University of Zurich[UZH]

Epidemiology, Biostatistics
and Prevention Institute,
Epidemiology Department
+41 44 634 46 34
www.ebpi.uzh.ch

Multiple Imputation using STATA

Sarah R Haile (sarah.haile@uzh.ch)

Version 1.1 of March 3, 2016

Contents

Multiple imputation (MI) is a common statistical method used to analyze datasets where some values are missing. In this document we describe multiple imputation briefly, and show how to perform the analysis in STATA. The main and extended examples show a dataset where the outcome is binary, and logistic regression is used. After that, we show shorter examples for linear regression and Cox proportional hazards regression.

For more information about multiple imputation, you might find these other references helpful:

1. UW-Madison (2013) (http://www.ssc.wisc.edu/sscc/pubs/stata_mi_intro.htm)
2. Stata Corp (2013) (http://www.stata.com/manuals13/mi.pdf)
3. White et al. (2011) (http://doi.org/10.1002/sim.4067)
4. Royston and White (2011) (http://www.jstatsoft.org/v45/i04), and
5. White and Royston (2009) (http://doi.org/10.1002/sim.3618).

Papers 3 and 4 specifically deal with MI using STATA, however they use an older version of the code. Nevertheless they provide a good overview. Finally, paper 5 discusses multiple imputation when survival data are present.

1 Basic steps

The basic steps in STATA are as follows:

`mi set wide` Declare your data to be `mi`, and tell STATA which "style" to use.

`mi register imputed` Tell STATA which variable need to be imputed (also: `mi passive`)

`mi impute chained` Impute all missing variables in one step.

`mi estimate` Analyze each of the imputed datasets and combine the results.

2 Complete Cases analysis

The `mheart5` dataset has 6 variables, two of which have missing observations: `age` (n = 12), and `bmi` (n = 28).

```
. webuse mheart5
(Fictional heart attack data)

. describe

Contains data from http://www.stata-press.com/data/r13/mheart5.dta
  obs:           154                          Fictional heart attack data
 vars:             6                          19 Jun 2012 10:50
 size:         1,848
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
```

```
-------------------------------------------------------------------------------
attack          byte    %9.0g                Outcome (heart attack)
smokes          byte    %9.0g                Current smoker
age             float   %9.0g                Age, in years
bmi             float   %9.0g                Body Mass Index, kg/m^2
female          byte    %9.0g                Gender
hsgrad          byte    %9.0g                High school graduate
-------------------------------------------------------------------------------
Sorted by:

. misstable summ, all
                                                               Obs<.
                                                +------------------------------
               |                                | Unique
      Variable |     Obs=.     Obs>.     Obs<.  | values        Min         Max
  -------------+--------------------------------+------------------------------
        attack |                           154  |      2          0           1
        smokes |                           154  |      2          0           1
           age |        12                 142  |    142   20.73613    83.78423
           bmi |        28                 126  |    126   17.22643    38.24214
        female |                           154  |      2          0           1
        hsgrad |                           154  |      2          0           1
  -----------------------------------------------------------------------------
```

First, let's do the usual (complete cases) analysis of the association of our 5 predictors on the *probability of a heart attack* (`attack`), using only the complete cases.

```
. logistic attack smokes age bmi hsgrad female

Logistic regression                               Number of obs   =        126
                                                  LR chi2(5)      =      22.56
                                                  Prob > chi2     =     0.0004
Log likelihood = -75.802314                       Pseudo R2       =     0.1295

------------------------------------------------------------------------------
      attack | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   4.545809   1.844206     3.73   0.000     2.052505    10.06788
         age |   1.030523   .0181677     1.71   0.088     .9955232    1.066754
         bmi |   1.104937   .0553865     1.99   0.047     1.001543    1.219004
      hsgrad |   1.381645   .6161839     0.72   0.469      .576473    3.311418
      female |   1.321851   .6152168     0.60   0.549     .5309038    3.291163
       _cons |   .0050166   .0090925    -2.92   0.003     .0001438    .1750652
------------------------------------------------------------------------------
```

We would like to see if our results are affected by the missing values in `age` and `bmi`.

3 Logistic regression

3.1 A basic example

Now, we use MI to guess what the missing `age` and `bmi` "might have been", and then use our imputed datasets to estimate the regression coefficients again.

3.1.1 Declare your data to be `mi` using `mi set`.

```
mi set wide
```

We used `mi set wide` below to set the data as `mi`, but there are other "styles" (see `help mi styles`) we could have used. The `wide` style produces new imputed variables as we would expect to see. However, it is not the most efficient of the 4 styles, and may not work very well for very large datasets. If you have already set your `mi` data as `wide` and want to convert them, try `mi convert mlong, clear` or `mi convert flongsep, clear`.

```
mi describe
mi misstable summarize
tab _mi_miss
* tab _mi_miss _mi_m
```

At any point, you can use `mi describe` to see what variables have been imputed or `mi misstable summarize` to examine which variables have missing observations. Among the variables that are created is `_mi_miss`, which tells you whether the observation has missing values or not, and, if you are using `style mlong` or `flong`, `_mi_m`, which indicates which imputed dataset an observation belongs to, and `_mi_id`, indicating the subject.

3.1.2 Tell `mi` which variables we want to impute

Tell `mi` which variables have missing values that we want to impute. We'll look at the other main option (`passive`) in a later example. For now, just tell STATA which variables with missing values you want to include in your regression models.

```
mi register imputed age bmi
```

3.1.3 Impute the missing observations using `mi impute`.

There are various methods used to impute missing values for multiple variables at once. Unless you only have 1 continuous predictor with missing values, we recommend using `mi impute chained` ("multivariate imputation using chained equations") and specifying which regression method you want to use for each variable with missing values. Here we use the `regress` command to impute 20 additional datasets for both `age` and `bmi`, using all other variables (including the outcome!) as predictors in the imputation procedure (see Section 8.1). We also used a seed here (with option `rseed(#)`, or alternately with the command `set seed #`) to ensure that we will get the same results any time we run the code. For more on which regression methods to use or how many datasets to impute see Sections 8.3 and 8.4.

```
mi impute chained (regress) age bmi = attack smokes hsgrad female,
  add(20) rseed(158720)
```

3.1.4 Run our regression models on imputed data, and combine the results using `mi estimate`.

The MI estimation procedure `mi estimate` by default only prints the estimated coefficients. If you want odds ratios, use the option `or` (for other commands, the usual `eform`, `hr`, `rrr`, etc are available), and to save the estimates as `miest` use `saving(miest, replace)`. For more details on the variability seen in the analysis of the imputed datasets, use the `vartable` option. See Section 7) if are interested in making predictions or testing coefficients after `mi estimate`, in which case be sure to use the `saving()` option.

```
mi estimate, or saving(miest, replace): logistic attack smokes age bmi hsgrad female
mi predict xbmi using miest
mi xeq: generate predprobmi = invlogit(xbmi)
mi xeq: generate ORmi = exp(xbmi)
```

```
. mi set wide

. mi describe

  Style:  wide
          last mi update 06aug2015 10:48:57, 0 seconds ago

  Obs.:   complete          154
          incomplete          0  (M = 0 imputations)
          ---------------------
          total             154

  Vars.:  imputed:  0

          passive:  0

          regular:  0

          system:   1; _mi_miss

         (there are 7 unregistered variables)

. mi misstable summarize
                                                               Obs<.
                                                +------------------------------
                |                               | Unique
       Variable |     Obs=.     Obs>.     Obs<. | values        Min         Max
  -------------+--------------------------------+------------------------------
            age |        12                 142 |    142   20.73613    83.78423
            bmi |        28                 126 |    126   17.22643    38.24214
     predprobcc |        28                 126 |    126   .1455459    .9103615
  -----------------------------------------------------------------------------

. tab _mi_miss
```

```
   _mi_miss |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        154      100.00      100.00
------------+-----------------------------------
      Total |        154      100.00

.
. mi register imputed age bmi

. mi describe

  Style:  wide
          last mi update 06aug2015 10:48:57, 0 seconds ago

  Obs.:   complete          126
          incomplete         28  (M = 0 imputations)
          ---------------------
          total             154

  Vars.:  imputed:  2; age(12) bmi(28)

          passive:  0

          regular:  0

          system:   1; _mi_miss

         (there are 5 unregistered variables)

.
. set seed 29390

. mi impute chained (regress) age bmi = attack smokes hsgrad female, add(20)
note: missing-value pattern is monotone; no iteration performed

Conditional models (monotone):
               age: regress age attack smokes hsgrad female
               bmi: regress bmi age attack smokes hsgrad female

Performing chained iterations ...

Multivariate imputation                     Imputations =       20
Chained equations                                 added =       20
Imputed: m=1 through m=20                       updated =        0

Initialization: monotone                     Iterations =        0
```

```
                                                   burn-in =          0

              age: linear regression
              bmi: linear regression

------------------------------------------------------------------
                  |               Observations per m
                  |----------------------------------------------
         Variable |   Complete   Incomplete   Imputed |     Total
------------------+-----------------------------------+----------
              age |        142           12        12 |       154
              bmi |        126           28        28 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. mi describe

  Style:  wide
          last mi update 06aug2015 10:48:57, 0 seconds ago

  Obs.:   complete          126
          incomplete         28  (M = 20 imputations)
          ---------------------
          total             154

  Vars.:  imputed:  2; age(12) bmi(28)

          passive:  0

          regular:  0

          system:   1; _mi_miss

         (there are 5 unregistered variables)

.
. mi estimate, or saving(miest, replace): logistic attack smokes age bmi hsgrad
>  female

Multiple-imputation estimates                   Imputations     =         20
Logistic regression                             Number of obs   =        154
                                                Average RVI     =     0.0779
                                                Largest FMI     =     0.2439
DF adjustment:   Large sample                   DF:     min     =     331.67
                                                        avg     =  150865.39
                                                        max     =  759562.93
```

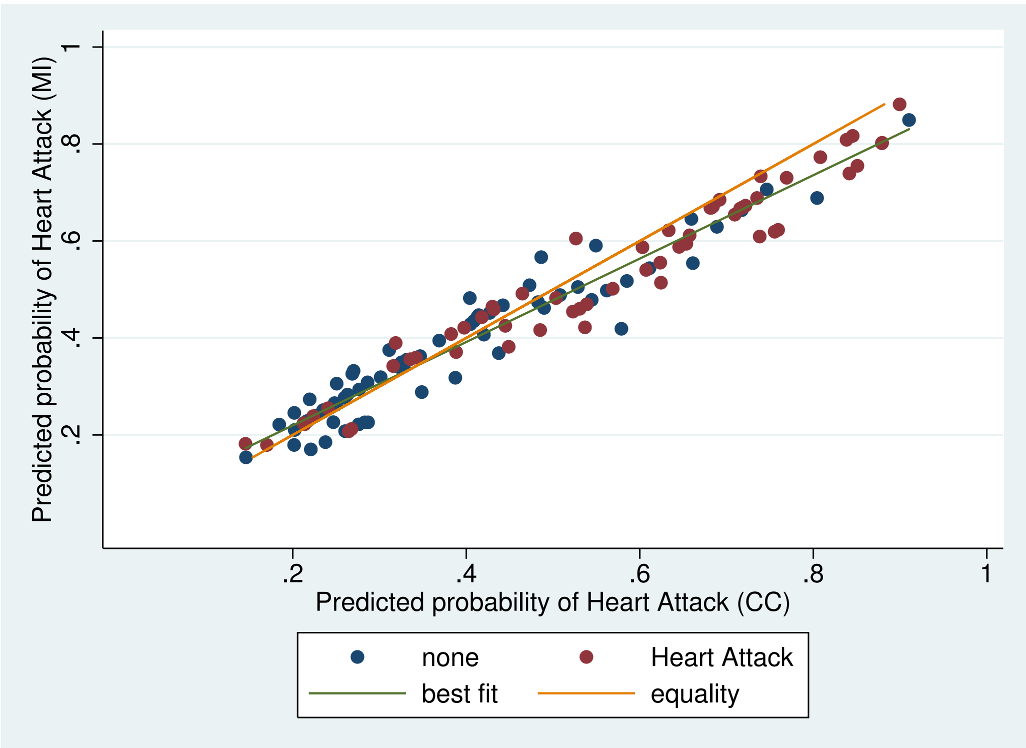
```
Model F test:       Equal FMI                    F(   5,12152.8) =      3.15
Within VCE type:          OIM                    Prob > F        =    0.0076

------------------------------------------------------------------------------
      attack | Odds Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   3.180032   1.134821     3.24   0.001     1.580028     6.40027
         age |   1.030013   .0167176     1.82   0.069     .9977382    1.063331
         bmi |   1.103508   .0563454     1.93   0.055     .9980537    1.220105
      hsgrad |   1.140957   .4595155     0.33   0.743     .5181403    2.512414
      female |   .9063112   .3739501    -0.24   0.812     .4037061    2.034648
       _cons |   .0071372   .0126159    -2.80   0.005     .0002214    .2301282
------------------------------------------------------------------------------

. mi predict xbmi using miest
(option xb assumed; linear prediction)

. quietly: mi xeq: generate predprobmi = invlogit(xbmi)

. quietly: mi xeq: generate ORmi = exp(xbmi)
```



We see in the graph that the predicted probabilities are quite close, but the MI probabilities are slight lower than those in the CC analysis.

3.2 Extended example

Now let's suppose that we want to consider age and bmi as categorical variables, and further that smoking status has some missing values. Here, we probably should impute age and bmi as continuous variables, and then convert the imputed variables to `age55` and `bmicut` using the `mi passive` command (also see Section 8.5). (Note that you will get an error if, for example, `age55` has been defined in your dataset before you define it with `mi passive`. Therefore, you should rename these variables, or wait to define them until you perform the `mi` analysis.)

When we impute the 3 variables, we need to also tell to use `logit` to impute values for `smokes`, rather than `regress`. We can do that by writing

```
mi impute chained (regress) age bmi (logit) smokes2 = attack hsgrad female, add(20)
```

```
. webuse mheart5
(Fictional heart attack data)

. gen smokes2 = smokes

. replace smokes2 = . if age > 60 & female==1
(15 real changes made, 15 to missing)

. tab smokes smokes2, miss

   Current |              smokes2
    smoker |         0          1          . |     Total
-----------+---------------------------------+----------
         0 |        84          0          6 |        90
         1 |         0         55          9 |        64
-----------+---------------------------------+----------
     Total |        84         55         15 |       1.
. mi set wide

. mi misstable summarize
                                                            Obs<.
                                                +------------------------------
               |                                | Unique
      Variable |     Obs=.     Obs>.     Obs<.  | values        Min         Max
  -------------+--------------------------------+------------------------------
           age |        12                 142  |    142   20.73613    83.78423
           bmi |        28                 126  |    126   17.22643    38.24214
       smokes2 |        15                 139  |      2          0           1
  -----------------------------------------------------------------------------

.
. mi register imputed age bmi smokes2

. mi passive: egen age55 = cut(age), at(20 55 85) label
m=0:
(12 missing values generated)

. mi passive: egen bmicut = cut(bmi), at(10 16 18.5 25 30 35 40) label
```

```
m=0:
(28 missing values generated)

.
. mi impute chained (regress) age bmi (logit) smokes2 = attack hsgrad female,
add(20)

Conditional models:
               age: regress age i.smokes2 bmi attack hsgrad female
           smokes2: logit smokes2 age bmi attack hsgrad female
               bmi: regress bmi age i.smokes2 attack hsgrad female

Performing chained iterations ...

Multivariate imputation                     Imputations =       20
Chained equations                                 added =       20
Imputed: m=1 through m=20                       updated =        0

Initialization: monotone                     Iterations =      200
                                                burn-in =       10

               age: linear regression
               bmi: linear regression
           smokes2: logistic regression

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
               age |        142           12        12 |       154
               bmi |        126           28        28 |       154
           smokes2 |        139           15        15 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. mi describe

  Style:  wide
          last mi update 06aug2015 10:49:16, 5 seconds ago

  Obs.:   complete          114
          incomplete         40  (M = 20 imputations)
          ---------------------
          total             154
```

```
  Vars.:  imputed:  3; age(12) bmi(28) smokes2(15)

          passive:  2; age55(12) bmicut(28)

          regular:  0

          system:   1; _mi_miss

         (there are 4 unregistered variables; attack smokes female hsgrad)


. mi estimate, or saving(miest2, replace): logistic attack smokes2 age55 bmicut
hsgrad female

Multiple-imputation estimates                    Imputations     =         20
Logistic regression                              Number of obs   =        126
                                                 Average RVI     =     0.0424
                                                 Largest FMI     =     0.0888
DF adjustment:   Large sample                    DF:     min     =    2451.89
                                                         avg     =   57738.66
                                                         max     =  154114.16
Model F test:       Equal FMI                    F(   5,46357.8) =       3.25
Within VCE type:          OIM                    Prob > F        =     0.0062

------------------------------------------------------------------------------
      attack | Odds Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     smokes2 |   4.876346   2.134777     3.62   0.000     2.066663    11.50587
       age55 |   1.561686   .6232266     1.12   0.264     .7143059     3.41431
      bmicut |   1.542118   .3785125     1.76   0.078     .9532098    2.494864
      hsgrad |   1.526762   .6917138     0.93   0.350     .6282369    3.710387
      female |   1.573723   .7395976     0.96   0.335     .6262893    3.954407
       _cons |   .0795238   .0707164    -2.85   0.004     .0139169    .4544137
------------------------------------------------------------------------------
.
. mi predict xbmi2 using miest2
(option xb assumed; linear prediction)
(28 missing values generated)

. quietly: mi xeq: generate predprobmi2 = invlogit(xbmi2)
```

4 Linear regression

In this example, we examine how BMI is associated with the other covariates, again using the `mheart5` dataset. Here, we compare the results obtained when we impute `age` and `bmi` with the `regress` command, with those obtained using the `pmm` (predictive mean matching).

```
. webuse mheart5
(Fictional heart attack data)

. regress bmi female age smokes attack hsgrad

      Source |       SS       df       MS              Number of obs =     126
-------------+------------------------------           F(  5,   120) =    0.90
       Model |  73.1449691     5  14.6289938           Prob > F      =  0.4853
    Residual |  1956.28756   120  16.3023963           R-squared     =  0.0360
-------------+------------------------------           Adj R-squared = -0.0041
       Total |  2029.43253   125  16.2354602           Root MSE      =  4.0376

------------------------------------------------------------------------------
         bmi |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      female |   -.109108   .8372703    -0.13   0.897    -1.766845    1.548629
         age |  -.0240674   .0318764    -0.76   0.452    -.0871805    .0390457
      smokes |  -.2470361   .7820658    -0.32   0.753    -1.795472      1.3014
      attack |      1.545   .7775581     1.99   0.049     .0054888    3.084511
      hsgrad |  -.4092541   .8225038    -0.50   0.620    -2.037754    1.219246
       _cons |   26.31799   1.961478    13.42   0.000      22.4344    30.20158
------------------------------------------------------------------------------

. mi set wide

. mi register impute age bmi

. mi impute chained (regress) age bmi, add(20)
note: missing-value pattern is monotone; no iteration performed

Conditional models (monotone):
               age: regress age
               bmi: regress bmi age

Performing chained iterations ...

Multivariate imputation                     Imputations =       20
Chained equations                                 added =       20
Imputed: m=1 through m=20                       updated =        0

Initialization: monotone                     Iterations =        0
```

```
                                              burn-in =        0

            age: linear regression
            bmi: linear regression

-----------------------------------------------------------------
               |               Observations per m
               |----------------------------------------------
      Variable |   Complete   Incomplete   Imputed |     Total
---------------+-----------------------------------+----------
           age |        142           12        12 |       154
           bmi |        126           28        28 |       154
-----------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. mi estimate: regress bmi female age smokes attack hsgrad

Multiple-imputation estimates                   Imputations       =         20
Linear regression                               Number of obs     =        154
                                                Average RVI       =     0.2243
                                                Largest FMI       =     0.2791
                                                Complete DF       =        148
DF adjustment:   Small sample                   DF:     min       =      75.64
                                                        avg       =      98.57
                                                        max       =     118.80
Model F test:       Equal FMI                   F(   5,   136.4)  =       0.65
Within VCE type:          OLS                   Prob > F          =     0.6606

------------------------------------------------------------------------------
         bmi |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      female |   .0130089   .8064126     0.02   0.987    -1.583796    1.609813
         age |  -.0222235    .032021    -0.69   0.489    -.0857989    .0413518
      smokes |  -.1754602   .8069476    -0.22   0.828    -1.782759    1.431838
      attack |   1.305557   .7720149     1.69   0.094    -.2277497    2.838864
      hsgrad |  -.2516483   .8441186    -0.30   0.766    -1.927272    1.423976
       _cons |   26.20658   1.891125    13.86   0.000     22.46062    29.95254
------------------------------------------------------------------------------

.
. mi impute chained (pmm) age bmi, add(20) replace
note: missing-value pattern is monotone; no iteration performed

Conditional models (monotone):
            age: pmm age
            bmi: pmm bmi age
```

```
Performing chained iterations ...

Multivariate imputation                      Imputations =       40
Chained equations                                  added =       20
Imputed: m=1 through m=40                        updated =       20

Initialization: monotone                      Iterations =        0
                                                 burn-in =        0

               age: predictive mean matching
               bmi: predictive mean matching

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
               age |        142           12        12 |       154
               bmi |        126           28        28 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. mi estimate: regress bmi female age smokes attack hsgrad

Multiple-imputation estimates                   Imputations     =         40
Linear regression                               Number of obs   =        154
                                                Average RVI     =     0.0813
                                                Largest FMI     =     0.1095
                                                Complete DF     =        148
DF adjustment:   Small sample                   DF:     min     =     125.51
                                                        avg     =     132.08
                                                        max     =     138.33
Model F test:       Equal FMI                   F(   5,  145.1) =       0.96
Within VCE type:          OLS                   Prob > F        =     0.4452

------------------------------------------------------------------------------
         bmi |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      female |   -.166588   .7663968    -0.22   0.828    -1.681955    1.348779
         age |  -.0375009   .0295147    -1.27   0.206    -.0958795    .0208778
      smokes |   .1029344   .7134225     0.14   0.886    -1.308559    1.514428
      attack |   1.280091   .7186201     1.78   0.077    -.1420903    2.702273
      hsgrad |  -.0299699   .7815607    -0.04   0.969    -1.576108    1.516169
       _cons |    26.6825   1.805129    14.78   0.000     23.11263    30.25238
------------------------------------------------------------------------------
```

5 Cox proportional hazards regression

In the data we use here, the survival time and event indicator have already been set with `stset`. As this is already taken care of, we don't need to do this for our `mi` data. If, however, we had not set this information prior to starting our MI analysis with `mi set`, we would have to use the command `mi stset` instead of the usual `stset`. (The same is true for other set commands, like `xtset`.) Here we include two additional variables, `HT` and `_d`, in the imputation process. For more details see Section 8.2.

```
. webuse drugtr
(Patient Survival in Drug Trial)

. gen age2 = age

. replace age2 = . if (drug == 1 & age > 60)
(6 real changes made, 6 to missing)

. replace age2 = . if (drug == 0 & age < 50)
(3 real changes made, 3 to missing)

. describe

Contains data from http://www.stata-press.com/data/r13/drugtr.dta
  obs:            48                          Patient Survival in Drug Trial
 vars:             9                          3 Mar 2013 02:12
 size:           768
------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
studytime       int     %8.0g                 Months to death or end of exp.
died            int     %8.0g                 1 if patient died
drug            int     %8.0g                 Drug type (0=placebo)
age             int     %8.0g                 Patient's age at start of exp.
_st             byte    %8.0g
_d              byte    %8.0g
_t              byte    %10.0g
_t0             byte    %10.0g
age2            float   %9.0g
------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved

. misstable summ
                                                               Obs<.
                                                +------------------------------
               |                                | Unique
```

```
      Variable |     Obs=.     Obs>.     Obs<.  | values        Min          Max
  -------------+--------------------------------+------------------------------
          age2 |         9                  39  |     18         47           67
  -----------------------------------------------------------------------------

. * data set already has stset information

. stcox drug age2, nolog

         failure _d:  died
   analysis time _t:  studytime

Cox regression -- Breslow method for ties

No. of subjects =           39                     Number of obs    =        39
No. of failures =           25
Time at risk    =          621
                                                   LR chi2(2)       =     29.31
Log likelihood  =   -60.864773                     Prob > chi2      =    0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |    .114668   .0633757    -3.92   0.000     .0388144    .3387597
        age2 |   1.125743   .0528197     2.52   0.012     1.026836    1.234178
------------------------------------------------------------------------------

. sts gen HT = na

. mi set wide

. mi register impute age2

. mi impute chained (regress) age2 = HT _d drug, add(20)
note: missing-value pattern is monotone; no iteration performed

Conditional models (monotone):
              age2: regress age2 HT _d drug

Performing chained iterations ...

Multivariate imputation                     Imputations =       20
Chained equations                                 added =       20
Imputed: m=1 through m=20                       updated =        0

Initialization: monotone                     Iterations =        0
                                                burn-in =        0
```

```
              age2: linear regression

-----------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
              age2 |         39            9         9 |        48
-----------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. mi estimate, hr: stcox drug age2

Multiple-imputation estimates                  Imputations     =         20
Cox regression: Breslow method for ties        Number of obs   =         48
                                               Average RVI     =     0.0992
                                               Largest FMI     =     0.1501
DF adjustment:   Large sample                  DF:     min     =     865.68
                                                       avg     =   22335.80
                                                       max     =   43805.93
Model F test:       Equal FMI                  F(   2, 3791.4) =      12.28
Within VCE type:          OIM                  Prob > F        =     0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .1562903   .0717446    -4.04   0.000     .0635593    .3843125
        age2 |   1.127032   .0531534     2.54   0.011      1.02739    1.236338
------------------------------------------------------------------------------
```

6 Multiple imputation with clustered data

Missing data of course also occur in settings with clustered or repeated data, where random effects should be included, as in with the `mixed` command (formerly `xtmixed`). The STATA FAQ (Eddings and Marchenko, 2015) suggests 3 strategies for dealing with such clustering:

1. Include indicator variables for clusters in the imputation model.
2. Impute data separately for each cluster.
3. Use a multivariate normal model to impute all clusters simultaneously.

Strategy 1 is appropriate if there are few clusters with many observations. Using the code from the STATA FAQ, this is equivalent to

```
use http://www.stata.com/support/faqs/data1, clear
mi set wide
mi register imputed x
mi impute regress x y i.cluster, add(5) noisily
mi estimate: mixed y x || cluster:
```

Note the use of `i.cluster` in the `mi impute` statement.

With Strategy 2, we allow the regression models in the imputation procedure to vary by cluster. In the above example, we replace the `mi impute` command with

```
mi impute regress x y, add(5) by(cluster)
```

If you try the imputation `by(cluster)` and run into problems, consider the option `nostop` in the `by()` option. Strategy 3 can be used if a) only continuous repeated outcomes have missing observations, and b) observations only occur at fixed timepoints. See the above FAQ for an example.

Multiple imputation in multilevel data is an area of ongoing methodological research. One additional possibility is including the clusters as random effects in the imputation step, which is not yet possible in STATA, but can be performed in some cases in R. See van Buuren and Groothuis-Oudshoorn (2011) for more details. If none of these 4 strategies work, and the estimated random effects are small in the complete cases model, you may consider ignoring the clustering in the imputation process.

7 Postestimation commands after `mi estimate`

After fitting and combining model results with `mi estimate`, you may wish to

1. perform tests of the (transformed) coefficients with `mi test` (`mi testtransform`); or
2. obtain (nonlinear) predictions with `mi predict` (`mi predictnl`).

Each of these postestimation commands assumes that results from `mi estimate` have been saved using the `saving()` option. Then the postestimation command calls the saved estimates with `using`. Suppose we want to save the estimates with the name "`myest`", and make sure that they can replace previous results with that name. We could then write:

```
mi estimate, saving(myest, replace): regress y x1 x2 x3
mi test x1 x2 using myest
mi predict using myest
```

7.1 Testing coefficients

To test that a subset of coefficients, for example `x1` and `x2`, are *jointly* equal to zero ($x1 = x2 = 0$), use for example

```
mi estimate, saving(miest): regress y x1 x2
mi test x1 x2 using miest
```

If, however, we want to instead test that $x1 = x2$, we need to first tell `mi estimate` to also estimate `diff` $= x1 - x2$, and then use `mi testtransform` to test `diff`. For example, as shown in the example for `mi test`:

```
mi estimate (diff: _b[x1]-_b[x2]), saving(miest2):
            regress y x1 x2 x3 x4
mi testtransform diff using miest2
```

Note the use of parentheses around the definition of `diff`. Multiple differences or transformations would be defined individually in parentheses:

```
mi estimate (diff1: _b[x1]-_b[x2]) (diff2: _b[x1]-_b[x3]), saving(miest3): ...
mi testtransform diff1 diff2 using miest3
```

Similar code could be used to estimate and test a ratio of two variables.

7.2 Generating predictions

Linear predictions `xb_mi` (default) and their standard errors `se_mi` (with the option `stdp` for "standard error of the prediction") can be calculated using

```
mi estimate, saving(mi_est): regress y x1 x2 x3
mi predict xb_mi using mi_est
mi predict se_mi using mi_est, stdp
```

For cases where the outcome is non-linear, it is not possible to predict, say, probabilities (for logistic regression) directly. We can however easily convert the linear predictions to predicted probabilities using `invlogit` as follows:

```
mi estimate, saving(mi_est2): logistic y x1 x2 x3
mi predict xb_mi using mi_est2
mi xeq: generate phat = invlogit(xb_mi)
```

A similar procedure with `exp` instead of `invlogit` should theoretically work for `poisson` or `nbreg` models. The non-linear version `mi predictnl` can additionally be used to calculate a wide range of statistics related to the coefficients (`predict()`, `xb()`, `se()`, `var()`, `wald()`, `p()`, `ci()`) as well as the imputation process (`bvar()` [between-imputation variance], `wvar()` [within-imputation variance], `rvi()` [relative-variance increase], `fmi()` [fraction of missing information], `re()` [relative efficiency]). All of these options require the name(s) of the new variable(s) within the parentheses. Of particular interest is perhaps the confidence interval:

```
mi predictnl xb_mi = predict(xb) using miest, ci(lower upper) fmi(fmi)
```

In this line of code, we have simultaneously predicted the linear predictor `xb_mi`, corresponding confidence limits `lower` and `upper`, and the fraction of missing information `fmi`.

The usual postestimation commands such as `estat ic`, `estat gof`, `margins`, `predict` and `test` do *not* work after `mi estimate`. While clear rules exist for how to combine estimated coefficients across imputed datasets, it is unclear how likelihood estimates, random effects estimates, covariance matrixes or R^2 estimates, for example, should be combined. Thus, it is of great importance of consider issues related to model fit, variable selection, and so on prior to the multiple imputation process.

8 Frequently Asked Questions

8.1 Which variables should be included in the `mi impute` step?

All variables you were considering as predictors or confounders, as well as the outcome. White et al. (2011, Section 5) recommend using covariates and the outcome from the analysis models, as well as predictors of the incomplete variable. In general, you should include *all variables and interactions* in this step that you want to include in your final model, *even if that means imputing more than just the outcome variable*. In addition, if you plan on stratifying on any factors (e.g. `sex`), you should do this when you impute the data using the `by()` option (however it is not possible to both impute `sex` and perform the imputation `by(sex)`).

8.2 Are they any additional variables I should include in `mi impute` when preforming Cox regression?

All variables you were considering as predictors or confounders, as well as the outcome. White et al. (2011, Section 5) recommend using covariates and the outcome from the analysis models, as well as predictors of the incomplete variable. Further, White and Royston (2009) recommend using the

- the Nelson-Aalen estimate of the cumulative hazard (`HT` computed using: `sts gen HT = na`), as well as
- the event indicator (`_d`), which is created with `stset` command.

8.3 Which method do I use to impute the variables?

Assuming you have missing values in more than one variable, using `mi impute chained` to impute them all at once. Then, you have to specify in parentheses which type of regression (`regress`, `logit`, `ologit`, `mlogit`, `poisson`, `nbreg`, as well as `pmm` [good for continuous but skewed/non-normal variables], `truncreg` [for variables with a truncated range], `intreg` [interval censored]) you will use for each variable. The specific choice of regression methods here depends only on the type of variable with missing observations, and not on the method you will use in your final analysis. For example, we use `regress` here because `age` and `bmi` are continuous variables, even though we will use `logistic` to analyze associations with `attack` in the main analysis.

8.4 How many imputations?

White et al. (2011, Section 7.3) suggest a rule of thumb that we should impute at least `100 (incomplete / total)` (that is, the percentage of incomplete cases, which we can find using `mi describe`). Here we have `28 / 154 = 0.18` incomplete cases. This would indicate we need at least 20 imputations.

8.5 Note on categorical variables

Be careful when setting the categories on imputed variables. I had a number of errors when imputing and using `bmicut` until I made the lowest category large enough that a) all subjects were in a category, and b) all category had at least a few subjects. In other words, make sure the lowest and highest categories are large enough to cover imputed values more extreme than in the original dataset. For example, starting the lowest category at say `bmi = 13` may fail to include subjects if they have an imputed value of 12.5. On the other hand, if we have categories for `bmi` 10 to 14 and `bmi` 14 to 16, we will have trouble with the `mi estimate` command if there are no subjects with `bmi` between 10 and 14.

8.5.1 Other things to consider

If your imputation step is taking a long time, you may want to use the `dots` option. If your imputation step is producing errors, try the `noisily` option to help you figure out where the problem is occuring.

References

EDDINGS, W. and MARCHENKO, Y. (2015). How can I account for clustering when creating imputations with `mi impute`?
URL http://www.stata.com/support/faqs/statistics/clustering-and-mi-impute/

ROYSTON, P. and WHITE, I. R. (2011). Multiple imputation by chained equations (MICE): Implementation in STATA. *Journal of Statistical Software* **45** 1–20.
URL http://www.jstatsoft.org/v45/i04

STATA CORP (2013). *STATA 13 Multiple Imputation Reference Manual*. Stata Press, College Station, TX.
URL http://www.stata.com/manuals13/mi.pdf

UW-MADISON, S. S. C. C. (2013). Multiple imputation in STATA: Introduction.
URL http://www.ssc.wisc.edu/sscc/pubs/stata_mi_intro.htm

VAN BUUREN, S. and GROOTHUIS-OUDSHOORN, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software* **45**.
URL https://www.jstatsoft.org/article/view/v045i03

WHITE, I. R. and ROYSTON, P. (2009). Imputing missing covariate values for the Cox model. *Statistics in Medicine* **28** 1982–1998.
URL http://dx.doi.org/10.1002/sim.3618

WHITE, I. R., ROYSTON, P. and WOOD, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine* **30** 377–399.
URL http://dx.doi.org/10.1002/sim.4067

A Recent Changes

Version 1.1

- multiple imputation for clustered data, where models with random effects are used;
- "postestimation" commands available after `mi estimate`.
- Frequently asked questions have been moved to a separate section.