Missing The Point

Hanne Oberman

7-9-2020

Missing The Point: Non-Convergence in Iterative Imputation Algorithms

Abstract

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. Our study provides insight into identifying non-convergence in iterative imputation algorithms, since there is no consensus on how to evaluate the convergence properties currently. 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

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.

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 (Zhu & Raghunathan, 2015; Takahashi, 2017), the current practice is to visually inspect imputations for signs of non-convergence.

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. Our aim is 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.
- What does non-convergence look like? –> non-stat + non-mix in a certain parameter, e.g., chain means.
- What are the consequences? –> bias, under-coverage –> Refer to van Buuren (2018) instead of reproducing this
- Existing non-convergence identifiers for MCMC algorithms are not all applicable, e.g. N_eff is estimated using varcov matrix of the set of samples, while this is not the case in it im per se.
- We consider autocorr and rhat, and monitor four parameters: chain means, chain variances, the scientific estimand, and lambda.

Simulation

We investigate non-convergence in iterative imputation through model-based simulation in R (version 4.0.2; 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.

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

Our aim is to 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 ($n_{it} = 1, 2, ..., 50$). 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 sample of $n_{obs} = 1000$ cases, before inducing missingness. Let the complete set be three multivariately normal random variables

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

The complete set is amputed according to the 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 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 we propose, λ.

Performance measures

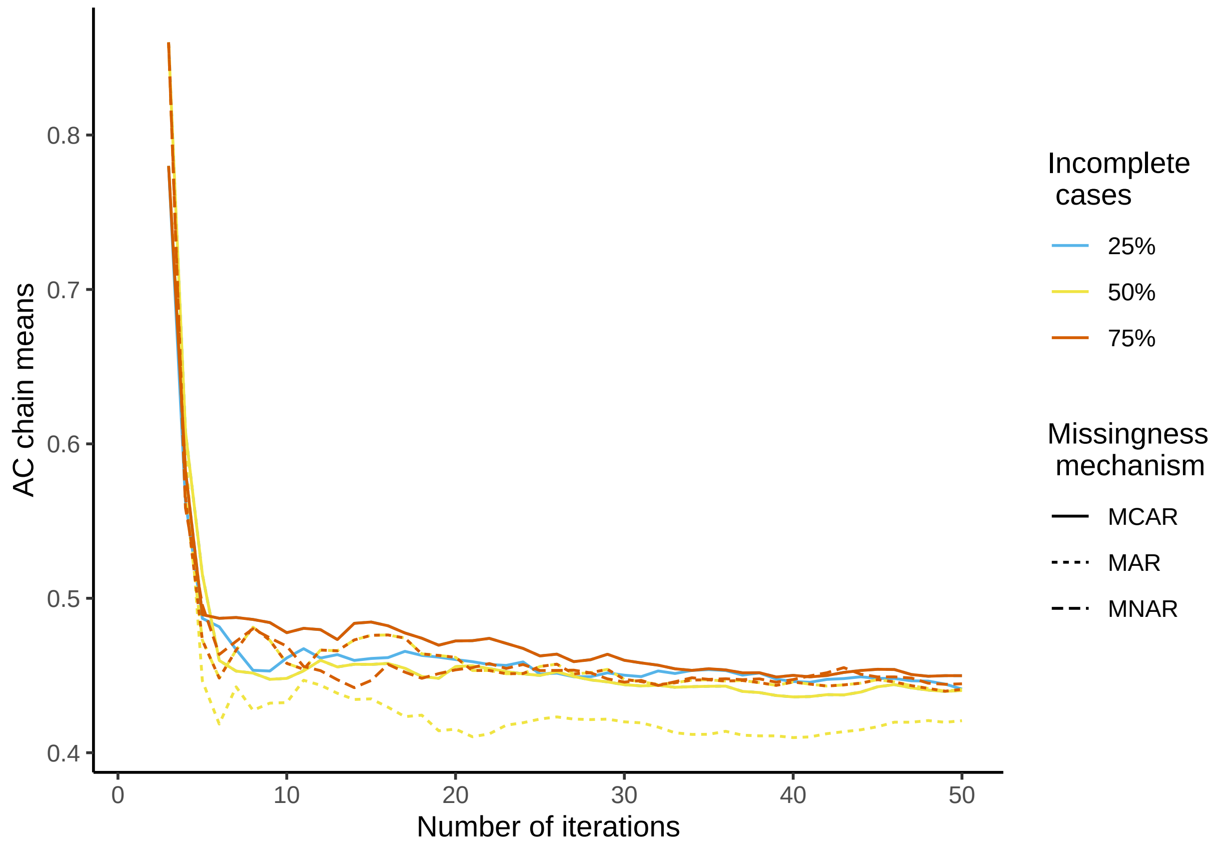
As recommended by Van Buuren (2018), our performance measures are bias ($E(\bar{Q}) - Q$), average confidence interval width ($\bar{Q}_{LL} - \bar{Q}_{UL}$, where LL and UL denote the lower and upper limit of the 95% confidence interval respectively) and empirical coverage rate ($Pr(\bar{Q}_{LL} \geq Q \geq \bar{Q}_{UL})$) of the estimands.

Results

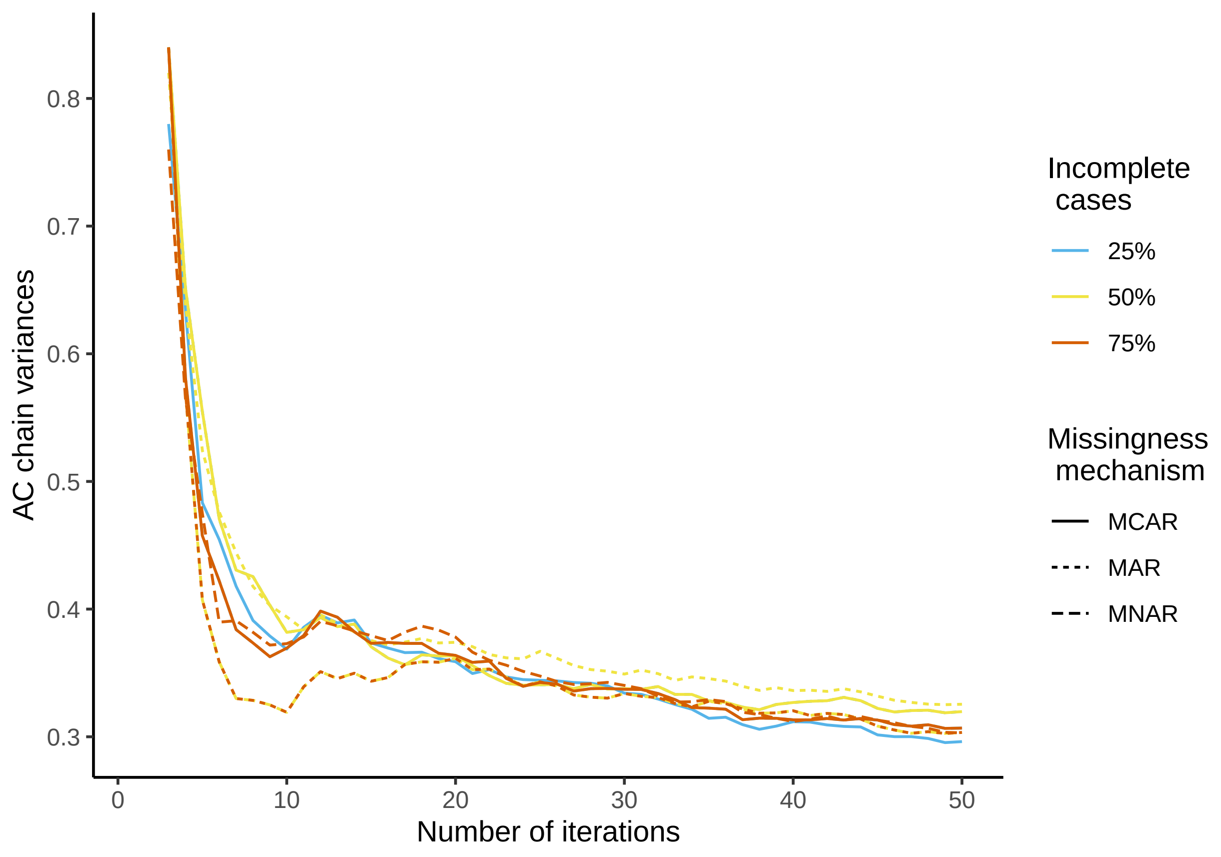
We present the results of our simulation study visually because of the many simulation conditions. Within each plot, the results are split according to the missingness conditions (missingness mechanisms as line types, and proportion of incomplete cases as colors).

Non-Convergence Identifiers

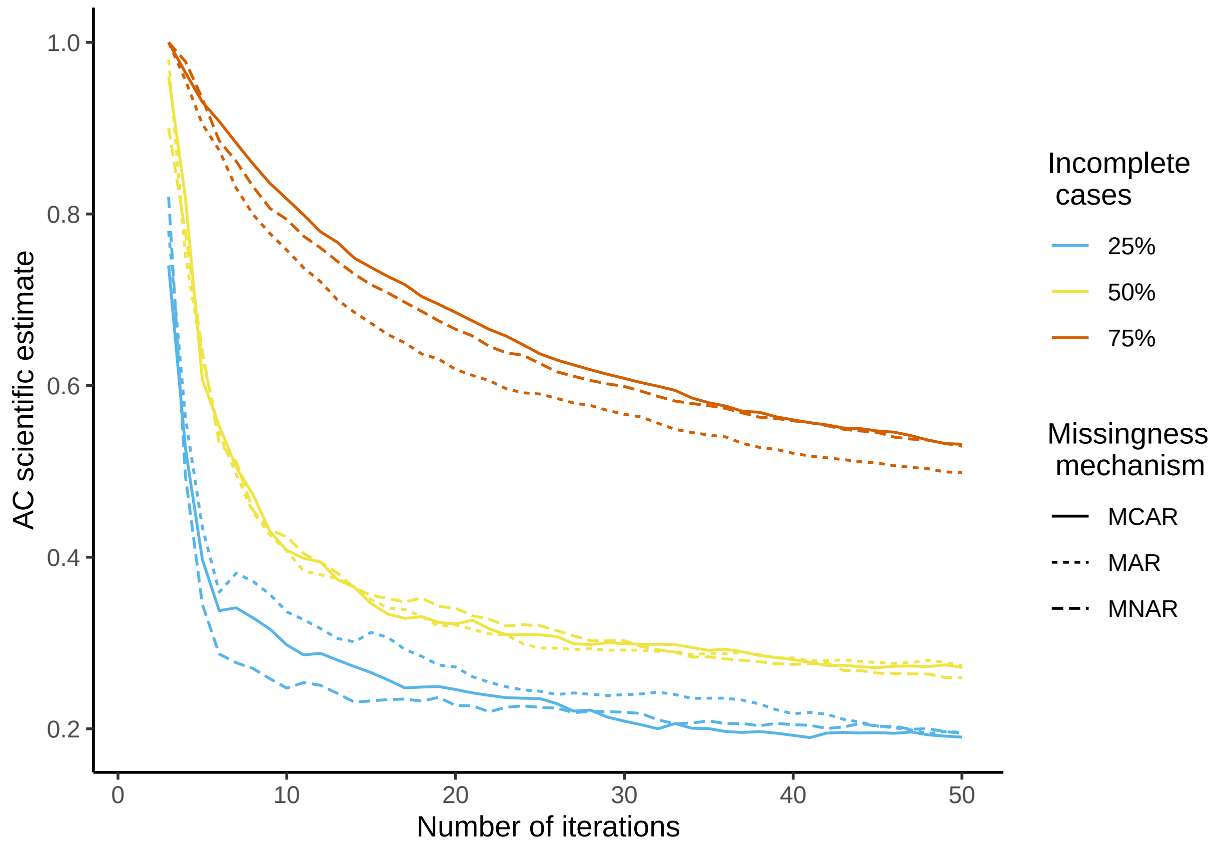
The next four plots show the autocorrelations in each parameter plotted against the number of iterations.



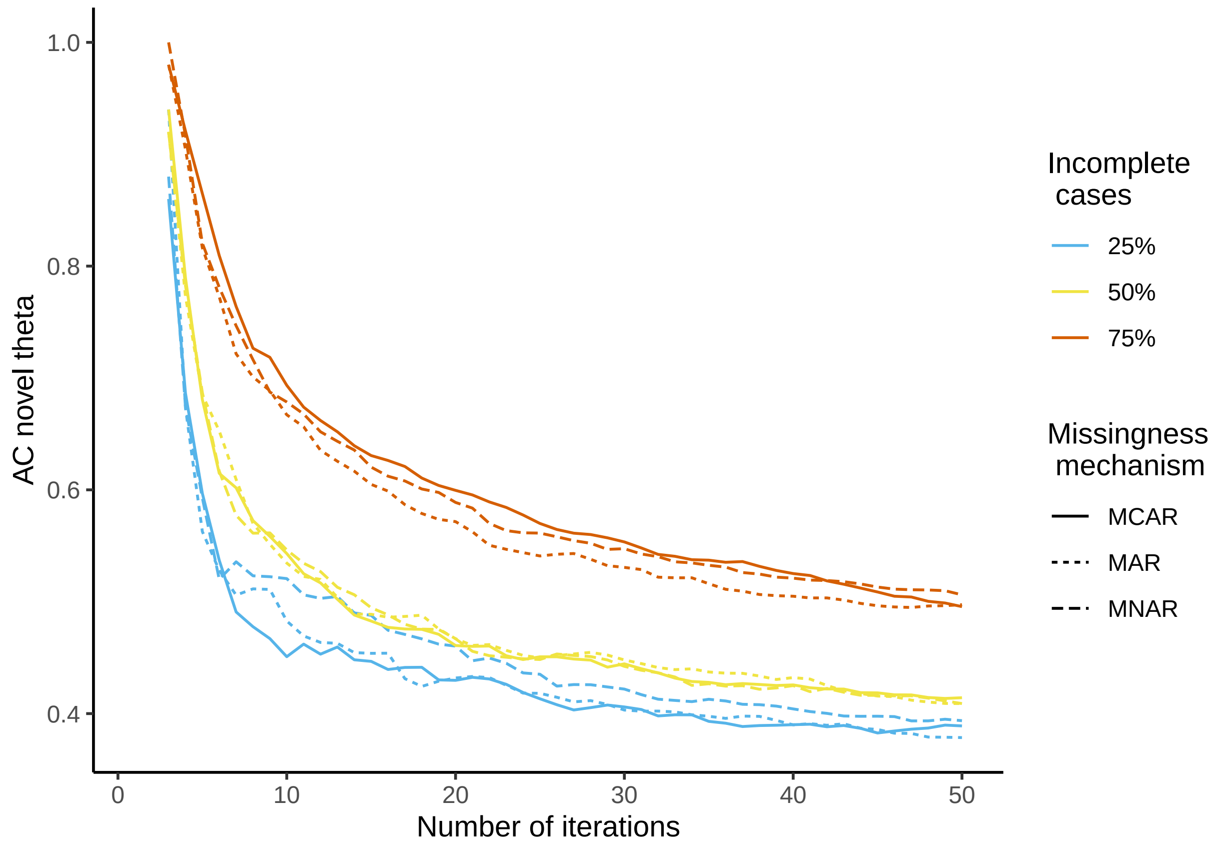
This plot shows that irrespective of the missingness condition, autocorrelation in the chain means decreases rapidly until $it \geq 6$. This means that there is some initial trending within chains, but the stationarity does not improve substantively after the first few iterations.



Autocorrelations in the chain variances do not show a systematic effect of the missingness conditions either. The number of iterations that is required to reach non-improving autocorrelations is somewhat more ambiguous, but roughly when $it \geq 10$.

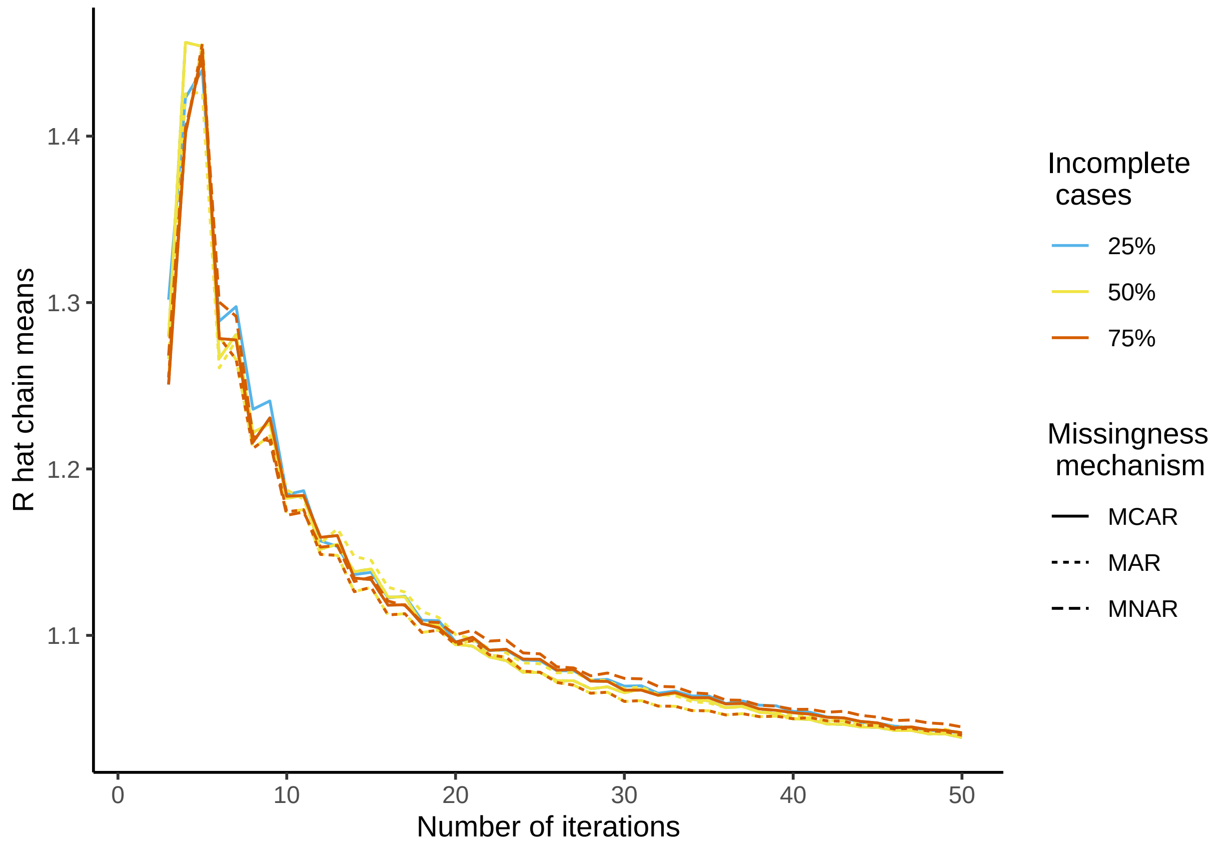


We see that the autocorrelations in the scientific estimates are more pronounced than in the univariate parameters (chain means and chain variances), especially when the proportion of missing cases is moderate to extreme. We observe the worst stationarity in conditions with 75% of cases having missing values. There is no clear effect of the missingness mechanisms. Overall, the autocorrelations reach a plateau when $it \geq 20$ to 30.

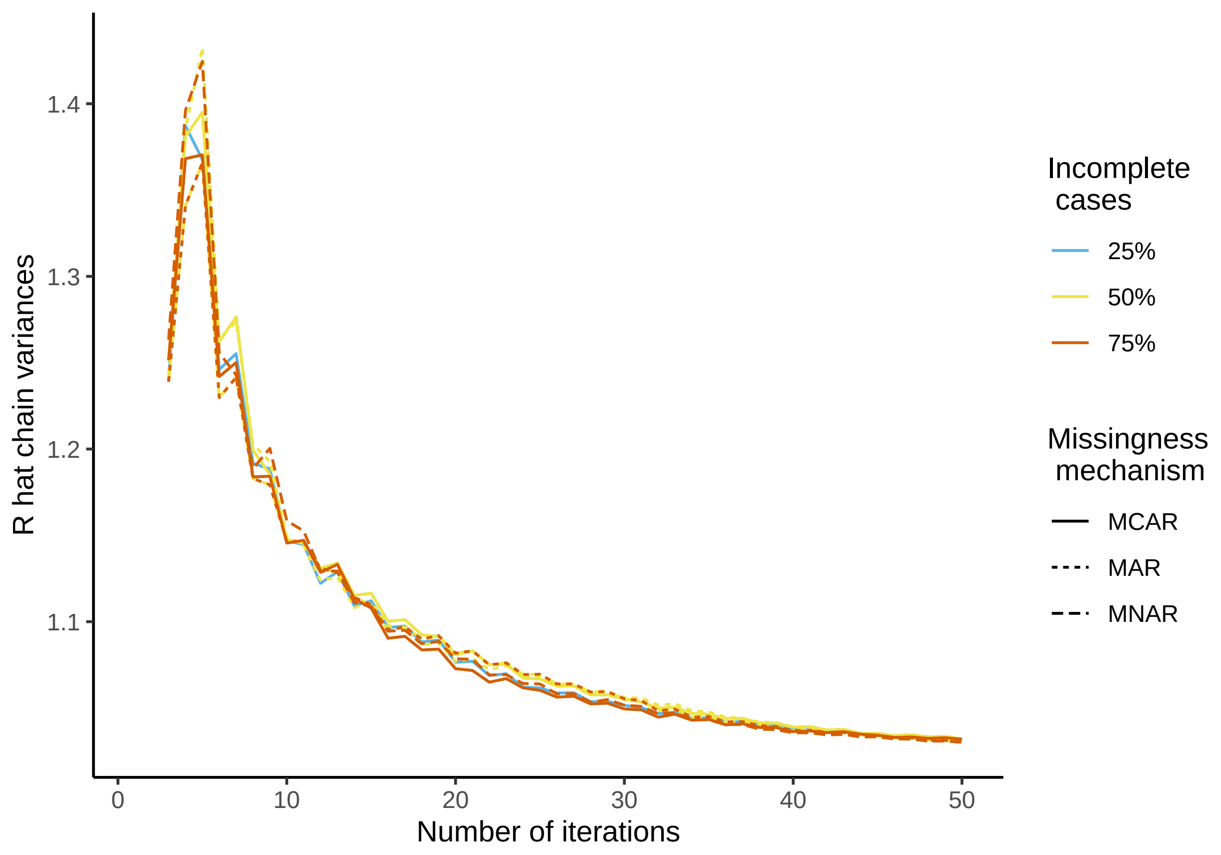


Autocorrelations in the novel parameter show a similar trend to the autocorrelations in the scientific estimates. Around $it \geq 20$ is required before trending in this parameter diminishes.

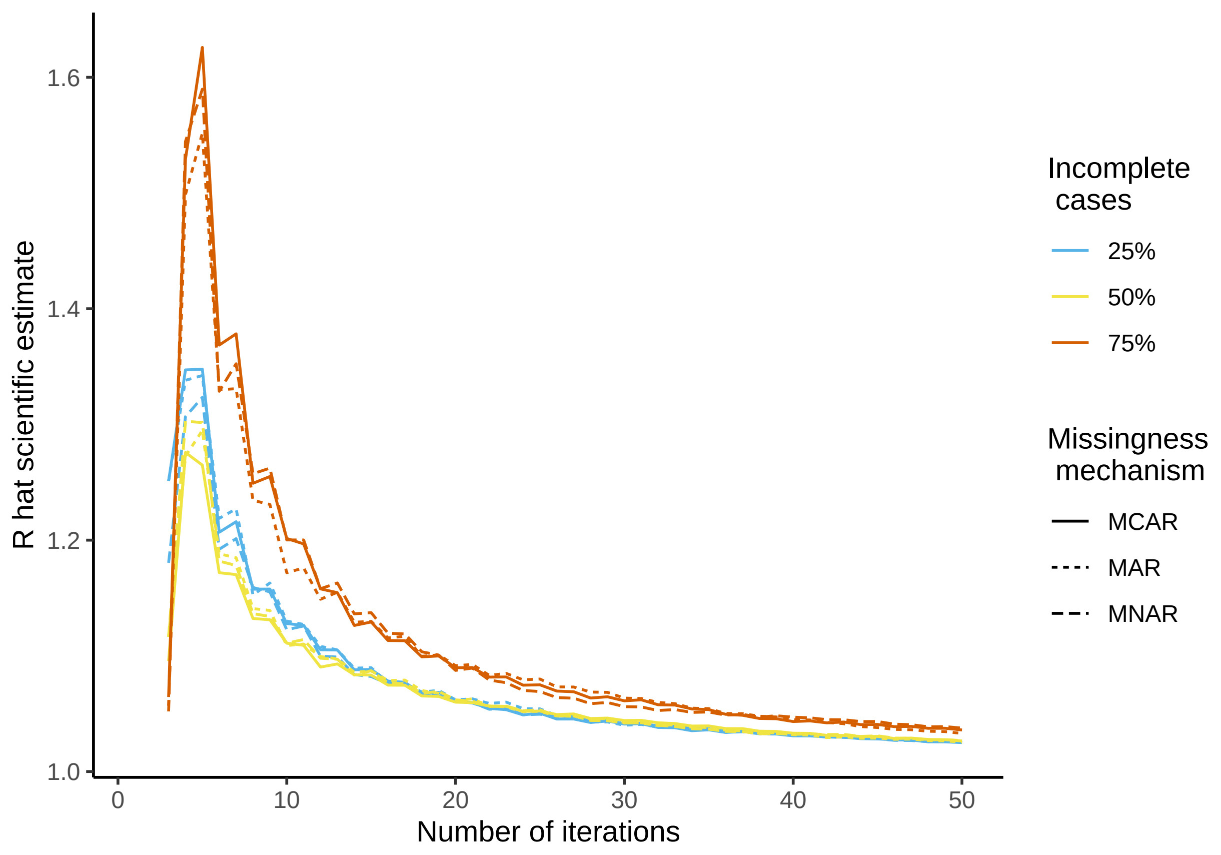
The next four plots show the $\widehat{R}$ in the different parameters.



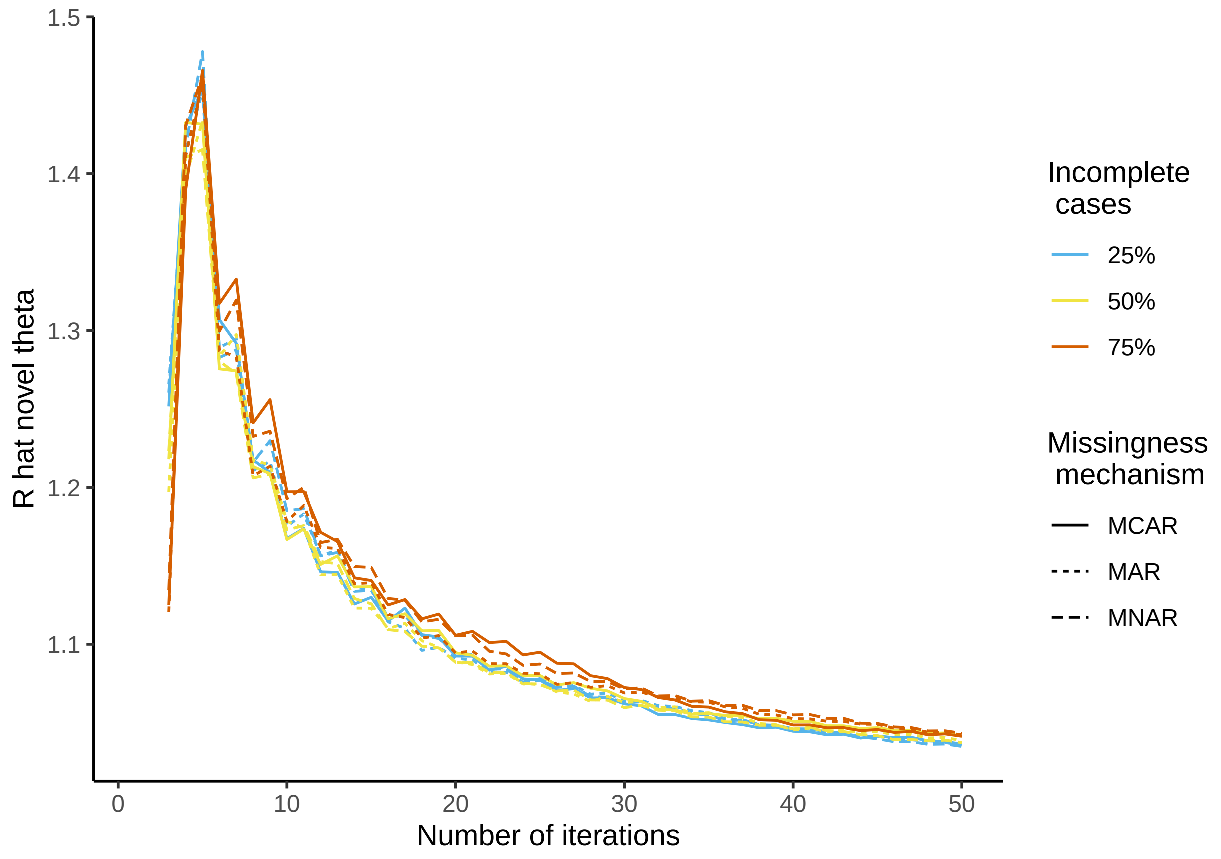
In this plot we observe that $\widehat{R}$ in the chain means generally decreases as a function of the number of iterations—with the exception of an initial increase when $3 < it < 5$. There is no apparent differentiation between the missingness conditions. The mixing between chain means generally improves until $it \geq 30$ to 40.



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



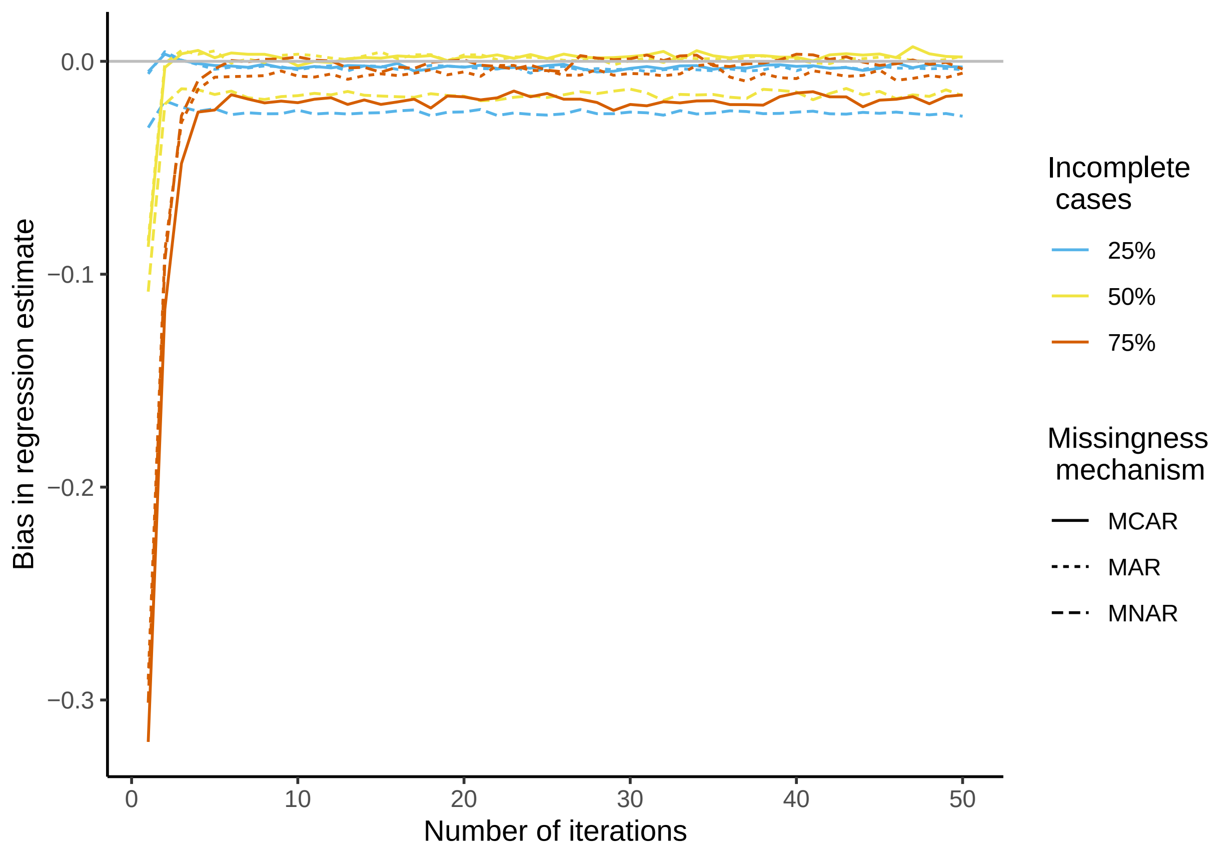
With the scientific estimate as parameter we observe very similar $\widehat{R}$-values again. We do, however, see some differentiation between missingness conditions. Conditions with 75% of cases being incomplete show more extreme non-mixing. The overall trend remains the same: about 30 iterations are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



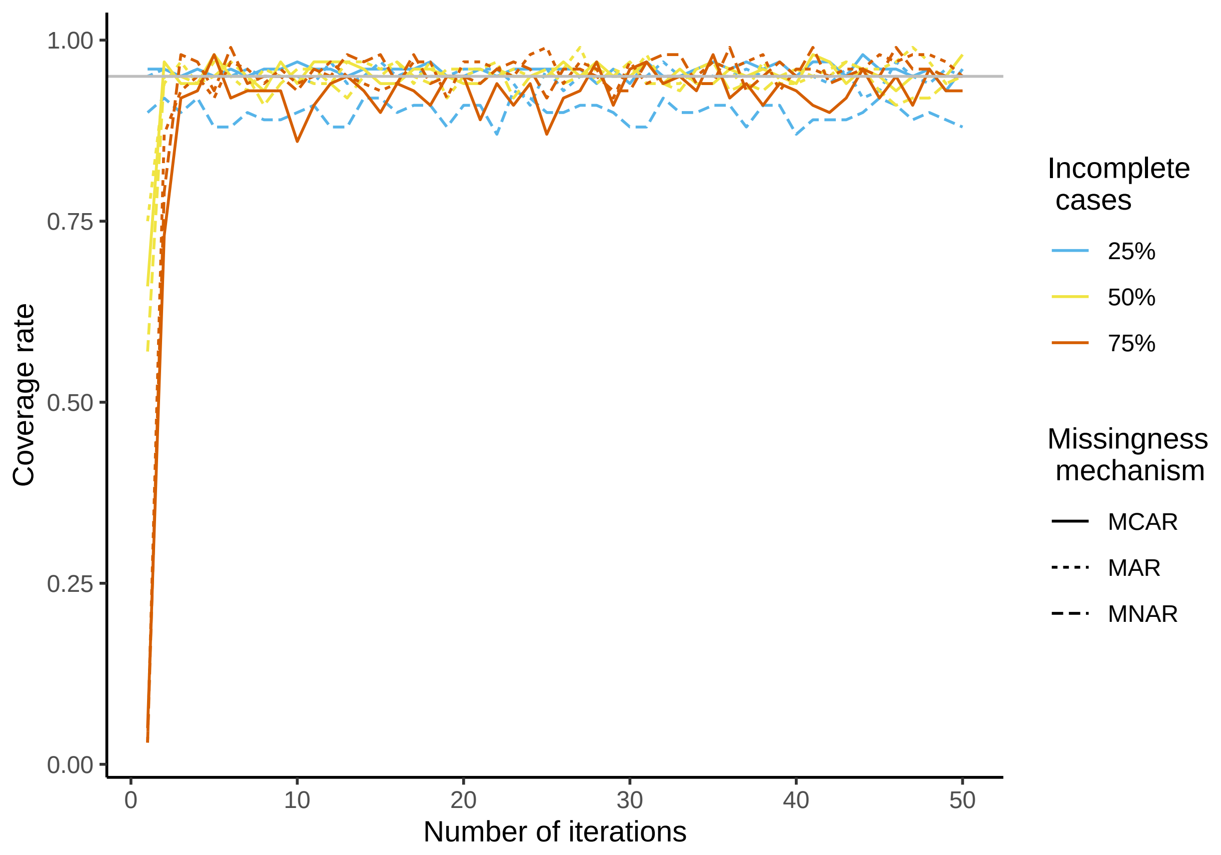
$\widehat{R}$-values in the chains of the novel parameter show a trend similar to the chain means and chain variances. [add something about conditions??]

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

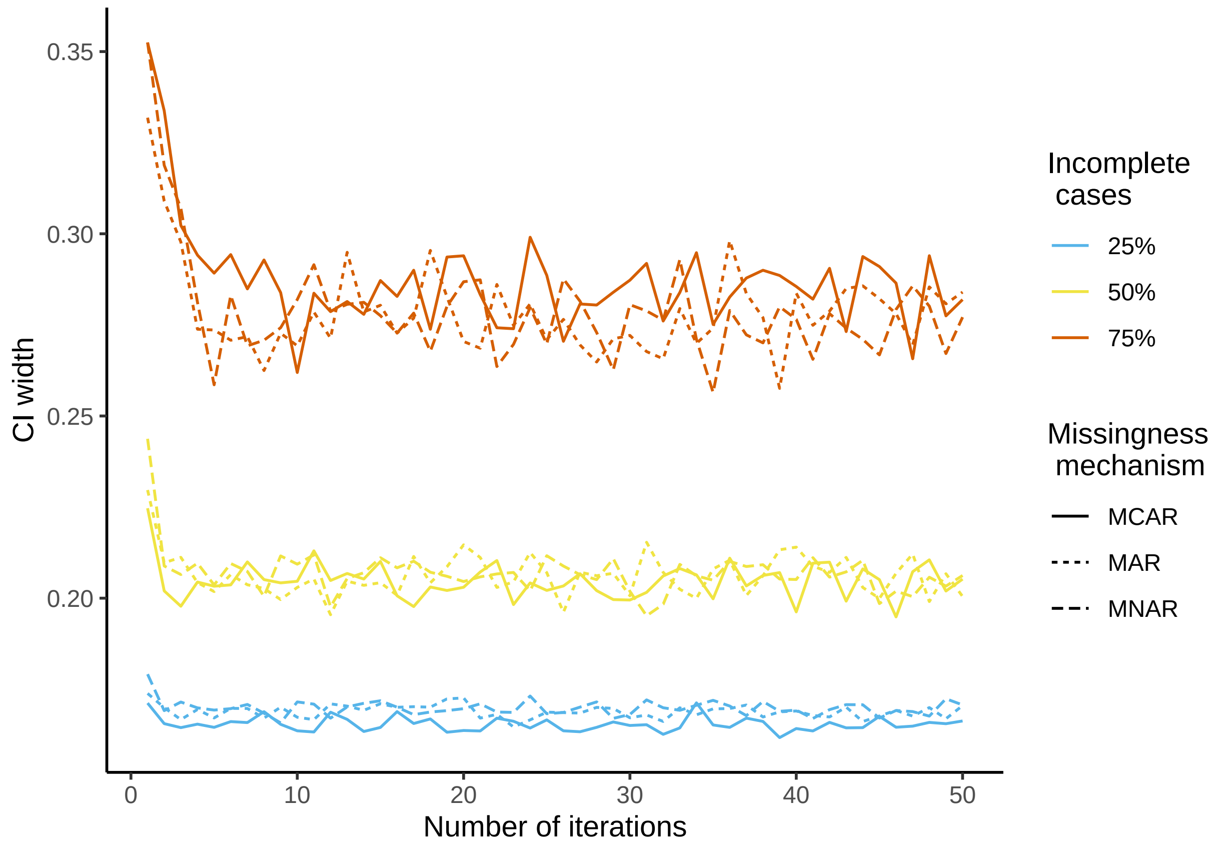
Performance Measures



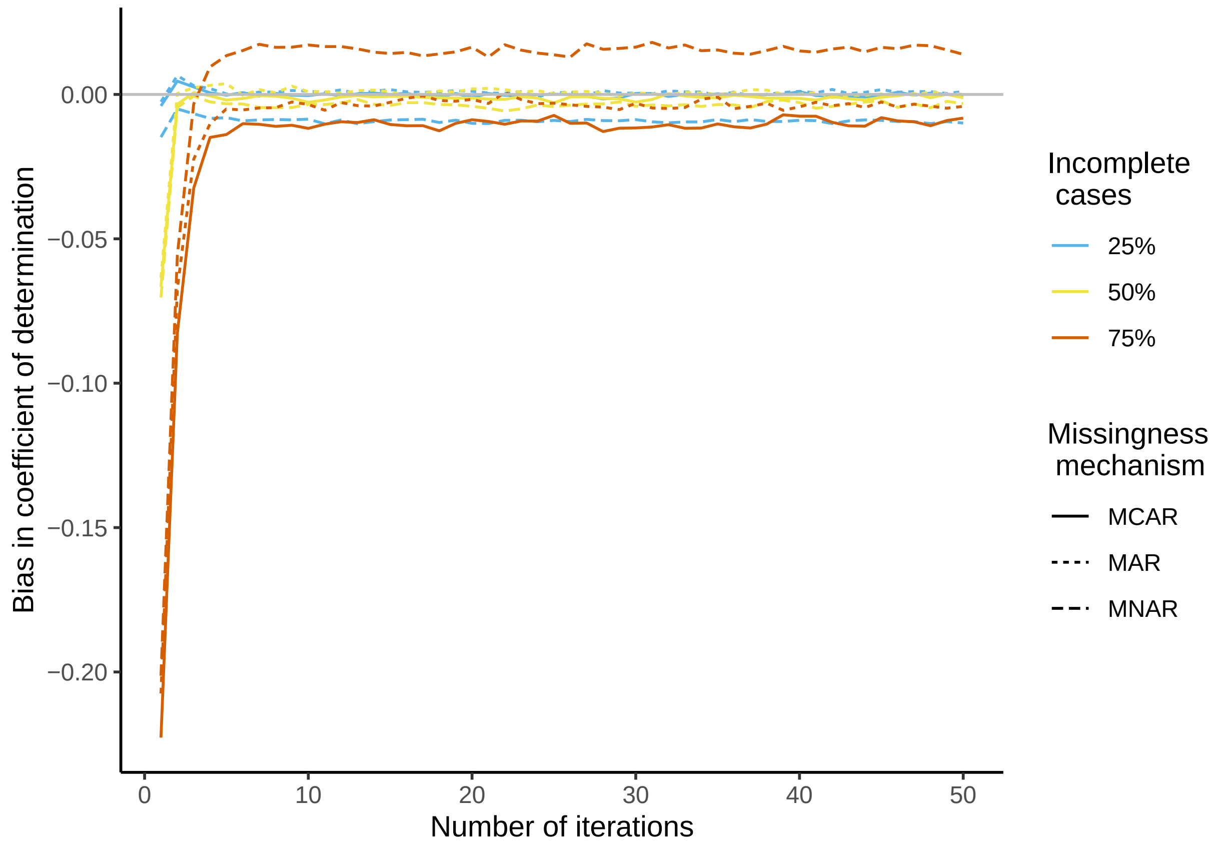
Bias in estimate depends mostly on the amount of missingness (less info to estimate the relation??). Overall 75% incomplete cases performs the worst, but is stable for $t \geq 6$.



Cov is more or less fine, except for MNAR with only 0.05% incomplete cases –> due to bias in the estimate in combination with very narrow CI (see CIW!). However, all stable after it=3??



CIW clearly determined by the proportion of incomplete cases. Worst for p=0.95, then stable after it 7.



Magnitude of the bias in r^2 perfectly follows the missingness mechanisms: least bias for MCAR, most for MNAR. Worst effect of non-convergence for $p = 0.95$, MNAR, but stable from it=5.

Discussion

- need for multivar parameters!!
- Estimates etc do not improve after it $= 7$
- Convergence diagnostics keep improving, but are all stable after 20-30 it
- Univariate thetas do not differentiate between missingness mechanisms or proportions of incomplete cases
- Determining non-stationarity with lambda is more difficult than with qhat :(

methodological explanation is that rhat and ac have a lag (few it to inform your statistic) –> will always indicate conv slower than the est

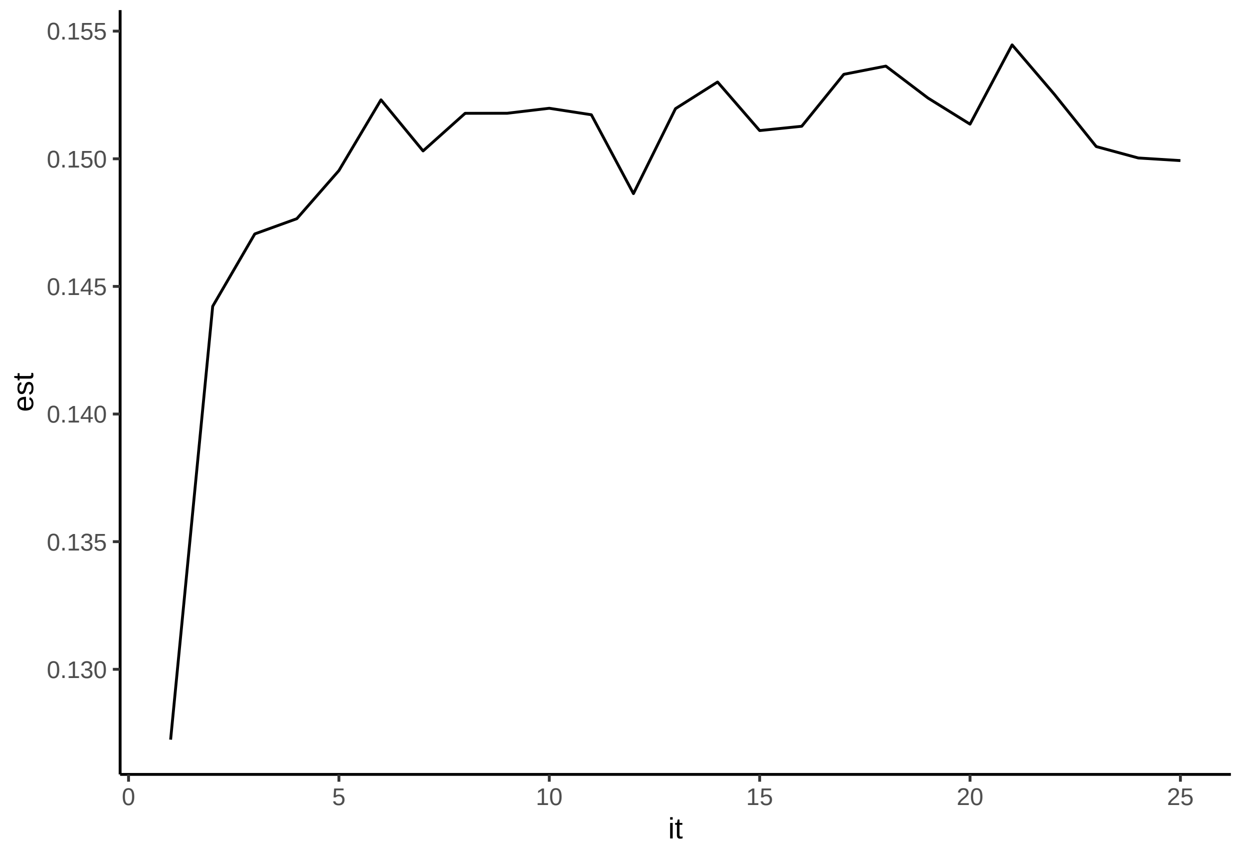
calc rhat and cov for each block of 5 it

disc: conf val is likely to happen much sooner

as soon as we start down-weighting the first few it from the calc, the memory effect would

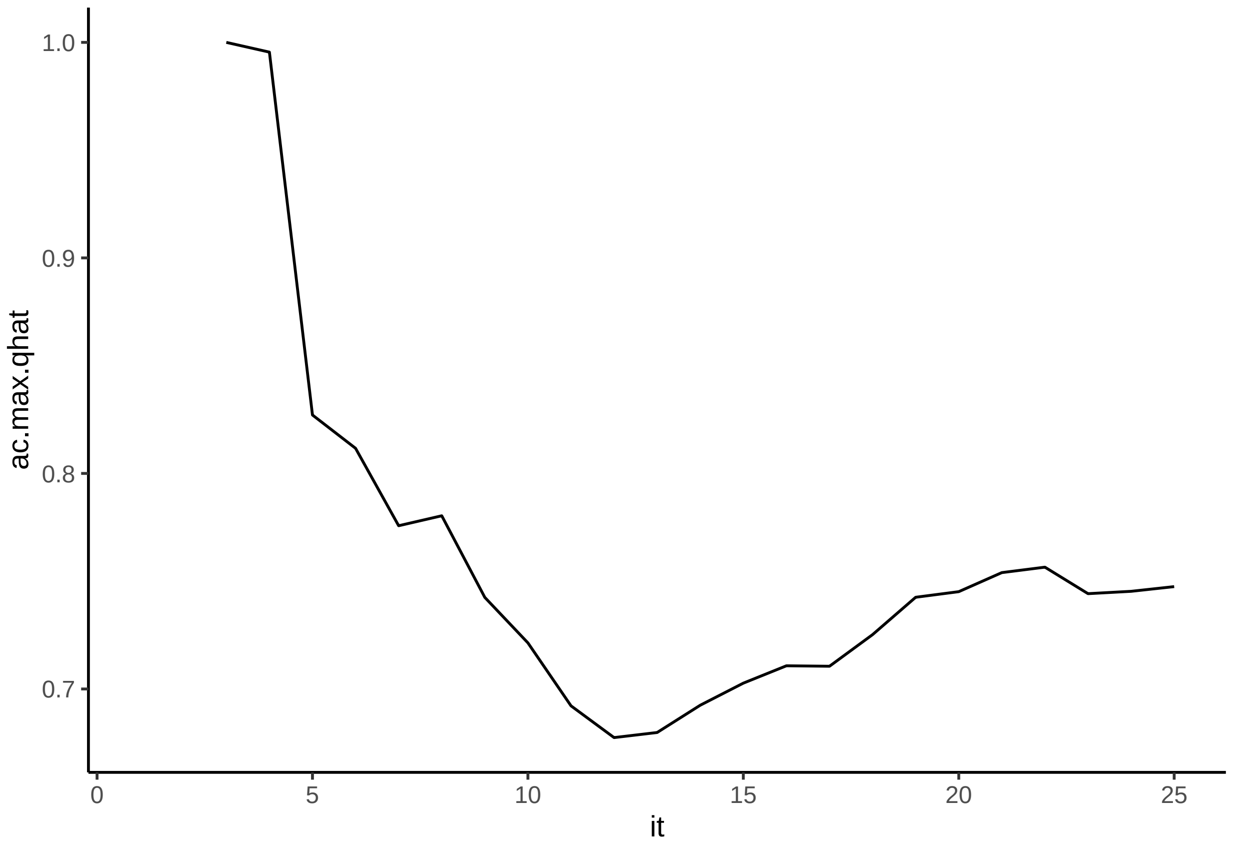
Example data

```r
# load data
load("Data/example.Rdata")
load("Data/example_mids.Rdata")

# plot each var of example data
plotfun <- function(v, ...){
ggplot(example, aes(x = .data[["it"]], y = .data[[v]])) +
  geom_line() +
  theme_classic()
}

plotfun(v="est")
```



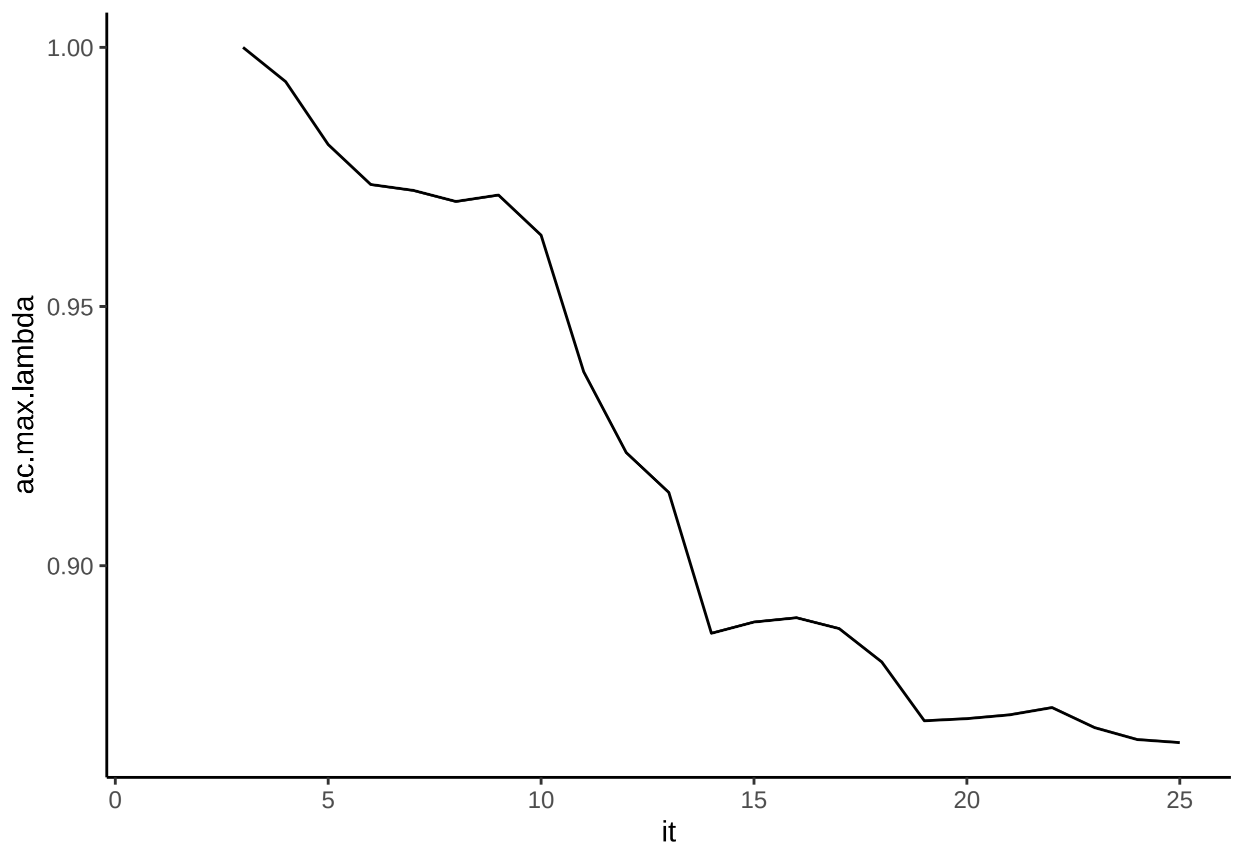
```r
plotfun(v="ac.max.qhat") #+ list(scale_y_continuous(limits = c(0.65,1)))
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



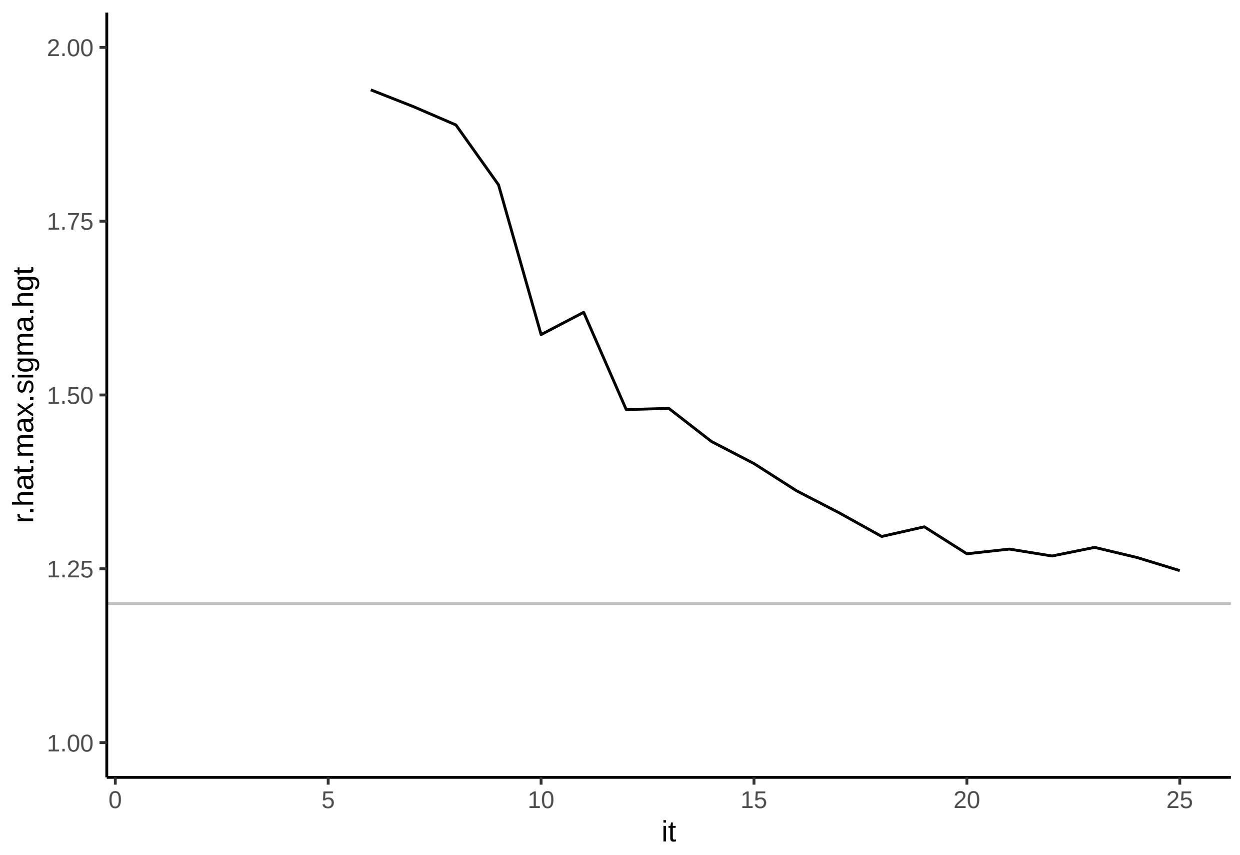
```
plotfun(v="ac.max.lambda")
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



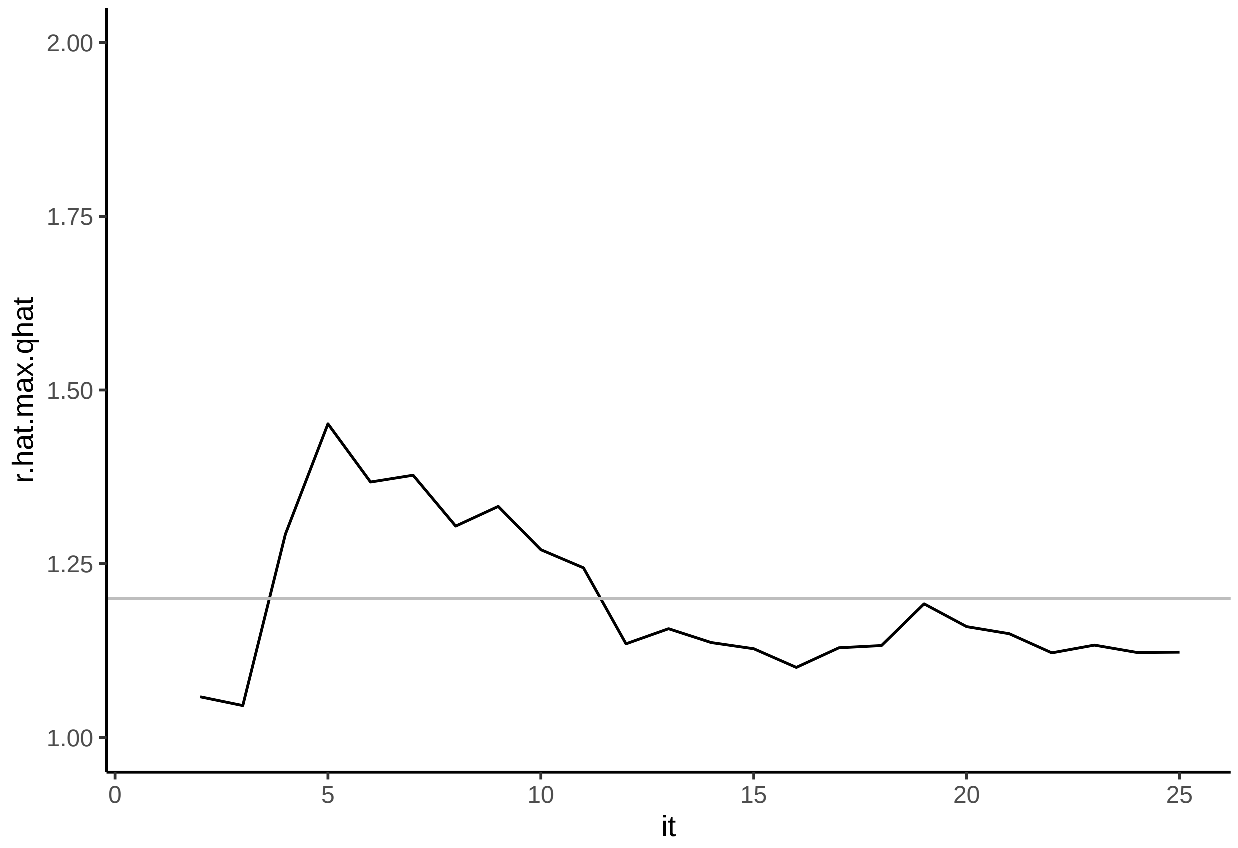
```
plotfun(v="r.hat.max.sigma.hgt")+list(geom_hline(yintercept = 1.2, color = "grey"), scale_y_continuous(
```

```
## Warning: Removed 5 row(s) containing missing values (geom_path).
```



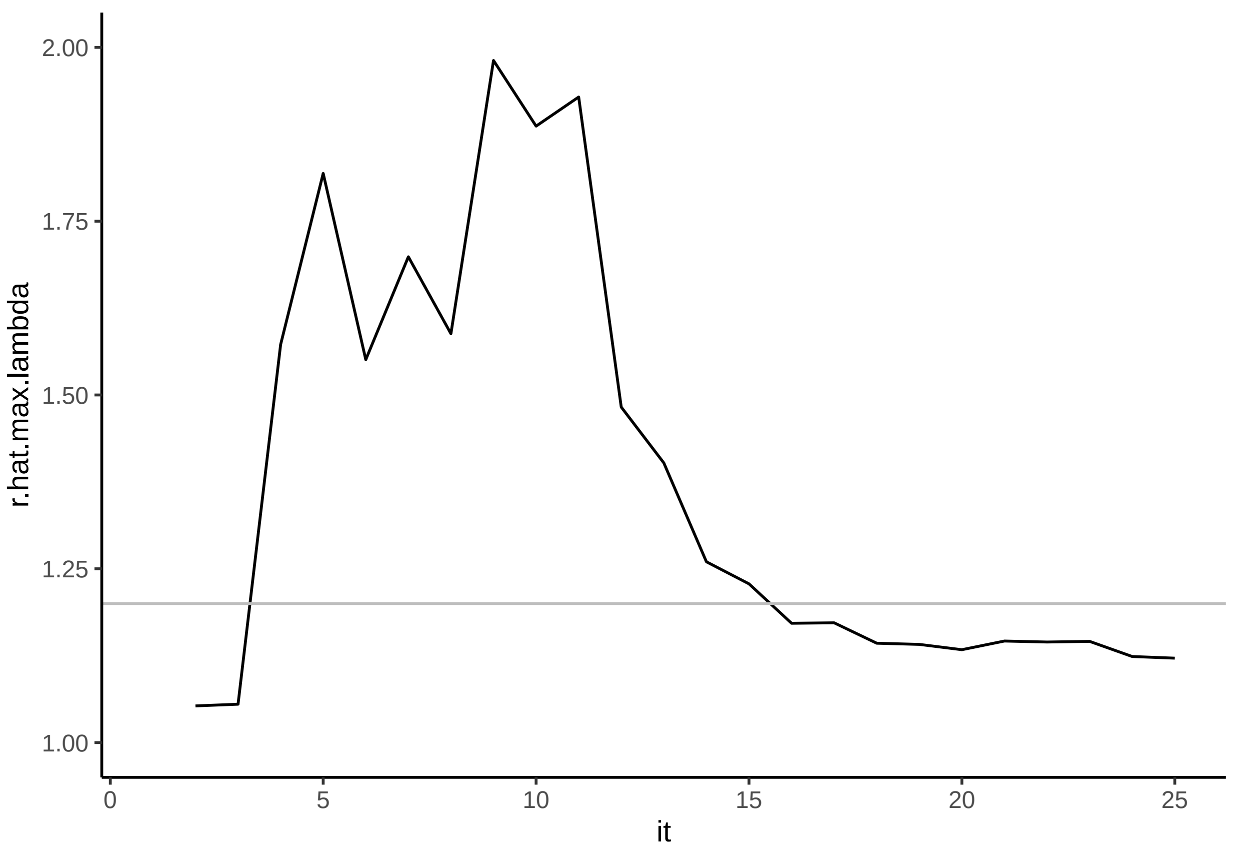
```
plotfun(v="r.hat.max.qhat")+list(geom_hline(yintercept = 1.2, color = "grey"), scale_y_continuous(limit
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

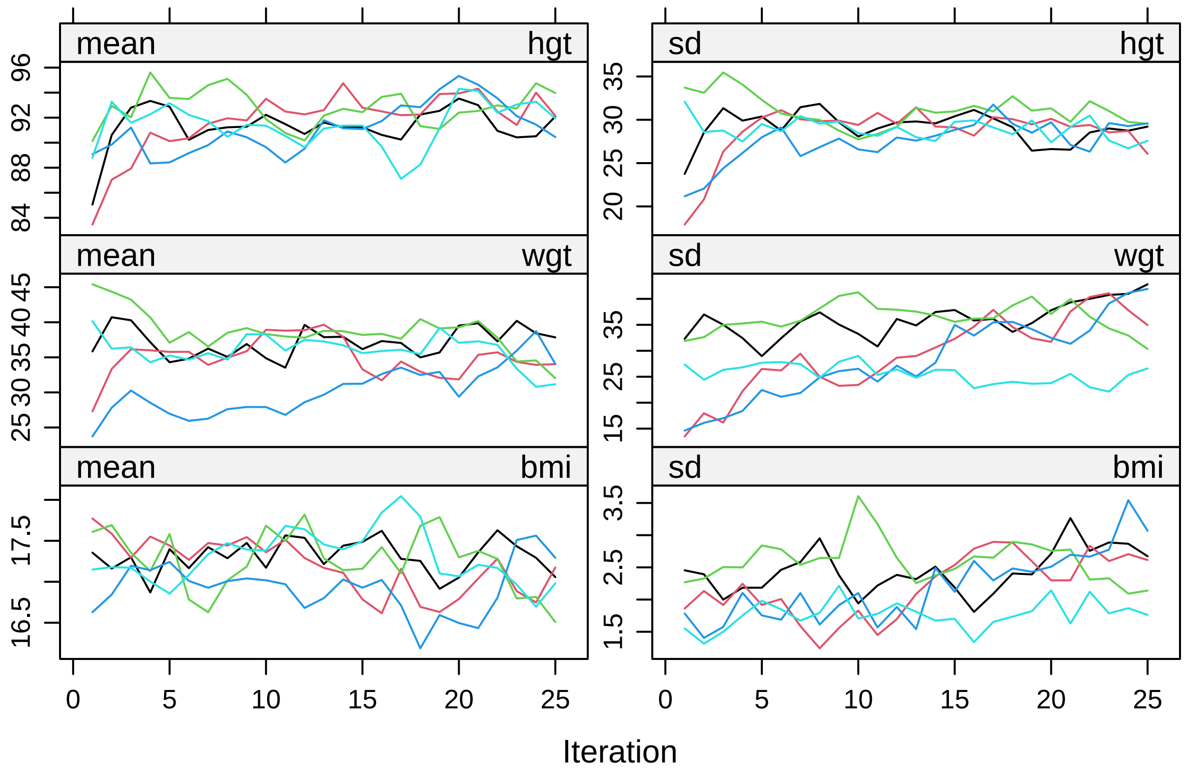


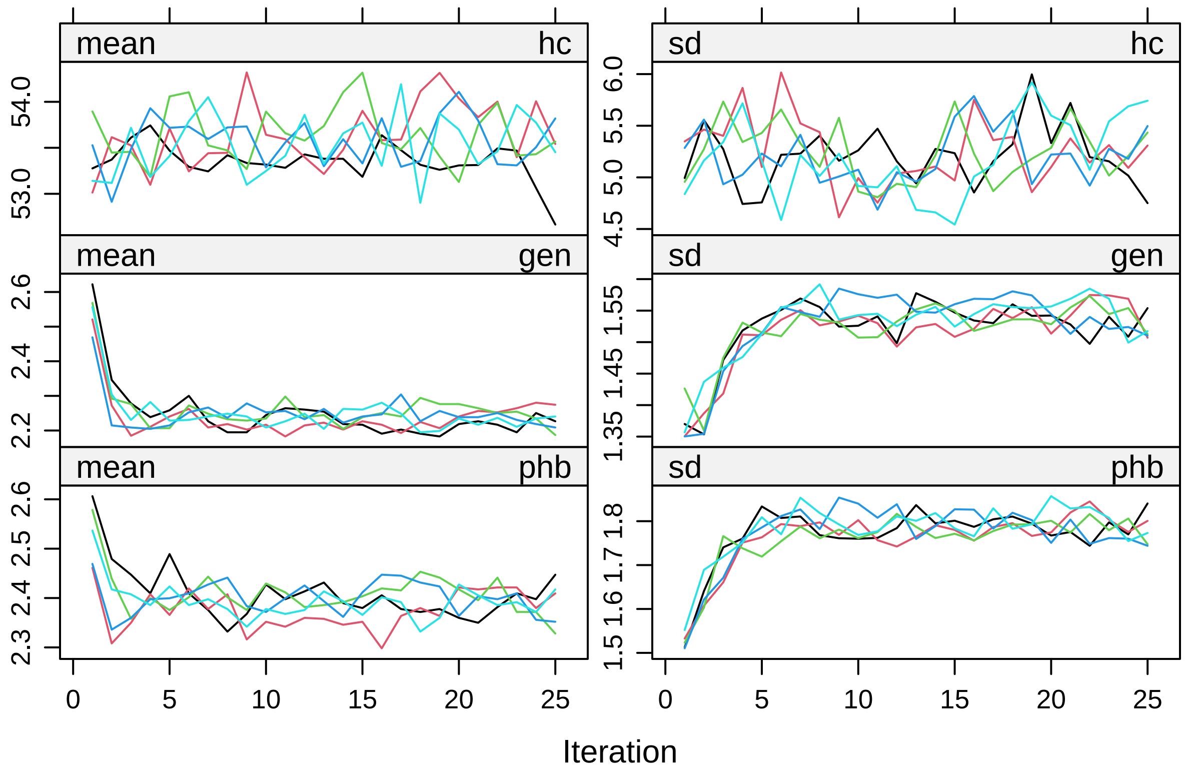
```
plotfun(v="r.hat.max.lambda")+list(geom_hline(yintercept = 1.2, color = "grey"), scale_y_continuous(lim
```

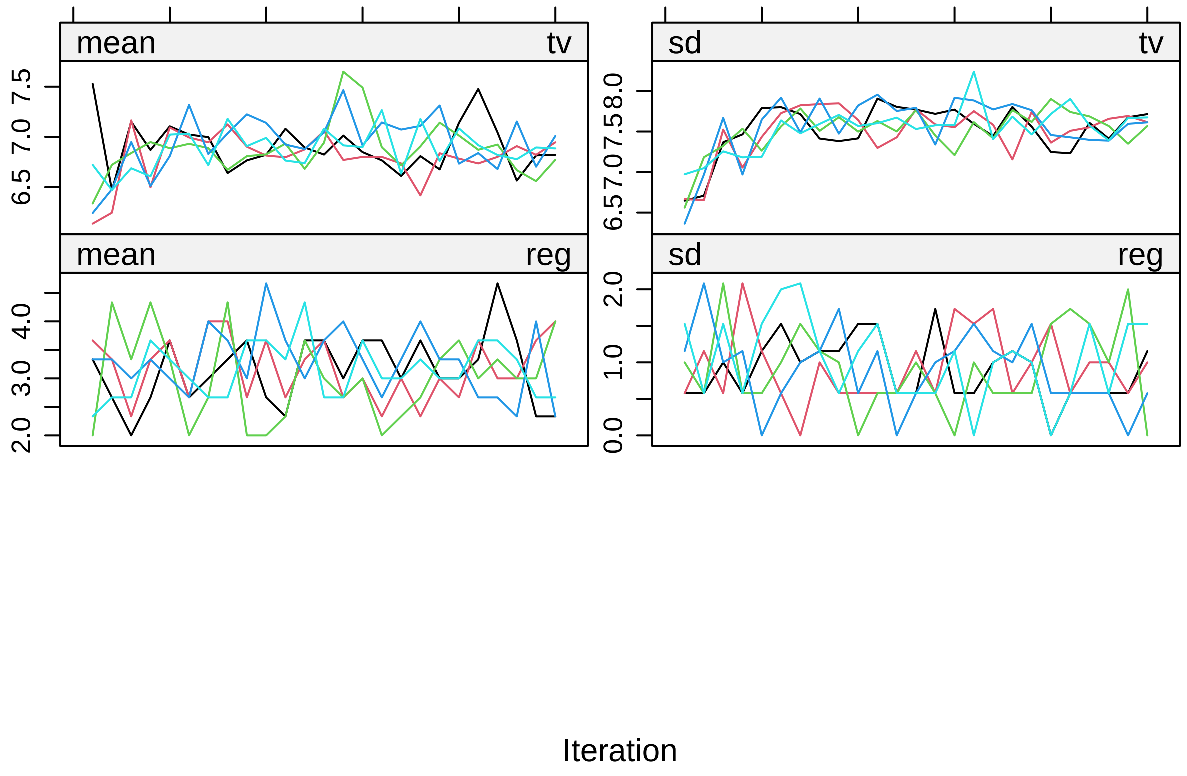
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



```r
plot(mids)
```


mean hgt
sd hgt
mean wgt
sd wgt
mean bmi
sd bmi
Iteration


mean
hc
sd
hc
mean
gen
sd
gen
mean
phb
sd
phb
Iteration
0
5
10
15
20
25
53.0
54.0
4.5
5.0
5.5
6.0
2.2
2.4
2.6
1.35
1.45
1.55
2.3
2.4
2.5
2.6
1.5
1.6
1.7
1.8



References (incomplete)

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

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.