Multiple Imputation with Categorical Variables

Stats Camp 2018: Missing Data Analysis

Kyle M. Lang

Department of Methodology & Statistics
Tilburg University

19–21 October 2018

Outline

- Discuss imputation diagnostics
  - Assessing imputation model convergence
  - Checking the imputations' plausibility
- Look at graphical and numerical options for both

Example Data

These data were analyzed by Lang, Salter, and Adams (2009).

- $N = 87$
- $V = 33$
- Variables assessing:
  - Perceptions of and Definitions of Racism
  - Political Affiliation
  - Support for Affirmative Action Policies
  - Belief in meritocratic ideals
- Almost no missing data
  - I've artificially imposed 30% MAR missing data on all variables (expect political affiliation) using political affiliation as the MAR predictor.

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 not be plausible.

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.

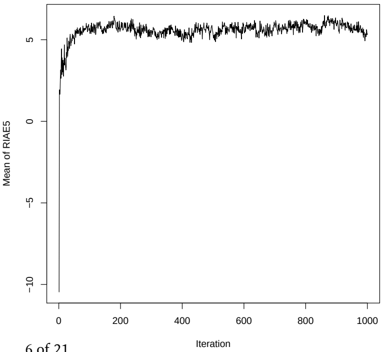
- Gibbs sampled parameters form a *Markov Chain*.
  - Each draw is dependent on only its immediate predecessor in the chain.
  - $\theta^{(t)} | \theta^{(t-1)} \perp \theta^{(t-j)} \, \forall j > 1$
- Early elements of a Markov chain are similar to the starting values.
  - Samples are poor approximations 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

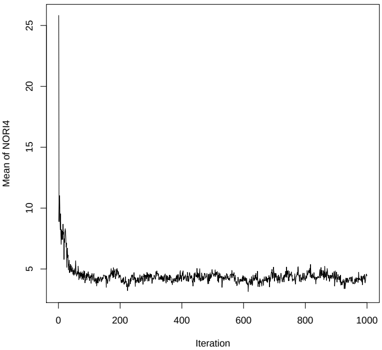
Once converged, each sampled imputation model parameter should “bounce” around an equilibrium point.

- The draws will never converge onto a single point.
- That would defeat the purpose of simulation-based inference.

Trace of RIAE5's Estimated Mean



Trace of NORI4's Estimated Mean



Potential Scale Reduction Factor

Suppose we have two Markov chains for the same parameter.

- If these chains have converged, the average distance between any two points on separate chains should be the same as the average distance between two points on the same chain.
- The *between-chain* variance should, on average, equal the *within-chain* variance.

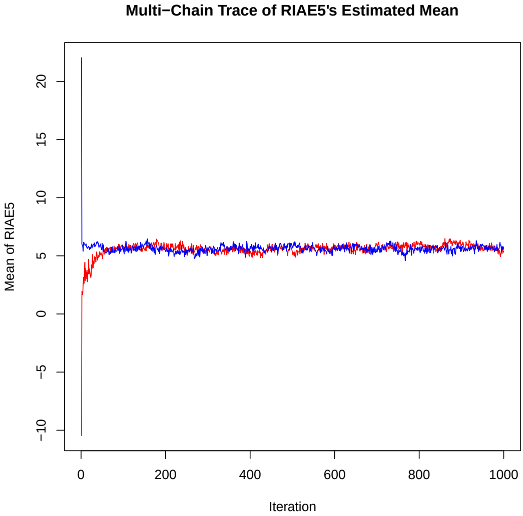
The *Potential Scale Reduction Factor* $\widehat{R}$ quantifies this concept:

$$\widehat{R} = \frac{\hat{\sigma}^2_{between}}{\hat{\sigma}^2_{within}}$$

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

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

Example: Potential Scale Reduction Factor



Example: Potential Scale Reduction Factor

```
## Create matrices of the full and burnt-in chains:
iterMat  <- cbind(chain1, chain2)
burntMat <- iterMat[201 : 1000, ]

## Full Chain R-Hat:
wVar1 <- mean(apply(iterMat, 2, var))
bVar1 <- mean(apply(iterMat, 1, var))
rHat1 <- bVar1 / wVar1
rHat1

## [1] 1.682921

## Burnt-In R-Hat:
wVar2 <- mean(apply(burntMat, 2, var))
bVar2 <- mean(apply(burntMat, 1, var))
rHat2 <- bVar2 / wVar2
rHat2

## [1] 1.104803
```

More Imputation Model Convergence

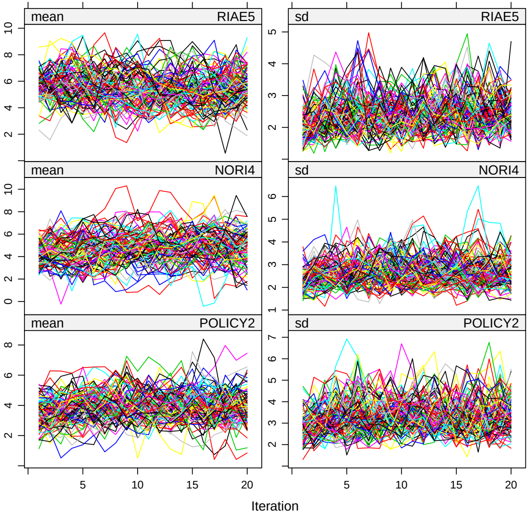
A convergent imputation model will produce imputed values that fluctuate around an equilibrium point.

- 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
miceOut1 <- readRDS(paste0(dataDir, "miceOut1.rds"))
plot(miceOut1, c("RIAE5", "NORI4", "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
rawMeans <- colMeans(missData, na.rm = TRUE)
impMeans <- colMeans(do.call("rbind", impList))

rawSds <- apply(missData, 2, sd, na.rm = TRUE)
sdList <- lapply(impList, function(x) sapply(x, FUN = sd))
impSds <- colMeans(do.call(rbind, sdList))

rawRanges <- apply(missData, 2, range, na.rm = TRUE)
impRanges <- sapply(do.call("rbind", impList), range)
```

Numeric Imputation Checks

```
round(rawMeans[1 : 5], 3)

## RIAE2 RIAE3 RIAE7 RIAE8 RIAE9
## 3.677 3.108 3.774 3.092 3.726

round(impMeans[1 : 5], 3)

## RIAE2 RIAE3 RIAE7 RIAE8 RIAE9
## 3.697 3.122 3.340 3.134 3.010
```

Numeric Imputation Checks

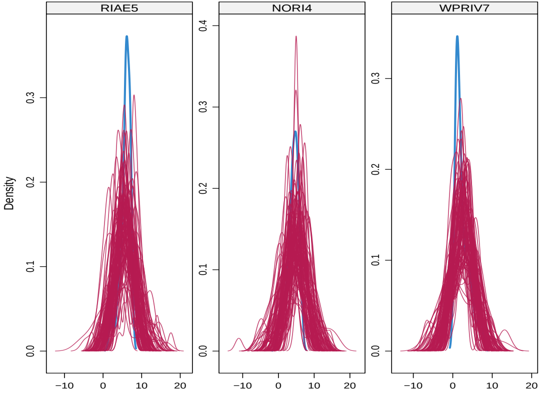
```
round(rawSds[1 : 5], 3)

## RIAE2 RIAE3 RIAE7 RIAE8 RIAE9
## 1.696 1.522 2.060 1.693 1.700

round(impSds[1 : 5], 3)

## RIAE2 RIAE3 RIAE7 RIAE8 RIAE9
## 2.223 1.906 2.917 2.157 2.283
```

Numeric Imputation Checks

```
round(rawRanges[ , 1 : 5], 3)

##      RIAE2 RIAE3 RIAE7 RIAE8 RIAE9
## [1,]     1     1     1     1     1
## [2,]     7     6     7     7     7

round(impRanges[ , 1 : 5], 3)

##       RIAE2  RIAE3   RIAE7  RIAE8   RIAE9
## [1,] -7.819 -8.486 -17.790 -9.473 -10.184
## [2,] 21.724 15.941  15.869 16.551  11.201
```

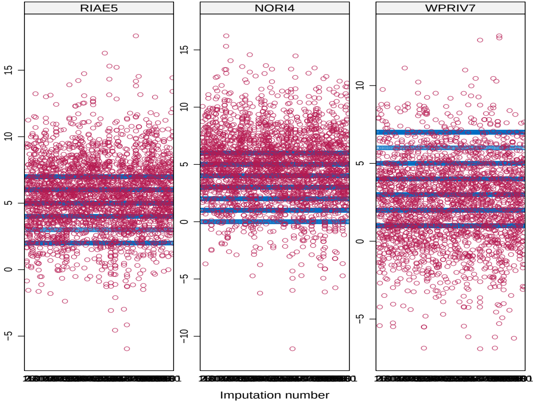
Graphical Imputation Checks

```
## Overlaid density plots of imputed vs. observed values:
densityplot(miceOut1, data = ~RIAE5 + NORI4 + WPRIV7,
            layout = c(3, 1))
```



Graphical Imputation Checks

```
## Scatterplots of imputed vs. observed values:
stripplot(miceOut1, data = RIAE5 + NORI4 + WPRIV7 ~ .imp,
          layout = c(3, 1))
```

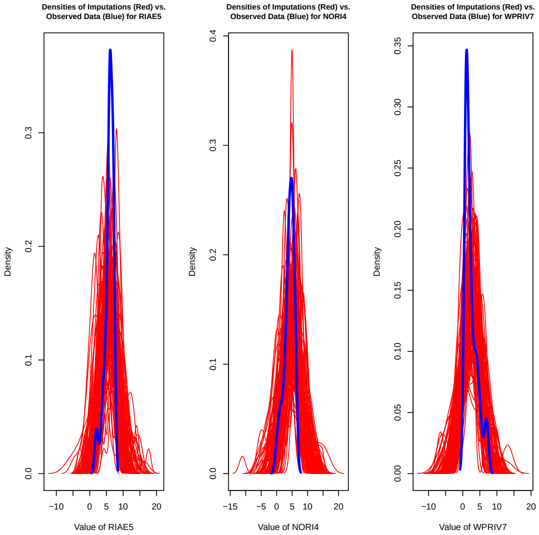


Graphical Imputation Checks

We can use the `plotImps` function from the **SURF** package to generate overlaid density plots for arbitrary lists of imputed data.

```
## Overlaid density plots of imputed vs. observed values:
par(mfrow = c(1, 3), cex.main = 0.9)

rMat                       <- is.na(miceOut1$data)
type                       <- miceOut1$method
type[type == "norm"] <- "con"

plotImps(impList   = impList,
         rMat      = rMat,
         typeVec   = type,
         targetVar = c("RIAE5", "NORI4", "WPRIV7"))
```

Graphical Imputation Checks



References

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.*