

Missing The Point

Hanne Oberman

7-9-2020

Missing The Point: Non-Convergence in Iterative Imputation Algorithms

Notation in this manuscript:

- square brackets are comments for myself/stuff I need to review
- bullet points are things I need to expand

Abstract

has become the de facto approach

in data science?

Iterative imputation is a popular tool to accommodate 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. Since there is no consensus on how to evaluate the convergence properties of iterative imputation algorithms, our study provides insight into identifying non-convergence. We found that—in the cases considered—inferential validity was 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.

Intro

De facto standard

- ① often used
- ② convergence no consensus
- ③ therefore identify nonconvergence
- ④ more important is if the estimates are biased
- ⑤ How does nonconvergence affect the validity of our conclusions?

Missing data pose a ubiquitous threat to anyone who aims to obtain unbiased, confidence-valid statistical inferences. A popular technique to accommodate missing data is to 'impute' (i.e., fill in) any missing values in an incomplete dataset. Imputation procedures like 'Multiple Imputation by Chained Equations' (MICE) have proven to be powerful tools to draw valid inference under many missing data circumstances (Rubin 1987; Van Buuren 2018). To obtain imputations, most imputation software packages rely on iterative algorithms.

zon leeft niet lekker tot verrijke..

With iterative imputation, the validity of the inference depends on the state-space of the algorithm at the final iteration. This 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 using the imputed data? These remain open questions since the convergence properties of iterative imputation algorithms have not been systematically studied (Van Buuren, 2018, 6.5.2). While there is no scientific consensus on how to evaluate the convergence of imputation algorithms (Liu et al. 2014; Zhu and Raghunathan 2015; Takahashi 2017), the current practice is to visually inspect imputations for signs of non-convergence.


Waarom niet \hat{R} enz; dat mist nu

Identifying non-convergence through visual inspection 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). Therefore, a quantitative diagnostic method to identify non-convergence would be preferred. ✓

In this paper, we explore diagnostic methods for iterative imputation algorithms. For reasons of brevity, we focus on the iterative imputation algorithm implemented in the popular `mice` package (Van Buuren and Groothuis-Oudshoorn 2011) in `R` (R Core Team 2020). We consider two non-convergence identifiers for iterative algorithms: autocorrelation (conform Lynch, 2007, p. 147) and potential scale reduction factor \widehat{R} (conform Vehtari et al., 2019, p. 5). Aside from the usual parameters to monitor—chain means and chain variances—we also investigate convergence in the multivariate state-space of the algorithm.

We propose a novel multivariate parameter to check for non-convergence in iterative algorithms. We aim to show which method is the most informative about non-convergence in iterative imputation [explain that method = parameter + diagnostic + interpretation].

Identifying non-convergence

- Iterative imputation is a sort of MCMC algorithm, so it makes sense to investigate non-convergence in a similar fashion to typical MCMC algorithms.
- With MCMC, the generated values will vary even after convergence, so there is not a unique point at which convergence is established (Gelman et al. 2013). Therefore, diagnostic methods may only identify signs of *non*-convergence (Hoff 2009). Non-convergence occurs when one of two requirements for convergence is not met.
- There are two requirements for convergence of iterative algorithms: mixing and stationarity (Gelman et al. 2013). Mixing implies that generated values intermingle nicely, and stationarity is characterized by the absence of trending between successive draws.
- What are the consequences? → bias, under-coverage [refer to van Buuren (2018) instead of reproducing this]
- We consider autocorrelation and \widehat{R} , and monitor four parameters: chain means, chain variances, a scientific estimate, and lambda.
- Explain lambda 

Simulation Set-Up

We investigate non-convergence in iterative imputation through model-based simulation in `R` (version 4.0.3; 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 (<https://github.com/hanneoberman/MissingThePoint>).

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

```

for each simulation repetition (1:1000)
  1. simulate complete data
  for each missingness condition (1:9)
    2. create missingness
    for each iteration (1:50)
      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

```

Aims

With this simulation, we assess the impact of non-convergence on the validity of scientific estimates obtained using `mice` (Van Buuren and Groothuis-Oudshoorn 2011). Inferential validity is reached when estimates are both unbiased and have nominal coverage across simulation repetitions ($n_{sim} = 1000$). To induce non-convergence, we terminate the iterative algorithm at different imputation chain lengths ($n_{it} = 1, 2, \dots, 50$). We differentiate between nine different missingness scenarios: three missingness mechanisms versus three proportions of incomplete cases [remove here if already defined above/below??].

Data generating mechanism

Data are generated in each simulation repetition for a complete set of $n_{obs} = 1000$ cases (i.e., before inducing missingness). We define three multivariately normal random variables, let

$$\begin{pmatrix} Y \\ X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & & \\ 0.5 & 1 & \\ -0.5 & 0.5 & 1 \end{pmatrix} \right].$$

[Add mvtnorm package here??] The complete set is amputed according to nine missingness conditions. We use a 3×3 factorial design consisting of three missingness mechanisms and three proportions of incomplete cases. [Briefly describe MCAR, MAR, MNAR + 25, 50, 75% of cases incomplete (not using 5% anymore, because according to Vink, n.d., 25% is more representative of missing data problems in the social sciences). And add the `ampute` function?]

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 to predict outcome variable Y from the other two variables

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

where Y' is the expected value of the outcome. Our estimands are the regression coefficient β_1 and coefficient of determination r^2 that we obtain after pooling the regression results across the imputations.

Methods

We use eight different diagnostic methods to identify non-convergence: a combination of two non-convergence identifiers—autocorrelation and \widehat{R} —and four parameters of interest—chain means, chain variances, a scientific estimate, and the novel parameter that we propose, λ .

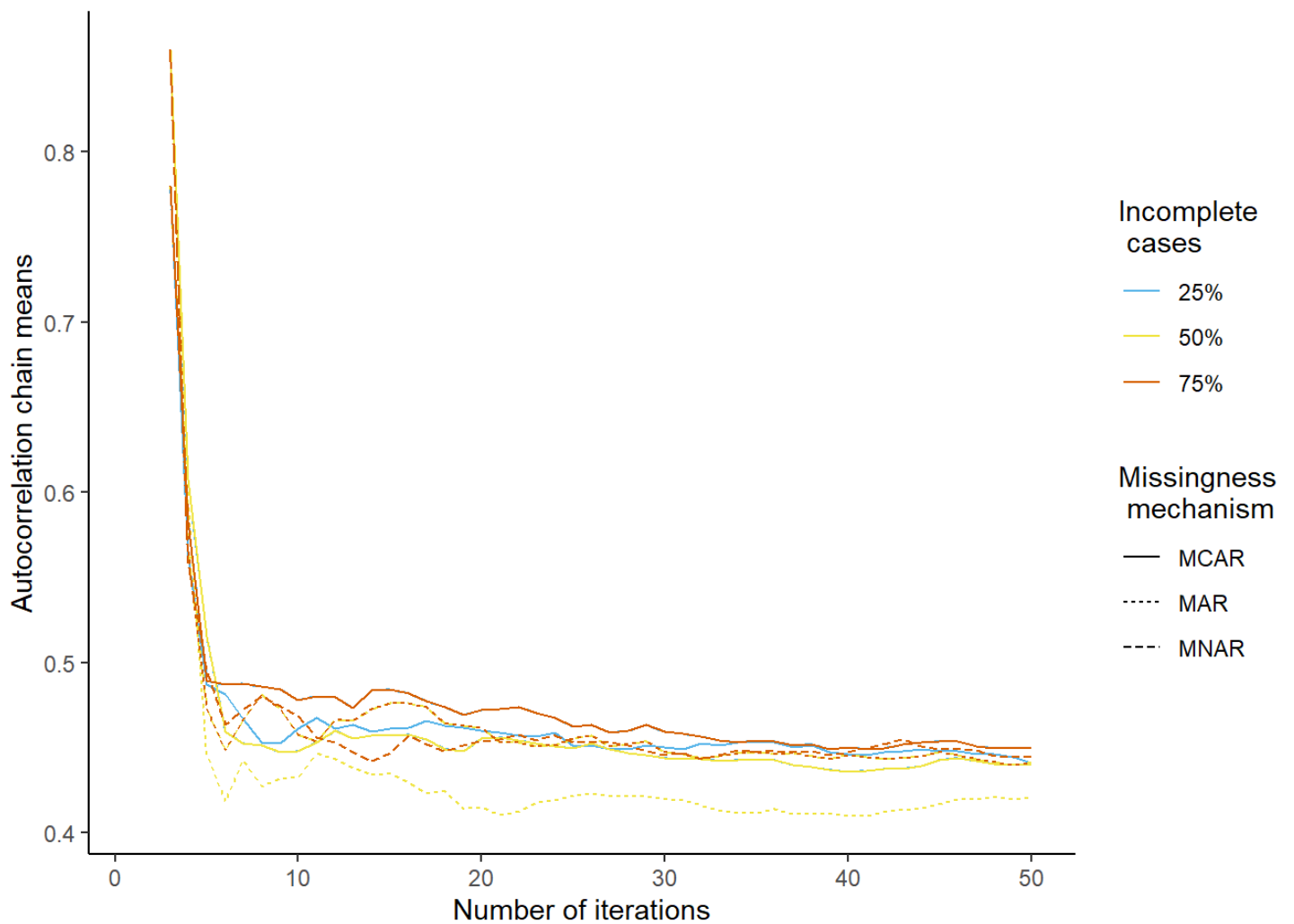
Performance measures

As recommended by Van Buuren (2018), our performance measures are bias ($E(\bar{Q}) - Q$), average confidence interval width ($E(\bar{Q}_{LL} - \bar{Q}_{UL})$, where LL and UL denote the lower and upper limit of the 95% confidence interval respectively) and empirical coverage rate ($Pr(\bar{Q}_{LL} \geq Q \geq \bar{Q}_{UL})$) of the estimands [or just of the regression estimate?? otherwise add the ciw and cov of r^2 too!! and check if definition of ciw is correct].

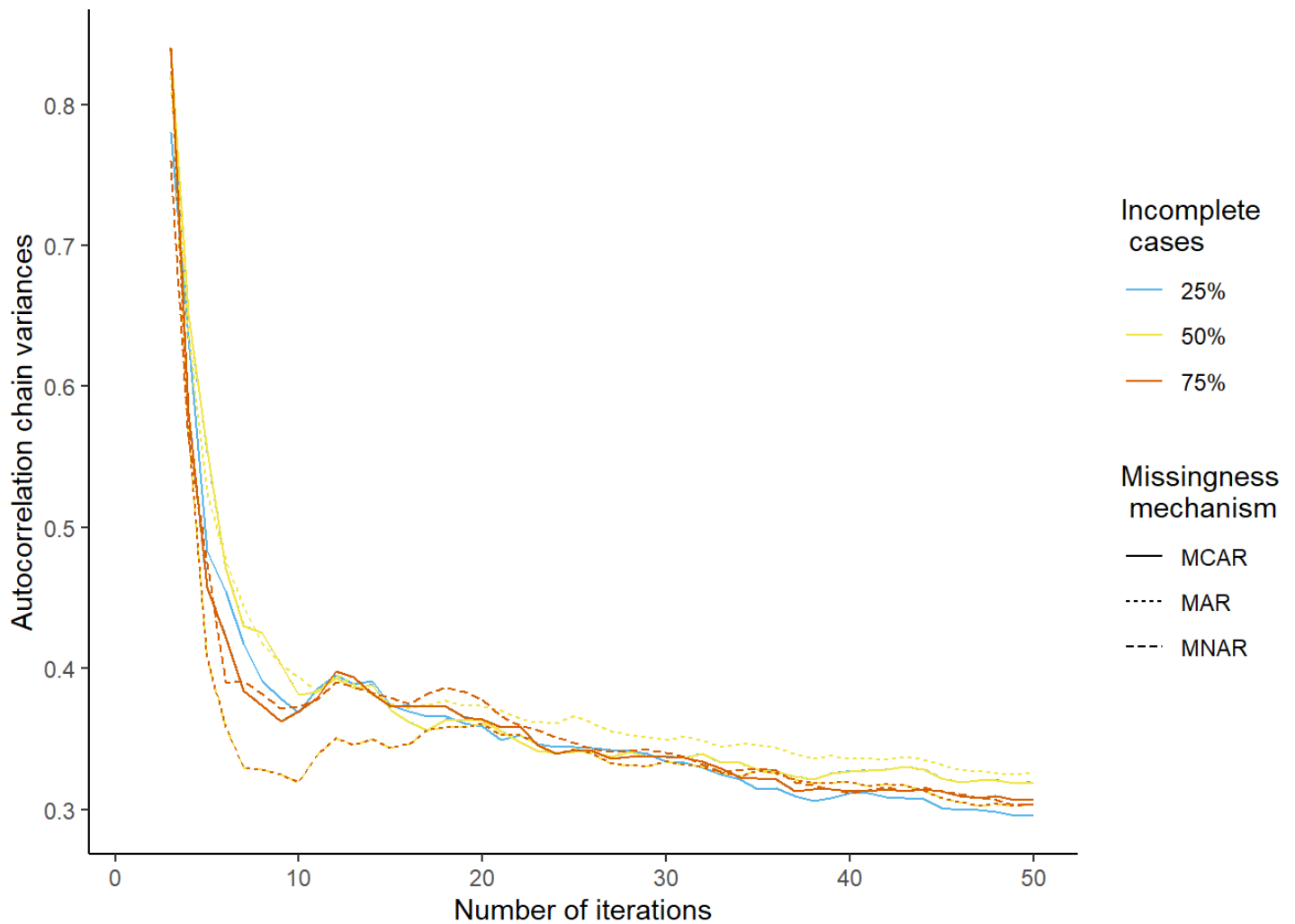
Simulation Results

The following figures display the simulation results for the eight diagnostic methods and four performance measures, contrasted to the number of iterations in the imputation algorithm. Within the figures, we split the results according to the missingness conditions [missingness mechanisms as line types, and proportion of incomplete cases as colors]. Note that these results are averages of the $n_{sim} = 1000$ simulation repetitions.

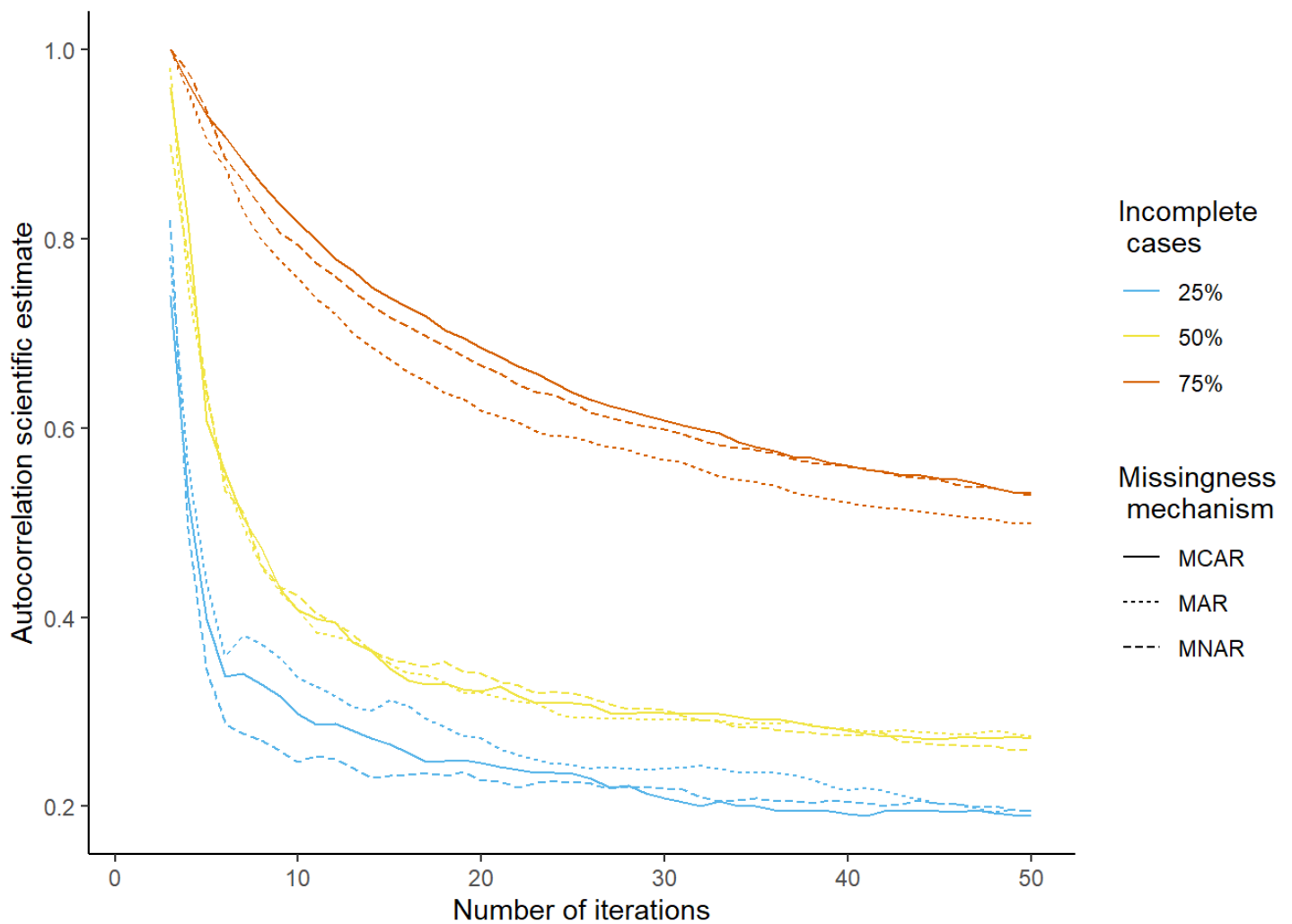
Diagnostic Methods



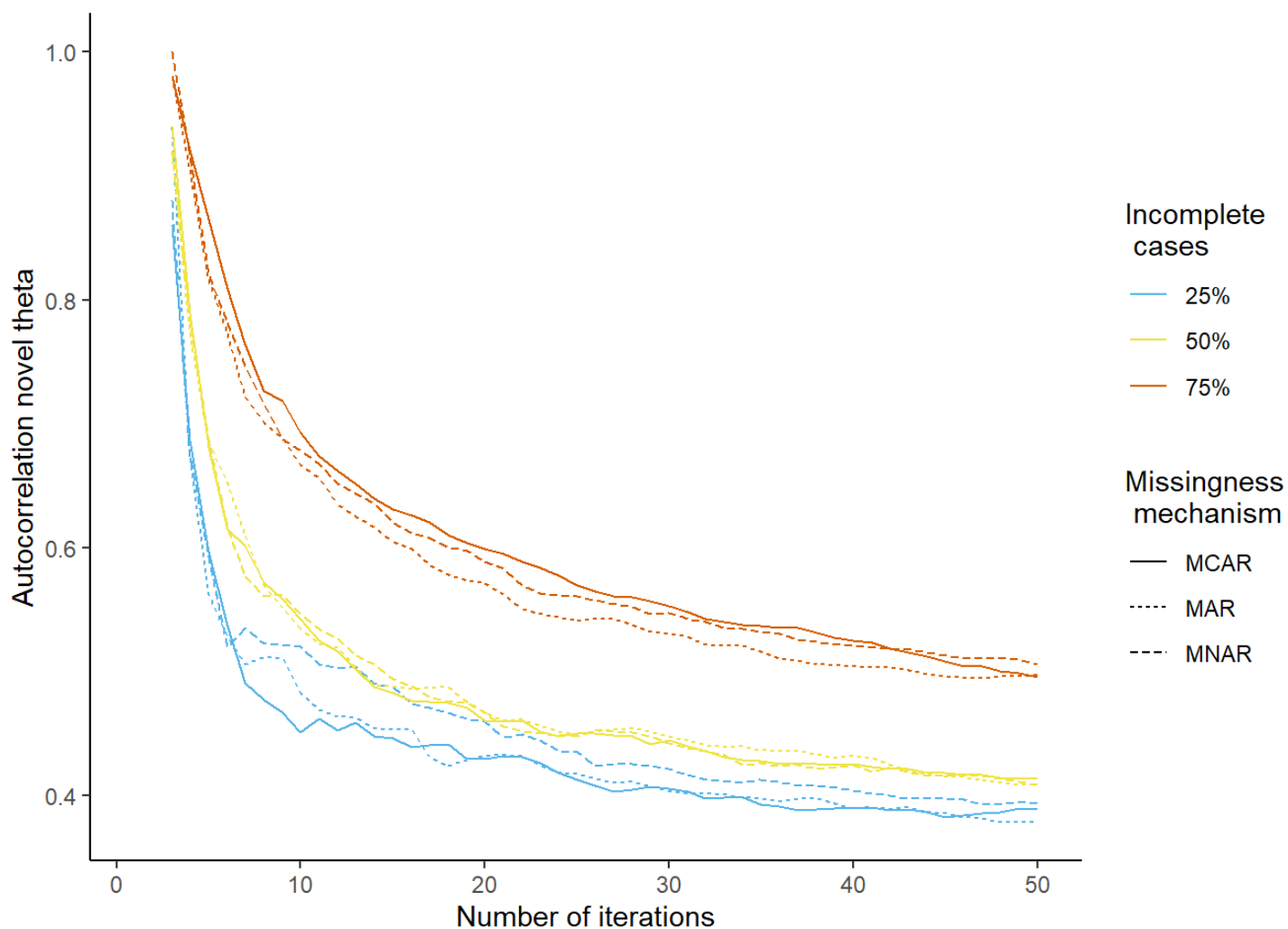
We observe that autocorrelation in the chain means rapidly decreases until $it \geq 6$. This means that there is some initial trending within chains, but stationarity of the average imputed value does not improve substantively after the first few iterations. These results hold irrespective of the missingness condition [necessary to add?].



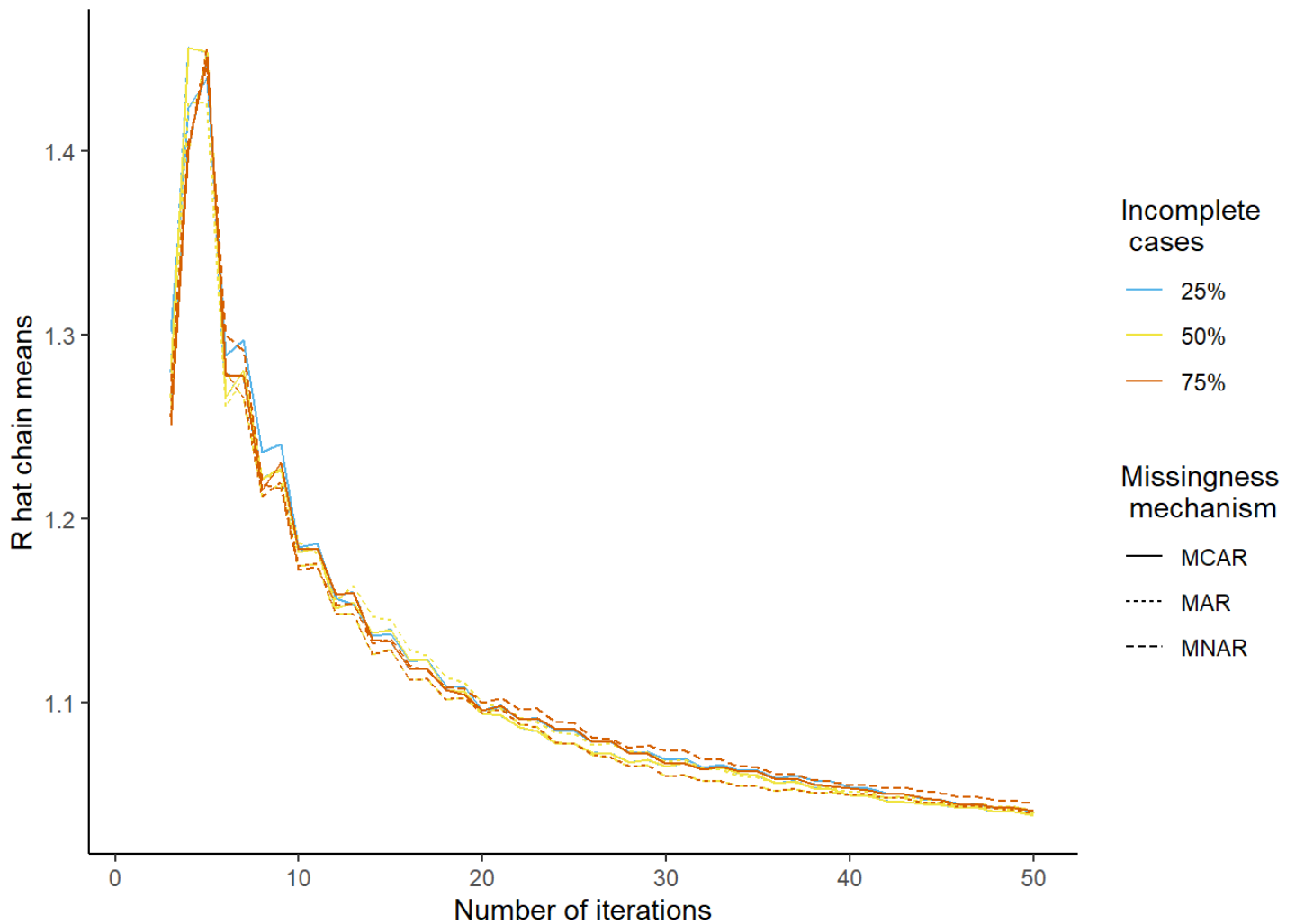
Autocorrelation in the chain variances show us something similar. The number of iterations that is required to reach non-improving autocorrelations is somewhat more ambiguous than for chain means, but generally around $it \geq 10$. We do not observe a systematic difference between missingness conditions here either.



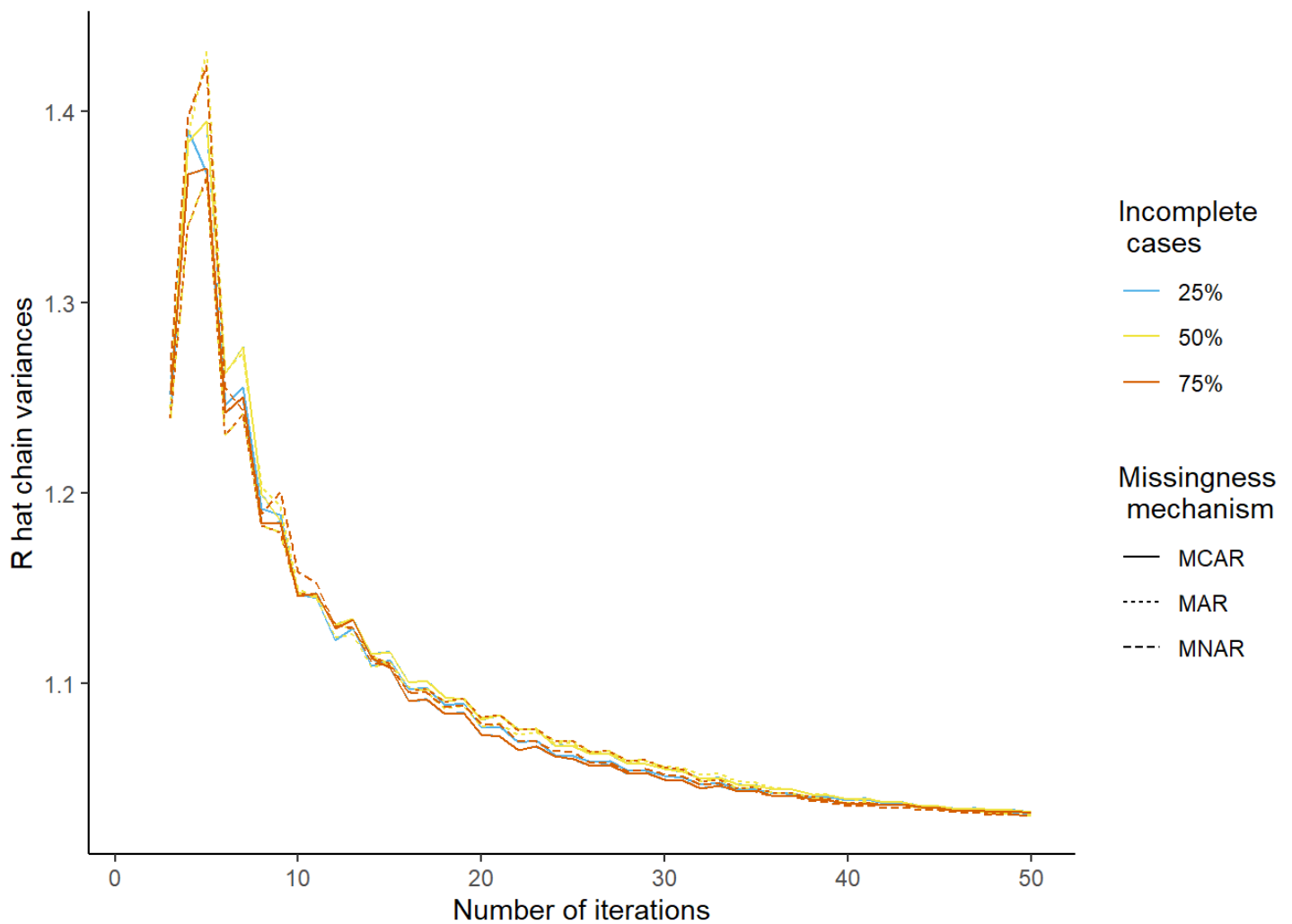
There is more autocorrelation in the scientific estimates than in the univariate parameters (chain means and chain variances). We observe the highest autocorrelations in conditions where 75% of cases are incomplete. Overall, the autocorrelations reach a plateau when $it \geq 20$ to 30. There is no clear effect of the missingness mechanisms [unnecessary?].



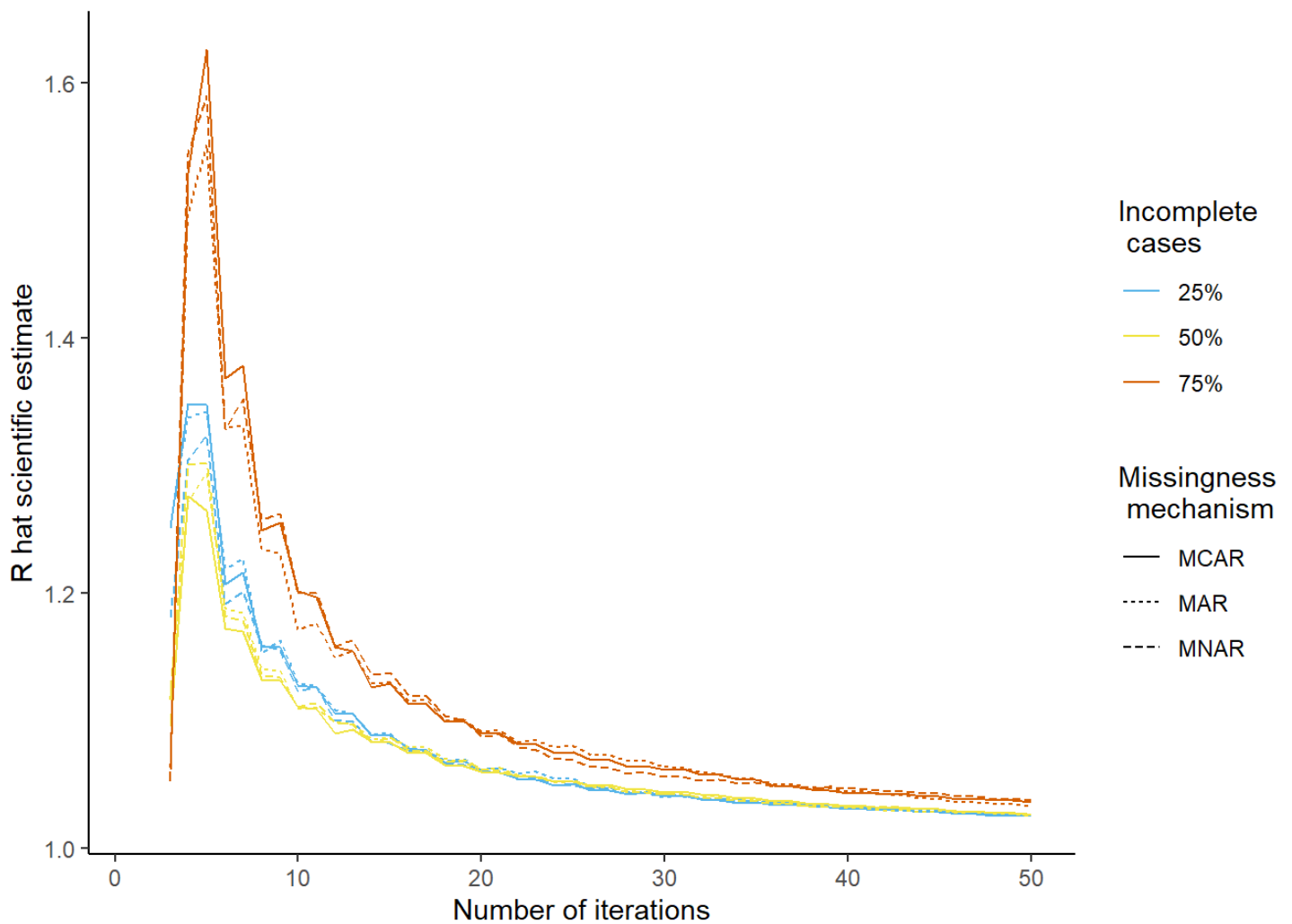
The autocorrelation in the novel parameter exhibits a similar trend to the autocorrelation in the scientific estimates. Trending in this parameter diminishes when $it \geq 20$.



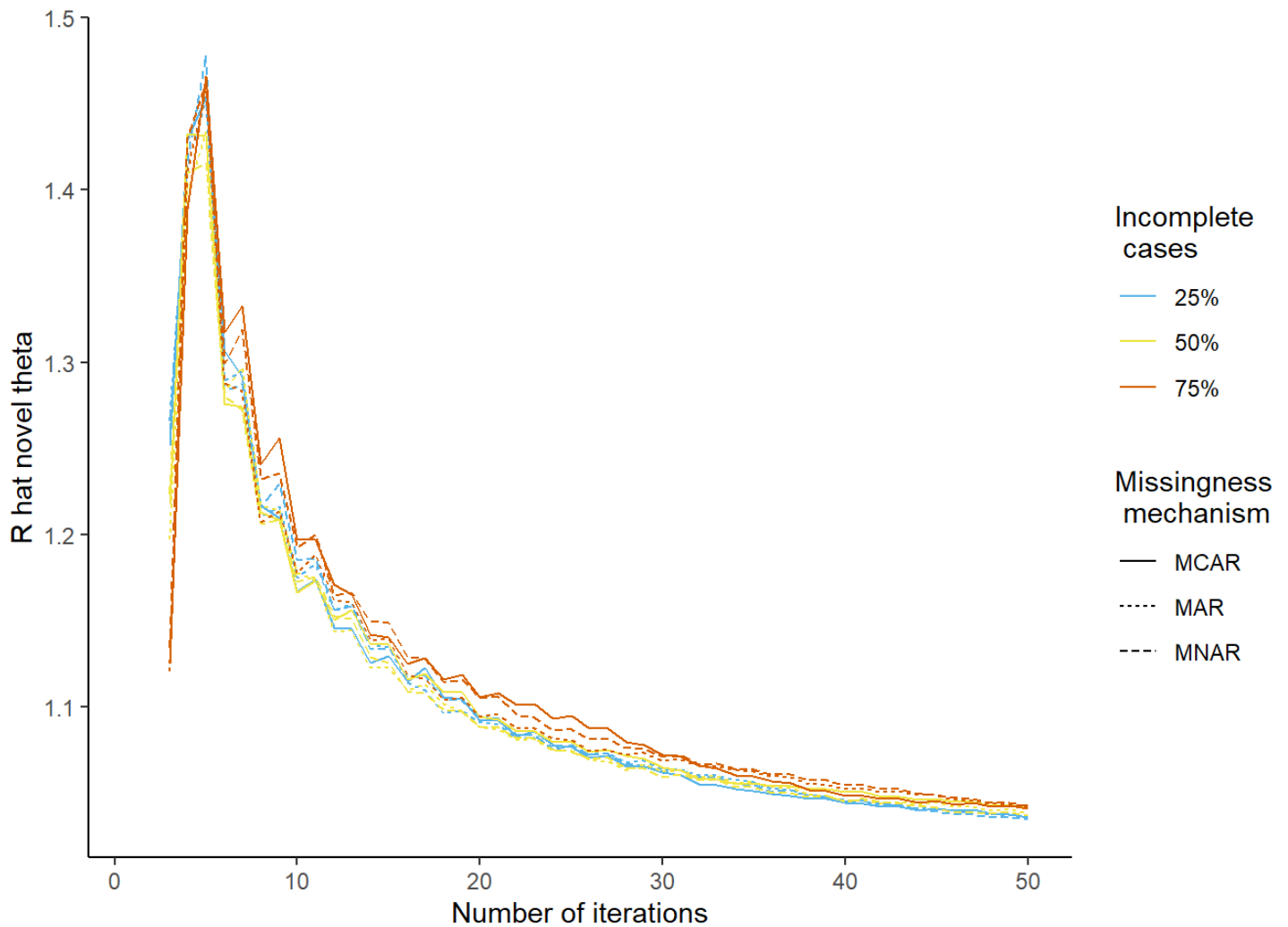
We observe that \widehat{R} -values of the chain means generally decreases as a function of the number of iterations. An exception to this observation is the initial increase when $3 \leq it \leq 5$ [interpret?? due to initialization or is there really more mixing initially??]. After the first couple of iterations, the mixing between chain means generally improves until $it \geq 30$ to 40. There is no apparent differentiation between the missingness conditions.



The mixing between chain variances mimics the mixing between chain means almost perfectly. Irrespective of the missingness condition, the \hat{R} -values taper off around $it \geq 30$.



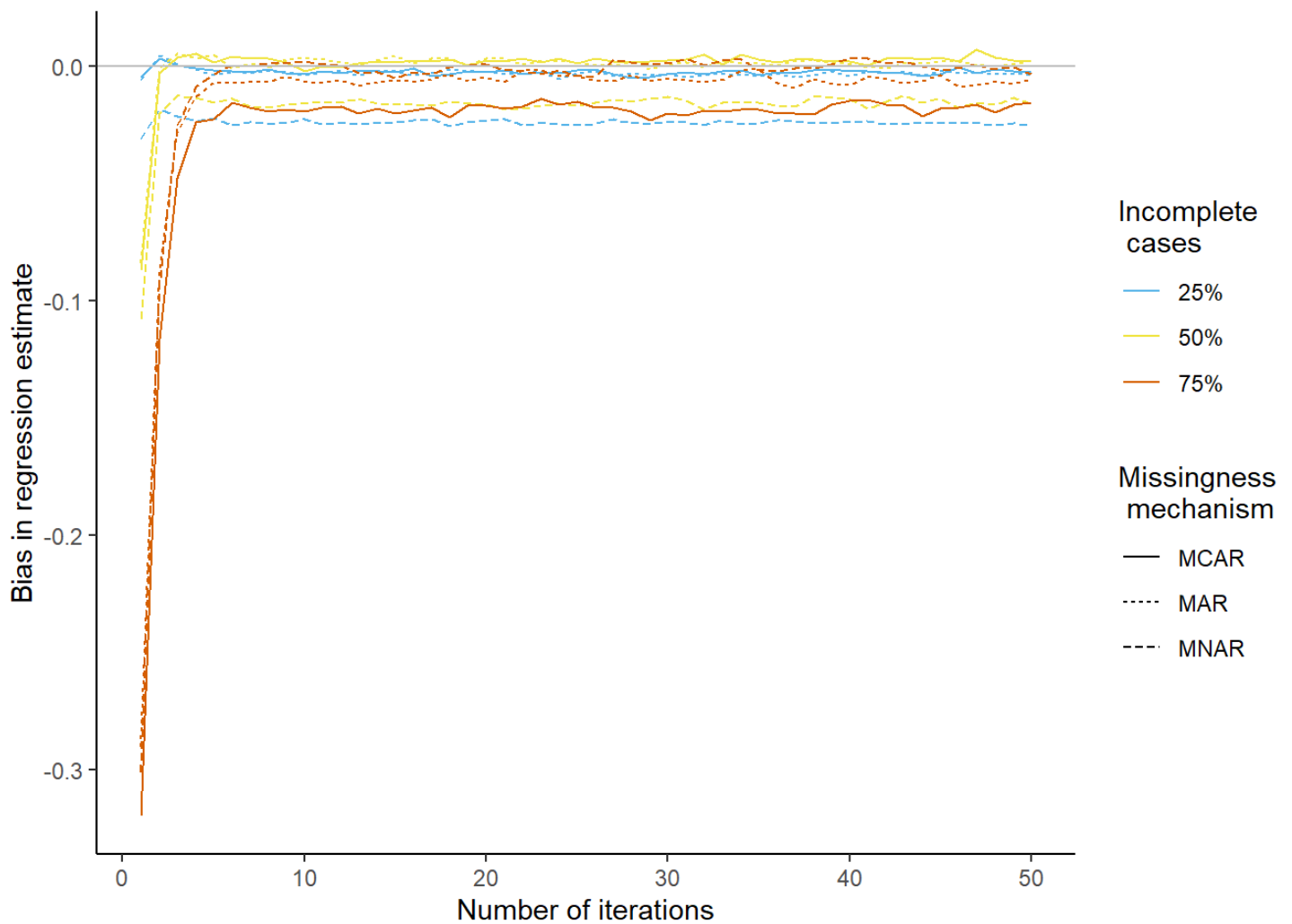
With the scientific estimate as parameter we observe very similar \widehat{R} -values again. We do, however, see some differences between missingness conditions. Conditions where 75% of the cases are incomplete show more extreme non-mixing. The overall trend remains the same: about 30 iterations are required before mixing stops improving substantially.



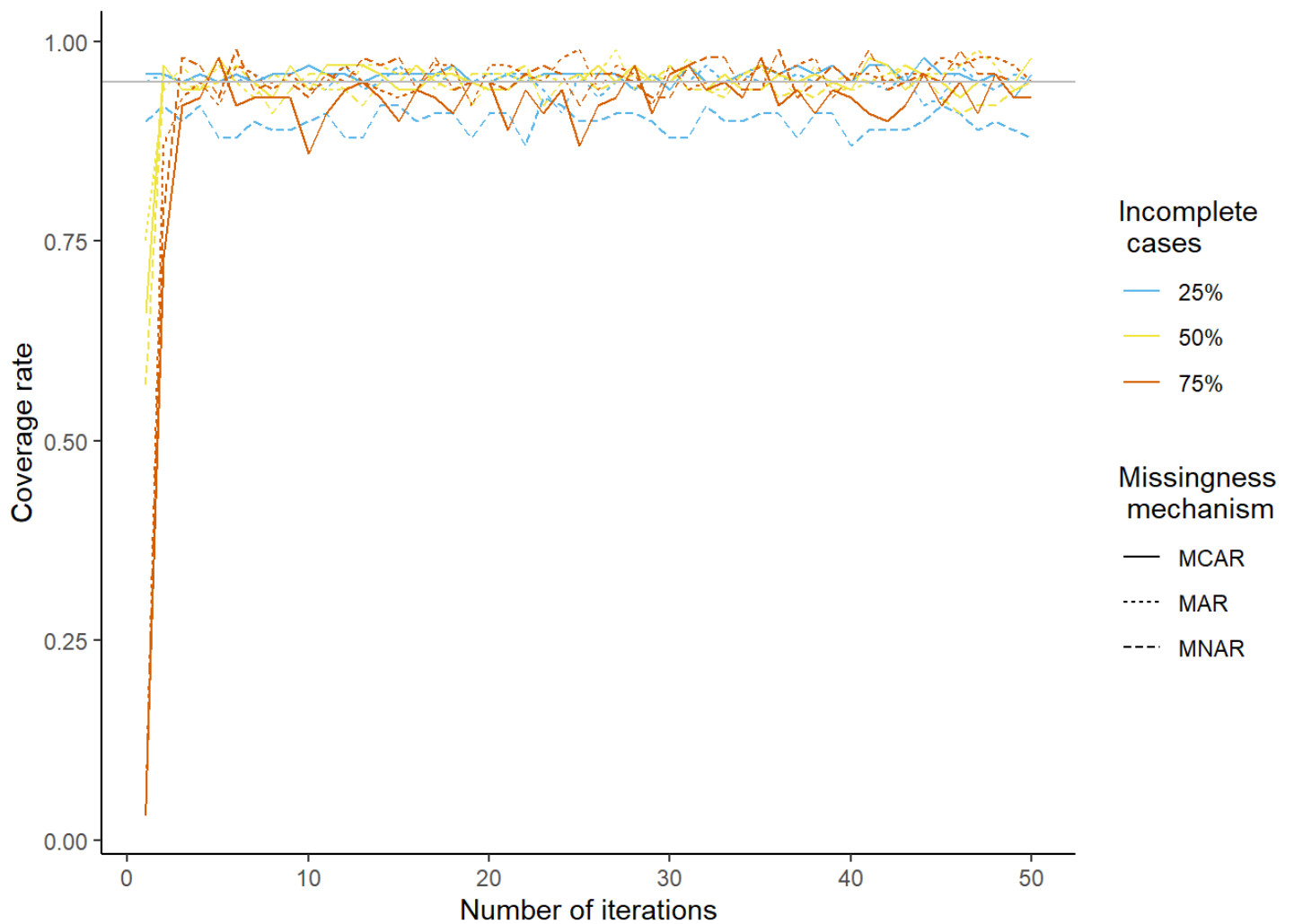
\hat{R} -values of the novel parameter show a trend similar to the chain means and chain variances. [add something about conditions??]

[These rhat plots all show some initialization before the fifth iteration: is rhat useful before that??]

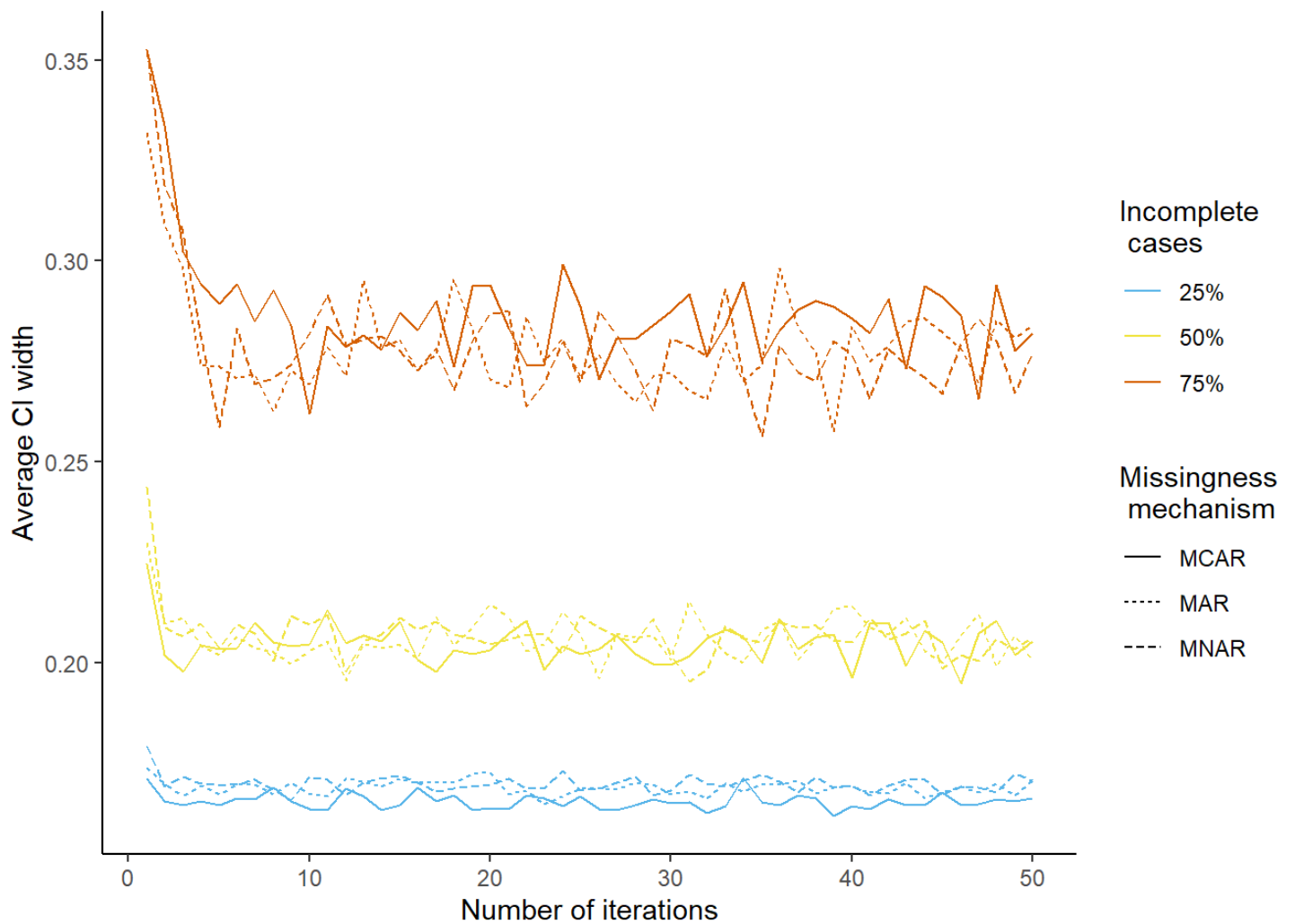
Performance Measures



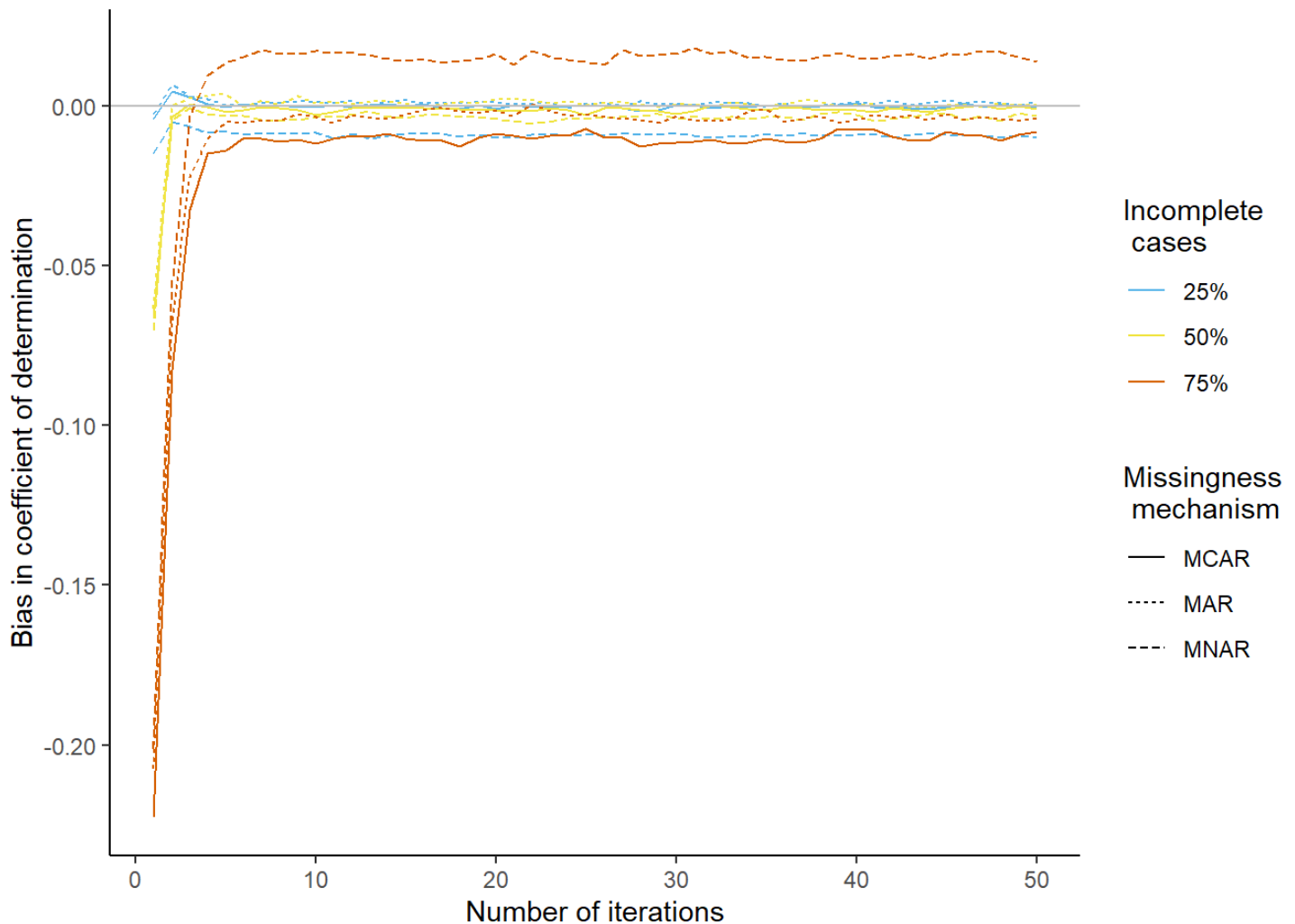
We see that within a few iterations the average bias approaches zero. When $it \geq 6$, even the worst-performing conditions (e.g., with a proportion of incomplete cases of 75%) produce stable, non-improving estimates [regression coefficient is underestimated because there is less info to estimate the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



Nominal coverage is quickly reached. 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. Depending on the proportion of incomplete cases this takes up-to $it \geq 9$.



Equivalent to the bias in the regression estimate, the bias in the coefficient of determination tapers off within a couple of iterations. We observe stable estimates in all conditions when $it \geq 6$ [interpret the over-estimation in MNAR+75% condition?].

Discussion

With this study, we show 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.

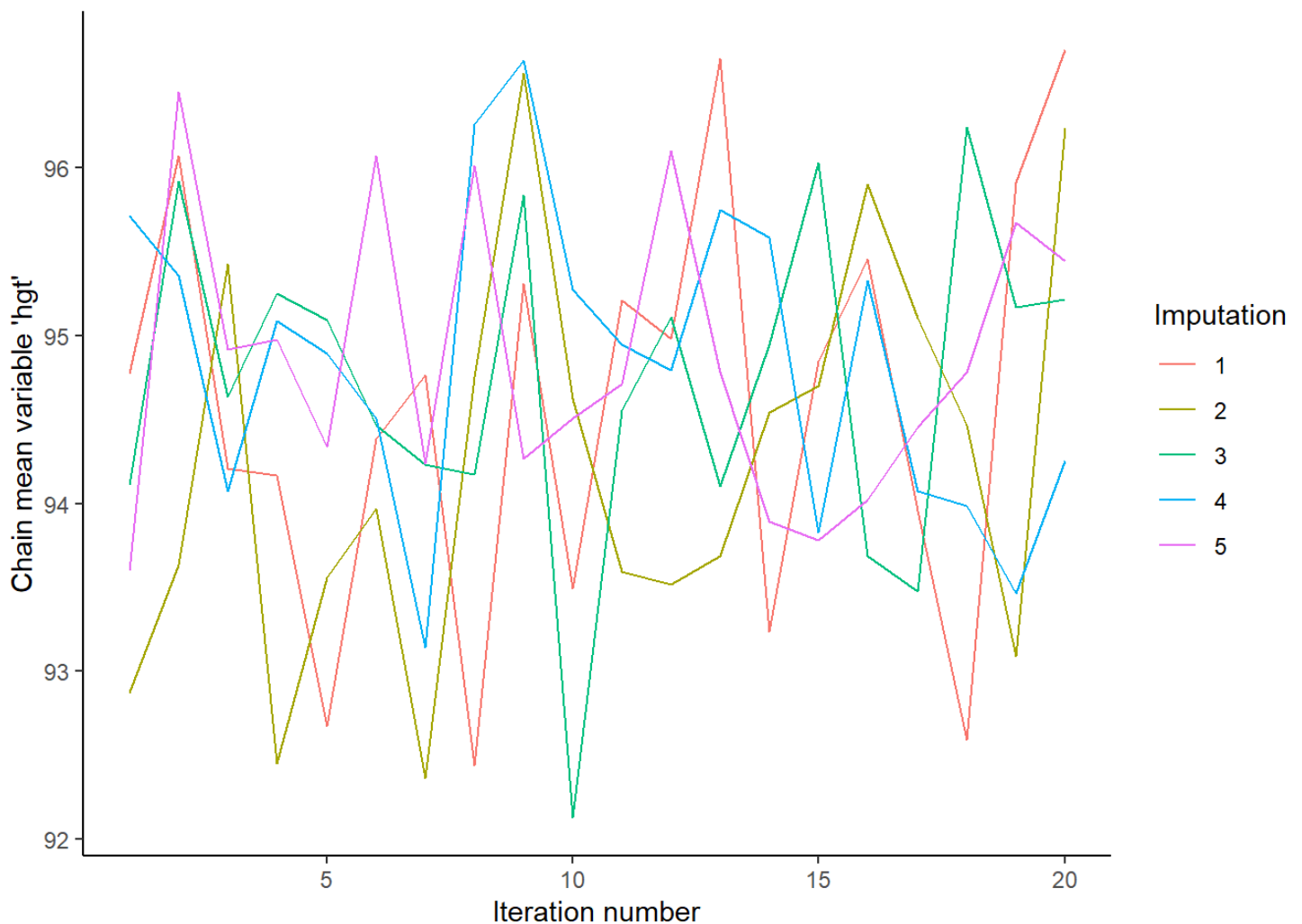
Our study found that—in the cases considered—inferential validity was achieved after five to ten iterations, much earlier than indicated by the non-convergence identifiers. Of course, it never hurts to iterate longer, but such calculations hardly bring added value.

- Convergence diagnostics keep improving substantially until $it = 20-30$
- Performance measures do not improve after $it = 9$
- [methodological explanation is that \hat{r} and \hat{ac} have a lag (few it to inform your statistic) → will always indicate convergence slower than inferential validity is reached]
- Univariate thetas may under-estimate non-convergence.
- Determining non-stationarity with λ is more difficult than with \hat{q} :

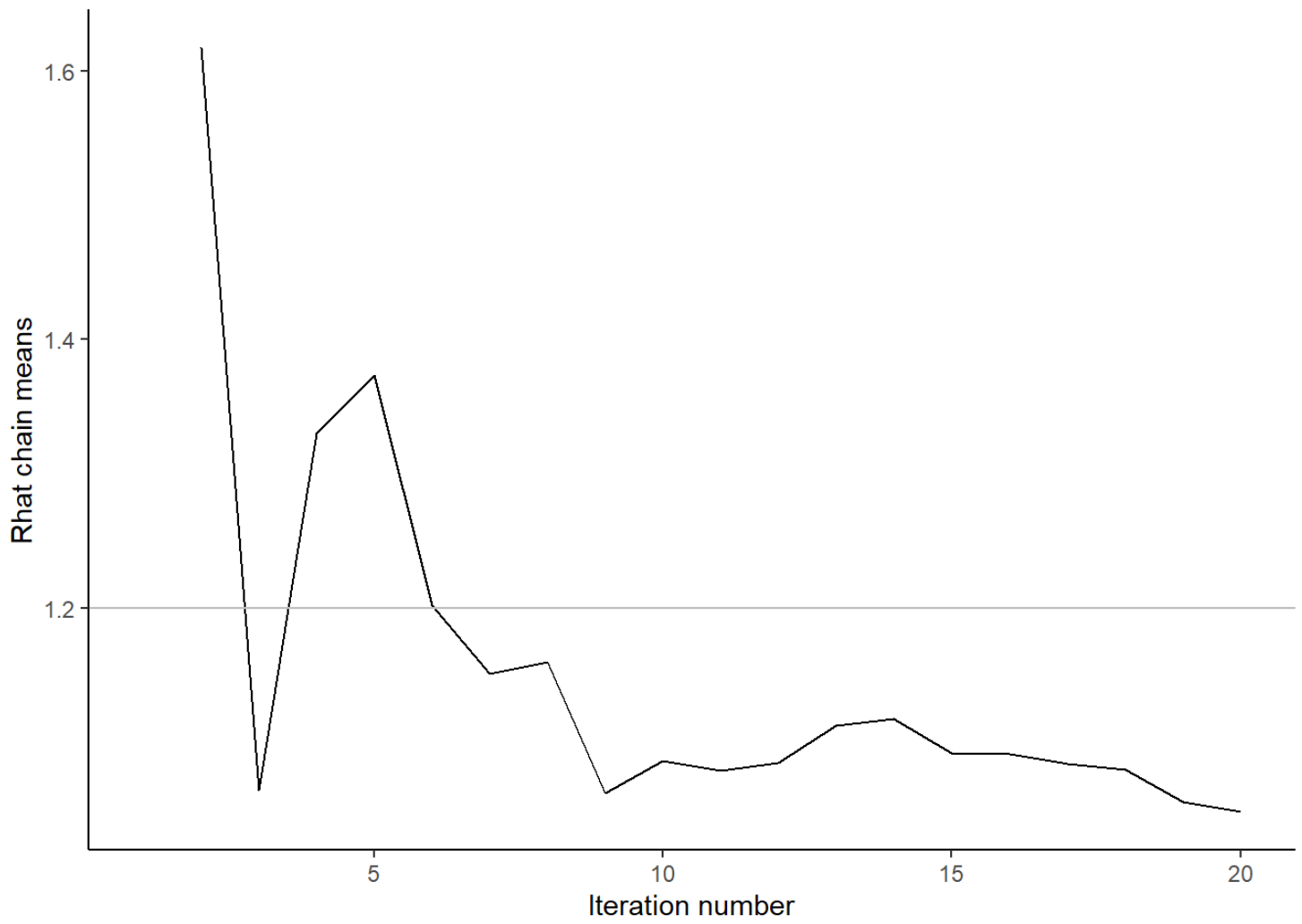
- Idea: calculate \hat{r}_{hat} for each block of 5 it

Case Study

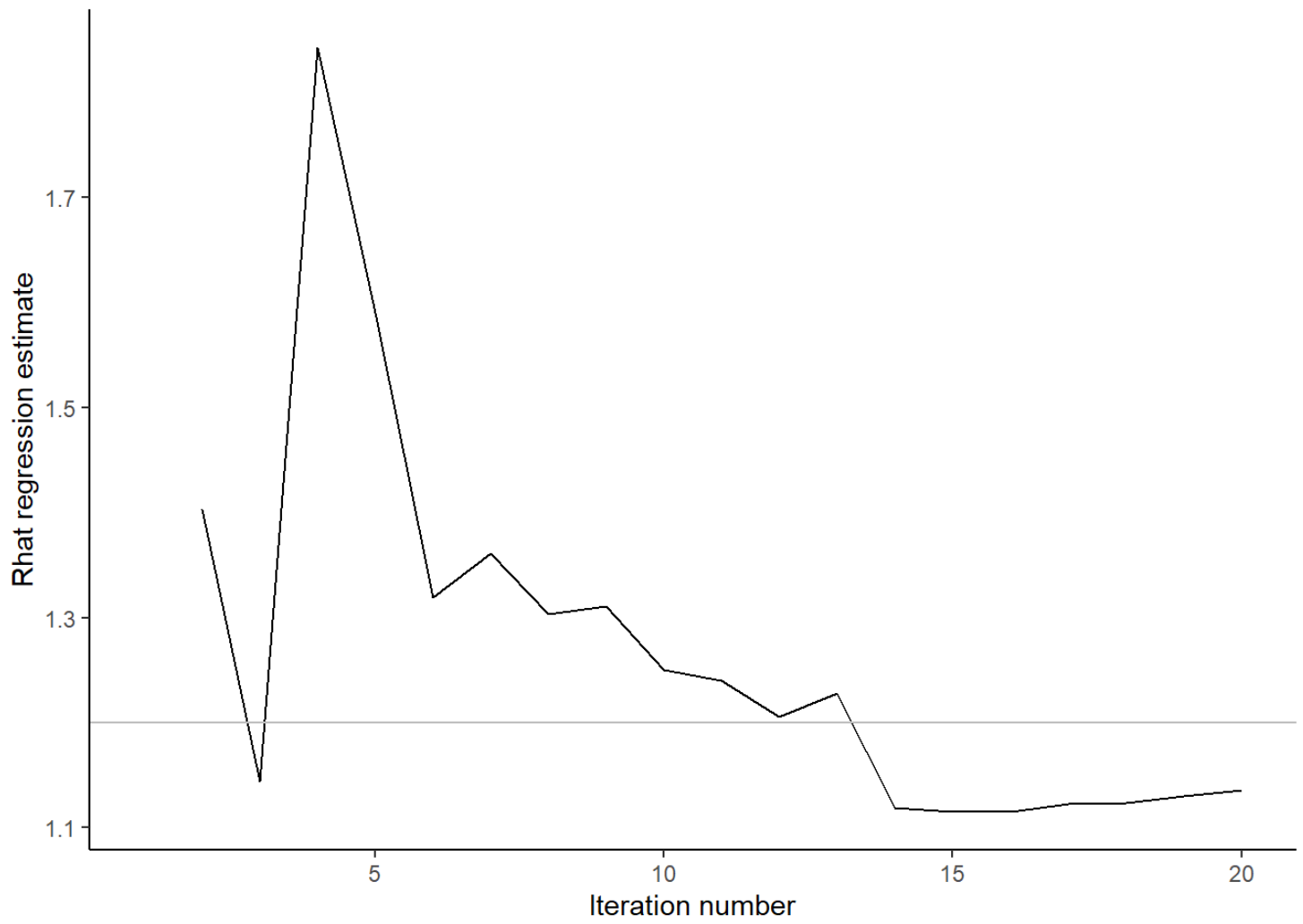
- We use real data: the `boys` dataset from the `mice` package
- We are interested in predicting age from the other variables, in particular in the regression coefficient of `hgt`
- We compare non-convergence identified using visual inspection versus \hat{r}_{hat} in the chain variances, scientific estimate and λ .
- The figures show results of a `mice` run with 20 iterations but otherwise default settings.



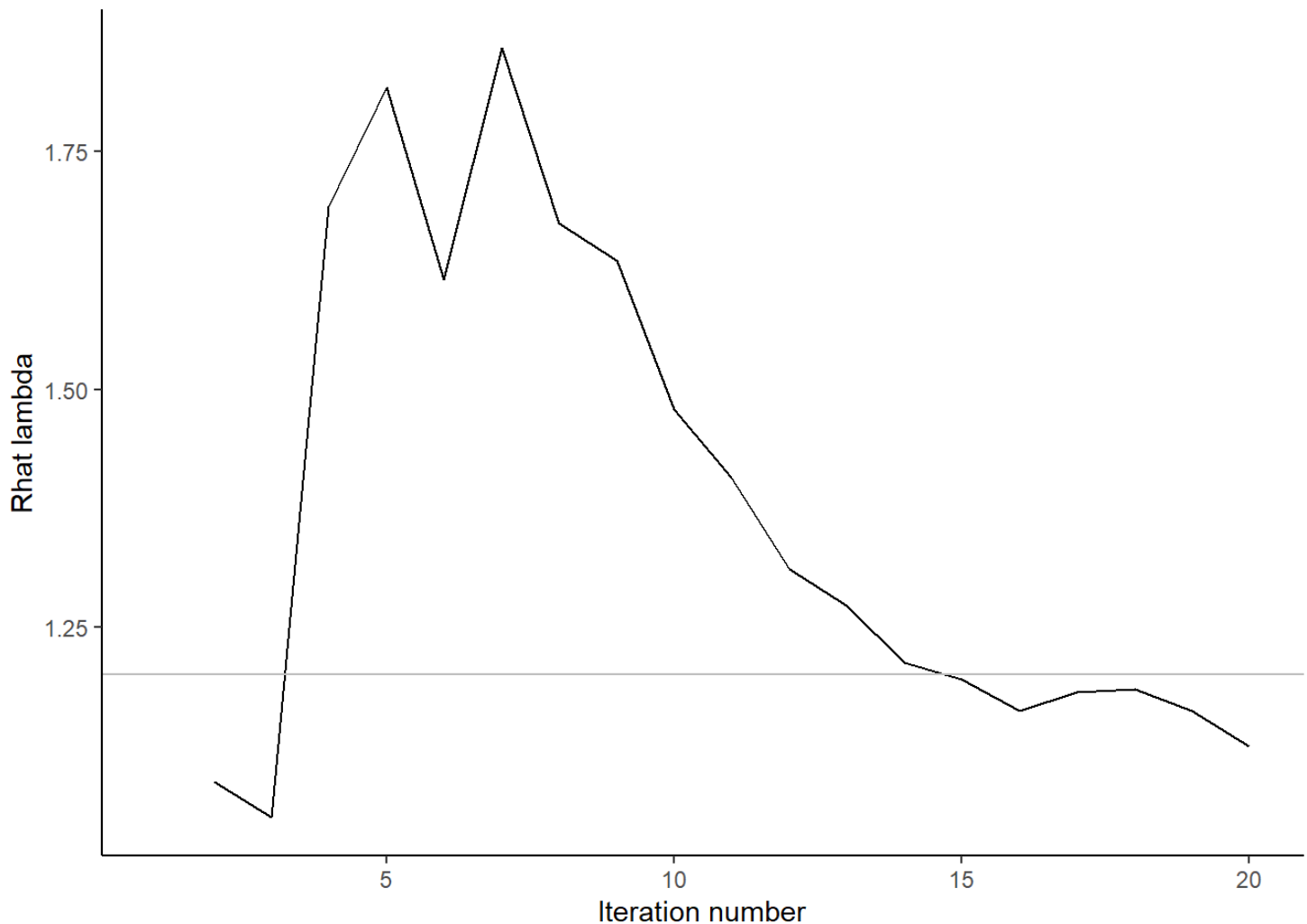
From the traceplot of the chain means it seems that mixing improves up-to 10 iterations, while trending is only apparent in the first three iterations.



This figure shows that 7 iterations are required before the \widehat{R} -values of the chain means drop below the threshold for non-convergence.



The scientific estimate reaches the threshold much sooner, when $it = 14$.



According to the \widehat{R} -values of the novel parameter, at least 15 iterations are required.

[Is this a nice example? Why is convergence so slow? Don't the defaults work? Is this the bmi+wgt+wgt problem? Maybe find an easier problem?]

References (incomplete)

- 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.
- 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> (<https://doi.org/10.1007/978-0-387-92407-6>).
- Liu, J., A. Gelman, J. Hill, Y.-S. Su, and J. Kropko. 2014. "On the Stationary Distribution of Iterative Imputations." *Biometrika* 101 (1): 155–73. <https://doi.org/10.1093/biomet/ast044> (<https://doi.org/10.1093/biomet/ast044>).
- R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria.
- Rubin, Donald B. 1987. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Mathematical Statistics Applied Probability and Statistics. New York, NY: Wiley.
- 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 (July): 37. <https://doi.org/10.5334/dsj-2017-037> (<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> (<https://doi.org/10.18637/jss.v045.i03>).

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> (<https://doi.org/10.1080/01621459.2014.948117>).