Journal of Statistical Software

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

A Note on Convergence Diagnostics for Multiple Imputation using Chained Equations (MICE)

Hanne Oberman
Utrecht University

Abstract

This Research Report contains a simulation study that serves as the basis of the technical paper that will be submitted for publication in *Journal of Statistical Software*. I have chosen to use the first format from the Research Report guidelines: "*It is written as a (mini) thesis, with an introduction, methods section, some results (i.e., preliminary analyses, or pilot simulations), and a discussion of results. The length of the research report should be maximally 2500 words of text (without references list and or tables and figures). Please do not include appendices, and no more than 6 tables or figures. Table and Figure captions do not count towards the word limit. An abstract may be included, but is not necessary*". I aim to publish a pre-print of this research report on ArXiv. That way, I can refer to the simulation study described here, without 'bulking up' the technical paper that I want to submit for publication in JSS. Another option would be to attach this note as online appendix to the technical paper on ShinyMICE. **TO ADD: appendix with extra results tables with different ampute and or very high or very low correlations between predictors**.

Keywords: multiple imputation, convergence, **mice**, R.

1. Introduction

At some point, any scientist conducting statistical analyses will run into a missing data problem (Allison 2001). Missingness is problematic because statistical inference cannot be performed on incomplete data without employing *ad hoc* solutions (e.g., listwise deletion), which can yield wildly invalid results (Van Buuren 2018). A popular answer to the ubiquitous problem of missing information is to use the framework of multiple imputation (MI), proposed by Rubin (1987). MI is an iterative algorithmic procedure in which each missing data point is 'guessed' (i.e. imputed) several times. The variablity between imputations is used to reflect how much uncertainty in the inference is introduced by the missingness. Therefore, MI can

provide valid inferences despite of missing information.

ADD some references in this paragraph! To obtain valid inferences with MI, the variability between imputations should be correct. If the variance between imputations is underestimated, confidence intervals around estimates will be too narrow, which can yield spurious results. On the other hand, over-estimation of the variance between imputations results in unnecessarily wide confidence intervals, which can be costly because it requires more observations than strictly required. Since both of these situations are undesirable, imputations and their variability should be properly evaluated. Such evaluation measures are currently missing or under-developed in MI software, like the world leading R package **mice** (Van Buuren and Groothuis-Oudshoorn 2011). The aim of this note is to provide guidelines for the evaulation of imputations to applied researchers who use MI methods. More specifically, this note adresses the question: 'How to diagnose convergence of the MI algorithm implemented in **mice**?'.

The convergence properties of **mice** are investigated by means of model-based simulation in R (Core Team 2017). All programming code used in this note is available on Github repository 'ShinyMICE'. The results of this simulation study are guidelines for applied researchers to evaluate convergence of MI algorithms. These guidelines will be implemented in the interactive evaluation tool for **mice**, 'ShinyMICE', which is currently under development.

1.1. Terminology

The intended audience of this note consists of empirical researchers and statisticians who use multiple imputation to solve missing data problems. Basic familiarity with MI methodology is assumed. For the theoretical foundation of MI, see Rubin (1987). For an accessible and comprehensive introduction to MI from an applied perspective, see Van Buuren (2018).

The convergence guidelines introduced in this paper are developed to be integrated into the **mice** environment (Van Buuren and Groothuis-Oudshoorn 2011) in R (Core Team 2017). This note therefore follows notation and conventions of Van Buuren and Groothuis-Oudshoorn (2011). Deviations from the 'original' notation by Rubin (1987), are described in (Van Buuren 2018, § XYZ).

Let Y denote an $n \times p$ matrix containing the data values on p variables for all n units in a sample. The collection of observed data values in Y is denoted as Y_{obs}; the missing part of Y is referred to as Y_{mis}. Response indicator R shows whether a data value in Y is missing or observed. The relation between R, Y_{obs}, and Y_{mis} determines the missingness mechanism. This paper only considers a 'missing completely at random' (MCAR) mechanism, where the probability to be missing is equal for all $n \times p$ cells in Y.

In this note, the terms 'unobserved' and 'missing' data are used interchangibly. Both refer to Y_{mis}, e.g. 'NA' values in R or '999' in SPSS. The terms 'incomplete' or 'observed' data denote Y_{obs}. Incomplete data is the starting point of the multiple imputation procedure. Figure 1.1 provides a schematic overview of the steps involved with MI.

Missing data in Y is 'imputed' (i.e., filled in) m times. The imputed data is combined with observed data Y_{obs} to create m completed data sets. On each completed data set, the analysis of scientific interest (or 'complete data model') is performed. The quantity of scientific interest (e.g., a regression coefficient) is denoted with Q. Since Q is estimated on each completed data set, m separate $\hat{Q}$ values are obtained. These m values are combined into a single pooled estimate $\bar{Q}$.

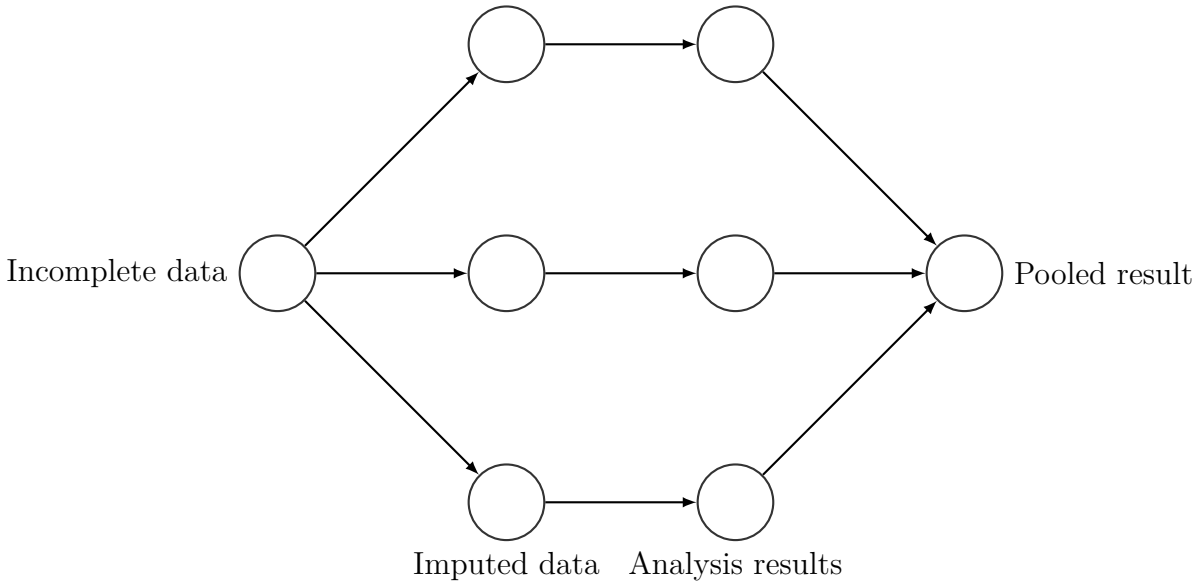


Figure 1: Scheme of main steps in multiple imputation (Adapted from (Van Buuren 2018, § 1.4.1))

This note is primarily concerned with the algorithmic properties of the imputation step. The algorithm employed within this step has an iterative nature. That is, before drawing m imputed values for each missing data point in Y_{mis}, a 'chain' of potential values is considered. Each of the m chains starts with an initial value, drawn randomly from Y_{obs}. The chains are terminated after a predefined number of iterations. Only the ultimate value that a chain lands on is imputed, and hence used in the next steps (analysis and pooling). The collection of values (or summary statistics) across iterations of one of the m imputations will be referred to as 'imputation chain'.

ADD THIS? The process of deliberately creating missingness in an otherwise complete data set is referred to as 'amputing' the data.

[explain difference MI, FCS, mice]

[Explain 'simulation condition' (the number of iterations the algorithm has before imputing), ...?]

1.2. Theoretical Background (NOG NIET AF VANAF HIER)

Explain the measures that I'm investigating here. R hat is ... Auto-correlation is... How are they computed? What do they say? How are they computed on MI data? Looking at imputation chains. Explain that there is no baseline to compare the measures with. Only current convergence check is to visually inspect trace plots. (But in the traceplots convergence seems immediate).

What is convergence in general?

We can never be certain of convergence, therefore convergence diagnostics evaluate signs of non-convergence (Hoff 2009). Convergence has two interpretations **[look up source!!!]**: mixing between chains and stability over iterations within chains. Ideally, we want the chains

to intermingle nicely, without trends within chains. Diagnose non-convergence in the mixing sense: $\widehat{R}$. Diagnose non-convergence in the stability sense: auto-correlation.

In the latter interpretation of convergence, convergence is interpreted as stability over iterations, or non-recurring. Non-recurrence can be evaluated with auto-correlation. Auto-correlation shows how dependent subsequent draws of an imputation chain are on the previous value. If there is a lot of dependence, draws at e.g. iteration five are significantly correlated with the value of the first draw. Auto-correlation above the confidence level **[explain that this is a critical value for correlation based on alpha .05]** indicate dependence within chains. Also, possible to use MC error, or Geweke statistic. But outside of scope of this study. Here, the focus is mainly on $\widehat{R}$.

The potential scale reduction factor $\widehat{R}$ tells us how much variance of an estimate could be shrunken down by running the chains infinitely long Gelman and Rubin (1992). That then tells us something about how dependent the chains are on the starting values. If there is no dependence on the initial values anymore, the chains have converged (in the mixing sense of the word). $\widehat{R}$ is then equal to one. Conventional threshold was $\widehat{R} < 1.2$ is acceptable. Most recent guideline by Vehtari, Gelman, Simpson, Carpenter, and Bürkner (2019) is more stringent: $\widehat{R} < 1.01$.

We can apply $\widehat{R}$ to the mean (or to the first two moments) of the variables of interest. $\widehat{R}$ is computed as follows:

REPHRASE: In the equations below, N is the number of draws per chain, M is the number of chains, and $S = MN$ is the total number of draws from all chains. For each scalar summary of interest θ, we compute B and W, the between- and within-chain variances:

$$\text{“}B = \frac{N}{M-1}\sum_{m=1}^{M}\left(\bar{\theta}^{(\cdot m)} - \bar{\theta}^{(\cdot\cdot)}\right)^2, \text{ where } \bar{\theta}^{(\cdot m)} = \frac{1}{N}\sum_{n=1}^{N}\theta^{(nm)}, \quad \bar{\theta}^{(\cdot\cdot)} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}^{(\cdot m)}$$

$$W = \frac{1}{M}\sum_{m=1}^{M}s_j^2, \text{ where } s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}\left(\theta^{(nm)} - \bar{\theta}^{(\cdot m)}\right)^2 \text{”}. \text{ (Vehtari } et\ al. \text{ 2019, p. 5)}$$

The proportion of within chain variance W against the weighted average of B and W, $\widehat{\text{var}}^+$ tells us how much variance of θ could be shrunken down if $N \to \infty$. $\widehat{\text{var}}^+$ is computed as follows:

$$\widehat{\text{var}}^+(\theta|y) = \frac{N-1}{N}W + \frac{1}{N}B.$$

The potential scale reduction factor $\widehat{R}$ is obtained as:

$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^+(\theta|y)}{W}}.$$

Convergence of MI

A fundamental assumption of MICE is convergence of the algorithm. MICE is a sort of Gibbs sampler, a type of Markov chain Monte Carlo (MCMC) algorithm. In general, the validity of

inference resulting from MCMC algorithms is threatened by non-convergence. Hence we use convergence diagnostics to flag non-convergence. But it is not known whether conventional convergence diagnostics for MCMC methods work on MI data.

Why is $\widehat{R}$ not applicable on MI data? $\widehat{R}$ is not appropriate because it assumes over-dispersed initial values, which means that the initial values of the m imputation chains are 'far away' from the distribution that the chains are converging to. With MI procedures, initial values are chosen randomly from the observed data. This means that one initial value is necessary for each imputation chain.

Without over-dispersed initial states, $\widehat{R}$ can falsely diagnose convergence (Brooks and Gelman 1998). This suggests that $\widehat{R}$ would not be sensitive enough to flag non-convergence of MI algorithms. Empirical finding, however, show that the opposite may be true: Lacerda, Ardington, and Leibbrandt (2007) report $\widehat{R}$ values above the threshold of $\widehat{R} < 1.1$ after fifty iterations.

How is convergence checked now (current practice)?

Conventional thresholds to diagnose non-convergence– e.g., Gelman and Rubin's 1992 statistic $\widehat{R} < 1.1$ –are not applicable on multiply imputed data (Lacerda *et al.* 2007) [**Expand or remove Rhat here! Refer to future section?**]. Therefore, empirical researchers have to rely on visual inspection procedures that are theoretically equivalent to $\widehat{R}$ (White, Royston, and Wood 2011) [**explain why this is appropriate, and numerical is not**]. Visually assessing convergence is not only difficult to the untrained eye, it might also be futile. The convergence properties of MI algorithms lack scientific consensus (Takahashi 2017), and some default MICE techniques might not converge to stable distributions at all (Murray 2018). Moreover, convergence diagnostics for MI methods have not been systematically studied (Van Buuren 2018).

1.3. Simulation Hypothesis

The aim is to evaluate whether the imputation procedure has converged. The primary research interest is in determining whether $\widehat{R}$ is an appropriate convergence diagnostic, and if so, which level of stringency suits MI data.

Hypothesis of this simulation study is that $\widehat{R}$ will over-estimate non-convergence. [**Or use:**] **Hypothesis based on Lacerda *et al.* (2007) is that the conventional acceptable level of $\widehat{R}$ is too strict for MI data. We expect that the simulation diagnostics will indicate valid inference before $\widehat{R}$ will.**

2. Methods

In this study, 1000 simulations are performed for each of the 100 simulation conditions. The simulation set-up consists of several steps, summarized in the pseudo-code below. The complete R script of the simulation study is available on Github.

```
# pseudo-code of simulation
simulate data
for (number of simulation runs from 1 to 1000)
  for (number of iterations from 1 to 100)
```

```
    create missingness
    impute the missingness
    compute convergence diagnostics
    perform analysis
    pool results
    compute simulation diagnostics
aggregate convergence and simulation diagnostics
```

2.1. Data

The simulated dataset is a finite population of $N = 1000$. The data are simulated to solve a multiple linear regression complete data problem. The quantity of scientific interest is the estimated regression coefficient of predictor X on outcome variable Y. The linear regression model is $Y \sim X + Z_1 + Z_2$, where Y is the dependent variable, X is an independent variable, and Z_1 and Z_2 are covariates. The data generating model of the predictors is a multivariate normal distribution with means structure μ, and variance-covariance matrix Σ. Outcome variable Y is deduced from the predictor variables as follows:

$$\begin{pmatrix} X \\ Z_1 \\ Z_2 \\ \epsilon \end{pmatrix} \sim N \left[\begin{pmatrix} 12 \\ 3 \\ 0.5 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 & 4 & 1.8 & 0 \\ 4 & 16 & 4.8 & 0 \\ 1.8 & 4.8 & 9 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix} \right]$$

$$Y = 2 \times X + 0.5 \times Z_1 - 1 \times Z_2 + \epsilon$$

2.2. Amputation

After data generation, the complete data is 'amputed'. That is, the **mice** function `ampute()` in R is used to impose an MCAR missingness mechanism upon the data. The missingness is univariate, and the probability to be missing is the same for all variables, namely 20%. This leaves 20% of the rows completely observed. Table XYZ shows a summary of the generated complete data, and the amputed data. The amputed dataset is the starting point of each of the 1000 simulations per simulation condition.

2.3. Imputation

Missing data points are imputed with **mice** in R. All simulations are performed with imputation method 'norm' (Bayesian linear regression imputation), and five imputation chains ($m = 5$). The number of iterations is varied over simulations ('maxit' argument between 1 and 100). Each simulation condition (i.e., each number of iterations) is simulated 1000 times. Simulation and convergence diagnostics are aggregated over the 1000 MCMC simulations.

2.4. Convergence Diagnostics

$\widehat{R}$ is computed within imputation chains: for each variable, for each simulation condition, for each simulation. The maximum value across variables within the same simulation is reported. Maximum $\widehat{R}$ values per simulation condition are aggregated across simulations.

Autocorrelation (AC) is computed within imputation chains, as the correlation between θ at the i^{th} iteration and θ at the $i + 1^{th}$ iteration. For now, we only consider the chain mean as scalar of interest θ. The AC of the variable with the highest absolute AC value is reported per simulation. These values are then averaged per simulation condition.

2.5. Analysis

Linear regression is performed using the R function `lm()`. The quantity of scientific interest Q is the regression coefficient of X on Y. Q is estimated in each imputation, in each simulation condition, in each repetition. The $m = 5$ estimated regression coefficients $\hat{Q}$ are pooled according to Rubin's 1987 rules with **mice** function `pool()`.

2.6. Simulation diagnostics

The simulation diagnostics are as recommended by Van Buuren (2018). Namely, average bias, average confidence interval width, and empirical coverage rate (coverage probability) across simulations. We could also look at distributional characteristics, and plausibility of imputed values, see Vink. For now, this is outside of the scope of this study.

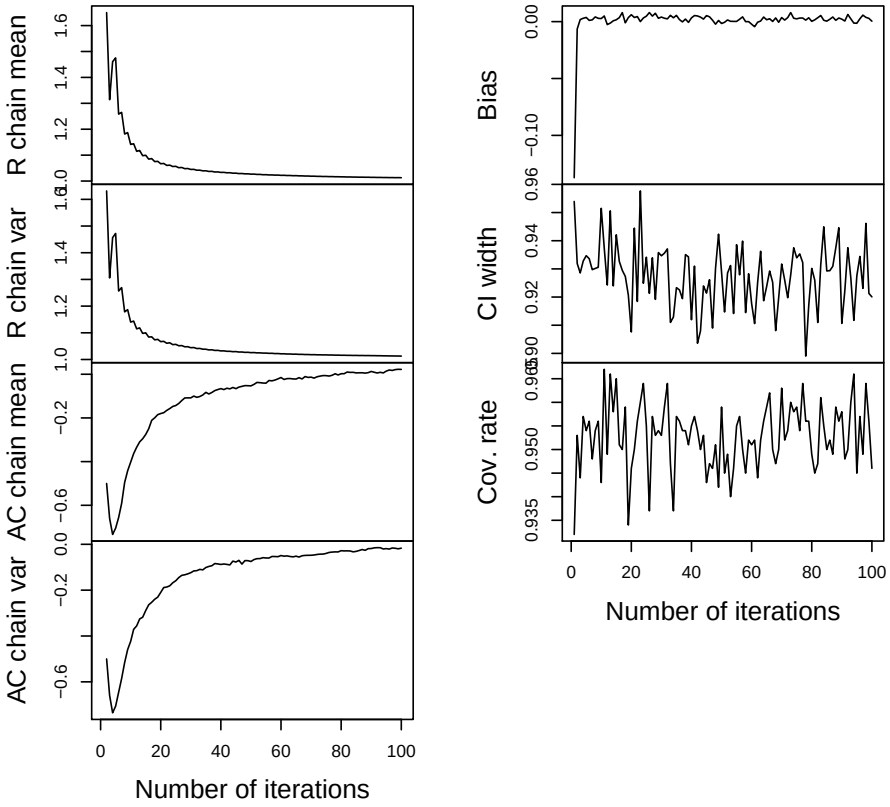
Bias is computed as the estimated regression coefficient after MI minus the true regression coefficient. Bias is averaged over all simulations with the same simulation condition.

Confidence interval width (CIW) is computed as the diffence between the lower and upper bound of the 95% confidence interval (CI95%). The CI95% bounds are computed as the estimated regression coefficient plus or minus (respectively) the pooled SE across imputations times 2.66 (the quantile of a t distribution with m-1 degrees of freedom). CIW is averaged over all simulations with the same simulation condition.

Coverage rate is computed as the proportion of simulations (with the same simulation condition) in which the true regression coefficient is between the bounds of the CI95%.

3. Results

Figure **??** shows the results over 1000 simulations. A subset of simulation conditions is presented in Table 1.



Try to put the chain means and vars together in one panel

Evaluate simulation diagnostics one by one: bias, empirical SE, confidence interval width, and coverage rate. Then evaluate convergence diagnostics $\widehat{R}$ and auto-correlation.

3.1. Convergence diagnostics

$\widehat{R}$. Steep drop down up-to iteration twenty, then more gradual decrease upto iteration fourty, more or less stable after that. Conventional threshold $\widehat{R} < 1.2$ reached after seven or eight iterations, $\widehat{R} < 1.1$ after fourteen iterations for chain means and sixteen for chain variances, $\widehat{R} < 1.01$ not reached within the 100 iterations (see online appendix for full table of results).

Auto-correlation. Within the 100 iterations considered in this study, AC decreases steeply, then slowly increases. Lowest AC observed is -0.735 at four iterations. There appears to be a trend towards zero from four iterations upwards. Auto-correlation does not plateau completely. The average absolute maximum ACs are negative up to 77 iterations for chain means, and for chain variances all iterations considered here have negative ACs.

3.2. Simulation diagnostics

Bias. From two iterations upwards, average bias across repititions is stable. Across iterations two to one hundred, bias fluctuates within a narrow range around zero.

Table 1: Simulation and convergence diagnostics over 1000 MCMC simulations.

It.	Bias	CI width	Cov. rate	$\widehat{R}_{mean}$	$\widehat{R}_{var}$	AC_{mean}	AC_{var}
1	-0.137	0.954	0.932	NA	NA	NA	NA
2	-0.006	0.932	0.953	1.650	1.632	-0.500	-0.500
3	0.002	0.929	0.944	1.314	1.306	-0.660	-0.659
4	0.003	0.933	0.957	1.461	1.457	-0.733	-0.735
5	0.004	0.935	0.954	1.475	1.472	-0.705	-0.706
6	0.001	0.934	0.956	1.258	1.256	-0.656	-0.646
7	0.002	0.930	0.948	1.265	1.269	-0.591	-0.585
8	0.004	0.930	0.954	1.181	1.178	-0.495	-0.516
9	0.003	0.931	0.956	1.187	1.187	-0.442	-0.459
10	0.003	0.952	0.943	1.141	1.140	-0.403	-0.423
15	0.002	0.942	0.965	1.100	1.100	-0.276	-0.289
25	0.005	0.934	0.955	1.057	1.057	-0.143	-0.159
50	-0.002	0.929	0.959	1.027	1.026	-0.053	-0.075
100	0.000	0.920	0.946	1.013	1.013	0.022	-0.017

Confidence interval width. CIW does not show a clear trend across iterations. One might argue that there is some downward trending upto fourty iterations.

Coverage rate. Coverage rate is more or less stable from two iterations upwards. On average, the coverage rate is somewhat higher then the expected nominal coverage of 95% **?**, namely 0.953.

4. Summary and discussion

Summary. This note illustrates that conventional convergence diagnostics behave differently on MI data than other MCMC methods like Gibbs samplers in Bayesian analyses. The most recent recommended threshold for $\widehat{R}$ ($\widehat{R} < 1.01$) is too stringent for MI data.

From the simulation diagnostics it appears that as little as two iterations could be sufficient to draw valid inference. Convergence diagnostics $\widehat{R}$ and auto-correlation, however, indicate deviant behavior upto twenty or even fourty iterations.

R hat below 1. Moreover, $\widehat{R}$ could theoretically not be smaller than one, yet it happened several times in this study (see online appendix XYZ). How could $\widehat{R}$ smaller than 1 occur? The number of simulations is smaller than in 'regular' MCMC processes [explain that less iterations is an advantage of MI, not a disadvantage compared to MCMC]. Therefore, the '$(n - 1/n)$' [add equation number] correction factor can influence the estimated potential scale reduction factor. This downwards bias is in the opposite direction than expected: "The mixture-of-sequences variance, V, should stabilize as a function of n. (Before convergence, we expect σ^2 to decrease with n, only increasing if the sequences explore a new area of parameter space, which would imply that the original sequences were not overdispersed for the particular scalar summary being monitored.)" (Brooks and Gelman 1998, p 438).

Negative ACs. Auto-correlation is dangerous when positive. These auto-correlations are negative. Still, we want the iterations to be stable and independent. Default maxit is five iterations now, should this be different? Are the 'waves' most pronounced at iteration 5? But

why the dip??

Implication of dip in AC is that default maxit value of five iterations is the worst possible number of iterations.

Future research. We would like to have convergence measures for multivariable statistics (scalars?) of interest. This is, however, dependent of the complete data model. The eigenvector decomposition method proposed by McKay (?) should be implemented. I could not find any resources to apply this method and it is outside the scope of this thesis to investigate how this approach could be implemented. **[this paragraph is too negative, explain scope in intro instead.]**

Computational details

The results in this paper were obtained using R 3.6.1 with the **mice** 3.6.0.9000 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

This paper is written by the sole author (Hanne Oberman, BSc.), with guidance from Master thesis supervisors prof. dr. Stef van Buuren, and dr. Gerko Vink.

References

Allison PD (2001). *Missing data*, volume 136. Sage publications.

Brooks SP, Gelman A (1998). "General Methods for Monitoring Convergence of Iterative Simulations." *Journal of Computational and Graphical Statistics*, **7**(4), 434–455. ISSN 1061-8600. doi:10.1080/10618600.1998.10474787. URL https://www.tandfonline.com/doi/abs/10.1080/10618600.1998.10474787.

Core Team (2017). *: A language and environment for statistical computing*. Vienna, Austria. Tex.organization: Foundation for Statistical Computing, URL https://www.R-project.org/.

Gelman A, Rubin DB (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science*, **7**(4), 457–472. ISSN 0883-4237, 2168-8745. doi:10.1214/ss/1177011136. URL https://projecteuclid.org/euclid.ss/1177011136.

Hoff PD (2009). *A First Course in Bayesian Statistical Methods*. Springer Texts in Statistics. Springer New York, New York, NY. ISBN 978-0-387-92299-7 978-0-387-92407-6. doi:10.1007/978-0-387-92407-6. URL http://link.springer.com/10.1007/978-0-387-92407-6.

Lacerda M, Ardington C, Leibbrandt M (2007). "Sequential regression multiple imputation for incomplete multivariate data using Markov chain Monte Carlo."

Murray JS (2018). "Multiple Imputation: A Review of Practical and Theoretical Findings." *Statistical Science*, **33**(2), 142–159. ISSN 0883-4237. doi:10.1214/18-STS644. URL https://projecteuclid.org/euclid.ss/1525313139.

Rubin DB (1987). *Multiple Imputation for nonresponse in surveys*. Wiley series in probability and mathematical statistics Applied probability and statistics. Wiley, New York, NY. ISBN 978-0-471-08705-2.

Takahashi M (2017). "Statistical Inference in Missing Data by MCMC and Non-MCMC Multiple Imputation Algorithms: Assessing the Effects of Between-Imputation Iterations." *Data Science Journal*, **16**, 37. ISSN 1683-1470. doi:10.5334/dsj-2017-037. URL http://datascience.codata.org/articles/10.5334/dsj-2017-037/.

Van Buuren S (2018). *Flexible imputation of missing data*. Chapman and Hall/CRC. ISBN 0-429-96035-2.

Van Buuren S, Groothuis-Oudshoorn K (2011). "mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, **45**(1), 1–67. ISSN 1548-7660. doi:10.18637/jss.v045.i03. URL https://www.jstatsoft.org/index.php/jss/article/view/v045i03.

Vehtari A, Gelman A, Simpson D, Carpenter B, Bürkner PC (2019). "Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC." *arXiv:1903.08008 [stat]*. ArXiv: 1903.08008, URL http://arxiv.org/abs/1903.08008.

Vink G (????). "Towards a standardized evaluation of multiple imputation routines."

White IR, Royston P, Wood AM (2011). "Multiple imputation using chained equations: Issues and guidance for practice." *Statistics in Medicine*, **30**(4), 377–399. ISSN 1097-0258. doi:10.1002/sim.4067.

Affiliation:

Hanne Ida Oberman, BSc.
Methodology and Statistics for the Behavioural, Biomedical and Social Sciences
Department of Methodology and Statistics
Faculty of Social and Behavioral Sciences
Utrecht University
Heidelberglaan 15
3500 Utrecht, The Netherlands
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