Stata Summer Series

Stata 201 – Data Cleaning

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 201 will step things up a bit, so do not worry if things are slightly confusing or a bit more difficult to follow along on. We will be around to answer questions and debug during and after the session. Up to now we have been using pristine, pre-cleaned data, the goal of this session is to lay out the basics of importing, cleaning, and manipulating 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
- » Find and deal with duplicate records
- » Convert string to numeric and date data or vice versa
- » 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 classes:

- » Stata 202 – **Data Manipulation** (Thursday, July 23, 2:00 pm-3:00 pm on Zoom: https://zoom.us/j/93212540371)
- » Stata 301 – **Automating Tasks and Exporting Output** (Wednesday, July 30, 2:00 pm-3:00 on Zoom: https://zoom.us/j/91742344531)

```
 1    clear
 2    capture log close
 3    capture ssc install savesome                  /*Some commands have to be installed, but
 4                                                   only once, capture prevents the error from
 5                                                   popping up*/
 6
 7    set more off, permanently
 8
 9    *Set a working directory where all your project files will be located (replace my username
      with your own):
10    cd "D:\Users\CLou\Desktop\"
11    *cd "C:\Users\urbanmeet\Desktop\"
12
13
14    log using "StataClass_$S_DATE.log", replace        /*$S_DATE saves the date in
15                                                       the name of your log file, so
16                                                       you save a log daily. This is
17                                                       helpful if you're changing
18                                                       things and need to go back*/
19
20
21    /*Easiest way to import a file is to use the pull down menu, it will give you the stata
22    code for the command, otherwise you can use StatTransfer*/
23
24    /*Importing an excel file*/
25    import excel "Stata Class File_long.xlsx",  /// Imports the excel file
26                sheet("Sheet1") /// Tells stata which sheet to import
27                firstrow        /// Tells stata first row is variable names
28                case(lower)     /// Makes variable names lowercase (Stata is case sensitive)
29                allstring       /// Forces all to be string, some numeric won't import well
30                clear
31
32    describe
33    list in 1/10                    /* Lists the first 10 observations*/
34
35    *DUPLICATES
36    /*Checking for duplicates, this should include one observation per person per arrest date*/
37    duplicates report id            /*more than one arrest for most of these individuals*/
38    duplicates report id arrdate    /*reports number of duplicates on id arrdate*/
39    duplicates tag id arrdate, gen(t_dup) /*generates a variable called t_dup to identify
40                                          duplicates*/
41    list if t_dup!=0
42    duplicates drop                 /*drops any true duplicates*/
43    duplicates report id arrdate    /*reports number of duplicates on id arrdate*/
44
45
46    /*Converting from string to numeric and vice versa*/
47    list id arrdate in 1/10               /*displays first 10 observations of id and arrest date*/
48    describe id arrdate                    /*describes the characteristics of id and arrestdate*/
49
50    destring id, gen(t_id)                /*Turns string into numeric, need all characters
51                                          to be numeric. Having temporary variables start
52                                          with t_ helps you keep track of which variables
53                                          to drop*/
54    destring arrdate, gen(t_arrdate)      /*This doesn't work because arrdate has "/"s
55                                          in it.*/
56    list id arrdate in 1/10               /*displays first 10 observations of id and arrest date*/
57    describe id arrdate                    /*describes the characteristics of id and arrestdate*/
58
59    tostring t_id, replace                /*Turns numeric back to string*/
60    drop t_id
61
62    /*Cleaning Dates*/
63    list arrdate in 1/10
64    gen arr_d=date(arrdate, "MDY")            /*Need to input format of the date of birth,
65                                              "MDY" (Month Day Year) or
66                                              "DMY" (Day Month Year)*/
67        label var arr_d "Arrest Date"        /*Adds a label to the variable*/
68        list arr_d arrdate in 1/10
69        format arr_d %td                     /*dates in stata are numeric values that are
```

```
70                                           the number of days (or other denomination)
71                                           since 1/1/1960, they need to be formatted*/
72      list arr_d arrdate in 1/10
73      assert arr_d!=. if arrdate!=""      /*Assert checks the statement, if it's not true
74                                          then it will stop your program. This is helpful
75                                          if you're running the same program on different
76                                          data it checks your assumptions*/
77      drop arrdate
78
79   /*Once it's in stata date format it's easy to transform*/
80      gen arr_m=month(arr_d)              /*generates month of arrest variable*/
81      gen arr_y=year(arr_d)              /*generates year of arrest variable*/
82      gen arr_my=mofd(arr_d)             /*generates variable equal to the month/year of
83                                          arrest, note numerically this is the
84                                          number of months since 1/1/1960, need to
85                                          format*/
86      list arr_m arr_y arr_my arr_d in 1/10
87      format arr_my %tm                  /*formats it as year month*/
88      list arr_m arr_y arr_my arr_d in 1/10
89      gen arr_jan25=(arr_d==td(25jan2014)) /*Shortcut for creating dummies is to put
90                                          the if statement after the equal sign in
91                                          parenthesis, downside is only for zero/one
92                                          variables, problem for missings*/
93                                          /*Note td() calls the label otherwise
94                                          you would have to calculate days from
95                                          1/1/1960*/
96      tab arr_d if arr_jan25==1           /*Check it is defined correctly*/
97      drop arr_jan25 arr_my arr_m
98   /*Clean arrest reason*/
99   tab violdesc, m
100     replace violdesc=lower(violdesc) /*makes all lower case, opposite is upper()*/
101     replace violdesc=strtrim(violdesc) /*drops excess spaces at the beginning and end of a
    string*/
102     tab violdesc, m
103
104  /*Say we wanted to create a variable for arrest reason*/
105  gen arrestreason=0
106     replace arrestreason=1 if strpos(violdesc, "theft")!=0 /*strpos tells you the
107                                                        position of the string
108                                                        in the variable, 0 means
109                                                        it's not in the variable*/
110     replace arrestreason=2 if (strpos(violdesc, "liquor")!=0 | strpos(violdesc, "alcohol")!=
    0)
111     tab violdesc arrestreason, m
112     label define arrestreason 0 "Other" 1 "Theft" 2 "Alcohol"
113     label values arrestreason arrestreason
114     label var arrestreason "Reason For Arrest"
115     tab violdesc arrestreason, m
116
117  /*Other useful string commands: */
118     tab violdesc
119     gen t_violdesc = subinstr(violdesc, "-", " ",.) /*pull out certain characters*/
120     tab t_violdesc, m
121     split violdesc, p("-") gen(t_)                   /*Divides the string into a bunch
122                                                      of strings based on where the parse
123                                                      is, useful for full names*/
124     help string functions                           // Stata HELP file listing various
    string functions...
125
126     list violdesc t_1 t_2 if t_2!="" in 1/100
127     drop t_*               /* a * after a variable name calls all variables that start
128                            with that name, use a t_ prefix  for all your temporary
129                            variable and then you can drop with a drop t_*
130                            when your done*/
131  /*renaming variables*/
132  rename arrestreason arr_r
133     describe
134  rename arr_*    arrest_*    /*You can use * to rename a bunch of variable */
135     describe
136  rename *_d  d_*              /*or to add/remove a prefix*/
```

```
137          describe
138     rename d_*  *_d
139     rename arrest_* arr_*
140          describe
141
142
143     /*Say we want to get rid of duplicates so that a person has their arrest reasons
144     listed horizontally*/
145     duplicates tag id arr_d, gen(t_dup) /*generates a variable called t_dup to identify
146                                      duplicates*/
147
148     list if t_dup!=0
149
150     *This code will create indicators for if each individual (ID) was ever arrested for each of
151     the 3 reasons we specified on each date:
151     gen t_arr=(arr_r==1)
152     list id arr_d arr_r t_arr in 1/10
153     by id arr_d, sort: egen arr_theft=max(t_arr)
154     list id arr_d arr_r t_arr arr_theft in 1/10
155     drop t_arr
156     gen t_arr=(arr_r==2)
157     by id arr_d, sort: egen arr_alcohol=max(t_arr)
158     drop t_arr
159     gen t_arr=(arr_r==0)
160     by id arr_d, sort: egen arr_other=max(t_arr)
161     drop t_arr
162     label define yesno 0 "No" 1 "Yes"
163     label values arr_theft arr_alcohol arr_other yesno
164
165     list if t_dup!=0
166     *Now we can drop the specific arrest descriptions as well as the categorical description
167         *since we have indicators for our categories of arrest for each date:
168     drop arr_r violdesc t_*
169     duplicates drop
170     duplicates report id arr_d
171
172
173     save "long.dta", replace /*Save the long form of this data set*/
174
175
176
177
178     ***EXERCISES****
179     *(It will be helpful to run the code above 1st,
180     * as the questions below use some of the data files that are created.)
181
182     /*Open the long.dta data set. How many duplicates are there on id?*/
183
184     *Answer:
185     use "long.dta", clear
186         duplicates report id
187
188
189     /*import the csv file called main_names.csv  (Hint: you can either search help import
190     or use the interface to determine the code for importing a csv)*/
191
192     import delimited "main_names.csv", clear
193
194     /*How many true duplicates are there? How many duplicates on id? Drop any true duplicates*/
195
196     duplicates report
197     duplicates report id
198     duplicates drop
199
200     /*Generate a variable which identifies the duplicates, sort by id and then
201     list the first few duplicates (hint: set more off)*/
202
203     duplicates tag id, gen(t_dup)
204     list if t_dup==1 in 1/100  /*Note that in 1/100 command just shows the first 100
205                              observations not the first 100 that meet the criteria*/
```

```stata
206    set more on
207    list if t_dup==1              /*Set more on then click the red (x) at the top to break */
208    set more off
209    /*The duplicates show different versions of the same name, use the string commands
210    learned above so that the names are all in trimmed and in lower case, then remove
211    any remaining true duplicates*/
212
213    replace full_name=trim(full_name)
214    replace full_name=lower(full_name)
215    duplicates drop
216
217    /*How many duplicates on id do you have now?*/
218
219    duplicates report id
220
221    /*Now split the name into f_name for their first name and l_name for their
222    last name. Drop any middle initials.*/
223
224    split full_name, p(" ")
225
226    rename full_name1 f_name
227    rename full_name2 l_name
228    replace l_name=full_name3 if full_name3!="" /*There is only a third name if they
229                                            had a middle name*/
230    replace l_name=full_name3 if length(l_name)<3  /*Another way: All the middle
231                                                names are middle initials
232                                                less than 3 characters*/
233    replace l_name=full_name3 if strpos(l_name, ".")!=0 /*Another way: All the middle
234                                                names are middle initials
235                                                with a period*/
236    drop full_name3
237
238    /*Tab  last name to view the values.  Do you notice any non-alphabetical characters?
239    Remove them from the first and/or last names*/
240
241    replace l_name=subinstr(l_name, "/", "", .)
242
243    /*Extra credit, do a google search to see how you could remove non-alphabetic characters
244    in the name varaibles without knowing which non-aphabetic characters where in the
       variable*/
245
246     capture noisily ssc install egenmore
247     egen new_l_name=sieve(l_name), keep(a)
248
249
250
251    capture log close
252    exit
253
```