




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 accommodated, 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.

Warning: package 'ggplot2' was built under R version 4.3.2

1.1. overview of software

The popular **mice** package in R [R Core Team \(2017\)](#)

1.2. scope

This paper 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 (?).

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 ?) 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, ?);
- **impact** from the **metamisc** package (empirical data on traumatic brain injuries, $n = 11,022$ patients across $N = 15$ studies with data that are MAR, ?);
- **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).

2. Background

2.1. concepts in multilevel data

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 (?). 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.

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

missing data mechanisms etc.

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 (??). For example, it is possible

	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					

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. Sporadic missingness is visualized in Figure XYZ.

`plot_na()`

Column X_1 in Figure 1 is completely observed, column X_2 is systematically missing in cluster 2, and column X_3 is sporadically missing. To analyze these incomplete data, we have to take the nature of the missingness and the cluster structure into account. For example, the sporadic missingness in X_3 could be easily amended if this would be a cluster-level variable (and thus constant within clusters). We could then just extrapolate the true (but missing) value of X_3 for unit 1 from unit 2, and the value for unit 4 from unit 3. If X_3 would instead be a unit-level variable (which may vary within clusters), we could not just recover the unobserved ‘truth’, but would need to use some kind of missing data method, or discard the incomplete units altogether (i.e., complete case analysis). Complete case analysis can however introduce bias in statistical inferences and lowers statistical power. Further, with the systematic missingness in X_2 , it would be impossible to fit a multilevel model without accommodating the missingness in some way. Complete case analysis in that case would mean excluding the entire cluster from the analyses. The wrong choice of missing data handling method can thus be extremely harmful to the inferences.

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 (? and ?). We here consider the general case that data are MAR, and expand on certain MNAR situations.

2.3. imputation with mice

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 3. The predictor matrix. The entries corresponding to the level-1 predictors are coded with a 3, indicating that both the original values as well as the cluster means of the predictor are included into the imputation model. The entry of 4 in the predictor matrix adds three variables to the imputation model for the imputation model predictor: the value of the predictor, the cluster means of the predictor and the random slopes of the predictor.

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")]
```

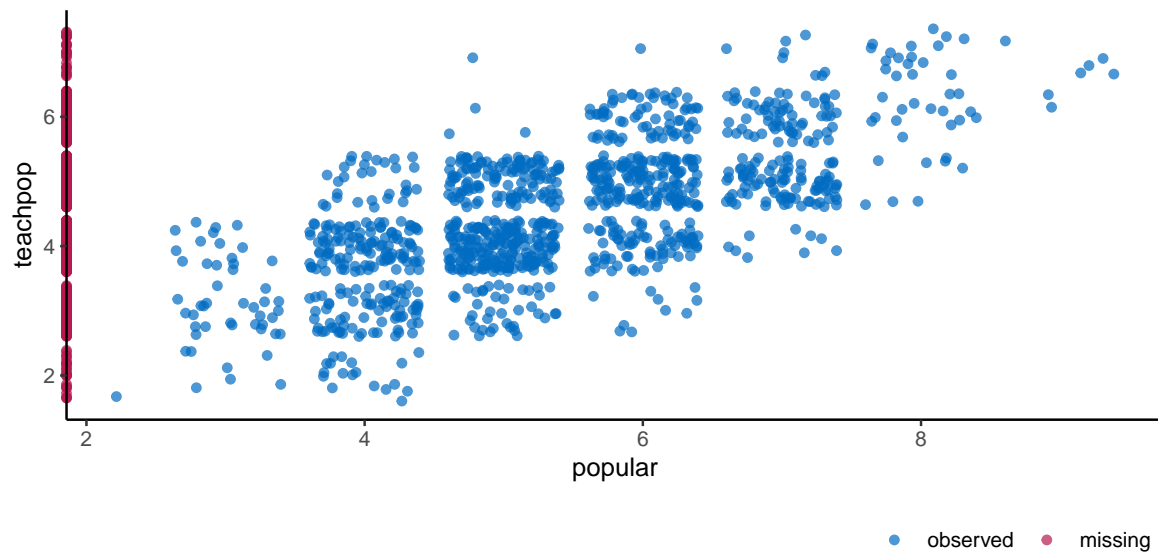


Figure 1: Polar axis plot

```
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"	""	""

```
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

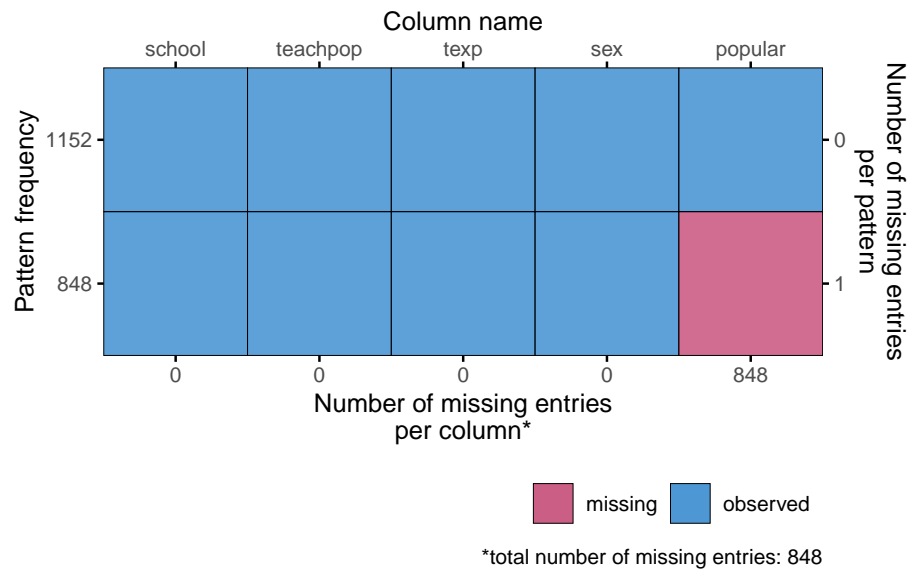


Figure 2: Missing data pattern.

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.

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

Evaluate the convergence.

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

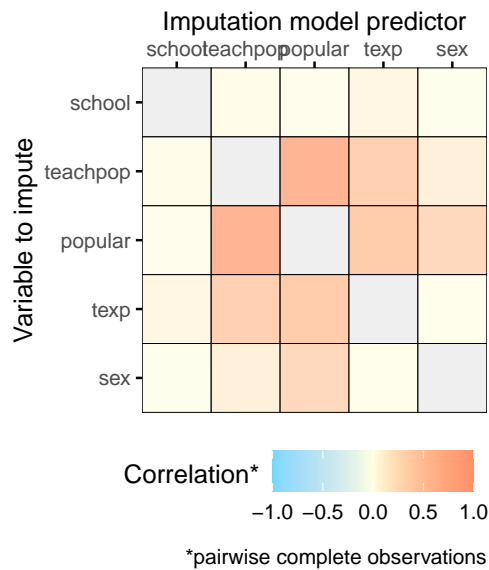


Figure 3: Pair-wise correlations.

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)
```

Analyze the imputed data.

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

Pool the estimates.

```
pool(fit)
```


	Imputation model predictor					
	school	teachpop	popular	texp	sex	
Variable to impute	school	0	0	0	0	Imputation method
	teachpop	0	0	0	0	
	popular	-2	1	0	1	
	texp	0	0	0	0	
	sex	0	0	0	0	

2l.pmm

cluster variable
 not used
 predictor
 random effect

```

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)
```

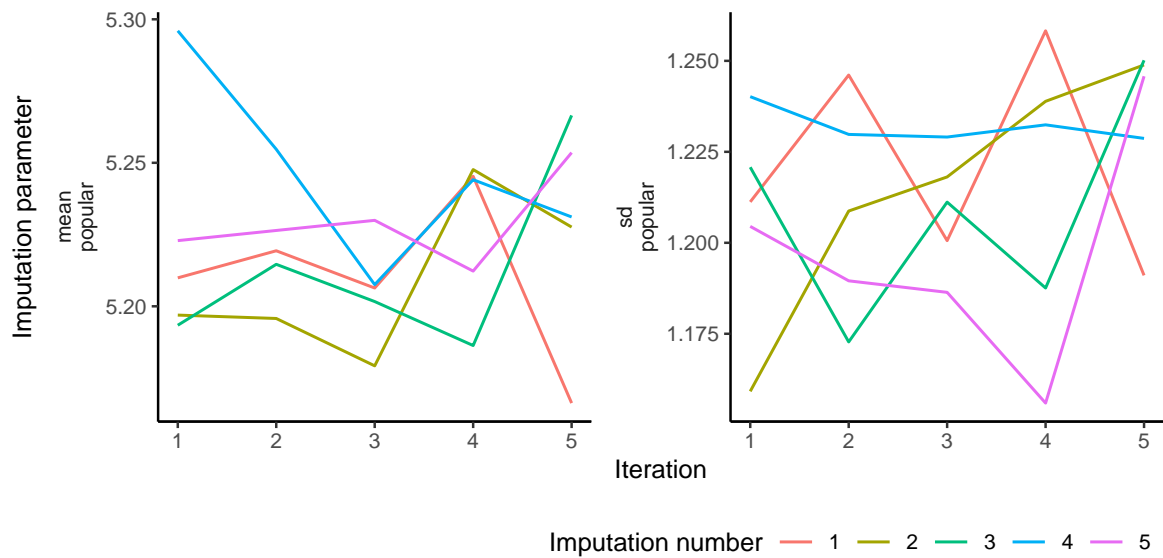
Call:

```
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

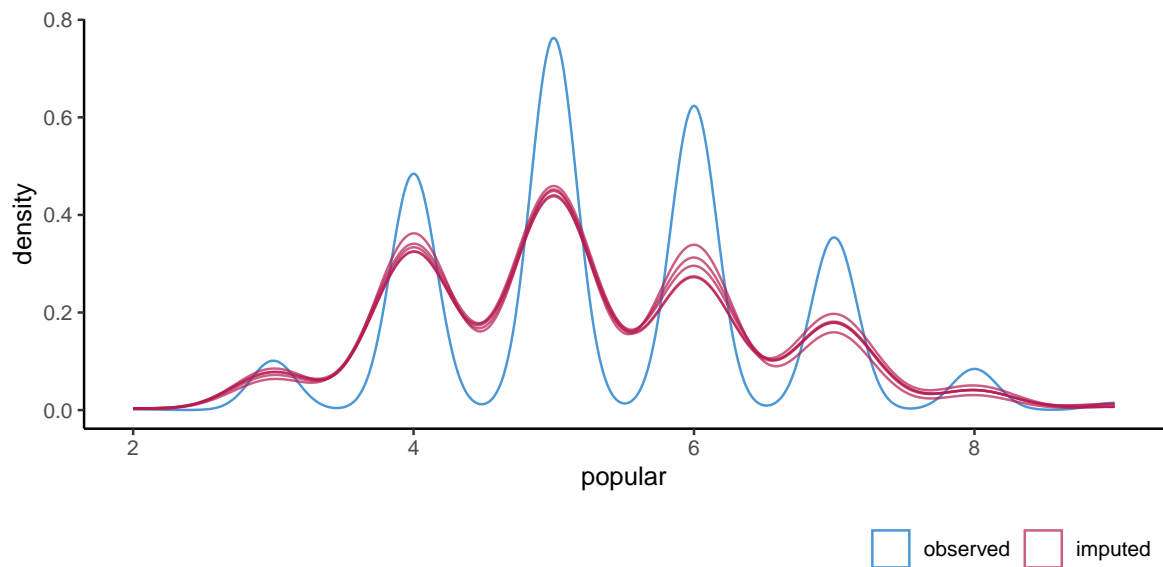
The results in this paper were obtained using R~4.3.0. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at [<https://CRAN.R-project.org/>].

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

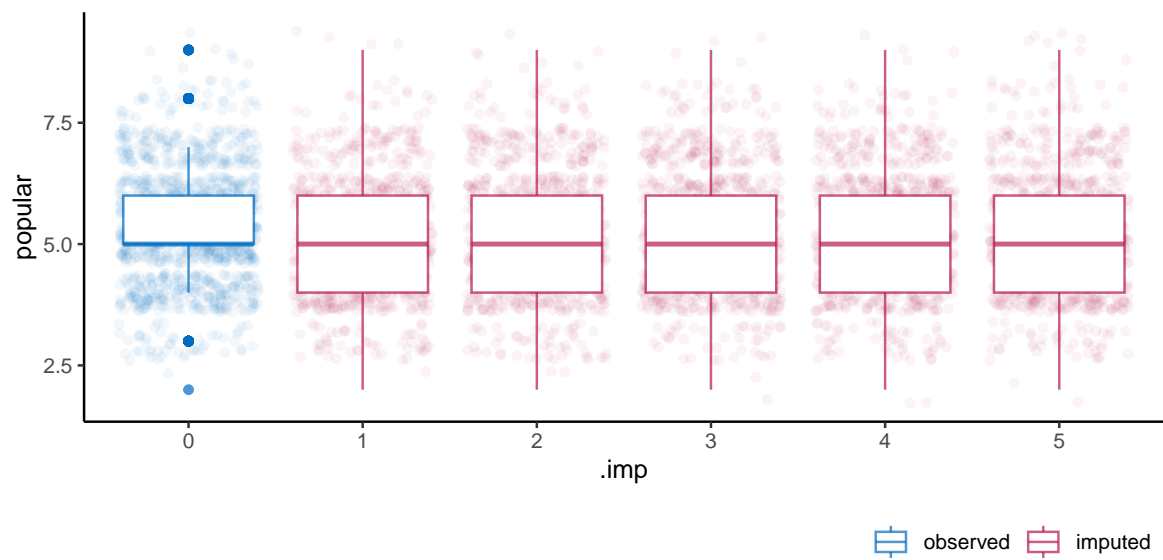
Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*). For more technical style details, please check out JSS's style FAQ at [<https://www.jstatsoft.org/pages/view/style#frequently-asked-questions>]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

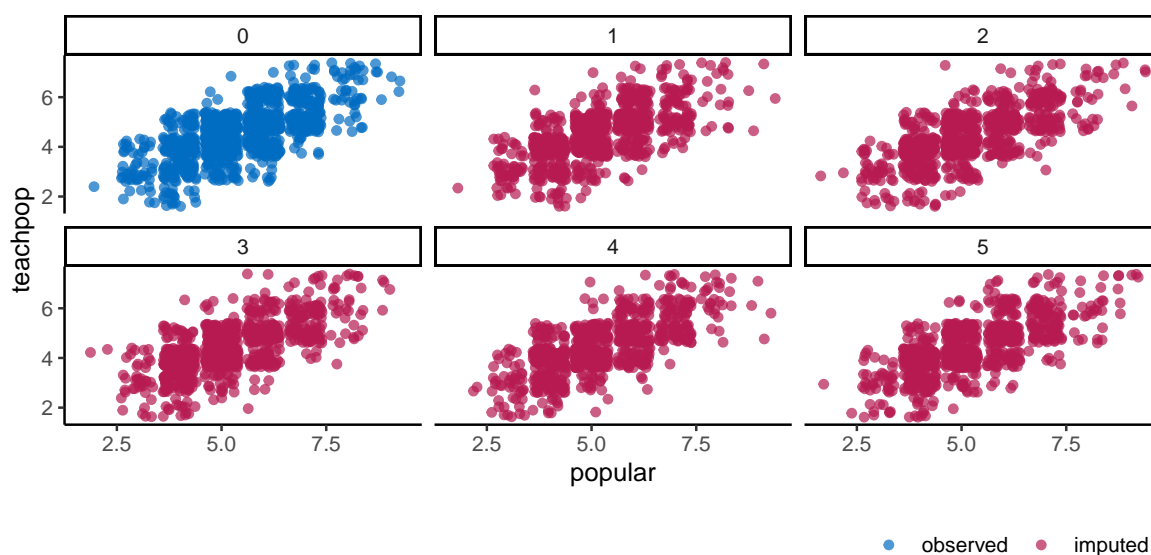
Using BibTeX

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