Sociological Methods & Research
2020, Vol. 49(?):1–24

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

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 imputation algorithms. We conclude that in the case of drawing inference from incomplete data, convergence of the iterative imputation algorithm is often a convenience but not a necessity.

Keywords

missing data, iterative imputation, non-convergence, mice

Overview of sections

Intro

- This is missingness
- This is how we solve missingness
- For that we need an algorithm, but what if it has not converged?
- That is what we solve in this paper

Algorithmic convergence

- What is algorithmic convergence?

[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

- When is it converged? Stationarity versus mixing.
- How is convergence currently investigated? And how should it be investigated? Multivariate versus univariate.

Methods

- Which methods are there? And how do they relate to mixing and stationarity?
- $\widehat{R}$
- AC
- We can use these on any θ

The Simulation study, Results, Discussion.

Introduction

Anyone who analyzes person-data may run into a missing data problem. Missing data is not only ubiquitous across science, but treating it can also be a tedious task. If a dataset contains just one incomplete observation, calculations are not defined, **hence** statistical models cannot do fitted to the data. To circumvent this, many statistical packages employ list-wise deletion by default (i.e., ignoring incomplete observations). Unfortunately, this *ad hoc* solution may yield invalid results (Van Buuren 2018). An alternative is to apply imputation. With imputation, we 'fill in' the missing values in an incomplete dataset. Subsequently, the model of scientific interest can be fitted to 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 draw valid inferences from incomplete data under many circumstances (Van Buuren 2018).

Figure 1 provides an overview of the steps involved with MI. The missing part y_{mis} of an incomplete dataset is imputed m times. This creates m sets of imputed data $\dot{y}_{\text{imp},\ell}$, where $\ell = 1, 2, ..., m$. The imputed data is then combined with the observed data y_{obs} to create m completed datasets. On each of these datasets the analysis of scientific interest is performed to estimate Q: the quantity of scientific interest (e.g., a regression coefficient). Since Q is estimated on each completed dataset, m separate $\hat{Q}_\ell$-values are obtained. Finally, the m $\hat{Q}_\ell$-values are combined into a single pooled estimate $\bar{Q}$. The premise of multiple imputation is that $\bar{Q}$ is an unbiased and confidence-valid estimate of the true—but unobserved—inference (Rubin 1996).

A popular method to obtain imputations is to use the 'Multiple Imputation by Chained Equations' algorithm, shorthand 'MICE'(Van Buuren and Groothuis-Oudshoorn 2011). With MICE, imputed values $\dot{y}_{\text{imp},\ell}$ are drawn from the posterior predictive distribution of the missing values y_{mis}. The algorithm is named after the iterative algorithmic procedure by which imputed values are generated: a multivariate distribution is obtained by iterating over a sequence of univariate imputations. The iterative nature of algorithms like MICE introduces a potential threat to the validity of the imputations: What if the algorithm has

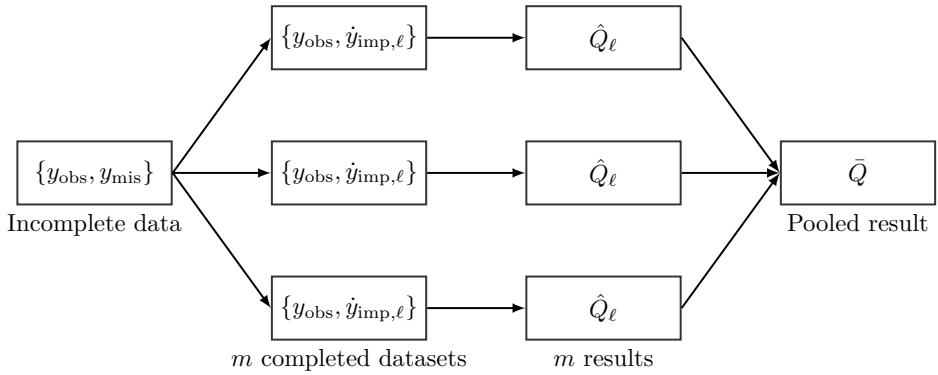

Figure 1. Scheme of the main steps in multiple imputation (where $m = 3$)—from an incomplete dataset, to m multiply imputed datasets, to m estimated quantities of scientific interest $\hat{Q}$s, to a single pooled estimate $\bar{Q}$.

not converged? Are the imputations then to be trusted? And can we rely on the inference obtained on the completed data?

These remain open questions, since the convergence properties of iterative imputation algorithms have not been systematically studied (Van Buuren 2018). There is no scientific consensus on how to evaluate convergence of imputation algorithms (Takahashi 2017). **Moreover, the convergence of chained equations algorithms such as MICE under certain default imputation models (e.g., 'predictive mean matching') is an entirely open question (Murray 2018).** Therefore, algorithmic convergence should be monitored carefully.

Currently, the recommended practice for evaluating the convergence of iterative imputation algorithms is through visual inspection. After running the imputation algorithm for a certain number of iterations, its convergence is monitiored by plotting a scalar summary of the state space of the algorithm θ against the iteration number. Typically, the θs under evaluation are chain means and variances—the average and the variance of the imputed values per imputation. Monitoring convergence by inspecting traceplots of these θs may be undesirable for several reasons 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) and, 3) there is not an objective measure that quantifies convergence. Therefore, a quantitative, diagnostic evaluation of convergence would be preferred—although not straightforward. Iterative imputation algorithms such as MICE are special cases of 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 (Gelman et al. 2013). Since MCMC algorithms do not reach a unique point at which convergence is established, diagnostics may only identify signs of *non*-convergence (Hoff 2009). Several of such non-convergence

diagnostics exist, but it is not known whether these are appropriate for iterative imputation algorithms.

In this paper, we study different methods for assessing non-convergence in iterative imputation algorithms. We define several diagnostics and evaluate how these diagnostics could be appropriate for iterative imputation applications. We then adress the impact of inducing non-convergence in iterative imputation algorithms by means of model-based simulation in `R` (R Core Team 2020). The aim of the simulation is to determine whether unbiased, confidence-valid inferences may be obtained under different levels of non-convergence (**and at which point convergence may safely be assumed**). Additionally, we will assess how well different non-convergence diagnostics perform in identifying signs of non-convergence (**and on which scalar summary of the state space of the algorithm these diagnostics should be applied**). We translate the results of the study into guidelines for applied researchers. **[add more societal relevance: what is the consequence of non-convergence? see example of non-convergence in Figure 2 (e.g., biased estimates, invalid inference, etc.)]** 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).

[Include sub-questions and make structure of the paper clear!] 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 Q be correct when the algorithm is not (yet) converged, and vice versa?

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 value, divided by the total number of units n, is called the missingness proportion 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 $m \times T$ plausible values, where m is the number of imputations ($\ell = 1, 2, \ldots, m$) and T is the number of iterations ($t = 1, 2, \ldots, T$) in the imputation algorithm. 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'.

Algorithmic non-convergence

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 the distribution of imputed values differs across imputations. Chains may be 'stuck' at a local optimum, instead of sampling imputed values from

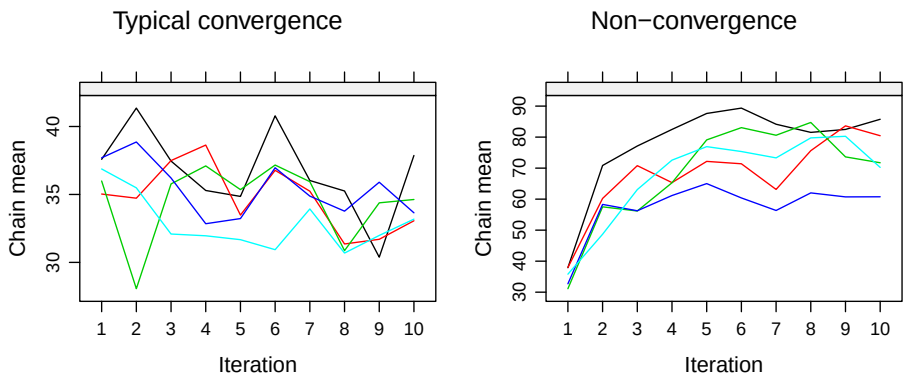

Figure 2. Typical convergence versus pathological non-convergence. Please note that this is the same data with a different imputation model, leading to different imputations (e.g., see the units on the y-axis).

the entire predictive posterior distribution of the missing values. This may cause underestimation of the variance within chains, which results in spurious, invalid inferences. Without stationarity, there is trending within imputation chains. Trending implies that further iterations would yield a systematically lower or higher set of imputations. Iterative imputation algorithms that have not yet reached stationarity, may thus yield biased estimates.

[SEGWAY to this section] To illustrate what non-convergence looks like in iterative imputation algorithms, we reproduce an example from van Buuren (2018, § 6.5.2). Figure 2 displays a subset of the example: a variable that we will call j. We see two traceplots of the chain means for j—i.e., the average of the imputed values for j in each imputation $\dot{y}_{\text{imp},\ell}$, plotted against the iteration number. Each line portrays one imputation. The plot on the left-hand side of the figure shows typical convergence of an iterative imputation algorithm. The right-hand side displays pathological non-convergence, induced by purposefully mis-specifying the imputation model. **In the typical convergence situation, the imputation chains intermingle nicely and there is little to no trending. In the non-convergence 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 imputation ℓ) are very different between the two plots. The algorithm with the mis-specified model yields imputed values that are on average a factor two larger than those of the typically converged algorithm. That means that non-convergence has a impact on the mean of j in $\dot{y}_{\text{imp},\ell}$. This difference (presumably) translates to bias in the m completed data estimates $\hat{\mu}_{j,\ell}$, and consequently also to the pooled estimate $\bar{\mu}_j$. Non-convergence thus leads to biased estimates. This shows the importance of reaching converged states in iterative imputation algorithms.

[MERGE with section below]. Inspecting non-convergence through traceplots of chain means and chain variances may be insufficient because they are univariate

summaries of the state space of the algorithm, while MICE is not only concerned with a single column, but the entire multivariate distribution of y_{imp}. Ideally, a multivariate θ should be monitored. A suggestion by Van Buuren (2018) for a multivariate θ to monitor is the quantity of scientific interest Q (e.g., a regression coefficient), which usually is multivariate in nature. Implementing this, however, might be somewhat too technical for empirical researchers. Moreover, this scalar summary is not model-independent, i.e., it only applies to one model of scientific interest. Yet, one of the advantages of MI is that the missing data problem and scientific problem are solved independently. On these grounds, such a θ can be considered insufficient too. Van Buuren (2018, § 4.5.2) also proposed multivariable evaluation of the MICE algorithm through eigenvalue decomposition, building on the work of MacKay and Mac Kay (2003). Eigenvalues of a variance-covariance matrix are a measure of the data's total covariance. Such a θ would be a model-independent, multivariate scalar summary to monitor, but it is not implemented in `mice`.

[MERGE with section above] Convergence has historically been inspected visually, by monitoring the imputation algorithm over iterations. This is typically done through traceplots. In a traceplot, the iteration number is plotted against a certain θ. θs are scalar summaries of the state space of the algorithm at a specific iteration. The default scalar summaries in `mice` are chain means and chain variances. Additionally, researchers may "monitor some statistic of scientific interest" (Van Buuren 2018, § 6.5.2). As Van Buuren (2018) describes, researchers can specify their quantity of scientific interest Q (e.g., a regression coefficient) as scalar summary θ. Such a user-defined scalar summary, however, may be somewhat advanced for empirical researchers. And moreover, it is not universal to all complete data problems. Focusing on the convergence of outcome parameters may influence the iterative imputation procedure in the sense that the model of evaluation favors the model of interest. The downside to these two 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. Ideally, one would like to evaluate a model-independent parameter that summarizes the multivariate nature of the data. Therefore, we propose a θ that summarizes the multivariate state space of the algorithm, but is independent from the model of scientific interest. We propose $\lambda_{1,\ell}$ as such a scalar summary. We define $\lambda_{1,\ell}$ as the first eigenvalue of the variance-covariance matrix of the completed data. Let $\lambda_{1,\ell} \geq \lambda_{2,\ell} \geq ... \geq \lambda_{j,\ell}$ be the eigenvalues of Σ_ℓ in each of the m completed datasets $\{y_{\text{obs}}, \dot{y}_{\text{imp},\ell}\}$. $\lambda_{1,\ell}$ is measure that summarizes the covariances in the completed datasets. **The first eigenvalue has the appealing property that is not dependent on the substantive model of interest. Eigenvalue decomposition is inspired by MacKay and Mac Kay (2003).**

Non-convergence diagnostics

Non-convergence diagnostics for MCMC algorithms typically identify problems in either one of the two requirements for convergence: mixing or stationarity. 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, we use $\widehat{R}$ **[WHICH? Explain that we'll use both]** and AC to evaluate mixing and stationarity separately, as recommended by e.g., Cowles and Carlin (1996).

[Add: we don't use effective sample size and MC error because they assume independent samples *within* chains. In MI, we only use the final estimate. Also, computing n_eff assumes infinitely many iterations, which is fine in Bayesian analyses where T is often at least one thousand. But the default in iterative imp is often substantially lower.]

Potential scale reduction factor

The potential scale reduction factor $\widehat{R}$ was coined by Gelman and Rubin (1992). An updated version of $\widehat{R}$ has been proposed by Vehtari et al. (2019, p. 5). **This version is better suited to detect non-convergence in the tails of distributions, by using three transformations on the scalar summary that it is applied to. Namely, rank-normalization, folding, and localization. The Vehtari et al. version of $\widehat{R}$ may be suitable for iterative imputation. We therefore use their definition of the diagnostic.** Let m be the total number of chains, T the number of iterations per chain ($T \geq 2$), and θ the scalar summary of interest (e.g., chain means). For each chain ($\ell = 1, 2, \ldots, m$), we estimate the variance of θ, and average these to obtain within-chain variance W.

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

We then estimate between-chain variance B as the variance of the collection of average θ per chain. **[Note that this is not the usual B in MI]**

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

From the between- and within-chain variances we compute a weighted average, $\widehat{\text{var}}^+$, which approximates the total variance of θ. $\widehat{R}$ is then obtained as a ratio between the 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 the sampled values in each chain are independent of the chain's initial value, the mixing component of convergence is satisfied. The variance between chains is then equivalent to the variance within chains, and $\widehat{R}$-values will be close to one. High $\widehat{R}$-values thus indicate non-convergence.

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. AC-values close to zero indicate independence between subsequent θs in imputation chains. 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. Positive AC-values occur when θ-values of subsequent iterations are similar. When high θ-values are followed by high θ-values, and low θ-values are followed by low θ-values, we speak of recurrence. Recurrence within imputation chains indicates trending. Ergo, non-stationarity may be diagnosed when AC values are larger than zero.

Thresholds

It is unlikely that an MCMC algorithm will achieve the ideal values of $\widehat{R} = 1$ and $AC = 0$. Because of its convergence to a distribution, the algorithm will show some signs of non-mixing and non-stationarity even in the most converged state. Upon approximate convergence, the imputation chains intermingle such that the only difference between the chains is caused by the randomness induced by the algorithm ($\widehat{R} > 1$), and there is some dependency between subsequent iterations of imputation chains ($AC > 0$). In practice, non-convergence is therefore diagnosed when non-convergence diagnostics exeed a certain threshold. The conventional threshold to diagnose non-mixing is $\widehat{R} > 1.2$ (Gelman and Rubin 1992). Another generally accepted threshold is $\widehat{R} > 1.1$ (Gelman et al. 2013). And more recently, Vehtari et al. (2019) proposed an even more stringent threshold of $\widehat{R} > 1.01$. The magnitude of AC-values may be evaluated statistically, using a Wald test with $AC = 0$ as null hypothesis (Box et al. 2015). AC-values that are significantly higher than zero indicate non-stationarity. **[think about one-sided test]**

In practice

Before we evaluate the performance of the non-convergence diagnostics $\widehat{R}$ and AC quantitatively through simulation, we apply them to the example of pathological non-convergence by Van Buuren (2018) that we reproduced above. We want to be able to draw the same conclusions as we did from visually inspecting the two algorithms. But $\widehat{R}$ and AC should at least be able to distinguish between the two algorithms plotted in Figure 2. From visual inspection, we know that the typically converged algorithm initially portrayed some signs of non-mixing (around $t = 2$), but intermingled nicely overall. Additionally, there was very little trending. The algorithm with pathological non-convergence showed severe non-mixing, although this gradually improved (from $t = 7$). Moreover, there was a lot of trending as well initially (up-to $t = 6$), after which the chains reached a somewhat more stationary state. To assess whether $\widehat{R}$ and AC may be appropriate non-convergence identifiers for iterative imputation algorithms, they should identify the same trends as the visual inspection. But at least, the following condition must hold: the methods should indicate worse performance for the mis-specified model with pathological non-convergence than the typically converged algorithm (i.e., higher $\widehat{R}$- and AC-values).

In Figure 3A, the chain means from Figure 2 are plotted again—now together. Panel B shows two versions of calculating ACs applied to the θs of panel A: the default calculation with `R` function `stats::acf()` (R Core Team 2020), and manual calculation as the correlation between θ in iteration t and θ in iteration $t + 1$. Panel C shows the traditional computation of $\widehat{R}$, and panel D shows $\widehat{R}$ as computed by implementing Vehtari et al. (2019) 's recommendations. **[Add labels for the two types of Rhat to the figure!]**

When we look at panel B, we conclude something weird. The AC-values calculated with the default function indicate equal performance for the typical convergence and the pathological non-convergence (up-to $t = 5$), while there is obvious trending in the θs of 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 AC function is not suited for iterative imputation algorithms. The function is optimized for performance when $t \geq 50$ (Box et al. 2015), while the default number of iterations in iterative imputation is often much lower. Therefore, we compute AC manually, see the solid line. These AC values indicate non-stationarity as expected. We therefore calculate AC manually.

Describe panel C here and why it is not optimal: NB. In panel C: De grotere variantie overall zorgt er kennelijk voor dat $\widehat{R}$ groeit. De $\widehat{R}$ geeft de foutieve boodschap dat convergentie steeds slechter gaat. Verklaring: de overdispersie assumptie klopt niet als er trend is.] The $\widehat{R}$-values in panel D do meet the requirements specified above. $\widehat{R}$ as computed conform Vehtari et al. (2019) does indeed indicate less signs of non-convergence as the number of iterations goes up. **(explain the dip in Rhat values at t=2. Namely, because we can only use 2 of the 3 tricks by Vehtari et al.**

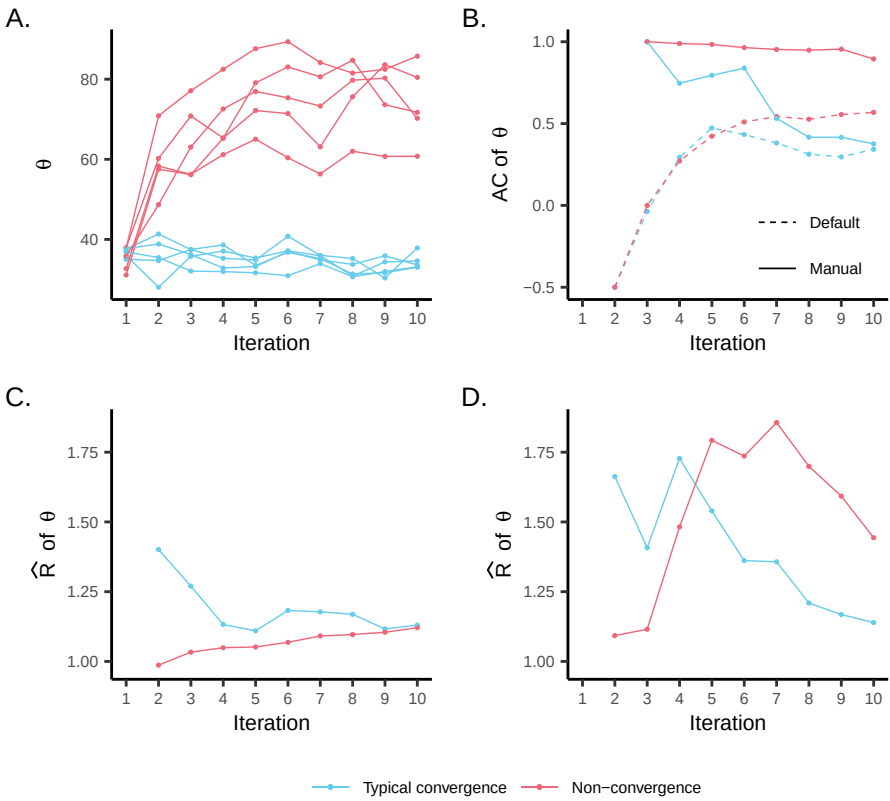

Figure 3. Convergence diagnostics applied on the imputation algorithms of Figure 2. θ = chain mean in $\dot{y}_{\text{imp},\ell}$.

2019, if the number of iterations is very low (t<4). That's why the $\widehat{R}$s are more similar to the traditional GR.)

In conclusion, we will use $\widehat{R}$ conform Vehtari et al. (2019) and AC computed manually as non-convergence diagnostics in the simulation study. The diagnostics will be applied to four θs of interest: chain means, chain variances, a quantity of scientific interest, and the first eigenvalue of the variance-covariance matrix.

Simulation study

The aim of the simulation study is to evaluate the **impact of inducing non-convergence ["early stopping" en "proportion of complete cases"]** 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 **[and performance of different thetas]**. And finally, we will formulate an informed advice on the requirements to safely conclude sufficient convergence in practice. That is, we conclude that convergence is sufficient when each estimate $\bar{Q}$ is an unbiased and confidence-valid estimate of the corresponding estimand Q. **[and 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 instantaneously. We expect that the incremental missingness proportions in our study will result in a corresponding increase in signs of non-convergence.

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`, $T = 5$.

[SEGWAY to this section] Q**s 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** Q**s on the** m **completed datasets, and finally pooling the** m **estimates** $\hat{Q}_\ell$ **to obtain a single** $\bar{Q}$ **for each** Q**. We try to predict the effect of non-convergence (i.e., the difference between** Q **and** $\bar{Q}$**) with the convergence diagnostics** $\widehat{R}$ **and** AC**, by applying these methods to several scalar summaries of the state space of the algorithm,** θ**s.**

Hypotheses

1. We expect that simulation conditions with a high missingness proportion p_{mis} and a low number of iterations T will more often 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 from $t = 5$ onward (Lacerda et al. 2007). We therefore suspect that $\widehat{R}$ may over-estimate signs of non-convergence in iterative imputation algorithms, compared to the validity of 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. We further hypothesize that multivariate θs are better at detecting non-convergence than univariate θs.

Set-up

We investigate non-convergence of the MICE algorithm through model-based simulation in `R` (version 3.6.3; R Core Team 2020). The number of simulation repetitions is 1000. 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. Let the predictor space be defined as

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

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

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

where $\epsilon \sim \mathcal{N}(0, 100)$. This results in a complete set y, with observed values for all units i on all variables j ($j = Y, X_1, X_2, X_3$). From the complete set we obtain the true values of the scientific estimands Q.

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 **(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.

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: $p_{\text{mis}} = .05, .25, .5, .75, .95$. We ampute the complete data with the `mice` package (function `mice::ampute()`; Van Buuren and Groothuis-Oudshoorn 2011). We consider all possible univariate and multivariate patterns of missingness, 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 each incomplete dataset y 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: $\dot{y}_{\text{imp},\ell}$, 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 $\dot{y}_{\text{imp},\ell}$ we obtain the chain means and chain variances for each variable j. These will serve as scalar summary θ to apply non-convergence diagnostics $\widehat{R}$ and AC on. We then use `mice::complete()` to combine the imputed data $\dot{y}_{\text{imp},\ell}$ with the observed data y_{obs}, resulting in five completed datasets $\{y_{\text{obs}}, y_{imp,m}\}$.

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

- Estimate $\hat{Q}_\ell$s for $Q = \mu$ and $Q = \sigma$ with `base::mean()`, and `stats::sd()` (R Core Team 2020). We pool the $\hat{Q}_\ell$s with `base::mean()` to get $\bar{Q}$s.
- Compute $\lambda_{1,\ell}$, to use as θ in the two convergence diagnostics. By definition, the first eigenvalue of the variance-covariance matrix, $\lambda_{1,\ell}$, is equal to the variance of the principal component solution of each $\{y_{\text{obs}}, y_{imp,m}\}$, which we calculate with `stats::princomp()` (R Core Team 2020).
- Perform multiple linear regression with `stats::lm()` (R Core Team 2020) to get the $\hat{Q}_\ell$s for $Q = \beta$ and $Q = r^2$. We pool the $\hat{Q}_\ell$s conform Vink and van Buuren (2014) to get $\bar{Q}$s for the βs and r^2 **[check finite pooling of r^2, instead of `mice::pool.r.squared()`]**. Additionally, we use the $\hat{Q}_\ell$s of $Q = \beta$ as θ to apply the convergence diagnostics to.

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 performance of the iterative imputation algorithm in each simulation condition 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 confidence-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 it is a measure of efficiency. Under nominal coverage, short CIs are preferred. Too short CIs, however, indicate underestimation of the variance in $\bar{Q}$, **which may yield inferences with low statistical power if CIW is large**.

With these performance measures, we also evaluate how well the non-convergence diagnostics $\widehat{R}$ and AC can identify simulation conditions in which the inferences are affected by non-convergence, i.e., where $\bar{Q}$ is *not* an unbiased, confidence-valid estimate of Q. **[REPHRASE this:] For performance of univariate Qs, we apply the non-convergence diagnostics to univariate θs. When Q is multivariate, we look at signs of non-convergence identified in multivariate θs.**

- For $Q = \mu$, θs are chain means **[ADD that these should be unbiased because MCAR!]**.
- For $Q = \sigma$, θs are chain variances.
- For $Q = \beta$, θs are estimated βs per imputation.
- For $Q = r^2$, θs are the first eigenvalue of the variance-covariance matrix, $\lambda_{1,\ell}$.

Results

[Add more info about figure legends and axes]

Our results show the impact of **inducing non-convergence ["early stopping" en "proportion of complete cases"]** in an iterative imputation algorithm on several scientific estimates, and whether the non-convergence diagnostics $\widehat{R}$ and AC identify the signs of non-convergence in the algorithm. For reasons of brevity, we only discuss the non-convergence diagnostics for estimates with the worst performance in terms of bias. Bias in the descriptive statistics ($Q = \mu$ and $Q = \sigma$) is most pronounced for the outcome variable Y. Of the regression coefficients ($Q = \beta$) we observe the largest bias for β_1 (the effect of X_1 on Y). There is only one estimate for $Q = r^2$ to consider. In the figures, we present simulation conditions where the number of iterations is $1 \leq T \leq 50$, since the results are more or less stable for conditions where $T \geq 30$. Full results are available from github.com/gerkovink/shinyMice.

Quantities of scientific interest

Figure 4 displays the influence of **inducing non-convergence ["early stopping" en "proportion of complete cases"]** on the estimated quantities of scientific interest, Qs. **[Add panel labels, describe panels]** Simulation conditions that are impacted by non-convergence portray more extreme bias in the estimated Qs, and non-nominal coverages.

$Q = \mu$. The estimated univariate means seem unaffected by the number of iterations in the algorithm. This implies that valid inferences may be obtained with as little as one iteration ($T \geq 1$). The magnitude of the bias in $\bar{Q}$ depends solely on the missingness proportion p_{mis}, with missingness proportions of 75% and 95% leading to more extreme biases than other missingness proportions **[note that these are curious]**. Unbiased estimates are obtained in all conditions where $p_{\text{mis}} \leq .50$. Note however, that even in the conditions where $p_{\text{mis}} \geq .75$, the bias is small: μ_Y is maximally under-estimated by 0.02 units, while the true value of μ_Y is 25.81 ($\sigma_Y = 11.32$). We conclude that non-convergence does not substantially impact the validity of the inferences for $Q = \mu$. **[To be expected, because MCAR]**

$Q = \sigma$. The estimated standard deviations are largely affected by the missingness proportions, but not to much by the number of iterations. Valid inferences are obtained in conditions with just two iterations. Interestingly, the bias in first iteration is actually *less* extreme than for any other number of iterations. **[Due to MCAR and randomly drawn initial values?]** We may therefore conclude that no iteration at all is needed to obtain approximately unbiased estimates for $Q = \sigma$. Completely unbiased estimates are only obtained in conditions where $p_{\text{mis}} = .05$. Conditions where $p_{\text{mis}} \geq .25$ are impacted by non-convergence in order of increasing missingness proportion. Similar to the bias for $Q = \mu$ the magnitude of the bias in $Q = \sigma$ is negligible. We therefore conclude that neither of the univariate Qs is affected substantially by non-convergence.

$Q = r^2$. Bias in the estimated variance explained depends both on missingness proportions and the number of iterations. The magnitude of the bias clearly follows

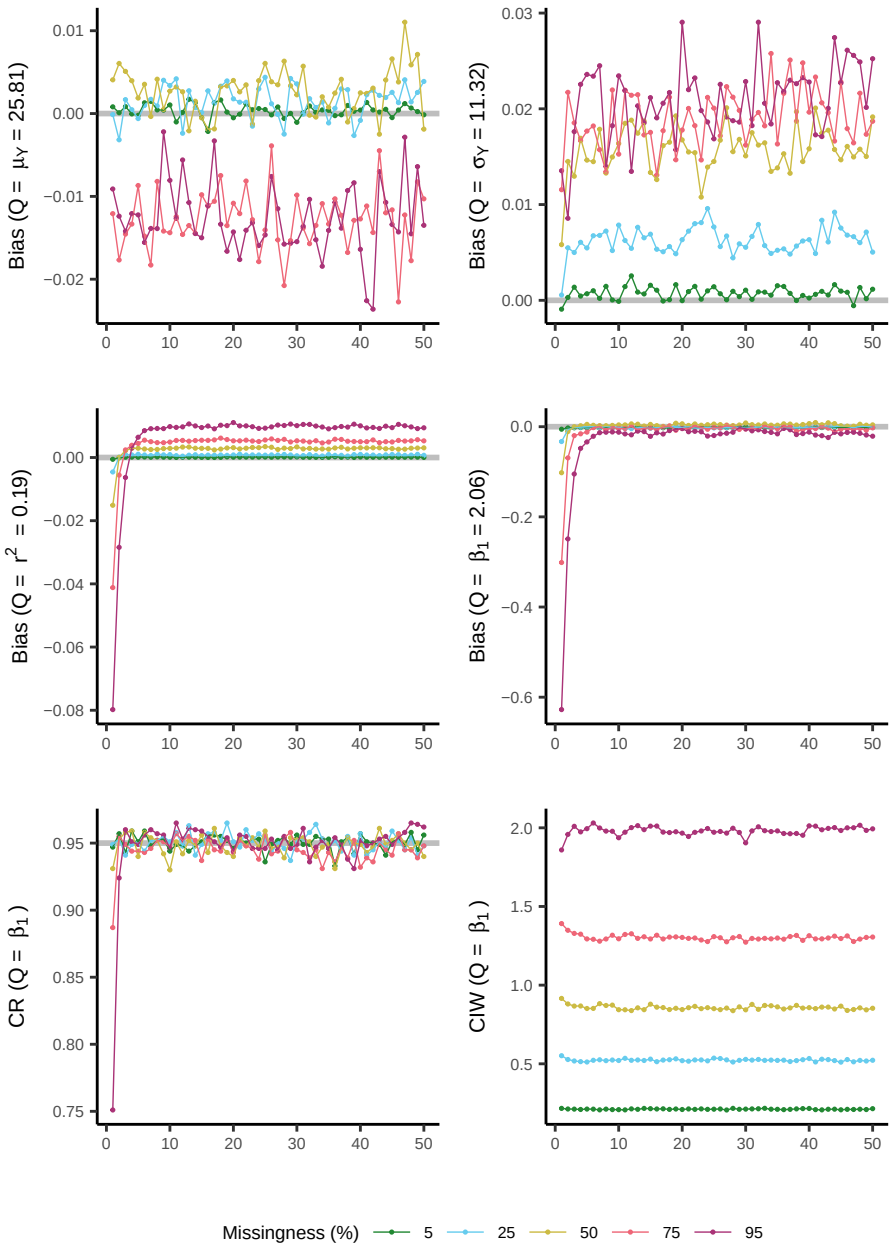

Figure 4. Impact of non-convergence on statistical inferences. Depicted are the bias, coverage rate (CR) and confidence interval width (CIW) of the estimates $\bar{Q}$. We present the subset of Qs with the worst performance in terms of bias: the mean and standard deviation of Y, and the regression coefficient of X_1.

the incremental order of the missingness proportions. Completely unbiased estimates may only be obtained if the missingness proportion is at most 25%. Approximately unbiased estimates are observed for any missingness proportion. For $p_{\text{mis}} \leq .25$, the minimum number of iterations is two ($T \geq 2$), for $p_{\text{mis}} = .50$ it is $T \geq 4$, and $p_{\text{mis}} = .75$ requires $T \geq 5$. The highest number of iterations needed to obtain valid inferences is seven ($p_{\text{mis}} = .95; T \geq 7$). **[interpret sign of bias? ADD what this implies: dat - zelfs bij een gebiasede methode - je maar 7 iteraties nodig hebt om het eindresultaat te bereiken.]**

$Q = \beta$. For the estimated regression coefficients, we consider bias, confidence interval width (CIW), and coverage rate (CR) as performance measures. **[Split panels]** Unbiased estimates are obtained in conditions where $p_{\text{mis}} \leq .50$ and $T \geq 3$. Approximately confidence-valid estimates are obtained in any condition where $T \geq 3$—despite of persistant bias in conditions where $p_{\text{mis}} \geq .75$. These approximately nominal coverages are explained by the CIWs: as expected, conditions with higher missingness proportions result in more uncertainty in the estimates (i.e., wider CIs), which diminish the effect of the bias. confidence-valid inferences may be obtained when $T \geq 3$, while we need at least seven iterations to reach valid inferences in terms of bias. Continued iteration after $T \geq 7$ seems futile. **[add max T for other betas?]**

Summary Overall, we see that conditions with a higher proportion of missingness are affected more severely by non-convergence. Multivariate Qs are also substantially influenced by the number of iterations. This means that the algorithm has indeed not converged at $t = 1$, and converges gradually as t moves to T. The point at which an additional iteration does not lead to an improvement in the estimates depends on quantity of scientific interest Q and the missingness proportion p_{mis}. The highest number of iterations that is necessary to reach this point across all simulation conditions is $T = 7$. This implies that, under the current specifications, no more than seven iterations are necessary to obtain sufficient performance in terms of bias and confidence-validity.

Non-convergence diagnostics

We evaluate the non-convergence diagnostics $\widehat{R}$ and AC against the performance measures described above (i.e., bias and confidence-validity of the quantities of scientific interest Q). The diagnostics should identify those simulation conditions that are negatively impacted by non-convergence. We expect the most signs of non-convergence in conditions where valid estimates are not obtained (high missingness proportions, $T < 7$), and at least some signs of non-convergence in conditions where estimates are not approximately unbiased ($p_{\text{mis}} = .95$, any number of iterations).

Figure 5 displays the non-convergence diagnostics $\widehat{R}$ and AC for each of the four θs under consideration. **[Add panel labels and explain how to read the plots]**

$\theta =$ *chain mean.* **[As expected, $\widehat{R}$ lowers with the number of iterations. At a cut-point of 1.2 , we have convergence at about 8 iteration, whereas we need 12 and 25 iteration for more conservative cut-off 1.1 and 1.05. We do not reach the very strict 1.01.]**

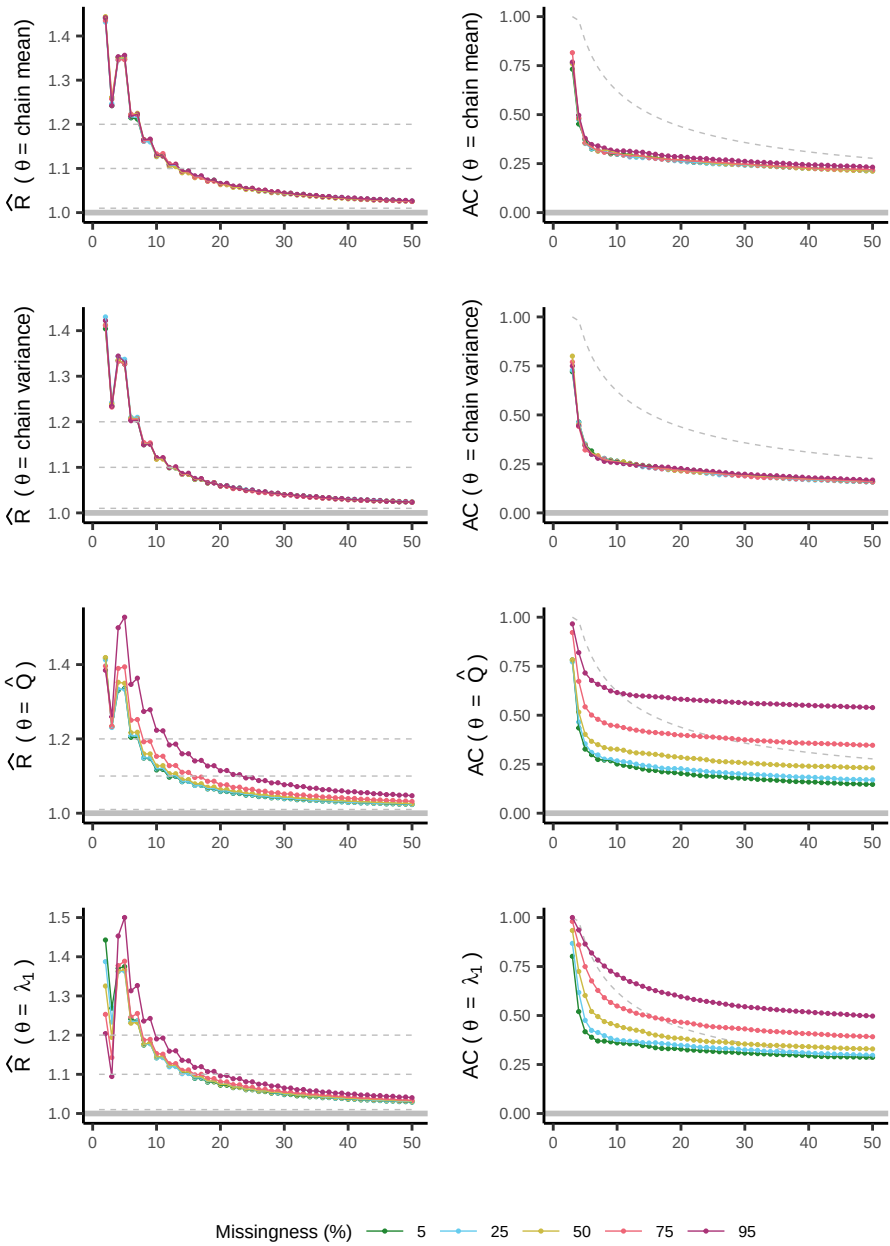

Figure 5. Non-convergence diagnostics. [Split up into 2 or 4 figures??]

We evaluate the performance of the non-convergence diagnostics applied to chain means against the bias in $Q = \mu$. Based on the bias in Q, we concluded that one iteration is sufficient to obtain valid inferences, irrespective of the missingness proportion. The non-convergence diagnostics, however, identify some signs of non-convergence in conditions with as many as thirty iterations. This does not correspond to the impact of non-convergence in the univariate Qs. **Compared to the multivariate Qs [add before that we evaluate the thetas against all Qs?], $\widehat{R}$ and AC applied to the chain means fail to identify the non-convergence in conditions where $p_{\text{mis}} = .95$.**

$\theta =$ *chain variance.* The non-convergence diagnostics applied to the chain variances as θ yield highly similar results to the chain means. **[Segway to performance of $Q = \sigma$] While the bias in the estimated standard deviations mainly depends on the missingness proportion of each condition, $\widehat{R}$ and AC are influenced mostly by the number of iterations. We thus conclude that there is no added value of evaluating non-convergence with chain variances as θ in the diagnostics.**

$\theta = \hat{Q}_\ell$. **[We zien weer hetzelfde voor laag-medium, beetje meer voor extreme misisnh data.]**

The non-convergence diagnostics are applied to the estimated regression coefficients in each imputation, and evaluated against the bias in the corresponding pooled estimate.

$\widehat{R}$ indicates signs of non-convergence across all missingness proportions and each number of iterations, in the expected order: simulation conditions with higher missingness proportions and less iterations are identified as the worst converged. The thresholds to diagnose non-convergence are overcome in different conditions, depending on the number of iterations and the missingness proportion. The most lenient threshold ($\widehat{R} > 1.2$) is overcome at $T \geq 9$ for $p_{\text{mis}} = 0.05$, and $T \geq 12$ for $p_{\text{mis}} = 0.95$. The conventional $\widehat{R} > 1.1$ is surpassed at $T \geq 12$ and $T \geq 24$, respectively.

For the autocorrelations applied to the $\hat{Q}_\ell$s, we see an even stronger effect of the missingness proportions. Especially the conditions where $p_{\text{mis}} = 0.75$ and $p_{\text{mis}} = 0.95$ are identified as non-stationary. The threshold used to evaluate whether $AC = 0$, however, is only reached in conditions with at least ten iterations. This suggests that the default number of iterations may be too low to detect non-stationarity with AC. However, the critical value for diagnosing non-convergence with AC decreases as a function of the number of iterations. The conclusion to identify non-stationarity in conditions where $T \geq 10$ and $p_{\text{mis}} = 0.95$ might therefore be spurious.

Compared to the bias in $\bar{Q}$, the diagnostics applied to the $\hat{Q}_\ell$s over-estimate the severity of the non-convergence.

Summary In general, the diagnostics identify more signs of non-convergence in simulation conditions with a lower number of iterations. The missingness proportions, however, only seem to impact multivariate convergence. When $\widehat{R}$ and AC are applied to univariate θs, they identify the same amount of non-convergence for any missingness proportion. When the diagnostics are applied to multivariate θs, they generally identify more signs of non-convergence for higher missingness proportions.

As expected, none of the simulation conditions that we consider resulted in complete convergence of the algorithm (defined as $\widehat{R} = 1$ and $AC = 0$). However, the thresholds used in practice to diagnose non-convergence seem too conservative for imputation purposes. $\widehat{R}$ and AC identify signs of non-convergence in conditions where the $\bar{Q}$s are unbiased and confidence-valid estimates of the Qs. The most recent recommended threshold for diagnosing non-convergence with $\widehat{R}$ is never overcome in this study (all $\widehat{R}$>1.01). Apparently, this threshold is too stringent for iterative imputation algorithms, and will not be discussed any further.

$\theta = \lambda_{1,\ell}$. Finally, we evaluate the non-convergence diagnostics applied to the first eigenvalue of the variance-covariance matrices of the completed data. Similar to the results the the regression coefficients as θs, we see effects of both the missingness proportion and number of iterations in each simulation condition. The most signs of non-convergence are identified in conditions where p_{mis} is large and T is small.

Notwithstanding the similarities with $\theta = \hat{Q}_\ell$, there is a big difference in $\widehat{R}$-values for conditions where $T < 4$. In these conditions, $\widehat{R}$ behaves oppositely of our expectation: conditions with the lowest missingness proportions yield the highest $\widehat{R}$-values. In conditions with a higher number of iterations, this artifact disappears. This suggests that the combination of $\widehat{R}$ and $\lambda_{1,\ell}$ is only appropriate for imputation algorithms where the number of iterations at least four. Interpreting the $\widehat{R}$ values of conditions where $T < 3$ may lead to the incorrect conclusion that only conditions with *low* missingness proportions show signs of non-convergence. For applications as a model-independent multivariate θ, we should therefore only interpret the $\widehat{R}$-values when $T \geq 4$. Using this cut-off we conveniently ignore the initial 'dip' at $T = 3$ as well. The traditional threshold $\widehat{R} > 1.2$ is overcome in conditions where $p_{\text{mis}} = .95$ and $T \geq 10$, and for all other missingness proportions in conditions where $T \geq 8$. With the $\widehat{R} > 1.1$ threshold, the number of iterations are $T \geq 20$ and $T \geq 16$, respectively.

The AC-values largely depend on the missingness proportion in each condition. Conditions where $p_{\text{mis}} = .95$ show statistically significant autocorrelations for any number of iterations. The other extreme, conditions where $p_{\text{mis}} = .05$, does not yield AC values above the threshold. The AC values with this missingness proportion are equivalent for all conditions where $T \geq 6$, which suggests that the algorithm is as stationary as it can be. A similar point is only observed after twenty iterations in conditions with higher missingness proportions.

In comparison to the Qs, the diagnostics again over-estimate the signs of non-convergence. Conditions with low missingness proportions (e.g., 5% or 25% of the complete data) yield unbiased, confidence-valid estimates with as little as two iterations. The convergence diagnostics, however, indicate improved convergence up-to eight iterations according to AC and thirty to forty iterations according to $\widehat{R}$. The model-independent θ $\lambda_{1,\ell}$ seems as well-equipped to diagnose signs of non-convergence as $\theta = \hat{Q}_\ell$. The univariate θs do not differentiate between bias induced by the missingness proportions.

Discussion

We arrive back at the original questions that we posed.

- Non-convergence can be identified with non-convergence diagnostics $\widehat{R}$ and AC. But common thresholds are too conservative. Why? Part of the distribution is already determined by the observed data. in regular MCMC the entire distribution is unknown (**priors??**). In iterative imputation algorithms we have the same characteristics as MCMC in the limit, but because we combine the known distribution with the unknown distribution, convergence is reached much sooner.
- Inferences are unbiased much quicker than the algorithm has converged. All things univariate are converged almost intantly after initiating the iterative algorithm with a valid imputation model. But less quick with multivariate parameters.

In short, iterative imputation algorithms may yield approximately unbiased, confidence-valid estimates under different levels of induced non-convergence. Univariate estimates seem robust against terminating the algorithm early: There is no clear effect of the number of iterations on the bias in estimated means and standard deviations. The bias in these Qs was only impacted by the missingness proporion p_{mis}, with higher p_{mis} resulting in more bias. Bias in multivariate estimates depended on both p_{mis} and the number of iterations. However, the only effect of the number of iterations was observed when the algorithm was terminated at seven iterations or earlier. This suggests that continued iteration beyond seven iterations does not yield better estimates. Yet, the non-convergence diagnostics $\widehat{R}$ and AC indicated that the iterative imputation algorithm did not reach a stable state before twenty to thirty iterations. **[MERGE or remove]** $\widehat{R}$ **and autocorrelation indicate that there are signs of non-convergence in the algorithm up-to at least twenty iterations, while unbiased, confidence-valid estimates may be obtained with as little as one iteration. This observation is in agreement with** one of [still add the others!]** the simulation hypotheses: $\widehat{R}$ over-estimates the severity of non-convergence when applied to MI procedures.**

Over-estimation of the signs of non-convergence may be due to the methods (and their thresholds) or due to the Qs (descriptive 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. Which might be why $\widehat{R}$ and AC identify non-convergence.

Some other interesting findings are **[REPHRASE]**

- Is there more bias in conditions where non-convergence was induced? (YES for multivar., NOPE for univariates).
- 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? (YES, while it doesn't really matter as long as it's multivariate, $\lambda_{1,\ell}$ is model-independent and therefore preferred).
- What should the thresholds be? (much HIGHER than recommended for $\widehat{R}$, difficult to say for AC).

- And the default nr of iterations? (MAYBE increase to 10 to detect significant AC and be sure not to miss the dip in $\widehat{R}$?).

Recommendations for empirical researchers

After imputation it is always wise to take the following steps:

1. First, check traceplots for signs of pathological non-convergence. Adjust imputation model if necessary.
2. Monitor non-convergence diagnostics $\widehat{R}$ and AC. Do not use the traditional calculation for $\widehat{R}$, or the `R` function `stats::acf()` to compute AC. Instead, calculate $\widehat{R}$ conform Vehtari et al. (2019) and compute autocorrelations manually (see e.g., github.com/gerkovink/shinyMice).
3. Track θs over iterations, e.g., the novel θ that is 'substantive model-independent'.
4. Track your own scalar summary of interest (see e.g., Van Buuren (2018)). Compute $\widehat{R}$ and autocorrelation values for this scalar summary.
5. Use the threshold $\widehat{R} > 1.2$ for mixing, and assess stationarity by plotting the autocorrelation over iterations. Keep iterating until the threshold for mixing is reached, and until a reasonably decreasing asympototic AC value is obtained. At that point, inferences are unlikely to improve by continued iteration.

Recommendations for future research

Much remains unknown. about non-convergence in iterative imputation algorithms. Further research is needed to investigate the performance of iterative imputation algorithms under clear violation of convergence, e.g. dependency between predictors (predictors with very high correlations). For example:

- Induce non-convergence with mis-specified imputation model.
- Introduce more difficult missingness problems, i.e., M(N)AR instead of MCAR. That is, 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 imputation software, e.g., evaluation environment for applied researchers.
- Develop an additional convergence diagnostic unique for iterative imputation: AC across imputations, instead of iterations.

In short, we have shown that an iterative imputation algorithm can yield correct outcomes, even when a converged state has not yet formally been reached. Any further iterations would then burn computational resources without improving the statistical inferences. To conclude, in the case of drawing inference from incomplete data,

convergence of the iterative imputation algorithm is often a convenience but not a necessity. **[Our study found that - in the cases considered - inferential validity was achieved after 5-10 iterations, much earlier than indicated by the $\widehat{R}$ and AC diagnostics. Of course, it never hurts to iterate longer, but such calculations hardly bring added value.]**

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.

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.

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

Rubin DB (1996) Multiple Imputation After 18+ Years. *Journal of the American Statistical Association* 91(434): 473–489. DOI:10.2307/2291635.

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.

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.

Vink G and van Buuren S (2014) Pooling multiple imputations when the sample happens to be the population URL http://arxiv.org/abs/1409.8542.