SOCY7708: Hierarchical Linear Modeling
Instructor: Natasha Sarkisian

Missing data, Part 2

So in some, there are two algorithms we might want to follow to impute a nested dataset using MICE in Stata.

I. If we can use dummies for each level 2 unit in level 1 imputation, then we:

1. Impute level 1 using dummies for level 2 units
2. Aggregate level 1 variables to level 2, merge to level 2 file separately for each imputation
3. For each imputation separately, impute level 2 including aggregated level 1 variables (aggregated DV and level 1 IVs)
4. Merge each imputed level 2 file with the overall imputed level 1 data with correct imputation number

II. If we cannot use dummies for each level 2 unit in level 1 imputation because there are too few cases per level 2 unit or the model does not converge or computing resources are inadequate:

1. Aggregate all level 1 variables to level 2, merge to level 2 file
2. Impute level 2 using aggregates
3. Merge imputed level 2 file to level 1, one imputation at a time
4. Impute level 1, one file at a time
5. Merge all imputations together

For algorithm 1, in order to use dummies for each school in level 1 imputation, we need to make sure we have no schools with one unit per school; if we find them, we will combine all units that area one unit per school only into the same cluster (this is done in level 1 or combined file):

```
. gen count=1

. bysort sch_id: egen countt=total(count)

. tab countt
```

countt	Freq.	Percent	Cum.
1	19	0.14	0.14
2	28	0.20	0.34
3	72	0.52	0.86
4	60	0.43	1.30
5	110	0.80	2.09
6	162	1.17	3.26
7	189	1.37	4.63
8	320	2.32	6.95
9	477	3.45	10.40
10	440	3.18	13.58
11	561	4.06	17.64
12	732	5.30	22.93
13	936	6.77	29.71
14	1,316	9.52	39.23
15	1,290	9.33	48.56
16	1,136	8.22	56.78

```
         17 |      1,020        7.38       64.16
         18 |      1,152        8.33       72.49
         19 |      1,026        7.42       79.92
         20 |        380        2.75       82.67
         21 |        441        3.19       85.86
         22 |        418        3.02       88.88
         23 |        299        2.16       91.04
         24 |        216        1.56       92.61
         25 |         75        0.54       93.15
         26 |        208        1.50       94.65
         27 |         27        0.20       94.85
         28 |        140        1.01       95.86
         29 |        116        0.84       96.70
         30 |         90        0.65       97.35
         31 |         62        0.45       97.80
         33 |         66        0.48       98.28
         34 |         34        0.25       98.52
         35 |         70        0.51       99.03
         42 |         42        0.30       99.33
         45 |         45        0.33       99.66
         47 |         47        0.34      100.00
------------+-----------------------------------
      Total |     13,822      100.00

. list sch_id if countt==1

       +--------+
       | sch_id |
       |--------|
  211. |   6461 |
  233. |   6656 |
 1654. |   7658 |
 2426. |   8902 |
 5242. |  26792 |
       |--------|
 5378. |  29757 |
 5491. |  34614 |
 5497. |  34798 |
 5855. |  45252 |
 6091. |  45394 |
       |--------|
 6932. |  45895 |
 7018. |  45928 |
 7032. |  45939 |
 7176. |  46238 |
 7188. |  46301 |
       |--------|
 7562. |  47583 |
 9857. |  70245 |
10548. |  72251 |
12251. |  77520 |
       +--------+

. sum sch_id

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      sch_id |     13822    46240.13    26486.03       1249      91991

. gen cluster=sch_id

. replace cluster=91999 if countt==1
(19 real changes made)
```

```
. save "L:\socy7708\nels_science.dta"
file L:\socy7708\nels_science.dta saved
```

Following this strategy (algorithm 1), we add dummies to our level 1 imputation model. When we do this imputation, we will impute race5 using m. prefix. Note that I am omitting level 2 variables because I am including dummies for schools which would be collinear with level 2 variables.

Since we are using a lot of dummies, we are likely to run into errors, especially in logit-based models, so we will use persist right away to see the errors and troubleshoot.

```
. set matsize 1200

. use "L:\socy7708\nels_science.dta", clear

. ice science female spoken pared m.race5 i.cluster, cmd(spoken: ologit)
saving("L:\socy7708\nels_science_level1imp.dta", replace) m(1) dryrun
```

If this model does not converge, we could consider adding the clusters with 2 units per cluster into that combined cluster and try again. If it does converge, we will then use these imputed level 1 data to create level 2 aggregates and then generate the separate level 2 file and impute it (as discussed above). Once you have that imputed level 2 file, we will have to merge it with the imputed level 1 file, matching not only on school ID but also on imputation number, _mj:

```
. use "L:\socy7708\nels_science_level1imp.dta", clear
. merge m:1 sch_id _mj using "L:\socy7708\nels_imputed.dta"
. tab _merge
. drop _merge
. tab _mj
```

If the algorithm I strategy does not succeed, we can go back to the second strategy outlined above and use level 2 variables and level 2 aggregates in level 1 imputation. To do that, now that we imputed level 2 variables, we can merge them with our level 1 dataset separately for each imputation, e.g., for the first one:

```
. use "L:\socy7708\nels_imputed.dta", clear

. tab _mj
 imputation |
     number |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,013       16.67       16.67
          1 |      1,013       16.67       33.33
          2 |      1,013       16.67       50.00
          3 |      1,013       16.67       66.67
          4 |      1,013       16.67       83.33
          5 |      1,013       16.67      100.00
------------+-----------------------------------
      Total |      6,078      100.00

 keep if _mj==1
(5065 observations deleted)

. merge 1:m sch_id using "L:\socy7708\nels_science.dta"
```

```
    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                            13,822  (_merge==3)
    -----------------------------------------

. tab _merge

                _merge |      Freq.     Percent        Cum.
-----------------------+-----------------------------------
            matched (3) |     13,822      100.00      100.00
-----------------------+-----------------------------------
                  Total |     13,822      100.00

. drop _merge _mi _mj
. save "L:\socy7708\nels_imputed_1.dta"
```

We will create such datasets for each imputation; you can create a loop to do this:

```
use "L:\socy7708\nels_imputed.dta", clear
for num 1/5: preserve \ keep if _mj==X \ merge 1:m sch_id using
"L:\socy7708\nels_science.dta" \ tab _merge \ drop _merge _mi _mj \ save
"L:\socy7708\nels_imputed_X.dta", replace \ restore
```

Next, we will run that imputation without school dummies for each imputation separately, generating single imputation, and we will include level 2 variables:

```
for num 1/5: use nels_imputed_X.dta, clear \ ice science female spoken pared m.race5
public pct_esl morale sciencem femalem blackm latinom asianm nativem spokenm paredm,
cmd(spoken morale:ologit) saving(nels_science_impX, replace) m(1)
```

We would repeat this process 5 times, and then merge all 5 imputations together. First, let's drop imputation 0 from datasets 2-5:

```
. for num 2/5: use nels_science_impX.dta \ drop if _mj==0 \ replace _mj=X \ save
nels_science_impX.dta, replace
. clear all
. for num 1/5: append using nels_science_impX.dta
. tab _mj
. save nels_imputed_merged.dta, replace
```

<u>Longitudinal Example</u>

To deal with the longitudinal version of nested data (time points nested within cases), we will use a small, NLSY-based dataset on marriage and employment. Since this dataset is longitudinal, we should impute it in wide format. Stata allows us to reshape between wide and long formats easily.

```
. use marriage.dta, clear
. reshape wide interv  mar  educ emp enrol, i(id) j(year)
(note: j = 83 84 85 86 87 88 89 90 91 92 93 94)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                    72972   ->    6081
Number of variables                  11   ->      65
j variable (12 values)             year   ->   (dropped)
xij variables:
                                  interv   ->   interv83 interv84 ... interv94
                                     mar   ->   mar83 mar84 ... mar94
                                    educ   ->   educ83 educ84 ... educ94
```

```
                                  emp   ->   emp83 emp84 ... emp94
                                enrol   ->   enrol83 enrol84 ... enrol94
------------------------------------------------------------------------------
```

As discussed above, we can use interval option to limit the range of a variable during imputation:

```
gen paredll=pared
gen paredul=pared
replace paredll=0 if pared==.
replace paredul=20 if pared==.

ice mar* educ* emp* enrol*  birthdate m.race parpres pared paredll paredul,
saving(L:\socy7708\marriage_imputed.dta, replace) m(5) interval(pared: paredll
paredul) genmiss(mv_) dryrun
```

Obtaining Estimates after MI

To obtain final estimates of the parameters of interest and their standard errors, one would fit a model in each imputation and carry out the appropriate post-MI averaging procedure on the results. In Stata, you can do that using mi estimate -- you would merge all the imputed datasets into a single file for that, and make sure each imputation is appropriately marked in the _mj variable. You would also have to do mi import to convert your data from MICE format into the format used by MI commands (see help mi import).

First, if we generated our imputed data using the separate ICE module, we have to do mi import to convert the data from MICE format into the format used by MI commands. Let's do it for this longitudinal example:

```
. use marriage_imputed.dta, clear

. tab _mj

 imputation |
     number |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      6,081       16.67       16.67
          1 |      6,081       16.67       33.33
          2 |      6,081       16.67       50.00
          3 |      6,081       16.67       66.67
          4 |      6,081       16.67       83.33
          5 |      6,081       16.67      100.00
------------+-----------------------------------
      Total |     36,486      100.00

. mi import ice

. tab  _mi_m

      _mi_m |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      6,081       16.67       16.67
          1 |      6,081       16.67       33.33
          2 |      6,081       16.67       50.00
          3 |      6,081       16.67       66.67
          4 |      6,081       16.67       83.33
          5 |      6,081       16.67      100.00
------------+-----------------------------------
```

```
      Total |     36,486      100.00

. tab  _mi_miss

   _mi_miss |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      6,081      100.00      100.00
------------+-----------------------------------
      Total |      6,081      100.00

. sum  _mi_id

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      _mi_id |     36486        3041    1755.458          1       6081

. mi reshape long mar educ emp enrol mv_interv mv_mar mv_educ mv_emp mv_enrol,
>  i(id) j(year)

reshaping m=0 data ...
(note: j = 83 84 85 86 87 88 89 90 91 92 93 94)
(note: mv_interv83 not found)
(note: mv_interv84 not found)
(note: mv_interv85 not found)
(note: mv_interv86 not found)
(note: mv_interv87 not found)
(note: mv_interv88 not found)
(note: mv_interv89 not found)
(note: mv_interv90 not found)
(note: mv_interv91 not found)
(note: mv_interv92 not found)
(note: mv_interv93 not found)
(note: mv_interv94 not found)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                     6081   ->   72972
Number of variables                 125   ->      39
j variable (12 values)                   ->   year
xij variables:
                  mar83 mar84 ... mar94   ->   mar
               educ83 educ84 ... educ94   ->   educ
                  emp83 emp84 ... emp94   ->   emp
            enrol83 enrol84 ... enrol94   ->   enrol
mv_interv83 mv_interv84 ... mv_interv94   ->   mv_interv
         mv_mar83 mv_mar84 ... mv_mar94   ->   mv_mar
      mv_educ83 mv_educ84 ... mv_educ94   ->   mv_educ
         mv_emp83 mv_emp84 ... mv_emp94   ->   mv_emp
   mv_enrol83 mv_enrol84 ... mv_enrol94   ->   mv_enrol
-----------------------------------------------------------------------------

reshaping m=1 data ...

reshaping m=2 data ...

reshaping m=3 data ...

reshaping m=4 data ...

reshaping m=5 data ...

assembling results ...
```

```
. tab _mi_m

      _mi_m |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     72,972       16.67       16.67
          1 |     72,972       16.67       33.33
          2 |     72,972       16.67       50.00
          3 |     72,972       16.67       66.67
          4 |     72,972       16.67       83.33
          5 |     72,972       16.67      100.00
------------+-----------------------------------
      Total |    437,832      100.00
```

Here, we will use educ as our dependent variable; therefore, we might have to delete its imputed values (using MID strategy):

```
. tab mv_educ _mi_m, m

           |                    _mi_m
   mv_educ |         0          1          2          3 |     Total
-----------+--------------------------------------------+----------
         0 |         0     65,148     65,148     65,148 |   325,740
         1 |         0      7,824      7,824      7,824 |    39,120
         . |    72,972          0          0          0 |    72,972
-----------+--------------------------------------------+----------
     Total |    72,972     72,972     72,972     72,972 |   437,832


           |         _mi_m
   mv_educ |         4          5 |     Total
-----------+----------------------+----------
         0 |    65,148     65,148 |   325,740
         1 |     7,824      7,824 |    39,120
         . |         0          0 |    72,972
-----------+----------------------+----------
     Total |    72,972     72,972 |   437,832

. gen educ_dv=educ
(7,824 missing values generated)

. replace educ_dv=. if mv_educ==1
(39,120 real changes made, 39,120 to missing)

. mi xtset, clear

. mi estimate: mixed educ_dv mar  _Irace_2 _Irace_3 emp enrol birthdate parpres pared
|| id:

Multiple-imputation estimates                   Imputations       =          5
Mixed-effects ML regression                     Number of obs     =     65,148

Group variable: id                              Number of groups  =      6,081
                                                Obs per group:
                                                              min =          6
                                                              avg =       10.7
                                                              max =         12
                                                Average RVI       =     0.0498
                                                Largest FMI       =     0.1604
DF adjustment:   Large sample                   DF:     min       =     175.58
                                                        avg       =  38,966.95
                                                        max       = 275,303.36
Model F test:       Equal FMI                   F(   8, 7015.3)   =     913.50
```

```
                                                Prob > F          =     0.0000

------------------------------------------------------------------------------
     educ_dv |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mar |   .2167192   .0067137    32.28   0.000     .2034692    .2299692
    _Irace_2 |   .3792421    .062897     6.03   0.000     .2559484    .5025359
    _Irace_3 |   .2628567   .0776357     3.39   0.001     .1106915     .415022
         emp |   .0621397    .006392     9.72   0.000     .0496095    .0746699
       enrol |   -.612483   .0103154   -59.38   0.000    -.6327994   -.5921667
   birthdate |  -.0001555   .0000309    -5.03   0.000    -.0002161   -.0000948
     parpres |   .0274338   .0025974    10.56   0.000     .0223303    .0325373
       pared |   .2744962   .0096453    28.46   0.000     .2555902    .2934022
       _cons |     8.5588   .1173961    72.91   0.000     8.328587    8.789013
------------------------------------------------------------------------------
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Identity                 |
                  sd(_cons) |   1.957336   .0180025      1.922367    1.992942
-----------------------------+------------------------------------------------
               sd(Residual) |   .5297213   .0015621      .5266679    .5327925
------------------------------------------------------------------------------
```

Useful options for mi estimate

It might be helpful to be aware of a few options that can be used within mi estimate: prefix.

- nimputations(#) -- specify number of imputations to use; default is to use all in the file imputations(numlist) -- specify which imputations to use
- esampvaryok -- allow estimation when estimation sample varies across imputations
- cmdok -- allow estimation when estimation command is not supported by mi estimate
- post -- post estimated coefficients and VCE to e(b) and e(V)
- eform_option -- display coefficients table in exponentiated form; specific options depend on the command used:
    - eform exponentiated coefficient, string is exp(b)
    - hr hazard ratio, string is Haz. Ratio
    - shr subhazard ratio, string is SHR
    - irr incidence-rate ratio, string is IRR
    - or odds ratio, string is Odds Ratio
    - rrr relative-risk ratio, string is RRR

If you have to use some command not supported by mi estimate and you cannot resort to cmdok option for some reason (e.g., different software), you can estimate the model separately for each dataset, e.g.:

```
. for num 1/5: mixed educ_dv emp enrol mar birthdate parpres pared _Irace_2 _Irace_3
if  _mi_m==X || id:
```

The coefficients would be simple averages of coefficients from separate regression models, and the standard errors can be calculated using Rubin's (1987) formula:

$$\sqrt{\frac{1}{M}\sum_k s_k^2 + \left(1+\frac{1}{M}\right)\left(\frac{1}{M-1}\right)\sum_k (b_k - \bar{b})^2}$$

where b_k is the estimated regression coefficient in the imputed dataset k of the M imputed datasets, andt s_k is its estimated standard error. b bar is the average of coefficients across M imputations.

Essentially, we calculate the average squared standard error and then add to it the variance of coefficient estimates (multiplied by a correction factor 1 + 1/M).

Note that it is better to do exploratory runs and diagnostics on a single imputation--that would be much faster. Therefore, in addition to the main set of imputations, it is a good idea to separately generate one more imputation to be used in preliminary runs. But the results could end up being different.

Brief note: Methods for nonignorable missing data

All the methods of missing data handling considered above require that the data meet the MAR assumption. There are circumstances, however, when cases are considered missing due to non-ignorable causes. In such instances the investigator may want to consider the use of a selection model or a pattern-mixture model.

1. Selection models.
Social researchers have traditionally dealt with NMAR data by using selection models. In a selection model, you simultaneously model Y and the probability that Y is missing. In Stata, these models are implemented in heckman and heckprob for cross-sectional data, and xtheckman for longitudinal data. For an example of the latter, see:
https://www.stata.com/new-in-stata/xtheckman/

2. Pattern mixture models.
An alternative to selection models is multiple imputation with pattern mixture. In this approach, you perform multiple imputations under a variety of assumptions about the missing data mechanism.

Pattern-mixture models categorize the different patterns of missing values in a dataset into a predictor variable, and this predictor variable is incorporated into the statistical model of interest. The investigator can then determine if the missing data pattern has predictive power in the model, either by itself (a main effect) or in conjunction with another predictor (an interaction effect).

In ordinary multiple imputation, you assume that those people who report their weights are similar to those who don't. In a pattern-mixture model, you may assume that people who don't report their weights are an average of 20 pounds heavier. This is of course an arbitrary

assumption; the idea of pattern mixture is to try out a variety of plausible assumptions and see how much they affect your results.

Although pattern mixture is more natural, flexible, and interpretable approach, it appears that social researchers more often use selection models – partly because of tradition, partly because they are easier to use. Pattern mixture models can be implemented in SAS using PROC MI or PROC MIXED, but still, this requires some custom programming. Also, if the number of missing data patterns and the number of variables with missing data are large relative to the number of cases in the analysis, the model may not converge due to insufficient data.

In Stata, you could look at twofold command:
https://www.stata.com/meeting/uk16/slides/welch_uk16.pdf

To approximate pattern-mixture modeling, researchers also use regular MI but then modify multiply-imputed data to reflect possible departures from the MAR assumption. This process involves the following steps:

1. Use MI to impute the missing values under an MAR assumption.
2. Modify the imputed data to reflect a range of plausible MNAR scenarios, for example, by multiplying the imputed values by some constant c, or by adding a fixed amount delta (this is called "delta adjustment approach; for an example, see pp. 184-186 of van Buuren, S (2012), Flexible imputation of missing data, CRC Press).
3. Analyze the resulting dataset as one would a usual MI dataset, fitting the analysis model to each imputed dataset and combining the results using Rubin's rules.

With this approach, it is possible to do sensitivity testing by using a range of c or delta values. Also see:

Carpenter JR, Kenward MG, White IR. 2007. "Sensitivity analysis after multiple imputation under missing at random: a weighting approach." Statistical Methods in Medical Research, 16: 259-275.

Baptiste Leurent, Manuel Gomes, Rita Faria, Stephen Morris, Richard Grieve & James R. Carpenter. 2018. Sensitivity Analysis for Not-at-Random Missing Data in Trial-Based Cost-Effectiveness Analysis: A Tutorial. PharmacoEconomics, 36, 889–901.

Dealing with Attrition

Since attrition also results in missing data, it also can be either MCAR, MAR, or NMAR; and similarly to regular missing data, it is easier to deal with MCAR or MAR types of attrition. Researchers often do sensitivity analyses by imputing missing data for cases lost to follow-up and compare the results; they case also use weighing techniques where cases that are in the dataset are used to represent cases lost to follow-up using weights. As simulations show, if data are MCAR or MAR, the results without imputation or weights are pretty much as good as those with imputation and weights; they are also good if attrition is related to baseline values of outcome variable but not the follow-up value in addition to that. If it's NMAR with regard to

both baseline and follow-up, there is bias with all approaches, especially if attrition rates are higher (above 25%).

See:

Kristman, Vicky, Michael Manno, and Pierre Côté. 2005. "Methods to Account for Attrition in Longitudinal Data: Do They Work? A Simulation Study." European Journal of Epidemiology 20(8):657-62).

Gustavson, K., von Soest, T., Karevold, E. et al. 2012. Attrition and Generalizability in Longitudinal Studies: Findings from a 15-year Population-based Study and a Monte Carlo Simulation Study. BMC Public Health 12: 918-28.

So in cases when attrition may be directly related to your outcomes of interest, Heckman models or pattern mixture modeling could be useful.

Sometimes, studies also introduce refreshment samples—new, randomly sampled respondents who participate in the survey at the same time as a regular wave of the panel; these samples offer information that can be used to diagnose and adjust for bias due to attrition. See:

Yiting Deng, D. Sunshine Hillygus, Jerome P. Reiter, Yajuan Si and Siyu Zheng. 2013. Handling Attrition in Longitudinal Studies: The Case for Refreshment Samples. Statistical Science 28(2), pp. 238-256.

Also see this article for a discussion of different approaches for sensitivity analyses to deal with both NMAR nonparticipation and deaths:

Biering K, Hjollund NH, Frydenberg M. 2015. Using Multiple Imputation to Deal with Missing Data and Attrition in Longitudinal Studies with Repeated Measures of Patient-reported Outcomes. Clinical Epidemiology 7:91-106.