Imputation Diagnostics

Utrecht University Winter School: Missing Data in R

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

Imputation Model Convergence

Plausibility of Imputed Values

Example Data

The example data are synthesized from questionnaire data collected by Lang, Salter, and Adams (2009).

- $N = 87$
- $P = 33$ Likert-type variables assessing:
    - Perceptions and definitions of racism
    - Political affiliation
    - Support for affirmative action policies
    - Belief in meritocratic ideals

The data synthesis involved:

1. Resampling the original data to produce a new sample of 250 cases
2. Adding Gaussian noise
3. Imposing 25% MAR missing
    - MAR Predictors = Political Affiliation, Definition of Racism

Imputation Diagnostics

After we run an MI routine, we need to make sure that the procedure has performed as expected.

Problems can arise to two different places:

1. The imputation model may fail to converge.
2. The imputed values may be invalid.

We need to examine our results to check for these problems.

Imputation Model Convergence

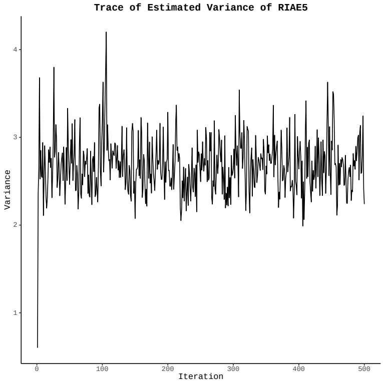
The imputation model is usually estimated through some form of Bayesian simulation.

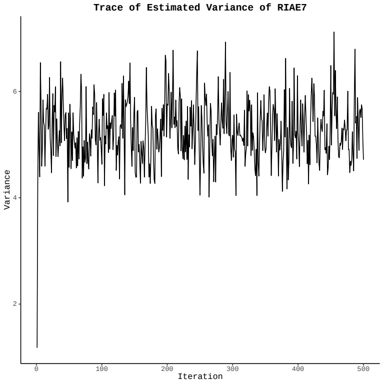
- Gibbs sampled parameters form a *Markov Chain*.
    - Each draw is dependent on only its immediate predecessor in the chain.
    - $\theta^{(\boldsymbol{t})} | \theta^{(\boldsymbol{t}-\mathbf{1})} \perp \theta^{(\boldsymbol{t}-\boldsymbol{j})} \; \forall \boldsymbol{j} > \mathbf{1}$
- Early elements of a Markov chain are similar to the starting values.
    - Samples are poor approximations of the true posterior.
- We must let the sampler iterate for a while to allow the estimates time to separate from their starting values.
    - We call these initial iterations “burn-in” or “warm-up” iterations.

Traceplots

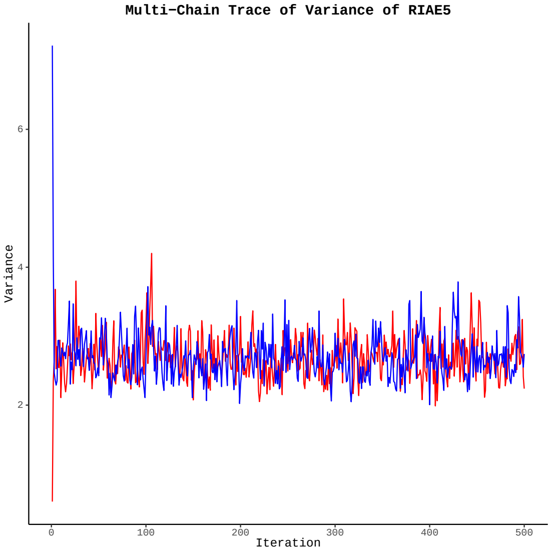
Once converged, each sampled parameter should “bounce” around some equilibrium point.

- The draws will never converge to a single point.
- Deterministic convergence would defeat the purpose of simulation.


Trace of Estimated Variance of RIAE5


Trace of Estimated Variance of RIAE7

Traceplots


Multi−Chain Trace of Variance of RIAE5
Variance
2
4
6
Iteration
0
100
200
300
400
500

Potential Scale Reduction Factor

Suppose we have M length-N Markov chains for the same parameter, θ.

- If these chains have converged, all M chains should be sampling from the same parameter space.
- The pooled *total variance* and the *within-chain* variances should be about the same.

The Gelman and Rubin (1992) *Potential Scale Reduction Factor*, $\widehat{R}$, quantifies this concept:

$$\widehat{R} = \sqrt{\frac{T}{W}}$$

$\widehat{R}$ will approach 1.0 at convergence.

- $\widehat{R} < 1.1$ or 1.2 suggests acceptable convergence.

Potential Scale Reduction Factor

$$\widehat{R} = \sqrt{\frac{T}{W}}$$

The total variance, T, is the weighted average of the within-chain variance, W, and the between-chain variance, B.

$$T = \frac{N-1}{N} W + \frac{1}{N} B$$

$$W = \frac{1}{M} \sum_{m=1}^{M} \mathrm{var}\left(\theta_m\right)$$

$$B = \frac{N}{M-1} \sum_{m=1}^{M} \left(\bar{\theta}_m - \bar{\theta}\right)^2$$

Example: Potential Scale Reduction Factor

We can compute $\hat{R}$ statistics for `mice()` models using the `Rhat.mice()` function from the **miceadds** package.

```r
## Impute missing values:
miceOut <- mice(data   = incompleteData,
                m      = 25,
                maxit  = 50,
                method = "norm",
                seed   = 235711)

## Compute PSR factors:
Rhat.mice(miceOut)
```

```
  variable MissProp Rhat.M.imp Rhat.Var.imp
1    riae2     22.8   1.004677     1.005062
2    riae3     24.0   1.009668     1.014859
3    riae7     26.4   1.001777     1.005287
4    riae8     22.8   1.008973     1.017599
5    riae9     24.0   1.014319     1.002627
6   riae11     25.2   1.006530     1.005156
```

More Imputation Model Convergence

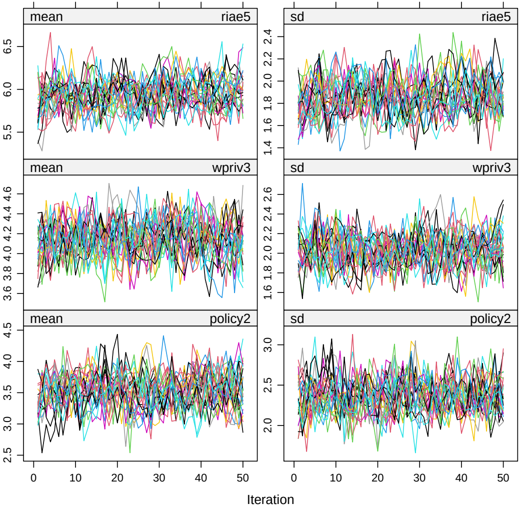
A convergent imputation model will produce imputed values that fluctuate around an equilibrium point.

- Imputation model convergence can be assessed indirectly by looking at plots of the item-level sufficient statistics for each imputation.

This approach is automated for **mice** via `plot.mice()`.

```r
## Impute missing values:
miceOut <- mice(data   = incompleteData,
                m      = 25,
                maxit  = 50,
                method = "norm",
                seed   = 235711)

## Create diagnostic traceplots:
plot(miceOut, c("riae5", "wpriv3", "policy2"))
```

More Imputation Model Convergence



Imputed Value Plausibility

We need to ensure that the imputations are sensible.

- Imputed values shouldn't be *too* dissimilar from their observed counterparts.
  - What constitutes *too* much dissimilarity is subjective and problem-specific.

We can assess dissimilarity graphically or through summary statistics.

- Out-of-bounds values for the imputations are perfectly acceptable.
  - MI is *NOT* designed to maintain the range.
  - We don't want wildly extreme values, though.
- The means of the observed and imputed components of each variable shouldn't differ too much.
  - Again, how much is *too* much is subjective.

Numeric Imputation Checks

```r
## Fill the missing values with imputations:
impList <- complete(miceOut, "all")

## Computes means:
rawMeans <- colMeans(missData, na.rm = TRUE)
impMeans <- do.call("rbind", impList) %>% colMeans()

## Compute standard deviations:
rawSds <- sapply(missData, sd, na.rm = TRUE)
impSds <- lapply(impList, function(x) sapply(x, sd)) %>%
    do.call(rbind, .) %>%
    colMeans()

## Compute ranges:
rawRanges <- sapply(missData, range, na.rm = TRUE)
impRanges <- do.call("rbind", impList) %>% sapply(range)
```

Numeric Imputation Checks

Compare observed and imputation-based means:

```
vars <- grep("policy\\d", colnames(missData))

round(rawMeans[vars], 3)

policy1 policy3 policy4 policy5 policy6 policy2
  3.020   3.724   3.564   3.746   4.483   3.558

round(impMeans[vars], 3)

policy1 policy3 policy4 policy5 policy6 policy2
  3.343   4.010   3.223   3.413   4.435   3.666
```

Numeric Imputation Checks

Compare observed and imputation-based standard deviations:

```
round(rawSds[vars], 3)

policy1 policy3 policy4 policy5 policy6 policy2
  2.045   2.181   2.035   2.015   1.956   2.237

round(impSds[vars], 3)

policy1 policy3 policy4 policy5 policy6 policy2
  2.259   2.309   2.115   2.167   2.093   2.187
```

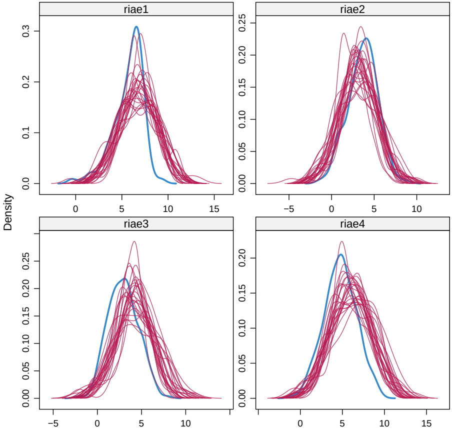
Numeric Imputation Checks

Compare observed and imputation-based ranges:

```
round(rawRanges[ , vars], 3)

     policy1 policy3 policy4 policy5 policy6 policy2
[1,]  -1.383  -1.342  -2.260  -1.106   0.126  -2.221
[2,]   8.641   9.551   9.471   8.885   9.098  10.237

round(impRanges[ , vars], 3)

     policy1 policy3 policy4 policy5 policy6 policy2
[1,]  -3.290  -4.887  -4.717  -2.696  -3.450  -4.241
[2,]  10.751  13.251   9.471  12.595  11.724  12.047
```

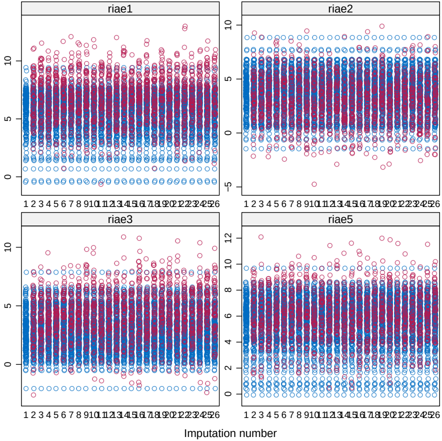
Graphical Imputation Checks

```r
## Overlaid density plots of imputed vs. observed values:
densityplot(miceOut, ~ riae1 + riae2 + riae3 + riae4, layout = c(2, 2))
```



Graphical Imputation Checks

```
## Stripplots of imputed vs. observed values:
stripplot(miceOut, riae1 + riae2 + riae3 + riae5 ~ .imp, layout = c(2, 2))
```



References

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, *7*(4), 457–472.

Lang, K. M., Salter, P. S., & Adams, G. (2009, April). What drives the relationship between conservatism and racism? A mediation analysis. In *Proceedings of the annual meeting of the Southwestern Psychological Association.*