Non-convergence in iterative imputation

H. I. Oberman

Iterative imputation has become the de facto standard to accommodate for the ubiquitous problem of missing data. While it is widely accepted that this technique can yield valid inferences, these inferences all rely on algorithmic convergence. Our study provides insight into identifying non-convergence in iterative imputation algorithms. We show that these algorithms can yield correct outcomes even when a converged state has not yet formally been reached. In the cases considered, inferential validity is achieved after five to ten iterations, much earlier than indicated by diagnostic methods. We conclude that it never hurts to iterate longer, but such calculations hardly bring added value.

Aims

- can we diagnose non-convergence quantitatively?
- ~~which parameter should we track?~~
- how many iterations are needed before convergence is reached?
- how many iterations are needed before valid inferences are reached?

Introduction (newest)

Iterative imputation has become the de facto standard to accommodate missing data in social scientific research [ref: e.g., rubin after 18 years paper? murray 2018?]. The technique enables researchers to draw valid inferences when faced with incomplete observations, without resorting to ad hoc solutions such as list-wise deletion, which may bias results. With iterative imputation, every missing cell in an incomplete dataset is imputed (i.e., filled in) algorithmically. And once the dataset is completed, the analysis of scientific interest can be performed. This analysis will yield unbiased and confidence-valid estimates if–and only if–both the missing data problem and the scientific problem are appropriately considered. The validity of this whole process naturally depends on the convergence of the algorithm that was used to to generate the imputed values. Yet, there has not been a systematic study on how to evaluate the convergence of iterative imputation algorithms.

Determining whether an algorithm has converged is not trivial, especially in the context of iterative imputation. Since the aim of iterative imputation is to converge to a distribution and not to a single point, the algorithm may produce some fluctuations even after it has converged. Because of this property, it may be more desirable to focus on *non*-convergence. A widely accepted practice is visual inspection of the algorithm (Raghunathan and Bondarenko 2007; Van Buuren 2018). Diagnosing non-convergence through visual inspection may, however, be undesirable: 1) it may be challenging to the untrained eye, 2) only severely pathological cases of non-convergence may be diagnosed, and 3) there is not an objective measure that quantifies convergence (Van Buuren 2018). [add argument: "Inspection of such plots is a notoriously unreliable method of assessing convergence and in addition is unwieldy when monitoring a large number of quantities of interest, such as can arise in complicated hierarchical models" (Gelman et al., 2013, p. 285).] Therefore, a quantitative diagnostic method to assess convergence would be preferred. Fortunately, there are non-convergence identifiers for *other* iterative algorithms, but the validity of these identifiers has not been systematically evaluated on imputation algorithms.

In this study, we explore different methods for identifying non-convergence in iterative imputation algorithms. We evaluate whether these methods are able to cover the extent of the non-convergence, and we also investigate the relation between non-convergence and the validity of the inferences. We translate the results of our simulation study into guidelines for practice, which we demonstrate by means of a case study.

Simulation set-up

We investigate non-convergence in iterative imputation through model-based simulation in R (version 4.4.2; R Core Team (2020)). We provide a summary of the simulation set-up in Algorithm 1; the complete script and technical details are available from github.com/hanneoberman/MissingThePoint [TODO: make this a Zenodo DOI]. The number of simulation repetitions $n_{\text{sim}} = 2000$.

Algorithm 1: simulation set-up (pseudo-code)

```
for each simulation repetition
  1. simulate complete data
  for each missingness condition
    2. create missingness
    for each iteration
      3. impute missingness
      4. perform analysis of scientific interest
      5. apply non-convergence identifiers
      6. pool results across imputations
      7. compute performance measures
```

```
    8. combine outcomes from all iterations
  9. combine outcomes from all missingness conditions
10. aggregate outcomes from all simulation repetitions


# pseudo-code of simulation
for (number of simulation runs) {
  1. simulate complete data
  for (missingness conditions) {
    2. create missingness
    for (number of iterations) {
      3. impute missingness
      4. estimate quantities of scientific interest
      5. apply performance measures to the estimates
      6. compute non-convergence diagnostics
    }
  }
7. aggregate outcomes across simulation runs
```

Aims

With this simulation, we assess the impact of non-convergence on the validity of scientific estimates obtained using the imputation package `{mice}` (Van Buuren and Groothuis-Oudshoorn 2011). Inferential validity is reached when estimates are both unbiased and have nominal coverage across simulation repetitions ($n_{\text{sim}} = 2000$). To evaluate convergence, we terminate the imputation algorithm after a varying number of iterations ($n_{\text{it}} = 1, 2, ..., 100$). We differentiate between six different missingness scenarios that are defined in the data generating mechanism.

Data generating mechanism

In each simulation repetition, we first generate a complete set of $n_{\text{obs}} = 200$ cases, representing person-data in a multiple linear regression problem. The predictor space consists of three multivariately normal random variables,

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & & \\ 0.5 & 1 & \\ 0.5 & 0.5 & 1 \end{pmatrix} \right].$$

The outcome variable Y is a linear combination of the three predictors, such that for each unit $i = 1, 2, ..., n_{\text{obs}}$,

Missingness conditions		
missing data mechanism		proportion of incomplete cases
	×	
MCAR, MAR		25%, 50%, 75%

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

where $\epsilon \sim \mathcal{N}(0, 1)$. This results in a complete dataset of size $n_{\text{obs}} \times p$, with units $i = 1, 2, ..., n_{\text{obs}}$ and variables $j = Y, X_1, X_2, X_3$). Multivariate normal data are generated using the `mvtnorm` package (Genz and Bretz 2009).

Subsequently, the complete are 'amputed' (i.e., made incomplete) according to six missingness conditions. We use a 2×3 factorial design with two missing data mechanisms and three proportions of incomplete cases, see Table XYZ.

[TODO: make this a box] Missingness mechanisms refer to the probability of being missing for any given entry in a dataset. There are three distinct types of mechanisms, as defined by Rubin (1976). Roughly translated, the probability of being missing may be equal for all entries (MCAR; Missing Completely At Random), may depend on observed information (MAR; Missing At Random), or may depend on *un*observed information, making the missingness non-ignorable (MNAR; Missing Not At Random).

The missing data mechanisms under consideration are 'missing completely at random' [MCAR; Rubin (1987)], where the probability to be missing is the same for all $n_{\text{obs}} \times p$ cells in y, and a right-tailed 'missing at random' (MAR) mechanism, where the probability to be missing is a function of the observed data, and higher values are more likely to be missing. The proportion of incomplete cases π_{inc} is set to 25%, 50%, and 75%. We use the 'mice' package [function `mice::ampute()`; Van Buuren and Groothuis-Oudshoorn (2011)] to obtain an incomplete dataset $\{y_{\text{obs}}, y_{\text{mis}}\}$ for every missingness condition. The missing data patterns are all possible combinations of multivariate missingness.

[TODO: add left-tailed M(N)AR mechanism, md pattern, and maybe refer to ampute paper or "Dance of…" by Schouten 2018?].

Estimands

We impute the missing data five times ($m = 5$) using Bayesian linear regression imputation with `mice` (Van Buuren and Groothuis-Oudshoorn 2011). On each imputed dataset, we perform multiple linear regression as our analysis of scientific interest. Our estimands are the regression coefficients β,

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3,$$

Non-convergence diagnostics				
identifier		parameter		variable
	×		×	
AC, $\widehat{R}$		chain means, chain variances		Y, X_1, X_2, X_3

where $\hat{Y}$ is the predicted value of the outcome. We estimate these coefficients using the `lm()` function (R Core Team 2020) in each imputed dataset, and subsequently pool the estimates across the five imputations using Rubin's rules [function `mice::pool()`; Van Buuren and Groothuis-Oudshoorn (2011)].

Performance measures

We evaluate the pooled estimates against their true population values using the performance measures bias, confidence interval width, and coverage rate, as recommended by van Buuren ((2018) § 2.5.2). We calculate bias as $\bar{Q} - Q$. CR is defined as the percentage of simulation repetitions in which the 95% confidence interval (CI) around $\bar{Q}$ covers the true estimand Q. 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, since wider CIs indicate lower statistical power.

Diagnostic methods

- non-convergence in iterative algorithms is diagnosed using an identifier and a parameter
- the parameter can be any statistic that we track across iterations, for example the average imputed value per imputation (i.e., chain means)
- the identifier is a calculation of some sort that quantifies non-convergence
- identifiers are historically focused on either non-mixing between chains or non-stationarity within chains
- the popular non-mixing identifier rhat has recently been updated by Vehtari et al, and should now work for non-stationarity as well
- just to be sure, we also use autocorrelation to quantify trending within chains
- we apply these identifiers to the two parameters that we typically evaluate after imputation using visual inspection: chain means and chain variances

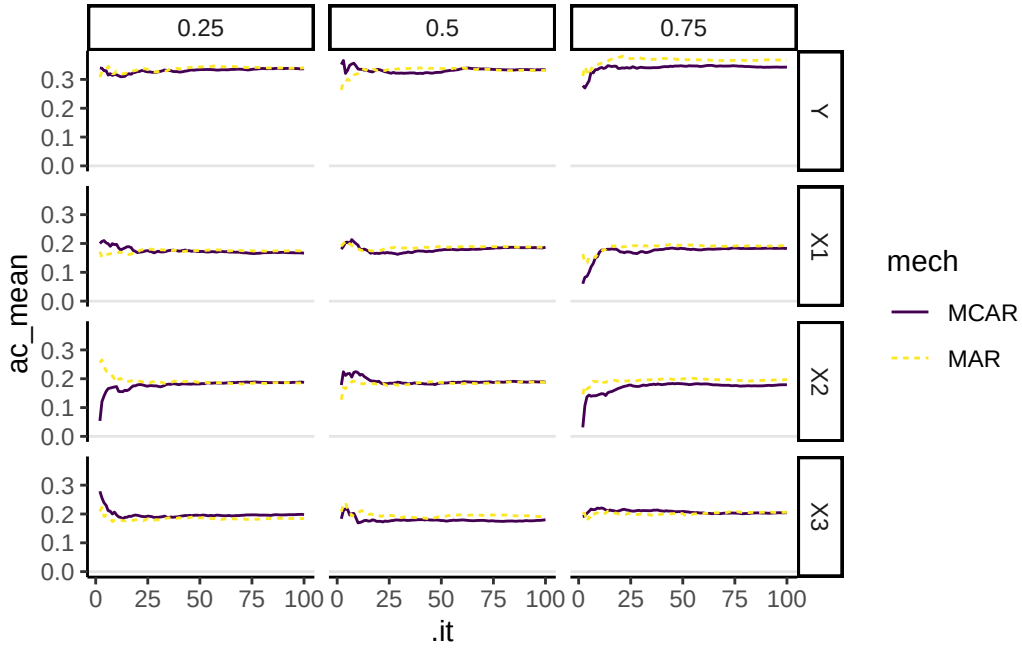
We use two non-convergence identifiers—autocorrelation and $\widehat{R}$—to diagnose non-convergence in the imputation models of the four incomplete variables. For each variable we apply the two identifiers on two parameters—chain means and chain variances—and four variables, resulting in 16 sets of identifier-parameter-variable pairs.

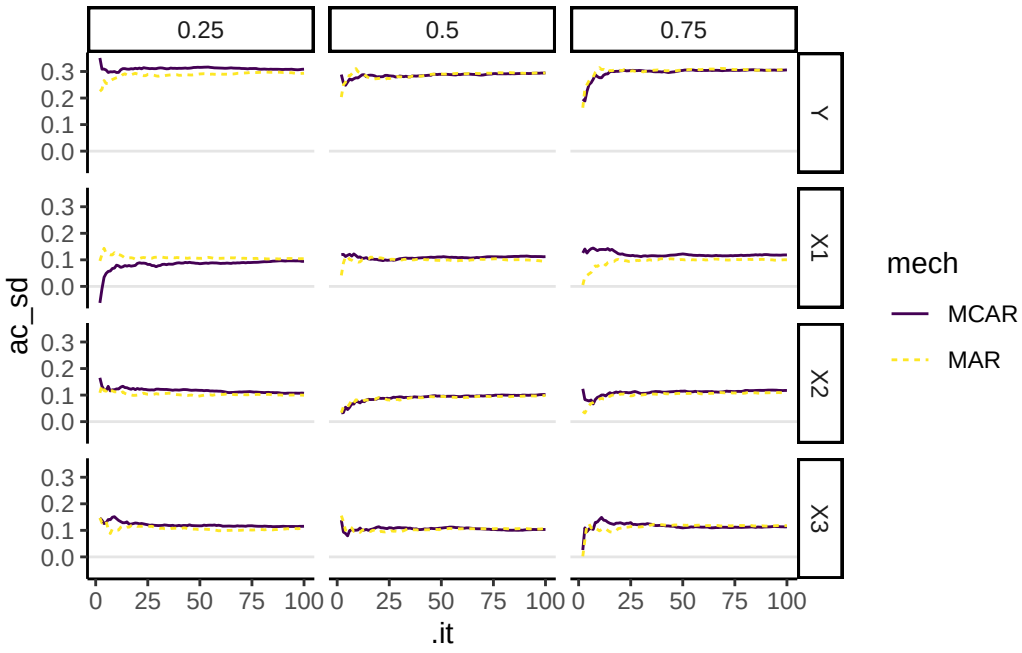
Simulation Results

The following figures display the simulation results for the sixteen diagnostic identifier-parameter-variable pairs, and sixteen performance measure-variable pairs, contrasted to the number of iterations in the imputation algorithm. Within the figures, we split the results according to the missingness conditions [TODO: missingness mechanisms as line types, and proportion of incomplete cases as colors?]. Note that these results are averages of the $n_{sim} = 2000$ simulation repetitions.

Diagnostic Methods

Figure @ref(fig:ac) shows the autocorrelations in the chain means (panel A) and chain variances (panel B).





Autocorrelation in the chain means decreases rapidly in the first few iterations (see @ref(fig:ac)A). The decrease is substantive until $n_{it} \geq 6$. This means that there is some initial trending within chains, but the average imputed value quickly reaches stationarity. These results hold irrespective of the missingness condition. Autocorrelation in the chain variances show us something similar (see @ref(fig:ac)B). The number of iterations that is required to reach non-improving autocorrelations is somewhat more ambiguous than for chain means, but generally around $n_{it} \geq 10$. We do not observe a systematic difference between missingness conditions here either.

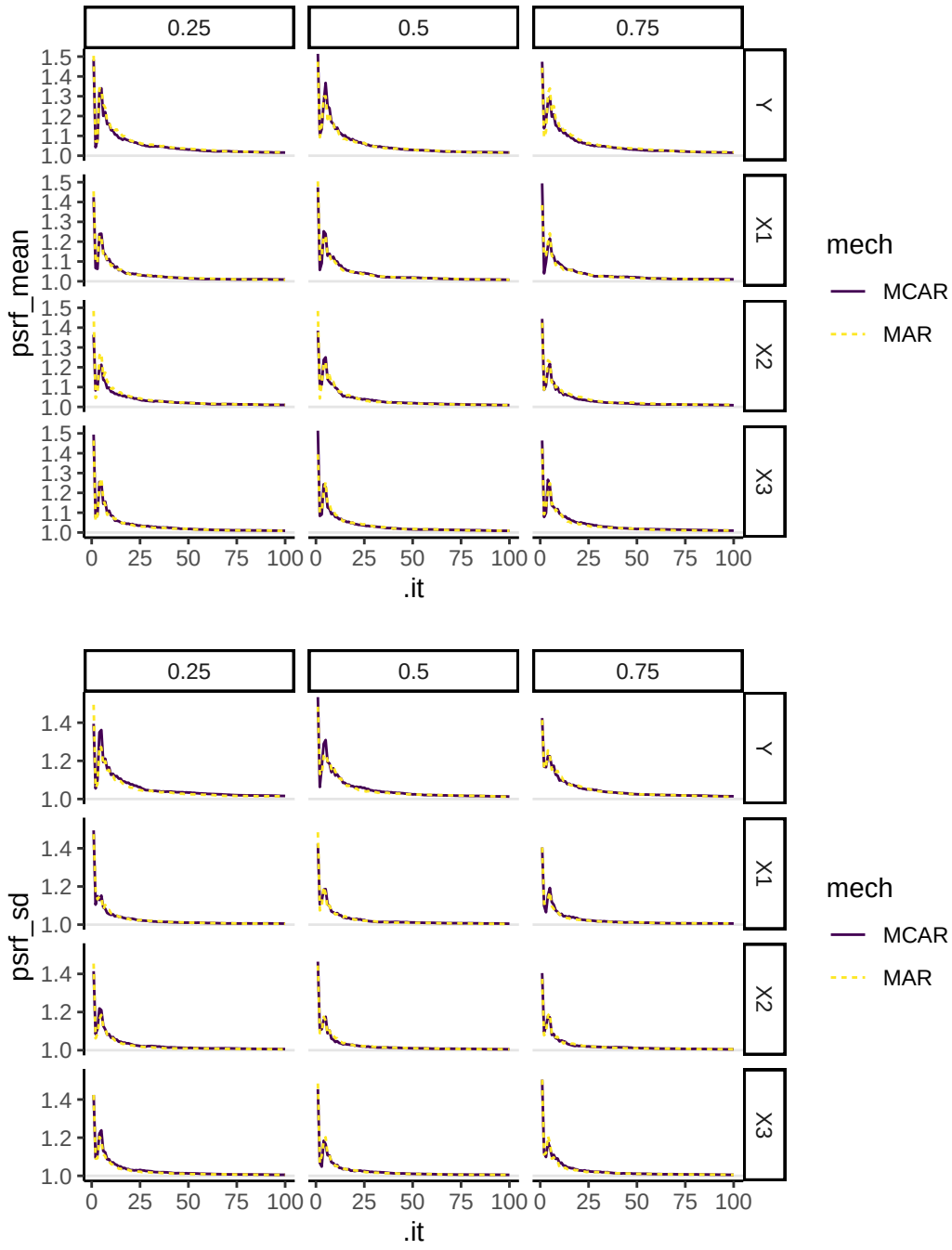
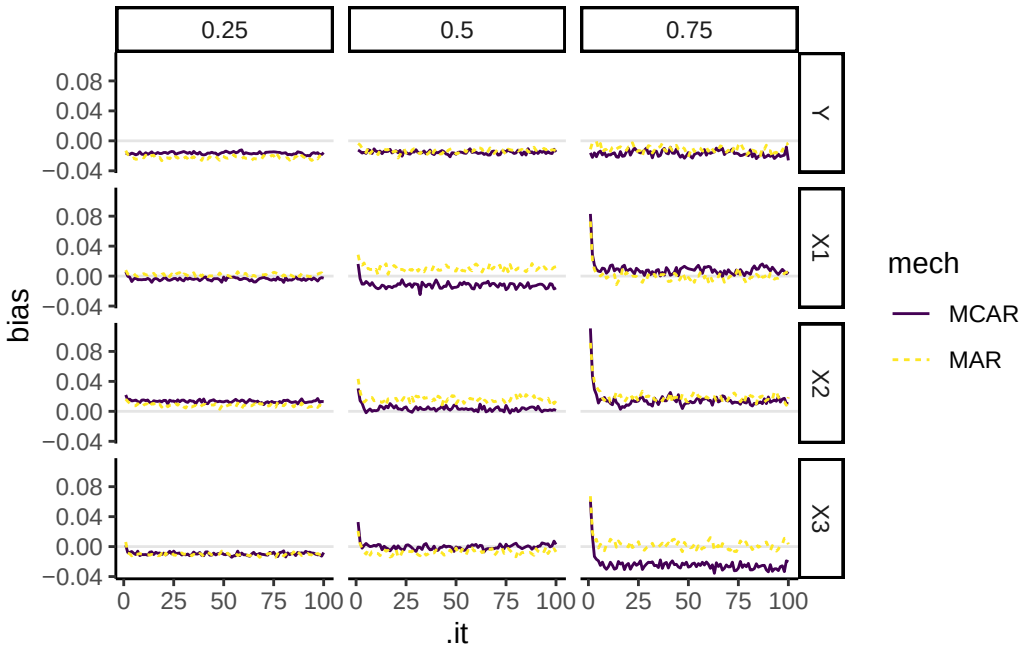


Figure @ref(fig:rh) shows the potential scale reduction factor in the chain means (panel A) and chain variances (panel B).

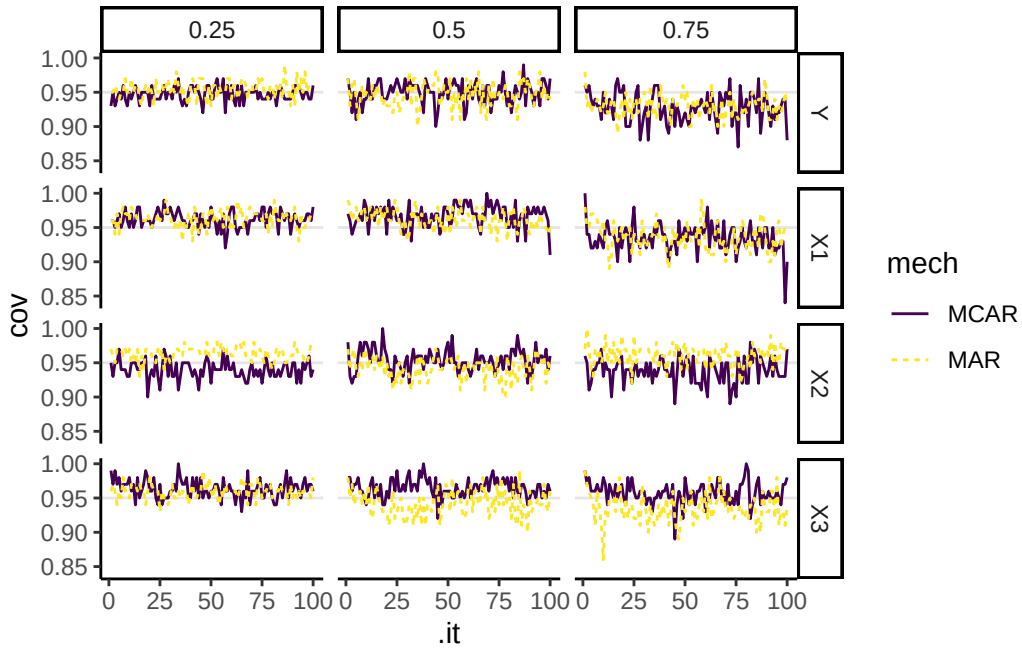
We observe that $\widehat{R}$-values of the chain means generally decrease as a function of the number of iterations (see @ref(fig:rh)A). An exception to this observation is a steep increase in iterations

$3 \leq n_{it} \leq 5$ [TODO: interpret?? due to initialization or is there really more mixing initially??]. After the first couple of iterations, the mixing between chain means generally improves until $n_{it} \geq 30$ to 40. There is no apparent differentiation between the missingness conditions. The mixing between chain variances mimics the mixing between chain means almost perfectly (see @ref(fig:rh)B). Irrespective of the missingness condition, the $\widehat{R}$-values taper off around $n_{it} \geq 30$. [The rhat plots all show some initialization before the fifth iteration: is rhat useful before that??]

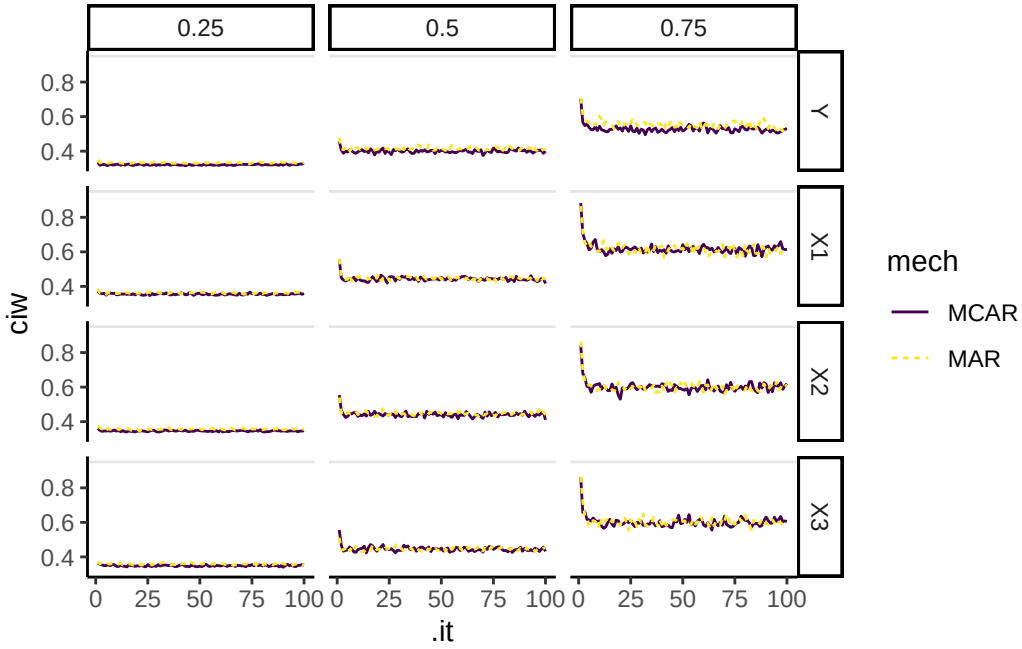
Performance Measures



- more missingness = more extreme bias, but also steeper decrease over iterations



- no clear trend



- more missingness = wider CIs, with steeper decrease in CIW, but not stababiling at same value between missingness conditions.

In Figure @ref(fig:perf) we show the performance measures: bias in the regression estimate (panel A), the empirical coverage rate of the regression estimate (panel C) and the average confidence interval width of this estimate (panel D).

We see that within a few iterations the bias in the regression estimate approaches zero (see @ref(fig:perf)A). When $n_{it} \geq 6$, even the worst-performing conditions (e.g., with a proportion of incomplete cases of 75%) produce stable, non-improving estimates [regression coefficient is underestimated because there is less info to estimate the relation??].

Nominal coverage is quickly reached (see @ref(fig:perf)C). After just three iterations, the coverage rates are non-improving in every missingness condition [but MNAR with 5% incomplete cases does not reach nominal coverage –> due to bias in the estimate in combination with very narrow CI (see CIW!)].

The average confidence interval width decreases quickly with every added iteration until a stable plateau is reached (see @ref(fig:perf)D). Depending on the proportion of incomplete cases this takes up-to $n_{it} \geq 9$.

Discussion

Our study found that—in the cases considered—inferential validity was achieved after five to ten iterations, much earlier than indicated by the non-convergence identifiers. Of course, it never hurts to iterate longer, but such calculations hardly bring added value.

- Convergence diagnostics keep improving substantially until n_it = 20-30
- Performance measures do not improve after n_it = 9
- [methodological explanation is that rhat and ac have a lag (few it to inform your statistic) –> will always indicate convergence slower than inferential validity is reached]

In short, the validity of iterative imputation stands or falls with algorithmic convergence—or so it's thought. We have shown that iterative imputation algorithms can yield correct outcomes even when a converged state has not yet formally been reached. Any further iterations just burn computational resources without improving the statistical inferences.

Case Study

- We use real data: the `boys` dataset from the `mice` package
- We are interested in predicting age from the other variables, in particular in the regression coefficient of `hgt`

- We compare non-convergence identified using visual inspection versus rhat in the chain variances, scientific estimate and lambda.
- The figures show results of a `mice` run with 20 iterations but otherwise default settings.

From the traceplot of the chain means (see @ref(fig:case)A) it seems that mixing improves up-to 10 iterations, while trending is only apparent in the first three iterations.

This figure (@ref(fig:case)B) shows that 7 iterations are required before the $\widehat{R}$-values of the chain means drop below the threshold for non-convergence.

The $\widehat{R}$-values for the scientific estimate reaches the threshold much sooner, when $n_{it} = 14$ (see @ref(fig:case)C).

According to the $\widehat{R}$-values with λ as parameter, at least 15 iterations are required (see @ref(fig:case)D).

Introduction (new-ish)

Iterative imputation has become the de facto standard to accommodate for missing data. The aim is usually to draw valid inferences, i.e. to get unbiased, confidence-valid estimates that incorporate the effects of the missingness. Such estimates are obtained with iterative imputation by separating the missing data problem from the scientific problem. The missing values are imputed (i.e., filled in) using some sort of algorithm. And subsequently, the scientific model of interest is performed on the completed data. To obtain valid scientific estimates, both the missing data problem and the scientific problem should be appropriately considered. The validity of this whole process naturally depends on the convergence of the algorithm that was used to to generate the imputations.

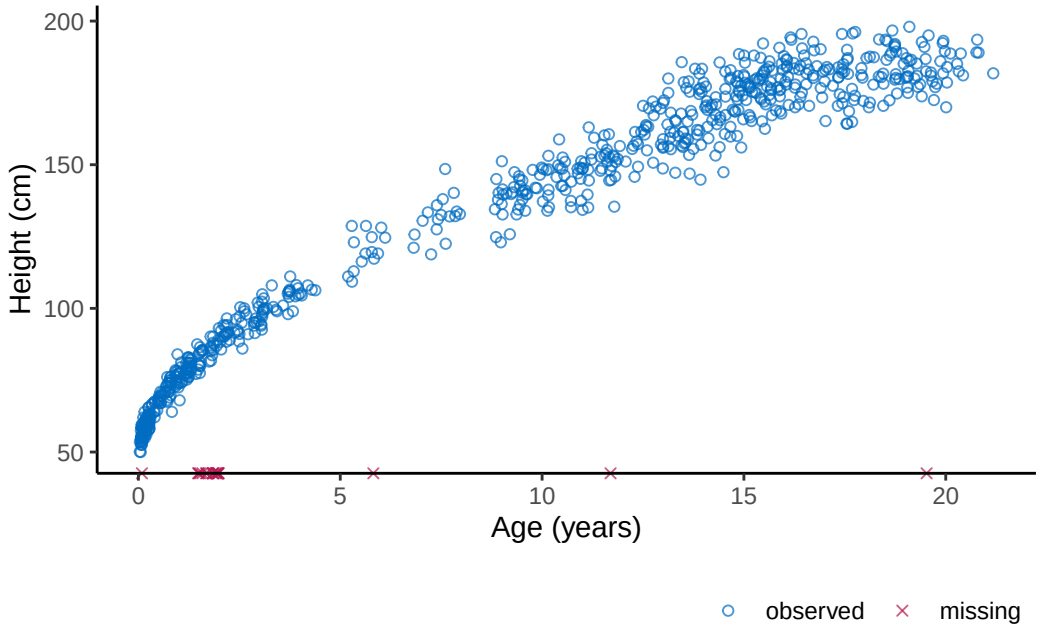
All inferences with imputed data rely on the convergence of the imputation algorithm. Yet, determining whether an algorithm has converged is not trivial. There has not been a systematic study on how to evaluate the convergence of iterative imputation algorithms. A widely accepted practice is visual inspection of the algorithm. Diagnosing convergence through visual inspection, however, may be undesirable for several reasons: 1) it may be challenging to the untrained eye, 2) only severely pathological cases of non-convergence may be diagnosed, and 3) there is not an objective measure that quantifies convergence (Van Buuren 2018). Therefore, a quantitative diagnostic method to assess convergence would be preferred.

It is challenging to arrive upon a single point at which convergence has been reached. Since the aim is to converge to a distribution and not to a single point, the algorithm may produce some fluctuations even after it has converged. Because of this property, it may be more desirable to focus on *non*-convergence. Fortunately, there are non-convergence identifiers for other iterative algorithms, but the validity of these identifiers has not been systematically evaluated on imputation algorithms.

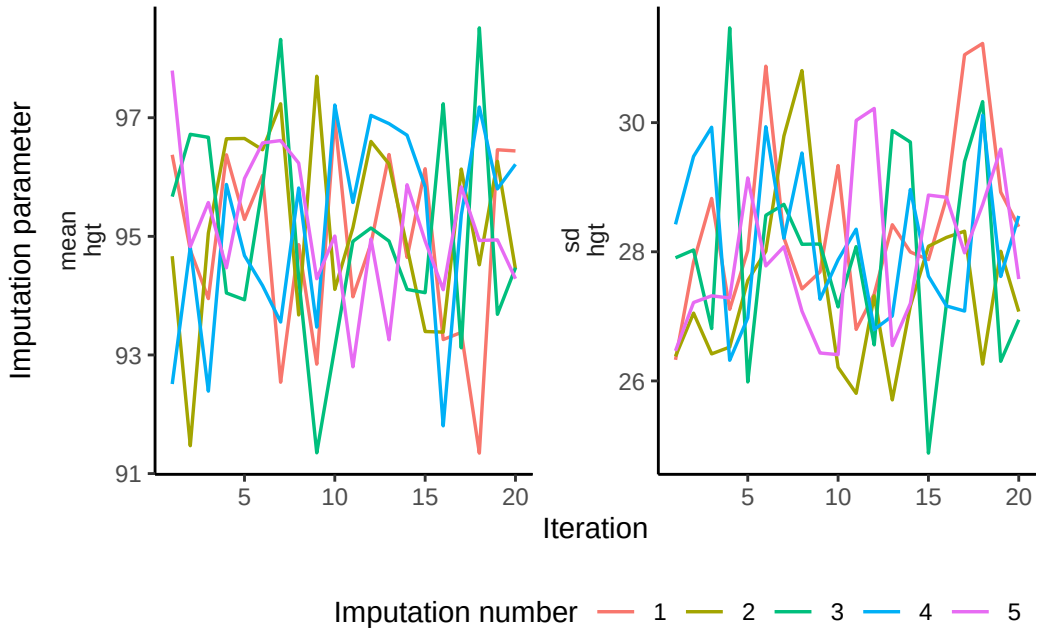
In this study, we explore different methods for identifying non-convergence in iterative imputation algorithms. We evaluate whether these methods are able to cover the extent of the non-convergence, and we also investigate the relation between non-convergence and the validity of the inferences. We translate the results of our simulation study into guidelines for practice, which we demonstrate by means of a motivating example.

Motivating Example

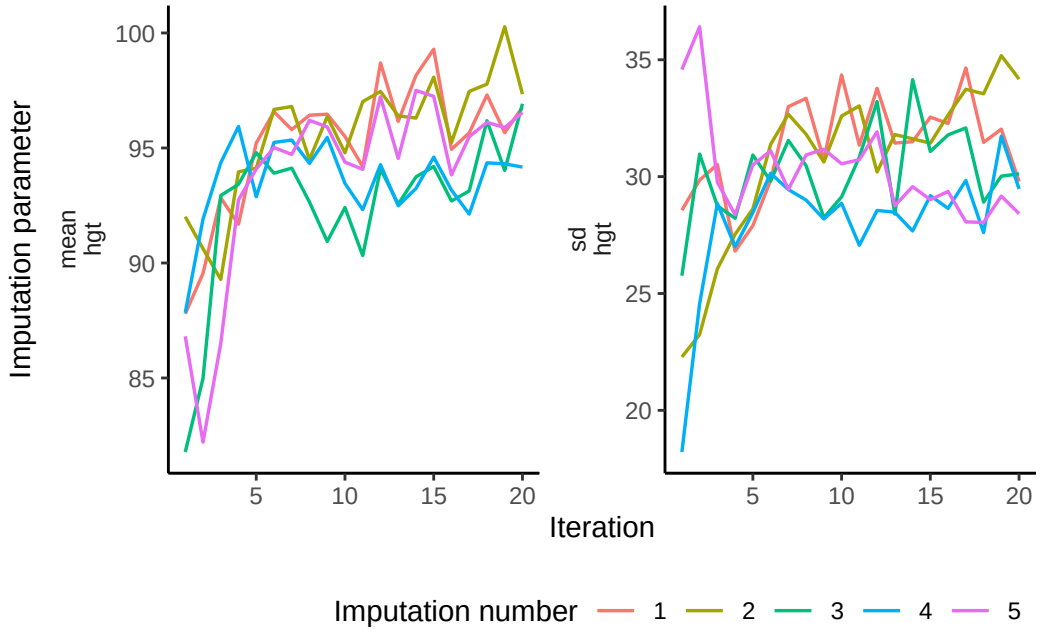
We use empirical incomplete data: the `boys` dataset from the `mice` package, which contains health-related data for 748 Dutch boys (Van Buuren and Groothuis-Oudshoorn 2011). Say, we're interested in the relation between children's heights and their respective ages, we could use a linear regression model to predict `hgt` from `age`. However, as figure XYZ shows, the variable `hgt` is not completely observed. To be able to analyze these data, we need to solve the missing data problem first.



The incomplete height variable can be imputed based on auxiliary variables, such as weight. After imputation with `mice`, one would conventionally inspect the trace plots for signs of non-convergence.



A mis-specified imputation model can lead to non-convergence in the imputation algorithm.



Introduction (old)

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, and statistical models cannot be 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 tool to draw valid inferences from incomplete data under many circumstances (Van Buuren 2018).

Figure **??** 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 $\hat{Q}_\ell$-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—scientific estimand Q (Rubin 1996).

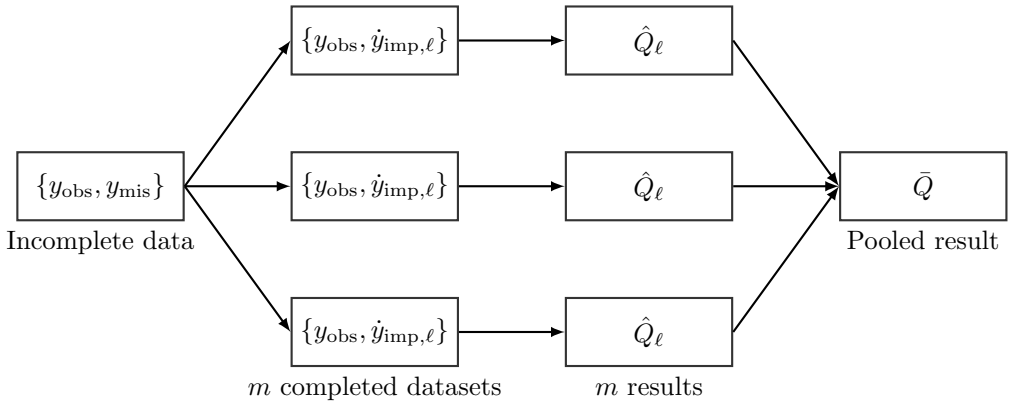


Figure 1: Scheme of the main steps in multiple imputation—from an incomplete dataset, to $m = 3$ multiply imputed datasets, to $m = 3$ estimated quantities of scientific interest $\hat{Q}$, to a single pooled estimate $\bar{Q}$.

A popular method to obtain imputations is to use the 'Multiple Imputation by Chained Equations' algorithm, shorthand 'MICE'(Van Buuren and Groothuis-Oudshoorn 2011). With

MICE, imputed values are drawn from the posterior predictive distribution of the missing values. The algorithm is named after its iterative nature: a multivariate distribution is obtained by iterating over a sequence of univariate imputations. Iteration, however, also introduces a potential threat to the validity of the imputations: What if the algorithm has 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 the convergence of imputation algorithms (Zhu and Raghunathan 2015; Takahashi 2017). Moreover, the behavior of such algorithms under certain default imputation models (e.g., 'predictive mean matching') is an entirely open question (Murray 2018). Therefore, algorithmic convergence should be monitored carefully—although this is 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, diagnostic methods may only identify signs of *non*-convergence (Hoff 2009). Several non-convergence diagnostics exist, but it is not known whether these are appropriate within the imputation framework.

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 address the impact of inducing non-convergence in iterative imputation algorithms through model-based simulation in `R` (R Core Team 2020). For reasons of brevity, we only focus on the iterative imputation algorithm implemented in the popular `mice` package in `R` (Van Buuren and Groothuis-Oudshoorn 2011). The aim of the simulation study is to determine whether unbiased, confidence-valid inferences may be obtained if the algorithm has not (yet) converged. And additionally, to evaluate the behavior and performance of several diagnostic methods to identify non-convergence. With that, we formulate an informed advice on when it is safe to conclude that the algorithm is converged *enough* for valid inferences. We translate the results of the study into guidelines for applied researchers, which may facilitate drawing valid inferences from incomplete data.

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 ($i = 1, 2, \dots, n$) on variable j ($j = 1, 2, \dots, k$) may be either observed or missing. The number of units i with at least one missing value, divided by the total number of units n, is called the proportion of incomplete cases p_{inc} in dataset y. 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 ($\ell = 1, 2, \dots, m$) and T is the number of iterations in the imputation algorithm

$(t = 1, 2, ..., T)$. The state-space of the algorithm at a certain iteration t may be summarized by a scalar summary θ (e.g., the average of the imputed values). The collection of θ-values between $t = 1$ (summarizing the state-space of the algorithm at initialization) and $t = T$ (summarizing the state-space for the imputed values) 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). In iterative imputation algorithms, mixing implies that imputation chains intermingle nicely, and stationarity is characterized by the absence of trending across iterations. If one of the two requirements is not met, we speak of non-convergence. Without mixing, chains may be 'stuck' at a local optimum, instead of sampling imputed values from the entire predictive posterior distribution of the missing values. The distribution of imputed values then differs across imputations. This may cause under-estimation of the variance between 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.

To illustrate what non-mixing and non-stationarity look like in iterative imputation algorithms, we reproduce an example from van Buuren (2018, § 6.5.2). Figure **??** displays two scenarios from the example. The panel 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. Each line portrays one imputation chain (i.e., the values of a scalar summary θ across iterations). The θ depicted here is the 'chain mean' of variable j, which is defined as the average of variable j in the m sets of imputations $\dot{y}_{\text{imp},\ell}$.

In the typical convergence scenario, the imputation chains intermingle nicely and there is little to no trending. In the non-convergence scenario, there is a lot of trending and some chains do not intermingle. Importantly, the chain means at the last iteration (the imputed values per imputation ℓ) are very different between the two scenarios. 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. It is obvious that non-convergence in this example impacts the distribution of the imputed values per imputation $\dot{y}_{\text{imp},\ell}$. This effect may translate into the distribution of the m sets of completed data $\{y_{\text{obs}}, \dot{y}_{\text{imp},\ell}\}$, which consequently affects the estimated quantities of scientific interest $\hat{Q}_\ell$, and finally the pooled estimate $\bar{Q}$. $\bar{Q}$ is then a biased, invalid estimate of Q. Therefore, it is important to reach algorithmic convergence in iterative imputation algorithms.

Identifying non-convergence

Currently, the recommended practice for evaluating the convergence of the MICE algorithm is through visual inspection. After running the imputation algorithm for a certain number of iterations, researchers are encouraged to produce traceplots. In a traceplot, a scalar summary of the state-space of the algorithm θ is plotted against the iteration number, as depicted in Figure **??**. The default scalar summaries to inspect for the MICE algorithm are chain means and chain variances. Non-convergence is diagnosed if the imputation chains are not freely intermingled with one another, or if the chains show definite trends (Van Buuren 2018, § 6.5.2).

Identifying non-convergence by inspecting traceplots may be undesirable for several reasons: 1) it may be challenging to the untrained eye, 2) only severely pathological cases of non-convergence may be diagnosed, and 3) there is not an objective measure that quantifies convergence (Van Buuren 2018, § 6.5.2). Moreover, traceplots are typically only used to inspect univariate θs. Monitoring univariate summaries of the state-space of the algorithm may be insufficient because MICE is not only concerned with a single column, but the entire multivariate distribution of the imputations. Ideally, we would monitor a multivariate θ.

A suggestion by Van Buuren (2018) for a multivariate θ to monitor is the estimated quantity of scientific interest $\hat{Q}$, which usually is multivariate in nature. This θ is computed as the estimate in each imputation, $\hat{Q}_\ell$, across iterations. Implementing this, however, might be somewhat too technical for empirical researchers. Besides, this scalar summary is not model-independent. That is, $\theta = \hat{Q}$ is not universal to all complete data problems, while one of the advantages of iterative imputation is that the missing data problem and scientific problem are solved independently. 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. On these grounds, such a θ can be considered insufficient too.

As alternative, van Buuren (2018, § 4.5.2) proposed multivariate evaluation of the MICE algorithm through eigenvalue decomposition, building on the work of MacKay and Mac Kay (2003). This technique may yield a model-independent, multivariate scalar summary to monitor, but it is not implemented in the `mice` package (Van Buuren and Groothuis-Oudshoorn 2011).

A novel scalar summary

To monitor non-convergence in iterative imputation algorithms, we may summarize the state-space of the algorithm with several θs. The downside to the current θs 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 θ that summarizes the multivariate nature of the data.

We propose λ_1 as such a scalar summary. We define λ_1 as the first eigenvalue of the variance-covariance matrix in the m completed datasets. The eigenvalues of a variance-covariance

matrix S summarize the total set of covariances in the data. Let $\lambda_{1,\ell} \geq \lambda_{2,\ell} \geq ... \geq \lambda_{j,\ell}$ be the eigenvalues of S_ℓ in each imputation. The first eigenvalue, $\lambda_{1,\ell}$, then summarizes the largest possible amount of covariance in each completed dataset $\{y_{\text{obs}}, \dot{y}_{\text{imp},\ell}\}$. By definition, λ_1-values are equal to the variance of the first component that would be obtained by performing principal component analysis (PCA) on the completed data. As scalar summary, λ_1 has the appealing property that it is not dependent on the model of scientific interest, yet still summarizes the multivariate state-space of the algorithm.

In this study, we consider each of the four θs that we discussed. Namely, two univariate scalar summaries (chain mean and chain variance), a model-dependent multivariate summary $\hat{Q}$, and the novel model-independent multivariate summary λ_1.

Non-convergence diagnostics

There are many diagnostic tools to identify non-convergence in iterative (MCMC) algorithms (Brooks and Gelman 1998; El Adlouni, Favre, and Bobée 2006). We consider only two of them that may be appropriate for imputation algorithms—one to monitor signs of non-mixing, and one for non-stationarity. As recommended by e.g. Cowles and Carlin (1996) we will use the potential scale reduction factor $\widehat{R}$ to evaluate mixing ['Gelman-Rubin statistic'; Gelman and Rubin (1992)], and autocorrelation to diagnose trending [AC; Schafer (1997); Gelman et al. (2013)]. 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. 2021). Therefore, we will evaluate the appropriateness of both $\widehat{R}$ versions, in addition to AC. Other methods are outside the scope of this study because they e.g. assume that values within chains represent independent samples, whereas the MICE algorithm only uses the final iteration to produce imputations.

Potential scale reduction factor

In 2019, Vehtari et al. proposed an updated version of the potential scale reduction factor $\widehat{R}$, originally coined in 1992. The adapted version would be better suited to detect non-mixing in the tails of distributions, and even identify non-stationarity. This version uses three transformations on the scalar summary θ, before computing $\widehat{R}$-values. Namely, rank-normalization, folding, and localization. The definition of the diagnostic itself is equal to the original $\widehat{R}$. Therefore, we may follow Vehtari et al.'s formulation (2019, p. 5) to define both $\widehat{R}$ versions. Let m be the total number of chains, T the number of iterations per chain (where $T \geq 2$), and θ the scalar summary of interest. For each chain ($\ell = 1, 2, ..., m$), we estimate the variance of θ, and average these to obtain within-chain variance W.

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

We then estimate between-chain variance B (note that we diverge from the typical notation in MI, where B denotes the variance between the estimated quantities of scientific interest $\hat{Q}_\ell$). B is defined as the variance of the collection of average θs per chain:

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

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

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

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). The assumption underlying this interpretation is 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, B, is then equivalent to the variance within chains, W, and $\widehat{R}$-values will be close to one. High $\widehat{R}$-values thus indicate non-convergence.

Autocorrelation

Autocorrelation is defined as the correlation between two subsequent θ-values within the same chain (Lynch 2007, 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. Following the same notation as for $\widehat{R}$,

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

We can interpret AC-values as a measure of non-stationarity. If there is dependence between subsequent θ-values in imputation chains, AC-values are non-zero. Positive AC-values occur when θ-values are recurring (i.e., high θ-values are followed by high θ-values, and low θ-values are followed by low θ-values). Recurrence within imputation chains may lead to trending. Negative AC-values occur when θ-values of subsequent iterations are less similar, or diverge from one another. Divergence within imputation poses no threat to the convergence of the algorithm—it may even speed up convergence. Complete stationarity is reached when $AC = 0$. As non-convergence diagnostic, our interest is in positive AC-values.

Thresholds

It is unlikely that iterative algorithms such as MICE 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. The aim, therefore, is to reach approximate convergence (Gelman et al. 2013). 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 little dependency between subsequent iterations of imputation chains ($AC > 0$). In practice, we diagnose non-convergence when approximate convergence is violated, i.e., when $\widehat{R}$ and AC exceed a certain threshold. The conventional thresholds to diagnose non-mixing are $\widehat{R} > 1.2$ (Gelman and Rubin 1992) or $\widehat{R} > 1.1$ (Gelman et al. 2013). Vehtari et al. (2021) proposed a much 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.

In practice

Before we evaluate the performance of the non-convergence diagnostics $\widehat{R}$ and AC quantitatively through simulation, we first assess their appropriateness qualitatively. We do this by applying them to the example of pathological non-convergence that we reproduced form Van Buuren (2018). Ideally, the diagnostics are as informative as, or better than visual inspection of the traceplots. The methods should at least indicate worse performance (higher $\widehat{R}$- and AC-values) for the scenario with pathological non-convergence, compared to the typically converged algorithm.

Each panel in Figure **??** depicts one method to identify non-convergence, applied to the two scenarios from Figure **??**. Panel A consists of the two traceplots that may be evaluated through visual inspection. Panel B shows two versions of the AC: 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 displays the traditional computation of $\widehat{R}$ conform Gelman and Rubin (original), and in panel D we see $\widehat{R}$ as computed by implementing Vehtari et al.'s recommendations (adapted).

By visually inspecting the imputation chains in panel A, we conclude that the two scenarios show very different convergence. The typically converged algorithm initially portrays some signs of non-mixing (around $t = 2$), but intermingles nicely overall. Additionally, there is very little trending in this scenario. The algorithm with pathological non-convergence shows severe non-mixing, although this gradually improves (beyond $t = 7$). In this scenario, we see a lot of trending initially (up-to $t = 6$), after which the chains reached a somewhat more stationary state.

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 scenarios (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 $t = 2$ for both scenarios. However, if we look at the chain means of the non-convergence scenario, there should be signs of trending up-to iteration number seven. After consulting the documentation on `stats::acf()`, 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 (presented with solid lines in panel B). These AC-values behave as expected from visual inspection. We will, therefore, only consider manually calculated AC-values as non-convergence diagnostic.

In panel C, we do not see a lot of difference between the two scenarios either. The $\widehat{R}$-values are even somewhat lower for the non-convergence scenario than for the typical convergence. And apparently, the large variance causes an increase in $\widehat{R}$-values. Convergence seems to worsen with every additional iteration. This paints the wrong picture. An explanation for this behavior may be that under severe trending, the assumption of over-dispersion is violated. Taken together, this version of $\widehat{R}$ is sub-optimal for assessing non-convergence in MI.

The $\widehat{R}$-values in panel D are conform our expectations. The adapted version of $\widehat{R}$ depicted here does indicate less signs of non-convergence as the number of iterations goes up. An exception to this general trend is the 'dip' around $t = 3$. As diagnostic tool, this may lead to the incorrect conclusion that the non-convergence scenario has superior convergence compared to the typical scenario. We can explain this as a downside of the low number of iterations in iterative imputation: the adapted version of $\widehat{R}$ can only be completely employed if the number of iterations is at least four. Otherwise, it is not possible to perform all three transformations to the chains of θ-values. For low values of t, the $\widehat{R}$-values conform Vehtari et al. are, therefore, more similar to the original $\widehat{R}$ by Gelman and Rubin.

In short, both non-convergence diagnostics have a version that may be appropriate for iterative imputation algorithms. We will compute $\widehat{R}$ conform Vehtari et al. (2021), and calculate AC manually. The diagnostics will be applied to the four types of scalar summaries θ described earlier: chain means, chain variances, a quantity of scientific interest, and the first eigenvalue of the variance-covariance matrix. We evaluate the performance of these eight sets of diagnostic methods (two diagnostics; four θs) through simulation.

Simulation study

The simulation study aims to determine to what extend non-convergence affects iterative imputation algorithms. Specifically, we are interested in the validity of inferences drawn from incomplete data using the MICE algorithm. And, whether invalid inferences due to non-convergence may be detected using eight sets of diagnostic methods. If the estimated quantities of scientific interest $\bar{Q}$ are unbiased and confidence-valid estimates of Q, we will conclude that the algorithm is sufficiently converged for practical purposes. In contrast to this, we defined *approximate* convergence as the most converged state that the algorithm can obtain. Continued iteration after reaching approximate convergence does not yield better mixing or stationarity (indicated by non-improving $\widehat{R}$- and AC-values). Continued iteration after obtaining valid inferences may lead to a more converged state of the algorithm, but not better estimates.

To induce non-convergence, we consider two sets of simulation conditions: 'missingness' and 'early stopping'. Missingness refers to the severity of the missing data problem, which we determine by the proportion of incomplete cases in dataset y, p_{inc}. Early stopping is defined by the number of iterations in the imputation algorithm, T.

The missingness conditions are chosen to reflect the difficulty of the missingness problem. The underlying assumption is that a low proportion of incomplete cases p_{inc} leads to quick algorithmic convergence, since there is a lot of information in the observed data. A higher p_{inc} yields slower convergence—unless there is so little information in y_{obs} that this is outweighed by the random component in the imputation algorithm. Then, convergence to a stable but highly variable state may be reached instantly. Still, we expect that missingness conditions with higher p_{inc} will result in more signs of non-convergence.

The assumption inherent to the early stopping conditions 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 starting values [which are sampled randomly from the set of observed datapoints; Van Buuren (2018)]. As the number of iterations increases, the imputation chains should become independent of the initial values, until approximate convergence is reached (i.e., when an additional iteration does not lead to a more converged state). We expect that we can induce non-convergence by early stopping in at least those conditions where T is smaller than the default number of iterations in `mice`, $T = 5$ (Van Buuren and Groothuis-Oudshoorn 2011).

Hypotheses

1. We expect that simulation conditions with a high proportion of incomplete cases p_{inc} and a low number of iterations T will more often result in biased, invalid estimates of the quantities of scientific interest Q.

2. We hypothesize that $\widehat{R}$ and AC will correctly identify signs of non-convergence in those simulation conditions where any $\bar{Q}$ is *not* an unbiased and confidence-valid estimate of Q.

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, Ardington, and Leibbrandt 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, signs of non-convergence can be under-estimated by $\widehat{R}$, in exceptional cases where the initial values of the algorithm are not appropriately over-dispersed (Brooks and Gelman 1998, 437). In `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. 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 4.4.2; R Core Team (2020)]. The number of simulation repetitions is 2000, and the simulation set-up is summarized in the pseudo-code below. The complete `R` script of the simulation study is available from github.com/hanneoberman/MissingThePoint [TODO: make Zenodo DOI].

```
# pseudo-code of simulation
for (number of simulation runs) {
  1. simulate complete data
  for (missingness conditions) {
    2. create missingness
    for (number of iterations) {
      3. impute missingness
      4. estimate quantities of scientific interest
      5. apply performance measures to the estimates
      6. compute non-convergence diagnostics
```

```
    }
  }
7. aggregate outcomes across simulation runs
```

Data-generating mechanism.

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

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix} \right].$$

In each simulation repetition, 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

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

where $\epsilon \sim \mathcal{N}(0, 1)$. This results in a dataset y, where $y_{\text{mis}} = \emptyset$. In other words, y contains the complete set of observations for all units i on all variables j (where $j = Y, X_1, X_2, X_3$).

Scientific estimands.

We consider four multivariate quantities of scientific interest (Qs). Namely, the regression coefficients of the analysis model,

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

where Y' is the expected value of the outcome. From the data-generating mechanism, we obtain the true values of the scientific estimands: $\beta_0 = 0$, $\beta_1 = \beta_2 = \beta_3 = 1$. The scientific estimands are estimated by $\bar{Q}$ in each simulation condition.

Simulation conditions.

We consider two sets of simulation conditions, 'missingness conditions' and 'early stopping' conditions. The simulation study is a fully factorial design, as visualized in Table XYZ.

In each simulation repetition, we *ampute* the complete dataset conform six missingness conditions: two missing data mechanisms and three proportions of incomplete cases. The missing data mechanisms are 'missing completely at random' [MCAR; Rubin (1987)], where the probability to be missing is the same for all $N \times k$ cells in y, and a right-tailed 'missing at random' (MAR) mechanism, where the probability to be missing is a function of the observed data and higher values are more likely to be missing. The proportion of incomplete cases p_{inc} is set to 25%, 50%, and 75%.

We consider all possible multivariate patterns of missingness, as visualized in Figure **??**.

We use the 'mice' package [function `mice::ampute()`; Van Buuren and Groothuis-Oudshoorn (2011)] to obtain an incomplete dataset $\{y_{\text{obs}}, y_{\text{mis}}\}$ for each of the six missingness conditions.

We impute the missing values in $\{y_{\text{obs}}, \dot{y}_{\text{mis}}\}$ according to the early stopping conditions. That is, we vary the number of iterations in the imputation algorithm between one and one hundred ($T = 1, 2, ., 100$). All imputation procedures are performed using `mice` [function `mice()`; Van Buuren and Groothuis-Oudshoorn (2011)], with Bayesian linear regression imputation, and five imputation chains ($m = 5$). This results in $m = 5$ sets of imputations for each of the one hundred early stopping conditions, $\dot{y}_{\text{imp},\ell}$.

Subsequently, we obtain the completed data per imputation, $\{y_{\text{obs}}, \dot{y}_{\text{imp},\ell}\}$, by combining the imputed data with the observed data [function `mice::complete()`; Van Buuren and Groothuis-Oudshoorn (2011)]. Estimates of scientific quantities Qs are obtained after performing multiple linear regression (function `stats::lm()`, R Core Team 2020). We pool the resulting m estimates, $\hat{Q}_\ell$, into a single $\bar{Q}$ conform Rubin's rules [function `mice::pool()`; Van Buuren and Groothuis-Oudshoorn (2011)].

Methods.

We consider eight sets of diagnostic methods to identify non-convergence, by applying $\widehat{R}$ and AC on four different θs. Univariate θs are obtained from $\dot{y}_{\text{imp},\ell}$. The model-independent multivariate θ, λ_1, is obtained from $\{y_{\text{obs}}, \dot{y}_{\text{imp},\ell}\}$ as the variance in the first PCA component [function `stats::princomp`; R]. The model-dependent multivariate θ is one of the quantities of scientific interest: the estimated regression coefficients in $\{y_{\text{obs}}, \dot{y}_{\text{imp},\ell}\}$. We apply the two non-convergence diagnostics on the θs by implementing the adapted version of $\widehat{R}$ and by manually programming the AC function.

Performance measures.

As recommended by Van Buuren (2018), we evaluate the performance of the imputation algorithm with bias, coverage rate (CR), and confidence interval width (CIW). Bias is calculated for each Q, whereas CR and CIW are only obtained for $Q = \beta$. We calculate bias as $\bar{Q} - Q$. 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

$$\text{CI} = \bar{Q} \pm t_{(m-1)} \times \text{SE}_{\bar{Q}},$$

where $t_{(M-1)}$ is the quantile of a t-distribution with $m-1$ degrees of freedom, and $\text{SE}_{\bar{Q}}$ is the square root of the pooled variance estimate. If we obtain nominal coverage (CR = 95%), we can conclude that $\bar{Q}$ is a confidence-valid estimate of Q. 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, since wider CIs indicate lower statistical power.

We evaluate the diagnostic methods to identify non-convergence against these performance measures. $\widehat{R}$ and AC should identify non-convergence in simulation conditions, where $\bar{Q}$ is *not* an unbiased, confidence-valid estimate of Q.

Results

Our results show the performance of the MICE algorithm under different conditions of missingness and early stopping. For reasons of brevity, we only discuss results for the worst-performing estimates in terms of bias. For univariate scientific estimands ($Q = \mu$ and $Q = \sigma$) we observe the largest bias in the outcome variable Y. And for the regression coefficients ($Q = \beta$) the bias is most pronounced in β_1 (the effect of X_1 on Y). Since there is just one estimate for $Q = r^2$ to evaluate, we consider $Q = \mu_Y, \sigma_Y, r^2, \beta_1$. In the figures, we present all missingness conditions, but only early stopping conditions where $1 \leq T \leq 50$. That is, the results are more or less stable for conditions where $T \geq 30$. Full results are available from github.com/hanneoberman/MissingThePoint.

Quantities of scientific interest

Figure **??** displays the influence of missingness and early stopping on the estimated quantities of scientific interest, Qs. In the first row of the figure, we see the bias in the estimated descriptive statistics ($Q = \mu_Y$ and $Q = \sigma_Y$; panels A and B). The second row shows the bias in the multivariate scientific estimands ($Q = r^2$ and $Q = \beta_1$; C and D). The third row consists of the coverage rate (CR) and confidence interval width (CIW) of the estimated regression coefficient ($Q = \beta_1$; E and F).

$Q = \mu$.

The estimated univariate means seem unaffected by early stopping. This implies that approximately unbiased estimates may be obtained with as little as one iteration ($T \geq 1$). The bias in $\bar{Q}$ depends solely on the proportion of incomplete cases, with $p_{\text{inc}} \geq .75$ leading to more extreme biases than other conditions. Completely unbiased estimates are only obtained in conditions where $p_{\text{inc}} \leq .50$. This is curious since the MCAR missingness mechanism should yield unbiased univariate estimates without employing multiple imputation. Note however, that the magnitude of the bias in these conditions 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, therefore, conclude that non-convergence does not substantially impact the validity of the inferences for $Q = \mu$.

$Q = \sigma$.

The estimated standard deviations are not substantially impacted by early stopping either. While the estimates for $T = 1$ are not equal to other conditions, the bias is actually *less* severe than for $T \geq 2$. Therefore, no iteration at all is needed to obtain approximately unbiased estimates for $Q = \sigma$. Completely unbiased estimates are only obtained in conditions where $p_{\text{inc}} = .05$. The other missingness conditions are impacted by non-convergence in the order of increasing p_{inc}. Similar to $Q = \mu$, the magnitude of the bias in $Q = \sigma$ is negligible. We, therefore, conclude that neither one of the univariate Qs is affected substantially by non-convergence.

$Q = r^2$.

In contrast to the univariate estimates, the estimated coefficient of determination is clearly affected by early stopping, see Figure **??**C. The bias in Q appears to decrease with each additional iteration, while the magnitude of the bias still depends on the proportion of incomplete cases. Therefore, we see that the number of iterations that is necessary to reach stable, non-improving estimates also differs across missingness conditions. In conditions where $p_{\text{inc}} = .05$, estimates are unbiased after one iteration. The highest number of iterations necessary to reach approximate unbiasedness is seven ($p_{\text{inc}} = .95$; $T \geq 7$). Completely unbiased estimates are only obtained in conditions where $p_{\text{inc}} \leq .25$. This implies that even for biased estimates, $T \geq 7$ would suffice to reach a stable solution.

$Q = \beta$.

For the estimated regression coefficients, we first consider bias, before discussing CR and CIW. As Figure **??**D shows, the bias in $\bar{Q}$ is affected by both the proportion of incomplete cases and the number of iterations. Similar to $Q = r^2$, we observe approximately unbiased estimates after at least one, and at most seven iterations, depending on p_{inc}. Completely unbiased

estimates are only obtained in conditions where $T \geq 3$ and $p_{\text{inc}} \leq .50$. Despite persistent bias in conditions where $p_{\text{inc}} \geq .75$, the coverage rates are non-improving after just three iterations, irrespective of p_{inc}. We conclude that any condition where $T \geq 3$ yields approximately nominal coverage rates, and thus confidence-valid estimates. The nominal coverages can be explained by the CIWs. The CIWs depicted in Figure **??**F are conform the theoretical foundation of MI: CIs are wider in conditions with higher proportions of incomplete cases (i.e., there is less information in the data, and thus more uncertainty due to missingness). Since conditions with higher p_{inc} result in both wider CIs and more severe bias, the true value of Q may be included in the CI, despite of the bias in $\bar{Q}$. We conclude that approximately unbiased estimates may be obtained in conditions where $T \geq 7$, whereas confidence-valid estimates require at most three iterations.

Summary.

These results demonstrate that the estimates of univariate scientific estimands Q are not impacted by early stopping of the MICE algorithm or the proportion of incomplete cases in y. Unbiased estimates may be obtained after just one iteration. Multivariate estimates, by contrast, are affected by both the number of iterations and the proportion of missing cases. Completely unbiased estimates are only obtained under low to moderate missingness ($p_{\text{inc}} \leq .50$), after at most three iterations. We observe approximately unbiased estimates after at most seven iterations (for any p_{inc} considered). This implies that the algorithm produces stable, non-improving estimates when $T \geq 7$.

Non-convergence diagnostics

Figure **??** displays results of the eight sets of diagnostic methods. The two columns in the figure represent the non-convergence diagnostics $\widehat{R}$ and AC; the rows depict the four scalar summaries θ under consideration. The first row in Figure **??** thus shows $\widehat{R}$ and AC applied on the chain means (panels A and B). In the second row, we see the diagnostics applied on the chain variances (C and D). The third row depicts the same for $\theta = \hat{Q}$ (E and F). And in the last row, we see the diagnostics applied on the novel θ to consider: λ_1 (G and H). We evaluate the performance of the methods by establishing whether they correctly identify conditions in which $\bar{Q}$ is *not* an unbiased, confidence-valid estimate of Q. As we concluded in the last section, substantially biased estimates were only obtained in conditions where $T \leq 6$, and we observed non-nominal coverage only when $T \leq 3$. However, there was some persistent bias in conditions where $p_{\text{inc}} \geq .75$, irrespective of the number of iterations. If the diagnostic methods are appropriate for MI, we will detect non-convergence in these conditions.

$\theta =$ **chain mean.**

As expected, $\widehat{R}$ lowers with the number of iterations. After an initial 'dip' in conditions where $T = 3$, we observe a gradual decrease in $\widehat{R}$-values with every additional iteration. This implies improving convergence, until the decrease tapers off after 30 to 40 iterations. Using the threshold $\widehat{R} > 1.2$, we diagnose non-convergence in conditions where $T \leq 7$, whereas the threshold 1.1 is exceeded when $T \leq 13$. The very strict threshold 1.01 is surpassed in all conditions $T \leq 100$ (not shown), and therefore not informative under the current specifications. With this set of methods, we fail to identify non-convergence in conditions with persistent bias due to missingness.

Similarly, AC decreases with higher Ts. AC indicates improving stationarity in conditions where $T \leq 6$. The AC-values do not exceed the threshold defined by statistical significance of the AC-values. Based on this threshold, we fail to diagnose non-convergence in any condition.

$\theta =$ **chain variance.**

As panel C and D in Figure **??** show, the results of this set of methods are highly similar to those with chain means as θ. This holds for both $\widehat{R}$ and AC. The only apparent difference is in the $\widehat{R}$ threshold 1.1. Instead of diagnosing non-convergence in conditions where $T \leq 13$, we now diagnose this when $T \leq 11$. All other observations are equivalent.

$\theta = \hat{Q}$**.**

The $\widehat{R}$-values in Figure **??**E show a similar trend across iterations as we observed for chain means and chain variances. Only conditions with a very high proportion of missing cases ($p_{\text{inc}} \geq .75$) diverge from the earlier observations. The $\widehat{R}$-values in these conditions do not taper off after 30 to 40 iterations, but rather between 40 to 50 iterations. The highest number of iterations at which non-convergence is still diagnosed according to the thresholds are 11 for $\widehat{R} > 1.2$, 23 for $\widehat{R} > 1.1$, and 100 for $\widehat{R} > 1.01$ (not shown).

The AC-values also follow earlier observations, with the same exception as for $\widehat{R}$ (when $p_{\text{inc}} \geq .75$). To reach maximum stationarity, these conditions require at most seven iterations instead of five. Moreover, the AC-values in these conditions exceed the threshold to diagnose non-convergence. Interestingly, this only occurs after ten or even thirty iterations, while we would expect more signs of trending early in the iterations. Since the AC-values are not improving or worsening when $T \geq 7$, we conclude that non-convergence is incorrectly diagnosed based on the threshold.

$\theta = \lambda_1$.

Once again, we observe $\widehat{R}$-values with a similar trend across iterations. For this θ, the results differ only in the most extreme missingness condition. When $p_{\text{inc}} = .95$, we diagnose non-convergence at $T \leq 9$ according to $\widehat{R} > 1.2$, $T \leq 19$ for $\widehat{R} > 1.1$, and 100 for $\widehat{R} > 1.01$ (not shown). The 'dip' in $\widehat{R}$-values at $T = 3$ is even more pronounced than with other θs. Some $\widehat{R}$-values even overcome the threshold of 1.2. If we would terminate the algorithm at $T = 3$, we would incorrectly conclude that the algorithm reached approximate convergence, according to this threshold.

AC-values are systematically higher for this θ than for other scalar summaries. The AC-values also taper off at a later point in the iterations. This suggests improving stationarity up-to ten, or even thirty iterations, depending on the missingness condition. According to the threshold, we should diagnose non-convergence at some point in any missingness condition. However, this occurs only after reasonably decreasing AC-values are obtained—again suggesting that non-convergence is incorrectly diagnosed.

Summary.

Overall, the methods to diagnose non-convergence perform as expected: they indicate more signs of non-convergence in conditions with worse performance in terms of bias and confidence-validity. Under low to moderate missingness, it does not seem to matter on which θ the non-convergence diagnostics are applied. If we look at complete unbiasedness, however, we notice that univariate θs fail to diagnose the persistent bias in conditions where $p_{\text{inc}} \geq .75$.

The number of iterations necessary to obtain approximately unbiased, confidence-valid estimates corresponds to the $\widehat{R}$ threshold 1.2. The thresholds 1.1 and 1.01 seem too strict compared to the validity of the inferences. The threshold to diagnose significant AC-values does not appear to be appropriate in at least the current set-up, and perhaps in iterative imputation in general. A better heuristic to diagnose non-stationarity with AC may be through evaluation of the AC-values across iterations: If the values do not substantially decrease with T, approximate stationarity may be concluded.

Discussion

Before drawing any further conclusions, let us first return to the hypotheses that we formulated for the simulation study.

1. As hypothesized, biased, invalid estimates of quantities of scientific interest occurred more often in simulation conditions with a high proportion of incomplete cases p_{inc} and a low number of iterations T. However, this only holds for multivariate scientific estimands. Univariate Qs are not substantially impacted by early stopping.

2. Our results also support the hypothesis that $\widehat{R}$ and AC would correctly identify signs of non-convergence. We observed higher $\widehat{R}$ and AC-values in conditions where $\bar{Q}$s were biased, invalid estimates of Qs.

3. We hypothesized that the recommended thresholds to diagnose non-convergence with $\widehat{R}$ would be too stringent for iterative imputation applications, compared to the number of iterations necessary for statistically valid inferences. This hypothesis is partially supported: On the one hand, the 1.01 and 1.1 thresholds indeed seem to over-estimate the signs of non-convergence. On the other hand, however, we observed that the traditional threshold of 1.2 roughly corresponds to the number of iterations required to obtain approximately unbiased, confidence-valid estimates.

4. The results of this simulation study do not support the hypothesis that high AC-values are implausible in iterative imputation algorithms with typical convergence. We expected that the randomness induced by the algorithm would effectively mitigate the risk of dependency within chains after a few iterations. In this study, however, we observed AC-values as high as theoretically possible ($AC = 1$). The second part of the hypothesis is not refuted: the AC-values did decrease quickly as a function of T.

5. Our final hypothesis was that multivariate θs would be better at detecting non-convergence than univariate θs. Multivariate θs did indeed show superior performance under certain conditions (e.g., multivariate Qs and extreme missingness).

Conclusion

We have shown that non-convergence in iterative imputation algorithms goes hand in hand with biased, invalid estimates—at least for the multivariate quantities of scientific interest considered in this study. Identifying non-convergence may be a crucial aspect of drawing valid statistical inferences from incomplete data. We concluded that the current practice of visually inspecting non-convergence through traceplots of univariate scalar summaries of the state-space of the algorithm does not suffice. We, therefore, considered eight sets of diagnostic methods to identify non-convergence: two non-convergence diagnostics ($\widehat{R}$ and AC), two univariate θs, a model-dependent multivariate θ, and a novel, model-independent multivariate θ. Signs of non-convergence due to early stopping were identified with each of the methods. Bias due to high proportions of incomplete cases was only identified with multivariate θs.

Since we obtained approximately unbiased, confidence-valid estimates after at most seven iterations, we conclude that the $\widehat{R}$ threshold 1.2 is the most appropriate diagnostic cut-off. The threshold to diagnose non-stationarity with AC does not seem to apply. However, some signs of non-convergence were detected in simulation conditions many more iterations. Under moderate missingness ($p_{\text{inc}} \leq .50$) $\widehat{R}$-values decreased substantially until 30 to 40 iterations, which implies that mixing in the algorithm still improved with each additional iteration. AC-values only improved until about six iterations, suggesting minimal improvement in trending was obtained beyond $T = 6$ in the current simulation set-up.

The main finding of this study is that valid inferences may be obtained much quicker than approximate algorithmic convergence is reached. Under the current specifications, univariate Qs did not require algorithmic convergence at all. They are unbiased almost instantly after the algorithm is initiated. Approximately unbiased, confidence-valid estimates of multivariate Qs were obtained after a maximum of seven iterations. Continued iterations beyond $T = 7$ did not yield better estimates.

Implications

Iterative imputation algorithms such as MICE are known to yield valid results under severe missingness. With this study, we have shown that valid inferences can also be obtained under a combination of severe missingness and early stopping. Based on our results, the traditional threshold of $\widehat{R} > 1.2$ would be appropriate for diagnosing non-convergence in iterative imputation algorithms. This is, however, not in line with the recent recommendations by Vehtari et al. (2021) to lower the threshold to 1.01. The discrepancy between our conclusion and Vehtari et al. may be explained by the nature of iterative imputation algorithms. In the limit, iterative imputation algorithms have the same characteristics as other MCMC algorithms. But, in imputation procedures, a part of the distribution is already determined by the observed data, whereas the entire distribution is unknown in many other MCMC applications. Since we combine a known distribution with an unknown distribution, valid estimates may be reached much sooner. Convergence may, therefore, be diagnosed at a less stringent threshold.

The threshold to diagnose non-convergence with AC does not seem appropriate at all in iterative imputation algorithms. A better diagnostic cut-off would be the number of iterations at which an additional iteration does not substantially decrease the AC-values. In practice, this implies that the default number of iterations in MICE is not sufficient. AC can only be computed if $T \geq 3$, while the current default in the `mice` package is five (Van Buuren and Groothuis-Oudshoorn 2011). Identifying a reasonable decrease in the AC-values across iterations (an 'elbow') requires more than three observations. We, therefore, suggest to increase the default number of iterations in MICE to ten. The computational cost of five additional iterations has become less of a burden since MICE was introduced (2011). Moreover, an increased number of iterations would grant the opportunity to exclude the first iterations from evaluation with $\widehat{R}$. Excluding $T \leq 3$ provides empirical researchers with a less ambiguous heuristic to diagnose non-convergence, because we expect the initial 'dip' in $\widehat{R}$-values to disappear.

The potential scale reduction factor metric assumes over-dispersed starting values of the iterative algorithm. The MICE algorithm does not start in an over-dispersed state. MICE does not rely on starting values for parameters at all: instead, the algorithm is initialized based on a 'zeroth' iteration, where missing values are filled in using a random draw from observed values. Under MCAR, this would typically not yield an over-dispersed state at all, depending on the parameter of interest: means and variances are not affected, regression coefficients and other multivariate parameters start biased downwards (because sampling starting values

distorts multivariate structures in the data). Whether this accrues to over-dispersion is questionable. Under MAR (or MNAR), some over-dispersion might occur when the observed data and imputed data differ impactfully. The algorithm should 'escape' the initial state (based on the starting values drawn from observed data alone). Over iterations, all parameters of interest (e.g. means, variances, multivariate estimands) should converge towards a stable state. Moreover, the MICE algorithm is initialized with more information than a typical MCMC algorithm: parameter estimates depend not only on imputed data, but also on observed data that does not change over iterations. Therefore, the initial state of the algorithm is 'too good' to observe a relative decrease in $\widehat{R}$.

Recommendations for empirical researchers

Based on our results, we formulate several recommendations for empirical researchers who employ iterative imputation to draw inferences from incomplete data. We suggest that non-convergence may still be evaluated visually, but in addition to inspecting univariate summaries of the state-space of the algorithm (θs, e.g., chain means), multivariate θs should be considered. We propose the following steps:

1. First, check traceplots of the default θs (e.g., chain means and chain variances) for signs of pathological non-convergence. Adjust the imputation model if necessary.
2. Subsequently, decide which multivariate θs to track across iterations. The novel θ that we propose in this study, λ_1 is scientific model-independent and may thus always be employed. Alternatively, specify your own scalar summary of interest (see e.g., Van Buuren 2018). Monitor these θs through visual inspection, or using a non-convergence diagnostic.
3. Compute non-convergence diagnostics $\widehat{R}$ and AC. Do *not* use the original implementation to calculate $\widehat{R}$ (Gelman and Rubin 1992), or the `R` function `stats::acf()` to compute AC (R Core Team 2020). Instead, calculate $\widehat{R}$ conform Vehtari et al. (2021) and compute autocorrelations manually (see e.g., github.com/hanneoberman/MissingThePoint).
4. And finally, use the threshold $\widehat{R} > 1.2$ to diagnose non-mixing, and assess stationarity by plotting the autocorrelation over iterations. Keep iterating until the threshold for mixing is overcome, and until reasonably decreasing asymptotic AC-values are obtained. At that point, inferences are unlikely to improve by continued iteration.

Limitations

Much remains unknown about non-convergence in iterative imputation algorithms. Even though we have demonstrated the appropriateness of $\widehat{R}$ and AC as non-convergence diagnostics in this study, results may not extrapolate to other situations. In this study, we only

considered the iterative imputation algorithm implemented in the `mice` package. The performance of the non-convergence diagnostics may depend on the characteristics of this specific algorithm. A potential threat to the ability to detect non-mixing with $\widehat{R}$, for example, is the assumption of over-dispersed initial states. In `mice`, the algorithm is initiated by sampling starting values at random from the observed data, whereas other imputation software may use the average of the observed cases. The latter could lead to a systematical under-estimation of the variability in the first few iterations and violate the assumption of over-dispersion. In those cases, it would be more difficult to detect non-convergence with $\widehat{R}$. Therefore, the diagnostic might not be appropriate outside of `mice`.

Moreover, the current simulation conditions were restricted to a single missingness mechanism. Proper performance under a 'missing completely at random' (MCAR) mechanism is a necessary condition for any missing data method. It does not, however, guarantee equal performance under different missingness mechanisms. Future research should determine the performance of $\widehat{R}$ and AC under 'missing at random' and 'missing not at random' mechanisms. Another parameter to consider in future research is the choice of the imputation method. As Murray (2018) concluded, the behavior of algorithms such as MICE under certain default imputation models is still an open research question. We have investigated the behavior of MICE under only one type of model (Bayesian linear regression). Different imputations models might converge more poorly.

The imputation model may even be mis-specified on purpose, to induce non-convergence of different severity levels. Qualitatively, we have shown that $\widehat{R}$ and AC are appropriate under clear violation of convergence. Quantitatively, we have only demonstrated that they can identify signs of non-convergence under MCAR, when the imputation model is a correctly specified Bayesian linear regression model.

In short, we have shown that iterative imputation algorithms 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. Our study found that—in the cases considered—inferential validity was achieved after five to ten 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.

Box, George E. P., Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2015. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.

Brooks, Stephen P., and Andrew Gelman. 1998. "General Methods for Monitoring Convergence of Iterative Simulations." *Journal of Computational and Graphical Statistics* 7 (4): 434–55. https://doi.org/10.1080/10618600.1998.10474787.

Cowles, Mary Kathryn, and Bradley P Carlin. 1996. "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review." *Journal of the American Statistical Association* 91 (434): 883–904.

El Adlouni, Salaheddine, Anne-Catherine Favre, and Bernard Bobée. 2006. "Comparison of Methodologies to Assess the Convergence of Markov Chain Monte Carlo Methods." *Computational Statistics & Data Analysis* 50 (10): 2685–2701. https://doi.org/10.1016/j.csda.2005.04.018.

Gelman, Andrew, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. *Bayesian Data Analysis*. Philadelphia, PA, United States: CRC Press LLC.

Gelman, Andrew, and Donald B. Rubin. 1992. "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science* 7 (4): 457–72. https://doi.org/10.1214/ss/1177011136.

Genz, Alan, and Frank Bretz. 2009. *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Heidelberg: Springer-Verlag.

Hoff, Peter D. 2009. *A First Course in Bayesian Statistical Methods*. Springer Texts in Statistics. New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-92407-6.

Lacerda, Miguel, Cally Ardington, and Murray Leibbrandt. 2007. "Sequential Regression Multiple Imputation for Incomplete Multivariate Data Using Markov Chain Monte Carlo." University of Cape Town, South Africa.

Lynch, Scott M. 2007. *Introduction to Applied Bayesian Statistics and Estimation for Social Scientists*. Springer Science & Business Media.

MacKay, David JC, and David JC Mac Kay. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge university press.

Murray, Jared S. 2018. "Multiple Imputation: A Review of Practical and Theoretical Findings." *Statistical Science* 33 (2): 142–59. https://doi.org/10.1214/18-STS644.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Raghunathan, Trivellore, and Irina Bondarenko. 2007. "Diagnostics for Multiple Imputations." {{SSRN Scholarly Paper}} ID 1031750. Rochester, NY: Social Science Research Network.

Rubin, Donald B. 1976. "Inference and Missing Data." *Biometrika* 63 (3): 581–92. https://doi.org/10.2307/2335739.

———. 1987. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Mathematical Statistics Applied Probability and Statistics. New York, NY: Wiley.

———. 1996. "Multiple Imputation After 18+ Years." *Journal of the American Statistical Association* 91 (434): 473–89. https://doi.org/10.2307/2291635.

Schafer, Joseph L. 1997. *Analysis of Incomplete Multivariate Data*. Chapman and Hall/CRC.

Takahashi, Masayoshi. 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 (0): 37. https://doi.org/10.5334/dsj-2017-037.

Van Buuren, Stef. 2018. *Flexible Imputation of Missing Data*. Chapman and Hall/CRC.

Van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2011. "Mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software* 45 (1): 1–67. https://doi.org/10.18637/jss.v045.i03.

Vehtari, Aki, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. 2021. "Rank-Normalization, Folding, and Localization: An Improved R^ for Assessing Convergence of MCMC." *Bayesian Analysis* -1 (-1): 1–38. https://doi.org/10.1214/20-

BA1221.

Zhu, Jian, and Trivellore E. Raghunathan. 2015. "Convergence Properties of a Sequential Regression Multiple Imputation Algorithm." *Journal of the American Statistical Association* 110 (511): 1112–24. https://doi.org/10.1080/01621459.2014.948117.