Stata Introductory Series

Stata 202 – Data Manipulation

Our introductory Stata session #100 laid out the fundamentals of interacting with Stata; setting up DO and LOG files to easily save, replicate, modify, and share your analysis and its results; and how to detect and deal with missing and special values as you check your work.

Stata 202 will build off of last week's training on Data Cleaning (Stata 201), but do not worry if you did not attend last week or things are slightly confusing or a bit more difficult to follow along on. This week's session does not require attending prior trainings, and we will be around to answer questions and debug during and after the session. The goal of this session is to continue learning how to import, clean, and manipulate messy data that you may encounter in the real world from administrative, survey, or your own data collection.

What you will get out of this session:

- Learn how to utilize messy, real-world data
- Reshape, merge, and append your data sets
- Review how to find and deal with duplicate records
- Review how to clean and convert string to numeric and date data or vice versa
- Review how to convert information stored in text strings into indicators

Basic command structure

```
command   objects    conditions          , options
use       file.dta                       , clear
generate  age = 15   if AGE2 == 15
tabulate  state      if country == "US"  , missing
```

Helpful resources

- Stata manual: access by typing "help command" in the stata console
- Statalist: https://www.statalist.org/forums/forum/general-stata-discussion/general
  - Often will come up if you google a question that isn't covered by the documentation
- UCLA IDRE: https://stats.idre.ucla.edu/stata/
  - Provides helpful tips on how to use Stata as well as the statistics behind the programming
- UNC CPC: http://www.cpc.unc.edu/research/tools/data_analysis/statatutorial
  - Guide to working with and analyzing data in Stata

Remember: Getting errors is a normal part of programming! The best way to debug is to read through every line carefully.

Next class:

- Stata 301 – **Automating Tasks and Exporting Output**

```
clear
capture log close
capture ssc install savesome                /*Some commands have to be installed, but
                                              only once, capture prevents the error from
                                              popping up*/

set more off, permanently

*Set a working directory where all your project files will be located (replace my username
with your own):
*cd "D:\Users\CLou\Desktop\"
cd "C:\Users\urbanmeet\Desktop\"


log using "StataClass_$S_DATE.log", replace         /*$S_DATE saves the date in
                                                      the name of your log file, so
                                                      you save a log daily. This is
                                                      helpful if you're changing
                                                      things and need to go back*/


**This training builds off of the prior "Data Cleaning" session, though attending that
training first is not required or necessary to participate in this session.
    *All materials and files necessary from the prior Data Cleaning training session are
provided, and some concepts are reviewed.

*Open up the "long.dta" Stata data set created at the end of the "Data cleaning" training
session:
use "long.dta", clear
/*Reshaping the data set*/
sort id arr_d
list in 1/10                    /*Data is currently in long format*/

*Right now, the data is setup in LONG format, meaning there are repeated
observations/rows/records
    *of the same entity/unit of analysis, usually across time. Sometimes it is easier
    *or better to work with data in WIDE format, where there is one observation/row/record
    *for each entity or unit of analysis, as you do not have to worry about double counting.
    *The code below will help with converting from LONG to WIDE format; the code is similar
for WIDE to LONG.

/*Need to generate a number that indicates how to order the wide format data*/
by id (arr_d), sort: egen arrest_n=seq() /*Creates a variable of the number of the
                                          arrests by person id, the variable not in
                                          parenthesis is the variable that it's created
                                          by, the variable in parenthessis is sorted
                                          within that*/
by id (arr_d), sort: gen arrest_n2=_n  /* you can also use Stata's "_n" notation with just
regular generate.
                                          The "_n" is essentially the index or (usually)
observation number in Stata and quite powerful.
                                          When specifying groups with "by :", the "_n" index
actually resets for each group, which can be
                                          use to your advantage if you want to know how many
observations you have in a group or to mark the
                                          first or last observation within your group. "_N"
is the total # of observations overall
                                          or within your group when combined with "by :".
Also, using [_n-1] and [_n+1] or +2,-2, etc.
                                          indices directly or appending after a variable name
allows you to reference or use the values
                                          or prior or subsequent observations in sorted data,
which is also quite useful.*/
list id arr_d arrest_n arrest_n2
assert arrest_n == arrest_n2
drop arrest_n2

sort id arr_d
list id arr_d arrest_n in 1/10

```

```
*rename * *_  /*If I ran this line the variables generated in the reshape would be arr_d_1 */
                /*arr_d_2 instead of arr_d1 arr_d2*/

reshape wide arr_*, i(id) j(arrest_n)   /*Converts to wide format, i is the id, j
                                          is the order of the dates, note arr_n
                                          becomes suffix*/
list id arr_d* in 1/2
duplicates report id

gen t_arr1= arr_d2-arr_d1               /*Calculate the time between arrests*/
sum t_arr1

save "wide.dta", replace

import excel "Stata Class File_main.xlsx",          ///
            sheet("Sheet1")                 ///
            firstrow                        ///
            case(lower)                     ///
            allstring                       ///
            clear

describe
list in 1/10

duplicates report id

/*Clean birthdate*/
list dob in 1/10
gen birth_d=date(dob, "MDY")        /*Need to input format of the date of birth,
                                    "MDY" or "DMY"*/
    label var birth_d "Birth Date"
    format birth_d %td
    assert birth_d!=. if dob!=""            /*if date is not in the right format,
                                              the new var will be missing when the
                                              old var has a value always want to
                                              check this*/
    list dob birth_d if birth_d==. & dob!=""
    replace dob="01/01/1985" if dob=="01/001/1985"
    replace birth_d=date(dob, "MDY")
    assert birth_d!=. if dob!=""
    codebook birth_d, d
    list birth_d dob if (birth_d<=td(01jan1915) | birth_d>td(01jan1996)) /*list new and
                                                                           old var that
                                                                           might not be
                                                                           real*/
    replace birth_d=. if birth_d<=td(01jan1915)  /*replace with missing*/
    replace birth_d=. if birth_d>=td(01jan1996) /*replace with missing*/
    drop dob


gen age=floor((td(01jan2014)-birth_d)/365)      /*Generate age, floor rounds down to
                                                  the nearest interger*/
    sum age, d
    /*Generate categorical age variable with labels*/
    assert age<=100 | age==.            /*Note missing is considered infinity be
                                          careful with open ended greater/less than*/
    recode age                          ///
        (18/25=1 "18-25 Years Old")     ///
        (26/30=2 "26-30 Years Old")     ///
        (30/100=3 "30+ Years Old"),     ///
    gen(age_cat)
    label var age_cat "Age Category"
    tab age_cat, m

/*Cleaning string variables, generating sex categorical variables*/
tab sex, m
    replace sex=lower(sex)              /*makes string lower case*/
    replace sex=trim(sex)               /*removes leading/trailing spaces*/
    tab sex
gen gender=(substr(sex,1,1)=="f")           /*If the first letter is f, syntax is
```

```
                                                substr(varname, position, number)*/
    replace gender=. if sex==""
    replace gender=. if sex=="u"
    label var gender "Gender"
    capture label define gender 0 "Male" 1 "Female" /*Capture allows the program to
                                                      continue even if there's an error*/
    label values gender gender
    tab sex gender, m
    drop sex

save "main.dta", replace
use "main.dta", clear

/*Merge Data Sets*/
/*Don't need to sort either data set prior to merging*/
merge 1:m id using "long.dta"   /*need to specify variable on which to merge and the
                    extent of duplicates in each file no duplicates is a 1, any duplicates
                    is an m, can do 1:1, 1:m or m:1 never merge m:m (it probably doesn't do
                    what you think-- see the HELP file), use joinby instead,
                    which creates all pairwise combinations of the data based on the linking
                    variables specified, which is what the m:m merge sounds like it should
do.*/
                    /*if same varnames will override variables in the using data set*/
                    /*generates variable called _merge, indicating how well each
                    observation merged*/
    keep if _merge==3 /*Keep the ones that matched*/
    drop _merge
    sort id
    list in 1/10
save "data_long.dta", replace

/*Or you can merge 1:1 using the wide data set*/
use "main.dta", clear
    merge 1:1 id using "wide.dta"
    keep if _merge==3
    drop _merge
    sort id
    list in 1/10

save "data_wide.dta", replace
keep id gender birth_d age arr_d1 arr_y1


savesome                                ///
    id gender birth_d age arr_d1 arr_y1 ///
    if arr_y1==2014                     ///
    using "arr_2014.dta", replace /*Saves a portion of the data set*/

    sum arr_y1
    drop if arr_y1==2014 /*drops that same portion*/
    sum arr_y1

*Append essentially stacks one data set on top of another
    *as opposed to merge which places them side-by-side.
append using "arr_2014.dta" /*adds that data set back in*/
    sum arr_y1
    clear



***EXERCISES****
*(It will be helpful to run the code above 1st,
* as the questions below use some of the data files that are created.)


/*Open the main.dta data set.  How many
duplicates are there on id?*/

*Answer:
use "main.dta", clear
```

```
196       duplicates report id
197
198   /*Merge in the data_long.dta (Hint: type "help merge" into the command line
199   to find out the syntax for this kind of _merge)*/
200
201       merge 1:m id using "data_long.dta"
202
203   /*Keep only those observations in both files, that matched.
204   Save this data set in the temp folder using the name main_long.dta*/
205
206       keep if _merge==3
207       save "main_long.dta", replace
208
209   /*Reshape you data so that there is only one record or observation per id. Save your new
      data file under the name "main_wide.dta":*/
210   drop _merge
211   sort id arr_d
212   by id: gen arrest_n3=_n
213   reshape wide arr_* age* birth* gender , i(id) j(arrest_n3)
214
215
216   /*import the csv file called main_names.csv  (Hint: you can either search help import
217   or use the interface to determine the code for importing a csv)*/
218
219   import delimited "main_names.csv", clear
220
221   /*How many true duplicates are there? How many duplicates on id? Drop any true duplicates*/
222
223   duplicates report
224   duplicates report id
225   duplicates drop
226
227   /*Remove any true duplicates in terms of name (accounting for different letter casing and
      any leading or training spaces*/
228
229   replace full_name=trim(full_name)
230   replace full_name=lower(full_name)
231   duplicates drop
232
233
234   /*Try to merge in the main_long.dta data set that you saved in the temp folder*/
235
236       merge 1:m id using "main_long.dta"
237
238   /*Did you get an error message?  How can you address the error?*/
239
240       describe id
241       tostring id, replace
242
243       merge 1:m id using "main_long.dta"
244
245   /*Did you get another error message?  How can you address the error?*/
246       merge 1:m id using "main_long.dta", gen(new_merge)
247
248
249   capture log close
250   exit
251
```