Journal of Statistical Software

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

A Note on Convergence Diagnostics for Multiple Imputation Algorithms

Research Report (Preparation for Research Master Thesis, Research Seminar)
Methodology and Statistics for the Behavioural, Biomedical and Social Sciences

Hanne Oberman
Utrecht University

Abstract

The ubiquitous problem of missing information may be circumvented by implementing the framework of multiple imputation (MI). With MI, missing data points are 'filled in' (i.e. imputed) several times, and statistical inference is performed on each imputed data set. The validity of inferences may be threatened by non-convergence of this iterative algorithmic procedure, but diagnostics to evaluate algorithmic convergence have not been implemented in MI software. This note investigates the performance of two convergence diagnostics (($\widehat{R}$, Gelman and Rubin 1992); (auto-correlation, Schafer 1997)) through model-based simulation. To investigate convergence of the algorithm, the number of iterations is varied between one and one-hundred. Each of these 100 simulation conditions is replicated 1000 times. The convergence diagnostics are evaluated against three recommended simulation diagnostics (bias, confidence interval width, and coverage rate (Van Buuren 2018), since no benchmark diagnostic is available. This simulation study shows that three iterations may be sufficient to obtain ... **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 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 an online appendix to the technical paper on ShinyMICE.**

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., list-wise deletion), which may 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 'imputed' (i.e. filled in) several times. The variability between imputations is used to reflect how much uncertainty in the inference is introduced by the missingness. Therefore, MI can provide valid inferences despite missing information.

To obtain valid inferences with MI, the variability between imputations should be properly represented (Rubin 1987; Van Buuren 2018). If the variance between imputations is under-estimated, 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 lowers the statistical power. Since both of these situations are undesirable, imputations and their variability should be evaluated. Evaluation measures, however, are currently missing or under-developed in MI software, like the world-leading **mice** package (Van Buuren and Groothuis-Oudshoorn 2011) in R (Core Team 2019).

The goal of this research project is to develop novel methodology and guidelines for evaluating MI methods. These evaluation measures and guidelines will subsequently be implemented in an interactive evaluation framework for multiple imputation to aid applied researchers in drawing valid inference from incomplete datasets. This note provides the theoretical foundation towards the diagnostic evaluation of the convergence of MI algorithms. For reasons of brevity, this note only focuses on the MI algorithm implemented in **mice** Van Buuren and Groothuis-Oudshoorn (2011). To this end we will evaluate how convergence of the imputation algorithm can be diagnosed.

The convergence properties of the MI algoritm in **mice** are investigated through model-based simulation. The results of this simulation study are guidelines for assessing convergence of MI algorithms. These guidelines will be implemented in an interactive evaluation tool for **mice**, 'ShinyMICE', which is currently under development. All programming code used in this note is available from github.com/gerkovink/ShinyMICE/simulation.

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 2019). 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, § 2.2.3).

Let Y denote an $n \times p$ matrix containing the data values on p variables for all n units in a

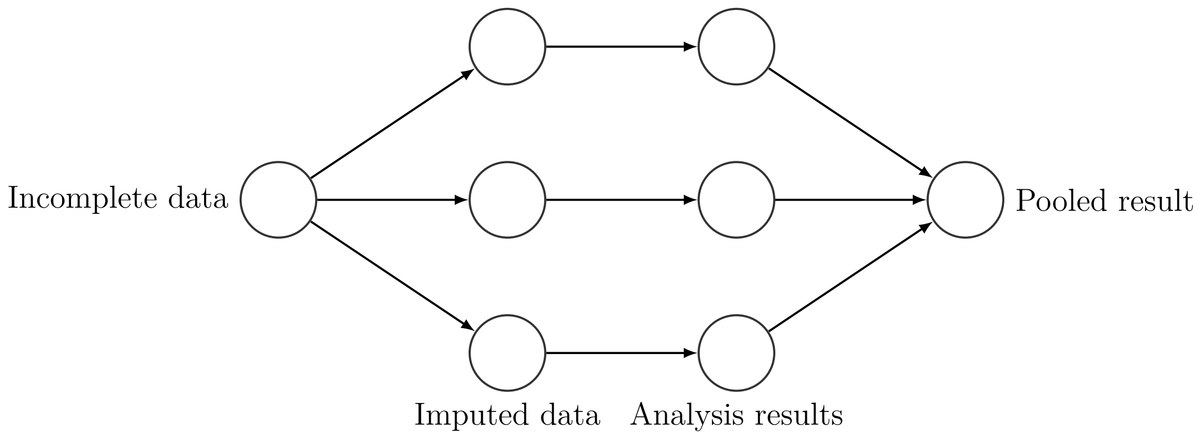

Figure 1: Scheme of the main steps in multiple imputation ($m = 3$). Adapted from (Van Buuren 2018, § 1.4.1).

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 note only considers a 'missing completely at random' (MCAR) mechanism, where the probability of being missing is equal for all $n \times p$ cells in Y.

In this note, the terms 'unobserved' and 'missing' data are used interchangeably. 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 $\bar{Q}$.

This note focuses on 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 sampled. 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 sample that a chain lands on is imputed and subsequently used in the analysis and pooling steps. The collection of samples across iterations (between the initial value and the imputed value) for each of the m imputations will be referred to as an 'imputation chain'. The number of iterations will be denoted with t.

1.2. Theoretical Background

The multiple imputation algorithm in **mice** is a special case of Markov chain Monte Carlo (MCMC) methods–a framework of iterative algorithmic procedures. While methodology for diagnosing convergence of MI algorithms has not been systematically studied (Van Buuren 2018), several convergence diagnostics have been proposed for other types of MCMC methods (e.g., Gibbs samplers) and the framework in general. This section provides a brief review of: 1) how convergence is defined in the general context of iterative algorithmic procedures; 2) what diagnostic tools are available; and 3) which of these may apply to MI algorithms.

Loosely speaking, the aim of MCMC algorithms (e.g., Gibbs samplers, or the MI algorithm in **mice**) is to sample values from an unknown target distribution (e.g., the posterior distribution of a parameter in Bayesian inference, or a distribution of missing values, respectively). Convergence is reached once the algorithm samples exclusively from the target distribution (Cowles and Carlin 1996). Since this target distribution is unknown by definition, convergence can only be monitored by evaluating signs of non-convergence (Hoff 2009). In practice, the definition of convergence consists of two components: 'mixing' of chains together, and 'stationarity' of individual chains (Gelman, Carlin, Stern, Dunson, Vehtari, and Rubin 2013, p. 284). The mixing component is satisfied when chains intermingle such that the only difference between the chains is caused by the randomness induced by the algorithm. Stationarity refers to the absence of trending across iterations within chains.

Most convergence diagnostics target either of the two components. The stationarity component of convergence may be evaluated with auto-correlation (Schafer 1997; Gelman *et al.* 2013), numeric standard error (Geweke 1992, which is commonly called 'MC error'), or Raftery and Lewis's (1991) procedure to determine how many iterations are necessary to deminish the effect of trending within chains. The mixing component can be assessed with the potential scale reduction factor $\widehat{R}$ (Gelman and Rubin 1992, widely referred to as 'the Gelman-Rubin statistic'). With an adapted version of $\widehat{R}$, proposed by Vehtari, Gelman, Simpson, Carpenter, and Bürkner (2019), we may also evaluate the stationarity component of convergence. This would make $\widehat{R}$ a general convergence diagnostic. The application of $\widehat{R}$ to assess stationarity has not been thourougly investigated. Therefore, this study employs both $\widehat{R}$ and auto-correlation to investigate algorithmic convergence, as recommended by (Cowles and Carlin 1996, p. 898). Other convergence diagnostics are outside the scope of this study.[1]

It is not known whether any convergence diagnostic is appropriate to diagnose convergence of **mice** algorithms. As by (Van Buuren 2018, § 6.5.2), the current practice is to visually inspect imputation chains for signs of non-mixing and non-stationarity. Visually assessing convergence, however, may be challenging to the untrained eye. **[This is problematic because]** There is no scientific consensus on the convergence properties of multiple imputation algorithms (Takahashi 2017), and in some situations, the default techniques in **mice** might not converge to stable target distributions at all (Murray 2018). Therefore, convergence of the **mice** algorithms should be monitored carefully.

The convergence diagnostics under consideration are $\widehat{R}$ and auto-correlation. Perfect convergence is reached when the variance between imputation chains is equivalent to the variance within chains ($\widehat{R} = 1$), and there is no dependency between subsequent samples within imputation chains (auto-correlation $= 0$). To define $\widehat{R}$, we follow notation by (Vehtari *et al.* 2019, p. 5). Let M be the number of chains, N the number of iterations per chain, and θ the scalar summary of interest (e.g., chain mean or chain variance). We compute the variance within each chain of θ values (W) and between the m chains (B) as follows:

[1] All of these methods evaluate the convergence of univariate scalar summaries (e.g., chain means or variances). These convergence diagnostics cannot diagnose convergence of multivariable statistics (i.e., relations between scalar summaries). Van Buuren (2018) proposed to implement multivariable evaluation by means of eigenvalue decomposition MacKay and Mac Kay (2003). This method is outside of the scope of the current study.

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

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

From the between and within chain variances we compute a weighted average, $\widehat{\text{var}}^+$, which over-estimates the total variance of θ. $\widehat{R}$ is then obtained as a ratio between the over-estimated total variance and the within chain variance:

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

High $\widehat{R}$ would indicate .... low $\widehat{R}$ values indicate .... The conventionally acceptable threshold was $\widehat{R} < 1.2$ Gelman and Rubin (1992). More recently, Vehtari *et al.* (2019) proposed a more stringent threshold: $\widehat{R} < 1.01$.

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. A high auto-correlation indicates dependence within chains. The magnitude of the AC can be interpreted qualitatively or quantitatively. Quantitative evaluation of the AC entails comparing the observed ACs to the critical values of a two-tailed 95% confidence interval, divided by the square root of the number of iterations.

Convergence diagnostics applied to MI

$\widehat{R}$ may not be an appropriate diagnostic to evaluate MI data because it assumes over-dispersed initial values. This means that the initial values of the m imputation chains are 'far away' from the distribution that the chains are converging to. In the **mice** algorithm, initial values are chosen randomly from the observed data. Therefore, we cannot be certain that the initial values are over-dispersed.

Without over-dispersed initial states, $\widehat{R}$ may falsely diagnose convergence (Brooks and Gelman 1998). This suggests that $\widehat{R}$ would not be sensitive enough to flag non-convergence of MI algorithms. An empirical finding, however, shows 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.

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. **Or if there is a discrepancy between the visual and numeric inspection. There is no baseline to compare the convergence diagnostics with. Therefore, we will only evaluate performance measures.**

This simulation study hypothesizes 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

To investigate the convergence properties of the **mice** algorithm, we perform model-based simulation of the MI procedure with varying imputation chain lengths. Each simulation condition has a different maximum number of iterations, from one to one-hundred iterations. We compute $\widehat{R}$ and auto-correlation ('convergence diagnostics') and performance measures ('simulation diagnostics') for each condition. The number of simulation runs per condition is 1000. The simulation set-up consists of several steps, summarized in the pseudo-code below.[2]

```
# 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 simulation

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 $Y \sim \beta_1 X + \beta_2 Z_1 + \beta_3 Z_2$, 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.

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

Outcome variable Y is subsequently calculated as $Y = 2X + .5Z_1 - Z_2 + \epsilon$.

2.2. Amputation

The complete data is the point of origin for all simulations. In each repetition of the simulation study, the complete data is 'amputed' once. That is, the **mice** function `ampute()` in R is used

[2] The complete R script of the simulation study is available on Github.

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. The resulting amputed data is equal for all simulation conditions in the same repetition.

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, each simulation condition, and each simulation. The maximum value across variables within the same simulation is reported. Maximum $\widehat{R}$ values per simulation condition are aggregated across simulations.

Auto-correlation (AC) is computed within imputation chains, as the correlation between θ at **the i^{th} iteration and θ at the $i+1^{th}$ iteration** [**CHECK VARIABLE IN NOTATION**], where θ is a scalar summary of interest (i.e., chain means and chain variances). 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, § 2.5.2). These evaluation criteria comprise of average bias, average confidence interval width, and empirical coverage rate across simulations. We could also look at distributional characteristics, and plausibility of imputed values, see Vink (n.d.). 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 difference 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

Figures 2 and 3 display results of the simulation study aggregated over 1000 repetitions. A subset of simulation conditions is presented in Table 1. This section describes convergence diagnostics ($\widehat{R}$ and auto-correlation) and simulation diagnostics (bias, confidence interval width and coverage rate) first separately, and then together.

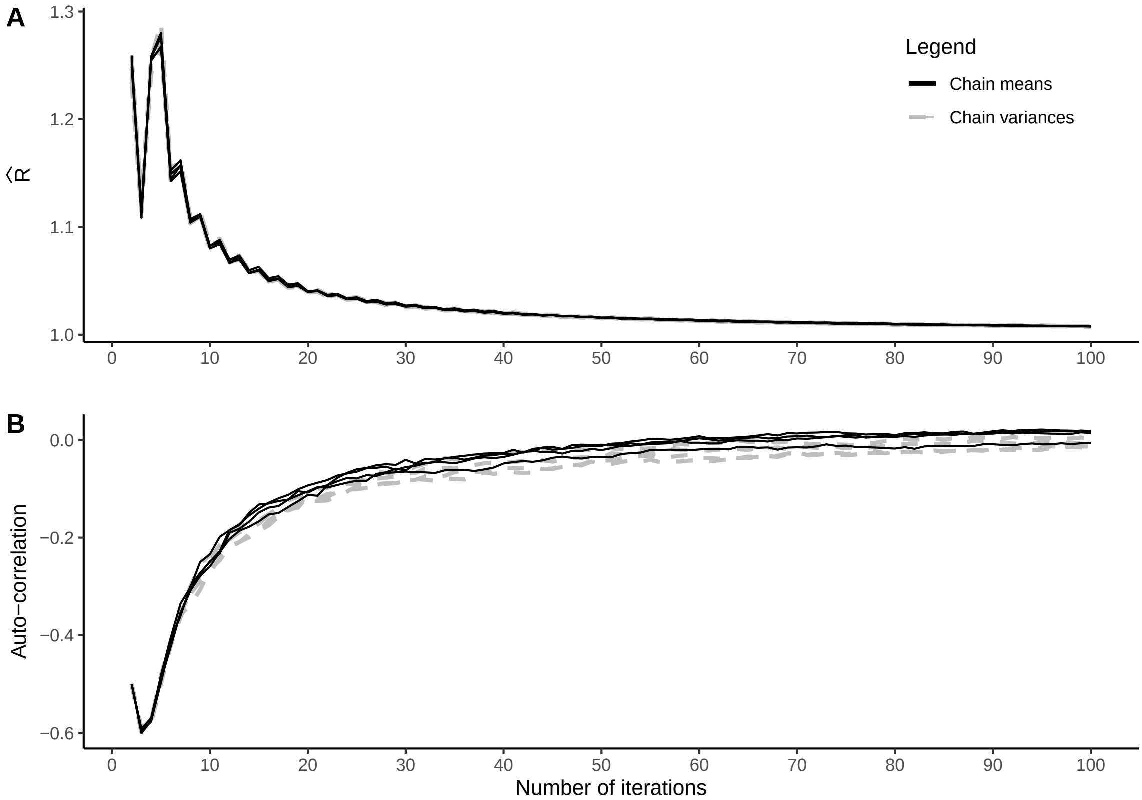

Figure 2: Convergence diagnostics over 1000 MCMC simulations.

3.1. Convergence diagnostics

$\widehat{R}$. The $\widehat{R}$ values for chain means and chain variances show equivalent trends, neither is there a clear difference between the four variables. Roughly speaking, we see that $\widehat{R}$ is higher for simulation conditions with a lower number of iterations, and approaches one for conditions with more iterations. Figure 2 shows an initial dip, a small bump upwards, and then a steep decline. This decline **houdt aan** up-to iteration twenty, after which a more gradual decrease can be observed up-to iteration forty. The $\widehat{R}$ values are more or less stable after that. The **conventionally acceptable threshold for convergence, maximum $\widehat{R} < 1.2$,[CHECK IN INTRO]** is reached in the simulation condition with three iterations, and conditions with six iterations or more. The widely used threshold $\widehat{R} < 1.1$ is met in conditions with ten or more iterations. The most recent recommended threshold is not reached within the 100 iterations considered in this simulation study. $\widehat{R} > 1.01$ for all simulation conditions (see online appendix for the full table of results). **Check is computing r ht is still correct! Also say something about Rh below 1.**

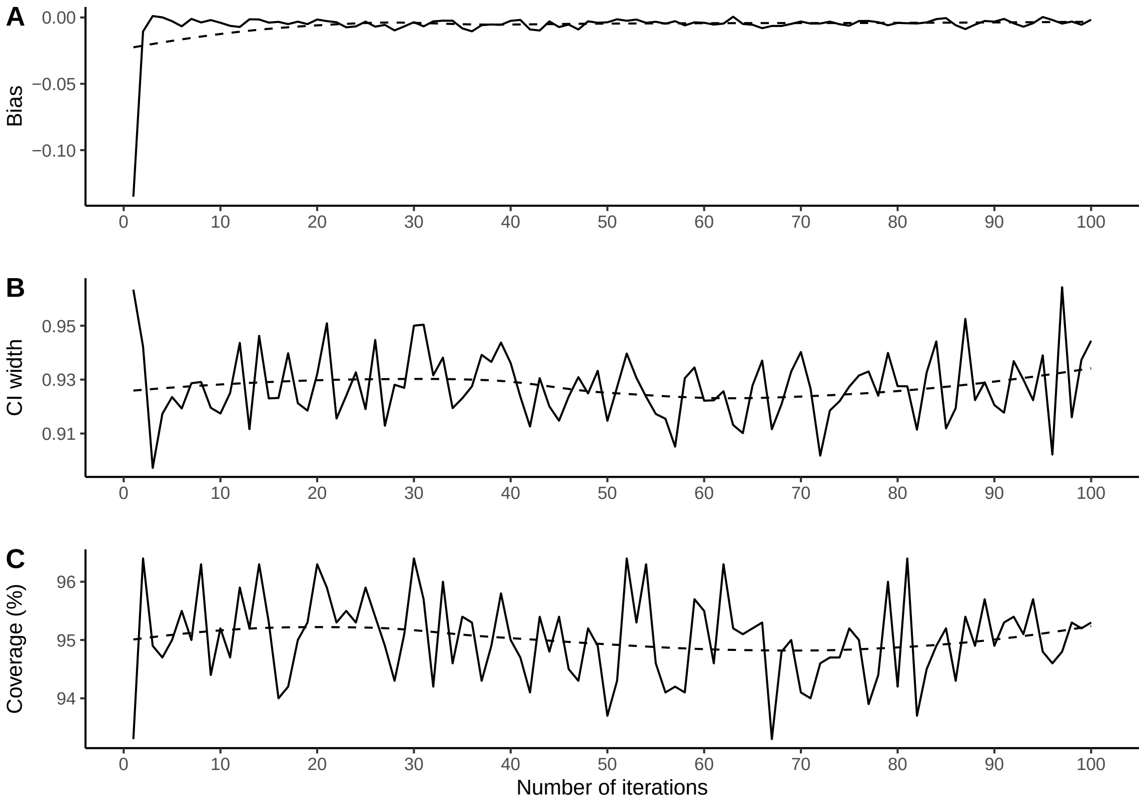

Figure 3: Simulation diagnostics over 1000 MCMC simulations.

Auto-correlation. All observed auto-correlations (aggregated over the 1000 repetitions) are negative or close to zero. There are some differences between variables, and the ACs of chain variances seem lower than the ACs of chain means. The general trend in figure 2 is that ACs decrease steeply as the number of iterations increases. **The lowest AC observed is at four iterations. Simulation conditions with a higher number of iterations show a slow increase towards zero. Auto-correlations do not plateau completely within the range of simulation conditions considered in this study. It seems that ACs might become positive if more iterations would be allowed. 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. Figure 3 shows the average bias between the regression coefficient of X on Y in the complete dataset, and $\bar{Q}$, **the pooled estimated regression coefficient and the ... Average bias is close to zero for simulation conditions with two iterations or more. The simulation condition with one iteration has a negative bias. From two iterations upwards, the average bias across repetitions is stable. Across iterations two to one hundred, bias fluctuates within a narrow range around zero. Loess line is flat around it = 20.**

Confidence interval width. CIW is only clearly divergent for the simulation condition with

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

one iteration. Conditions with two or more iterations have similar CIWs. The magnitude of differences from the loess line is somewhat larger up-to the condition with four iterations. From four iterations on, CIW seems to be stable across iterations. **Loess line is never flat.**

Coverage rate. The coverage rate is more or less stable from two iterations upwards. On average, the coverage rate is somewhat higher than the expected nominal coverage of 95% Neyman (1934), namely 95.3%. Loess line is never flat.

give some implications for these measures. connect convergence and simulation diagnostics.

4. Summary and discussion

Answer RQ. How to diagnose non-convergence? It's complicated.

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 $\widehat{R}$ ($\widehat{R} < 1.01$) may be too stringent for MI data. **However, using the 1.2 threshold may be to lenient because this was observed at three iterations and then again after six. If we would use 1.2, we might miss this increase after three iterations. The 1.1 threshold seems ok. Also, ten iterations are computationally not terrible. With increased performance and storage capacities, doubling the default is fine I guess.**

From the simulation diagnostics, it appears that as little as three iterations might be sufficient to obtain unbiased, confidence valid estimates. Convergence diagnostics $\widehat{R}$ and autocorrelation, however, indicate that convergence may only be reached after twenty or even forty iterations.

R hat below 1. $\widehat{R}$ could theoretically not be smaller than one, yet it occurred several times in this study (see online appendix XYZ). This can happen when the number of simulations is smaller than in 'regular' MCMC processes [explain that fewer 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??

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

Future research. This study consdered only a MCAR missingness mechanism. Necessary but not sufficient to demonstrate the performance of convergence diagnostics. This is just a proof of concept. Further reseach is needed to investigate the performance under violation of convergence, e.g. dependency between predictors (very high correlations).

Computational details

The results in this paper were obtained using R 3.6.1 Core Team (2019) with the **mice** 3.6.0.9000 package Van Buuren and Groothuis-Oudshoorn (2011). R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Acknowledgments

This note is part of a Research Master's thesis project. It was written by the sole author (Hanne Oberman, BSc.), with guidance from Master thesis supervisors prof. dr. Stef van Buuren and dr. Gerko Vink, and research seminar mentor dr. Daniela Cianci.

References

Allison PD (2001). *Missing data*. 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. `doi:10.1080/10618600.1998.10474787`.

Core Team (2019). *R: A language and environment for statistical computing*. Vienna, Austria. Tex.organization: R Foundation for Statistical Computing, URL `https://www.R-project.org/`.

Cowles MK, Carlin BP (1996). "Markov chain Monte Carlo convergence diagnostics: a comparative review." *Journal of the American Statistical Association*, **91**(434), 883–904. ISSN 0162-1459.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013). *Bayesian Data Analysis*. CRC Press LLC, Philadelphia, PA, United States.

Gelman A, Rubin DB (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science*, **7**(4), 457–472. doi:10.1214/ss/1177011136.

Geweke J (1992). "Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments." *Bayesian statistics*, **4**, 641–649.

Hoff PD (2009). *A First Course in Bayesian Statistical Methods*. Springer Texts in Statistics. Springer New York, New York, NY. doi: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." *Technical report*, University of Cape Town, South Africa.

MacKay DJ, Mac Kay DJ (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Murray JS (2018). "Multiple Imputation: A Review of Practical and Theoretical Findings." *Statistical Science*, **33**(2), 142–159. doi:10.1214/18-STS644.

Neyman J (1934). "On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection." *Journal of the Royal Statistical Society*, **97**(4), 558–625. doi:10.2307/2342192.

Raftery AE, Lewis S (1991). "How many iterations in the Gibbs sampler?" *Technical report*, Washington University Seatle, Department of Statistics, United States.

Rubin DB (1987). *Multiple Imputation for nonresponse in surveys*. Wiley series in probability and mathematical statistics Applied probability and statistics. Wiley, New York, NY.

Schafer JL (1997). *Analysis of incomplete multivariate data*. Chapman and Hall/CRC.

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. doi:10.5334/dsj-2017-037.

Van Buuren S (2018). *Flexible imputation of missing data*. Chapman and Hall/CRC.

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, URL http://arxiv.org/abs/1903.08008.

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

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