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**.

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

1. Introduction

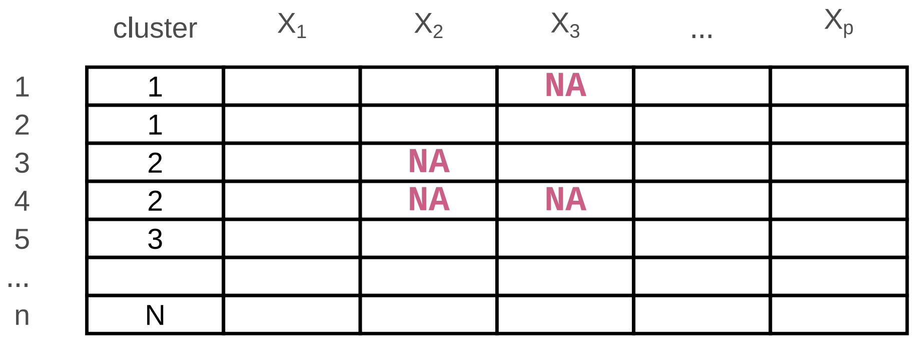
Many datasets include individuals that are clustered together, for example in geographic regions, or even different studies. In the simplest case, individuals (e.g., students) are nested within a single cluster (e.g., school classes). More complex clustered structures may occur when there are multiple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

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. A three-level
Fixed effect	Here we assume that the values of an independent variable are fixed, i.e., the values observed in the study are representative of all values in the in the dependent variable y e.g., blood pressure between treatments A and B.
Random effect	The values of an independent variable are assumed to be randomly drawn from admission we might select only certain hospitals that are representative of the difference of y between individual hospitals, but rather the variation of
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. Gelman and Hill (2006), Hox, Moerbeek, and van de Schoot (2017) and de Jong, Moons, Eijkemans, Riley, and Debray (2021)]. [TODO: add stratified intercept as concept..]

1.1. Missingness in multilevel data

As with any other dataset, clustered datasets may be impacted by missingness in much the



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

Figure 1: Sporadic missingness in multilevel data

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 (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 **miceheckman** 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")
R> devtools::install_github("hanneoberman/miceheckman")
```

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(miceheckman)  # 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, square = TRUE)
```

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)
```

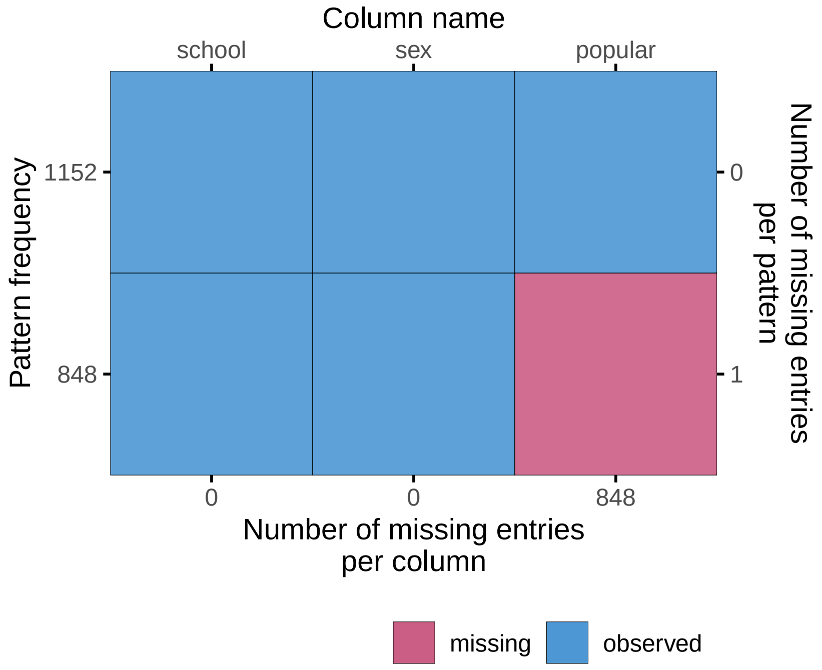
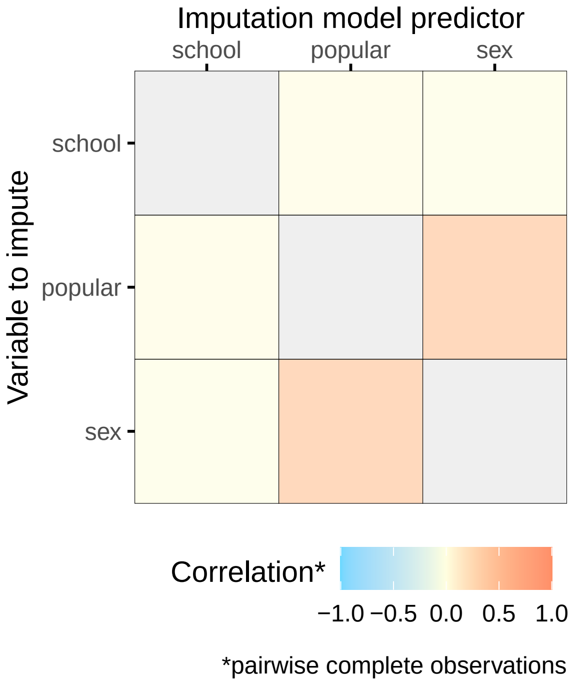
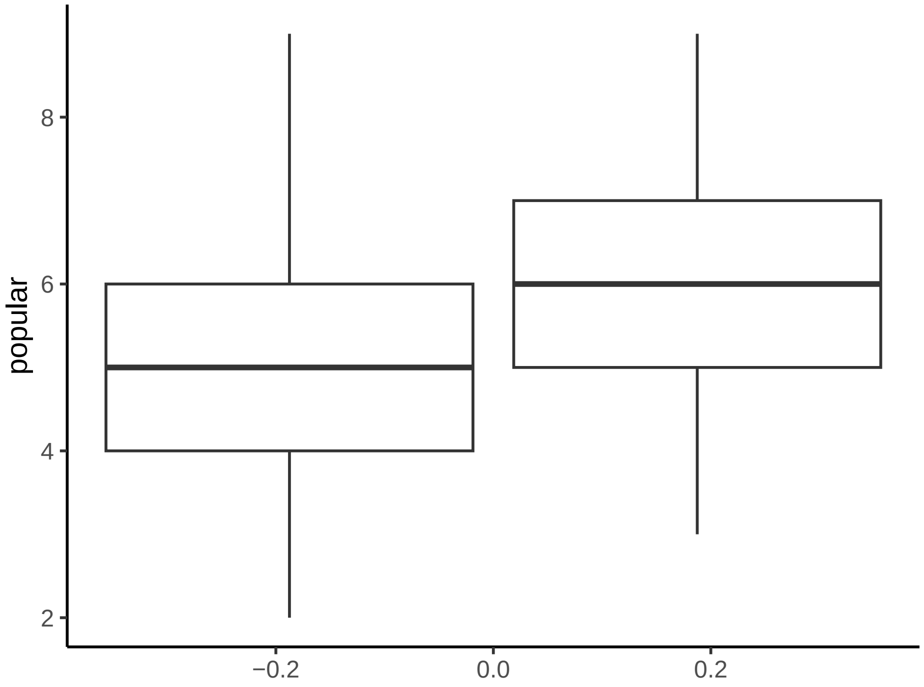


Figure 2: Missing data pattern in the popularity data



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()
```

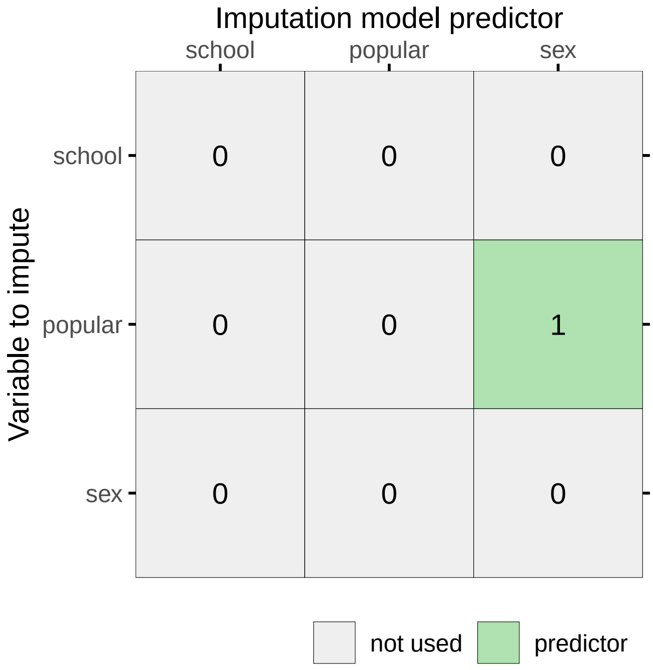


Imputation ignoring the cluster variable (not recommended)

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)
```

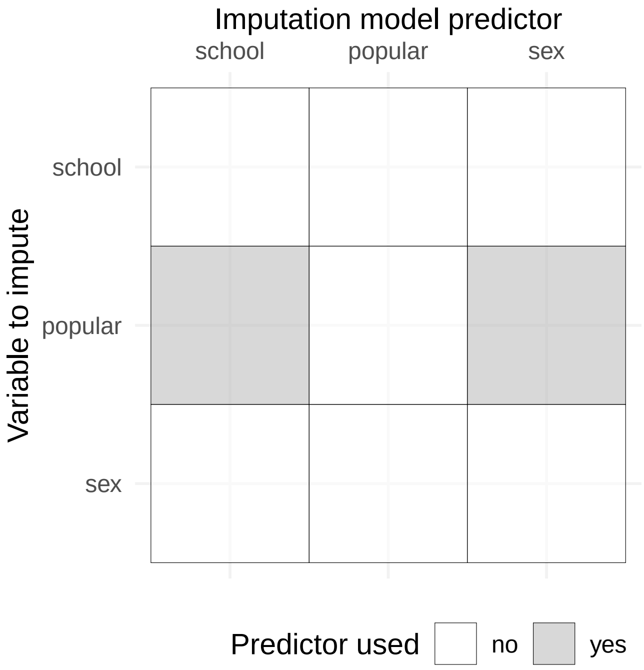
```

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)))
```

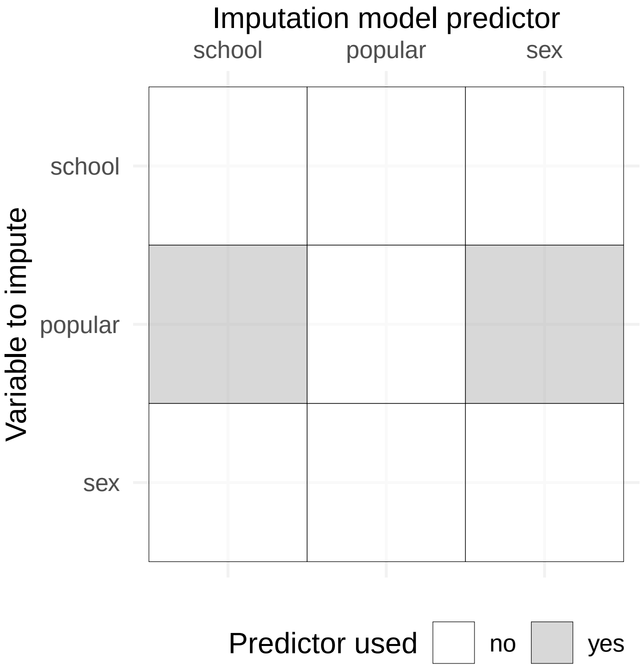
Print the estimates:

```
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)
```

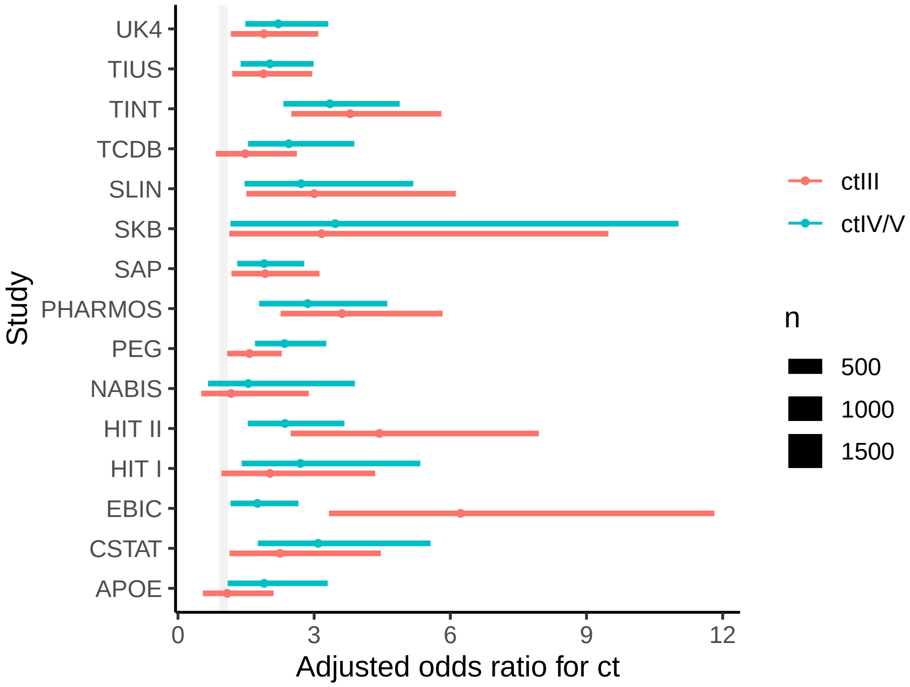
Note that variables `hypots`, `hypox`, `tsah` and `edh` are not part of the analysis model, and may thus serve as auxiliary variables for imputation.

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 ??.

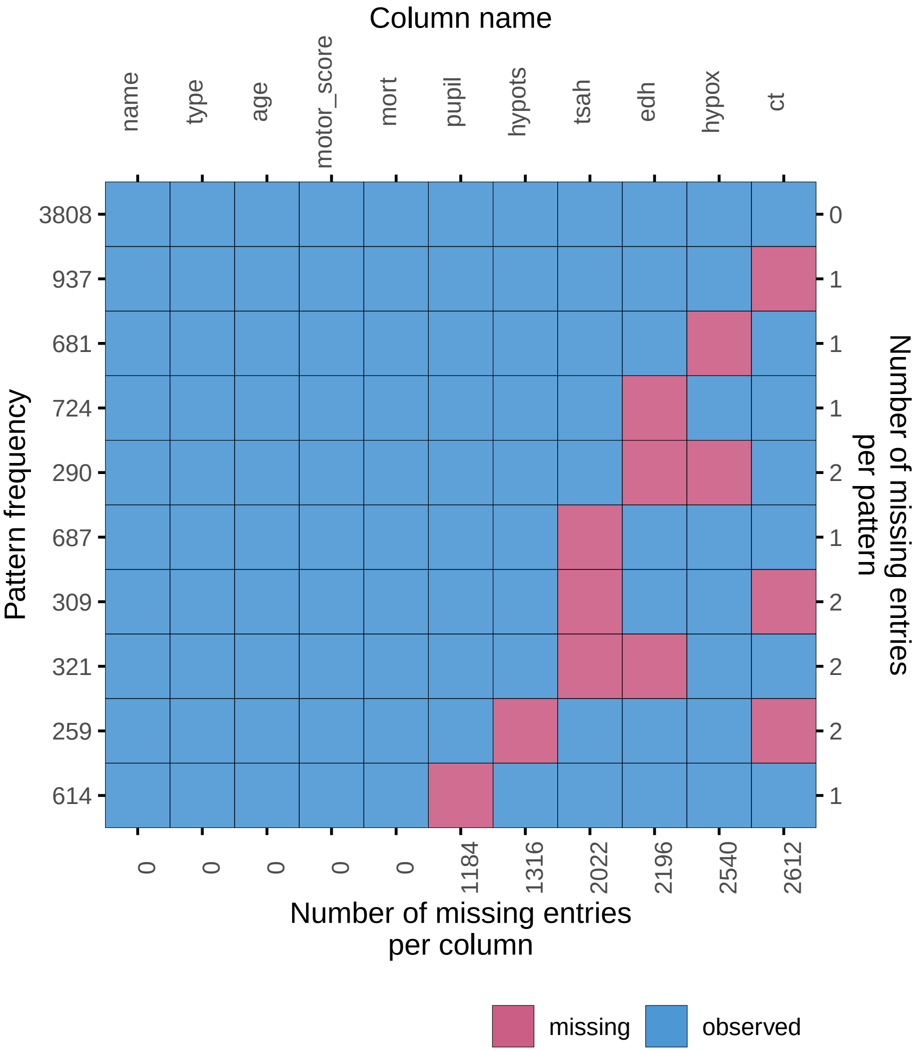


```
R> # fit <- glmer(mod, family = "binomial", data = impact) # fit the model
R> # tidy(fit, conf.int = TRUE, exponentiate = TRUE)       # print estimates
```

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, npat = 10L)  # plot missingness pattern
```

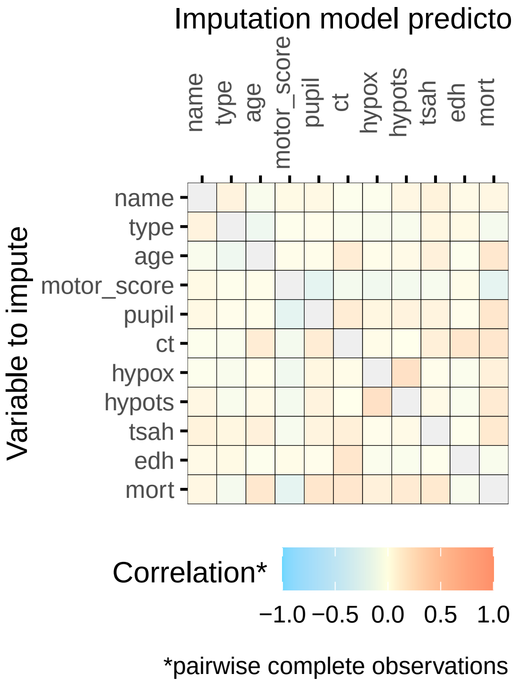


This shows that we need to impute `ct` and `pupil`.

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   2.99e-31   0.0571     0.130
 2 fixed   <NA>  type~   0.757    0.137       -1.54  1.22e- 1   0.531      1.08
 3 fixed   <NA>  age     1.03     0.00265     12.9   7.40e-38   1.03       1.04
 4 fixed   <NA>  moto~   0.651    0.0732      -3.82  1.34e- 4   0.522      0.811
 5 fixed   <NA>  moto~   0.489    0.0555      -6.30  2.97e-10   0.391      0.611
 6 fixed   <NA>  moto~   0.274    0.0321     -11.0   2.28e-28   0.218      0.345
 7 fixed   <NA>  pupi~   3.20     0.317       11.7   8.18e-32   2.63       3.88
 8 fixed   <NA>  pupi~   1.75     0.195        5.06  4.27e- 7   1.41       2.18
 9 fixed   <NA>  ctIII   2.41     0.268        7.89  3.05e-15   1.94       2.99
10 fixed   <NA>  ctIV~   2.30     0.214        8.95  3.55e-19   1.92       2.76
11 ran_pa~ name  sd__~   0.230   NA           NA    NA         NA         NA
```

As we can see, 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…

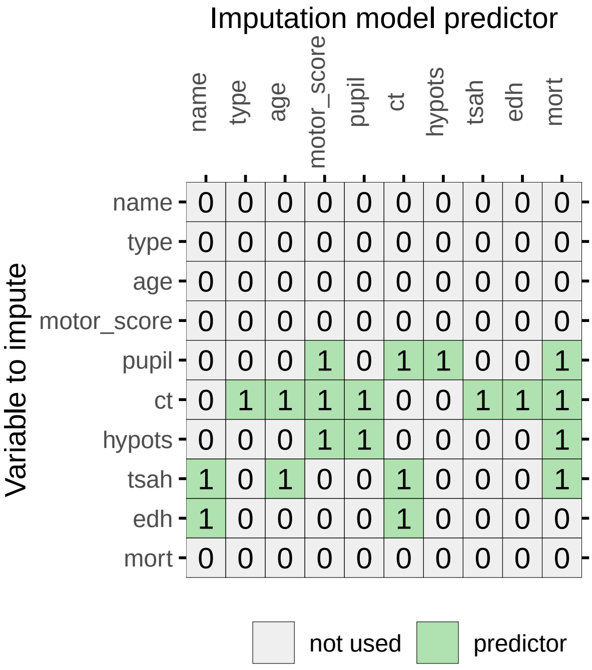
3.3. Imputation model

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

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

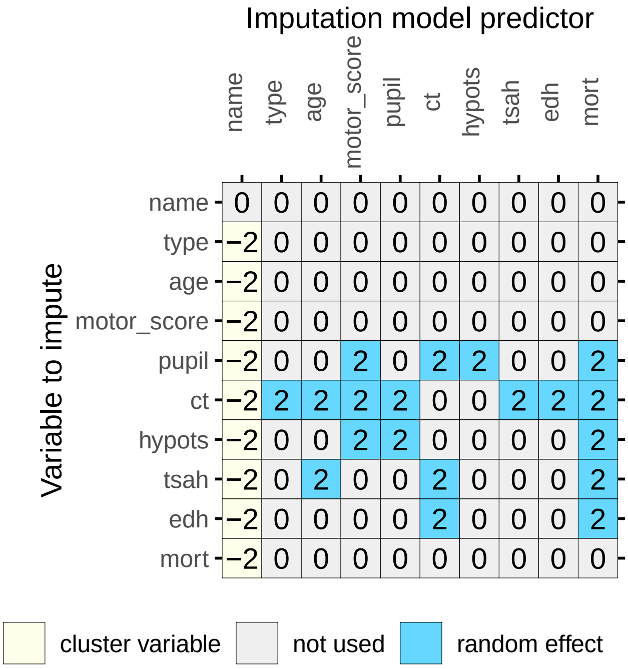
Create a methods vector and predictor matrix, and make sure `name` is not included as predictor, but as clustering variable:

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



	name	type	age	motor_score	pupil	ct	hypots	tsah	edh	mort
name	0	0	0	0	0	0	0	0	0	0
type	0	0	0	0	0	0	0	0	0	0
age	0	0	0	0	0	0	0	0	0	0
motor_score	0	0	0	0	0	0	0	0	0	0
pupil	0	0	0	1	0	1	1	0	0	1
ct	0	1	1	1	1	0	0	1	1	1
hypots	0	0	0	1	1	0	0	0	0	1
tsah	1	0	1	0	0	1	0	0	0	1
edh	1	0	0	0	0	1	0	0	0	0
mort	0	0	0	0	0	0	0	0	0	0

```
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)
```



	name	type	age	motor_score	pupil	ct	hypots	tsah	edh	mort
name	0	0	0	0	0	0	0	0	0	0
type	−2	0	0	0	0	0	0	0	0	0
age	−2	0	0	0	0	0	0	0	0	0
motor_score	−2	0	0	0	0	0	0	0	0	0
pupil	−2	0	0	2	0	2	2	0	0	2
ct	−2	2	2	2	2	0	0	2	2	2
hypots	−2	0	0	2	2	0	0	0	0	2
tsah	−2	0	2	0	0	2	0	0	0	2
edh	−2	0	0	0	0	2	0	0	0	2
mort	−2	0	0	0	0	0	0	0	0	0

```
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) -2.35199614 0.340174191  -6.914093 4.993068e-12
2                    type -0.41267936 0.180271952  -2.289204 2.208652e-02
3                     age  0.03049024 0.001570154  19.418637 1.233010e-81
4 as.factor(motor_score)2 -0.66765325 0.068737838  -9.713038 3.478858e-22
5 as.factor(motor_score)3 -1.05520488 0.070218678 -15.027410 2.536132e-50
6 as.factor(motor_score)4 -1.51238928 0.072304221 -20.917026 1.857373e-90
7                   pupil  0.48421506 0.038982924  12.421209 6.781796e-17
8                      ct  0.43474556 0.029968441  14.506779 2.341875e-36
             b          df dfcom         fmi      lambda m         riv
```

```
1 5.301620e-04 10113.60453 11013 0.005694385 0.005497777 5 0.005528170
2 4.905995e-05 10892.82806 11013 0.001994781 0.001811557 5 0.001814845
3 3.333628e-08  6323.59900 11013 0.016537098 0.016226102 5 0.016493731
4 4.194026e-05  8322.17049 11013 0.010889419 0.010651742 5 0.010766424
5 5.136864e-05  7631.13821 11013 0.012760548 0.012501842 5 0.012660117
6 1.403437e-04  2830.62134 11013 0.032897234 0.032214161 5 0.033286456
7 3.571667e-04    49.96887 11013 0.309144148 0.282035215 5 0.392825972
8 8.586575e-05   294.70102 11013 0.120676307 0.114728924 5 0.129597506
          ubar
1 1.150823e-01
2 3.243910e-02
3 2.425379e-06
4 4.674562e-03
5 4.869020e-03
6 5.059488e-03
7 1.091068e-03
8 7.950686e-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
              -2.37194                -0.41015                 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
               -2.32327                 -0.42365                  0.03023
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
               -0.67143                 -1.05776                 -1.51039
                  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
               -2.33576                  -0.40872                  0.03039
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
               -0.66477                  -1.05453                 -1.51860
                  pupil                        ct
                0.45928                   0.44419
optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 1 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
               -2.35183                  -0.40576                  0.03064
as.factor(motor_score)2  as.factor(motor_score)3  as.factor(motor_score)4
               -0.67469                  -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

Data are simulated and included in the micehecknan package. We will use 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,
- ‘rt“ Response time in minutes (inclusion-restriction variable).

The imputation of these data is based on the IPDMA Heckman Github repo Load the data:

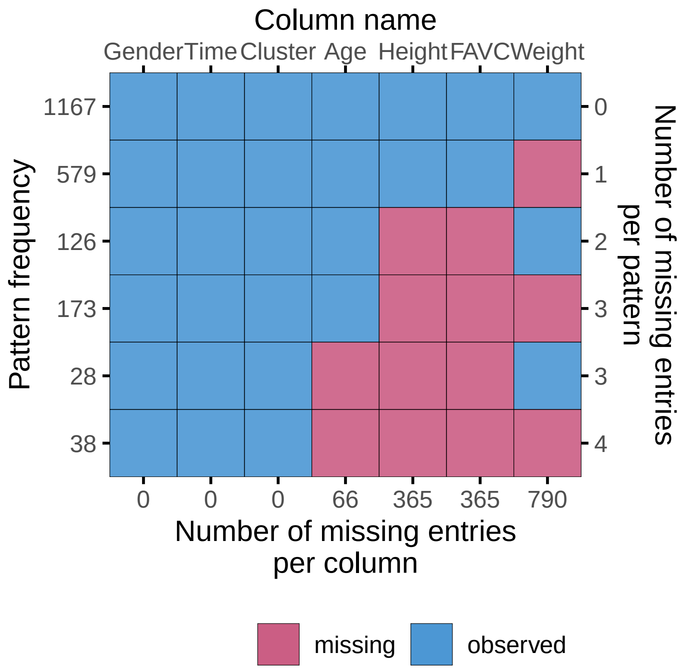
```
R> data("obesity", package = "miceheckman")
```

Select relevant variables:

```
R> dat <- obesity
```

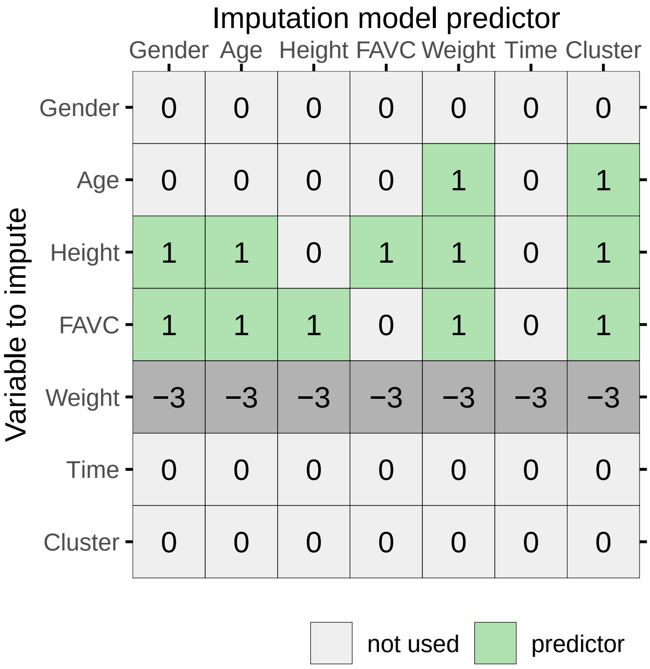
Visualize missing data pattern:

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



Create predictor matrix:

```
R> pred <- quickpred(dat)   # predictor matrix
R> pred["Weight","Cluster"]<- -2 # clustering variable
R> pred["Weight", ] <- -3 #inclusion-restriction variable
R> plot_pred(pred)
```



Imputation model predictor
Variable to impute
🟩 predictor
not used

	Gender	Age	Height	FAVC	Weight	Time	Cluster
Gender	0	0	0	0	0	0	0
Age	0	0	0	0	1	0	1
Height	1	1	0	1	1	0	1
FAVC	1	1	1	0	1	0	1
Weight	−3	−3	−3	−3	−3	−3	−3
Time	0	0	0	0	0	0	0
Cluster	0	0	0	0	0	0	0

Set the Heckman model as imputation method:

```
R> meth <- make.method(dat) # methods vector
R> meth["weight"]<-"2l.heckman"
```

Impute the missingness:

```
R> imp <- mice(
+   dat, # dataset with missing values
+   m = 5, # number of imputations
+   maxit = 10,
+   meth = meth, #imputation method vector
+   pred = pred, #imputation predictors matrix
+   meta_method = "reml",
+   printFlag = FALSE
+ )
```

5. Discussion

- 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/)

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 modelled by a random effect so the hierarchical model can be estimated even when there are few observations per cluster. However, this strategy is best suited for balanced data (Grund, 2017) and when 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

de Jong VMT, Moons KGM, Eijkemans MJC, Riley RD, Debray TPA (2021). "Developing More Generalizable Prediction Models from Pooled Studies and Large Clustered Data Sets." Statistics in Medicine, **40**(15), 3533–3559. ISSN 1097-0258. doi:10.1002/sim.8981.

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.

Gelman A, Hill J (2006). Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press. ISBN 978-1-139-46093-4.

Grund S, Lüdtke O, Robitzsch A (2018). "Multiple Imputation of Missing Data for Multilevel Models: Simulations and Recommendations." Organizational Research Methods, **21**(1), 111–149. ISSN 1094-4281. doi:10.1177/1094428117703686.

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

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

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

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