SOCY7706: Longitudinal Data Analysis
Instructor: Natasha Sarkisian
Introduction to Managing Longitudinal Data

Many longitudinal datasets are quite complex and require substantial data management efforts prior to use. Datasets can also vary considerably in terms of the ways that data are organized. We will look at Health and Retirement Study data in order to learn the basics of longitudinal data management. If interested, you can register to access the full data and get documentation at: http://hrsonline.isr.umich.edu/

Combining datasets

There are two commands in Stata for combining files: append and merge.

Appending datasets

Append works for datasets that both have the same set of variables but different observations – for example, when two waves of data are stored in separate files and variables have exactly the same variable names. Then we can open one dataset and then type:

```
. use dataset1.dta, clear
. append using dataset2.dta
```

Or we could just type:

```
.  append using dataset1 dataset2
```

In longitudinal context, we should make sure to create a time indicator to distinguish waves before appending. We will do that with two waves of HRS – 2006 and 2008.

```
. cd "L:\socy7706\"
. use H08A_R.dta
. keep  HHID PN LSUBHH LPN_SP LCSR LFAMR LFINR  LA099 LA100 LA019
. gen wave=2008
. rename (L*) (*)
. des

Contains data from L:\socy7706\H08A_R.dta
  obs:        17,217
 vars:            11                          8 Feb 2011 09:43
 size:       602,595 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
HHID            str6   %9s                    HOUSEHOLD IDENTIFICATION NUMBER
PN              str3   %9s                    RESPONDENT PERSON IDENTIFICATION NUMBER
SUBHH           str1   %9s                    2008 SUB HOUSEHOLD IDENTIFICATION NUMBER
PN_SP           str3   %9s                    2008 SPOUSE/PARTNER PERSON NUMBER
CSR             byte   %8.0g                  2008 WHETHER COVERSHEET RESPONDENT
FAMR            byte   %8.0g                  2008 WHETHER FAMILY RESPONDENT
FINR            byte   %8.0g                  2008 WHETHER FINANCIAL RESPONDENT
A019            int    %8.0g                  R CURRENT AGE CALCULATION
A099            byte   %8.0g                  NUMBER OF RESIDENT CHILDREN
```

```
A100            double %10.0g                 COUNT OF NONRESIDENT KIDS
wave            float  %9.0g
-------------------------------------------------------------------------------

. save short_2008.dta

. use H06A_R.dta, clear

. keep  HHID PN KSUBHH KPN_SP KCSR KFAMR KFINR  KA099 KA100 KA019

. gen wave=2006

. rename (K*) (*)

. des

Contains data from L:\socy7706\H06A_R.dta
  obs:        18,469
 vars:            11                          8 Feb 2011 11:51
 size:       775,698 (99.8% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
HHID            str6   %9s                    HOUSEHOLD IDENTIFICATION NUMBER
PN              str3   %9s                    RESPONDENT PERSON IDENTIFICATION NUMBER
SUBHH           str1   %9s                    2006 SUB HOUSEHOLD IDENTIFICATION NUMBER
PN_SP           str3   %9s                    2006 SPOUSE/PARTNER PERSON NUMBER
CSR             byte   %8.0g                  2006 WHETHER COVERSHEET RESPONDENT
FAMR            byte   %8.0g                  2006 WHETHER FAMILY RESPONDENT
FINR            byte   %8.0g                  2006 WHETHER FINANCIAL RESPONDENT
A019            int    %8.0g                  R CURRENT AGE CALCULATION
A099            double %10.0g                 NUMBER OF RESIDENT CHILDREN
A100            double %10.0g                 COUNT OF NONRESIDENT KIDS
wave            float  %9.0g
-------------------------------------------------------------------------------
. save short_2006.dta

. append using short_2008.dta

. tab wave

       wave |      Freq.     Percent        Cum.
------------+-----------------------------------
       2006 |     18,469       51.75       51.75
       2008 |     17,217       48.25      100.00
------------+-----------------------------------
      Total |     35,686      100.00
```

Then we can save the resulting merged file.

Or we could create a wave indicator on the go by specifying:

```
. append using short_2008.dta, gen(indicator)
. tab indicator
```

We can keep adding more waves to this dataset – the result will be adding more observations. We term that type of data setup as data being in the long format because different time points are represented by different observations, i.e., additional lines in the data.

When using append, beware of different variable names across waves! If two variables have different names, they will not be matched and appear as separate variables, with the corresponding observations from the other wave missing.

Even if variables are named the same, make sure there are no differences in the way things are coded across waves (e.g., 0 & 1 in one but 1 & 2 in another). If variable or value labels for some variables differ across appended dataset, the ones from the first dataset will be used.

Also note that both files we took for this exercise are respondent-level files –you cannot match files from different levels to each other using append. You can only stack "respondent" files with "respondent" files and "household" files with "household" files – combining them would require a different procedure based on merge command, which we will discuss next. It also would not make sense to use append to match the files from different modules (e.g., A_R and B_R) from the same wave because they contain different variables from the same people, and append does not match people.

Merging datasets

Another way to combine datasets is to create a wide format file. That is done using merge. There are four types of merges we could do: 1:1, 1:m, m:1, and m:m.

Merging 1:1

We will start with the simplest case, 1:1, and merge two respondent-level files. For that, however, we need to understand that there are different types of IDs in HRS:

```
. use H08A_R.dta, clear

. des

Contains data from L:\socy7706\H08A_R.dta
  obs:        17,217
 vars:            35
 size:     1,842,219 (99.6% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
HHID            str6   %9s                    HOUSEHOLD IDENTIFICATION NUMBER
PN              str3   %9s                    RESPONDENT PERSON IDENTIFICATION NUMBER
LSUBHH          str1   %9s                    2008 SUB HOUSEHOLD IDENTIFICATION NUMBER
KSUBHH          str1   %9s                    2006 SUB HOUSEHOLD IDENTIFICATION NUMBER
LPN_SP          str3   %9s                    2008 SPOUSE/PARTNER PERSON NUMBER
LCSR            byte   %8.0g                  2008 WHETHER COVERSHEET RESPONDENT
LFAMR           byte   %8.0g                  2008 WHETHER FAMILY RESPONDENT
LFINR           byte   %8.0g                  2008 WHETHER FINANCIAL RESPONDENT
…
```

In this file, every person who responded is uniquely identified with HHID and PN. If we need to merge such a file with another respondent's file, we would match them on these two variables:

```
. merge 1:1 HHID PN using H06A_R.dta

    Result                           # of obs.
    -----------------------------------------
    not matched                         2,700
        from master                       724  (_merge==1)
        from using                      1,976  (_merge==2)

    matched                            16,493  (_merge==3)
    -----------------------------------------
```

We need to carefully assess the results of merge and make sure these numbers make sense. Note the terminology – from master means from the file that was open when the merge was initiated; from using means from the file that was specified after "using" in the merge command.

Here are all the possibilities for codes in this table:

```
           numeric    equivalent
            code      word (results)     description
           ------------------------------------------------------------------------
              1       master             observation appeared in master only
              2       using              observation appeared in using only
              3       match              observation appeared in both
              4       match_update       observation appeared in both,
                                           missing values updated
              5       match_conflict     observation appeared in both,
                                           conflicting nonmissing values
           ------------------------------------------------------------------------
    Note: If codes of both 4 and 5 could pertain to an observation, then 5 is used.
```

Most cases give us a perfect merge – but there are some cases that are in 2006 dataset but not in 2008 (from using) and those that are in 2008 but not in 2006 (from master). Here, both types of situations are possible, and it makes sense that there are more cases that drop out from 2006 to 2008 than those that appear in 2008 but not in 2006. It would be easier to investigate these patterns if we started merging at wave 1 of the data.

Codes 4 and 5 can arise only if the update option is specified. Update option (as well as replace option) performs an update merge rather than a standard merge. In a standard merge, the data in the master always have priority and do not get changed. If both the master and using datasets contain the same variable but with different values, then matched observations will contain values from the master dataset, even if these values are missing in the master dataset, and unmatched observations will contain values from either master or using, depending on where these observations are from.

If the update option is specified, then matched observations will update missing values from the master dataset with values from the "using" dataset. Nonmissing values in the master dataset will be unchanged.

If replace option is specified, then matched observations will contain values from the "using" dataset, unless these values are missing, in which case the values from the master dataset are retained.

Merging 1:m and m:1

In situations when the data have some kind of nested structure (either because of the longitudinal component or because of another type of multilevel design such as individuals nested within households), we will often need to do merges where one case in file 1 will be matched to multiple ones in file 2, or vice versa. For instance, if one file has those characteristics of individuals that do not change over time (birth year, race/ethnicity, gender, etc.) and the other has time-varying data with multiple observations per person, then one unit of file 1 is person, and each person might be matched to multiple time-points in file 2. Or a single household may be matched to multiple individuals within household if multiple persons were interviewed in all or some households.

For our example, if we would want to merge information from household file to the individual file that we just created, we would want to match them on HHID and the SUBHH of the corresponding wave. SUBHH is used because households change across waves as individuals divorce or remarry.

```
. use H08A_H.dta, clear

. des
Contains data from L:\socy7706\H08A_H.dta
  obs:        11,897
 vars:            43
 size:     1,046,936 (99.8% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
HHID            str6   %9s                    HOUSEHOLD IDENTIFICATION NUMBER
LSUBHH          str1   %9s                    2008 SUB HOUSEHOLD IDENTIFICATION NUMBER
KSUBHH          str1   %9s                    2006 SUB HOUSEHOLD IDENTIFICATION NUMBER
LPN_CS          str3   %9s                    2008 COVERSCREEN RESP PERSON NUMBER
LPN_FAM         str3   %9s                    2008 FAMILY RESP PERSON NUMBER
LPN_FIN         str3   %9s                    2008 FINANCIAL RESP PERSON NUMBER
LPN_NCS         str3   %9s                    2008 NON-COVERSCREEN RESP PERSON NUMBER
LPN_NFAM        str3   %9s                    2008 NON-FAMILY RESP PERSON NUMBER
LPN_NFIN        str3   %9s                    2008 NON-FINANCIAL RESP PERSON NUMBER

. merge 1:m HHID  LSUBHH using H08A_R.dta
    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                            17,217  (_merge==3)
    -----------------------------------------
```

We can then add more datasets:

```
. merge 1:1 HHID PN using H06A_R.dta
_merge already defined
r(110);

. rename _merge merge_08_AH_AR
```

To avoid the need to rename _merge, we can give it a name right away using gen option, so we will do that for the next merge – here, we are merging individual data from 2008 with the addition of household information to individuals in 2006, so it's a 1:1 merge again.

```
. merge 1:1 HHID PN using H06A_R.dta, gen(merge_06_AR)

    Result                           # of obs.
    -----------------------------------------
    not matched                         2,700
        from master                       724  (_merge==1)
        from using                      1,976  (_merge==2)

    matched                            16,493  (_merge==3)
    -----------------------------------------
```

And now merging in the household information from 2006:

```
. merge m:1 HHID KSUBHH using H06A_H.dta, gen(merge_06_AH)

    Result                           # of obs.
    -----------------------------------------
    not matched                           565
        from master                       559  (_merge==1)
        from using                          6  (_merge==2)

    matched                            18,634  (_merge==3)
    -----------------------------------------
```

The important aspect of the merge process is to make sure that merging frequencies correspond to what you know about the data. For instance, if the data are longitudinal and no new cases are added after the first wave, then, if you start merging with wave 1, you can have observations that are in master but not using, but you cannot have observations that are in using but not in master.

Merging m:m

Such merges are pretty much not used. There are also examples of other very rare merges, using joinby and cross commands, that are used for very rare cases of combining datasets.

Some useful options of merge (see help merge for more):

keepusing(varlist) specifies the variables from the using dataset that are kept in the merged dataset. By default, all variables are kept.

force allows string/numeric variable type mismatches, resulting in missing values from the using dataset. If omitted, merge issues an error; if specified, merge issues a warning.

keep(results) specifies which observations are to be kept from the merged dataset. Using keep(match master), for example, specifies keeping only matched observations and unmatched master observations after merging.

Note that the biggest problems with mering stem from problems with the key variable or variables that are used for the merge. If one of your datasets contains duplicate cases, with the

same ID, your merge will fail and you need to deal with duplicates first. If you have multiple observations per person in your dataset and you are trying to merge only on ID, that will fail – a merge should be done on both ID and time variable in such cases to avoid problems.

Reshaping datasets

Once we merged datasets from different waves, we end up with a wide format dataset.
Wide format and long format each have their own advantages for both data management and analysis. For instance, for a lot of data management, we would typically want to change into long format, so it's only one variable per measure, rather than separate variables for each time point. But imputation is usually done in the wide format. So in most cases, you need to shift back and forth.

Reshaping wide to long

We can change the format it using reshape command; we will first get rid of variables from those waves we do not use, however (otherwise, reshape will assume we have three waves of data for everything and create a lot of blank rows).

```
. drop JSUBHH
```

Next, we need to list stems for all time-varying variables.

Time-varying vs time-invariant is an important distinction. Our dependent variable in longitudinal analysis should always be time-varying, while independent ones could be either, but some techniques restrict it further. In a wide format, we do not have a separate variable for time, but we will create it in the long format.

```
. reshape long  @SUBHH @PN_CS  @PN_FAM @PN_FIN @PN_NCS @PN_NFAM @PN_NFIN @A020 @A022
@A023 @A024 @A025 @A026 @A027 @A030 , j(wave) string i(HHID PN)
(note: j = K L)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                    19199   ->   38398
Number of variables                 142   ->     128
j variable (2 values)                     ->   wave
xij variables:
                          KSUBHH LSUBHH   ->   SUBHH
                          KPN_CS LPN_CS   ->   PN_CS
                        KPN_FAM LPN_FAM   ->   PN_FAM
                        KPN_FIN LPN_FIN   ->   PN_FIN
                        KPN_NCS LPN_NCS   ->   PN_NCS
                      KPN_NFAM LPN_NFAM   ->   PN_NFAM
                      KPN_NFIN LPN_NFIN   ->   PN_NFIN
                            KA020 LA020   ->   A020
                            KA022 LA022   ->   A022
                            KA023 LA023   ->   A023
                            KA024 LA024   ->   A024
                            KA025 LA025   ->   A025
                            KA026 LA026   ->   A026
                            KA027 LA027   ->   A027
                            KA030 LA030   ->   A030
-----------------------------------------------------------------------------
```

To bring it back into wide, we could just type:

```
. reshape wide
```

And back to long:

```
. reshape long
```

In long, we probably would want to make some things more clear:

```
. replace wave="2006" if wave=="K"
wave was str1 now str4
(19199 real changes made)

. replace wave="2008" if wave=="L"
(19199 real changes made)

. destring wave, replace
wave has all characters numeric; replaced as int

. tab wave

       wave |      Freq.     Percent        Cum.
------------+-----------------------------------
       2006 |     19,199       50.00       50.00
       2008 |     19,199       50.00      100.00
------------+-----------------------------------
      Total |     38,398      100.00
```

Now if we would want to return to wide format, we would need to specify the model again because we changed wave.

<u>Reshaping long to wide</u>

```
. reshape wide  SUBHH PN_CS  PN_FAM PN_FIN PN_NCS PN_NFAM PN_NFIN A020 A022 A023 A024
A025 A026 A027 A030 , j(wave) i(HHID PN)
(note: j = 2006 2008)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                    38398   ->   19199
Number of variables                 128   ->     142
j variable (2 values)              wave   ->   (dropped)
xij variables:
                                  SUBHH   ->   SUBHH2006 SUBHH2008
                                  PN_CS   ->   PN_CS2006 PN_CS2008
                                 PN_FAM   ->   PN_FAM2006 PN_FAM2008
                                 PN_FIN   ->   PN_FIN2006 PN_FIN2008
                                 PN_NCS   ->   PN_NCS2006 PN_NCS2008
                                PN_NFAM   ->   PN_NFAM2006 PN_NFAM2008
                                PN_NFIN   ->   PN_NFIN2006 PN_NFIN2008
                                   A020   ->   A0202006 A0202008
                                   A022   ->   A0222006 A0222008
                                   A023   ->   A0232006 A0232008
                                   A024   ->   A0242006 A0242008
                                   A025   ->   A0252006 A0252008
                                   A026   ->   A0262006 A0262008
                                   A027   ->   A0272006 A0272008
                                   A030   ->   A0302006 A0302008
-----------------------------------------------------------------------------
```

And now we can easily go back and force again.

```
. reshape long
(note: j = 2006 2008)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                    19199   ->   38398
Number of variables                 142   ->     128
j variable (2 values)                     ->   wave
xij variables:
                     SUBHH2006 SUBHH2008   ->   SUBHH
                     PN_CS2006 PN_CS2008   ->   PN_CS
                   PN_FAM2006 PN_FAM2008   ->   PN_FAM
                   PN_FIN2006 PN_FIN2008   ->   PN_FIN
                   PN_NCS2006 PN_NCS2008   ->   PN_NCS
                 PN_NFAM2006 PN_NFAM2008   ->   PN_NFAM
                 PN_NFIN2006 PN_NFIN2008   ->   PN_NFIN
                       A0202006 A0202008   ->   A020
                       A0222006 A0222008   ->   A022
                       A0232006 A0232008   ->   A023
                       A0242006 A0242008   ->   A024
                       A0252006 A0252008   ->   A025
                       A0262006 A0262008   ->   A026
                       A0272006 A0272008   ->   A027
                       A0302006 A0302008   ->   A030
-----------------------------------------------------------------------------
```

As was the case with merge, if id variables do not uniquely identify observations, you will get an error. Another reason for an error would be if a variable for which you do not specify a stem because it is supposed to be time invariant does in fact have different values for different observations. If you get this error, you can then use "reshape error" command to pinpoint where your time-invariant variables actually do vary even though they should not – it will list problem observations when reshape fails.

Reshaping into long will generate rows that are entirely empty for those people who were missing data on all variables for a specific year because they did not participate (e.g., attrition). It makes sense to drop those:

```
. egen all=rowmiss( A020- A030)

. tab all
        all |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |         59        0.15        0.15
          2 |      1,351        3.52        3.67
          3 |          2        0.01        3.68
          4 |          6        0.02        3.69
          5 |     22,848       59.50       63.20
          6 |     11,589       30.18       93.38
          8 |      2,543        6.62      100.00
------------+-----------------------------------
      Total |     38,398      100.00

. keep if all<8
```

Why do we care to get rid of empty rows? It obscures how much data we actually have and makes us believe we have balanced data.

Balanced vs unbalanced panel data:
Balanced = each unit is observed the same number of times (T)
Unbalanced = some units have fewer time points than others

Reasons for being unbalanced:

1. Temporary unit non-response xxx.x.xx
2. Panel attrition xxxx….
3. Late entry …xxxxx