Sociological Methods & Research
2020, Vol. 49(?):1–23

DOI: 10.1177/ToBeAssigned
journals.sagepub.com/home/smr
SAGE

Missing the Point: Non-Convergence in Iterative Imputation Algorithms

H. I. Oberman[1]

Abstract

Iterative imputation is a popular tool to accommodate missing data. While it is widely accepted that valid inferences can be obtained with this technique, these inferences all rely on algorithmic convergence. There is no consensus on how to evaluate the convergence properties of the method. This paper provides insight into identifying non-convergence of iterative impuation algorithms.

Keywords

missing data, iterative imputation, non-convergence, mice

Introduction

Anyone who analyzes person-data may run into a missing data problem. Missing data is not only ubiquitous, but treating it can also be tedious. If a dataset contains just one incomplete observation, statistical inferences are undefined and will not produce any results. To circumvent this, many statistical packages employ list-wise deletion by default (i.e., ignoring incomplete observations). Unfortunately, this *ad hoc* solution may yield wildly invalid results (Van Buuren 2018). An alternative is to *impute* (i.e., fill in) the missing values in the incomplete observations. Subsequently, statistical inferences can be performed on the completed dataset. By repeating this process several times, a distribution of plausible results may be obtained, which reflects the uncertainty in the data due to missingness. This technique is known as 'multiple imputation' (MI; Rubin 1976). MI has proven to be a powerful technique to yield unbiased and confidence

[1]Department of Methodology and Statistics, Utrecht University, Utrecht, The Netherlands

Corresponding author:
Hanne Oberman, Sjoerd Groenman building, Utrecht Science Park, Utrecht, The Netherlands.
Email: h.i.oberman@uu.nl

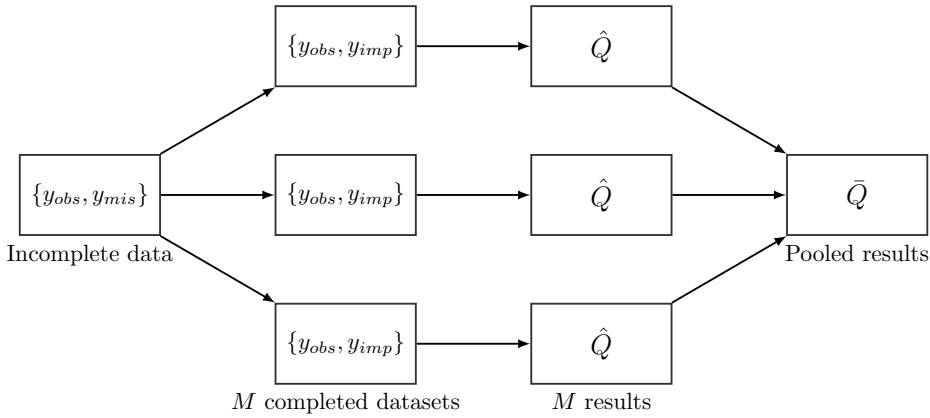

Figure 1. Scheme of the main steps in multiple imputation.

valid estimates of the true—but missing—data inference under many circumstances (Van Buuren 2018).

Figure 1 provides an overview of the steps involved with MI—from incomplete data, to M multiply imputed datasets, to M estimated quantities of interest $\hat{Q}$s, to a single pooled estimate $\bar{Q}$. Missing data in dataset y is imputed M times. The imputed data y_{imp} is combined with the observed data y_{obs} to create M completed datasets. On each completed dataset, the analysis of scientific interest is performed. The quantity of scientific interest (e.g., a regression coefficient) is denoted with Q. Since Q is estimated on each completed dataset, M separate $\hat{Q}$-values are obtained. These M values are combined into a single pooled estimate $\bar{Q}$.

A popular method to obtain imputations is to use the 'Multiple Imputation by Chained Equations' algorithm, shorthand 'MICE'(Van Buuren and Groothuis-Oudshoorn 2011). MICE is an iterative algorithmic procedure to draw imputations from the posterior predictive distribution of the missing values. This introduces a potential threat to the validity of the imputations: What if the algorithm has not converged? Are the implications then to be trusted? And can we rely on the inference obtained on the completed data? These are all open questions, because the convergence properties of iterative imputation algorithms have not been systematically studied (Van Buuren 2018). Moreover, there is no scientific consensus on how to evaluate convergence of MI algorithms (Takahashi 2017). Some default MICE techniques (e.g., 'predictive mean modeling') might not yield converged states at all (Murray 2018). Therefore, algorithmic convergence should be monitored carefully.

Currently, the recommended practice for evaluating convergence in iterative imputation algorithms is to visually inspect imputations for signs of non-convergence. **Add that this refers to chain means and variances already? It's insufficient that these are univariate, so multivariate state space of the algorithm may not be converged when univariates are ok. And Van Buuren (2018)'s suggestion to track**

user-defined statistics (e.g., regression coefficient) may be somewhat advanced for empirical researchers. Moreover, this is not model-independent, i.e., only apply to one substantive model, while one of the advantages of MI is that missing data problem and scientific problem are split. Then we can say 'we propose a novel…' and integrate this: Note that usually we only evaluate the convergence of univariate scalar summaries (e.g., chain means or variances). With these we cannot diagnose convergence of multivariable statistics (i.e., relations between scalar summaries). Van Buuren (2018, § 4.5.2) proposed to implement multivariable evaluation of the MICE algorithm through eigenvalue decomposition building on the work of MacKay and Mac Kay (2003). Eigenvalues of a covariance matrix are a measure of the data's covariance..

This method is insufficient on two counts: 1) it may be challenging to the untrained eye, and 2) only severely pathological cases of non-convergence may be diagnosed (Van Buuren 2018, § 6.5.2). Therefore, a quantitative, diagnostic evaluation of convergence would be preferred—yet not straightforward. Iterative imputation algorithms such as MICE are Markov chain Monte Carlo (MCMC) methods. In MCMC methods, convergence is not from a scalar to a point, but from one distribution to another. The values generated by the algorithm (e.g., imputed values) will vary even after convergence. Since MCMC algorithms do not reach a unique point at which convergence is established, diagnostics we may only identify signs of non-convergence (Hoff 2009). Several of such diagnostics exist, but it is not known whether these are appropriate for iterative imputation algorithms.

In this paper, we investigate how non-convergence in iterative imputation algorithms may be diagnosed, how well these methods perform, and at which point convergence may safely be assumed. For reasons of brevity, we only focus on the iterative imputation algorithm implemented in the popular `mice` package (Van Buuren and Groothuis-Oudshoorn 2011) in `R` (R Core Team 2020). The convergence properties of the MICE algorithm are investigated through model-based simulation. The results of this simulation study are guidelines for assessing convergence of MI algorithms, which will aid applied researchers in drawing valid inference from incomplete datasets. **add societal relevance: what is the consequence of non-convergence?**

Include sub-questions and add structure of the paper! How can non-convergence be identified diagnostically? Are common MCMC non-convergence diagnostics appropriate for MICE? And if so, which threshold should be used to diagnose non-convergence? How many iterations are sufficient/needed to be able to diagnose non-convergence? Are the default number of iterations sufficient (i.e., 5 in mice, 10 in SPSS and Stata, 30 in mi)? **How severe is it when the algorithm has not converged? And what are guidelines for practice? Can the parameter of interest, estimand Q, be correct when the algorithm is not (yet) converged, and vice versa?** *(Maybe add these too? What are the effects of continued iterations on estimates, predictions and inferences? Do the answers differ with varying missingness proportions? That is, we vary the nr of iterations and the missingness proportion because we assume that the algorithm has not reached convergence at $t = 1$, and performs worse with moderate to high missingness (not so much with little or a LOT of missingness).)*

Some notation

Let y denote an $n \times k$ matrix containing the data values on k variables for all n units in a sample. The data value of unit i ($i = 1, 2, \ldots, n$) on variable j ($j = 1, 2, \ldots, k$) may be either observed or missing. **The number of units i in dataset y with at least one missing data value, divided by the total number of units n, is called the missingness proportion in p_{mis}.** The collection of observed data values in y is denoted by y_{obs}; the missing part of y is referred to as y_{mis}. For each datapoint in y_{mis}, we sample $M \times T$ times plausible values, where M is the number of imputations ($m = 1, 2, \ldots, M$) and T is the number of iterations ($t = 1, 2, \ldots, T$). The collection of samples between the initial value (at $t = 1$) and the final imputed value (at $t = T$) will be referred to as an 'imputation chain'. **Add: θs are scalar summaries of interest in the iterative algorithm (e.g., chain means; the average of the imputed values in each imputation chain).**

Identifying non-convergence

The goal of imputation through e.g. MICE is to obtain a multivariate distribution by iterating over the sequence of univariate imputations. Traditionally, the inspection of the multivariate convergence is done by visually inspecting the univariate chain means and variances over the iterations, or by tracking the $\widehat{R}$ and autocorrelation. The downside to these approaches is that they either focus on the univariate state space, or primarily track the change over the iterations of a multivariate outcome conform the scientific model of interest. Focusing on the convergence of these outcome parameters may influence this procedure in the sense that the model of evaluation favors the model of interest. Ideally, one would like to evaluate a model-independent parameter that summarizes the multivariate nature of the data. We propose eigenvalue as such a parameter, because it. . . .

Mixing and stationarity have histroically been inspected visually, by evaluating traceplots of scalar summaries of interest (θs; e.g., chain means and chain variances). As Van Buuren (2018) describes, users can also specify a model-specific scalar summary (e.g., a regression coefficient). A user-specified scalar summary, however, is not universal to all complete data problems. Therefore, as inspired by (MacKay and Mac Kay 2003), we propose a θ that summarizes the multivariate state space of the algorithm. Namely, the first eigenvalue of the variance-covariance matrix of the M completed datasets. **The first eigenvalue has the appealing property that is not dependent on the substantive model of interest.**

Since convergence of iterative imputation algorithms is in distribution, there is not a unique point at which the algorithm reaches a converged state. We can only evaluate scalar summaries of the multivariate state space of the algorithm, θs. The recommended scalar summaries to evaluate are chain means and chain variances. Additionally, researchers may "monitor some statistic of scientific interest" (Van Buuren 2018, § 6.5.2). Because of the issues mentioned above, we propose a novel θ to monitor: the first eigenvalue of the variance-covariance matrix of the completed data. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_j$ be the eigenvalues of Σ in each of the

M **completed datasets** y_{obs}, y_{imp}**.** λ_1 **is measure that summarises the covariances in the completed datasets.**

There are two requirements for convergence of iterative algorithms: mixing and stationarity (Gelman et al. 2013). Without mixing, imputation chains do not intermingle nicely, **indicating that ....** Without stationarity, there is trending within imputation chains, which implies that further iterations would yiled a different set of imputations.

To illustrate what non-convergence looks like in MI, we reproduce the example from van Buuren (2018, § 6.5.2). Figure 2 shows the traceplot for one of the variables in the example. we see the average of the imputed values for a variable j in y_{imp}. The average imputed values in the left hand plot are about a magnitude 2 lower than the right hand plot. So non-convergence has a impact on the average imputed value for j in $y_{imp,m}$. This difference (presumably) translates to bias in the pooled estimate $\bar{\mu}_j$ as well. And maybe also to higher order statistics, e.g. variances, covariances, etc. Therefore, it's important to converge. **Explain what we see, namely example by Van Buuren (2018) reproduced, showing the traceplots of chain means for some variable. The first plot is typical convergence of MICE, the second is pathological non-convergence because of a mis-specified imputation model. Each line is an imputation. In the first plot, the chains intermingle nicely and there is little to no trending. In the second plot, there is a lot of trending and some chains do not intermingle. Importantly, the chain means at the last iteration (the imputed value per m) are very different between the two plots. The algorithm with the mis-specified model yields imputed values that are on average a magnitude two larger than those of the typically converged algorithm. This shows the importance of reaching converged states in iterative imputation algorithms.**

Methods under evaluation

Non-stationarity within chains may be diagnosed with e.g., autocorrelation (AC; Schafer 1997; Gelman et al. 2013), numeric standard error ('MC error'; Geweke 1992), or Raftery and Lewis's (1991) procedure to determine the effect of trending on the precision of estimates. A widely used diagnostic to monitor mixing between chains is the potential scale reduction factor $\widehat{R}$ ('Gelman-Rubin statistic'; Gelman and Rubin 1992). With a recently proposed adaptation, $\widehat{R}$ might also serve to diagnose non-stationarity, but this has not yet been thoroughly investigated (Vehtari et al. 2019). Therefore, use we $\widehat{R}$ and AC to evaluate mixing and stationarity separately, as recommended by e.g., Cowles and Carlin (1996).

Potential scale reduction factor An updated version of $\widehat{R}$ has been proposed by Vehtari et al. (2019) (p. 5). **This version may be suitable for iterative imputation.** Let M be the total number of chains, T the number of iterations per chain, and θ the scalar summary of interest (e.g., chain mean or chain variance). For each chain ($m = 1, 2, \ldots, M$), we estimate the variance of θ, and average these to obtain within-chain variance W.

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

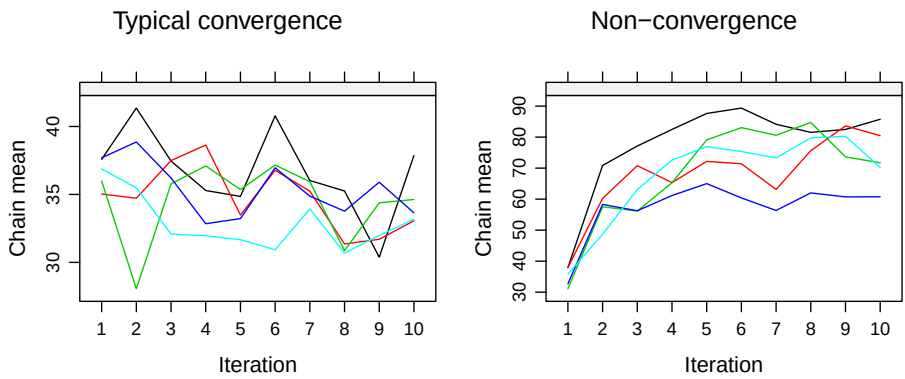

Figure 2. Typical convergence versus pathological non-convergence. Please note that this is the same data with a different imputation model, leading to different imputations (see the y axis!). Wel min of meer dezelfde origin op iteration 1.

We then estimate between-chain variance B as the variance of the collection of average θ per chain.

$$B = \frac{T}{M-1}\sum_{m=1}^{M}\left(\bar{\theta}^{(\cdot m)} - \bar{\theta}^{(\cdot\cdot)}\right)^2, \text{ where } \bar{\theta}^{(\cdot m)} = \frac{1}{T}\sum_{t=1}^{T}\theta^{(tm)}, \bar{\theta}^{(\cdot\cdot)} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}^{(\cdot m)}.$$

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

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

We can interpret $\widehat{R}$ as potential scale reduction factor since it indicates by how much the variance of θ could be shrunken down if an infinite number of iterations per chain would be run (Gelman and Rubin 1992). This interpretation assumes that chains are 'over-dispersed' at $t = 1$, and reach convergence as $T \to \infty$. Over-dispersion implies that the initial values of the chains are 'far away' from the target distribution and each other. When all chains sample independent of their initial values, the mixing component of convergence is satisfied, and $\widehat{R}$-values will be close to one. High $\widehat{R}$-values thus indicate non-convergence. The conventionally acceptable threshold for convergence was $\widehat{R} < 1.2$ (Gelman and Rubin 1992). More recently, Vehtari et al. (2019) proposed a more stringent threshold of $\widehat{R} < 1.01$.

Autocorrelation Following the same notation, we define autocorrelation as the correlation between two subsequent θ-values within the same chain (Lynch 2007, p. 147). In this study, we only consider AC at lag 1, i.e., the correlation between the t^{th} and $t+1^{th}$ iteration of the same chain.

$$AC = \left(\frac{T}{T-1}\right) \frac{\sum_{t=1}^{T-1}(\theta_t - \bar{\theta}^{(\cdot m)})(\theta_{t+1} - \bar{\theta}^{(\cdot m)})}{\sum_{t=1}^{T}(\theta_t - \bar{\theta}^{(\cdot m)})^2}.$$

We can interpret AC-values as a measure of stationarity. If AC-values are close to zero, there is no dependence between subsequent samples within imputation chains. Negative AC-values indicate divergence within imputation chains. **Subsequent sampled values within each imputation chain are less alike.** Positive AC-values indicate recurrence. If θ-values of subsequent iterations are similar, trending may occur. Negative AC-values show no threat to the stationarity component of convergence. On the contrary even—negative AC-values indicate that θ-values of subsequent iterations diverge from one another, which may increase the variance of θ and speed up convergence. As convergence diagnostic, the interest is therefore in positive AC-values. **Maybe remove:** Moreover, the magnitude of AC-values may be evaluated statistically, but that is outside of our scope.

In practice

Upon convergence, $\widehat{R} = 1$ and $AC = 0$, which are unlikely thresholds for MCMC algorithms, because of its convergence to a distribution. In practice, non-convergence is usually diagnosed when $\widehat{R} > 1.2$ or 1.1 or even 1.01. And a t-test is performed to assess whether AC is significantly different from zero.

We assume that complete convergence, defined as $\widehat{R} = 1$ and $AC = 0$, will not be observed. Because even in the most converged state, the algorithm will show some signs of non-mixing and non-stationarity. Upon *sufficient* convergence, the imputation chains will intermingle such that the only difference between the chains is caused by the randomness induced by the algorithm ($\widehat{R} > 1$), and there may be some dependency between subsequent iterations of imputation chains ($AC > 0$).

Before using them in the simulation, the two methods show at least be able to distinguish between the two algorithms plotted in Figure 2. From visual inspection we know that the typical convergence has some signs of non-mixing around iteration 2, and little trending. In the non-convergence situation, there is a lot of trending upto iteration 6, after which the chains reach a somewhat more stationary state. Mixing in this situation gets worse from the first iteration onward, and gradually gets better around iteration 7. To assess whether $\widehat{R}$ and AC may be appropriate non-convergence identifiers for iterative imputation algorithms, the following condition must hold: the methods indicate worse performance for the mis-specified model with pathological non-convergence (i.e., higher $\widehat{R}$- and AC-values than the typical performance). And additionally the methods should reflect the increasing convergence in the typical convergence situation as the number of iterations goes up.

In Figure 3A, the chain means from Figure 2 are plotted again, now together—as are the non-convergence diagnostics. Panel B shows $\widehat{R}$ as computed by implementing Vehtari et al. (2019) 's recommendations. As required, $\widehat{R}$ indicates less signs of non-convergence as the number of iterations goes up in the typical convergence situation**. The superior performance of the typical convergence over the pathological non-convergence is less prominent, even flipped for $t < 4$. Panel C displays the AC as computed with the R function `stats::acf()`. When we look at this panel, we conclude something weird. The AC-values indicate equal performance (up-to $t = 5$) for the typical convergence and the pathological non-convergence, while there is obvious trending in the latter. Moreover, the best convergence (as indicated by the lowest AC-value) is observed at $t = 2$, but looking at the chain means in panel A, there should be some signs of trending up-to iteration number seven. After consulting the documentation on stats::acf() (R Core Team 2020), we conclude that this implementation of AC is not suitable for iterative imputation algorithms. There is a correction factor for a mathematical shortcut that works in the limit. According to Box et al. (2015), the function works when the number of iterations > 50. The default number of iterations in iterative imputation, however, is often much lower. Therefore, we compute AC manually, see panel D. The AC values in this plot do meet the requirements (...) and will therefore be used in the simulation study.**

acf is niet bedoeld om weinig iteraties te... niet een fout in acf, alleen hier niet van toepassing

Simulation study

The aim of the simulation study is to evaluate the impact of inducing non-convergence in the MICE algorithm on several quantities of scientific interest Q. Subsequently we will evaluate how well $\widehat{R}$ and AC perform in identifying the effects of non-convergence (**add thetas**). And finally, we will formulate an informed advice on the requirements to safely assume *sufficient* convergence in practice. That is, we assume that the algorithm is *sufficiently* converged when each estimate $\bar{Q}$ is an unbiased and confidence valid estimate of the corresponding estimand Q. (**add to formulate advice on thetas?**)

Non-convergence will be induced by: 1) increasing the missingness proportion p_{mis} in dataset y incrementally, and 2) terminating the imputation algorithm at a varying number of iterations T. The first set of simulation conditions—the missingness proportions—is chosen to reflect the difficulty of the missingness problem. The underlying assumption is that low missingness proportions lead to quick algorithmic convergence, since there is a lot of information in the observed data. Higher missingness proportions should yield slower convergence. Unless, however, the fraction of missing information is so high that the random component in the imputation algorithm outweighs the information in the observed data. Then, convergence to a stable but highly variable state may be reached instanteneously. We assume that the incremental missingness proportions in our study will result in a corresponding increase in signs of non-convergence.

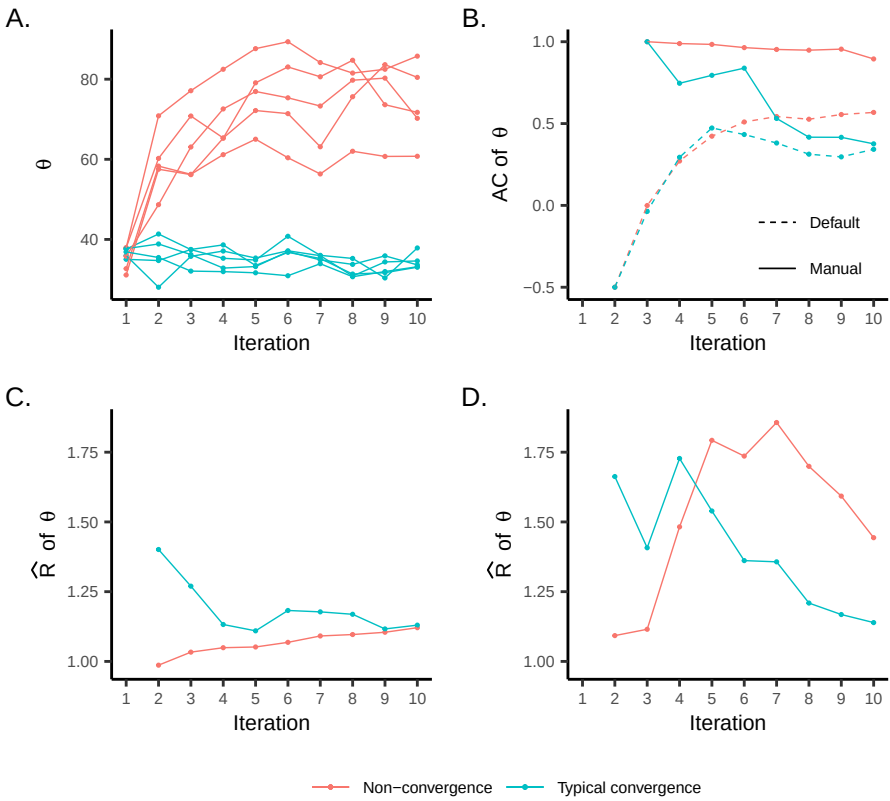

Figure 3. Convergence diagnostics applied on the imputation algorithms of Figure 2. θ = chain mean in $y_{imp,m}$.

The assumption inherent to the second set of simulation conditions—the number of iterations—is that terminating the imputation algorithm too early causes non-convergence. Generally, the algorithm will not reach convergence if $T = 1$, because the imputed values in the first iteration (at $t = 1$) depend on the initial values of the algorithm (which are sampled randomly from the set of observed datapoints). As the number of iterations increases, the imputation chains should become independent of the initial values, until the point at which adding an extra iteration does not lead to a more converged state. We assume that we can induce non-convergence at least in conditions where T is smaller than the default number of iterations in `mice`, $T = 5$.

Hypotheses

1. We expect that simulation conditions with a high missingness proportion p_{mis} and a low number of iterations T will result in biased, invalid estimates of the quantities of scientific interest, Qs.
2. We hypothesize that $\widehat{R}$ and AC will correctly identify signs of non-convergence in those simulation conditions where the $\bar{Q}$s are *not* unbiased and confidence valid estimates of the Qs.
3. We hypothesize that the recommended thresholds to diagnose non-convergence with $\widehat{R}$ ($\widehat{R} > 1.2$, $\widehat{R} > 1.1$, and $\widehat{R} > 1.01$) may be too stringent for iterative imputation applications. In an empirical study, where $\widehat{R}$ was used to inform the required imputation chain length, it took as many as 50 iterations to overcome the conventional non-convergence threshold $\widehat{R} > 1.2$. Yet, scientific estimates were insensitive to continued iteration after $t > 5$ (Lacerda et al. 2007). We therefore suspect that $\widehat{R}$ may over-estimate signs of non-convergence in iterative imputation algorithms, compared to the bias the estimates. In contrast to this, it may occur that signs of non-convergence are under-estimated by $\widehat{R}$, in exceptional cases where the initial values of the algorithm are not appropriately over-dispersed (Brooks and Gelman 1998, p. 437). In e.g. `mice`, initial values are chosen randomly from the observed data, hence we cannot be certain of over-dispersion in the initial values. In practice, we do not expect this to cause problems for identifying non-convergence with $\widehat{R}$.
4. We expect that high AC values are implausible in iterative imputation algorithms with typical convergence. That is, after only a few iterations, the randomness induced by the algorithm should effectively mitigate the risk of dependency within chains.
5. **Something about thetas**

Set-up

Convergence of the MICE algorithm is investigated through model-based simulation in `R` (version 3.6.3; R Core Team 2020). The simulation set-up is summarized in the pseudo-code below. The complete `R` script of the simulation study is available from github.com/gerkovink/shinyMice.

```
# pseudo-code of simulation
1. simulate data
for (number of simulation runs from 1 to 1000)
 for (missingness proportions 5%, 25%, 50%, 75% and 95%)
  2. create missingness
  for (number of iterations from 1 to 100)
   3. impute missingness
   4. perform analysis of scientific interest
   5. compute non-convergence diagnostics
   6. pool results across imputations
```

```
   7. compute performance measures
 8. combine outcomes of all missingness proportions
9. aggregate outcomes of all simulation runs
```

Data-generating mechanism. In this study, sampling variance is not of interest. Therefore, a single complete dataset may serve as comparative truth in all simulation repetitions (Vink and van Buuren 2014). The data-generating mechanism is a multivariate normal distribution, representing person-data on three predictor variables in a multiple linear regression problem **(from an unspecified social scientific field of study)**. Let the predictor space be defined as

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 12 \\ 3 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 4 & 4 & 1.8 & 0 \\ 4 & 16 & 4.8 & 0 \\ 1.8 & 4.8 & 9 & 0 \end{pmatrix}\right].$$

A finite population of $N = 1000$ is simulated using the `mvtnorm` package (Genz and Bretz 2009). Subsequently, a fourth variable is constructed as outcome Y. For each unit $i = 1, 2, ..., N$, let

$$Y_i = 1 + 2X_{1i} + .5X_{2i} - X_{3i} + \epsilon_i,$$

where $\epsilon \sim \mathcal{N}(0, 100)$. From the complete set, we obtain the true values of several quantities of scientific interest, Qs. **MOVE THIS NEXT PART TO TOP OF SIMULATION SECTION?** Qs are the targets that will be estimated in each simulation repetition, after first *amputing* the complete data with varying missingness proportions, then imputing the missingness with a varying number of iterations, then estimating the Qs on the M completed datasets, and finally pooling the M $\hat{Q}$s to obtain a $\bar{Q}$ for each Q. **Add that we try to predict the effect of non-converegence (i.e., the difference between Q and $\bar{Q}$) with the convergence diagnostics $\widehat{R}$ and AC, by appliying these methods to several scalar summaries of the state space of the algorithm, θs.**

Scientific estimands. We consider four types of Qs that are often of interest in empirical research. Namely, two descriptive statistics: the mean μ_j and standard deviation σ_j of each variable $j = Y, X_1, X_2, X_3$. We also consider the regression coefficients of the predictors: β_1, β_2, and β_3 in regression equation

$$Y' = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3,$$

where Y' is the expected value of the outcome. And finally, the proportion of variance explained by the set of predictors: coefficient of determination $r^2 \times 100$ **(note that lower case r is used to avoid confusion with non-convergence diagnostic $\widehat{R}$)**. The true values of these scientific estimands are calculated on the complete data with the R functions `base::mean()`, `stats::sd()`, and `stats::lm()` (R Core Team 2020).

Methods. In each simulation repetition, we ampute the complete dataset five times to obtain five different incomplete datasets, ys. The ratio between y_{obs} and y_{mis} in each y depends on the missingness proportion: $p_{mis} = .05, .25, .5, .75, .95$. We ampute the

complete data with the `mice` package (function `mice::ampute()`; Van Buuren and Groothuis-Oudshoorn 2011). All missingness is multivariate, and conform a 'missing completely at random' missingness mechanism (Rubin 1987). I.e., the probability to be missing is the same for all $N \times k$ cells in y, conditional on the missingness proportion p_{mis}.

Missing datapoints in the incomplete datasets, ys, are imputed with the `mice()` function (Van Buuren and Groothuis-Oudshoorn 2011). All imputation procedures are performed with Bayesian linear regression imputation, and five imputation chains ($M = 5$). Because $M = 5$, each run of the `mice()` function results in five sets of imputations, $y_{imp,m}$s, where $m = 1, 2, ..., 5$. The number of iterations in each run of the algorithm varies between simulation conditions ($T = 1, 2, ..., 100$).

From the 5 imputations $y_{imp,m}$: get chain means and chain variances from imputation object to use as scalar summary θ in convergence diagnostics. Then use `mice::complete()` to combine the imputed data $y_{imp,m}$ with the observed data y_{obs}, resulting in five completed datasets $\{y_{obs}, y_{imp,m}\}$.

On the 5 completed datasets $\{y_{obs}, y_{imp,m}\}$, we:

- Estimate $\hat{Q}$s for $Q = \mu$ and $Q = \sigma$ with `base::mean()`, and `stats::sd()` (R Core Team 2020). Then pool the $\hat{Q}$s with `base::mean()` to get $\bar{Q}$s.
- Compute λ_1, to use as θ in convergence diagnostics. By definition, the first eigenvalue of the variance-covariance matrix, λ_1, is equal to the variance of the principal component solution of each $\{y_{obs}, y_{imp,m}\}$, which we calculate with `stats::princomp()` (R Core Team 2020).
- Perform multiple linear regression with `stats::lm()` (R Core Team 2020) to get the $\hat{Q}$s for $Q = \beta$ and $Q = r^2$. Pool the $\hat{Q}$s conform Vink and van Buuren (2014) to get $\bar{Q}$s for the βs and r^2 (**check pooling of r^2, instead of `mice::pool.r.squared()`**). Additionally, use the $\hat{Q}$s of $Q = \beta$ as θ to compute convergence diagnostics.

We apply non-convergence diagnostics $\widehat{R}$ and AC to each scalar summary θ. We calculate $\widehat{R}$ by implementing Vehtari et al.'s (2019) recommendations, and AC as the correlation between the t^{th} and the $(t+1)^{th}$ iteration.

Performance measures. We assess the impact of inducing non-convergence in the iterative imputation algorithm by comparing $\bar{Q}$s with Qs. For each Q, we calculate bias as $\bar{Q} - Q$. For the regression coefficients $Q = \beta_{1,2,3}$, we also compute the empirical coverage rate (CR). CR is defined as the percentage of simulation repetitions in which the 95% confidence interval (CI) around $\bar{Q}$ covers the true estimand Q. Let

$$\text{CI} = \bar{Q} \pm t_{(M-1)} \times SE_{\bar{Q}},$$

where $t_{(M-1)}$ is the quantile of a t-distribution with $M - 1$ degrees of freedom, and $SE_{\bar{Q}}$ is the square root of the pooled variance estimate. If we obtain nominal coverage (CR = 95%), we can conclude that the $\bar{Q}$s are conficence valid estimates of the Qs. Finally, we inspect CI width (CIW): the difference between the lower and upper bound of the 95%

confidence interval around $\bar{Q}$. CIW is of interest because too short CIs indicate underestimation of the variance in $\bar{Q}$, which may yield spurious inferences.

With these performance measures, we also evaluate how well the non-convergence diagnostics $\widehat{R}$ and AC can identify simulation conditions in which the $\bar{Q}$s are *not* unbiased, confidence valid estimates of the Qs. **Rephrase and add corresponding Qs:** The two methods are applied to four scalar summaries θ: chain means (i.e., the mean in each $y_{imp,m,j}$), chain variances (i.e., the variance in each $y_{imp,m,j}$), the estimated regression coefficients $\beta_{1,2,3}$ in each $\{y_{obs}, y_{imp,m}\}$ (i.e., a user-defined statistic of scientific interest $\hat{Q}$), and the first eigenvalue of the variance-covariance matrix in each $\{y_{obs}, y_{imp,m}\}$, λ_1.

Results

Add to this section: 1) emphasize that the plots are averages across repetitions, not within MICE; 2) sample effects due to single complete dataset?? 3) Add more info about figure legends and axes.)

Primarily, we want to know what the effect is of inducing non-convergence on the estimated Qs. Specifically, we are interested in the bias of $\bar{Q}$ for all Qs and the confidence validity of $\bar{Q}$ for $Q = \beta$. Therefore, we plot these performance measures against the simulation conditions (the number of iterations T and the missingness proportion p_{mis}). We look at averages across simulation repetitions.

Secondly, we want to know if conditions in which the Qs are affected by non-convergence are also identified by the non-convergance diagnostics $\widehat{R}$ and AC. We do this, again, by looking at the performance measures for the Qs, but now we evaluate these against what $\widehat{R}$ and AC indicate about signs of non-the convergence in the θs.

- Is there more bias in conditions where non-convergence was induced? (NOPE for univariates, YES for multivar.).
- Is the point of *sufficient* convergence (at which the $\bar{Q}$s are unbiased and confidence valid estimates of the Qs) the same point as identified by the non-convergence diagnostics? (NOPE, they are too conservative).**
- Is one of the thetas better? (NAH, doesn't really matter).
- What should the thresholds be? (LOWER than recommended).
- And the default nr of iterarions? (MAYBE increase to 10?).

For reasons of brevity, we only discuss the convergence diagnostics for the Qs with the worst performance in terms of bias. Bias in the descriptve statistics ($Q = \mu$ and $Q = \sigma$) was most pronounced in the outcome variable Y. Of the regression coefficients, $Q = \beta$, the largest bias was observed in the effect of X_1 on Y, β_1. Results for $Q = r^2$ are reported as-is. In the figures, however, we only present simulation conditions where $T \leq 50$. Full results are available from **[link github repo]**.

In general

As expected, conditions with a higher proportion of missingness and/or a lower number of iterations show more signs of non-convergence. I.e., on average these conditions portet

more extreme bias in the estimated Qs, and non-nominal coverages. Roughly speaking this means that MICE is indeed not converged at $t = 1$, and converges gradually as t increases to T. The point at which an additional iteration does not lead to an improvement of the estimates depends on the difficulty of the missingness problem—between XYZ and XYZ. This means that the algorithm has converged sufficiently (**under the current specifications**).

As required for a method to diagnose non-convergence, the values of $\widehat{R}$ follow a trend similar to the performance measures. **Add split between convergence diagnostics versus performance measures, and convergence diagnostics versus simulation conditions.** $\widehat{R}$ values are generally lower in conditions with a higher number of iterations, and somewhat higher in conditions with a higher percentage of missing data. Autocorrelation values were indeed decreasing with a higher number of iterations, and lower in conditions with less missingness.

Evaluation with the performance measures shows that $\widehat{R}$ is conservative: it identifies signs of non-convergence in conditions where the $\bar{Q}$s are unbiased and/or confidence valid estimates of the Qs.

Means

Message: after one iteration (T=2), the algorithm is as well converged as the missingness proportion allows. Sooner even with lower missingness proportions (T=1 might be ok for p_{mis} = 5 and 25). Number of it has little influence on bias in $\hat{Q}$, only in first iteration some bias in missingness conditions of 25 and up. There is a clear different between the missingness conditions. The higher the missingness prop, the larger the magnitude of the bias. For this Q, t=2 would be sufficient. And only for p_{mis} = 5 the $\bar{Q}$s are unbiased. All other p_{mis} conditions have bias irrespective of the number of iterations.

$\widehat{R}$ differs a lot across iteration conditions. But not structurally for missingness proportions. Highest $\widehat{R}$s for the condition T=2, while the most bias is for (T=1), but there is no $\widehat{R}$ for that by definition. But for missingness, 25% seems the most signs of non-conv, while it has one of the least bias. Threshold of 1.2 reached in condition it = 3, p_{mis} = 5 and 75, and then for it>6, p_{mis} = 5 and 50, and T>10 for all p_{mis}. Threshold 1.1 reached in in T=12, p_{mis} = all but 75, 1.01 only reached in T = 93, p_{mis} = 75. While for this Q, T=2 would be sufficient.

AC mainly shows effects of the set of iteration conditions. AC indicates less signs of non-convergencein conditions with higher number of iterations. For iteration conditions where T<10, there are also some effects of the missingness proportions. The p_{mis} condtions 25 and 95 % indicate the most signs of non-convergence. For p_{mis} = 95% this corresponds to the bias in Qbar. For 25% miss it doesn't. The thresholds defined by the critical value is only met in the conditions it=3, p_{mis} = 5,75, and 95%.

Standard deviations

Message: even with possible effect of initial values (T=1), the algorithm is as well converged as the missingness proportion allows. For $Q = \sigma_Y$ there is no effect visible

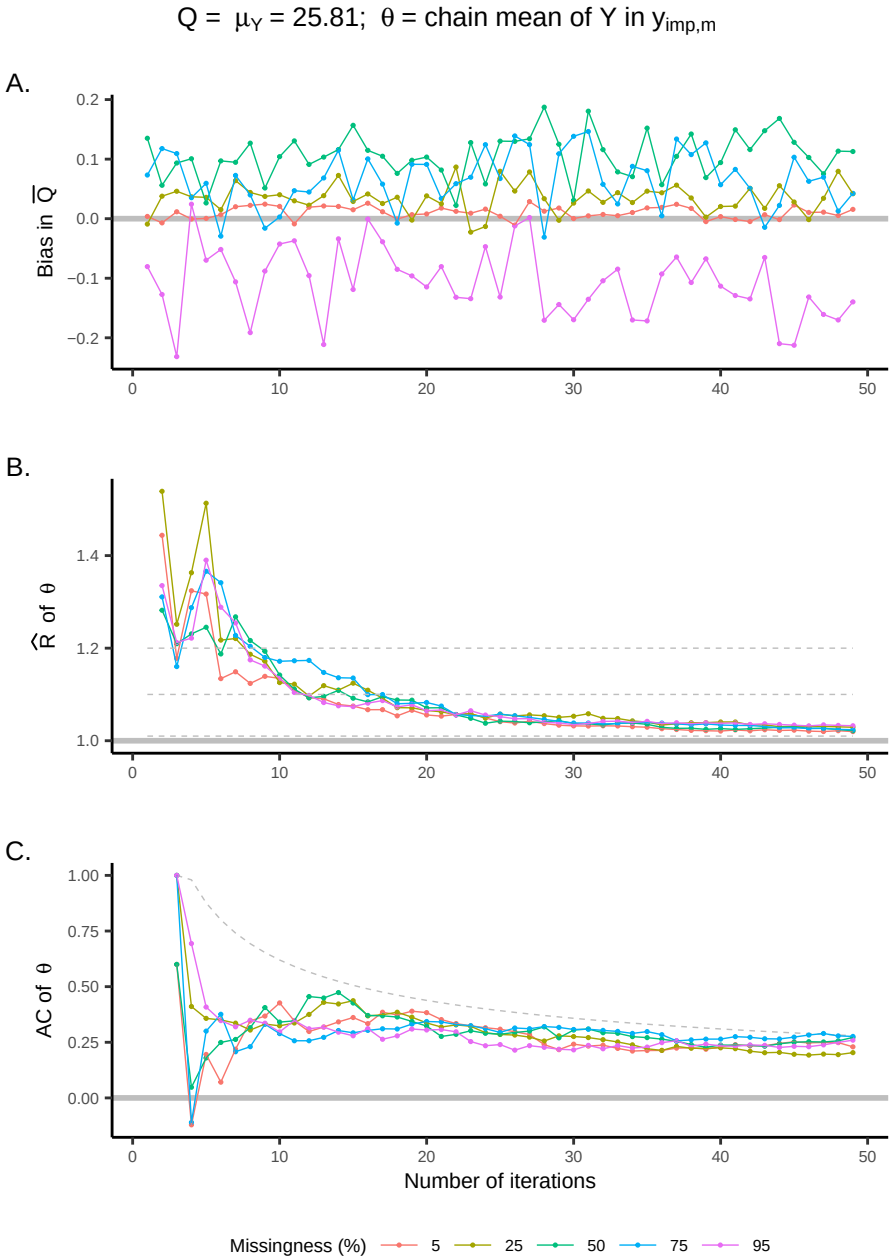

Figure 4. Convergence diagnostics chain mean.

of the iteration conditions, but clearly for the p_{mis} conditions. Most bias for p_{mis} = 95, followed by p_{mis} = 50 and 75. For the least possible bias we may use any number of iterations $T \geq 1$, but none of the p_{mis} results in unbiased $\bar{Q}$s.

There is a clear effect of the number of iterations on $\widehat{R}$. Conditions with lower Ts have more signs of non-conv, as identified by $\widehat{R}$. There is little difference between missingness conditions. Only for conditions where $T < 5$ the missingness proportions seem to affect whether $\widehat{R}$ indicates non-convergence The bias in $\bar{Q}$ seems unrelated to the magnitude of $\widehat{R}$. Conditions with more bias (p_{mis} = 50,75,95) are not identified to have more signs of non-convergence. The thresholds are therefore useless (because depend mostly on T, not so much p_{mis}).

AC is affected by both t and p_{mis} conditions. But conditions with worst bias (p_{mis} = 50, 75, 95) are not identified as worst converging according to AC. Moreover, only for T =3 the AC is significantly different from zero.

Regression coefficients

Message: after four iterations (T=5), the algorithm is as well converged as the missingness proportion allows. Sooner even with lower missingness proportions (T=1 might be ok for p_{mis} = 5 and 25). The figure for $Q = \beta_1$ shows that the $\bar{Q}$s are affected by both the number of iterations and the missingness proportion. For T =1, the magnitude of the bias (almost) follows the order of the missingness proportions. More missingness causes more bias in the $\bar{Q}$. In general, a lower nr of iterations results in more bias, but this depends on the missingness proportion. For p_{mis} = 95, there is more bias in conditions where T<6. When p_{mis} = 50, it's T<3. For p_{mis} = 75 and 25, it's only for T<2. And conditions where p_{mis} = 5 are unaffected on average. However, none of the conditions result in completely unbiased estimates of Q.

We should see this in the conv diag: higher $\widehat{R}$ and AC for p_{mis} = 75 and 95, and for condtions where $T < 2 - 6$, depending on p_{mis}.

$\widehat{R}$ depends somewhat on the missingness proportion. For conditions where T<10, the $\widehat{R}$ values follow the p_{mis} more or less. But what we actually want to know is whether they are indicators of bias. Conditions with the worst bias (high p_{mis}, low T) have the largest $\widehat{R}$ values. The thresholds are not appropriate to diagnose sufficient convergence. All thresholds over-estimate the signs of non-convergence compared to the bias in Qbar. However, a higher threshold than 1.2 may result in terminating the algorithm after 3 iterations, which is insufficient for the p_{mis} = 95 condition. Threshold of 1.25 or 1.3 could work for the other p_{mis} conditions.

AC is very affected by the p_mis. The two conditions with the most bias (p_{mis} = 75 and 95) are the only ones identified as non-converging, as they should. However, the p_{mis} = 75 condition is only identified in the T =3 condtion, while the bias is the least pronounced there. And the AC only identifies non-convergence in the p_{mis} = 95 conditions where T>10. In the default situation, T =5, this would be overlooked. Also, the AC shows the opposite of what you would expect from a theta that is based on the observed data: shouldn't there be higher autocorrelations in conditions with comparatively more observed data in the completed datasets? I don't understand this.

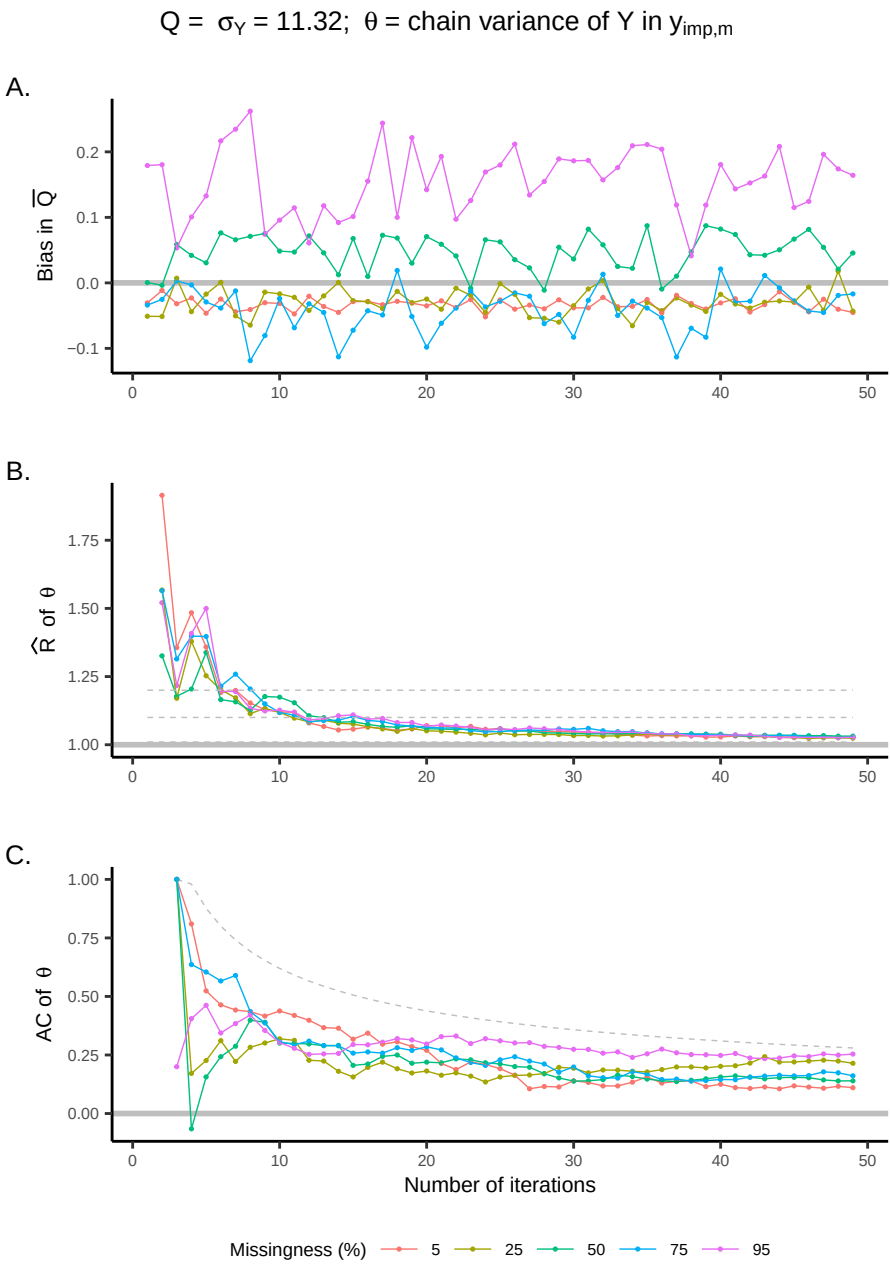

Figure 5. Convergence diagnostics chain variance.

Add how interesting it is because we literally track Q and are still not ok.

Variance explained

Message: after two iterations (T=3), the algorithm is as well converged as the missingness proportion allows. Sooner even with lower missingness proportions (T=1 might be ok for p_{mis} = 5 and 25). Bias is the worst for conditions where p_{mis} = 95, T <3. Followed by p_{mis} = 75, T =1; and p_{mis} = 50, T = 1. Conditions where p_{mis} = 25 or 5 seem unaffected by the number of iterations. Minimal bias is obtained in all iteration conditions. We should see this in $\widehat{R}$ and AC: worst convergence in p_{mis} = 95 and 50; some non-convergencein conditions where p_{mis} = 75 and T =1, and no non-convergencein conditions where p_{mis} = 25 or 5. The sign of the bias is logical: when there is less info in the data, the relations between the predictors and the outcome is under-estimated.

$\widehat{R}$ behaves oppositely to what we wanted. The lowest $\widehat{R}$ values for conditions where T<10 are observed for the two missingness conditions with the worst bias. However, the highest $\widehat{R}$ observed is indeed for the missingness condition with the worst bias, p_{mis} =95. But only at T =5 (while bias is no worse than anywhere else in this p_{mis} condition, conditional on iterations condition T>2). So neither of the thresholds works perfectly.

AC shows a similar trend to the biases: more signs of non-convergencefor higher p_{mis} and lower T. The threshold, however, performs less well. It identifies signs of non-convergence for all conditions where $T = 3$, irrespective of p_mis. Then nothing, and then for p_{mis} =95, T>7, and p_{mis} = 50, T >14, and p_{mis} = 75, T>20. This does not corresond to the bias in $\bar{Q}$. Maybe use AC>.5 at T =5 as threshold?

Discussion

In short, univariate estimates seem robust against terminating the algorithm early: There is no clear effect of the number of iterations on the bias in the estimated means and standard deviations. The bias in these Qs only depends on the p_{mis}, with higher p_{mis} resulting in more bias. Yet, the convergence diagnostics indicate that the algorithm did not reach a completely converged state (yet).

$\widehat{R}$ and autocorrelation indicate that algorithmic convergence may only be reached after twenty or even forty iterations, while unbiased, confidence valid estimates may be obtained with as little as one iteration. These results are in agreement with the simulation hypothesis: $\widehat{R}$ over-estimates the severity of non-convergence when applied to MI procedures.

With that, we show that MICE can lead to correct outcomes when they have not converged according to two common conv diags. This may be due to the methods (and their thresholds) or due to the Qs (descriptives and multivariate linear regression, not higher dimensional/more complex Qs). More 'complicated' Qs (e.g., higher-order effects or variance components) might show bias, under- or over-coverage at higher T, as indiated by $\widehat{R}$ and AC. Add what %miss has to do with it.

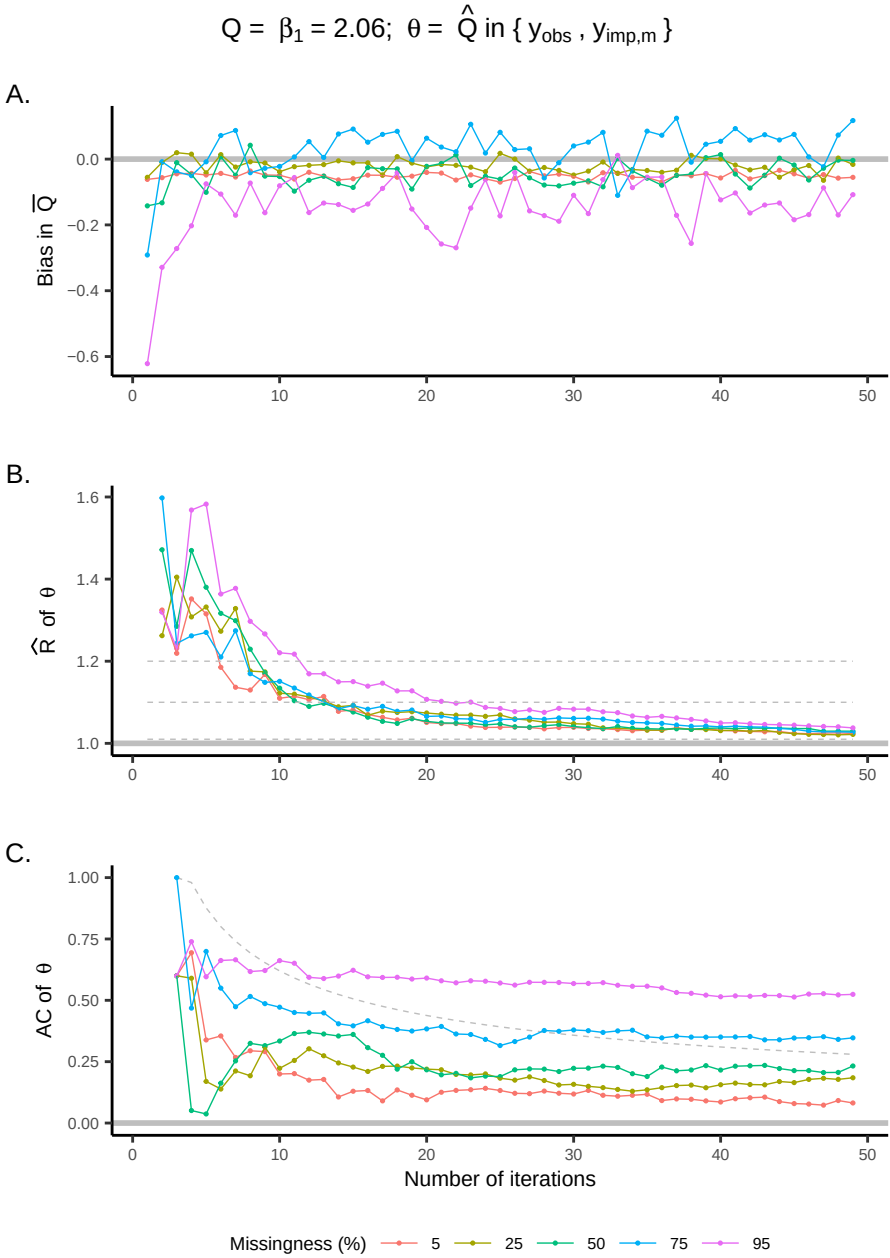

Figure 6. Convergence diagnostics regression coefficient.

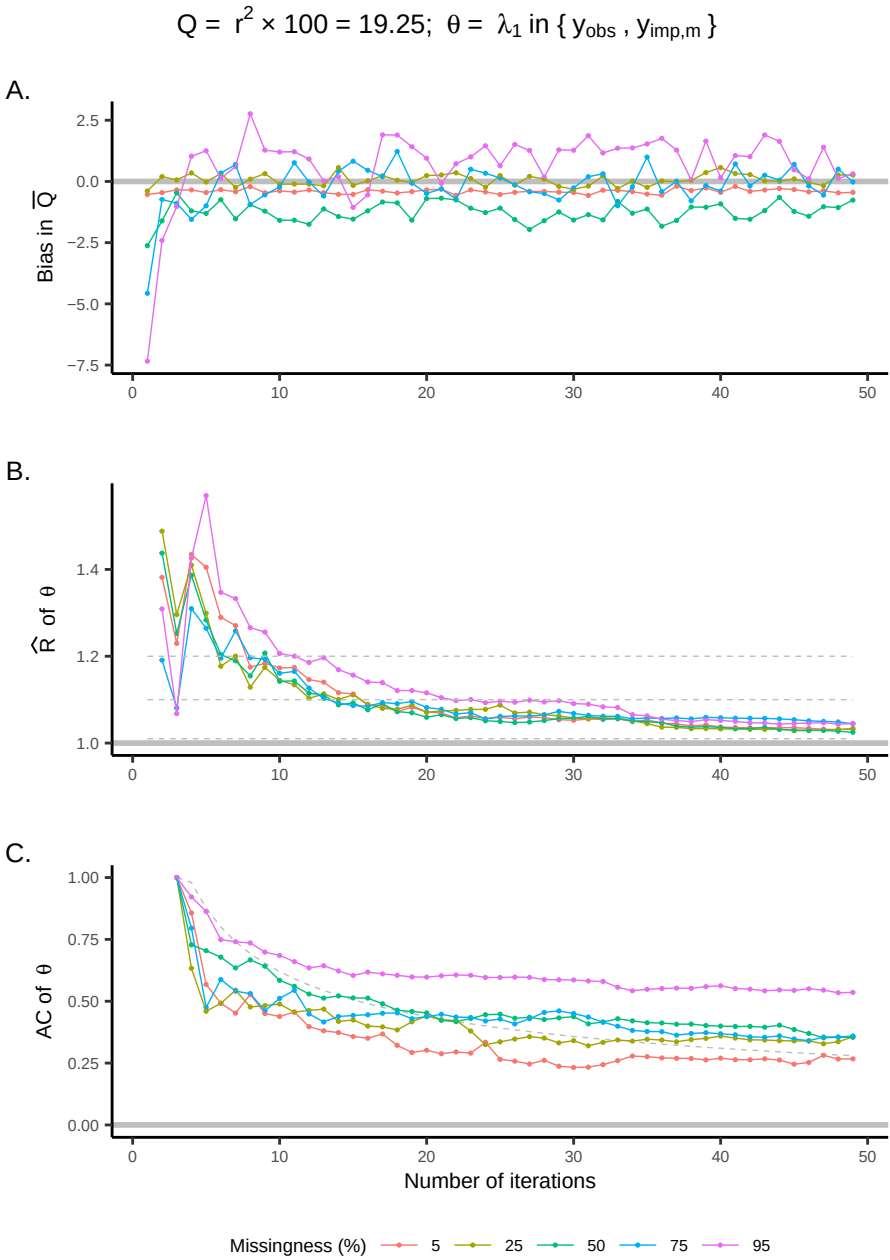

Figure 7. Convergence diagnostics regression coefficient.

Recommendations for empirical researchers

For empirical researchers:

1. Check trace plots for pathological non convergence and adjust imputation model if necessary.
2. use $\widehat{R}$ wait 1.1 ash threshold and autocorrelation with **?** as threshold. Keep iterating until these thresholds are reached.
3. Do not use the R function ACF. Instead, compute autocorrelations manually **(see e.g., [GitHub link])**.
4. Track your own scalar summary of interest. This is somewhat advanced but explained in Van Buren 2018. Compute $\widehat{R}$ and autocorrelation values for this scalar summary.
5. **Something** about the novel θ that is 'substantive model-independent'.

Recommendations for future research

Further research is needed to investigate their performance under clear violation of convergence, e.g. dependency between predictors (predictors with very high correlations). Also:

- Induce non-convergence with mis-specified imputation model.
- More difficult missingness problems, i.e., M(N)AR instead of MCAR. Proper performance under a 'missing completely at random' missingness mechanism is necessary to demonstrate the appropriateness of $\widehat{R}$ and AC as non-convergence diagnostics. However, results may not be extrapolated to other missingness mechanisms.
- Imputation technique with questionable convergence properties, e.g., PMM.
- Apply recommendations to empirical data and write a vignette for applied researchers.
- Implement recommendations in ShinyMICE.
- Develop an additional convergence diagnostic unique for iterative imputation: AC across imputations, instead of iterations.

Grading

- Scientific contribution (clear research question, convincing argumentation that the study adds to the scientific literature)
- Theory (use of theory, clear hypotheses, relevant literature)
- Method (correct data collection/handling, measures, choice and description of method of analysis)
- Results (correct interpretation of findings, clear presentation)
- Discussion and conclusions (summary, broader implications, limitations, future research)
- Written presentation (writing, structure, tables and figures, references)

References

Box GEP, Jenkins GM, Reinsel GC and Ljung GM (2015) *Time Series Analysis: Forecasting and Control*. John Wiley & Sons. ISBN 978-1-118-67492-5. Google-Books-ID: rNt5CgAAQBAJ.

Brooks SP and 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.

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

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

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

Genz A and Bretz F (2009) *Computation of multivariate normal and t probabilities*. Lecture notes in statistics. Heidelberg: Springer-Verlag. ISBN 978-3-642-01688-2.

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. New York, NY: Springer New York. DOI:10.1007/978-0-387-92407-6.

Lacerda M, Ardington C and Leibbrandt M (2007) Sequential regression multiple imputation for incomplete multivariate data using Markov chain Monte Carlo. Technical report, University of Cape Town, South Africa.

Lynch SM (2007) *Introduction to applied Bayesian statistics and estimation for social scientists*. Springer Science & Business Media.

MacKay DJ and 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.

R Core Team (2020) *R: A language and environment for statistical computing*. Vienna, Austria. URL https://www.R-project.org/. Tex.organization: R Foundation for Statistical Computing.

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

Rubin DB (1976) Inference and Missing Data. *Biometrika* 63(3): 581–592. DOI:10.2307/2335739. URL https://www.jstor.org/stable/2335739. Publisher: [Oxford University Press, Biometrika Trust].

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

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 and Groothuis-Oudshoorn K (2011) mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software* 45(1): 1–67. 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 and Bürkner PC (2019) Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC URL http://arxiv.org/abs/1903.08008. ArXiv: 1903.08008.

Vink G and van Buuren S (2014) Pooling multiple imputations when the sample happens to be the population. *arXiv:1409.8542 [math, stat]* URL http://arxiv.org/abs/1409.8542. ArXiv: 1409.8542 version: 1.