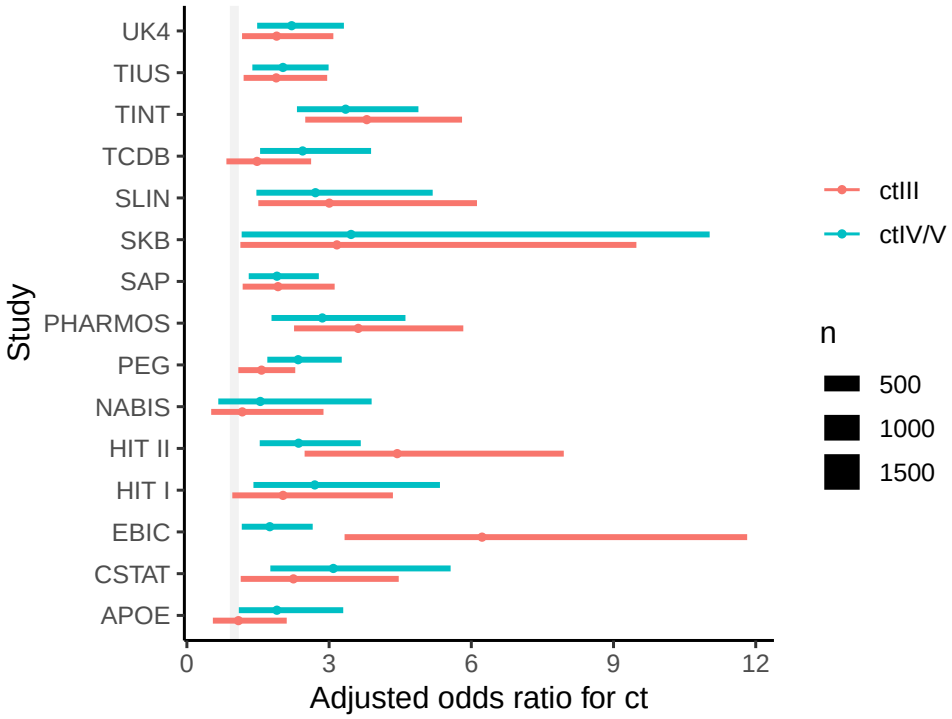
Note that variables `hypots`, `hypox`, `tsah` and `edh` are not part of the analysis model, and may thus serve as auxiliary variables for imputation.

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 **??**.

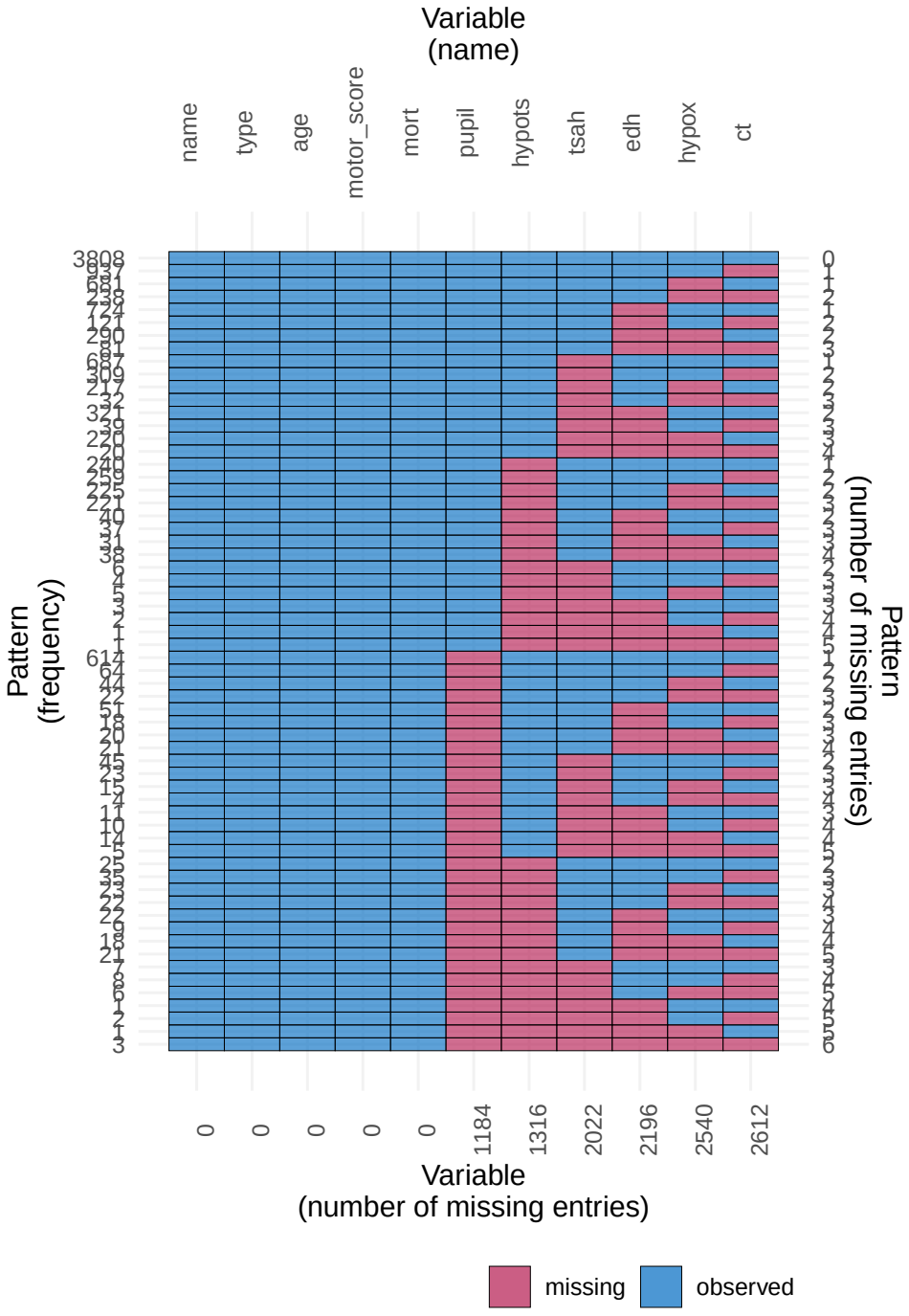



```
R> # fit <- glmer(mod, family = "binomial", data = impact) # fit the model
R> # tidy(fit, conf.int = TRUE, exponentiate = TRUE)       # print estimates
```

3.1. Missingness

To explore the missingness, it is wise to look at the missing data pattern. The ten most frequent missingness patterns are shown:

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



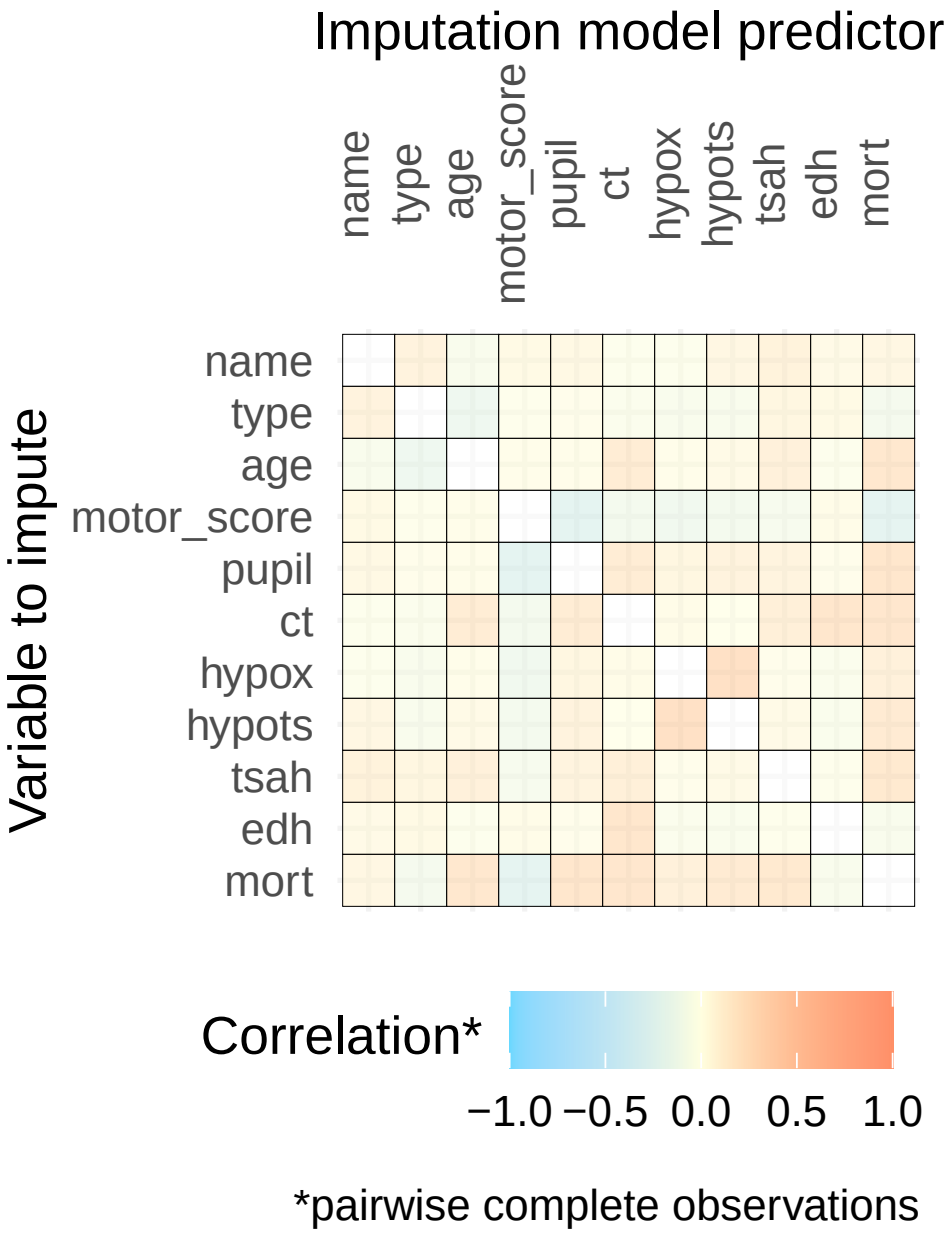
This shows that we need to impute `ct` and `pupil`.

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) # 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

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   3.00e-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  3.56e-19   1.92       2.76
11 ran_pa~ name  sd__~   0.230   NA           NA    NA         NA         NA
```

As we can see, a higher `ct` (Marshall Computerized Tomography classification) is associated with a lower odds of 6-month mortality, given by the odds ratio exp(0.42), CI ... to ..., when controlling for...

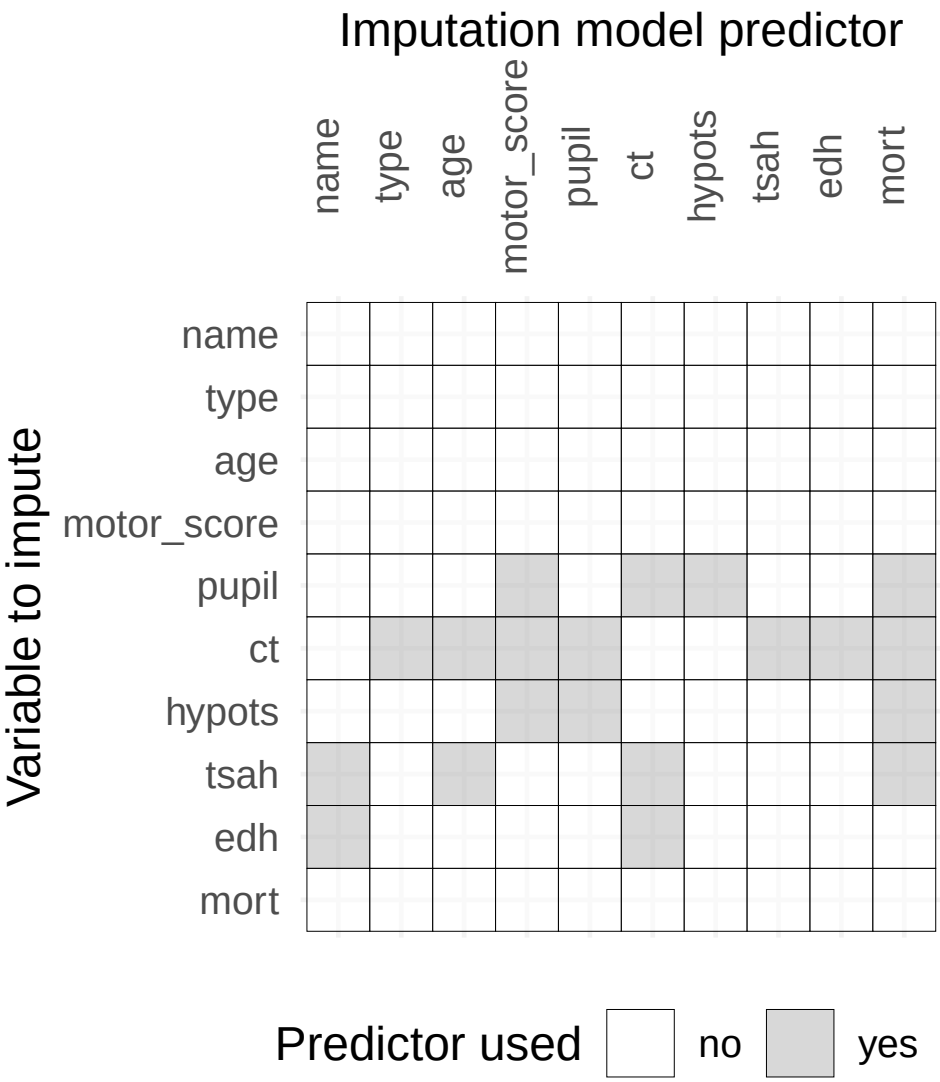
3.3. Imputation model

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

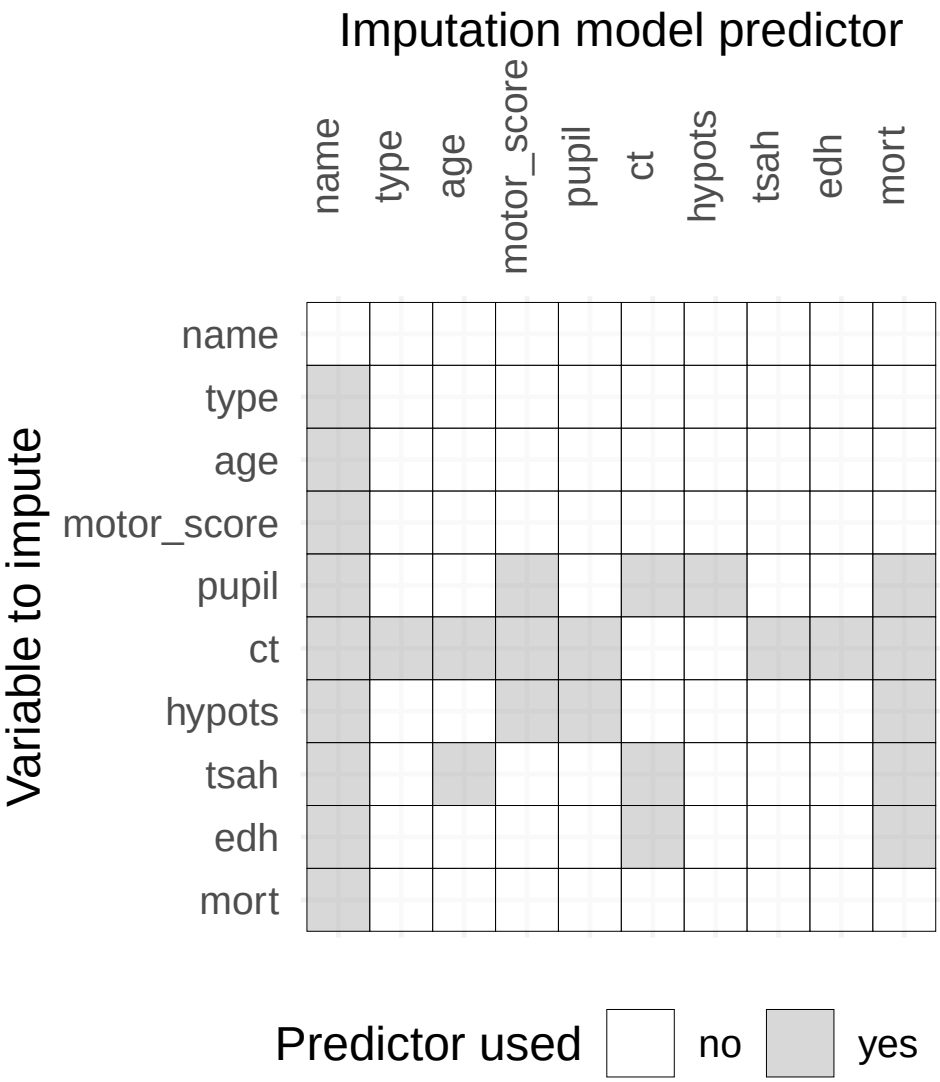
```
R> dat <- dat %>% mutate(across(everything(), as.integer))
```

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, rotate = TRUE)
```




```
R> pred[pred == 1] <- 2
R> pred["mort", ] <- 2
R> pred[, "mort"] <- 2
R> pred[c("name", "type", "age", "motor_score", "mort"), ] <- 0
R> pred[, "name"] <- -2
R> diag(pred) <- 0
R> plot_pred(pred, rotate = TRUE)
```




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

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

Impute the incomplete data

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

R> fit <- imp %>%
+   with(glmer(mort ~ type + age + as.factor(motor_score) + pupil + ct + (1 | name), famil
R> tidy(pool(fit))
```

```
                    term    estimate   std.error  statistic      p.value
1             (Intercept) -2.35203726 0.340181747  -6.914061 4.994037e-12
2                    type -0.41265892 0.180274846  -2.289054 2.209524e-02
3                     age  0.03049023 0.001570162  19.418521 1.238416e-81
4 as.factor(motor_score)2 -0.66764920 0.068737865  -9.712975 3.480413e-22
5 as.factor(motor_score)3 -1.05520001 0.070218940 -15.027285 2.540225e-50
6 as.factor(motor_score)4 -1.51238349 0.072304262 -20.916934 1.850073e-90
7                   pupil  0.48421447 0.038982800  12.421234 6.772069e-17
8                      ct  0.43474621 0.029968474  14.506785 2.342253e-36
             b           df dfcom         fmi      lambda m         riv
1 5.281599e-04 10119.70041 11013 0.005673265 0.005476771 5 0.005506932
2 4.881699e-05 10893.89378 11013 0.001985736 0.001802528 5 0.001805783
3 3.335358e-08  6320.89582 11013 0.016545467 0.016234340 5 0.016502243
4 4.188703e-05  8327.24771 11013 0.010875748 0.010638213 5 0.010752602
5 5.135721e-05  7632.20045 11013 0.012757637 0.012498967 5 0.012657168
6 1.403058e-04  2831.75462 11013 0.032888241 0.032205434 5 0.033277139
7 3.571497e-04    49.97295 11013 0.309130930 0.282023647 5 0.392803531
8 8.586645e-05   294.69765 11013 0.120677044 0.114729598 5 0.129598366
          ubar
1 1.150898e-01
2 3.244044e-02
3 2.425385e-06
4 4.674630e-03
5 4.869071e-03
6 5.059539e-03
7 1.091079e-03
8 7.950697e-04

R> as.mitml.result(fit)

[[1]]
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: mort ~ type + age + as.factor(motor_score) + pupil + ct + (1 |
    name)
      AIC       BIC    logLik  deviance  df.resid
10495.423 10561.192 -5238.712 10477.423     11013
Random effects:
 Groups Name        Std.Dev.
 name   (Intercept) 0.2843
Number of obs: 11022, groups:  name, 15
Fixed Effects:
            (Intercept)                     type                      age
               -2.37195                  -0.41014                  0.03052
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
               -0.65802                  -1.04611                  -1.51245
```

```
                pupil                      ct
              0.50405                 0.42496
optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings

[[2]]
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: mort ~ type + age + as.factor(motor_score) + pupil + ct + (1 |
    name)
     AIC      BIC   logLik deviance df.resid
10500.88 10566.65 -5241.44 10482.88    11013
Random effects:
 Groups Name        Std.Dev.
 name   (Intercept) 0.2917
Number of obs: 11022, groups:  name, 15
Fixed Effects:
           (Intercept)                     type                      age
              -2.37718                -0.41511                  0.03067
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
              -0.66935                -1.05211                 -1.49429
                pupil                      ct
              0.49013                 0.43835
optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings

[[3]]
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: mort ~ type + age + as.factor(motor_score) + pupil + ct + (1 |
    name)
      AIC       BIC    logLik  deviance  df.resid
10505.026 10570.795 -5243.513 10487.026     11013
Random effects:
 Groups Name        Std.Dev.
 name   (Intercept) 0.2908
Number of obs: 11022, groups:  name, 15
Fixed Effects:
           (Intercept)                     type                      age
              -2.32339                -0.42359                  0.03023
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
              -0.67142                -1.05776                 -1.51038
                pupil                      ct
              0.49756                 0.42474
optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings

[[4]]
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: mort ~ type + age + as.factor(motor_score) + pupil + ct + (1 |
    name)
      AIC       BIC    logLik  deviance  df.resid
10519.511 10585.280 -5250.755 10501.511     11013
Random effects:
 Groups Name        Std.Dev.
 name   (Intercept) 0.2961
Number of obs: 11022, groups:  name, 15
Fixed Effects:
            (Intercept)                     type                      age
               -2.33581                -0.40871                  0.03039
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
               -0.66477                -1.05451                -1.51858
                  pupil                       ct
                0.45928                 0.44419
optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 2 lme4 warnings

[[5]]
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: mort ~ type + age + as.factor(motor_score) + pupil + ct + (1 |
    name)
      AIC       BIC    logLik  deviance  df.resid
10522.038 10587.807 -5252.019 10504.038     11013
Random effects:
 Groups Name        Std.Dev.
 name   (Intercept) 0.2955
Number of obs: 11022, groups:  name, 15
Fixed Effects:
            (Intercept)                     type                      age
               -2.35187                -0.40574                  0.03064
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
               -0.67468                -1.06551                -1.52622
                  pupil                       ct
                0.47006                 0.44148
optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings

attr(,"class")
[1] "mitml.result" "list"



R> # testEstimates(as.mitml.result(fit))
```

4. Case study III: obesity data

In this example, we demonstrate a multilevel imputation of an intercept and slope random model with a continuous response. We use the obesity dataset included in the `micemd`@ package, a synthetic dataset that emulates an electronic survey in which individuals are asked to provide information about their weight and consumption habits in different countries. To easy the explanations we simulate data for 5 clusters and we reduce the dataset to the following variables:

- `region` Cluster variable,
- `gender` Gender (0=male, 1=female),
- `age` Age of the patient,
- `height` Height in meters,
- `weight` Weight in kilograms,
- 'time" Response time in minutes (inclusion-restriction variable).
- 'FamOb" Family obesity history (yes or not)

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. Researchers, based on observable data, are considering the inclusion of random intercept effects, as well as introducing a random slope for the Age variable.

The main model is given by:

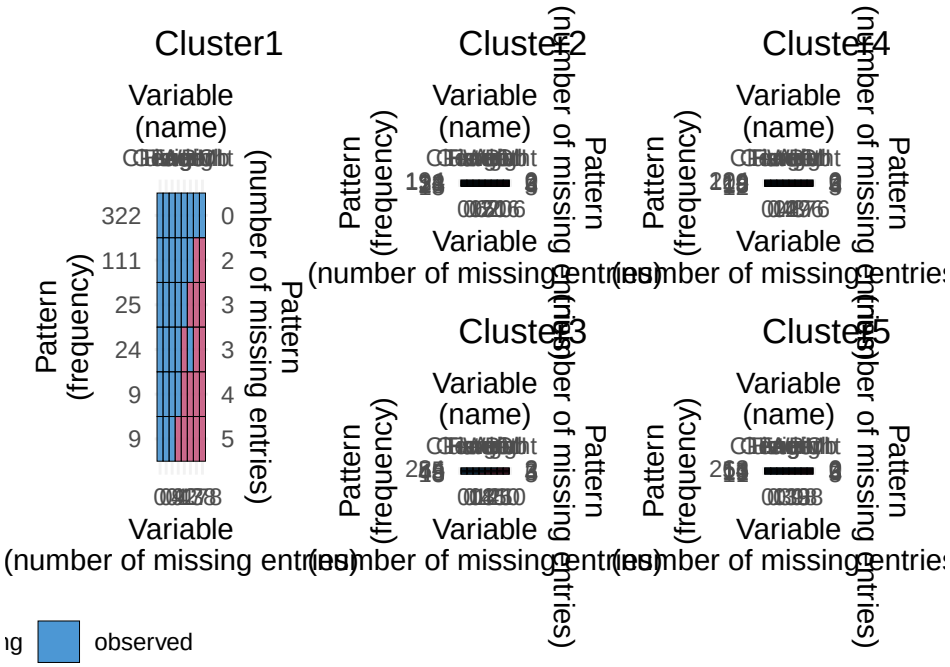
$$BMI_{ij} = (\beta_o + b_{oj}) + (\beta_1 + b_{oj}) * Age_{ij} + \beta_2 * FamOb_{ij} + \beta_3 Gender_{ij} + \epsilon_{ij}$$

We start by loading the data:

```
R> #data("data_heckman", package = "micemd")
R> #dat <- data_heckman
```

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

```
R> require(patchwork)
R>   myplots <- lapply(1:5, function(i) {
+          ggmice::plot_pattern(setDT(Obesity)[Cluster==i])+
+            ggtitle(paste0("Cluster", i))+
+       theme(legend.position = ifelse(i==1,"bottom","none"))
+       })
R>   myplots[[1]]+ myplots[[2]] /myplots[[3]]+ myplots[[4]] /myplots[[5]]
```



We observe that the missing pattern is quite similar across the clusters. However, regarding the weight variable, we notice that is systematically missing in cluster 3. In order to evaluate if it is required a imputation method for 1-level or 2-level we assess the Intraclass Correlation Coefficient (ICC) for the outcome variable, we use the "performance" package:

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

# Intraclass Correlation Coefficient

    Adjusted ICC: 0.342
  Unadjusted ICC: 0.342
```

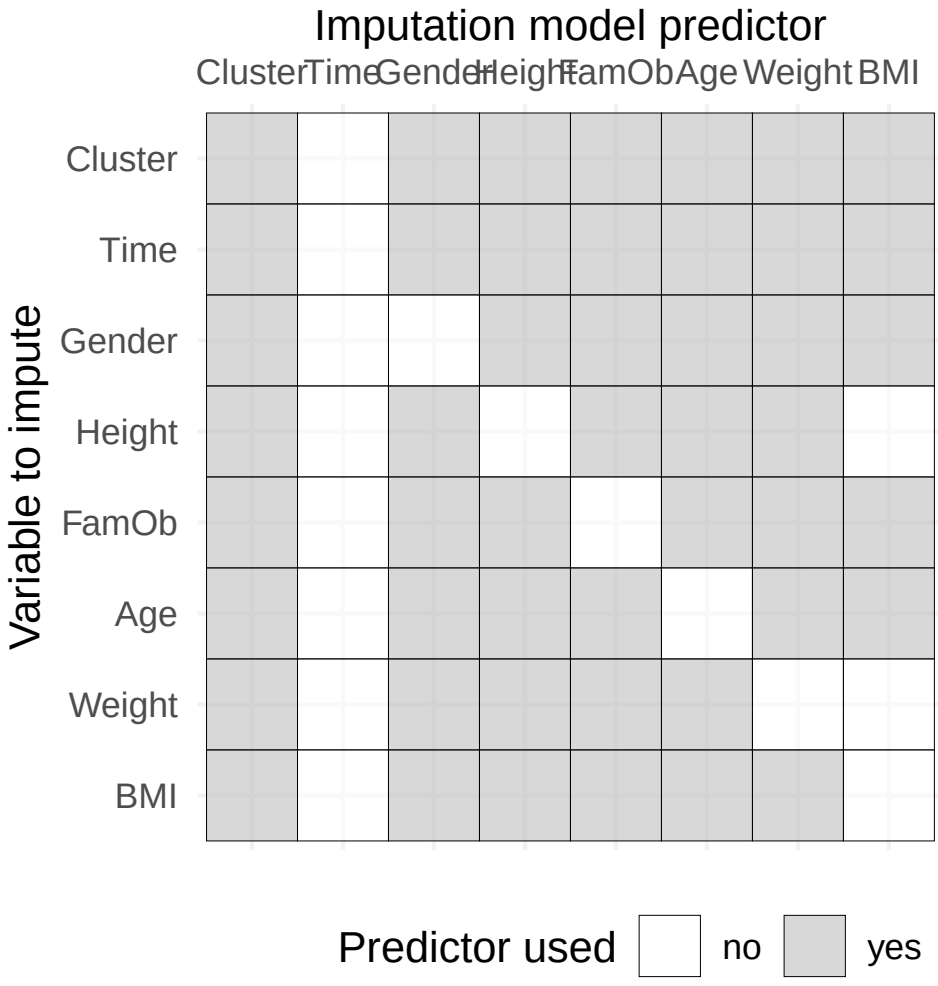
Since the ICC is above 0.1 and as the main analyis will be consider a mixed model, we decide to use two-level (2l) imputation methods. In this imputation process, we include all predictor variables in the main model and since BMI is a composite variable. We also incorporate weight and height variables.

To determine the best imputation method for these variables, we can use the `find.defaultMethod`function 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.

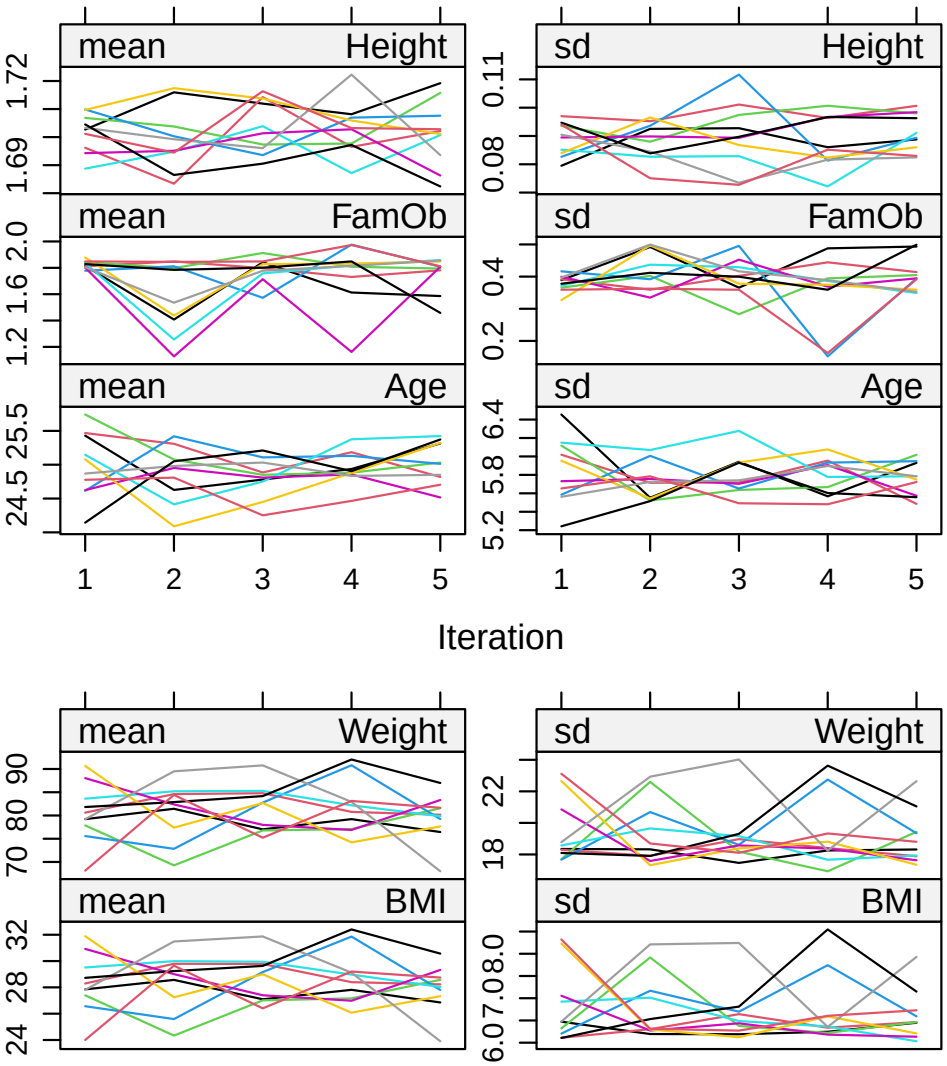
```
R> meth_mar <- meth_mar <- find.defaultMethod(Obesity, ind.clust=1, I.small = 7, ni.small
R> meth_mar["BMI"]<- "~ I(Weight / (Height)^2)"
R> meth_mar["Age"]<-"2l.2stage.pmm"
```

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

```
R> pred_mar <- mice(Obesity, maxit = 0)$pred   # predictor matrix
R> pred_mar[,"Cluster"]<- -2 # clustering variable
R> pred_mar[,"Time"]<-0
R> pred_mar[pred_mar==1]<-2
R> pred_mar[c("Height", "Weight"), "BMI"] <- 0
R> ggmice::plot_pred(pred_mar)
```

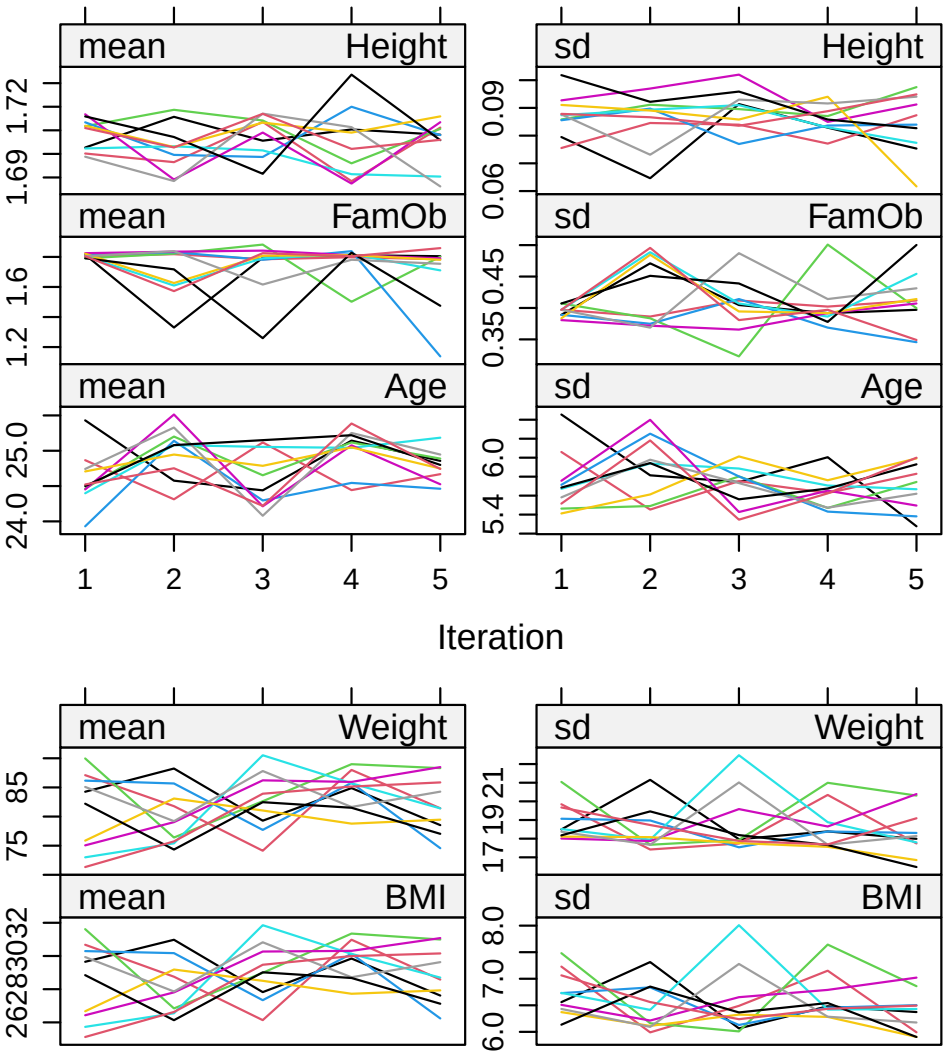



```
R> summary(complete(imp_mar,"long")$Weight)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.409  69.703  82.925  82.476  95.310 158.420

R> plot(imp_mar)
```



We are also contemplating the utilization of the "pmm" option, given that the imputed values' weights are lower than those of the observable values.

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

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  27.50   71.02   83.82   83.76   95.55  135.98

R> plot(imp_mar_pmm)
```



After confirming convergence, we proceed to save the results for future use. Users are considering the possibility that the weight variable may not have been selected randomly. 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. In contrast, the selection model approach includes methods such as the Heckman

model, which can be particularly useful in this case. Several methods, including Galimar2017, Hammon2021, and the recent Munoz method, designed for two-level data, allowing for variations in intercepts and exposure effects (random intercept and slope).

To apply the 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.

Here for the outcome equation we consider is 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 primary 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 is not have any influence on the weight assigned to the subject.

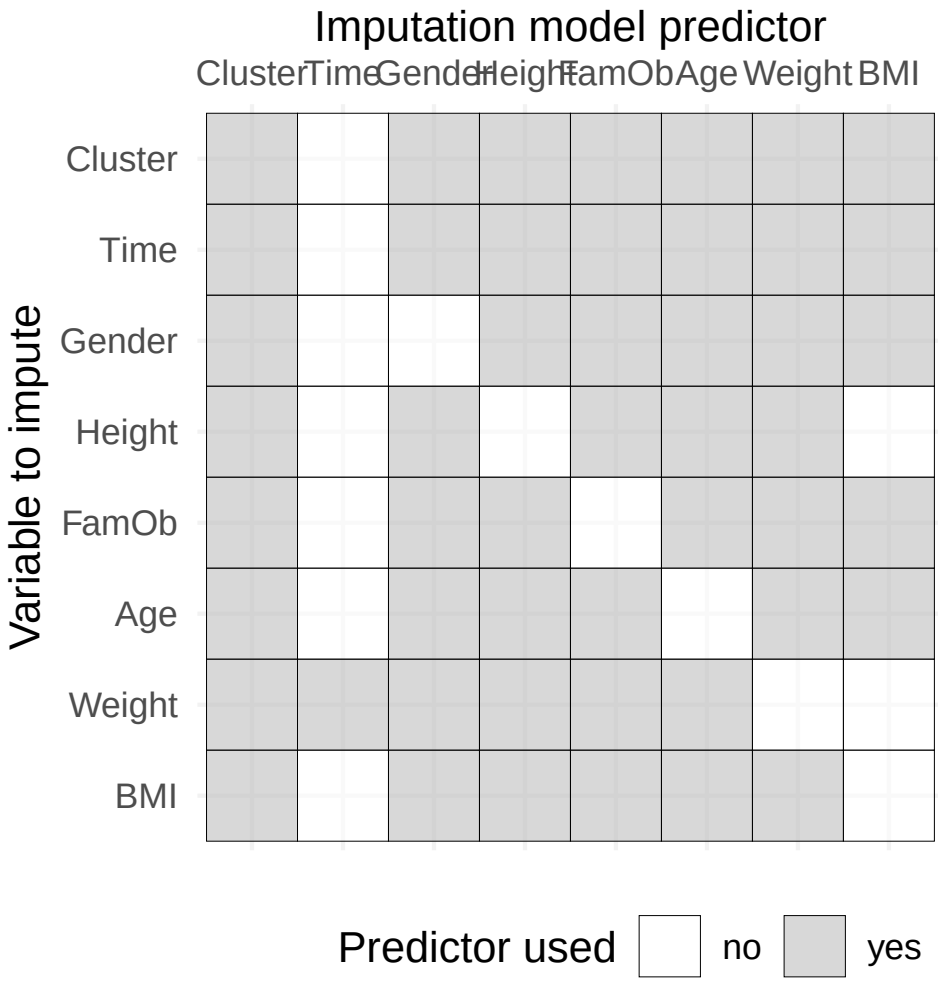
$$R_{ij} = \beta_o^S + \beta_1^S * Age_{ij} + \beta_2^S * FamOb_{ij} + \beta_3^S Gender_{ij} + \beta_4^S Time_{ij} + \epsilon_{ij}^S$$

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, you can refer to (Munoz).

To integrate 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's 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 labeled as (-3). Consequently, the only alteration needed in the predictor matrix pertains to the variable "Time."

```
R> pred_mnar <- pred_mar
R> pred_mnar["Weight","Time"]<- -3
R> ggmice::plot_pred(pred_mnar)
```



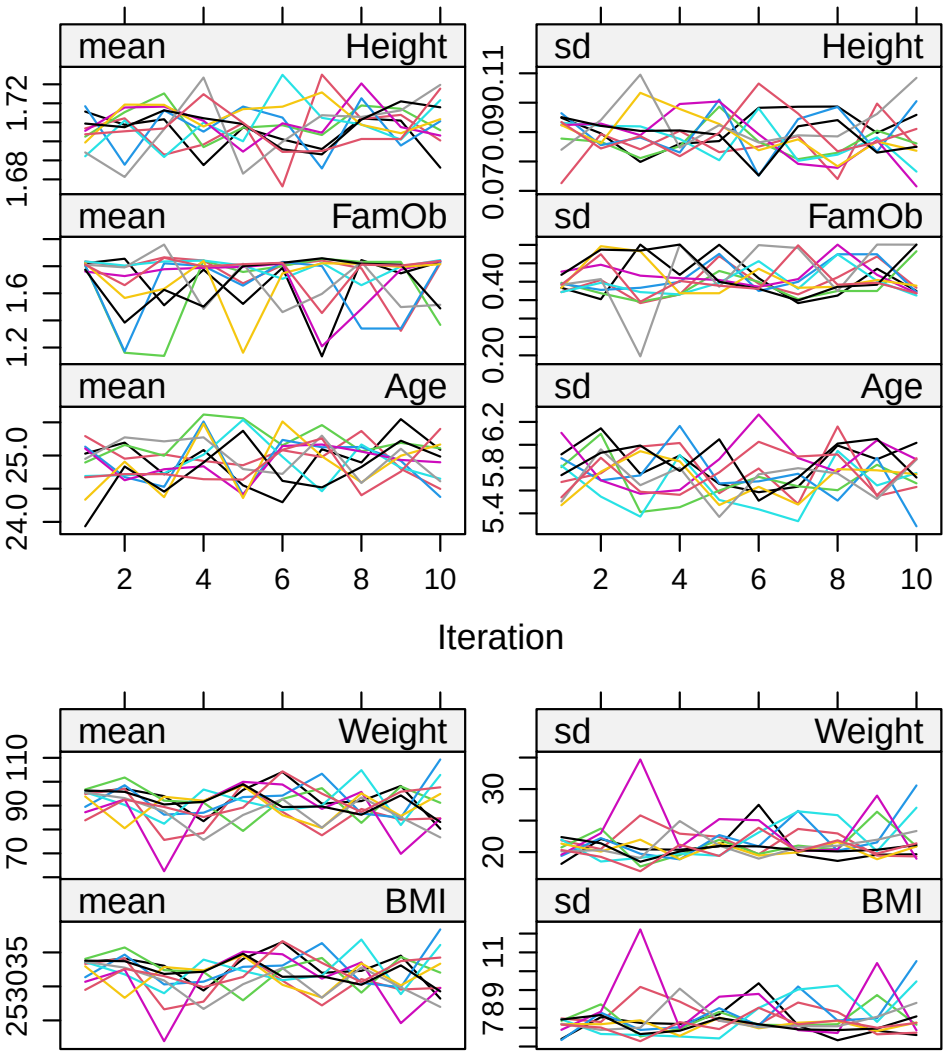
We also need to modify the method of the weight variable

```
R> meth_mnar <- meth_mar
R> meth_mnar["Weight"]<- "3l.2stage.heckman"
```

Then we proceed to run the imputation model as before Impute the missingness:

After executing these imputation procedures, it is essential to assess convergence and the coherence of the imputed values. Upon examining the weight variable, we noticed that the imputed range falls outside the realm of plausible values (as weight should be positive). Consequently, we are contemplating a different approach, specifically, the use of "pmm," which involves imputing values from donor observations. This approach ensures that the imputed values remain within the range of observable values.

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

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.26   73.41   86.94   88.27  101.73  186.47

R> plot(imp_mnar)
```



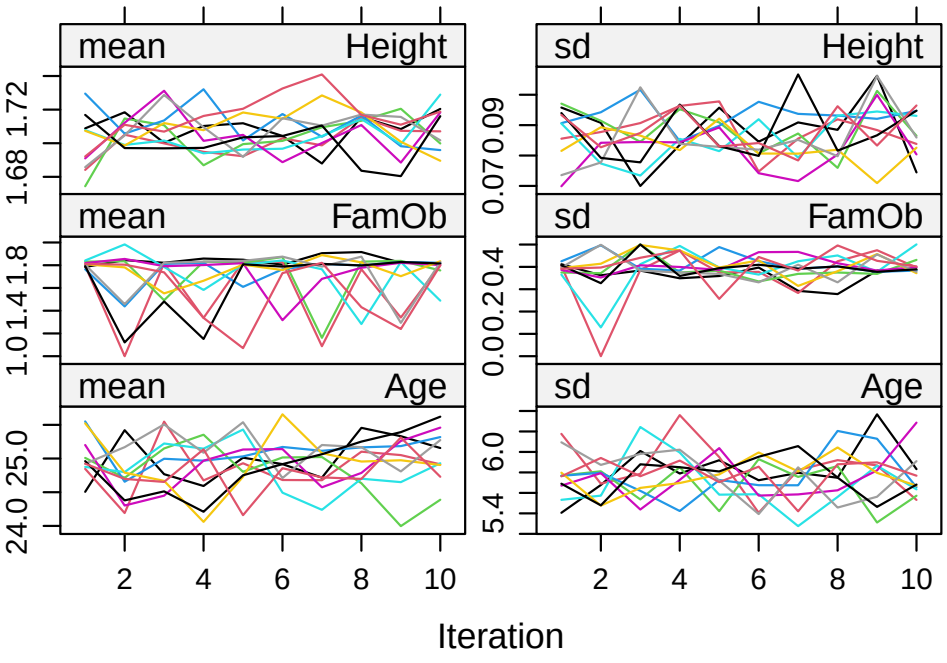
We then run the imputation model but this time using the option of pmm, to assure that weight values are in the range of the observable data, this can be implemented by seeting the pmm parameter as true.

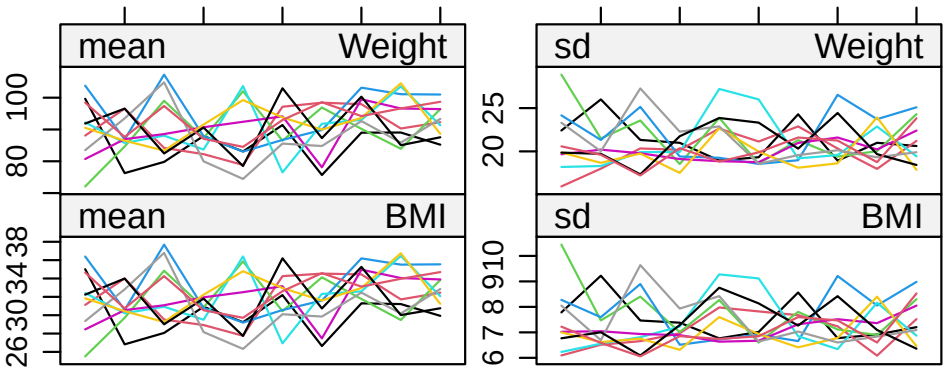
```
R> summary(complete(imp_mnar_pmm,"long"))
```

```
      .imp            .id          Cluster          Time          Gender
 Min.   : 1.0   Min.   :   1   Min.   :1.000   Min.   :1.016   Female:10720
 1st Qu.: 3.0   1st Qu.: 528   1st Qu.:2.000   1st Qu.:3.724   Male  :10390
```

```
Median : 5.5   Median :1056   Median :3.000   Median :5.151
Mean   : 5.5   Mean   :1056   Mean   :2.868   Mean   :5.211
3rd Qu.: 8.0   3rd Qu.:1584   3rd Qu.:4.000   3rd Qu.:6.698
Max.   :10.0   Max.   :2111   Max.   :5.000   Max.   :9.957
    Height       FamOb            Age             Weight           BMI
Min.   :1.450   no : 3794   Min.   :14.00   Min.   : 27.50   Min.   : 9.59
1st Qu.:1.642   yes:17316   1st Qu.:20.00   1st Qu.: 75.19   1st Qu.:25.84
Median :1.701               Median :25.00   Median : 88.88   Median :30.57
Mean   :1.702               Mean   :24.82   Mean   : 89.61   Mean   :31.02
3rd Qu.:1.760               3rd Qu.:29.00   3rd Qu.:103.78   3rd Qu.:35.89
Max.   :1.961               Max.   :47.00   Max.   :135.98   Max.   :63.21
```

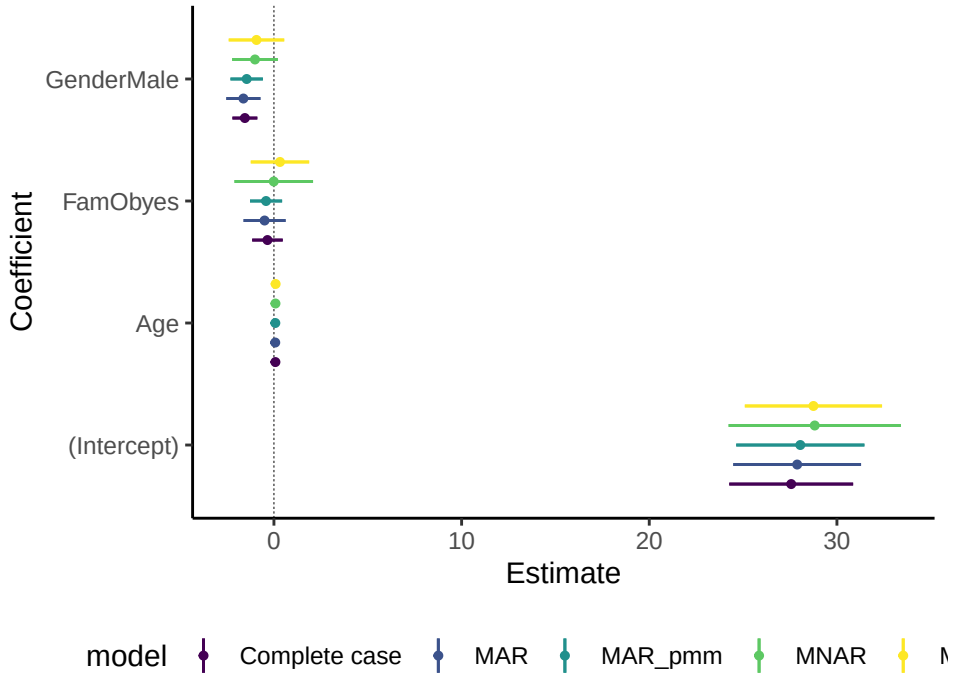
```
R> plot(imp_mnar_pmm)
```





Then after this modification we proceed to compare the effects on the model, therefore we run the analysis model on each of the completed datasets along with the dataset where the incomplete values are removed (CC).

```
R> library(ggplot2)
R> cc_rs<- with(Obesity[complete.cases(Obesity),],lme( BMI ~ Age + FamOb + Gender,random=~
R> mar_rs <- with(imp_mar,lme( BMI ~ Age + FamOb + Gender,random=~1+Age|Cluster))
R> mar_pmm_rs <- with(imp_mar_pmm,lme( BMI ~ Age + FamOb + Gender,random=~1+Age|Cluster))
R> mnar_rs<- with(imp_mnar,lme(BMI ~ Age + FamOb + Gender,random=~1+Age|Cluster))
R> mnar_pmm_rs<- with(imp_mnar_pmm, lme(BMI ~ Age + FamOb + Gender,random=~1+Age|Cluster))
R> list_models<-list(cc_rs,mar_rs,mar_pmm_rs,mnar_rs,mnar_pmm_rs)
R> plot_models(list_models,mod_name=c("Complete case", "MAR","MAR_pmm", "MNAR","MNAR_S", "
```



We note that there is minimal disparity in the age effect, FamObs, across the various imputation models under consideration. Nonetheless, with respect to Gender, we observe a notable distinction: in the M(C)AR assumption, the effect is significant, whereas under the MNAR assumption, it becomes insignificant. An analysis of the intercept reveals that, under the MNAR assumption, a higher average BMI is anticipated compared to the MAR assumption.

Visualize missing data pattern:

The matrix only shows the predictors for the main model, not the selection model.

5. Discussion

ORDER:

- summary
- congeniality, then in hierarchical models
- look whether we can fit cong. back in the main body
- alt. methods
- conclusion: mice is really easy!
- Additional levels of clustering
- More complex data types: timeseries and polynomial relationship in the clustering.
- FIML vs MI

An alternative approach to missing data is to use Full Information Maximum Likelihood (FIML). This method does not require the imputation of any missing values. Whereas MI consists of imputation, analyses and pooling steps, FIML analyses the data in a single step. When the assumptions are met the two approaches should produce equivalent results. [REF] As FIML requires specialised software, not all analyses can be performed with standard software. [REF]

- Survival / TTE, this could be put in the paragraph on congeniality

When the outcome is time-to-event, the Nelson-Aalen estimate of the time to event should be included as a covariate in the imputation model [REF]

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.

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).

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. References

8. Appendix

Table 3: Notation

Formula lme4	**Details**
`y ~ x1 + (1 | g1)`	Fixed `x1` predictor with random intercept varying among `g1`
`y~x1*x2+ (1| g1)`	Interactions of `x1` and `x2` only in fixed effect
`y ~ x1*x2+ (x2| g1)`	Interactions of `x1` and `x2` only in fixed effect with slope of `x2` randomly varying among `g1`
`y ~ x1*x2+ (x1*x2| g1)`	variance-covariance matrix estimated only with the variance terms of intercept, slope of `x1`, slope of `x2` and interaction `x1*x2`
`y ~ x1*x2+ (x1 | g1)+ (x2| g1)`	variance-covariance matrix estimated separately, i.e, one for intercept and `x1` and another for intercept and `x2`
`y ~ x1 + (x1 | g1) or 1 + x1 + (1 + x1 | g1)`	Fixed `x1` with correlated random intercept and random slope of `x`
`y ~ x1 + (x1 || g1) or 1 + x1 + (1 | g1) + (0 + x1 | g1)`	Fixed `x1` with uncorrelated random intercept and random slope of `x1`
`y ~ (1 | g1) + (1 | g2)`	Random intercept varying among `g1` and among `g2` `| |`y ~ (1

References

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.

Meng XL (1994). "Multiple-Imputation Inferences with Uncongenial Sources of Input." Statistical Science, **9**(4), 538–558. ISSN 0883-4237, 2168-8745. doi:10.1214/ss/1177010269.

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

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 I. Oberman
Methodology and Statistics
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