SOCY7706: Longitudinal Data Analysis
Instructor: Natasha Sarkisian

Missing Data in Longitudinal Studies

In most datasets (not only longitudinal ones), we will encounter the issue of missing data. In cross-sectional datasets, it stems from the problem of item non-response -- for various reasons, respondents often leave particular items blank on questionnaires or decline to give any response during interviews. Sometimes the portion of such missing data can be quite sizeable. This is a serious problem, and the more data points are missing in a dataset, the more likely it is that you will need to address the problem of incomplete cases.

Types of missing data

The most appropriate way to handle missing or incomplete data will depend upon how data points became missing. Little and Rubin (1987) define three unique types of missing data mechanisms.

Missing Completely at Random (MCAR):
MCAR data exists when missing values are randomly distributed across all observations. In this case, observations with complete data are indistinguishable from those with incomplete data. That is, whether the data point on Y is missing is not at all related to the value of Y or to the values of any Xs in that dataset. E.g. if you are asking people their weight in a survey, some people might fail to respond for no good reason – i.e. their nonresponse is in no way related to what their actual weight is, and is also not related to anything else we might be measuring.

MCAR missing data often exists because investigators randomly assign research participants to complete only some portions of a survey instrument – GSS does that a lot, asking respondents various subsets of questions. MCAR can be confirmed by dividing respondents into those with and without missing data, then using t-tests of mean differences on income, age, gender, and other key variables to establish that the two groups do not differ significantly. But in real life, MCAR assumption is too stringent for most situations other than such random assignment.

Missing at Random (MAR):
MAR data exist when the observations with incomplete data differ from those with complete data, but the pattern of data missingness on Y can be predicted from other variables in the dataset (Xs) and beyond that bears no relationship to Y itself – i.e., whatever nonrandom processes existed in generating the missing data on Y can be explained by the rest of the variables in the dataset. MAR assumes that the actual variables where data are missing are not the cause of the incomplete data -- instead, the cause of the missing data is due to some other factor that we also measured. E.g., one sex may be less likely to disclose its weight.

MAR is much more common than MCAR. MAR data are assumed by most methods of dealing with missing data. It is often but not always tenable. Importantly, the more relevant and related predictors we can include in statistical models, the more likely it is that the MAR assumption will be met. Sometimes, if the data that we already have are not sufficient to make our data

MAR, we can try to introduce external data as well – e.g., estimating income based on Census block data associated with the address of the respondent.

If we can assume that data are MAR, the best methods to deal with the missing data issue are multiple imputation and raw maximum likelihood methods. Together, MAR and MCAR are called ignorable missing data patterns, although that's not quite correct as sophisticated methods are still typically necessary to deal with them.

Not Missing at Random (NMAR or nonignorable):
The pattern of data missingness is non-random and it is not predictable from other variables in the dataset. NMAR data arise due to the data missingness pattern being explainable only by the very variable(s) on which the data are missing. E.g., heavy (or light) people may be less likely to disclose their weight. NMAR data are also sometimes described as having selection bias. NMAR data are difficult to deal with, but sometimes that's unavoidable; if the data are NMAR, we need to model the missing-data mechanism. Two approaches used for that are selection models and pattern mixture; however, we will not deal with them here.

Data Screening: Examining Patterns of Missing Data

It is a good idea to look at both the data and the codebook and/or questionnaire. I will use an example from GSS 2012 dataset:

```
. use https://www.sarkisian.net/socy7706/gss2012.dta
```

When examining missing data, the first thing is to make sure you know how the missing data were coded and take such codes into account when you do any recoding. It is also important to distinguish two main types of missing data – sometimes questions are not applicable and therefore not asked, but in other situations, questions are asked but not answered. It is very important to distinguish not applicable cases because those often would be cases that you might not want to include in the analyses or sometimes you might want to assign a certain value to them (e.g. if someone is not employed, their hours of work might be missing because that question was not relevant, but in fact we do know that it should be zero. Sometimes, however, datasets code some cases "not applicable" because a respondent has refused to answer some prior case – although coded not applicable, these cases are more likely to be an equivalent of "not answered" – i.e. truly missing data. "Don't know" is often a tough category – sometimes, on ordinal scales measuring opinions, you might be able to place them as the middle category, but in other situations, it becomes missing data.

For example, in our GSS2012 data, for the agekdbrn variable, some people refused to answer that question while many others were not applicable because they don't have children. We can see that in the questionnaire:

CHILDS: Categorical (Single)
How many children have you ever had? Please count all that were born alive at any time (including any you had from a previous marriage).

Categories:

{none}	NONE
{one}	ONE
{two}	TWO
{three}	THREE
{four}	FOUR
{five}	FIVE
{six}	SIX
{seven}	SEVEN
{eight_or_more}	EIGHT OR MORE
{dontknow}	DON'T KNOW
{refused}	REFUSED

If CHILDS.ContainsAny({one,two,three,four,five,six,seven,eight_or_more}) Then

AGEKDBRN: Long [10 .. 96]
How old were you when your first child was born?

RECORD AGE:

End If

Let's check that in the data:

```
. tab agekdbrn if childs==0
no observations

. tab agekdbrn if childs>0 & childs<., m

    R'S AGE |
   WHEN 1ST |
 CHILD BORN |      Freq.     Percent        Cum.
------------+-----------------------------------
         13 |          1        0.07        0.07
         14 |          7        0.49        0.56
         15 |         20        1.39        1.95
         16 |         30        2.09        4.04
         17 |         68        4.74        8.78
         18 |        101        7.04       15.82
         19 |        102        7.11       22.93
         20 |        110        7.67       30.59
         21 |        134        9.34       39.93
         22 |         78        5.44       45.37
         23 |         95        6.62       51.99
         24 |         82        5.71       57.70
         25 |         94        6.55       64.25
         26 |         70        4.88       69.13
         27 |         78        5.44       74.56
         28 |         58        4.04       78.61
         29 |         52        3.62       82.23
         30 |         55        3.83       86.06
         31 |         32        2.23       88.29
         32 |         32        2.23       90.52
         33 |         28        1.95       92.47
         34 |         29        2.02       94.49
         35 |         21        1.46       95.96
         36 |         13        0.91       96.86
         37 |         11        0.77       97.63
         38 |         10        0.70       98.33
         39 |          7        0.49       98.82
         40 |          6        0.42       99.23
         41 |          3        0.21       99.44
         42 |          1        0.07       99.51
         46 |          1        0.07       99.58
         50 |          1        0.07       99.65
          . |          5        0.35      100.00
------------+-----------------------------------
      Total |      1,435      100.00

. tab agekdbrn if childs==.
no observations
```

Let's create a new variable that will distinguish true missing from skip pattern missing values:

```
. gen agekdbrn_n=agekdbrn
(544 missing values generated)

. replace agekdbrn_n=.n if childs==0
(536 real changes made, 536 to missing)
```

We will also look at a measure of individual health: there seem to be two variables, health and health1, both have quite a few missing values. Let's crosstab them and then create a combined variable:

```
. tab health health1, m

 CONDITION |                      RS HEALTH IN GENERAL
 OF HEALTH | Excellent  Very good       Good       Fair       Poor          . |     Total
-----------+------------------------------------------------------------------+----------
 excellent |       140         31          1          3          0        175 |       350
      good |         6        103        179          6          0        304 |       598
      fair |         1          8         28         90          2        146 |       275
      poor |         0          0          0          7         27         49 |        83
         . |       107        204        207        104         36         10 |       668
-----------+------------------------------------------------------------------+----------
     Total |       254        346        415        210         65        684 |     1,974

. clonevar healthall=health
(668 missing values generated)

. replace healthall=1 if health1==1 & health==.
(107 real changes made)

. replace healthall=2 if (health1==2 | health1==3) & health==.
(411 real changes made)

. replace healthall=3 if health1==4 & health==.
(104 real changes made)

. replace healthall=4 if health1==5 & health==.
(36 real changes made)

. tab healthall, m

  CONDITION |
  OF HEALTH |      Freq.     Percent        Cum.
------------+-----------------------------------
```

```
  excellent |        457       23.15       23.15
       good |      1,009       51.11       74.27
       fair |        379       19.20       93.47
       poor |        119        6.03       99.49
          . |         10        0.51      100.00
------------+-----------------------------------
      Total |      1,974      100.00
```

Once you differentiated between truly missing data and the results of skip patterns, you should examine patterns of missing data. We will use the rest of the variables from our example model for agekdbrn except I will substitute hhrace instead of race (that's household race variable that actually has missing data) and add health variable.

```
. misstable summarize agekdbrn_n educ sex age born marital sibs hhrace healthall, all
showzeros
                                                               Obs<.
                                                +------------------------------
               |                                | Unique
      Variable |     Obs=.     Obs>.     Obs<.  | values        Min         Max
  -------------+--------------------------------+------------------------------
    agekdbrn_n |         8       536     1,430  |     32         13          50
          educ |         2         0     1,972  |     21          0          20
           sex |         0         0     1,974  |      2          1           2
           age |         5         0     1,969  |     72         18          89
          born |         3         0     1,971  |      2          1           2
       marital |         0         0     1,974  |      5          1           5
          sibs |         3         0     1,971  |     23          0          30
        hhrace |         5         0     1,969  |      5          1           5
     healthall |        10         0     1,964  |      4          1           4
  -----------------------------------------------------------------------------
```

```
. misstable patterns agekdbrn_n educ sex age born marital sibs hhrace healthall, freq
exok asis

         Missing-value patterns
           (1 means complete)

              |   Pattern
    Frequency |  1  2  3  4    5  6  7
  ------------+-----------------------
        1,943 |  1  1  1  1    1  1  1
              |
            8 |  1  1  1  1    1  1  0
            6 |  0  1  1  1    1  1  1
            5 |  1  1  0  1    1  1  1
            5 |  1  1  1  1    1  0  1
            2 |  1  1  1  1    0  1  1
            1 |  0  0  1  0    1  1  0
            1 |  0  1  1  1    0  1  1
            1 |  1  0  1  1    1  1  1
            1 |  1  1  1  0    1  1  0
            1 |  1  1  1  0    1  1  1
  ------------+-----------------------
        1,974 |

  Variables are  (1) agekdbrn_n  (2) educ  (3) age  (4) born  (5) sibs  (6) hhrace
(7) healthall
```

asis specifies that the order of the variables in the table be the same as the order in which they are specified on the misstable command. (The default is to sort by the number of missing values.)

freq specifies that the table should report frequencies instead of percentages.

exok specifies that the extended missing values .a, .b, ..., .z should be treated as if they do not designate missing. This allows to treat “not applicable” or deliberately skipped cases as distinct from truly missing.

Patterns of missing data across time points in a longitudinal dataset

In a longitudinal dataset, you may also want to understand patterns of attrition as well as patterns of item nonresponse across time for the same individuals. Here’s an example of exploring that. For the overall attrition, it’s definitely easier to explore that in a wide dataset:

```
. use "https://www.sarkisian.net/socy7706/hrs_hours.dta", clear

. for num 1/9: egen missX=rowmiss( rXworkhours80 rXpoorhealth rXmarried rXtotalpar
rXsiblog hXchildlg rXallparhelptw)
->  egen miss1=rowmiss( r1workhours80 r1poorhealth r1married r1totalpar r1siblog
h1childlg r1allparhelptw)

->  egen miss2=rowmiss( r2workhours80 r2poorhealth r2married r2totalpar r2siblog
h2childlg r2allparhelptw)

->  egen miss3=rowmiss( r3workhours80 r3poorhealth r3married r3totalpar r3siblog
h3childlg r3allparhelptw)

->  egen miss4=rowmiss( r4workhours80 r4poorhealth r4married r4totalpar r4siblog
h4childlg r4allparhelptw)

->  egen miss5=rowmiss( r5workhours80 r5poorhealth r5married r5totalpar r5siblog
h5childlg r5allparhelptw)

->  egen miss6=rowmiss( r6workhours80 r6poorhealth r6married r6totalpar r6siblog
h6childlg r6allparhelptw)

->  egen miss7=rowmiss( r7workhours80 r7poorhealth r7married r7totalpar r7siblog
h7childlg r7allparhelptw)

->  egen miss8=rowmiss( r8workhours80 r8poorhealth r8married r8totalpar r8siblog
h8childlg r8allparhelptw)

->  egen miss9=rowmiss( r9workhours80 r9poorhealth r9married r9totalpar r9siblog
h9childlg r9allparhelptw)

. for num 1/9: gen attX=1 if missX~=7

->  gen att1=1 if miss1~=7

->  gen att2=1 if miss2~=7
(259 missing values generated)

->  gen att3=1 if miss3~=7
(422 missing values generated)

->  gen att4=1 if miss4~=7
(471 missing values generated)

->  gen att5=1 if miss5~=7
(581 missing values generated)
```

```
->  gen att6=1 if miss6~=7
(636 missing values generated)

->  gen att7=1 if miss7~=7
(691 missing values generated)

->  gen att8=1 if miss8~=7
(764 missing values generated)

->  gen att9=1 if miss9~=7
(819 missing values generated)

. misstable patterns att*, freq

          Missing-value patterns
            (1 means complete)

              |   Pattern
    Frequency |  1  2  3  4    5  6  7  8
  ------------+--------------------------
        5,540 |  1  1  1  1    1  1  1  1
              |
          154 |  1  0  0  0    0  0  0  0
          137 |  0  0  0  0    0  0  0  0
           84 |  1  1  1  0    0  0  0  0
           81 |  1  1  1  1    0  0  0  0
           73 |  1  1  1  1    1  1  1  0
           69 |  1  1  0  0    0  0  0  0
           55 |  1  1  1  1    1  1  0  0
           49 |  1  1  1  1    1  0  0  0
           28 |  0  1  1  1    1  1  1  1
           27 |  1  1  1  0    1  1  1  1
           19 |  1  0  1  1    1  1  1  1
           15 |  1  1  1  1    1  1  0  1
           12 |  1  1  1  1    1  0  1  1
           11 |  0  1  0  0    0  0  0  0
           10 |  0  0  1  1    1  1  1  1
           10 |  1  1  0  0    1  1  1  1
           10 |  1  1  0  1    1  1  1  1
           10 |  1  1  1  1    0  1  1  1
            7 |  0  0  0  0    1  1  1  1
            7 |  1  0  1  1    1  0  0  0
            6 |  0  0  1  0    0  0  0  0
            6 |  1  0  0  1    1  1  1  1
            6 |  1  0  1  1    0  0  0  0
            6 |  1  1  1  1    1  0  0  1
            5 |  0  0  0  0    0  1  1  1
            5 |  0  0  0  1    1  1  1  1
            5 |  0  1  1  1    0  0  0  0
            5 |  1  1  1  0    1  0  0  0
            5 |  1  1  1  1    0  0  0  1
            5 |  1  1  1  1    0  0  1  1
            4 |  1  1  1  0    1  1  1  0
            4 |  1  1  1  1    0  1  0  0
            3 |  0  0  1  1    1  1  0  0
            3 |  0  1  1  1    1  1  1  0
            3 |  1  0  0  0    1  1  1  1
            3 |  1  0  1  1    0  1  1  1
            3 |  1  0  1  1    1  1  0  0
            3 |  1  1  0  0    1  1  0  0
            3 |  1  1  0  1    0  0  0  0
            3 |  1  1  0  1    1  0  0  0
            2 |  0  0  0  0    1  1  0  0
```

```
2  |   0   0   1   0     1   1   1   1
2  |   0   1   1   0     0   0   0   0
2  |   0   1   1   1     0   1   1   1
2  |   0   1   1   1     1   0   0   0
2  |   1   0   0   0     0   0   0   1
2  |   1   0   0   0     0   0   1   1
2  |   1   0   0   0     1   0   0   0
2  |   1   0   1   1     1   0   0   1
2  |   1   0   1   1     1   1   1   0
2  |   1   1   0   0     0   0   1   1
2  |   1   1   0   0     0   1   1   1
2  |   1   1   0   0     1   1   1   0
2  |   1   1   0   1     0   0   0   1
2  |   1   1   1   0     0   1   1   1
2  |   1   1   1   0     1   1   0   1
2  |   1   1   1   1     0   1   1   0
1  |   0   0   0   0     0   0   1   1
1  |   0   0   0   0     0   1   1   0
1  |   0   0   0   1     0   0   0   0
1  |   0   0   0   1     0   1   1   0
1  |   0   0   0   1     1   0   0   0
1  |   0   0   1   0     0   1   0   0
1  |   0   0   1   0     0   1   1   1
1  |   0   0   1   0     1   0   0   0
1  |   0   0   1   0     1   1   0   0
1  |   0   0   1   0     1   1   0   1
1  |   0   0   1   1     0   0   0   0
1  |   0   0   1   1     1   0   1   1
1  |   0   0   1   1     1   1   0   1
1  |   0   0   1   1     1   1   1   0
1  |   0   1   0   0     0   1   1   1
1  |   0   1   0   0     1   0   0   0
1  |   0   1   0   1     0   0   0   0
1  |   0   1   0   1     1   1   0   0
1  |   0   1   0   1     1   1   1   0
1  |   0   1   0   1     1   1   1   1
1  |   0   1   1   0     0   1   0   0
1  |   0   1   1   0     0   1   1   1
1  |   0   1   1   0     1   1   0   0
1  |   0   1   1   0     1   1   1   1
1  |   0   1   1   1     0   0   1   0
1  |   0   1   1   1     0   0   1   1
1  |   0   1   1   1     0   1   0   0
1  |   0   1   1   1     1   0   1   1
1  |   0   1   1   1     1   1   0   0
1  |   1   0   0   0     0   0   1   0
1  |   1   0   0   0     0   1   0   0
1  |   1   0   0   0     0   1   1   1
1  |   1   0   0   0     1   0   0   1
1  |   1   0   0   0     1   1   0   0
1  |   1   0   0   0     1   1   1   0
1  |   1   0   0   1     0   0   0   0
1  |   1   0   0   1     0   0   1   0
1  |   1   0   0   1     1   0   0   0
1  |   1   0   0   1     1   1   0   0
1  |   1   0   0   1     1   1   0   1
1  |   1   0   0   1     1   1   1   0
1  |   1   0   1   0     0   0   0   0
1  |   1   0   1   0     0   0   0   1
1  |   1   0   1   0     0   0   1   0
1  |   1   0   1   0     1   1   0   1
1  |   1   0   1   0     1   1   1   0
1  |   1   0   1   0     1   1   1   1
```

```
             1 |  1  0  1  1    0  1  0  1
             1 |  1  0  1  1    0  1  1  0
             1 |  1  1  0  0    0  1  0  0
             1 |  1  1  0  0    1  0  0  1
             1 |  1  1  0  0    1  0  1  1
             1 |  1  1  0  1    1  0  0  1
             1 |  1  1  0  1    1  1  0  0
             1 |  1  1  0  1    1  1  0  1
             1 |  1  1  1  0    0  0  0  1
             1 |  1  1  1  0    0  0  1  0
             1 |  1  1  1  0    0  0  1  1
             1 |  1  1  1  0    0  1  1  0
             1 |  1  1  1  0    1  0  1  1
             1 |  1  1  1  0    1  1  0  0
             1 |  1  1  1  1    0  1  0  1
             1 |  1  1  1  1    1  0  1  0
  ------------+--------------------------
         6,591 |
  Variables are  (1) att2  (2) att3  (3) att4  (4) att5  (5) att6  (6) att7  (7) att8
                 (8) att9
```

One important note: when examining attrition, it is often important to distinguish between non-participation and deaths, because, when respondents die, we do not consider it as a missing data problem, as their further outcome data doesn't exist when they are dead. Here, I do not have indicators of deaths in the dataset, but the full HRS dataset certainly contains this information, so we could further differentiate among cases. However, we don't want to exclude those who died from a study altogether either, because that kind of selection can bias the results (e.g., you will be focusing on the more healthy individuals). Some alternative approaches include assigning the "worst" value to the outcome variable for those who die during follow-up, or including an indicator of having been alive at next point of measurement. Both nonparticipation and deaths may introduce selection bias, however, if the reason for is related to the outcome of interest, however; researchers use multiple imputation (discussed below) and weighing techniques to explore the impact of attrition. There can be a lot of complexity in terms of nonparticipation, deaths, and item non-response; here's a good example:

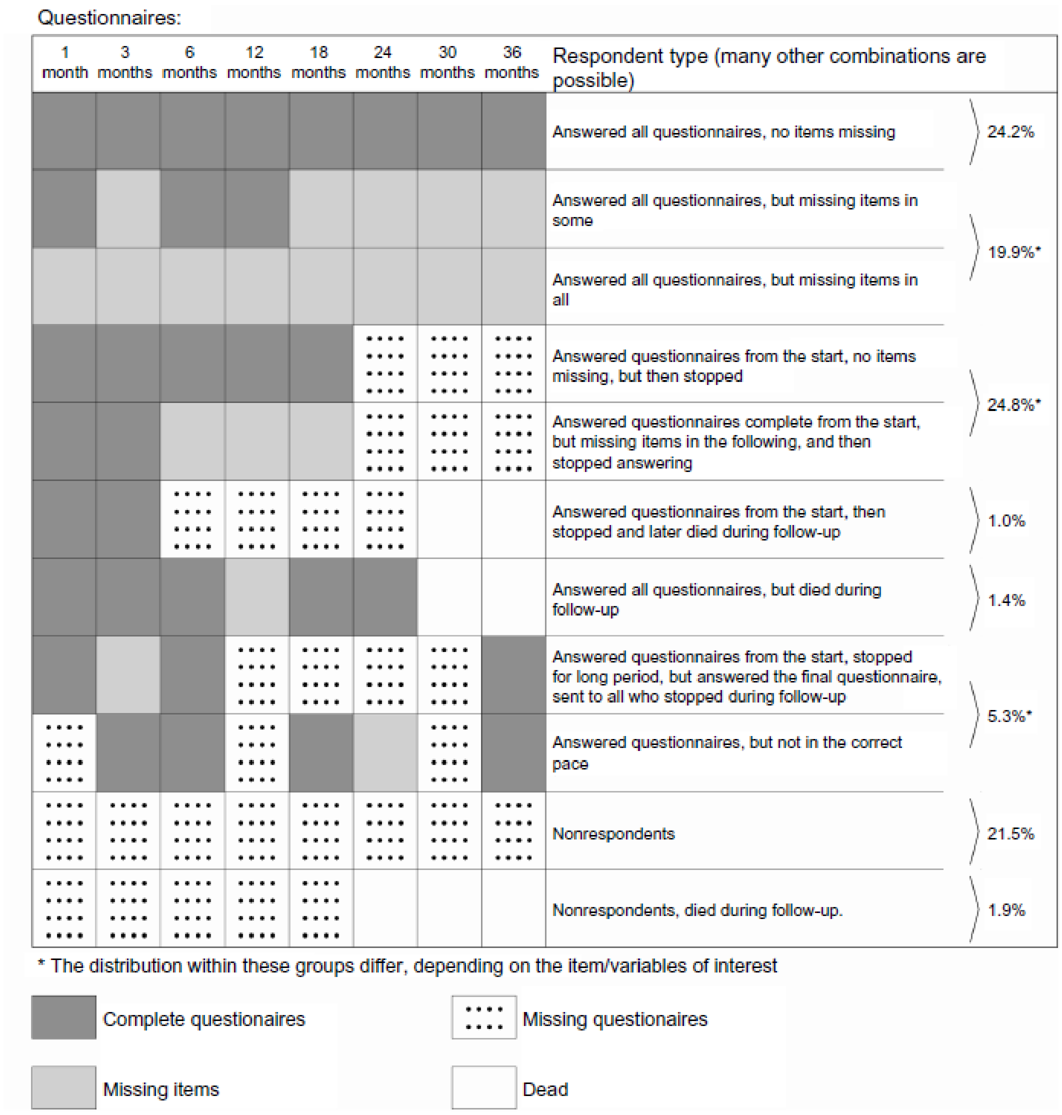

Questionnaires:
1 month
3 months
6 months
12 months
18 months
24 months
30 months
36 months
Respondent type (many other combinations are possible)
Answered all questionnaires, no items missing
24.2%
Answered all questionnaires, but missing items in some
Answered all questionnaires, but missing items in all
19.9%*
Answered questionnaires from the start, no items missing, but then stopped
Answered questionnaires complete from the start, but missing items in the following, and then stopped answering
24.8%*
Answered questionnaires from the start, then stopped and later died during follow-up
1.0%
Answered all questionnaires, but died during follow-up
1.4%
Answered questionnaires from the start, stopped for long period, but answered the final questionnaire, sent to all who stopped during follow-up
Answered questionnaires, but not in the correct pace
5.3%*
Nonrespondents
21.5%
Nonrespondents, died during follow-up.
1.9%
* The distribution within these groups differ, depending on the item/variables of interest
Complete questionaires
Missing questionaires
Missing items
Dead

Figure 1 Exemplified overview of some respondent types in a follow up study of PCI patients.

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

Next, we turn to the long dataset to explore the amount of item non-response per person (we could also do it in a wide dataset).

```
. use "https://www.sarkisian.net/socy7706/hrs_hours_reshaped.dta", clear
```

In the long format, we first generate a count of missing values per line (which is one person-year):

```
. egen miss=rowmiss( rworkhours80 rpoorhealth rmarried rtotalpar rsiblog hchildlg
rallparhelptw female age minority raedyrs)

. tab miss

       miss |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     30,541       55.86       55.86
          1 |     15,034       27.50       83.35
          2 |      1,436        2.63       85.98
          3 |        143        0.26       86.24
          4 |          7        0.01       86.26
          5 |          3        0.01       86.26
          6 |      7,506       13.73       99.99
          7 |          6        0.01      100.00
------------+-----------------------------------
      Total |     54,676      100.00
```

0 means no missing data at all in a given person-year; 7 means 7 out of 11 variables are missing data (probably all time-varying ones). Next, I generate an average number of missing values per year for each person (so an average of this count across time points for each person):

```
. bysort hhidpn: egen misstotal=mean(miss)
```

It makes sense to examine this per person since it's the average across years for each person. I'll mark one record per person (the first one) using the new variable I'll create, firstrecord. I could also create the same variable using tag function of egen, i.e. : egen taggedobs=tag(hhidpn)

```
. bysort hhidpn: gen firstrecord=1 if _n==1
(48,085 missing values generated)
```

Now I will tabulate misstotal with one observation per person:

```
. tab misstotal if firstrecord==1

  misstotal |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,228       18.63       18.63
   .1111111 |        378        5.74       24.37
       .125 |         15        0.23       24.59
   .1428571 |         18        0.27       24.87
   .1666667 |         11        0.17       25.03
         .2 |         25        0.38       25.41
   .2222222 |        354        5.37       30.78
        .25 |         38        0.58       31.36
   .2857143 |         12        0.18       31.54
   .3333333 |        379        5.75       37.29
       .375 |         17        0.26       37.55
         .4 |         21        0.32       37.87
   .4285714 |         14        0.21       38.08
   .4444444 |        338        5.13       43.21
         .5 |        102        1.55       44.76
   .5555556 |        330        5.01       49.76
   .5714286 |         16        0.24       50.01
         .6 |         17        0.26       50.27
       .625 |         22        0.33       50.60
   .6666667 |        479        7.27       57.87
   .7142857 |         13        0.20       58.06
        .75 |         30        0.46       58.52
   .7777778 |        410        6.22       64.74
         .8 |          9        0.14       64.88
   .8333333 |          6        0.09       64.97
   .8571429 |          5        0.08       65.04
       .875 |          8        0.12       65.16
   .8888889 |        129        1.96       67.12
          1 |        184        2.79       69.91
   1.111111 |         86        1.30       71.22
      1.125 |          5        0.08       71.29
   1.142857 |          2        0.03       71.32
   1.166667 |          4        0.06       71.39
        1.2 |          2        0.03       71.42
   1.222222 |         78        1.18       72.60
       1.25 |          8        0.12       72.72
   1.285714 |          1        0.02       72.74
   1.333333 |        146        2.22       74.95
      1.375 |          3        0.05       75.00
   1.428571 |          2        0.03       75.03
   1.444444 |         71        1.08       76.10
        1.5 |         19        0.29       76.39
   1.555556 |         69        1.05       77.44
   1.571429 |          1        0.02       77.45
        1.6 |          4        0.06       77.51
      1.625 |          1        0.02       77.53
   1.666667 |         74        1.12       78.65
   1.714286 |          3        0.05       78.70
       1.75 |          2        0.03       78.73
   1.777778 |         46        0.70       79.43
   1.833333 |          1        0.02       79.44
   1.857143 |          1        0.02       79.46
   1.888889 |         47        0.71       80.17
          2 |        110        1.67       81.84
   2.111111 |         45        0.68       82.52
   2.166667 |          1        0.02       82.54
        2.2 |          1        0.02       82.55
   2.222222 |         40        0.61       83.16
   2.333333 |         35        0.53       83.69
        2.4 |          1        0.02       83.71
   2.444444 |         32        0.49       84.19
        2.5 |          2        0.03       84.22
   2.555556 |          8        0.12       84.34
   2.666667 |        116        1.76       86.10
   2.777778 |         44        0.67       86.77
   2.888889 |         39        0.59       87.36
          3 |         48        0.73       88.09
   3.111111 |         16        0.24       88.33
   3.333333 |        128        1.94       90.27
   3.444444 |         36        0.55       90.82
        3.5 |          1        0.02       90.84
   3.555556 |         29        0.44       91.28
   3.666667 |         19        0.29       91.56
   3.777778 |          5        0.08       91.64
   3.888889 |          2        0.03       91.67
          4 |        181        2.75       94.42
   4.111111 |         40        0.61       95.02
   4.222222 |          6        0.09       95.11
   4.333333 |         11        0.17       95.28
   4.444445 |          3        0.05       95.33
   4.555555 |          2        0.03       95.36
   4.666667 |        154        2.34       97.69
   4.777778 |          9        0.14       97.83
   4.888889 |          8        0.12       97.95
          5 |          3        0.05       98.00
   5.111111 |          2        0.03       98.03
   5.222222 |          3        0.05       98.07
   5.333333 |        117        1.78       99.85
   5.444445 |         10        0.15      100.00
------------+-----------------------------------
      Total |      6,591      100.00
```

Next, I will calculate how many waves of complete data each person has. I will use the missing data indicator I already created and generate a dichotomy based on that which marks the complete waves (1 if miss is 0, 0 otherwise):

```
. tab miss

       miss |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     30,541       55.86       55.86
          1 |     15,034       27.50       83.35
          2 |      1,436        2.63       85.98
          3 |        143        0.26       86.24
          4 |          7        0.01       86.26
          5 |          3        0.01       86.26
          6 |      7,506       13.73       99.99
          7 |          6        0.01      100.00
------------+-----------------------------------
      Total |     54,676      100.00

. gen nomiss=(miss==0)

. tab nomiss

     nomiss |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     24,135       44.14       44.14
          1 |     30,541       55.86      100.00
------------+-----------------------------------
      Total |     54,676      100.00
```

And now I will calculate a count of years of non-missing data per person:

```
. bysort hhidpn: egen nomisstotal=total(nomiss)

. tab nomisstotal if firstrecord==1

nomisstotal |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        348        5.28        5.28
          1 |        445        6.75       12.03
          2 |        963       14.61       26.64
          3 |        886       13.44       40.08
          4 |        794       12.05       52.13
          5 |        660       10.01       62.15
          6 |        601        9.12       71.26
          7 |        537        8.15       79.41
          8 |        542        8.22       87.63
          9 |        815       12.37      100.00
------------+-----------------------------------
      Total |      6,591      100.00
```

So 815 people have all complete data for all 9 waves, 348 people have no complete waves, etc.

Methods of handling missing data

We will briefly address various naïve methods of dealing with missing data (that are no longer recommended), and learn to apply one of the more sophisticated techniques – multiple imputation by chained equations.

I. Available data approaches

1. Listwise (casewise) deletion
If an observation has missing data for any one variable used in a particular analysis, we can omit that observation from the analysis. This approach is the default method of handling incomplete data in Stata, as well as most other commonly-used statistical software.

There is no simple decision rule for whether to drop cases with missing values, or to impute values to replace missing values. Listwise deletion will produce unbiased results if the data are MCAR (but our sample will be smaller so the standard errors will be higher). When the data are MAR, listwise deletion produced biased results but they are actually less problematic than the results of many other common naïve methods of handling missing data. For instance, if the patterns of missing data on your independent variables are not related to the values of the dependent variables, listwise deletion will produce unbiased estimates.

Still, dropping cases with missing data can reduce our sample (and therefore also reduce the precision) substantially, and therefore we often want to avoid it. But when the number of cases with missing data is small (e.g., less than 5% in large samples), it is common simply to drop these cases from analysis.

2. Pairwise deletion
We can compute bivariate correlations or covariances for each pair of variables X and Y using all cases where neither X nor Y is missing – i.e., based upon the available pairwise data. To estimate means and variances of each of the variables, it uses all cases where that variable is not missing. We can then use these means and covariances in subsequent analyses.

Pairwise data deletion is available in a number of SAS and SPSS statistical procedures; Stata does not use it much and for a good reason – pairwise deletion produces biased results and shouldn't be used.

3. Missing data indicators
In this method, we would create a dummy variable equal to 1 in cases where X is missing, and 0 in cases where it is not. Then we would include both X and the dummy variable in the model predicting Y. This is method is very problematic and results in biased estimates.

II. Deterministic imputation methods

1. Mean substitution
The simplest imputation method is to use a variable's mean (or median) to fill in missing data values. This is only appropriate when the data are MCAR, and even then this method creates a spiked distribution at the mean in frequency distributions, lowers the correlations between the imputed variables and the other variables, and underestimates variance. Nevertheless, it is made available as an easy option in many SPSS procedures, and there is a procedure (STANDARD) available for that in SAS. Still, you should avoid using it.

A type of mean substitution where the mean is calculated in a subgroup of the non-missing values, rather than all of them, is also sometimes used; this technique also suffers from the same problems.

2. Single regression imputation (a.k.a. conditional mean substitution)
A better method than to impute using each variable's mean is to use regression analysis on cases without missing data, and then use those regression equations to predict values for the cases with missing data. This imputation technique is available in many statistical packages (for example, in Stata there is "impute" command). This technique still has the problem that all cases with the same values on the independent variables will be imputed with the same value on the missing variable, thus leading to an underestimate of variance; thus, the standard errors in your models will be lower than they should be.

3. Single random (stochastic) regression imputation
To improve upon the single regression imputation method, and especially to compensate for its tendency to lower the variance and therefore lead to an underestimation of standard errors, we can add uncertainty to the imputation of each variable so each imputed case would get a different value. This is done by adding a random value to the predicted result. This random value is usually the regression residual from a randomly selected case from the set of cases with no missing values. SPSS offers stochastic regression imputation – when doing regression imputation, SPSS by default adds the residual of a randomly picked case to each estimate. Impute command in Stata does not offer such an option, but one can use ice command we will learn soon to generate such imputations.

Single random regression imputation is better than regular regression imputation because it preserves the properties of the data both in terms of means and in terms of variation. Still, this residual is just a guess and it is likely that standard errors will be smaller than they should be. Another remaining problem, but it's a serious one, is that it uses imputed data as if they were real – it doesn't allow for the variation between different possible sets of imputed values. That's why we need to move beyond the traditional approaches to those that try to recognize the difference between real and imputed data.

4. Hot deck imputation
As opposed to regression imputation, hotdeck imputation is a nonparametric imputation technique (i.e., doesn't depend on estimating regression parameters). Hot deck imputation involves identifying the most similar case to the case with a missing value and substituting that most similar case's value for the missing value. We need to specify which variables are used to define such similarity – these variables should be related to the variable that's being imputed. Thus, a number of categorical variables are used to form groups, and then cases are randomly picked within those groups. For example:

Obs	Var 1	Var 2	Var 3	Var 4
1	4	1	2	3
2	5	4	2	5
3	3	4	2	.

Hot deck imputation examines the observations with complete records (obs 1 and 2) and substitutes the value of the most similar observation for the missing data point. Here, obs 2 is more similar to obs 3 than obs 1. New data matrix:

Obs	Var 1	Var 2	Var 3	Var 4
1	4	1	2	3
2	5	4	2	5
3	3	4	2	5

After doing this imputation, we analyze the data using the complete database. Stata offers a hot deck algorithm implemented in the hotdeck command. This procedure will tabulate the missing data patterns for the selected variables and will define a row of data with missing values in any of the variables as a `missing line' of data, similarly a `complete line' is one where all the variables contain data. The hotdeck procedure replaces the variables in the `missing lines' with the corresponding values in the `complete lines'. It does so within the groups specified by the "by" variables. Note that if a dataset contains many variables with missing values then it is possible that many of the rows of data will contain at least one missing value. The hotdeck procedure will not work very well in such circumstances. Also, hotdeck procedure assumes is that the missing data are MAR and that the probability of missing data can be fully accounted for by the categorical variables specified in the `by' option.

Hotdeck imputation allows imputing with real, existing values (so categorical variables remain categorical and continuous variables remain continuous). But it can be difficult to define "similarity." Also, once again this approach does not distinguish between real and imputed data and therefore will result in standard errors that are too low.

II. Maximum likelihood methods

The second group of methods we will consider are those based on maximum likelihood estimation. There are two types of techniques in this group.

1. Expectation Maximization (EM) approach:

EM approach is a technique uses ML algorithm to generate a covariance matrix and mean estimates given the available data, and then these estimates can be used in further analyses. All the estimates are obtained through an iterative procedure, and each iteration has two steps. First, in the expectation (E) step, we take estimates of the variances, covariances and means, perhaps from listwise deletion, use these estimates to obtain regression coefficients, and then fill in the missing data based on those coefficients. In the maximization (M) step, having filled in missing data, we use the complete data (using estimated values) to recalculate variances, covariances, and means. These are substituted back into the E step. The procedure iterates through these two steps until convergence is obtained (convergence occurs when the change of the parameter estimates from iteration to iteration becomes negligible). At that point we have maximum likelihood estimates of variances, covariances, and means, and we can use those to make the maximum likelihood estimates of the regression coefficients. Note that the actual imputed data are not generated in this process; only the parameter estimates.

The SPSS Missing Values Analysis (MVA) module uses the EM approach to missing data handling, and it's also available in SAS as SAS-MI; as far as I know, it is not available in Stata.

The strength of the approach is that it has well-known statistical properties and it generally outperforms available data methods and deterministic methods. The main disadvantage is that it adds no uncertainty component to the estimated data. Thus, it still underestimates the standard errors of the coefficients.

<u>2. Direct ML methods</u>
There are alternative maximum likelihood estimators that are better than the ones obtained by the EM algorithm; these involve direct ML method, also known as raw maximum likelihood method, or Full Information Maximum Likelihood estimation (FIML). This technique uses all available data to generate maximum likelihood-based statistics. Like EM algorithm, direct ML methods assume the missing values are MAR. Under an unrestricted mean and covariance structure, direct ML and EM return identical parameter estimate values. Unlike EM, however, direct ML can be employed in the context of fitting user-specified linear models. Direct ML methods are *model-based*, that is, implemented as part of a fitted statistical model. This produces standard errors and parameter estimates under the assumption that the fitted model is not false, so parameter estimates and standard errors are model-dependent. But it also makes it difficult to use variables that are not in the model in your imputation.

Direct ML has the advantage of convenience/ease of use and well-known statistical properties. Unlike EM, it also allows for the direct computation of appropriate standard errors and test statistics. Disadvantages include an assumption of joint multivariate normality of the variables used in the analysis and the lack of an imputed dataset produced by the analysis.

Direct ML method is implemented by the EM algorithm in the SPSS Missing Values option and MIXED procedure in SAS. It is also available in structural equation modeling packages, such as AMOS, LISREL, MPlus, as well as Stata SEM module.

<u>III. Multiple imputation</u>

An alternative to the maximum likelihood is called multiple imputation (MI). In multiple imputation we generate imputed values on the basis of existing data, but we do the imputation more than once, each time including a random component in our imputation. This allows us to prevent underestimating the standard errors of our regression coefficients. We do that by combining coefficient estimates from multiple datasets using special formulas for the standard errors. Specifically, the standard error of such estimates is equal to SQRT[(1 - 1/m)/B +W], where m is the number of replicates, B is the variance of the imputations, and W is the average of the estimated variance.

Multiple imputation is a sort of an approximation to Direct ML. In multiple imputation, we try a *few* plausible values of missing data. In maximum likelihood, we integrate over *all* possible data values, giving more weight to values that are more plausible. MI has the advantage of simplicity over Maximum Likelihood methods, making it particularly suitable for large datasets. The efficiency of MI is high even when the number of imputed datasets is low (3-10), although recent literature suggests that this depends on the amount of missing data--larger amount may necessitate more imputed datasets.

<u>1. Multiple hot deck imputation</u>
Multiple hot deck imputation combines the well-known statistical advantages of EM and direct ML with the ability of hot deck imputation to provide a raw data matrix. The primary difference between multiple hot deck imputation and regular hot deck imputation is that multiple imputation requires that we generate five to ten datasets with imputed values. We then analyze each database and summarize the results into one summary set of findings. Stata offers multiple hot deck imputation as a part of the same hotdeck command we discussed earlier.

<u>2. Multiple Imputation under the Multivariate Normal Model</u>

In this approach, multiple datasets are generated with methods somewhat similar to single random regression imputation, but with some important modifications. To impute the missing values, we use the information from available data to generate a distribution of plausible values for the missing data, and draw from that distribution at random multiple times to produce multiple datasets. The imputed value is affected by two sources of random variation:
(1) It is a random draw from a conditional distribution.
(2) The conditional distribution itself must be estimated, and the estimation contains some error.

Under this method, the model for the missing data given the observed is a fully specified joint model (e.g. multivariate normal). This is difficult to specify for a mixture of continuous and categorical data. Therefore this method assumes all data are normal. But luckily, tests suggest that this type of imputation is quite robust even when the simulation is based on an erroneous model (e.g., when normality is assumed even though the underlying data are not in fact normal). In Stata, it is available as a part of the MI package: see mi impute mvn. Since this method assumes that all variables are continuous (even though it is fairly robust to that violation), we will focus on learning another method, also implemented in Stata, that does not make that assumption.

<u>3. Multiple Imputation by Chained Equations (MICE)</u>
Chained equations method has a similar idea but adopts a different approach. No joint distribution is set up. Rather, we use a series of conditional distributions. E.g. if we have a continuous X, a count variable Y, and a binary Z, and some data for each are missing, we set up (1) a linear regression of X on Y and Z, (2) a Poisson regression of Y on X and Z, and (3) a logistic regression of Z on X and Y. We start by fitting (1) to the observed data, then simulate any missing X from that model. Then we fit (2) using observed Z and X (with missing values filled out by the simulations), and simulate any missing Y. Then we fit (3) using X and Y (with missing values filled by the simulations). We go through multiple iterations, fitting each model in turn, and updating the simulations with each iteration, waiting for the model to converge. We do that multiple times producing multiple datasets.

MICE is computationally simple to implement, and is available in Stata. The drawback is that the conditionals may not specify a unique joint distribution which can make the inferences problematic; but the simulation studies suggest it often works quite well, so it is increasingly used.

A few rules crucial for implementing any type of MI:

- All variables included in your models should be included in the imputation process; you can also add auxiliary variables that would improve predictions for some variables included in the model.
- Dependent variable should always be included in the imputation process, but there are different opinions on whether their imputed values should be subsequently discarded (this is called MID—multiple imputation then deletion) – for a while, the consensus was that MID was better than regular MI (i.e., that it is better to drop the imputed values of the dependent variable), but recent work suggests that the two approaches produce equivalent results. What is clear is that it is definitely beneficial to keep imputed values of the dependent variable in a situation when additional variables were used in imputing this dependent variable—such supplementary variables enhance the imputation of the dependent variable but are not included in final data analysis models. In this case, imputed values of the dependent variable contain extra information and should definitely be preserved. (To better understand this situation, see: von Hippel, Paul T. 2007. Regression with Missing Ys: An Improved Strategy for Analyzing Multiply Imputed Data. *Sociological Methodology 37(1),* 83-117.)
- If you have interactions or nonlinear relationships among the variables, you should create the variables for those before doing the imputations – otherwise the imputed values will only reflect the linear relationships among the variables. There is some disagreement, however, whether those interactions and nonlinear terms should be imputed separately (this is known as "Just Another Variable" approach) or be a product of imputations of main terms.
- If you are using categorical variables as sets of dummies, you need to create single variables representing those sets before doing the imputation, even if you only plan to use them as dummies later on.
- It is helpful to transform skewed variables before imputing, then back-transform them for analyses.
- Only "soft" missing data (those denoted by just a dot) get imputed – "hard" missing data – e.g., .a, .b, etc. – do not get imputed. Only "soft" missing data (those denoted by just a dot) get imputed – "hard" missing data – e.g., .a, .b, etc. – do not get imputed. In some versions of the command (mi impute chained rather than ice), hard missing data will present problems when imputing other values and you need to decide how to handle them – either exclude these observations from imputation, or allow for them to be soft missing and imputed and then delete, or exclude this variable from the imputation of the others, or replace with a value of 0 and include a separate dummy indicating the not applicable group (e.g., put 0 for marital satisfaction for the unmarried and include a married/unmarried dichotomy).
- Another issue to consider when using multiple imputation is the number of datasets to impute. The larger the amount of missing data, the more datasets you'd need to avoid loss of efficiency. The most common choice these days seems to be 20. You can consult the following article for the info on the number of imputations: Graham, J. W., A. E.

Olchowski, and T. D. Gilreath. 2007. How many imputations are really needed? Some practical clarifications of multiple imputation theory. *Prevention Science 8(3)*: 206-213.

- If you want to do exploratory analyses, generate an extra imputed dataset specifically for that part of the analysis; then, for the final results, run your model on a number of other imputed datasets, not including the preliminary one.
- Multiple imputation of a longitudinal dataset should be done on a wide rather than long dataset (i.e., there should be separate variables for each year of data, rather than multiple observations within each case). That will allow you to use observations from one year to impute data in another year, which will improve the quality of prediction as well as account for the clustered nature of the data. To change between long and wide format of data in Stata, you can use reshape command (or mi reshape after imputation).
- For a longitudinal dataset, additional problems will be created by attrition—loss of cases over time for various reasons. It is possible to estimate models even if you have different number of time points for different cases, but oftentimes it is useful to conduct sensitivity analyses and check how the results would look like if those cases that are missing entirely at certain time points are imputed (either under MAR assumption or using, for example, sensitivity testing approaches for NMAR; see below).
- If you plan to use mixed effects models with random slopes, you should try to include interactions of level 2 ID dummies with each of the independent variables that should have random slopes, although this will likely be too many variables, especially if you have multiple random slopes. (You might want to explore the issue of which random slopes you want to include using listwise deletion dataset in order to limit the number of interactions to include.) Alternatively, you may want to use a specialized R package for multilevel data.

Learning to use MICE

MICE is available both within the mi module in Stata as well as a separate package. I will mostly demonstrate its use with the separate package but also introduce the mi module version (mi impute chained).

For a separate module, you find the command using

```
. net search mice
```

You need to install st0067_4 from http://www.stata-journal.com/software/sj9-3

Here's the syntax:

```
ice mainvarlist [if] [in] [weight] [, ice_major_options ice_less_used_options]

    ice_major_options             description
    ------------------------------------------------------------------------
    clear                         clears the original data from memory and
                                    loads the imputed dataset into memory
    dryrun                        reports the prediction equations -- no
                                    imputations are done
    eq(eqlist)                    defines customized prediction equations
    m(#)                          defines the number of imputations
    match[(varlist)]              prediction matching for each member of
                                    varlist
    passive(passivelist)          passive imputation
    saving(filename [, replace])  imputed and nonimputed variables are
                                    stored to filename
    ------------------------------------------------------------------------

ice_less_used_options         description
    ------------------------------------------------------------------------
    boot[(varlist)]               estimates regression coefficients for
                                    varlist in a bootstrap sample
    by(varlist)                   imputation within the levels implied by
                                    varlist
    cc(varlist)                   prevents imputation of missing data in
                                    observations in which varlist has a
                                    missing value
    cmd(cmdlist)                  defines regression command(s) to be used
                                    for imputation
    conditional(condlist)         conditional imputation
    cycles(#)                     determines number of cycles of regression
                                    switching
    dropmissing                   omits all observations not in the
                                    estimation sample from the output
    eqdrop(eqdroplist)            removes variables from prediction
                                    equations
    genmiss(string)               creates missingness indicator variable(s)
    id(newvar)                    creates newvar containing the original
                                    sort order of the data
    interval(intlist)             imputes interval-censored variables
    monotone                      assumes pattern of missingness is
                                    monotone, and creates relevant
                                    prediction equations
    noconstant                    suppresses the regression constant
    nopp                          suppresses special treatment of perfect
                                    prediction
    noshoweq                      suppresses presentation of prediction
                                    equations
    noverbose                     suppresses messages showing the progress
                                    of the imputations
    nowarning                     suppresses warning messages
    on(varlist)                   imputes each member of mainvarlist
                                    univariately
    orderasis                     enters the variables in the order given
    persist                       ignores errors when trying to impute
                                    "difficult" variables and/or models
    restrict([varname] [if])      fits models on a specified subsample,
                                    impute missing data for entire
                                    estimation sample
    seed(#)                       sets random-number seed
    substitute(sublist)           substitutes dummy variables for multilevel
                                    categorical variables
    trace(trace_filename)         monitors convergence of the imputation
                                    algorithm
    ------------------------------------------------------------------------
```

Sets of imputed and nonimputed variables are stored to a new file specified using saving option. Option replace permits filename to be overwritten with new data. Any number of complete imputations may be created (defined by option m). Let's see an example, and then we'll examine a few options. We will use a small, NLSY-based dataset on marriage and employment. Since this dataset is longitudinal, as mentioned above, we should impute it in wide format.

```
. use https://www.sarkisian.net/socy7706/marriage.dta

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

```
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                    72972   ->    6081
Number of variables                  11   ->      65
j variable (12 values)             year   ->   (dropped)
xij variables:
                                 interv   ->   interv83 interv84 ... interv94
                                    mar   ->   mar83 mar84 ... mar94
                                   educ   ->   educ83 educ84 ... educ94
                                    emp   ->   emp83 emp84 ... emp94
                                  enrol   ->   enrol83 enrol84 ... enrol94
-----------------------------------------------------------------------------
```

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

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

=> xi: ice mar83 mar84 mar85 mar86 mar87 mar88 mar89 mar90 mar91 mar92 mar93 mar94
educ83 educ84 educ8
> 5 educ86 educ87 educ88 educ89 educ90 educ91 educ92 educ93 educ94 emp83 emp84 emp85
emp86 emp87 emp88
>  emp89 emp90 emp91 emp92 emp93 emp94 enrol83 enrol84 enrol85 enrol86 enrol87 enrol88
enrol89 enrol90
>  enrol91 enrol92 enrol93 enrol94 birthdate race i.race parpres pared,
cmd(race:mlogit) substitute(ra
> ce:i.race) saving(L:\socy7706\marriage_imputed.dta, replace) m(5) dryrun

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

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

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

```
           |         | birthdate _Irace_2 _Irace_3 parpres pared
[output omitted]

   _Irace_2 |         | [Passively imputed from (race==2)]
   _Irace_3 |         | [Passively imputed from (race==3)]
    parpres | regress | mar83 mar84 mar85 mar86 mar87 mar88 mar89 mar90 mar91
           |         | mar92 mar93 mar94 educ83 educ84 educ85 educ86 educ87
           |         | educ88 educ89 educ90 educ91 educ92 educ93 educ94 emp83
           |         | emp84 emp85 emp86 emp87 emp88 emp89 emp90 emp91 emp92
           |         | emp93 emp94 enrol83 enrol84 enrol85 enrol86 enrol87
           |         | enrol88 enrol89 enrol90 enrol91 enrol92 enrol93
           |         | enrol94 birthdate _Irace_2 _Irace_3 pared
      pared | regress | mar83 mar84 mar85 mar86 mar87 mar88 mar89 mar90 mar91
           |         | mar92 mar93 mar94 educ83 educ84 educ85 educ86 educ87
           |         | educ88 educ89 educ90 educ91 educ92 educ93 educ94 emp83
           |         | emp84 emp85 emp86 emp87 emp88 emp89 emp90 emp91 emp92
           |         | emp93 emp94 enrol83 enrol84 enrol85 enrol86 enrol87
           |         | enrol88 enrol89 enrol90 enrol91 enrol92 enrol93
           |         | enrol94 birthdate _Irace_2 _Irace_3 parpres
------------------------------------------------------------------------------

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

Note that we specified race as m.race – that tells Stata that we want to use mlogit to impute race and that we want to use race as a set of dummies when imputing other variables. If we want to use ologit instead to impute a variable, we would use o. instead of m. prefix. This is the best way to use multicategory variables in imputation. If you have a number of dummies instead (e.g., if we had separate dummies for black and Hispanic), it is a good idea to generate a multicategory variable first and use that variable rather than separate dummies in the imputation process—otherwise, your dummies will be imputed independently and therefore can end up overlapping. Some options for ice (see help ice for more):

cmd option: This option allows us to specify what command we want to use to impute a given variable. The default cmd for a variable is logit when there are two distinct values, mlogit when there are 3-5 and regress otherwise. Note that unless the dataset is large, use of the mlogit command may produce unstable estimates if the number of levels is too large.

For example, if we want educ to be used in other imputations as continuous but to be imputed using ordered logit, we can specify:

```
. ice mar* educ* emp* enrol*  birthdate m.race parpres pared,
saving(L:\socy7706\marriage_imputed.dta, replace) m(5) dryrun cmd(educ*: ologit)
```

eq(eqlist) option: This option allows one to define customized prediction equations for any subset of variables. The default is that each variable in mainvarlist with any missing data is imputed from all the other variables in mainvarlist. The option allows great flexibility in the possible imputation schemes. The syntax of eqlist is varname1:varlist1 [,varname2:varlist2 ...], where each varname# (or varlist#) is a member (or subset) of mainvarlist. It is important to ensure that each equation is sensible. ice places no restrictions except to check that all variables mentioned are indeed in mainvarlist. For example, we could specify:

```
ice mar*  educ* emp* enrol*  birthdate m.race parpres pared,
saving(L:\socy7706\marriage_imputed.dta, replace) eq(mar*: emp* educ* birthdate
_Irace_2 _Irace_3) m(5) dryrun
```

genmiss(name) option: This option creates an indicator variable for the missingness of data in any variable in mainvarlist for which at least one value has been imputed. The indicator variable is set to missing for observations excluded by if, in, etc. The indicator variable for xvar is named namexvar (of course, you get to specify the actual name).

Passive option: If we are using interactions, for example, between x1 and x2 (e.g., x12=x1*x2), we can use passive option, passive(x12:x1*x2). For example:

```
. ice mar* educ* emp* enrol*  birthdate m.race parpres pared paprespared,
saving(L:\socy7706\marriage_imputed.dta, replace) m(5) dryrun passive(parprespared:
parpres*pared)
```

However, some people recently have been arguing that interactions and nonlinear terms should not be imputed passively and instead should be imputed separately – this is known as "Just Another Variable" approach.

Persist option: If we are using a lot of dummies, we are likely to run into errors when running imputation, especially in logit-based models. You can tell Stata to ignore those errors by specifying "persist" option; however, the resulting imputation would be problematic. You might want to use it to see which variables keep creating errors and then exclude those variables from corresponding equations using eqdrop option.

Eqdrop option: This option specifies which variables to omit from certain equations. You should omit as little as possible in order to make imputation run. The syntax of eqdroplist is varname1:varlist1 [, varname2:varlist2 ...]. For example:

```
. ice mar*  educ* emp* enrol*  birthdate m.race parpres pared,
saving(L:\socy7706\marriage_imputed.dta, replace) eqdrop(emp*: birthdate  _Irace_2
_Irace_3, mar*: pared parpres) m(5) dryrun
```

Conditional option: This option specifies criteria for inclusion/exclusion of cases during imputation; it is useful when your variable should only have values for a subset for the whole sample. For example:

```
. ice mar* marsat* educ* emp* emphrs* enrol*  birthdate m.race parpres pared,
saving(marriage_imputed.dta, replace) m(5) dryrun conditional(emphrs: emp=1 \
marsat: mar==1)
```

Interval option: This option can be used to limit the range of a variable during imputation (because it should not be top or bottom coded afterwards!):

```
gen paredll=pared
gen paredul=pared
replace paredll=0 if pared==.
replace paredul=20 if pared==.
ice mar* educ* emp* enrol*  birthdate m.race parpres pared paredll paredul,
saving(L:\socy7706\marriage_imputed.dta, replace) m(5) interval(pared: paredll
paredul)
```

Seed option: This option allows to make a fixed starting point for random number generation, so that every time you run that command, you get exactly the same imputed datasets rather than different ones every time. For example:

```
ice mar* marsat* educ* emp* emphrs* enrol*  birthdate m.race parpres pared,
saving(marriage_imputed.dta, replace) m(5) dryrun seed(1234)
```

If your dataset is large and your model includes many variables, you might run into problems running imputation in Intercooled Stata (Stata IC). In that case, try Stata SE (that's what we are using on Citrix anyways). Note that imputation is often a pretty slow process, especially if your dataset is large. And you often have to keep adjusting the imputation model until it runs without errors.

Since our dry run looked good, we can generate imputed data:

```
. ice mar* educ* emp* enrol*  birthdate m.race parpres pared,
saving(L:\socy7706\marriage_imputed.dta, replace) m(5) genmiss(mv_)
[Output omitted]
Imputing
[Perfect prediction detected: using auglogit to impute enrol84]
..........1
[Perfect prediction detected: using auglogit to impute enrol84]
..........2
[Perfect prediction detected: using auglogit to impute enrol84]
..........3
[Perfect prediction detected: using auglogit to impute enrol84]
..........4
[Perfect prediction detected: using auglogit to impute enrol84]
..........5
(note: file L:\socy7706\marriage_imputed.dta not found)
file L:\socy7706\marriage_imputed.dta saved
```

<u>Stata built-in module</u>

To do the same thing with the Stata built-in module:

```
. mi set wide
. mi register imputed mar* educ* emp* enrol*  race parpres pared
. mi register regular birthdate


. mi impute chained (logit) mar* emp* enrol* (mlogit) race (reg) educ* parpres
pared = birthdate, add(5) augment
variable ident not found
    Your mi data are xtset and some of the variables previously declared by
xtset are not in the
    dataset.  mi verifies that none of the xtset variables are also
registered as imputed or
    passive.  Type mi xtset, clear to clear old no-longer-valid settings.
r(111);

. mi xtset, clear

. mi impute chained (logit) mar* emp* enrol* (mlogit) race (reg) educ*
parpres pared = birthdate, add(5) augment
```

```
Conditional models:
              race: mlogit race pared parpres i.enrol92 i.enrol94 i.enrol84
i.enrol89 i.enrol91 i.enrol90 i.enrol93 i.enrol83 i.enrol85 i.enrol87
i.enrol88 i.enrol86 i.emp83 i.emp85 i.emp84 educ83 educ90 educ92 i.emp93
i.emp87 educ84 educ93 educ85 educ91 educ94 educ87 i.emp86 i.emp88 i.emp91
i.emp89 educ89 i.emp92 i.emp94 educ88 educ86 i.emp90 i.mar93 i.mar90 i.mar94
i.mar91 i.mar92 i.mar84 i.mar87 i.mar85 i.mar89 i.mar86 i.mar83 i.mar88
birthdate , augment
             pared: regress pared i.race parpres i.enrol92 i.enrol94
i.enrol84 i.enrol89 i.enrol91 i.enrol90 i.enrol93 i.enrol83 i.enrol85
i.enrol87 i.enrol88 i.enrol86 i.emp83 i.emp85 i.emp84 educ83 educ90 educ92
i.emp93 i.emp87 educ84 educ93 educ85 educ91 educ94 educ87 i.emp86 i.emp88
i.emp91 i.emp89 educ89 i.emp92 i.emp94 educ88 educ86 i.emp90 i.mar93 i.mar90
i.mar94 i.mar91 i.mar92 i.mar84 i.mar87 i.mar85 i.mar89 i.mar86 i.mar83
i.mar88 birthdate
           parpres: regress parpres i.race pared i.enrol92 i.enrol94
i.enrol84 i.enrol89 i.enrol91 i.enrol90 i.enrol93 i.enrol83 i.enrol85
i.enrol87 i.enrol88 i.enrol86 i.emp83 i.emp85 i.emp84 educ83 educ90 educ92
i.emp93 i.emp87 educ84 educ93 educ85 educ91 educ94 educ87 i.emp86 i.emp88
i.emp91 i.emp89 educ89 i.emp92 i.emp94 educ88 educ86 i.emp90 i.mar93 i.mar90
i.mar94 i.mar91 i.mar92 i.mar84 i.mar87 i.mar85 i.mar89 i.mar86 i.mar83
i.mar88 birthdate
           enrol92: logit enrol92 i.race pared parpres i.enrol94 i.enrol84
i.enrol89 i.enrol91 i.enrol90 i.enrol93 i.enrol83 i.enrol85 i.enrol87
i.enrol88 i.enrol86 i.emp83 i.emp85 i.emp84 educ83 educ90 educ92 i.emp93
i.emp87 educ84 educ93 educ85 educ91 educ94 educ87 i.emp86 i.emp88 i.emp91
i.emp89 educ89 i.emp92 i.emp94 educ88 educ86 i.emp90 i.mar93 i.mar90 i.mar94
i.mar91 i.mar92 i.mar84 i.mar87 i.mar85 i.mar89 i.mar86 i.mar83 i.mar88
birthdate , augment

[output omitted]

Performing chained iterations ...

Multivariate imputation                     Imputations =        5
Chained equations                                 added =        5
Imputed: m=1 through m=5                        updated =        0

Initialization: monotone                     Iterations =       50
                                                burn-in =       10

             mar83: logistic regression
             mar84: logistic regression
             mar85: logistic regression
             mar86: logistic regression
             mar87: logistic regression
             mar88: logistic regression
             mar89: logistic regression
             mar90: logistic regression
             mar91: logistic regression
             mar92: logistic regression
             mar93: logistic regression
             mar94: logistic regression
             emp83: logistic regression
             emp84: logistic regression
             emp85: logistic regression
             emp86: logistic regression
```

```
              emp87: logistic regression
              emp88: logistic regression
              emp89: logistic regression
              emp90: logistic regression
              emp91: logistic regression
              emp92: logistic regression
              emp93: logistic regression
              emp94: logistic regression
            enrol83: augmented logistic regression
            enrol84: augmented logistic regression
            enrol85: logistic regression
            enrol86: logistic regression
            enrol87: logistic regression
            enrol88: logistic regression
            enrol89: logistic regression
            enrol90: logistic regression
            enrol91: logistic regression
            enrol92: logistic regression
            enrol93: logistic regression
            enrol94: logistic regression
               race: multinomial logistic regression
             educ83: linear regression
             educ84: linear regression
             educ85: linear regression
             educ86: linear regression
             educ87: linear regression
             educ88: linear regression
             educ89: linear regression
             educ90: linear regression
             educ91: linear regression
             educ92: linear regression
             educ93: linear regression
             educ94: linear regression
            parpres: linear regression
              pared: linear regression

------------------------------------------------------------------
                   |               Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
             mar83 |       5004         1077      1077 |      6081
             mar84 |       5041         1040      1040 |      6081
             mar85 |       5032         1049      1049 |      6081
             mar86 |       5016         1065      1065 |      6081
             mar87 |       5037         1044      1044 |      6081
             mar88 |       5003         1078      1078 |      6081
             mar89 |       5023         1058      1058 |      6081
             mar90 |       5124          957       957 |      6081
             mar91 |       5091          990       990 |      6081
             mar92 |       5088          993       993 |      6081
             mar93 |       5131          950       950 |      6081
             mar94 |       5105          976       976 |      6081
             emp83 |       5508          573       573 |      6081
             emp84 |       5471          610       610 |      6081
             emp85 |       5487          594       594 |      6081
             emp86 |       5416          665       665 |      6081
             emp87 |       5443          638       638 |      6081
             emp88 |       5413          668       668 |      6081
             emp89 |       5410          671       671 |      6081
             emp90 |       5390          691       691 |      6081
             emp91 |       5412          669       669 |      6081
             emp92 |       5409          672       672 |      6081
             emp93 |       5446          635       635 |      6081
             emp94 |       5406          675       675 |      6081
           enrol83 |       5823          258       258 |      6081
           enrol84 |       5843          238       238 |      6081
           enrol85 |       5819          262       262 |      6081
           enrol86 |       5808          273       273 |      6081
           enrol87 |       5819          262       262 |      6081
           enrol88 |       5813          268       268 |      6081
           enrol89 |       5840          241       241 |      6081
           enrol90 |       5835          246       246 |      6081
           enrol91 |       5839          242       242 |      6081
           enrol92 |       5857          224       224 |      6081
           enrol93 |       5825          256       256 |      6081
           enrol94 |       5855          226       226 |      6081
              race |       6019           62        62 |      6081
            educ83 |       5466          615       615 |      6081
            educ84 |       5440          641       641 |      6081
            educ85 |       5434          647       647 |      6081
            educ86 |       5395          686       686 |      6081
            educ87 |       5419          662       662 |      6081
            educ88 |       5399          682       682 |      6081
            educ89 |       5410          671       671 |      6081
            educ90 |       5455          626       626 |      6081
            educ91 |       5423          658       658 |      6081
            educ92 |       5450          631       631 |      6081
            educ93 |       5435          646       646 |      6081
            educ94 |       5422          659       659 |      6081
           parpres |       5868          213       213 |      6081
             pared |       5905          176       176 |      6081
------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)

Warning: the sets of predictors of the imputation model vary across
         imputations or iterations
```

Note that if any of your variables that are used but not imputed (variables after =) should be sets of dummies, use i. before these variables' names in the command.

Some helpful options of mi impute include:

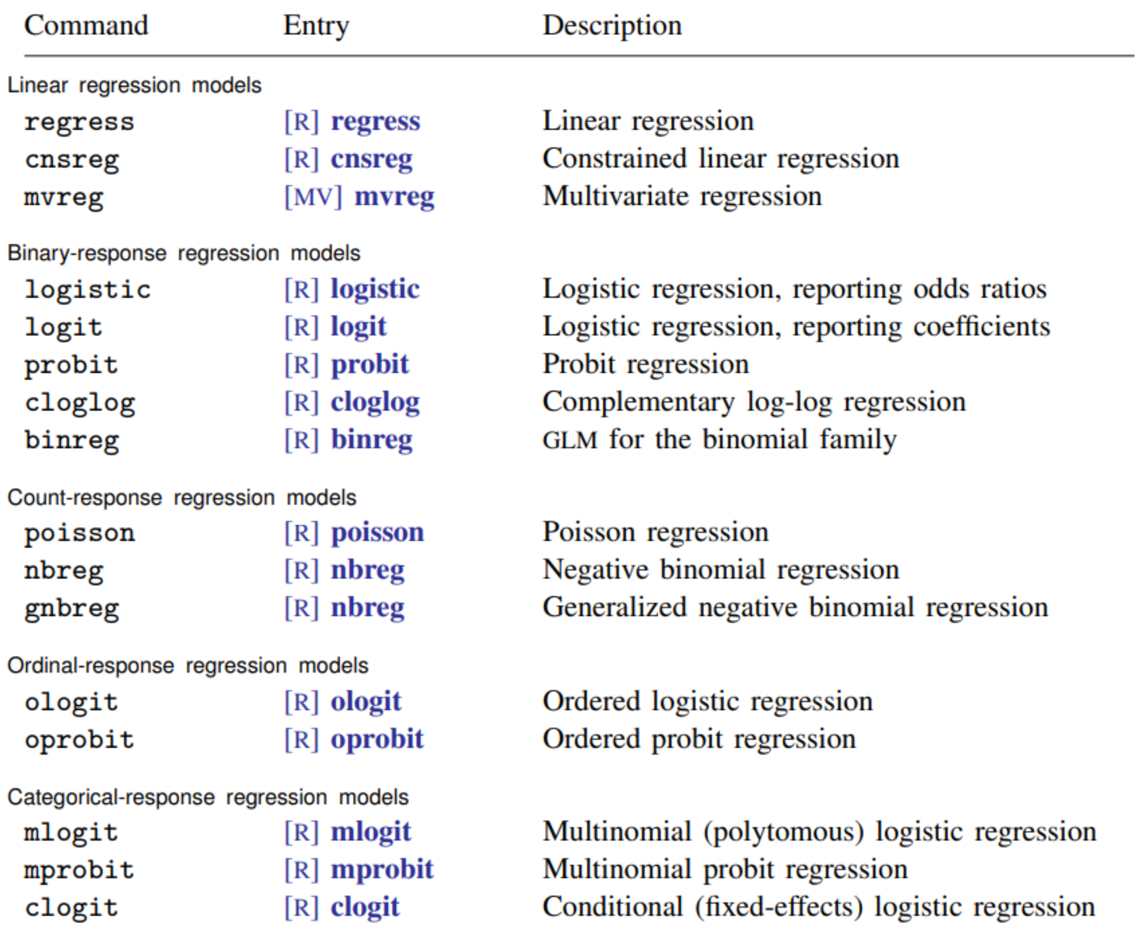
dryrun – same as for ice; specify equations but not run the actual imputation
rseed(#) – specify random-number seed in order to create imputations in a replicable fashion
by(varlist) – impute separately on each group formed by varlist
include – specify what to incude in the imputation equation
omit – to omit certain variables from some equations, e.g:

```
mi impute chained (logit) mar* emp* enrol* (mlogit, omit(birthdate)) race
(reg) educ* parpres pared = birthdate, add(5) augment
```

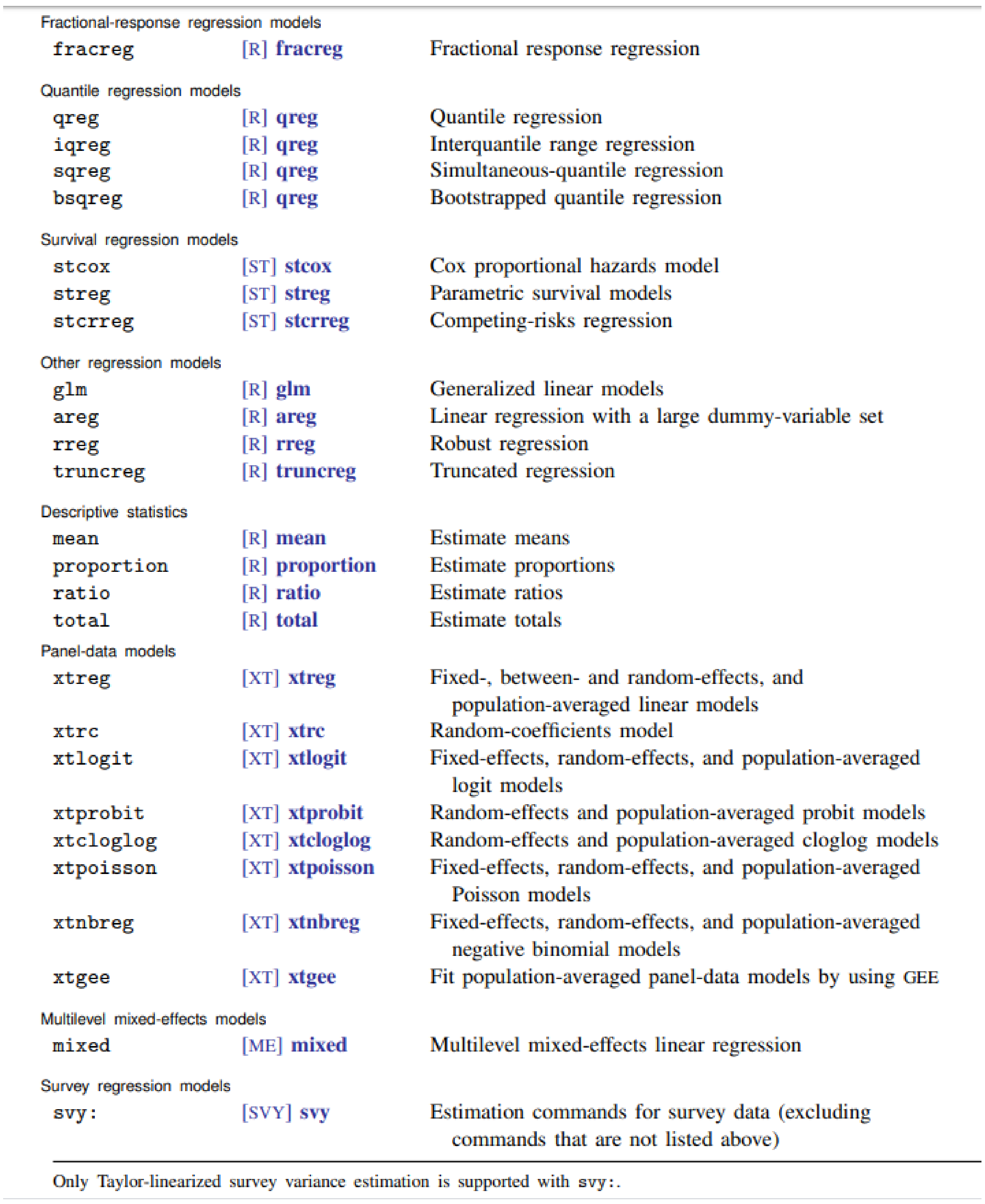
Estimation after multiple imputation

To obtain final estimates of the parameters of interest and their standard errors, one would fit a model in each imputation and carry out the appropriate post-MI averaging procedure on the results. In Stata, you can do that using mi estimate. It supports a wide range of commands:

The following estimation commands support the mi estimate prefix.



Command	Entry	Description
Linear regression models		
regress	[R] **regress**	Linear regression
cnsreg	[R] **cnsreg**	Constrained linear regression
mvreg	[MV] **mvreg**	Multivariate regression
Binary-response regression models		
logistic	[R] **logistic**	Logistic regression, reporting odds ratios
logit	[R] **logit**	Logistic regression, reporting coefficients
probit	[R] **probit**	Probit regression
cloglog	[R] **cloglog**	Complementary log-log regression
binreg	[R] **binreg**	GLM for the binomial family
Count-response regression models		
poisson	[R] **poisson**	Poisson regression
nbreg	[R] **nbreg**	Negative binomial regression
gnbreg	[R] **nbreg**	Generalized negative binomial regression
Ordinal-response regression models		
ologit	[R] **ologit**	Ordered logistic regression
oprobit	[R] **oprobit**	Ordered probit regression
Categorical-response regression models		
mlogit	[R] **mlogit**	Multinomial (polytomous) logistic regression
mprobit	[R] **mprobit**	Multinomial probit regression
clogit	[R] **clogit**	Conditional (fixed-effects) logistic regression



Fractional-response regression models		
`fracreg`	[R] **fracreg**	Fractional response regression
Quantile regression models		
`qreg`	[R] **qreg**	Quantile regression
`iqreg`	[R] **qreg**	Interquantile range regression
`sqreg`	[R] **qreg**	Simultaneous-quantile regression
`bsqreg`	[R] **qreg**	Bootstrapped quantile regression
Survival regression models		
`stcox`	[ST] **stcox**	Cox proportional hazards model
`streg`	[ST] **streg**	Parametric survival models
`stcrreg`	[ST] **stcrreg**	Competing-risks regression
Other regression models		
`glm`	[R] **glm**	Generalized linear models
`areg`	[R] **areg**	Linear regression with a large dummy-variable set
`rreg`	[R] **rreg**	Robust regression
`truncreg`	[R] **truncreg**	Truncated regression
Descriptive statistics		
`mean`	[R] **mean**	Estimate means
`proportion`	[R] **proportion**	Estimate proportions
`ratio`	[R] **ratio**	Estimate ratios
`total`	[R] **total**	Estimate totals
Panel-data models		
`xtreg`	[XT] **xtreg**	Fixed-, between- and random-effects, and population-averaged linear models
`xtrc`	[XT] **xtrc**	Random-coefficients model
`xtlogit`	[XT] **xtlogit**	Fixed-effects, random-effects, and population-averaged logit models
`xtprobit`	[XT] **xtprobit**	Random-effects and population-averaged probit models
`xtcloglog`	[XT] **xtcloglog**	Random-effects and population-averaged cloglog models
`xtpoisson`	[XT] **xtpoisson**	Fixed-effects, random-effects, and population-averaged Poisson models
`xtnbreg`	[XT] **xtnbreg**	Fixed-effects, random-effects, and population-averaged negative binomial models
`xtgee`	[XT] **xtgee**	Fit population-averaged panel-data models by using GEE
Multilevel mixed-effects models		
`mixed`	[ME] **mixed**	Multilevel mixed-effects linear regression
Survey regression models		
`svy:`	[SVY] **svy**	Estimation commands for survey data (excluding commands that are not listed above)

Only Taylor-linearized survey variance estimation is supported with `svy:`.

First, if we generated our imputed data using the separate ICE module, we have to do mi import to convert the data from MICE format into the format used by MI commands:

```
. use "L:\socy7706\marriage_imputed.dta", clear
. tab _mj

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

. mi import ice

. tab  _mi_m

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

. tab  _mi_miss

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

. sum  _mi_id

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

. mi reshape long interv mar educ emp enrol, i(id) j(year)

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

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                     6081   ->   72972
Number of variables                  67   ->      13
j variable (12 values)                    ->   year
xij variables:
         interv83 interv84 ... interv94   ->   interv
                  mar83 mar84 ... mar94   ->   mar
               educ83 educ84 ... educ94   ->   educ
                  emp83 emp84 ... emp94   ->   emp
            enrol83 enrol84 ... enrol94   ->   enrol
-----------------------------------------------------------------------------

reshaping m=1 data ...
reshaping m=2 data ...

reshaping m=3 data ...

reshaping m=4 data ...

reshaping m=5 data ...

assembling results ...

. tab _mi_m

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

We will use educ as our dependent variable; therefore, we have to delete its imputed values (using MID strategy). For that, we can use the variables that genmiss generated – specifically, mv_educ.

```
. clonevar educ_dv=educ
(7824 missing values generated)

. replace educ_dv=. If mv_educ==1
(39120 real changes made, 39120 to missing)

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

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

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

                                                Average RVI       =     0.0528
DF adjustment:   Large sample                   DF:     min       =     206.54
                                                        avg       =   28808.67
                                                        max       = 186339.77
Model F test:       Equal FMI                   F(   8, 8660.2)   =     952.02
Within VCE type: Conventional                   Prob > F          =     0.0000

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

<u>Useful options for mi estimate</u>

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

- nimputations(#) -- specify number of imputations to use; default is to use all in the file
  imputations(numlist) -- specify which imputations to use
- esampvaryok -- allow estimation when estimation sample varies across imputations
- cmdok -- allow estimation when estimation command is not supported by mi estimate
- post -- post estimated coefficients and VCE to e(b) and e(V)
- eform_option -- display coefficients table in exponentiated form; specific options depend on the command used:

eform	exponentiated coefficient, string is exp(b)
hr	hazard ratio, string is Haz. Ratio
shr	subhazard ratio, string is SHR
irr	incidence-rate ratio, string is IRR
or	odds ratio, string is Odds Ratio
rrr	relative-risk ratio, string is RRR

Syntax example:

```
. mi estimate, esampvaryok post ni(3): xtreg educ_dv mar  _Irace_2 _Irace_3 emp enrol
birthdate parpres pared, i(id)
```

If you have to use some command not supported by mi estimate, you can either try running it with cmdok, or if that does not work for some reason, you can estimate the model separately for each dataset, e.g.:

```
. for num 1/5: xtreg  educ_dv emp enrol mar birthdate parpres pared _Irace_2 _Irace_3
if  _mi_m==X, i(id)
```

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

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

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

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

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

Methods for nonignorable missing data

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

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

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

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

In ordinary multiple imputation, you assume that those people who report their weights are similar to those who don't. In a pattern-mixture model, you may assume that people who don't report their weights are an average of 20 pounds heavier. This is of course an arbitrary assumption; the idea of pattern mixture is to try out a variety of plausible assumptions and see how much they affect your results.

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

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

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

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

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

Carpenter JR1, Kenward MG, White IR. 2007. "Sensitivity analysis after multiple imputation under missing at random: a weighting approach." Statistical Methods in Medical Research, 16: 259-275.

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

Dealing with Attrition

Since attrition also results in missing data, it also can be either MCAR, MAR, or NMAR; and similarly to regular missing data, it is easier to deal with MCAR or MAR types of attrition. Researchers often do sensitivity analyses by imputing missing data for cases lost to follow-up and compare the results; they case also use weighing techniques where cases that are in the dataset are used to represent cases lost to follow-up using weights. As simulations show, if data are MCAR or MAR, the results without imputation or weights are pretty much as good as those with imputation and weights; they are also good if attrition is related to baseline values of outcome variable but not the follow-up value in addition to that. If it's NMAR with regard to both baseline and follow-up, there is bias with all approaches, especially if attrition rates are higher (above 25%).

See:

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

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

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

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

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

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

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