Journal of Statistical Software

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

Imputation of Incomplete Multilevel Data with R

Hanne I. Oberman
Utrecht University

Johanna Muñoz
University Medical Center Utrecht

Valentijn M.T. de Jong
University Medical Center Utrecht

Gerko Vink
University Medical Center Utrecht

Thomas P.A. Debray
University Medical Center Utrecht

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. Incomplete multilevel data requires careful consideration of the missing data problem and analysis strategy. In this tutorial, we focus on a popular strategy for accommodating missingness in multilevel data: replacing the missing data with one or more plausible values, i.e., imputation.Imputation separates the missing data problem from the main analysis and the completed data can be analyzed as if it has been fully observed. This tutorial illustrates the imputation of incomplete multilevel data with the statistical programming language R. We aim to show how imputation can yield less biased estimates from incomplete clustered data. We provide practical guidelines and code snippets for different missing data situations, including non-ignorable missingness mechanisms. For brevity, we focus on multilevel imputation using chained equations with the R mice package and its adjacent packages.

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

1. Introduction: Clustering and incomplete data

1. missing data occur often in data with human subjects

2. missing data may be resolved, but need to be handled in accordance with the analysis of scientific interest
3. in human-subjects research, there is often clustering, which may be captured with multilevel modeling techniques
4. if the analysis of scientific interest is a multilevel model, the missing data handling method should accommodate the multilevel structure of the data
5. both missingness and multilevel structures require advanced statistical techniques
6. this tutorial sets out to facilitate empirical researchers in accommodating both multilevel structures as well as missing data.
7. we illustrate the use of the software by means of three case studies from the social and biomedical sciences.

1.1. overview of software

The popular **mice** package in R R Core Team (2017)...

1.2. scope

2. Background

2.1. concepts in multilevel data

Box 1. The intraclass correlation coefficient.

In R, multilevel models may be fitted using the package **lme4**. For linear mixed-effects models, the function

```
lmer(formula, data, ...)
```

2.2. concepts in missing data

The R package **mice** provides a framework for imputing incomplete data on a variable-by-variable basis. The `mice()` function allows users to flexibly specify how many times and under what model the missing data should be imputed. This is reflected in the first four function arguments

```
mice(data, m, method, predictorMatrix, ...)
```

where `data` refers to the incomplete dataset, `m` determines the number of imputations, `method` denotes the functional form of the imputation model and `predictorMatrix` specifies the interrelational dependencies between variables and imputation models (i.e., the set of predictors to be used for imputing each incomplete variable).

Box 2. The `methods`.

Box 2. The predictor matrix.

3. Illustrations

In this section, we demonstrate the workflow using three case studies.

3.1. Setup

```
R> set.seed(123)
R> library(mice)
R> library(ggmice)
R> library(ggplot2)
R> library(miceadds)
R> library(lme4)
R> library(mitml)
R> library(broom.mixed)
```

3.2. Popularity data

```
R> data("popmis", package = "mice")
```

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

```
ggmice(dat, aes(popular, teachpop)) +
  geom_jitter()
```

With the `ggmice` unction `plot_pattern` we can visualize this.

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

```
R> plot_corr(dat)
```

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

```
  school teachpop  popular     texp      sex
      ""       ""    "pmm"       ""       ""
```

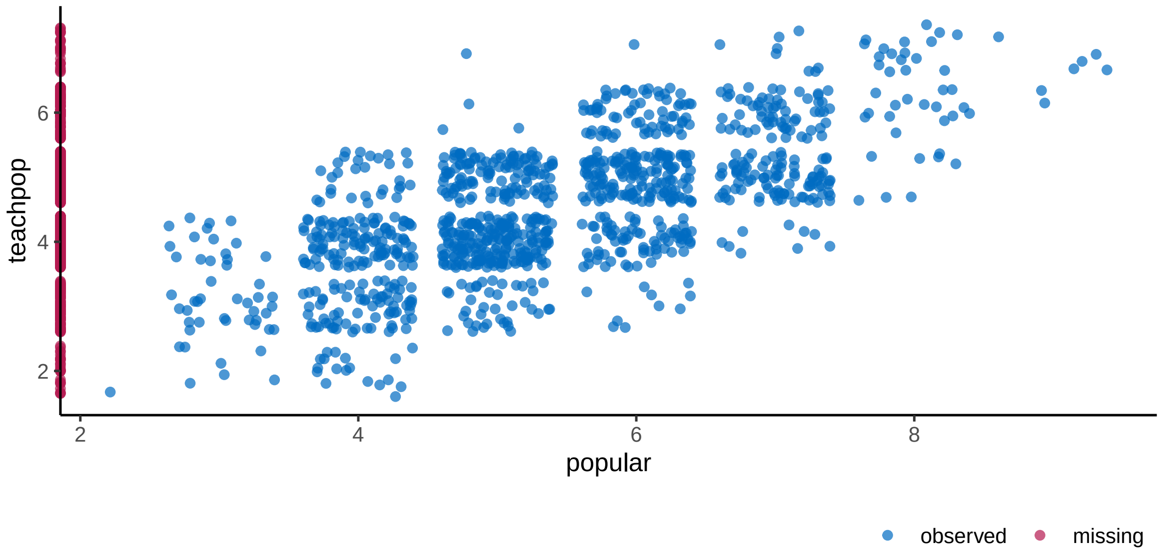


Figure 1: Polar axis plot

```
R> pred <- quickpred(dat)
R> pred
```

```
         school teachpop popular texp sex
school        0        0       0    0   0
teachpop      0        0       0    0   0
popular       0        1       0    1   1
texp          0        0       0    0   0
sex           0        0       0    0   0
```

Adjust the methods vector.

```
R> meth["popular"] <- "2l.pmm"
```

Adjust the predictor matrix.

```
R> pred["popular", "school"] <- -2
R> pred["popular", "sex"] <- 2
```

Visualize the imputation methods and predictors.

```
plot_pred(pred, method = meth)
```

Impute the data.

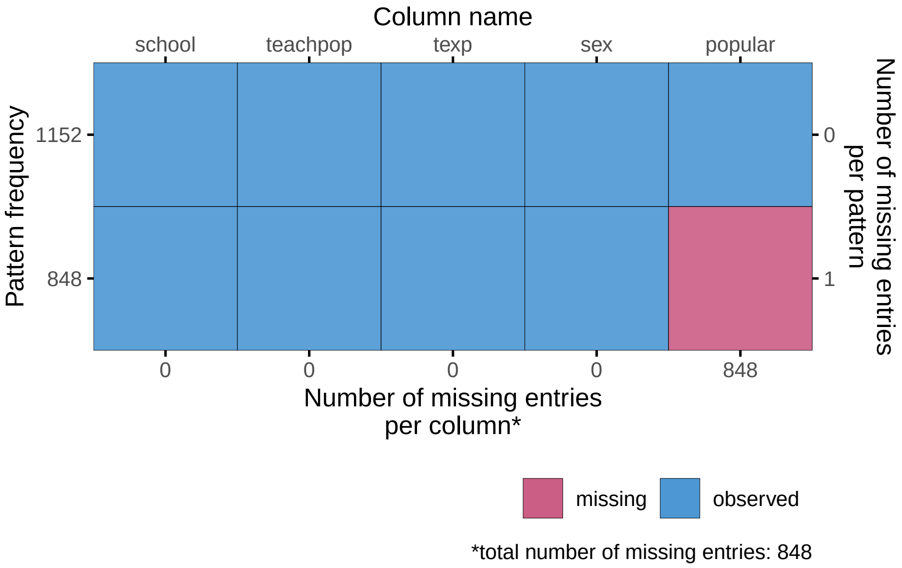


Figure 2: Missing data pattern.

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

Evaluate the convergence.

```
R> plot_trace(imp)
```

Evaluate the distribution of imputed values.

```
ggmice(imp, aes(popular, group = .imp)) +
  geom_density()
```

Evaluate the distribution of imputed values.

```
ggmice(imp, aes(.imp, popular)) +
  geom_jitter(alpha = 0.05) +
    geom_boxplot()


ggmice(imp, aes(popular, teachpop)) +
  geom_jitter() +
  facet_wrap(~ .imp)
```

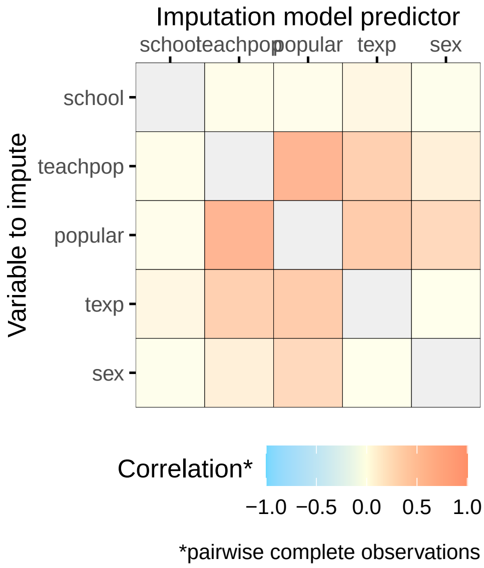


Figure 3: Pair-wise correlations.

Analyze the imputed data.

```
fit <- with(
  imp,
  lmer(teachpop ~ popular + texp + (1 | school))
)
```

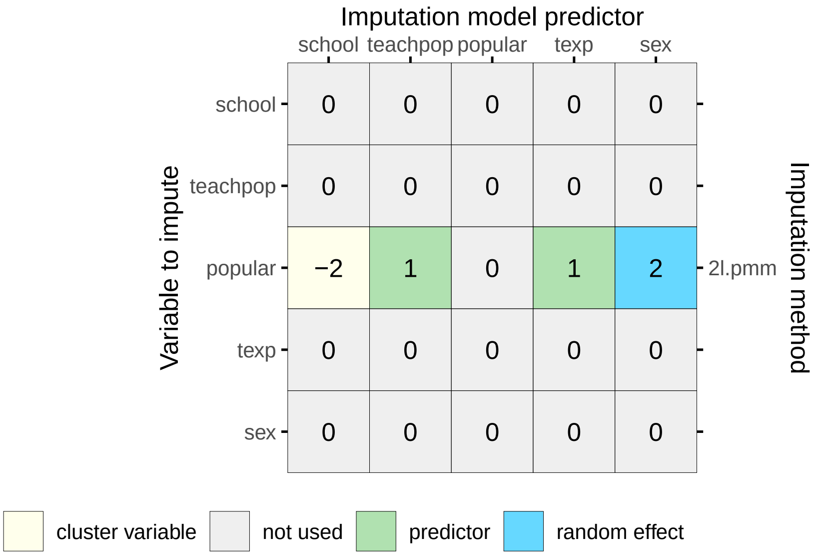
Pool the estimates.

```
pool(fit)
```

```
Class: mipo    m = 5
         term m  estimate         ubar            b            t dfcom
1 (Intercept) 5 2.4091354 2.241304e-02 1.712964e-03 2.446860e-02  1995
2     popular 5 0.2597284 2.353344e-04 1.209648e-04 3.804922e-04  1995
3        texp 5 0.0484727 7.728295e-05 3.236252e-06 8.116646e-05  1995
         df        riv     lambda        fmi
1 432.50579 0.09171257 0.08400798 0.08821454
2  26.88403 0.61681474 0.38149995 0.42289329
3 909.68447 0.05025044 0.04784615 0.04993264
```

Display results in table.

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



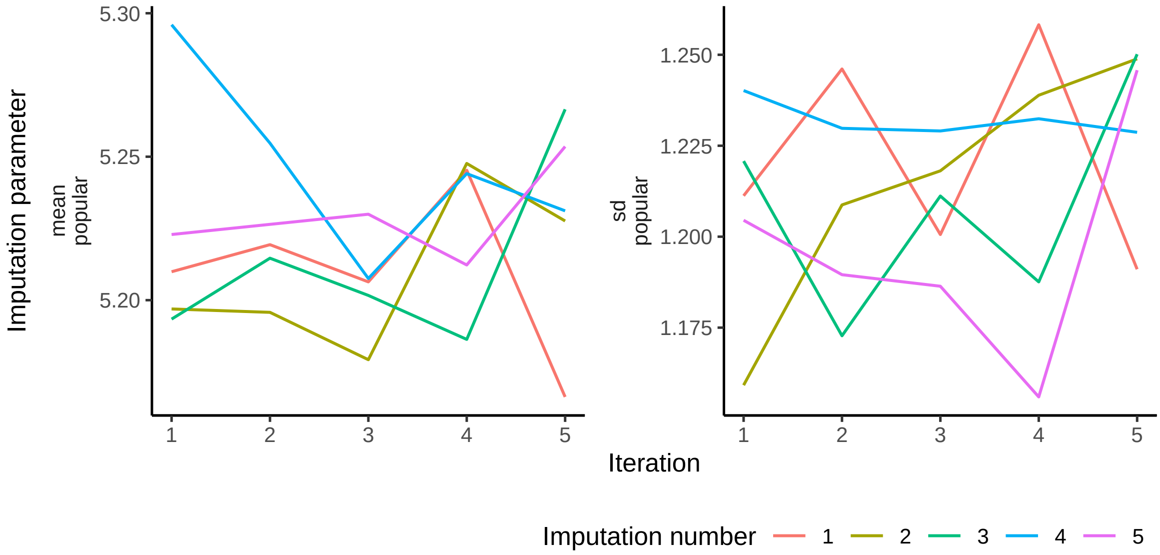

```
Call:

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)     2.409     0.156    15.401   566.786     0.000     0.092     0.087
popular         0.260     0.020    13.315    27.483     0.000     0.617     0.422
texp            0.048     0.009     5.380  1747.294     0.000     0.050     0.049

                            Estimate
Intercept~~Intercept|school    0.310
Residual~~Residual             0.307
ICC|school                     0.502

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

4. Summary and discussion

What is missing from this manuscript...

Computational details

The results in this paper were obtained using R~4.3.0. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at [https://CRAN.R-project.org/].

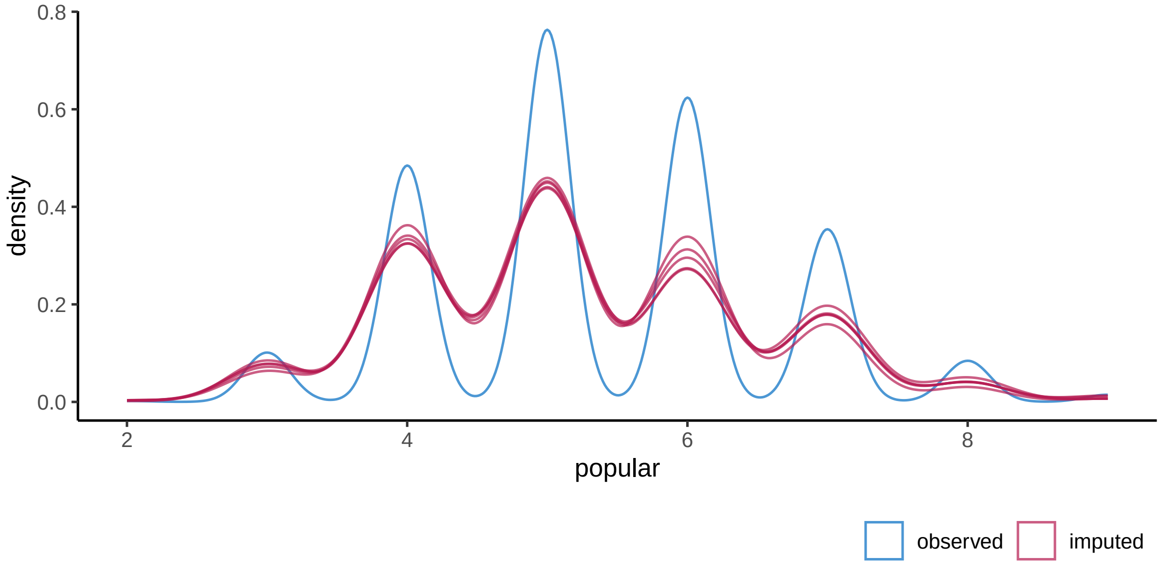


Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under ReCoDID grant agreement No 825746.

References

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.



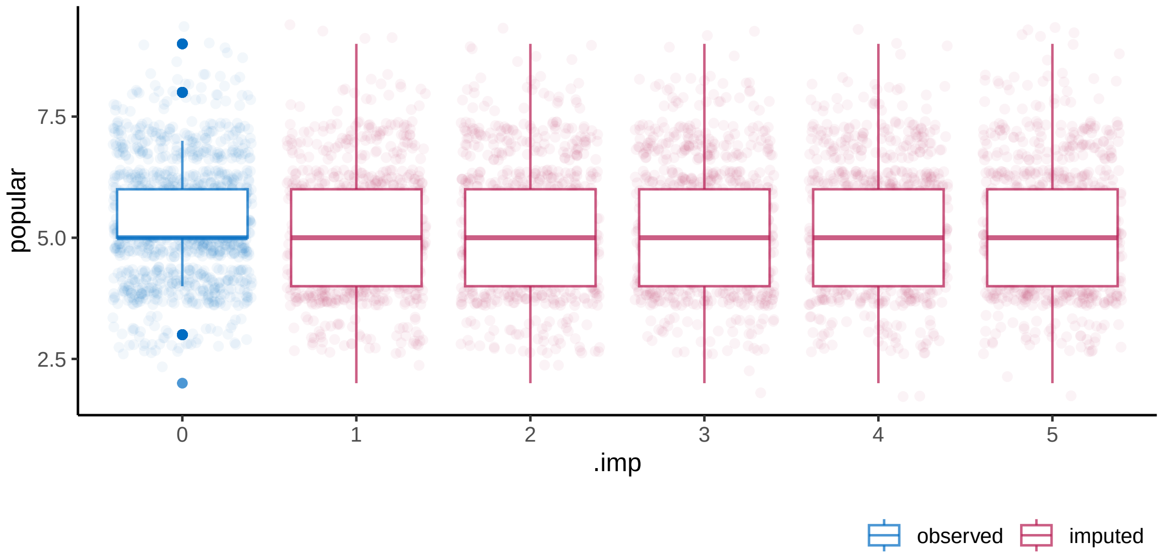
More technical details

Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*).
For more technical style details, please check out JSS's style FAQ at [https://www.jstatsoft.org/pages/view/style#frequently-asked-questions] which includes the following topics:

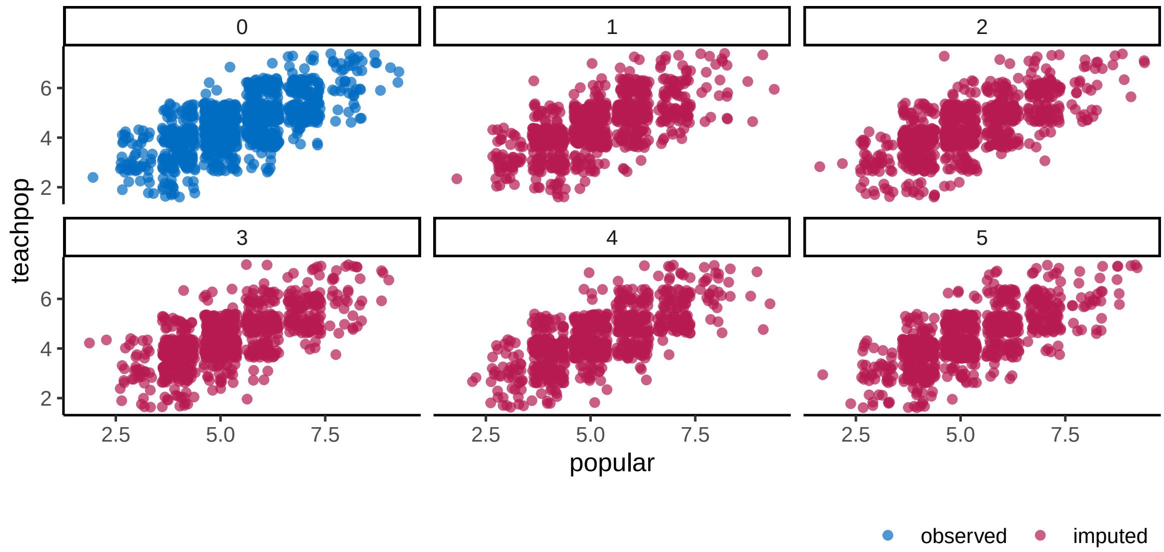
- Title vs. sentence case.
- Graphics formatting.
- Naming conventions.
- Turning JSS manuscripts into R package vignettes.
- Trouble shooting.
- Many other potentially helpful details…

Using BibTeX

References need to be provided in a BIBTEX file (`.bib`). All references should be made with `@cite` syntax. This commands yield different formats of author-year citations and allow to include additional details (e.g.,pages, chapters, …) in brackets. In case you are not familiar with these commands see the JSS style FAQ for details.
Cleaning up BIBTEX files is a somewhat tedious task – especially when acquiring the entries automatically from mixed online sources. However, it is important that informations are complete and presented in a consistent style to avoid confusions. JSS requires the following format.



- item JSS-specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- item Titles should be in title case.
- item Journal titles should not be abbreviated and in title case.
- item DOIs should be included where available.
- item Software should be properly cited as well. For R packages `citation("pkgname")` typically provides a good starting point.



Affiliation:

Hanne I. Oberman
Methodology and Statistics
Padualaan 14
Utrecht The Netherlands
E-mail: h.i.oberman@uu.nl
URL: https://www.hanneoberman.github.io

Johanna Muñoz
Julius Centre for Health Sciences and Primary Care
Universiteitsweg 100
Utrecht The Netherlands

Valentijn M.T. de Jong
Julius Centre for Health Sciences and Primary Care
Utrecht The Netherlands

Gerko Vink
Julius Centre for Health Sciences and Primary Care
Universiteitsweg 100
Utrecht The Netherlands

Thomas P.A. Debray
Julius Centre for Health Sciences and Primary Care
Universiteitsweg 100
Utrecht The Netherlands

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