Using Multiple Hot Deck Data Sets for Inference

Skyler Cranmer
Ohio State University

Jeff Gill
Washington University St. Louis

Natalie Jackson
The Huffington Post

Andreas Murr
University of Oxford

David A. Armstrong II
University of Wisconsin-Milwaukee

February 24, 2020

This document will walk you through some of the methods you could use to generate pooled model results that account for both sampling variability and across imputation variability. The package `hot.deck` does not come with a set of functions to do inference, so we will show you how you could use the data generated by `hot.deck` in combination with `glm.mids` (and similarly `lm.mids`) from the `mice` package, `zelig` from the `Zelig` package and by using `MIcombine` from the `mitools` package on a list of model objects.

1 Generating Imputations

The data we will use come from Poe, Tate and Keith (1999) dealing with democracy and state repression. First we need to call the `hot.deck` routine on the dataset.

```
library(hot.deck)
data(isq99)
out <- hot.deck(isq99, sdCutoff=3, IDvars = c("IDORIGIN", "YEAR"))

## Warning in hot.deck(isq99, sdCutoff = 3, IDvars = c("IDORIGIN", "YEAR")):  52 observations with no observed data.
These observations were removed
## Warning in hot.deck(isq99, sdCutoff = 3, IDvars = c("IDORIGIN", "YEAR")):  45 of 4661 imputations with # donors
< 5, consider increasing sdCutoff or using method='p.draw'
```

This shows us that there are still 47 observations with fewer than 5 donors. Using a different method or further widening the `sdCutoff` parameter may alleviate the problem. If you want to see the frequency distribution of the number of donors, you could look at:

```
numdonors <- sapply(out$donors, length)
numdonors <- sapply(out$donors, length)
numdonors <- ifelse(numdonors > 5, 6, numdonors)
numdonors <- factor(numdonors, levels=1:6, labels=c(1:5, ">5"))
table(numdonors)
```

```
## numdonors
##    1    2    3    4    5   >5
##   18   10   11    6   20 4596
```

Before running a model, three variables have to be created from those existing. Generally, if variables are deterministic functions of other variables (e.g., transformations, lags, etc...) it is advisable to impute the constituent variables of the calculations and then do the calculations after the fact. Here, we need to lag the `AI` variable and create percentage change variables for both population and per-capita GNP. First, to create the lag of `AI`, `PCGNP` and `LPOP`. To do this, we will make a little function.

```
tscslag <- function(dat, x, id, time){
        obs <- apply(dat[, c(id, time)], 1, paste, collapse=".")
        tm1 <- dat[[time]] - 1
        lagobs <- apply(cbind(dat[[id]], tm1), 1, paste, collapse=".")
        lagx <- dat[match(lagobs, obs), x]
}
for(i in 1:length(out$data)){
    out$data[[i]]$lagAI <- tscslag(out$data[[i]], "AI", "IDORIGIN", "YEAR")
    out$data[[i]]$lagPCGNP <- tscslag(out$data[[i]], "PCGNP", "IDORIGIN", "YEAR")
    out$data[[i]]$lagLPOP <- tscslag(out$data[[i]], "LPOP", "IDORIGIN", "YEAR")
}
```

Now, we can use the lagged values of `PCGNP` and `LPOP`, to create percentage change variables:

```
for(i in 1:length(out$data)){
    out$data[[i]]$pctchgPCGNP <- with(out$data[[i]], c(PCGNP-lagPCGNP)/lagPCGNP)
    out$data[[i]]$pctchgLPOP <- with(out$data[[i]], c(LPOP-lagLPOP)/lagLPOP)
}
```

2 Running Models on Multiple Hot Decking Result

2.1 Using Zelig

In version ≥ 5.0 of `Zelig`, the output from `hot.deck` will have to be converted into a format that looks like Amelia's. You can do this as follows:

```
out <- hd2amelia(out)
```

Then, with the output in the appropriate format, we can use `Zelig` to do the modeling.

```
library(Zelig)
z <- zelig(AI ~ lagAI + pctchgPCGNP + PCGNP + pctchgLPOP + LPOP + MIL2 + LEFT +
    BRIT + POLRT + CWARCOW + IWARCOW2, data=out, model="normal", cite=FALSE)
summary(z)

## Model: Combined Imputations
##
##              Estimate Std.Error z value Pr(>|z|)
## (Intercept)  5.31e-01  1.34e-01    3.95  7.8e-05
## lagAI        4.68e-01  2.20e-02   21.29  < 2e-16
## pctchgPCGNP  5.44e-03  7.35e-03    0.74   0.4594
## PCGNP       -1.99e-05  3.03e-06   -6.56  5.5e-11
## pctchgLPOP  -5.07e-01  1.08e+00   -0.47   0.6374
## LPOP         7.28e-02  8.09e-03    9.01  < 2e-16
## MIL2         1.16e-01  4.41e-02    2.63   0.0085
## LEFT        -1.41e-01  4.93e-02   -2.85   0.0043
## BRIT        -1.34e-01  3.12e-02   -4.28  1.8e-05
## POLRT       -6.96e-02  1.13e-02   -6.14  8.5e-10
## CWARCOW      6.12e-01  5.58e-02   10.96  < 2e-16
## IWARCOW2     1.68e-01  5.65e-02    2.98   0.0029
##
## For results from individual imputed datasets, use summary(x, subset = i:j)
## Next step: Use 'setx' method
```

Note that the summary indicates that the results have been combined across 5 multiply imputed datasets.

2.2 Using MIcombine

You can use the `MIcombine` command from the `mitools` package to generate inferences, too. Here, you have to produce a list of model estimates and the function will combine across the different results.

```
# initialize list
results <- list()
# loop over imputed datasets
for(i in 1:length(out$imputations)){
    results[[i]] <- lm(AI ~ lagAI + pctchgPCGNP + PCGNP + pctchgLPOP + LPOP + MIL2 + LEFT +
    BRIT + POLRT + CWARCOW + IWARCOW2, data=out$imputations[[i]])
}
summary(mitools::MIcombine(results))

## Multiple imputation results:
##       MIcombine.default(results)
##                   results           se         (lower         upper) missInfo
## (Intercept)  5.306640e-01 1.343256e-01  2.662818e-01  7.950461e-01     12 %
## lagAI        4.684916e-01 2.200209e-02  4.220845e-01  5.148987e-01     54 %
## pctchgPCGNP  5.437420e-03 7.348930e-03 -1.311326e-02  2.398810e-02     90 %
## PCGNP       -1.988956e-05 3.032863e-06 -2.589274e-05 -1.388637e-05     19 %
## pctchgLPOP  -5.073886e-01 1.076611e+00 -2.869658e+00  1.854881e+00     65 %
## LPOP         7.284485e-02 8.087428e-03  5.699280e-02  8.869691e-02      1 %
## MIL2         1.159230e-01 4.407117e-02  2.488961e-02  2.069563e-01     46 %
## LEFT        -1.406836e-01 4.929611e-02 -2.403317e-01 -4.103552e-02     35 %
## BRIT        -1.337468e-01 3.122174e-02 -1.950569e-01 -7.243661e-02      8 %
## POLRT       -6.963110e-02 1.134924e-02 -9.367434e-02 -4.558786e-02     55 %
## CWARCOW      6.119057e-01 5.583945e-02  5.009920e-01  7.228195e-01     23 %
## IWARCOW2     1.680549e-01 5.646443e-02  5.600692e-02  2.801029e-01     22 %
```

2.3 Using mids

The final method for combining results is to convert the data object returned by the `hot.deck` function to an object of class `mids`. This can be done with the `datalist2mids` function from the `miceadds` package.

```
out.mids <- miceadds::datalist2mids(out$imputations)

## Warning:  Number of logged events:  1

s <- summary(mice::pool(mice::lm.mids(AI ~ lagAI + pctchgPCGNP + PCGNP + pctchgLPOP + LPOP + MIL2 + LEFT +
BRIT + POLRT + CWARCOW + IWARCOW2, data=out.mids)))
print(s, digits=4)
```

```
##               estimate std.error statistic      df   p.value
## (Intercept)  5.387e-01 1.491e-01    3.6135  50.491 6.956e-04
## lagAI        4.708e-01 1.941e-02   24.2476  41.152 0.000e+00
## pctchgPCGNP  3.573e-03 5.049e-03    0.7076   6.714 5.030e-01
## PCGNP       -1.963e-05 3.791e-06   -5.1778  16.724 7.966e-05
## pctchgLPOP  -8.040e-01 1.377e+00   -0.5840   6.472 5.790e-01
## LPOP         7.150e-02 9.197e-03    7.7742  72.690 3.829e-11
## MIL2         1.209e-01 5.345e-02    2.2619  11.150 4.464e-02
## LEFT        -1.342e-01 4.810e-02   -2.7912  52.087 7.324e-03
## BRIT        -1.369e-01 3.951e-02   -3.4650  22.430 2.156e-03
## POLRT       -6.755e-02 1.090e-02   -6.1990  19.904 4.793e-06
## CWARCOW      6.027e-01 5.755e-02   10.4732  65.585 1.332e-15
## IWARCOW2     1.604e-01 5.362e-02    2.9921 319.263 2.986e-03
```

References

Poe, Steven, C. Neal Tate and Linda Camp Keith. 1999. “Repression of the Human Right to Personal Integrity Revisited: A Global, Cross-National Study Covering the Years 1976-1993.” *International Studies Quarterly* 43:291–313.