Missing Data Part II: Multiple Imputation & Maximum Likelihood

Richard Williams, University of Notre Dame, https://www3.nd.edu/~rwilliam/
Last revised September 20, 2024

Note: This is a very long handout!!! But most of it consists of appendices you can either skim through or check out if interested.

Note: Pp. 1-6 of this handout are designed to give you basic competency with Multiple Imputation, i.e. you will know what it does and why it is used and how to do some simple MI problems yourself. If you actually want to heavily use MI, you should read pp. 7-11, Appendix E, and some of the other appendices.

Warning: I teach about Multiple Imputation with some trepidation. You should know what it is and at least have reading competency with it. However, I have seen people try incredibly complicated imputation models before they have a lot of other basics down. For many/most purposes, at least for the work typically done in this class, listwise deletion is fine and MI adds little. Some people say to not even consider MI unless at least 15% or 20% of your data are missing. For your own papers, if you use it at all, MI should probably be one of the last things you do, rather than the first. And, if you do want to seriously use it, you should do a lot more reading than is in these notes. Some additional online sources (as of September 20, 2024) for information on MI are

http://www.ssc.wisc.edu/sscc/pubs/stata_mi_intro.htm (This is especially good)

https://www.stata.com/support/faqs/statistics/#mi (point #15 in the FAQ)

https://stats.idre.ucla.edu/stata/seminars/mi_in_stata_pt1_new/ (Also really good)

I. Advanced methods: Maximum Likelihood Estimation and Multiple Imputation.

Allison concludes that, of the conventional methods listed in Part I, listwise deletion often works the best. However, he argues that, under certain conditions, Maximum Likelihood Methods and Multiple Imputation Methods can work better.

Allison also argues that, while Maximum Likelihood techniques may be superior when they are available, either the theory or the software needed to estimate them is often lacking. Therefore this handout will primarily focus on multiple imputation. However if you are primarily interested in linear regression models, you may prefer ML to MI. Appendix D briefly discusses ML.

In a 2000 Sociological Methods and Research paper entitled "Multiple Imputation for Missing Data: A Cautionary Tale" Allison summarizes the basic rationale for multiple imputation:

> Multiple imputation (MI) appears to be one of the most attractive methods for general-purpose handling of missing data in multivariate analysis. The basic idea, first proposed by Rubin (1977) and elaborated in his (1987) book, is quite simple:
>
> 1. Impute missing values using an appropriate model that incorporates random variation.
>
> 2. Do this M times producing M "complete" data sets.
>
> 3. Perform the desired analysis on each data set using standard complete-data methods.
>
> 4. Average the values of the parameter estimates across the M samples to produce a single point estimate.

5. Calculate the standard errors by (a) averaging the squared standard errors of the M estimates (b) calculating the variance of the M parameter estimates across samples, and (c) combining the two quantities using a simple formula.

Allison adds that

- Introducing appropriate random error into the imputation process makes it possible to get approximately unbiased estimates of all parameters. No deterministic imputation method can do this in general settings.
- Repeated imputation allows one to get good estimates of the standard errors. Single imputation methods don't allow for the additional error introduced by imputation (without specialized software of very limited generality).

More of Allison's argument (including a discussion of the assumptions required for MI and how they can be violated) appear in *Appendix E*.

We now show some of the ways Stata can handle multiple imputation problems.

II. Using Stata 11 or higher for Multiple Imputation for One Variable

This example is adapted from pages 1-14 of the Stata 12 Multiple Imputation Manual (which I highly recommend reading) and also quotes directly from the Stata 12 online help. If you have Stata 11 or higher the entire manual is available as a PDF file. This is a simple example and there are other commands and different ways to do multiple imputation, so you should do a lot more reading if you want to use MI yourself.

NOTE: This example focuses on using regress to impute missing values for a single continuous variable. *Appendix A* shows other examples, such as logit and mlogit for categorical variables. It also shows how to use Predictive Mean Matching (PMM), a sometimes attractive alternative to regress for continuous variables with missing data. *Appendix B* shows how to do multiple imputation when more than one variable has missing data. *Appendix C* shows roughly how multiple imputation works its magic. *Appendix D* discusses Full Information Maximum Likelihood, which is a great alternative to MI in those situations where it works.

The file `mheart0.dta` is a fictional data set with 154 cases, 22 of which are missing data on `bmi` (Body Mass Index). The dependent variable for this example is `attack`, coded 0 if the subject did not have a heart attack and 1 if he or she did.

```
. version 12.1
. * Imputation for a single continuous variable using regress
. webuse mheart0, clear
(Fictional heart attack data; bmi missing)

. sum

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      attack |       154    .4480519    .4989166          0          1
      smokes |       154    .4155844    .4944304          0          1
         age |       154    56.48829    11.73051   20.73613   87.14446
         bmi |       132    25.24136    4.027137   17.22643   38.24214
      female |       154    .2467532    .4325285          0          1
-------------+--------------------------------------------------------
      hsgrad |       154    .7532468    .4325285          0          1
   marstatus |       154    1.941558    .8183916          1          3
     alcohol |       154    1.181818    .6309506          0          2
     hightar |       154    .2077922     .407051          0          1

. mi set mlong
```

[From the Stata 12 online help:] `mi set` is used to set a regular Stata dataset to be an mi dataset. An mi set dataset has the following attributes:

- The data are recorded in a style: wide, mlong, flong, or flongsep.
- Variables are registered as imputed, passive, or regular, or they are left unregistered.
- In addition to m=0, the data with missing values, the data include M>=0 imputations of the imputed variables.

For this example, the Stata 12 Manual says "we choose to use the data in the marginal long style (`mlong`) because it is a memory-efficient style." Type `help mi styles` for more details.

```
. mi register imputed bmi
(22 m=0 obs. now marked as incomplete)
. mi register regular attack smokes age hsgrad female
```

An *imputed* variable is a variable that has missing values and for which you have or will have imputations. All variables whose missing values are to be filled in must be registered as imputed variables. A *passive* variable (not used in this example) is a variable that is a function of imputed variables (e.g. an interaction effect) or of other passive variables. A passive variable will have missing values in m=0 (the original data set) and varying values for observations in m>0 (the imputed data sets). A *regular* variable is a variable that is neither imputed nor passive and that has the same values, whether missing or not, in all m; registering regular variables is optional but recommended. In the above, we are telling Stata that the values of `bmi` will be imputed while the values of the other variables will not be.

```
. mi impute regress bmi attack smokes age hsgrad female, add(20) rseed(2232)

Univariate imputation                    Imputations =       20
Linear regression                              added =       20
Imputed: m=1 through m=20                    updated =        0

               |               Observations per m
               |----------------------------------------------
      Variable |   complete   incomplete   imputed |     total
---------------+-----------------------------------+----------
           bmi |        132           22        22 |       154
--------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled in observations.)
```

The `mi impute` command fills in missing values (.) of a single variable or of multiple variables using the specified method. In this case, the use of `regress` means use a linear regression for a continuous variable; i.e. `bmi` is being regressed on `attack smokes age hsgrad & female`. The Stata 12 manual includes guidelines for choosing variables to include in the imputation model.

- One of the most common/important recommendations is that *the analytic model and the imputation model should be congenial, i.e. the imputation model should include the same variables (including the dependent variable) that are in the analytic model; otherwise relationships with the variables that have been omitted will be biased toward 0.*
- Other methods include `logit`, `ologit` and `mlogit`, e.g. you would use `logit` if you had a binary variable you wanted to impute values for.
- The `add` option specifies the number of imputations, in this case 20. (Stata recommends using at least 20 although it is not unusual to see as few as 5.) The `rseed` option sets the random number seed which makes results reproducible (different seeds will produce different imputed data sets).
- Case 8 is the first case with missing data on `bmi`, so let's see what happens to it after imputation:

```
. list bmi attack smokes age hsgrad female _mi_id _mi_miss _mi_m if  _mi_id ==8

     +------------------------------------------------------------------------------------+
     |      bmi   attack   smokes        age   hsgrad   female   _mi_id   _mi_miss   _mi_m |
     |------------------------------------------------------------------------------------|
  8. |        .        0        0   60.35888        0        0        8          1       0 |
155. | 20.58218        0        0   60.35888        0        0        8          .       1 |
177. | 27.40752        0        0   60.35888        0        0        8          .       2 |
199. |  22.1714        0        0   60.35888        0        0        8          .       3 |
221. | 22.45379        0        0   60.35888        0        0        8          .       4 |
     |------------------------------------------------------------------------------------|
243. | 31.89095        0        0   60.35888        0        0        8          .       5 |
265. | 27.42568        0        0   60.35888        0        0        8          .       6 |
287. | 27.62364        0        0   60.35888        0        0        8          .       7 |
309. | 33.36433        0        0   60.35888        0        0        8          .       8 |
331. | 21.90939        0        0   60.35888        0        0        8          .       9 |
     |------------------------------------------------------------------------------------|
353. | 26.93499        0        0   60.35888        0        0        8          .      10 |
375. | 25.82896        0        0   60.35888        0        0        8          .      11 |
397. |  24.6579        0        0   60.35888        0        0        8          .      12 |
419. | 23.59406        0        0   60.35888        0        0        8          .      13 |
441. | 24.35756        0        0   60.35888        0        0        8          .      14 |
     |------------------------------------------------------------------------------------|
463. | 28.23293        0        0   60.35888        0        0        8          .      15 |
485. | 31.92563        0        0   60.35888        0        0        8          .      16 |
507. | 31.16652        0        0   60.35888        0        0        8          .      17 |
```

```
529. | 20.54303        0        0   60.35888        0        0        8         .      18 |
551. | 21.39175        0        0   60.35888        0        0        8         .      19 |
     |-----------------------------------------------------------------------------------|
573. | 27.27427        0        0   60.35888        0        0        8         .      20 |
     +-----------------------------------------------------------------------------------+
```

`bmi` is missing in the original unimputed data set (`_mi_m = 0`). For each of the 20 imputed data sets, a different value has been imputed for `bmi`. The imputation of multiple plausible values will let the estimation procedure take into account the fact that the true value is unknown and hence uncertain.

The Stata 12 Manual recommends checking to see whether the imputations appear reasonable. In this case we do so by running the `mi xeq` command, which executes command(s) on individual imputations. Specifically, we run the `summarize` command on the original data set (m = 0) and on the (arbitrarily chosen) first and last imputed data sets. The means and standard deviations for `bmi` are all similar and seem reasonable in this case:

```
. mi xeq 0 1 20: summarize bmi

m=0 data:
-> summarize bmi

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         bmi |       132    25.24136    4.027137   17.22643   38.24214

m=1 data:
-> summarize bmi

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         bmi |       154    25.11855    3.990918   15.47331   38.24214

m=20 data:
-> summarize bmi

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         bmi |       154    25.37117    4.051929    15.4505   38.24214
```

The `mi estimate` command does estimation using multiple imputations. The desired analysis is done on each imputed data set and the results are then combined into a single multiple-imputation result (the `dots` option just tells Stata to print a dot after each estimation; it helps you track progress and an X gets printed out if there is a problem doing one of the estimations):

```
. mi estimate, dots: logit attack smokes age bmi hsgrad female

Imputations (20):
  .........10.........20 done

Multiple-imputation estimates                  Imputations     =         20
Logistic regression                            Number of obs   =        154
                                               Average RVI     =     0.0404
                                               Largest FMI     =     0.1678
DF adjustment:   Large sample                  DF:     min     =     694.17
                                                       avg     =  115477.35
                                                       max     =  287682.25
Model F test:       Equal FMI                  F(   5,43531.9) =       3.74
Within VCE type:          OIM                  Prob > F        =     0.0022

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.239172   .3630877     3.41   0.001     .5275236    1.950821
         age |   .0354929   .0154972     2.29   0.022     .0051187     .065867
         bmi |   .1184188   .0495676     2.39   0.017     .0210985    .2157391
      hsgrad |    .185709   .4075301     0.46   0.649    -.6130435    .9844615
      female |  -.0996102   .4193583    -0.24   0.812    -.9215408    .7223204
       _cons |  -5.845855    1.72309    -3.39   0.001    -9.225542   -2.466168
------------------------------------------------------------------------------
```

Note that you don't always get the same information as you do with non-imputed data sets (e.g. Pseudo R^2), partly because these things don't always make sense with imputed data or because it is not clear how to compute them.

Compare this to the results when we only analyze the original unimputed data:

```
. mi xeq 0: logit attack smokes age bmi hsgrad female, nolog

m=0 data:
-> logit attack smokes age bmi hsgrad female, nolog

Logistic regression                            Number of obs   =        132
                                               LR chi2(5)      =      24.03
                                               Prob > chi2     =     0.0002
Log likelihood =  -79.34221                    Pseudo R2       =     0.1315

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.544053   .3998329     3.86   0.000     .7603945    2.327711
         age |    .026112    .017042     1.53   0.125    -.0072898    .0595137
         bmi |   .1129938   .0500061     2.26   0.024     .0149837     .211004
      hsgrad |   .4048251   .4446019     0.91   0.363    -.4665786    1.276229
      female |   .2255301   .4527558     0.50   0.618    -.6618549    1.112915
       _cons |  -5.408398   1.810603    -2.99   0.003    -8.957115    -1.85968
------------------------------------------------------------------------------
```

The most striking difference is that the effect of age is statistically significant in the imputed data, whereas it wasn't in the original data set.

Note: The rest of this handout is optional!!! You should read it if you want to actually do Multiple Imputation, rather than just get a general understanding of what it does. I may cover a few key points in class.

III. Estimating adjusted predictions and marginal effects after using multiple imputation. The `margins` command does not work after using `mi estimate`. Daniel Klein's user-written `mimrgns` (available from SSC) does. While `mimrgns` can be extremely helpful, it is also a use at your own risk sort of routine. Be sure to carefully read the help file first, which warns that "There might be good reasons why `margins` does not work after `mi estimate`." The help further warns that, if you also use `marginsplot`, the DF and confidence intervals may be a little off (which may be a reason for not including the CIs when using `marginsplot`). In an email to me Klein further warned that he was not sure "whether predicted probabilities at fixed values qualify for pooling according to Rubin rules." Having said all that, `mimrgns` may be as good as it gets for now if you want to use both multiple imputation and adjusted predictions/ marginal effects. The help file includes links that explain the approach `mimrgns` uses.

Here is an example (thanks to both Christopher Quiroz and Daniel Klein for helping come up with this). On the `mimrgns` command, note the use of `predict(pr)` to get predicted probabilities (otherwise you would get log odds); and the `cmdmargins` option, which is needed if you also want to use `marginsplot`.

```
. * Use mimrgns -- but with caution
. mi estimate, dots: logit attack i.smokes age bmi i.hsgrad i.female

Imputations (20):
  .........10.........20 done

Multiple-imputation estimates                   Imputations       =         20
Logistic regression                             Number of obs     =        154
                                                Average RVI       =     0.0404
                                                Largest FMI       =     0.1678
DF adjustment:   Large sample                   DF:     min       =     694.17
                                                        avg       = 115,477.35
                                                        max       = 287,682.25
Model F test:       Equal FMI                   F(   5,43531.9)   =       3.74
Within VCE type:          OIM                   Prob > F          =     0.0022

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    1.smokes |   1.239172   .3630877     3.41   0.001     .5275236    1.950821
         age |   .0354929   .0154972     2.29   0.022     .0051187     .065867
         bmi |   .1184188   .0495676     2.39   0.017     .0210985    .2157391
    1.hsgrad |    .185709   .4075301     0.46   0.649    -.6130435    .9844615
    1.female |  -.0996102   .4193583    -0.24   0.812    -.9215408    .7223204
       _cons |  -5.845855    1.72309    -3.39   0.001    -9.225542   -2.466168
------------------------------------------------------------------------------

. mimrgns smokes, at (age = (20 (10) 90)) predict(pr) cmdmargins vsquish

Imputations (20):
  .........10.........20 done

Multiple-imputation estimates                   Imputations       =         20
```

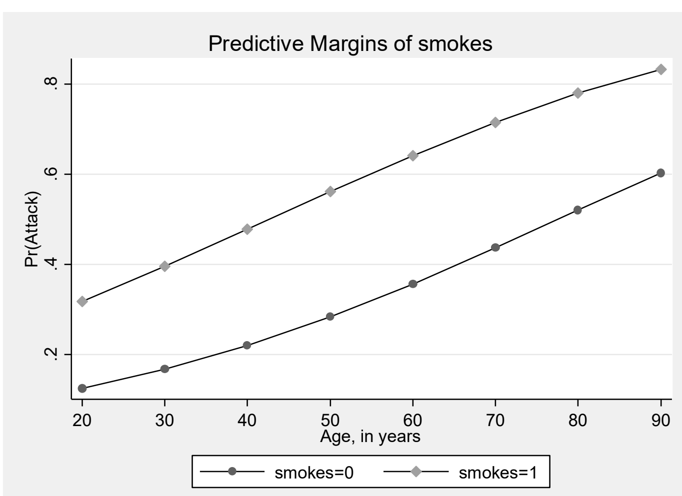
```
Predictive margins                              Number of obs     =        154
                                                Average RVI       =     0.0231
                                                Largest FMI       =     0.0123
DF adjustment:   Large sample                   DF:     min       = 126,876.37
                                                        avg       = 2178580.27
Within VCE type: Delta-method                           max       =   2.11e+07

Expression   : Pr(attack), predict(pr)
1._at        : age             =          20
2._at        : age             =          30
3._at        : age             =          40
4._at        : age             =          50
5._at        : age             =          60
6._at        : age             =          70
7._at        : age             =          80
8._at        : age             =          90

------------------------------------------------------------------------------
             |     Margin   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  _at#smokes |
        1 0  |   .1246298   .0653831     1.91   0.057     -.003519    .2527785
        1 1  |   .3182818   .1262326     2.52   0.012      .070868    .5656955
        2 0  |   .1670511   .0641717     2.60   0.009     .0412767    .2928255
        2 1  |   .3951355   .1084744     3.64   0.000     .1825276    .6077434
        3 0  |    .219916   .0578518     3.80   0.000     .1065285    .3333035
        3 1  |   .4774527   .0855896     5.58   0.000      .309699    .6452064
        4 0  |   .2834091   .0498432     5.69   0.000     .1857182       .3811
        4 1  |   .5611137   .0661128     8.49   0.000     .4315346    .6906929
        5 0  |   .3564808   .0515087     6.92   0.000     .2555255    .4574361
        5 1  |   .6417033   .0603787    10.63   0.000     .5233633    .7600434
        6 0  |   .4366583    .070939     6.16   0.000     .2976196     .575697
        6 1  |   .7153617   .0673338    10.62   0.000     .5833898    .8473337
        7 0  |    .520223   .0993719     5.24   0.000     .3254559      .71499
        7 1  |   .7794453    .075908    10.27   0.000     .6306682    .9282224
        8 0  |   .6027925   .1261848     4.78   0.000     .3554726    .8501125
        8 1  |   .8327803   .0797243    10.45   0.000     .6765233    .9890373
------------------------------------------------------------------------------

. marginsplot, noci scheme(sj) name(mimrgnsplot)

  Variables that uniquely identify margins: age smokes
```



IV. Already existing MI data sets. If you are lucky, somebody else may have already done the imputation for you (although it is possible that you might do even better since you know what variables are in your analytic models); and if you are super-lucky, the MI data will already be in Stata format. If not, you'll have to convert it to Stata yourself. The `mi import` command may be useful for this purpose. Once the data are in Stata format, the `mi describe` command can be used to provide a detailed report. Using the above data,

```
. mi describe

  Style:  mlong

  Obs.:   complete          132
          incomplete         22  (M = 20 imputations)
          ---------------------
          total             154

  Vars.:  imputed:  1; bmi(22)

          passive: 0

          regular: 5; attack smokes age hsgrad female

          system:  3; _mi_m _mi_id _mi_miss

         (there are 3 unregistered variables; marstatus alcohol hightar)
```

V. Other comments on multiple imputation

Imputation is pretty easy when only one variable has missing data. It can get more complicated in the more typical case when several variables have missing data. Again, this handout is just a brief introduction; read the manual and some related articles if you want to use multiple imputation in your own analyses.

Random number generator. Stata's random number generator has changed across versions, so even if you do specify rseed you may not get identical results, e.g. some results I got using Stata 11 were not the same as results I got using Stata 12. Using version control should keep things consistent. For more, see `help version` and, possibly (for Stata 14+) , `help set rng`.

Soft versus hard missing data codes. Stata has "soft" missing codes (coded as .) and "hard" missing codes (.a, .b, .c, …, .z). The former are eligible for imputation, the latter are not. This distinction can be useful when variables should not be imputed, e.g. "Number of times pregnant" is not applicable for men; either code it as zero or leave it as missing. Depending on the nature of the variable, you may need to change some soft codes to hard or hard codes to soft. Otherwise you may fail to impute values when you should or else impute values when you shouldn't. As stated before, you need to understand why data are missing.

Auxiliary variables. UCLA (https://stats.idre.ucla.edu/stata/seminars/mi_in_stata_pt1_new/) says

> Auxiliary variables are variables in your data set that are either correlated with a missing variable(s) (the recommendation is r > 0.4) or are believed to be associated with missingness. These are factors that are not of particular interest in your analytic model , but they are added to the imputation model to increase power and/or to help make the assumption of MAR more plausible. These variables have been found to improve the quality of imputed values generate from multiple imputation. Moreover, research has demonstrated their particular importance when imputing a dependent variable and/or when you have variables with a high proportion of missing information (Johnson and Young, 2011; Young and Johnson, 2010; Enders , 2010). You may a priori know of several variables you believe would make good auxiliary variables based on your knowledge of the data and subject matter. Additionally, a good review of the literature can often help identify them as well.

Multiple imputation on the dependent variable. Multiple imputation on the independent variables can be good because it lets you use the non-missing information on the other independent variables. Multiple imputation of the dependent variable, however, tends to gain you little or nothing. (One possible exception is when you have auxiliary variables that are strongly correlated with the dependent variable, e.g. r = .5 or greater, such as the same variable measured at different points in time.) Of course, the dependent variable in one part of the analysis may be an independent variable in a different part, so you may go ahead and do the imputation on the variable anyway.

UCLA (https://stats.idre.ucla.edu/stata/seminars/mi_in_stata_pt1_new/) adds this advice:

> Additionally, using imputed values of your DV is considered perfectly acceptable when you have good auxiliary variables in your imputation model (Enders, 2010; Johnson and Young, 2011; White et al., 2010). However, if good auxiliary variables are not available then you still INCLUDE your DV in the imputation model and then later restrict your analysis to only those observations with an observed DV value. Research has shown that imputing DV's when auxiliary variables are not present can add unnecessary random variation into your imputed values (Allison, 2012) .

Other programs for multiple imputation. User-written programs like `ice` and `mim` can also be used for imputation and estimation. I think Stata 12 largely eliminates the need for those programs. But even if you have Stata 12, the articles that have been written about these programs may be helpful to you in understanding how the ICE method works.

Note: In a 2017 Statalist discussion, some people claimed that `ice` worked better in some situations, e.g. when `mlogit` was being used as the imputation method (but they also expressed concern that they weren't sure `ice` was giving correct results in these situations). If you are having trouble with `mi impute`, you may wish to look at

http://www.statalist.org/forums/forum/general-stata-discussion/general/1371095-multiple-imputation

https://stats.idre.ucla.edu/stata/faq/how-can-i-perform-multiple-imputation-on-longitudinal-data-using-ice/

Passive imputation versus "just another variable" (JAV) approach. Passive imputation is somewhat controversial. With passive imputation, you would, for example, impute values for x1 and x2, and then multiply those values together to create the interaction term x1x2. The alternative is to multiply x1 * x2 before imputation, and then impute values for the resulting x1x2 interaction term, i.e. the “just another variable” (JAV) approach. Perhaps surprisingly, some people (including Paul Allison) claim that the JAV approach is superior. The issue was discussed on Stata List in February 2009. If interested, see

http://www.stata.com/statalist/archive/2009-02/msg00602.html

http://www.stata.com/statalist/archive/2009-02/msg00613.html

In the latter message, Paul Allison says “In multiple imputation, interactions should be imputed as though they are additional variables, not constructed by multiplying imputed values. The same is true if you have x and x^2 in a model. The x^2 term should be imputed just like any other variable, not constructed by squaring the imputed values of x. While this principle may seem counterintuitive, it is easily demonstrated by simulation that the more “natural" way to do it produces biased estimates.”

For more good discussion of JAV vs Passive Imputation, as well as several other issues, see

White, Ian R., Royston, Patrick, Wood, Angela M. 2011. “Multiple imputation using chained equations: Issues and guidance for practice.” Statistics in Medicine. Pp. 377-399.

NOTE: There is at least one exception. Suppose you are trying to compute a scale that is the sum of several items. In an email to me, Allison said “It's better, when possible, to impute at the item level rather than the scale level. Otherwise you lose a lot of data. This is one case where JAV doesn't apply.”

Multiple Imputation with Panel/ Longitudinal Data. See the following Stata FAQ, “How can I account for clustering when creating imputations with mi impute?”

https://www.stata.com/support/faqs/statistics/clustering-and-mi-impute/

> Excerpt: As of Stata 11.1, the mi estimate command can be used to analyze multiply imputed clustered (panel or longitudinal) data by fitting several clustered-data models, such as xtreg, xtlogit, and mixed; see mi estimation for the full list. However, we must also account for clustering when creating multiply imputed data; this FAQ will show how.

Appendix A: More Examples of Multiple Imputation for a Single Variable

These examples (and much of the text) are pretty much copied straight from the Stata 12 or 13 Multiple Imputation Manual. Read the manual for more details. Further, multiple methods can be used if you specify `mi impute chained` (see Appendix B). Read the manual if you want to get into other methods or more complicated imputations. I will either go over these quickly or not at all in class.

PMM – Predictive Mean Matching. PMM is an alternative to regress when imputing values for continuous variables. It may be preferable to linear regression when the normality of the variable is suspect (which is likely the case with BMI). The basic idea is that you again use regression methods to come up with an estimate of the missing value for variable X. However, rather than use that estimate, you identify one or more neighbors who have similar estimated values. (Note that it is the estimated value for the neighbor, not the neighbor's observed value.) The observed value of the nearest neighbor (or the randomly chosen nearest neighbor) is then used for the imputed value for the case with missing data on X.

So, for example, suppose that case 8 is missing on X, and the estimated value for X is 18.71. Suppose the nearest neighbor has an estimated value of 18.73, with an observed value of 20. Twenty will be used as the imputed value of X for case 8. (If the nearest neighbor was a big outlier, e.g. estimated value of 18.73 with observed value of 50, you would still use the observed value of 50 as the imputed value.) Or, if you have specified, say, 5 nearest neighbors, one of them will be chosen at random and their observed value on X will be used as the imputed value for case 8.

In other words, the method identifies neighbors who have complete data that have estimated values on X that are close to the estimated value for the person with incomplete data. One of these neighbors is chosen as a "donor", and the donor's observed value on the variable replaces the recipient's missing value.

You have to choose how many neighbors are to be used. If you only choose 1, your MI estimates may be highly variable from one imputation to the next. Including too many neighbors may bias your point estimates. In other words there is a tradeoff between biased estimators and estimators that have larger standard errors. The Stata Manual seems to use 1, 3 or 5 neighbors in its examples.

Here is an example from the manual. It uses the same data we used in our earlier example but uses PMM instead of regress to impute values for BMI.

```
. webuse mheart0, clear
(Fictional heart attack data; bmi missing)
. mi set mlong
. mi register imputed bmi
(22 m=0 obs. now marked as incomplete)
. mi impute pmm bmi attack smokes age hsgrad female, add(20) knn(5) rseed(2232)

Univariate imputation                       Imputations =       20
Predictive mean matching                          added =       20
Imputed: m=1 through m=20                       updated =        0

                                      Nearest neighbors =        5

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
               bmi |        132           22        22 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)
```

As the Stata Manual explains, "By default, `mi impute pmm` uses one nearest neighbor to draw from. That is, it replaces missing values with an observed value whose linear prediction is the closest to that of the missing value. Using only one nearest neighbor may result in high variability of the MI estimates. You can increase the number of nearest neighbors from which the imputed value is drawn by specifying the knn() option." In the example above I told Stata to select a donor from the 5 nearest neighbors. If you look at the imputed values, you may even be able to figure out who the donor was (e.g. if the imputed value for case 8 is 20 and case 47 is the only case with an observed value of 20, then case 47 must be the donor).

```
. mi estimate: logit attack smokes age bmi hsgrad female

Multiple-imputation estimates                     Imputations     =         20
Logistic regression                               Number of obs   =        154
                                                  Average RVI     =     0.0419
                                                  Largest FMI     =     0.1801
DF adjustment:   Large sample                     DF:     min     =     603.59
                                                          avg     =  287949.70
                                                          max     =  751953.76
Model F test:       Equal FMI                     F(   5,40396.1) =       3.63
Within VCE type:          OIM                     Prob > F        =     0.0028

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.215069   .3622206     3.35   0.001     .5051101    1.925029
         age |   .0362938   .0154764     2.35   0.019     .0059605     .066627
         bmi |   .1133446   .0505589     2.24   0.025     .0140518    .2126374
      hsgrad |   .1702272   .4049114     0.42   0.674    -.6233872    .9638415
      female |  -.0961759   .4171239    -0.23   0.818     -.913725    .7213732
       _cons |  -5.741508   1.753138    -3.27   0.001    -9.180562   -2.302453
------------------------------------------------------------------------------
```

While PMM may be superior to regress in some cases, it barely matters here. Recall that this is what we got earlier when we used regress to impute the values of BMI:

```
Multiple-imputation estimates                     Imputations     =         20
Logistic regression                               Number of obs   =        154
                                                  Average RVI     =     0.0404
                                                  Largest FMI     =     0.1678
DF adjustment:   Large sample                     DF:     min     =     694.17
                                                          avg     =  115477.35
                                                          max     =  287682.25
Model F test:       Equal FMI                     F(   5,43531.9) =       3.74
Within VCE type:          OIM                     Prob > F        =     0.0022

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.239172   .3630877     3.41   0.001     .5275236    1.950821
         age |   .0354929   .0154972     2.29   0.022     .0051187     .065867
         bmi |   .1184188   .0495676     2.39   0.017     .0210985    .2157391
      hsgrad |    .185709   .4075301     0.46   0.649    -.6130435    .9844615
      female |  -.0996102   .4193583    -0.24   0.812    -.9215408    .7223204
       _cons |  -5.845855    1.72309    -3.39   0.001    -9.225542   -2.466168
------------------------------------------------------------------------------
```

I suppose if you were really worried about whether pmm or regress was most appropriate, you could try both and see if it makes much difference.

Logit. Logit imputation is used when the variable with missing data has only two possible values, 0 and 1. In this example, hsgrad (coded 1 if high school graduate, 0 otherwise) has the missing data.

```
. webuse mheart2, clear
(Fictional heart attack data; hsgrad missing)
. mi set mlong
. * This will show us how much missing data, and the ranges of observed values
. mi misstable summarize
                                                            Obs<.
                                                +------------------------------
               |                                | Unique
      Variable |     Obs=.     Obs>.     Obs<.  | values        Min         Max
  -------------+--------------------------------+------------------------------
        hsgrad |        18                 136  |      2          0           1
  -----------------------------------------------------------------------------

. mi register imputed hsgrad
(18 m=0 obs. now marked as incomplete)
. mi impute logit hsgrad attack smokes age bmi female, add(10) rseed(2232)

Univariate imputation                       Imputations =       10
Logistic regression                               added =       10
Imputed: m=1 through m=10                       updated =        0

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
            hsgrad |        136           18        18 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. * Estimates before imputation
. mi xeq 0: logit attack smokes age bmi female hsgrad, nolog

m=0 data:
-> logit attack smokes age bmi female hsgrad

Logistic regression                               Number of obs   =        136
                                                  LR chi2(5)      =      23.99
                                                  Prob > chi2     =     0.0002
Log likelihood = -81.903374                       Pseudo R2       =     0.1278

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.475308   .3901501     3.78   0.000     .7106284    2.239989
         age |   .0294918   .0166343     1.77   0.076    -.0031108    .0620944
         bmi |   .1168109   .0498207     2.34   0.019     .0191641    .2144578
      female |    .170943   .4452731     0.38   0.701    -.7017761    1.043662
      hsgrad |   .3634346    .436017     0.83   0.405    -.4911431    1.218012
       _cons |  -5.688296   1.791735    -3.17   0.001    -9.200032    -2.17656
------------------------------------------------------------------------------
```

```
. * Estimates after imputation
. mi estimate: logit attack smokes age bmi female hsgrad

Multiple-imputation estimates                     Imputations     =         10
Logistic regression                               Number of obs   =        154
                                                  Average RVI     =     0.0244
                                                  Largest FMI     =     0.1267
DF adjustment:   Large sample                     DF:     min     =     588.19
                                                          avg     =   7.02e+07
                                                          max     =   2.75e+08
Model F test:       Equal FMI                     F(   5,47292.4) =       3.85
Within VCE type:          OIM                     Prob > F        =     0.0017

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.274902   .3654074     3.49   0.000     .5587127    1.991092
         age |   .0369741   .0154912     2.39   0.017     .0066119    .0673363
         bmi |   .1236749   .0464216     2.66   0.008     .0326902    .2146596
      female |  -.1111262   .4195926    -0.26   0.791    -.9335126    .7112603
      hsgrad |   .3176137   .4394874     0.72   0.470    -.5455419    1.180769
       _cons |  -6.169885   1.680838    -3.67   0.000    -9.464291   -2.875478
------------------------------------------------------------------------------
```

mlogit. Multinomial logit can be used when a variable is nominal and has more than 2 categories. Marital Status (1 = single, 2 = married, 3 = divorced) is the missing data victim this time.

```
. webuse mheart3, clear
(Fictional heart attack data; marstatus missing)
. mi set mlong
. mi misstable summarize
                                                               Obs<.
                                                +------------------------------
               |                                | Unique
      Variable |     Obs=.     Obs>.     Obs<.  | values        Min         Max
  -------------+--------------------------------+------------------------------
     marstatus |         7                 147  |      3          1           3
  -----------------------------------------------------------------------------

. mi register imputed marstatus
(7 m=0 obs. now marked as incomplete)

. mi impute mlogit marstatus attack smokes age bmi female hsgrad, add(20) rseed(2232)

Univariate imputation                       Imputations =       20
Multinomial logistic regression                   added =       20
Imputed: m=1 through m=20                       updated =        0

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
         marstatus |        147            7         7 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)
```

```
. * Estimates before imputation
. mi xeq 0: logit attack smokes age bmi female hsgrad i.marstatus

m=0 data:
-> logit attack smokes age bmi female hsgrad i.marstatus

Iteration 0:   log likelihood =   -101.126
Iteration 1:   log likelihood = -87.825045
Iteration 2:   log likelihood = -87.797081
Iteration 3:   log likelihood = -87.797076

Logistic regression                               Number of obs   =        147
                                                  LR chi2(7)      =      26.66
                                                  Prob > chi2     =     0.0004
Log likelihood = -87.797076                       Pseudo R2       =     0.1318

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.439608   .3786929     3.80   0.000     .6973834    2.181832
         age |    .035506   .0164938     2.15   0.031     .0031787    .0678333
         bmi |   .1076301    .047991     2.24   0.025     .0135695    .2016907
      female |   .1777255   .4391299     0.40   0.686    -.6829532    1.038404
      hsgrad |   .0844021   .4171585     0.20   0.840    -.7332135    .9020176
             |
   marstatus |
          2  |   .7620136   .4520608     1.69   0.092    -.1240092    1.648036
          3  |  -.0357522   .4601057    -0.08   0.938    -.9375427    .8660383
             |
       _cons |  -5.882399   1.734636    -3.39   0.001    -9.282223   -2.482575
------------------------------------------------------------------------------

. * Estimates after imputation
. mi estimate: logit attack smokes age bmi female hsgrad i.marstatus

Multiple-imputation estimates                     Imputations     =         20
Logistic regression                               Number of obs   =        154
                                                  Average RVI     =     0.0131
                                                  Largest FMI     =     0.0479
DF adjustment:   Large sample                     DF:     min     =    8349.75
                                                          avg     = 3041131.92
                                                          max     = 7758178.52
Model F test:       Equal FMI                     F(   7,584619.8)=       3.14
Within VCE type:          OIM                     Prob > F        =     0.0026

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.345395   .3736001     3.60   0.000     .6131516    2.077638
         age |   .0398306   .0159254     2.50   0.012     .0086173    .0710438
         bmi |   .1254246   .0466702     2.69   0.007     .0339526    .2168966
      female |   .0114877   .4303667     0.03   0.979    -.8320164    .8549917
      hsgrad |    .072225   .4139451     0.17   0.861    -.7390926    .8835427
             |
   marstatus |
          2  |   .7599448   .4521048     1.68   0.093    -.1262735    1.646163
          3  |  -.0337952   .4612619    -0.07   0.942     -.937983    .8703927
             |
       _cons |  -6.497155   1.708516    -3.80   0.000    -9.845784   -3.148525
------------------------------------------------------------------------------
```

ologit. The ordered logistic regression imputation method can be used to fill in missing values of an ordinal variable (e.g. the variable is coded high, medium low; Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree; Poor, Fair, Good, Excellent).

```
. use http://www.stata-press.com/data/r13/mheart4, clear
(Fictional heart attack data; alcohol missing)
. tabulate alcohol, missing

    Alcohol consumption: |
none, <2 drinks/day, >=2 |
               drinks/day |      Freq.     Percent        Cum.
-------------------------+-----------------------------------
             Do not drink |         18       11.69       11.69
  Less than 3 drinks/day |         83       53.90       65.58
Three or more drinks/day |         44       28.57       94.16
                        . |          9        5.84      100.00
-------------------------+-----------------------------------
                    Total |        154      100.00

. mi set mlong
. mi register imputed alcohol
(9 m=0 obs. now marked as incomplete)

. mi impute ologit alcohol attack smokes age bmi female hsgrad, ///
>         add(10) rseed(2232)

Univariate imputation                      Imputations =       10
Ordered logistic regression                      added =       10
Imputed: m=1 through m=10                      updated =        0

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
           alcohol |        145            9         9 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. mi estimate: logit attack smokes age bmi female hsgrad i.alcohol

Multiple-imputation estimates                   Imputations     =         10
Logistic regression                             Number of obs   =        154
                                                Average RVI     =     0.0139
                                                Largest FMI     =     0.0769
DF adjustment:   Large sample                   DF:     min     =    1569.99
                                                        avg     =   3.73e+07
                                                        max     =   1.35e+08
Model F test:       Equal FMI                   F(   7,228385.6)=       2.79
Within VCE type:          OIM                   Prob > F        =     0.0067

------------------------------------------------------------------------------------------
                   attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------------------+----------------------------------------------------------------
                   smokes |   1.238867   .3625929     3.42   0.001     .5281983    1.949536
                      age |   .0359612   .0156974     2.29   0.022     .0051949    .0667274
                      bmi |    .122124   .0461608     2.65   0.008     .0316505    .2125975
                   female |  -.1121927   .4223499    -0.27   0.791    -.9399833     .715598
                   hsgrad |   .1365788   .4127786     0.33   0.741    -.6724527    .9456104
                          |
                  alcohol |
  Less than 3 drinks/day  |  -.2666661   .5692751    -0.47   0.640    -1.383099    .8497667
Three or more drinks/day  |  -.2682957   .6090222    -0.44   0.660    -1.462878    .9262868
                          |
                    _cons |  -5.690434   1.710711    -3.33   0.001    -9.043387   -2.337482
------------------------------------------------------------------------------------------
```

Poisson & Nbreg – Count variables. `mi impute poisson` fills in missing values of a count variable using a Poisson regression imputation method. `mi impute nbreg` fills in missing values of an overdispersed count variable using a negative binomial regression imputation method (and will usually be better than the Poisson method). This won't mean a lot to you unless/until you have some background in categorical data analysis. For now, we will just briefly note the following:

Variables that count the # of times something happens are common in the Social Sciences.

- Hausman looked at effect of R & D expenditures on # of patents received by US companies
- Grogger examined deterrent effects of capital punishment on daily homicides
- King examined effect of # of alliances on the # of nations at war
- Long looked at # of publications of scientists

Count variables are often treated as though they are continuous and the linear regression model is applied; but this can result in inefficient, inconsistent and biased estimates. Fortunately, there are many models that deal explicitly with count outcomes. These include the Poisson and the (usually superior) Negative Binomial Regression method.

These examples illustrate another feature you can use when imputing: conditional imputation. In this example, most men are coded zero for number of pregnancies they have had. But 7 men, and 3 women, have missing values on the pregnancy variable. Imputing values for men would be a bit silly, as we can be cautiously optimistic that the true value for men on # of pregnancies is zero. With the conditional imputation procedure used below, the 7 men with missing values get assigned zero while the value for # of pregnancies is imputed for the 3 women with missing values.

```
. *Poisson
. use http://www.stata-press.com/data/r13/mheartpois, clear
(Fictional heart attack data; npreg missing)

. misstable summarize
                                                            Obs<.
                                                +------------------------------
               |                                | Unique
      Variable |     Obs=.     Obs>.     Obs<.  | values        Min         Max
  -------------+--------------------------------+------------------------------
         npreg |        10                 144  |      6          0           5
  -----------------------------------------------------------------------------

. tab2 female npreg, missing

-> tabulation of female by npreg

           |                           Number of pregnancies
    Gender |         0          1          2          3          4          5          . |     Total
-----------+-----------------------------------------------------------------------------+----------
      Male |       109          0          0          0          0          0          7 |       116
    Female |        14          8          3          8          1          1          3 |        38
-----------+-----------------------------------------------------------------------------+----------
     Total |       123          8          3          8          1          1         10 |       154


. mi set mlong
. mi register imputed npreg
(10 m=0 obs. now marked as incomplete)
```

```
. mi impute poisson npreg attack smokes age bmi hsgrad, ///
>         add(20) conditional(if female==1) rseed(2232)

Univariate imputation                          Imputations =       20
Poisson regression                                   added =       20
Imputed: m=1 through m=20                          updated =        0

Conditional imputation:
  npreg: incomplete out-of-sample obs. replaced with value 0

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
             npreg |        144           10        10 |       154
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

. mi estimate: logit attack smokes age bmi female hsgrad npreg

Multiple-imputation estimates                   Imputations     =         20
Logistic regression                             Number of obs   =        154
                                                Average RVI     =     0.0183
                                                Largest FMI     =     0.1143
DF adjustment:   Large sample                   DF:     min     =    1485.04
                                                        avg     =   6.31e+09
                                                        max     =   2.15e+10
Model F test:       Equal FMI                   F(   6,243443.3)=       3.20
Within VCE type:          OIM                   Prob > F        =     0.0038

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.248284   .3609721     3.46   0.001     .5407915    1.955776
         age |   .0367254   .0154729     2.37   0.018      .006399    .0670518
         bmi |    .121511   .0459844     2.64   0.008     .0313833    .2116388
      female |  -.0904122   .5574949    -0.16   0.871    -1.183272    1.002448
      hsgrad |   .1155058   .4053936     0.28   0.776    -.6790511    .9100627
       npreg |  -.0136322    .276589    -0.05   0.961    -.5561788    .5289144
       _cons |  -5.938172   1.642572    -3.62   0.000    -9.157553   -2.718791
------------------------------------------------------------------------------
```

. *nbreg

```
. use http://www.stata-press.com/data/r13/mheartpois, clear
(Fictional heart attack data; npreg missing)

. mi set mlong
. mi register imputed npreg
(10 m=0 obs. now marked as incomplete)

. mi impute nbreg npreg attack smokes age bmi hsgrad, ///
>         add(20) conditional(if female==1) rseed(2232)

Univariate imputation                          Imputations =       20
Negative binomial regression                         added =       20
Imputed: m=1 through m=20                          updated =        0

Dispersion: mean
```

```
Conditional imputation:
  npreg: incomplete out-of-sample obs. replaced with value 0

-----------------------------------------------------------------
                   |               Observations per m
                   |---------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
             npreg |        144           10        10 |       154
-----------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)
```

. mi estimate: logit attack smokes age bmi female hsgrad npreg

```
Multiple-imputation estimates                     Imputations     =         20
Logistic regression                               Number of obs   =        154
                                                  Average RVI     =     0.0094
                                                  Largest FMI     =     0.0617
DF adjustment:   Large sample                     DF:     min     =    5047.18
                                                          avg     =   2.26e+10
                                                          max     =   9.44e+10
Model F test:       Equal FMI                     F(   6,909484.4)=       3.23
Within VCE type:          OIM                     Prob > F        =     0.0035

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.245543   .3607367     3.45   0.001     .5385126    1.952574
         age |   .0367316   .0154561     2.38   0.017     .0064383     .067025
         bmi |   .1213715   .0459979     2.64   0.008     .0312173    .2115257
      female |  -.1467292   .5452724    -0.27   0.788    -1.215486    .9220277
      hsgrad |   .1090878   .4050585     0.27   0.788    -.6848123    .9029879
       npreg |   .0268187    .269944     0.10   0.921    -.5023888    .5560262
       _cons |  -5.929065    1.64196    -3.61   0.000    -9.147247   -2.710883
------------------------------------------------------------------------------
```

Appendix B: Using Stata 12+ for Multiple Imputation for Multiple Variables

Stata 12 introduced several new procedures and commands for multiple imputation. Among these is the `mi impute chained` command, which supports multivariate Imputation using Chained Equations (ICE). ICE uses iterative procedures to impute missing values when more than one variable is missing. These variables can be of different types, e.g. they might be binary, ordinal or continuous. Variables can have an arbitrary missing-data pattern. `mi impute chained` has numerous options, and Stata warns that you should do checks to make sure the imputation is working correctly. I am just going to give a simple example adapted from the Stata Manual; you should read the whole manual and/or related literature if you want to do a more detailed analysis of your own.

> *NOTE:* Other commands for imputing multiple variables include `mi impute monotone` and `mi impute mvn`. While these can be good (or even better) than `mi impute chained`, the assumptions required to use these commands are often violated. `mi impute mvn` may be good if all your imputed variables happen to be continuous, e.g. you don't need to impute any dichotomies, but in practice you often will have mixed types of variables to impute.

First, we retrieve another version of the fictitious heart attack data, in which some data are missing for bmi and age.

```
. webuse mheart8s0, clear
(Fictional heart attack data; bmi and age missing; arbitrary pattern)

. mi describe

  Style:  mlong
          last mi update 25mar2011 11:00:38, 122 days ago

  Obs.:   complete          118
          incomplete         36  (M = 0 imputations)
          ---------------------
          total             154

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

          passive:  0

          regular:  4; attack smokes female hsgrad

          system:   3; _mi_m _mi_id _mi_miss

         (there are no unregistered variables)
```

The above shows that the data have previously been `mi set` in mlong format. bmi and age have previously been specified as variables whose missing values are to be imputed. bmi has 28 missing cases, age has 12. M = 0 means that no imputed data sets have been computed yet, i.e. you just have the original data.

```
. mi misstable patterns, frequency

   Missing-value patterns
     (1 means complete)

              |   Pattern
    Frequency |  1  2
  ------------+-------------
          118 |  1  1
              |
           24 |  1  0
            8 |  0  1
            4 |  0  0
  ------------+-------------
          154 |

  Variables are  (1) age  (2) bmi
```

In the above table, a value of 1 indicates not missing, 0 indicates missing. So, we see that there are 118 cases with non-missing values on both age and bmi. Another 24 cases are missing bmi but not age, 8 cases are missing age but not bmi, and 4 cases have missing data on both age and bmi. Next we impute missing values using the `mi impute chained` command.

```
. mi impute chained (regress) bmi age = attack smokes hsgrad female, add(20) rseed(2232)

Conditional models:
               age: regress age bmi 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 =      200
                                                burn-in =       10

               bmi: linear regression
               age: linear regression

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

The (regress) option on the command told Stata that both bmi and age were continuous and that OLS regression should be used for imputation. If, instead, the two variables were dichotomies, we would have specified (logit) instead. (We could have also mixed different types on the same command, we could have used (logit), (regress), and (ologit) for different variables if that was appropriate, see the help for `mi impute chained` for more complicated examples where different methods are mixed.) Like before, the `add` option told Stata to create 20 imputed data sets and the `rseed` option was used so we can exactly reproduce our results later.

The conditional models show us that age was regressed on every variable (both from the left and right hand side) except itself. The same is true for bmi. This is the default behavior, i.e. all variables except the one being imputed are included in the prediction equation. This will work well in many situations but there are numerous options for changing this behavior if you need more flexibility.

Having done the imputation, we can proceed as before. To get the unimputed results,

```
. mi xeq 0: logit attack smokes age bmi hsgrad female, nolog

m=0 data:
-> logit attack smokes age bmi hsgrad female, nolog

Logistic regression                               Number of obs   =        118
                                                  LR chi2(5)      =      20.89
                                                  Prob > chi2     =     0.0008
Log likelihood = -71.278532                       Pseudo R2       =     0.1278

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.404968   .4163181     3.37   0.001     .5889992    2.220936
         age |   .0381199   .0184258     2.07   0.039      .002006    .0742338
         bmi |   .1004817   .0513924     1.96   0.051    -.0002455    .2012089
      hsgrad |   .2705538   .4530665     0.60   0.550    -.6174402    1.158548
      female |   .3143023   .4777947     0.66   0.511    -.6221581    1.250763
       _cons |  -5.654463   1.879328    -3.01   0.003    -9.337879   -1.971048
------------------------------------------------------------------------------
```

The analysis is limited to the 118 cases that had complete data, i.e. we have lost almost a third of the sample (36 cases) because of missing data. With multiple imputation, the results are

```
. mi estimate: logit attack smokes age bmi hsgrad female

Multiple-imputation estimates                     Imputations     =         20
Logistic regression                               Number of obs   =        154
                                                  Average RVI     =     0.0734
                                                  Largest FMI     =     0.2627
DF adjustment:   Large sample                     DF:     min     =     286.49
                                                          avg     =   41220.53
                                                          max     =  144975.75
Model F test:       Equal FMI                     F(   5,13852.4) =       3.46
Within VCE type:          OIM                     Prob > F        =     0.0039

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.170431    .362968     3.22   0.001     .4589862    1.881875
         age |   .0382372    .015968     2.39   0.017     .0069384    .0695361
         bmi |   .1038031   .0519485     2.00   0.047     .0015538    .2060523
      hsgrad |   .1471189   .4062852     0.36   0.717    -.6492007    .9434386
      female |  -.0986277    .419447    -0.24   0.814    -.9207355      .72348
       _cons |  -5.560604   1.778105    -3.13   0.002    -9.052313   -2.068894
```

In this particular example, the coefficients and standard errors for the two imputed variables, age and bmi, change little. The other independent variables show modest changes.

Appendix C (Optional):
Approximate Do it Yourself Multiple Imputation for a Single Continuous Variable

> Warning: I am NOT recommending that you use the approach shown in this handout! I am just giving it to you for pedagogical purposes. The goal of this handout is to give you a general idea of how multiple imputation works by showing you an approximate procedure by which it can be done without using mi commands. Additional (and somewhat complicated) adjustments should be made to take into account the fact that the regression coefficients from the imputation model are themselves estimated rather than known.

This (roughly) is the formula for an imputed value when a single continuous variable has missing data and you are using regress to impute values for the missing cases.

$$\text{Imputed Value } \mathrm{X_i} = \hat{X}_i + rmse * \varepsilon_i,\ \ \varepsilon_i \sim N(0,1)$$

An alternative but equivalent formula is

$$\text{Imputed Value } \mathrm{X_i} = \hat{X}_i + \varepsilon_i,\ \ \varepsilon_i \sim N(0,\ \text{rmse})$$

Basically, the procedure for multiple imputation with a single continuous variable is as follows.

- Regress X (the continuous variable with missing values) on the other variables in the imputation model.

- Retrieve the root mean square error (rmse) also known as the standard error of the estimate.

  - Recall that the standard error of the estimate (s_e) indicates how close the actual observations fall to the predicted values on the regression line. About 68.3% of the observations should fall within ± $1s_e$ units of the regression line, 95.4% should fall within ± $2s_e$ units, and 99.7% should fall within ± $3s_e$ units. (At least, that would be true in the population.)

- For those cases where X is missing, compute X hat, i.e. the predicted value for X given the values of the other variables in the equation.

- Add random variability to the imputed X. We do this by multiplying the rmse by a random variable epsilon that has a normal (0, 1) distribution. For example, for the first case, epsilon might equal .5. For the next case, epsilon might equal -1; etc.

- Estimate your analytic model using the imputed X and the other variables in the model.

- Repeat this process M times, where M is the number of imputations. Combine the estimates from the different imputations into a single set of estimates and standard errors. I won't show you how to do this but formulas are available for this purpose.

Here is how you can do this in Stata. Again, the results are approximate. Since the regression coefficients are themselves just estimates, I ought to estimate a different bmihat and rmse for each imputation, rather than using the same values for every imputation. Nonetheless, this should give you the general idea.

We begin by estimating the imputation model for bmi. We then compute bmihat for the cases that are missing bmi. We save the rmse so we can use it in our subsequent calculations.

```
. version 12.1
. webuse mheart0, clear
(Fictional heart attack data; bmi missing)
. * Imputation model for bmi
. regress bmi attack smokes age hsgrad female

      Source |       SS       df       MS              Number of obs =     132
-------------+------------------------------           F(  5,   126) =    1.24
       Model |  99.5998228     5  19.9199646           Prob > F      =  0.2946
    Residual |  2024.93667   126   16.070926           R-squared     =  0.0469
-------------+------------------------------           Adj R-squared =  0.0091
       Total |   2124.5365   131  16.2178358           Root MSE      =  4.0089

------------------------------------------------------------------------------
         bmi |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      attack |    1.71356   .7515229     2.28   0.024     .2263179    3.200801
      smokes |  -.5153181    .761685    -0.68   0.500     -2.02267    .9920341
         age |   -.033553   .0305745    -1.10   0.275    -.0940591     .026953
      hsgrad |  -.4674308   .8112327    -0.58   0.566    -2.072836    1.137975
      female |  -.3072767   .8074763    -0.38   0.704    -1.905249    1.290695
       _cons |   26.96559   1.884309    14.31   0.000      23.2366    30.69458
------------------------------------------------------------------------------

. predict bmihat if missing(bmi)
(option xb assumed; fitted values)
(132 missing values generated)
. scalar rmse = e(rmse)
. * As shown in the output, the rmse is a little over 4. To confirm,
. display rmse
4.0088559
```

For the cases with missing data, we now generate 20 random variables, e1-e20, each of which has a normal (0, 1) distribution.

```
. * Impute the values for missing cases 20 times
. set seed 2232
. gen e = 0 if !missing(bmi)
(22 missing values generated).
* Compute 20 random error terms
. forval i = 1/20 {
  2.         quietly gen e`i' = rnormal() if missing(bmi)
  3. }
```

We now generate 20 imputed values for bmi. The imputed values = bmihat + random variability.

```
. * Compute 20 imputed values for each case
. * Imputed value = bmihat + random variation
. forval i = 1/20 {
  2.          quietly clonevar bmi`i' = bmi
  3.          quietly replace bmi`i' = bmihat + rmse * e`i' if missing(bmi)
  4. }
```

We will now show what some of the imputed values look like.

```
. * Compare the imputed values, first 3 imputations
. list bmihat bmi bmi1 bmi2 bmi3 e1 e2 e3 if missing(bmi)

     +------------------------------------------------------------------------------------+
     |   bmihat   bmi       bmi1       bmi2       bmi3          e1          e2          e3 |
     |------------------------------------------------------------------------------------|
  8. | 24.94037     .   20.26646   29.12453   23.24085   -1.165896     1.04373    -.423942 |
 11. | 23.73765     .   21.25652   22.11762   10.19627    -.618912   -.4041111   -3.377865 |
 18. |  24.9775     .   29.89542   28.12042     23.199    1.226764    .7839947   -.4436428 |
 19. | 23.67863     .   29.40268   20.53067   25.60752    1.427849    -.785253    .4811565 |
 23. |  23.9666     .    28.3487   19.97047   24.61176    1.093105   -.9968241    .1609331 |
     |------------------------------------------------------------------------------------|
 25. | 26.75463     .   25.36388   27.01864   26.36398   -.3469186    .0658563   -.0974478 |
 34. | 25.92009     .   22.23172   29.00936   27.42771   -.9200564    .7706099     .376072 |
 38. | 25.57589     .   26.58776   26.44681   22.82028    .2524079     .217249   -.6873806 |
 47. | 24.89569     .   24.81721   14.63025   29.79111   -.0195766   -2.560692     1.22115 |
 51. | 26.64985     .   31.43427   33.36421   31.75082    1.193464    1.674883    1.272426 |
     |------------------------------------------------------------------------------------|
 62. |  24.2647     .   28.86235   18.59231   27.27137    1.146874   -1.414963    .7500079 |
 64. |  24.4496     .   26.98282   25.35096   22.36778    .6319062    .2248423   -.5193049 |
 66. | 26.43693     .   20.43546    27.4691   25.57946   -1.497052    .2574717   -.2138942 |
 68. | 25.05331     .   23.13218   23.87722   22.08071   -.4792223   -.2933746   -.7415092 |
 70. | 24.45155     .   29.27277   28.11036   17.34405    1.202641     .912681   -1.772951 |
     |------------------------------------------------------------------------------------|
107. | 23.73576     .   20.52475   25.12961   30.02304   -.8009791    .3476921    1.568349 |
111. | 25.13115     .   20.08341   14.03033   31.53078   -1.259147   -2.769074    1.596376 |
116. | 25.28776     .   33.68123   21.27818   27.55823    2.093732    -1.00018    .5663627 |
122. | 24.75772     .   20.78931   18.22538   19.79584   -.9899129   -1.629478   -1.237731 |
134. | 24.01538     .   27.33225   28.59109   17.61796    .8273882    1.141402   -1.595822 |
     |------------------------------------------------------------------------------------|
141. | 25.00403     .   25.42831   16.91262   22.28859    .1058361   -2.018384   -.6773593 |
150. | 24.62909     .   25.80076   23.54937   30.08058    .2922693   -.2693351    1.359861 |
     +------------------------------------------------------------------------------------+
```

As you can see, a different value of bmi is imputed with each imputation. You can easily see where the imputed values came from. For example, for case 8, bmihat = 24.94037 and e1 (which was created to have a normal (0, 1) distribution) equals -1.165896. Since rmse = 4.0089, for case 8 bmi1 = bmihat + rmse * e1 = 24.94037 + (4.0089 * -1.165896) = 24.94037 – 4.67396 = 20.266. The 24.94037 is the point estimate for the predicted value while -4.67396 is the random variability added to the point estimate.

```
. * Compare the summary stats, first 3 imputations
. sum bmihat bmi bmi1 bmi2 bmi3 e1 e2 e3, sep(5)

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      bmihat |        22    24.92336    .9089033   23.67863   26.75463
         bmi |       132    25.24136    4.027137   17.22643   38.24214
        bmi1 |       154    25.28435    4.015896   17.22643   38.24214
        bmi2 |       154    25.02149    4.224837   14.03033   38.24214
        bmi3 |       154    25.13252    4.204955   10.19627   38.24214
-------------+--------------------------------------------------------
          e1 |        22    .1543893    1.033153  -1.497052   2.093732
          e2 |        22   -.3046026    1.222324  -2.769074   1.674883
          e3 |        22   -.1107343    1.230458  -3.377865   1.596376
```

Even though different values were imputed each time, the means and standard deviations of bmi stay about the same with each imputation. Each of the random error terms has a mean of about 0 and a standard deviation of 1, as they should. The same is true if you look at all 20 imputations.

We will now convert this into an MI data set, so we can use the `mi estimate` command. The data are currently in a wide format, i.e. we have one record per case, with several variables containing imputed values. The variable bmi contains the original values, while bmi1-bmi20 contain the imputed values. We will also treat e as imputed because its values differ with each imputation. We therefore import as wide, and then convert to the more efficient mlong format.

```
. * Convert to mi format. We will use wide, as each case now has
. * one record with several imputed variables
. mi import wide, imputed(e = e1-e20 bmi = bmi1-bmi20) clear drop
. * We can now convert to the more efficient mlong format
. mi convert mlong, clear
```

Finally, to do the analytic model,

```
. * Now do the analytic model
. mi estimate: logit attack smokes age bmi hsgrad female

Multiple-imputation estimates                   Imputations     =         20
Logistic regression                             Number of obs   =        154
                                                Average RVI     =     0.0562
                                                Largest FMI     =     0.2277
DF adjustment:   Large sample                   DF:     min     =     379.99
                                                        avg     =  148236.68
                                                        max     =  509974.85
Model F test:       Equal FMI                   F(   5,22572.1) =       3.61
Within VCE type:          OIM                   Prob > F        =     0.0029

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.228637   .3653374     3.36   0.001     .5125445    1.944729
         age |   .0359155   .0154973     2.32   0.020     .0055414    .0662897
         bmi |   .1143556   .0513896     2.23   0.027     .0133119    .2153993
      hsgrad |   .1773898   .4080986     0.43   0.664    -.6224833    .9772628
      female |  -.0972318    .418408    -0.23   0.816    -.9172998    .7228363
       _cons |  -5.758118   1.772189    -3.25   0.001    -9.236463   -2.279773
------------------------------------------------------------------------------
```

Note that, even though we didn't do the imputations exactly right, we get results that are very similar to the correct results we got earlier in the main part of this handout. (I can't guarantee that this will always be true though). Repeating the earlier results,

```
Imputations (20):
  .........10.........20 done

Multiple-imputation estimates                     Imputations     =         20
Logistic regression                               Number of obs   =        154
                                                  Average RVI     =     0.0404
                                                  Largest FMI     =     0.1678
DF adjustment:   Large sample                     DF:     min     =     694.17
                                                          avg     =  115477.35
                                                          max     =  287682.25
Model F test:       Equal FMI                     F(   5,43531.9) =       3.74
Within VCE type:          OIM                     Prob > F        =     0.0022

------------------------------------------------------------------------------
      attack |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      smokes |   1.239172   .3630877     3.41   0.001     .5275236    1.950821
         age |   .0354929   .0154972     2.29   0.022     .0051187     .065867
         bmi |   .1184188   .0495676     2.39   0.017     .0210985    .2157391
      hsgrad |    .185709   .4075301     0.46   0.649    -.6130435    .9844615
      female |  -.0996102   .4193583    -0.24   0.812    -.9215408    .7223204
       _cons |  -5.845855    1.72309    -3.39   0.001    -9.225542   -2.466168
------------------------------------------------------------------------------
```

Conclusion. Again, I am not recommending do it yourself multiple imputation (unless maybe you are stuck with software that lacks built-in routines). But, by looking at this appendix, multiple imputation should be a little less magical to you. Taking into account the rmse, i.e. how much variability there can reasonably be about the predicted value for a case with missing data, you can generate a series of imputed values which will give you reasonable estimates of the coefficients and the standard errors.

Incidentally, we didn't actually need to compute the e variables. Here is a slightly simpler coding that uses the second variation of the imputed value formula presented earlier. The results are identical.

```
***********************
*** Alternative coding; no need to generate the e vars
* MD Part 3: Approximate do it yourself multiple imputation
version 12.1
webuse mheart0, clear
* Imputation model for bmi
regress bmi attack smokes age hsgrad female
predict bmihat if missing(bmi)
scalar rmse = e(rmse)
* As shown in the output, the rmse is a little over 4. To confirm,
display rmse
```

```
* Impute the values for missing cases 20 times
set seed 2232
* Compute 20 imputed values for each case
* Imputed value = bmihat + random variation
forval i = 1/20 {
       quietly clonevar bmi`i' = bmi
       quietly replace bmi`i' = bmihat + rnormal(0, rmse) if missing(bmi)
}
* Compare the imputed values, first 3 imputations
list bmihat bmi bmi1 bmi2 bmi3  if missing(bmi)
* Compare the summary stats, first 3 imputations
sum bmihat bmi bmi1 bmi2 bmi3 , sep(5)

* Convert to mi format. We will use wide, as each case now has
* one record with several imputed variables
mi import wide, imputed(bmi = bmi1-bmi20) clear drop
* We can now convert to the more efficient mlong format
mi convert mlong, clear

* Now do the analytic model
mi estimate: logit attack smokes age bmi hsgrad female
```

Appendix D (Optional): Full information Maximum Likelihood

As Newman (2003, p. 334) notes, "MI [multiple imputation] is a procedure by which missing data are imputed several times (e.g. using regression imputation) to produce several different complete-data estimates of the parameters. The parameter estimates from each imputation are then combined to give an overall estimate of the complete-data parameters as well as reasonable estimates of the standard errors." Maximum Likelihood (ML) approaches "operate by estimating a set of parameters that maximize the probability of getting the data that was observed" (Newman, p. 332).

Paul Allison has some excellent blog entries on the use of Full Information Maximum Likelihood (FIML) as a technique for dealing with missing data:

http://statisticalhorizons.com/ml-better-than-mi

http://statisticalhorizons.com/ml-is-better-than-mi

He also has a working paper at

http://statisticalhorizons.com/wp-content/uploads/MissingDataByML.pdf

Key advantages of FIML compared to MI that he mentions are

- ML is simpler to implement (if you have the right software).
- Unlike multiple imputation, ML has no potential incompatibility between an imputation model and an analysis model.
- ML produces a deterministic result rather than a different result every time.

FIML is simpler in that all you generally have to do is specify it as an option to your estimation command. You don't have to first create multiple imputations and you don't have to worry about the imputation and estimation models being congenial. The results you get won't be dependent on the way you set a random number seed.

"Right software" is a non-trivial matter. MPlus may be the best program if you want to use fiml. In Stata 14, you can use fiml with the `sem` command, but only for a limited range of conditions. `sem` estimates linear models; it can't estimate things like logistic regression models. When using the `sem` command to use fiml you specify `method(mlmv)`. Here is an excerpt from the sem manual (Section Intro 4):

> For method MLMV to perform what might seem like magic, joint normality of all variables is assumed and missing values are assumed to be missing at random (MAR). MAR means either that the missing values are scattered completely at random throughout the data or that values more likely to be missing than others can be predicted by the variables in the model. Method MLMV formally requires the assumption of joint normality of all variables, both observed and latent. If your observed variables do not follow a joint normal distribution, you may be better off using ML, QML, or ADF and simply omitting observations with missing values.

In his paper and blogs Allison suggests that violations of the multivariate normality assumption may not be that important, meaning that it is hopefully ok to have things like binary independent variables.

To illustrate the use of FIML, the following is adapted from example 2 of the mi estimate section of the Stata 14 manual. I will compare the results when sem and OLS regression are run on the original data that has missing values. I will then compare the results you get using multiple imputation and FIML.

```
. * Adapted from Example 2 of the mi estimate chapter of the Stata 14 MI manual
. webuse mhouses1993s30, clear
(Albuquerque Home Prices Feb15-Apr30, 1993)

. * Regular Regression with missing data
. mi xeq 0: regress price tax sqft age nfeatures ne custom corner

m=0 data:
-> regress price tax sqft age nfeatures ne custom corner

      Source |       SS           df       MS      Number of obs   =        66
-------------+----------------------------------   F(7, 58)        =     51.86
       Model |  9164658.61         7  1309236.94   Prob > F        =    0.0000
    Residual |  1464105.15        58  25243.1922   R-squared       =    0.8623
-------------+----------------------------------   Adj R-squared   =    0.8456
       Total |  10628763.8        65  163519.442   Root MSE        =    158.88

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         tax |   .4988701   .1584853     3.15   0.003     .1816273    .8161128
        sqft |   .3522184   .0957476     3.68   0.001     .1605588    .5438779
         age |  -.5650817   2.002529    -0.28   0.779     -4.57358    3.443416
   nfeatures |   4.389607   18.55499     0.24   0.814    -32.75223    41.53145
          ne |  -17.38534   47.27462    -0.37   0.714    -112.0158     77.2451
      custom |   174.9411   53.72371     3.26   0.002     67.40139    282.4808
      corner |  -73.58234   49.13007    -1.50   0.140    -171.9269    24.76218
       _cons |    92.7448    101.607     0.91   0.365    -110.6438    296.1334
------------------------------------------------------------------------------

. * Corresponding sem model with missing data
. sem (price <- tax sqft age nfeatures ne custom corner) if _mi_m ==0, nolog
(51 observations with missing values excluded)

[some output deleted]

Structural equation model                       Number of obs     =         66
Estimation method  = ml
Log likelihood     = -1803.2121

------------------------------------------------------------------------------
             |                 OIM
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Structural   |
  price <-   |
         tax |   .4988701   .1485699     3.36   0.001     .2076784    .7900617
        sqft |   .3522184   .0897573     3.92   0.000     .1762973    .5281394
         age |  -.5650817   1.877245    -0.30   0.763    -4.244414     3.11425
```

```
   nfeatures |   4.389607   17.39413     0.25   0.801    -29.70226    38.48147
          ne |  -17.38534   44.31697    -0.39   0.695     -104.245    69.47433
      custom |   174.9411   50.36258     3.47   0.001     76.23223    273.6499
      corner |  -73.58234   46.05634    -1.60   0.110    -163.8511    16.68642
       _cons |    92.7448   95.25019     0.97   0.330    -93.94214    279.4317
-------------+----------------------------------------------------------------
 var(e.price)|   22183.41   3861.636                      15770.78    31203.51
------------------------------------------------------------------------------
LR test of model vs. saturated: chi2(0)   =      0.00, Prob > chi2 =      .
```

As we would expect, the coefficients are identical in the two approaches, with very slight differences in the standard errors and significance tests. Note that there are only 66 cases, since 51 of the other cases have some missing data.

```
. * Multiple Imputation Model
. mi estimate: regress price tax sqft age nfeatures ne custom corner

Multiple-imputation estimates                   Imputations       =         30
Linear regression                               Number of obs     =        117
                                                Average RVI       =     0.0648
                                                Largest FMI       =     0.2533
                                                Complete DF       =        109
DF adjustment:   Small sample                   DF:     min       =      69.12
                                                        avg       =      94.02
                                                        max       =     105.51
Model F test:       Equal FMI                   F(   7,  106.5)   =      67.18
Within VCE type:          OLS                   Prob > F          =     0.0000

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         tax |   .6768015   .1241568     5.45   0.000     .4301777    .9234253
        sqft |   .2118129    .069177     3.06   0.003     .0745091    .3491168
         age |   .2471445   1.653669     0.15   0.882    -3.051732    3.546021
   nfeatures |   9.288033   13.30469     0.70   0.487    -17.12017    35.69623
          ne |   2.518996   36.99365     0.07   0.946    -70.90416    75.94215
      custom |   134.2193   43.29755     3.10   0.002     48.35674    220.0818
      corner |  -68.58686    39.9488    -1.72   0.089    -147.7934    10.61972
       _cons |   123.9118   71.05816     1.74   0.085    -17.19932    265.0229
------------------------------------------------------------------------------

. * Correspondign SEM model using fiml
. sem (price <- tax sqft age nfeatures ne custom corner) if _mi_m ==0, method(mlmv) nolog
note: Missing values found in observed exogenous variables. Using the noxconditional
behavior. Specify the
      forcexconditional option to override this behavior.

[Some output deleted]
Structural equation model                       Number of obs     =        117
Estimation method  = mlmv
Log likelihood     = -2972.8657

--------------------------------------------------------------------------------------
                     |                 OIM
                     |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
---------------------+----------------------------------------------------------------
Structural           |
  price <-           |
                 tax |   .6610302   .1360297     4.86   0.000     .3944168    .9276436
                sqft |   .2197247    .075478     2.91   0.004     .0717906    .3676588
                 age |  -.0334027   2.095612    -0.02   0.987    -4.140726    4.073921
           nfeatures |   8.518805    13.1256     0.65   0.516     -17.2069    34.24451
```

```
                ne |   5.868605   36.37018     0.16   0.872    -65.41564    77.15285
            custom |    133.974    41.9925     3.19   0.001     51.67024    216.2778
            corner |  -69.66022   38.59107    -1.81   0.071    -145.2973    5.976884
             _cons |   129.3903   72.05301     1.80   0.073      -11.831    270.6116
-------------------+----------------------------------------------------------------

[extraneous output deleted]
```

At least in this case, FIML and MI produced very similar results. (Coefficients that did differ were coefficients with very large standard errors.) In this case the imputed data sets had already been created but if that were not the case you would have had to do a lot of other work first.

It would be nice if more of Stata commands (e.g. `regress`) added an `mlmv` option. Nicer still would be if Stata could catch up with MPlus and support FIML in non-linear models. If all you want to do is run linear regressions though, you may want to consider using `sem` commands with `mlmv` rather than using multiple imputation. Or do both and see if the results differ much.

For more examples using FIML with linear models, see

https://www3.nd.edu/~rwilliam/dynamic/mi_xtdpdml.pdf

https://www3.nd.edu/~rwilliam/dynamic/SJPaper.pdf

Appendix E (Optional): Allison's "Multiple Imputation for Missing Data: A Cautionary Tale"

In a 2000 Sociological Methods and Research paper entitled "Multiple Imputation for Missing Data: A Cautionary Tale" Allison summarizes the basic rationale for multiple imputation:

> Multiple imputation (MI) appears to be one of the most attractive methods for general- purpose handling of missing data in multivariate analysis. The basic idea, first proposed by Rubin (1977) and elaborated in his (1987) book, is quite simple:
>
> 1. Impute missing values using an appropriate model that incorporates random variation.
> 2. Do this <u>M</u> times producing <u>M</u> "complete" data sets.
> 3. Perform the desired analysis on each data set using standard complete-data methods.
> 4. Average the values of the parameter estimates across the <u>M</u> samples to produce a single point estimate.
> 5. Calculate the standard errors by (a) averaging the squared standard errors of the <u>M</u> estimates (b) calculating the variance of the <u>M</u> parameter estimates across samples, and (c) combining the two quantities using a simple formula.

Allison adds that

> Multiple imputation has several desirable features:
>
> • Introducing appropriate random error into the imputation process makes it possible to get approximately unbiased estimates of all parameters. No deterministic imputation method can do this in general settings.
>
> • Repeated imputation allows one to get good estimates of the standard errors. Single imputation methods don't allow for the additional error introduced by imputation (without specialized software of very limited generality).

With regards to the assumptions needed for MI, Allison says that

- First, the data must be missing at random (MAR), meaning that the probability of missing data on a particular variable Y can depend on other observed variables, but not on Y itself (controlling for the other observed variables).
    - Example: Data are MAR if the probability of missing income depends on marital status, but within each marital status, the probability of missing income does not depend on income; e.g. single people may be more likely to be missing data on income, but low income single people are no more likely to be missing income than are high income single people.

- Second, the model used to generate the imputed values must be "correct" in some sense.

- Third, the model used for the analysis must match up, in some sense, with the model used in the imputation.

- The problem is that it's easy to violate these conditions in practice. There are often strong reasons to suspect that the data are not MAR. Unfortunately, not much can be done about this. While it's possible to formulate and estimate models for data that are not MAR, such models are complex, untestable, and require specialized software. Hence, any general-purpose method will necessarily invoke the MAR assumption.