SC708: Hierarchical Linear Modeling
Instructor: Natasha Sarkisian

Missing data, Part 2

As we discussed last time, there are two algorithms we might want to follow.

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
3. Impute level 2 including aggregates (especially DV aggregate)
4. Merge with imputed level 1 data

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 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 2, 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 "C:\Users\sarkisin\Documents\SC708\nels_science.dta"
file C:\Users\sarkisin\Documents\SC708\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.

Also, because of so many dummies, to run the following command, I need to use Stata SE rather than Stata IC. Stata SE is available at apps.bc.edu – you would want to specify larger matsize:

```
. set matsize 1200

. use "C:\Users\sarkisin\nels_science.dta", clear

ice science female spoken pared m.race5 i.cluster, cmd(spoken: ologit)
saving("C:\Users\sarkisin\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, _mj:

```
.use "C:\Users\sarkisin\nels_science_level1imp.dta", clear
. merge m:1 sch_id _mj using "C:\Users\sarkisin\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 "C:\Users\sarkisin\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 "C:\Users\sarkisin\Documents\Teaching Grad
Statistics\SC708\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 "C:\Users\sarkisin\nels_imputed_1.dta"
```

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

```
use "C:\Users\sarkisin\nels_imputed.dta", clear
for num 1/5: preserve \ keep if _mj==X \ merge 1:m sch_id using
"C:\Users\sarkisin\Documents\SC708\nels_science.dta" \ tab _merge \ drop _merge _mi
_mj \ save "C:\Users\sarkisin\nels_imputed_X.dta" \ restore
```

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

```
ice  public pct_esl morale sciencem femalem blackm latinom asianm nativem
spokenm paredm science female spoken pared m.race5 cmd(spoken morale: ologit)
saving("C:\Users\sarkisin\nels_science_imp1.dta", replace) m(1)
```

We would repeat this process 5 times, and then merge all 5 imputations together.

```
. use "C:\Users\sarkisin\nels_imputed_1.dta", clear
. drop if _mj==0
. for num 2/5: use "C:\Users\sarkisin\nels_imputed_X.dta"      \ drop if _mj==0 \
replace _mj==X \ save  "C:\Users\sarkisin\nels_imputed_X.dta", replace
. use "C:\Users\sarkisin\nels_imputed_1.dta", clear
. for num 2/5: append using "C:\Users\sarkisin\nels_imputed_X.dta"
. tab _mj
. save use "C:\Users\sarkisin\nels_imputed_merged.dta", replace
```

Longitudinal Example

To deal with the longitudinal version of nested data (which is what we will discuss next in terms of analyses), 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.

```
. 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
-------------------------------------------------------------------------------
```

Before we run actual multiple imputation using ICE, we will do a dry run – check that everything looks correct, without generating actual datasets:

```
. ice mar* educ* emp* enrol*  birthdate m.race parpres pared,
saving(marriage_imputed.dta, replace) m(5) dryrun

=> xi: ice mar83 mar84 mar85 mar86 mar87 mar88 mar89 mar90 mar91 mar92 mar93 mar94
educ83 educ84 educ8
> 5 educ86 educ87 educ88 educ89 educ90 educ91 educ92 educ93 educ94 emp83 emp84 emp85
emp86 emp87 emp88
>  emp89 emp90 emp91 emp92 emp93 emp94 enrol83 enrol84 enrol85 enrol86 enrol87 enrol88
enrol89 enrol90
>  enrol91 enrol92 enrol93 enrol94 birthdate race i.race parpres pared,
cmd(race:mlogit) substitute(ra
> ce:i.race) saving(C:\Users\sarkisin\marriage_imputed.dta, replace) m(5) dryrun

i.race            _Irace_1-3          (naturally coded; _Irace_1 omitted)

   #missing |
     values |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         43        0.71        0.71
          1 |        228        3.75        4.46
          2 |        494        8.12       12.58
          3 |        821       13.50       26.08
          4 |      1,029       16.92       43.00
          5 |      1,015       16.69       59.69
          6 |        833       13.70       73.39
          7 |        617       10.15       83.54
          8 |        459        7.55       91.09
          9 |        254        4.18       95.26
         10 |        137        2.25       97.52
         11 |         77        1.27       98.78
         12 |         31        0.51       99.29
         13 |         29        0.48       99.77
         14 |         12        0.20       99.97
         15 |          2        0.03      100.00
------------+-----------------------------------
      Total |      6,081      100.00

   Variable | Command | Prediction equation
------------+---------+-------------------------------------------------------
      mar83 | logit   | mar84 mar85 mar86 mar87 mar88 mar89 mar90 mar91 mar92
            |         | mar93 mar94 educ83 educ84 educ85 educ86 educ87 educ88
            |         | educ89 educ90 educ91 educ92 educ93 educ94 emp83 emp84
            |         | emp85 emp86 emp87 emp88 emp89 emp90 emp91 emp92 emp93
            |         | emp94 enrol83 enrol84 enrol85 enrol86 enrol87 enrol88
            |         | enrol89 enrol90 enrol91 enrol92 enrol93 enrol94
            |         | birthdate _Irace_2 _Irace_3 parpres pared
[output omitted]

   _Irace_2 |         | [Passively imputed from (race==2)]
   _Irace_3 |         | [Passively imputed from (race==3)]
    parpres | regress | mar83 mar84 mar85 mar86 mar87 mar88 mar89 mar90 mar91
```

```
            |         | mar92 mar93 mar94 educ83 educ84 educ85 educ86 educ87
            |         | educ88 educ89 educ90 educ91 educ92 educ93 educ94 emp83
            |         | emp84 emp85 emp86 emp87 emp88 emp89 emp90 emp91 emp92
            |         | emp93 emp94 enrol83 enrol84 enrol85 enrol86 enrol87
            |         | enrol88 enrol89 enrol90 enrol91 enrol92 enrol93
            |         | enrol94 birthdate _Irace_2 _Irace_3 pared
      pared | regress | mar83 mar84 mar85 mar86 mar87 mar88 mar89 mar90 mar91
            |         | mar92 mar93 mar94 educ83 educ84 educ85 educ86 educ87
            |         | educ88 educ89 educ90 educ91 educ92 educ93 educ94 emp83
            |         | emp84 emp85 emp86 emp87 emp88 emp89 emp90 emp91 emp92
            |         | emp93 emp94 enrol83 enrol84 enrol85 enrol86 enrol87
            |         | enrol88 enrol89 enrol90 enrol91 enrol92 enrol93
            |         | enrol94 birthdate _Irace_2 _Irace_3 parpres
-------------------------------------------------------------------------------

End of dry run. No imputations were done, no files were created.
```

Limiting variable values to an interval when using ice:

You 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(C:\Users\sarkisin\marriage_imputed.dta, replace) m(5) interval(pared: paredll
paredul)
```

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 interv mar educ emp enrol, i(id) j(year)

reshaping m=0 data ...
(note: j = 83 84 85 86 87 88 89 90 91 92 93 94)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                     6081   ->   72972
Number of variables                  67   ->      13
j variable (12 values)                    ->   year
xij variables:
         interv83 interv84 ... interv94   ->   interv
                  mar83 mar84 ... mar94   ->   mar
                educ83 educ84 ... educ94   ->   educ
                   emp83 emp84 ... emp94   ->   emp
             enrol83 enrol84 ... enrol94   ->   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). For that, it would have been useful to use genmiss to create a missing value indicator for educ, but since we did not, we will identify those using the following process:

```
. gen educm=(educ==.)

. tab educm

      educm |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    430,008       98.21       98.21
          1 |      7,824        1.79      100.00
------------+-----------------------------------
      Total |    437,832      100.00

. tab educm if  _mi_m==0

      educm |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     65,148       89.28       89.28
          1 |      7,824       10.72      100.00
------------+-----------------------------------
      Total |     72,972      100.00

. bysort id year: egen educm_all=total(educm)

. tab educm_all

  educm_all |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    390,888       89.28       89.28
          1 |     46,944       10.72      100.00
------------+-----------------------------------
      Total |    437,832      100.00


. for num 0/5: tab educm_all if  _mi_m==X

->  tab educm_all if _mi_m==0

  educm_all |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     65,148       89.28       89.28
          1 |      7,824       10.72      100.00
------------+-----------------------------------
      Total |     72,972      100.00

->  tab educm_all if _mi_m==1

  educm_all |      Freq.     Percent        Cum.
------------+-----------------------------------
```

```
          0 |     65,148       89.28       89.28
          1 |      7,824       10.72      100.00
------------+-----------------------------------
      Total |     72,972      100.00

->  tab educm_all if _mi_m==2

  educm_all |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     65,148       89.28       89.28
          1 |      7,824       10.72      100.00
------------+-----------------------------------
      Total |     72,972      100.00

->  tab educm_all if _mi_m==3

  educm_all |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     65,148       89.28       89.28
          1 |      7,824       10.72      100.00
------------+-----------------------------------
      Total |     72,972      100.00

->  tab educm_all if _mi_m==4

  educm_all |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     65,148       89.28       89.28
          1 |      7,824       10.72      100.00
------------+-----------------------------------
      Total |     72,972      100.00

->  tab educm_all if _mi_m==5

  educm_all |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     65,148       89.28       89.28
          1 |      7,824       10.72      100.00
------------+-----------------------------------
      Total |     72,972      100.00

. gen educ_dv=educ
(7824 missing values generated)

. replace educ_dv=. if educm_all==1
(39120 real changes made, 39120 to missing)

. mi estimate: xtreg educ_dv mar  _Irace_2 _Irace_3 emp enrol birthdate parpres pared,
i(id)

Multiple-imputation estimates                   Imputations        =         5
Random-effects GLS regression                   Number of obs      =     65148

Group variable: id                              Number of groups   =      6081
                                                Obs per group: min =         6
                                                               avg =      10.7
                                                               max =        12

                                                Average RVI        =    0.0528
DF adjustment:   Large sample                   DF:     min        =    206.54
                                                        avg        =  28808.67
                                                        max        = 186339.77
Model F test:       Equal FMI                   F(   8, 8660.2)    =    952.02
```

```
Within VCE type: Conventional                   Prob > F          =    0.0000

------------------------------------------------------------------------------
     educ_dv |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mar |   .2166079   .0067589    32.05   0.000     .2032827    .2299331
    _Irace_2 |   .3797767   .0553935     6.86   0.000      .271202    .4883514
    _Irace_3 |   .2705604   .0689158     3.93   0.000      .135483    .4056379
         emp |   .0662441   .0065246    10.15   0.000     .0534498    .0790384
       enrol |  -.6087065    .009938   -61.25   0.000    -.6281906   -.5892223
   birthdate |  -.0001589   .0000274    -5.80   0.000    -.0002126   -.0001052
     parpres |   .0268971   .0023521    11.44   0.000     .0222631    .0315311
       pared |   .2757336    .008757    31.49   0.000     .2585457    .2929215
       _cons |   8.561667   .1027696    83.31   0.000     8.360216    8.763118
-------------+----------------------------------------------------------------
     sigma_u |  1.7055971
     sigma_e |  .52961769
         rho |  .91205833   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Note: sigma_u and sigma_e are combined in the original metric.
```

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: xtreg  educ_dv emp enrol mar birthdate parpres pared _Irace_2 _Irace_3
if  _mi_m==X, i(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.

Using MI datasets in HLM

But we will use HLM. For HLM, we need to use our imputed files as separate files (which is what we typically get when using second algorithm; with the first algorithm, you can separate them using "keep if _mj==1" etc command).

We will then import the first one into MDM, generate .mdmt file, and then resave .mdmt file 9 more times and change files names. After that, we can just open them one by one, run, and generate .mdm files for each.

Then we start with one dataset, select Other Settings→Estimation Settings→Multiple imputation, then select up to 10 datasets.

After that, estimate the model as you normally would. Note that it is better to do exploratory runs on a single imputation--that would be much faster. 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. These models are implemented in heckman and heckprob in Stata. Unfortunately, a number of practical difficulties are often encountered in estimating selection models, and these models are not implemented in HLM software. You can, however, estimate two step selection models – in the first step, selection probability is predicted, and in the second step, that selection probability is used to predict Y.

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 an 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.