Missing Data 2

MSBBSS01: Survey data analysis - SGG C128

Stef van Buuren, Gerko Vink

Nov 20, 2023

Generating imputations, multivariate

Workflow after generating imputation

Special topic 1: Practicalities

Special topic 2: Multilevel data

Wrap up

Schedule

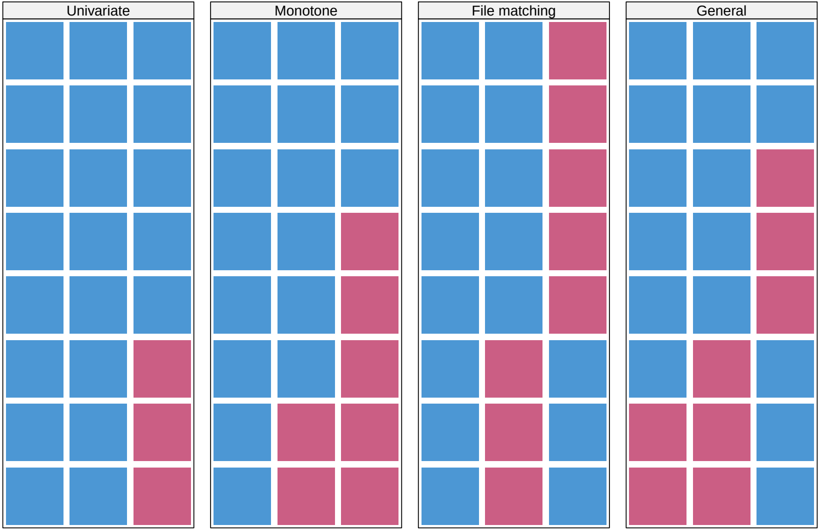
Slot	Time	What	Topic
A	16.30-17.30	L	Generating imputations
	17.30-17.45		COFFEE/TEA
B	17.45-18.15	L	Workflows, special topics
C	18.15-19.00	P	Three vignettes

Generating imputations, multivariate

Issues in multivariate imputation

- The predictors Y_{-j} themselves can contain missing values;
- "Circular" dependence can occur, where Y_j^{mis} depends on Y_h^{mis}, and vice versa;
- Variables are often of different types (e.g., binary, unordered, ordered, continuous);
- Especially with large p and small n, collinearity or empty cells can occur;
- The ordering of the rows and columns can be meaningful, e.g., as in longitudinal data;
- The relation between Y_j and predictors Y_{-j} can be complex, e.g., nonlinear, or subject to censoring processes;
- Imputation can create impossible combinations, such as pregnant grandfathers.

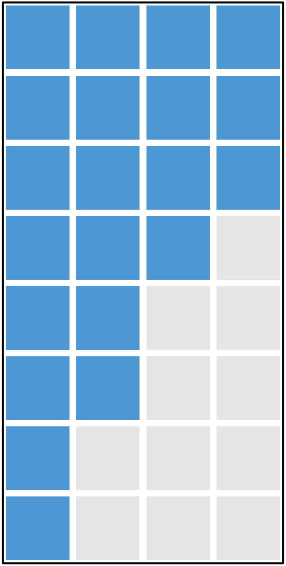
Missing data patterns



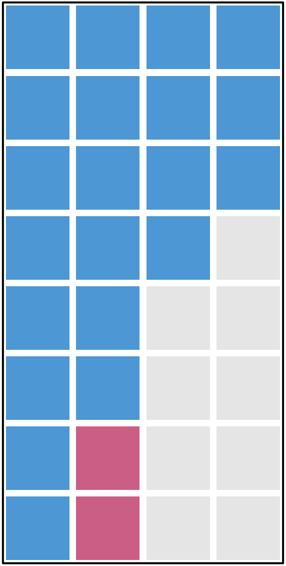
Three general strategies

- Monotone data imputation
- Joint modeling
- Fully conditional specification (FCS)

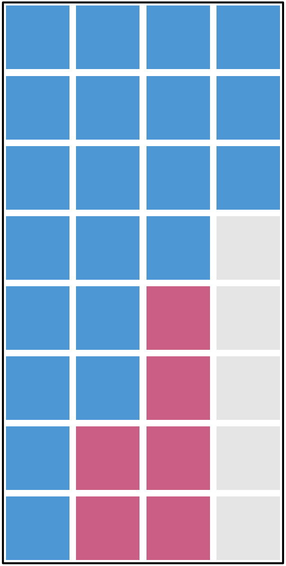
Imputation of monotone pattern



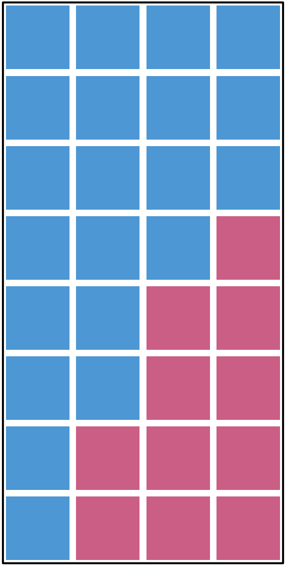
Imputation of monotone pattern



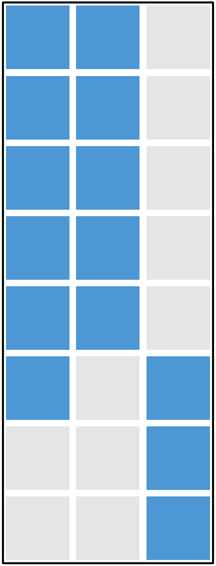
Imputation of monotone pattern



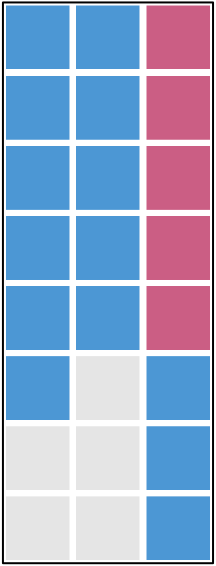
Imputation of monotone pattern



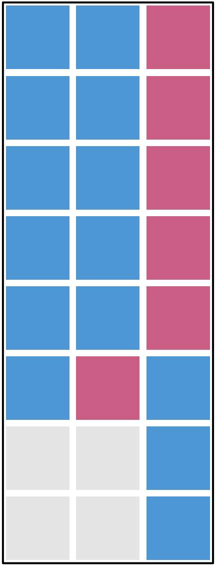
Imputation by joint modelling



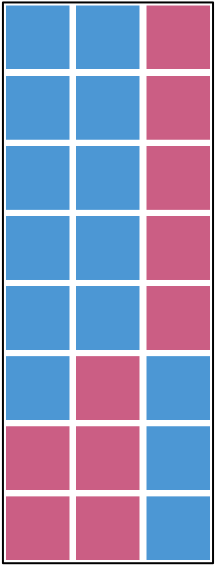
Imputation by joint modelling



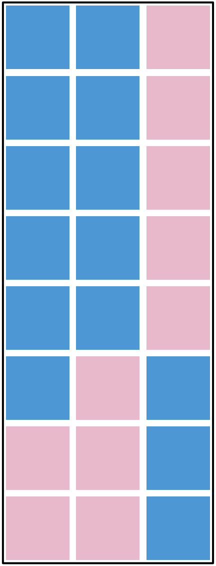
Imputation by joint modelling



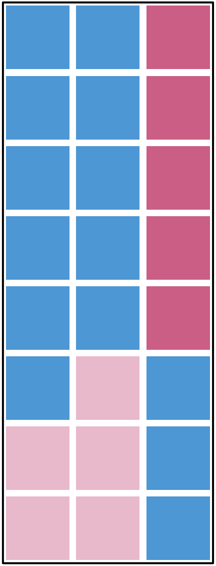
Imputation by joint modelling



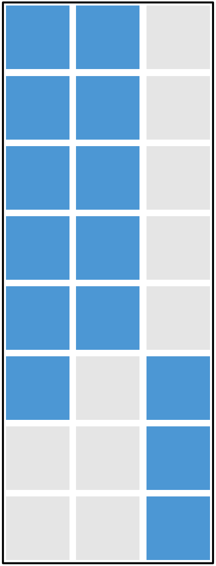
Imputation by joint modelling - next iteration



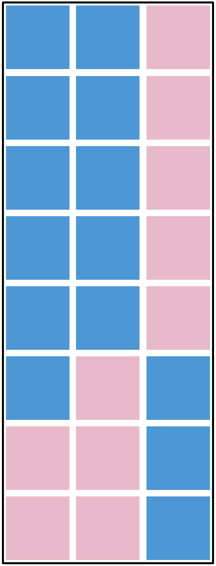
Imputation by joint modelling - next iteration



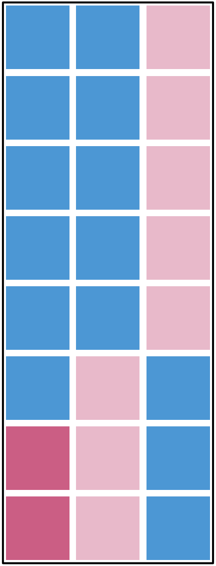
Imputation by fully conditional specification



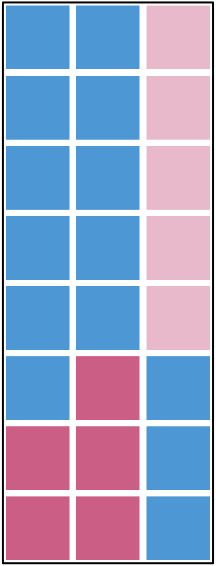
Imputation by fully conditional specification



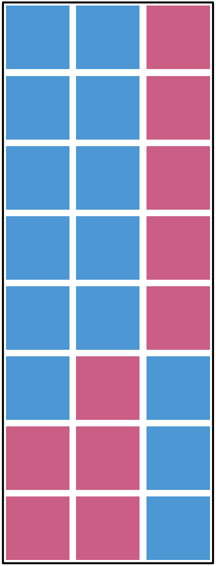
Imputation by fully conditional specification



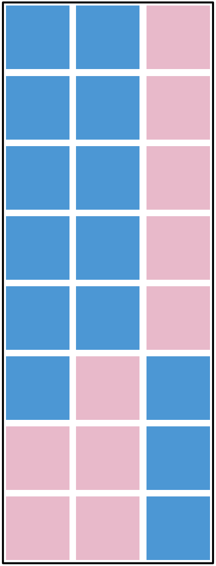
Imputation by fully conditional specification



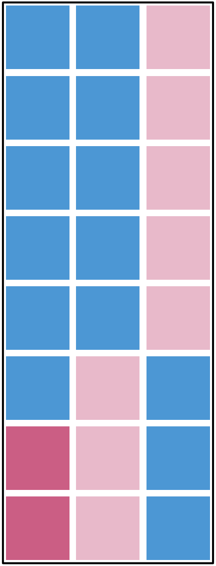
Imputation by fully conditional specification



Imputation by fully conditional specification - next iteration



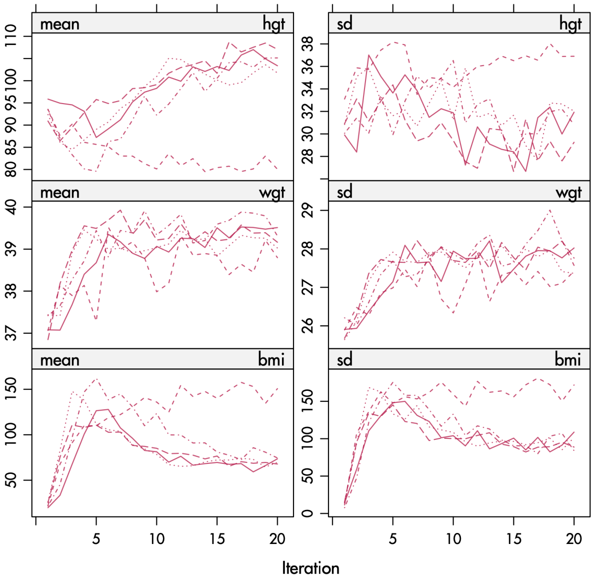
Imputation by fully conditional specification - next iteration



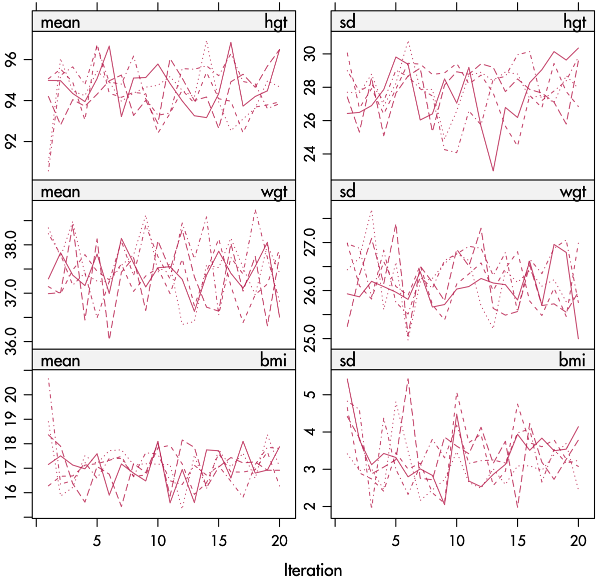
How many iterations?

- Quick convergence
- 5–10 iterations is adequate for most problems
- More iterations is λ is high
- Inspect the generated imputations
- Monitor convergence to detect anomalies

Non-convergence



Convergence



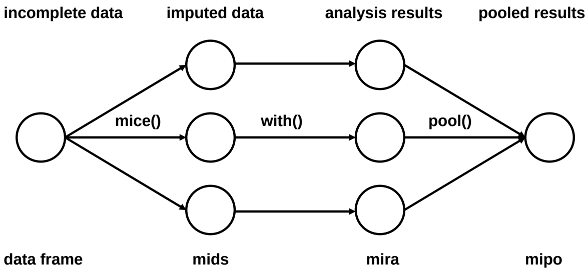
Number of iterations

Watch out for situations where

- the correlations between the Y_j's are high;
- the missing data rates are high; or
- constraints on parameters across different variables exist.

Workflow after generating imputation

Multiple imputation in `mice`



Workflow 1: `mids` workflow using saved objects

```r
# mids workflow using saved objects
library(mice)
imp <- mice(nhanes, seed = 123, print = FALSE)
fit <- with(imp, lm(chl ~ age + bmi + hyp))
est1 <- pool(fit)
```

Workflow 2: `mids` workflow using pipes

```
# mids workflow using pipes
library(magrittr)
est2 <- nhanes %>%
  mice(seed = 123, print = FALSE) %>%
  with(lm(chl ~ age + bmi + hyp)) %>%
  pool()
```

Workflow3: mild workflow using base::lapply

```r
# mild workflow using base::lapply
est3 <- nhanes %>%
  mice(seed = 123, print = FALSE) %>%
  mice::complete("all") %>%
  lapply(lm, formula = chl ~ age + bmi + hyp) %>%
  pool()
```

Workflow4: mild workflow using pipes and base::Map

```
# mild workflow using pipes and base::Map
est4 <- nhanes %>%
  mice(seed = 123, print = FALSE) %>%
  mice::complete("all") %>%
  Map(f = lm, MoreArgs = list(f = chl ~ age + bmi + hyp)) %
  pool()
```

Workflow5: `mild` workflow using `purrr::map`

```
# mild workflow using purrr::map
library(purrr)
est5 <- nhanes %>%
  mice(seed = 123, print = FALSE) %>%
  mice::complete("all") %>%
  map(lm, formula = chl ~ age + bmi + hyp) %>%
  pool()
```

Workflow6: long workflow using base::by

```
# long workflow using base::by
est6 <- nhanes %>%
  mice(seed = 123, print = FALSE) %>%
  mice::complete("long")  %>%
  by(as.factor(.$.imp), lm, formula = chl ~ age + bmi + hyp
  pool()
```

Workflow7: long workflow using a dplyr list-column

```
# long workflow using a dplyr list-column
library(dplyr)
est7 <- nhanes %>%
  mice(seed = 123, print = FALSE) %>%
  mice::complete("long") %>%
  group_by(.imp) %>%
  do(model = lm(formula = chl ~ age + bmi + hyp, data = .))
  as.list() %>%
  .[[-1]] %>%
  pool()
```

Special topic 1: Practicalities

How to set up the imputation model

1. MAR or MNAR
2. Form of the imputation model
3. Which predictors
4. Derived variables
5. What is m?
6. Order of imputation
7. Diagnostics, convergence

Which predictors?

- Include all variables that appear in the complete-data model, including transformations and interactions
- Include the variables that are related to the nonresponse
- Include variables that explain a considerable amount of variance
- Remove variables that have too many missing values within the subgroup of incomplete cases

Functions `mice::quickpred()` and `mice::flux()`

Derived variables

- ratio of two variables
- sum score
- index variable
- quadratic relations
- interaction term
- conditional imputation
- compositions

Derived variables: summary

- Derived variables pose special challenges
- Plausible values should respect data dependencies
- If you can, create derived variables after imputation
- Best option: Probably model-based imputation
- More work needed to verify

Special topic 2: Multilevel data

Imputation of multilevel data

- Avoid multilevel imputation . . . if you can
- Considerably more complex than *flat-file* imputation
- One of the hot spots in statistical technology
- Standard multilevel model does not deal with missing predictors
- Know the complete-data statistical analysis

brandsma data

- Brandsma and Knuver, Int J Ed Res, 1989.
- Extensively discussed in Snijders and Bosker (2012), 2nd ed.
- 4106 pupils, 216 schools, about 4% missing values

```
library(mice)
head(brandsma[, c(1:6, 9:10, 13)], 3)
```

```
##   sch pup   iqv   iqp sex    ses lpr lpo den
## 1   1   1 -1.35 -3.72   1 -17.67  33  NA   1
## 2   1   2  2.15  3.28   1     NA  44  50   1
## 3   1   3  3.15  1.27   0  -4.67  36  46   1
```

brandsma data subset

```r
d <- brandsma[, c("sch", "lpo", "sex", "den")]
head(d, 2)
```

```
##   sch lpo sex den
## 1   1  NA   1   1
## 2   1  50   1   1
```

- `sch`: School number, cluster variable, $C = 216$;
- `lpo`: Language test post, outcome at pupil level;
- `sex`: Sex of pupil, predictor at pupil level (0-1);
- `den`: School denomination, predictor at school level (1-4).

Model of scientific interest

Predict `lpo` from the

- level-1 predictor `sex`
- level-2 predictor `den`

Level notation - Bryk and Raudenbush (1992)

$$\texttt{lpo}_{ic} = \beta_{0c} + \beta_{1c}\texttt{sex}_{ic} + \epsilon_{ic} \tag{1}$$

$$\beta_{0c} = \gamma_{00} + \gamma_{01}\texttt{den}_c + u_{0c} \tag{2}$$

$$\beta_{1c} = \gamma_{10} \tag{3}$$

- lpo_{ic} is the test score of pupil i in school c
- sex_{ic} is the sex of pupil i in school c
- den_c is the religious denomination of school c
- β_{0c} is a random intercept that varies by cluster
- β_{1c} is a sex effect, assumed to be the same across schools.
- $\epsilon_{ic} \sim N(0, \sigma_\epsilon^2)$ is the within-cluster random residual at the pupil level

Level 2 equations: interpretation

The first level-2 model

$$\beta_{0c} = \gamma_{00} + \gamma_{01}\mathsf{den}_c + u_{0c},$$

describes the variation in the mean test score between schools as a function of

- the grand mean γ_{00},
- a school-level effect γ_{01} of denomination, and a
- school-level random residual $u_{0c} \sim N(0, \sigma^2_{u_0})$

The second level 2 model

$$\beta_{1c} = \gamma_{10},$$

specifies β_{1c} as a fixed effect equal in value to γ_{10}

Unknown parameters

$$\texttt{lpo}_{ic} = \beta_{0c} + \beta_{1c}\texttt{sex}_{ic} + \epsilon_{ic} \tag{4}$$
$$\beta_{0c} = \gamma_{00} + \gamma_{01}\texttt{den}_c + u_{0c} \tag{5}$$
$$\beta_{1c} = \gamma_{10} \tag{6}$$

The unknowns to be estimated are the fixed parameters:

- γ_{00},
- γ_{01}, and
- γ_{10},

and the variance components:

- σ^2_ϵ and
- $\sigma^2_{u_0}$.

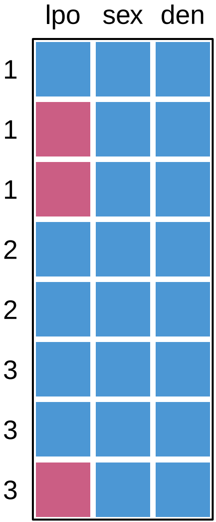
Where are the missings?

In single level data, missingness may be in the outcome and/or in the predictors

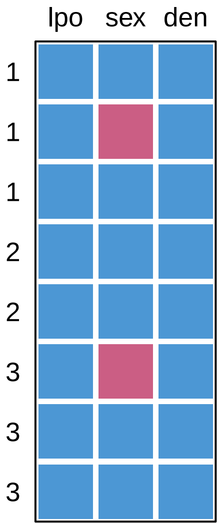
With multilevel data, missingness may be in:

1. the outcome variable;
2. the level-1 predictors;
3. the level-2 predictors;
4. the class variable.

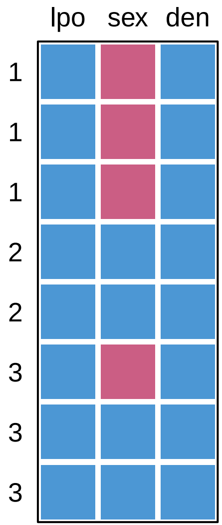
Univariate missing, level-1 outcome



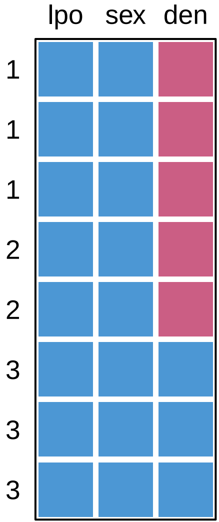
Univariate missing, level-1 predictor, sporadically missing



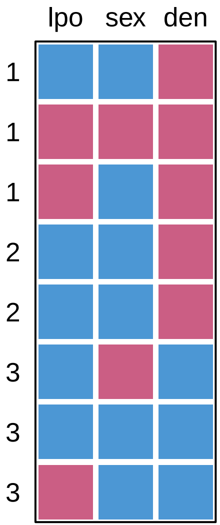
Univariate missing, level-1 predictor, systematically missing



Univariate missing, level-2 predictor



Multivariate missing



Fully conditional specification

$$\dot{\texttt{lpo}}_{ic} \sim N(\beta_0 + \beta_1 \texttt{den}_c + \beta_2 \texttt{sex}_{ic} + u_{0c}, \sigma_\epsilon^2) \quad (7)$$

$$\dot{\texttt{sex}}_{ic} \sim N(\beta_0 + \beta_1 \texttt{den}_c + \beta_2 \texttt{lpo}_{ic} + u_{0c}, \sigma_\epsilon^2) \quad (8)$$

Theoretical problem with FCS

Conditional expectation of $\texttt{sex}_{ic}$ in a random effects model depends on

- $\texttt{lpo}_{ic}$,
- $\overline{\texttt{lpo}}_i$, the mean of cluster i, and
- n_i, the size of cluster i.

Resche-Rigon & White (2018) suggest the imputation model

- should incorporate the cluster means of level-1 predictors
- be heteroscedastic if cluster sizes vary

Methods for multilevel imputation in mice

Table 7.2: Overview of methods to perform univariate multilevel imputation of continuous data. Each of the methods is available as a function called `mice.impute.[method]` in the specified `R` package.

Package	Method	Description
Continuous		
`mice`	`2l.lmer`	normal, `lmer`
`mice`	`2l.pan`	normal, `pan`
`miceadds`	`2l.continuous`	normal, `lmer` , `blme`
`micemd`	`2l.jomo`	normal, `jomo`
`micemd`	`2l.glm.norm`	normal, `lmer`
`mice`	`2l.norm`	normal, heteroscedastic
`micemd`	`2l.2stage.norm`	normal, heteroscedastic
Generic		
`miceadds`	`2l.pmm`	pmm, homoscedastic, `lmer`
`micemd`	`2l.2stage.pmm`	pmm, heteroscedastic, `mvmeta`

Methods for multilevel imputation in mice

Table 7.3: Methods to perform univariate multilevel imputation of missing discrete outcomes. Each of the methods is available as a function called `mice.impute.[method]` in the specified `R` package.

Package	Method	Description
Binary		
`mice`	`2l.bin`	logistic, `glmer`
`miceadds`	`2l.binary`	logistic, `glmer`
`micemd`	`2l.2stage.bin`	logistic, `mvmeta`
`micemd`	`2l.glm.bin`	logistic, `glmer`
Count		
`micemd`	`2l.2stage.pois`	Poisson, `mvmeta`
`micemd`	`2l.glm.pois`	Poisson, `glmer`
`countimp`	`2l.poisson`	Poisson, `glmmPQL`
`countimp`	`2l.nb2`	negative binomial, `glmmadmb`
`countimp`	`2l.zihnb`	zero-infl neg bin, `glmmadmb`

Methods for multilevel imputation in `mice`

Table 7.4: Overview of `mice.impute.[method]` functions to perform univariate multilevel imputation.

Package	Method	Description
Level-2		
`mice`	`2lonly.mean`	level-2 manifest class mean
`miceadds`	`2l.groupmean`	level-2 manifest class mean
`miceadds`	`2l.latentgroupmean`	level-2 latent class mean
`mice`	`2lonly.norm`	level-2 class normal
`mice`	`2lonly.pmm`	level-2 class pmm
`miceadds`	`2lonly.function`	level-2 class, generic
`miceadds`	`ml.lmer`	≥ 2 levels, generic

Wrap up

Summary

- Impact of missing data
- Ad-hoc techniques
- Theory of multiple imputation
- Generating imputations
- Workflows
- Specification of imputation model
- Multilevel data