

Stata Summer Series

Stata 100 – Intro to Stata

What you will get out of this session:

- » What is Stata? What can it do?
- » Why use Stata? How can we streamline and document our work?
- » How do I import, examine, and save a dataset?
- » What resources are available if I need help?

Basic command structure

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

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 101 – **Hands-on Intro Stata Workshop** (Wednesday, July 17, 2:00 pm-3:00 pm in Rm. 3030)
- » Stata 201 – **Data Cleaning** (Wednesday, July 24, 2:00 pm-3:00 pm in Rm. 3030)
- » Stata 202 – **Data Manipulation** (Wednesday, July 31, 2:00 pm-3:00 pm in Rm. 3030)
- » Stata 301 – **Automating Tasks and Exporting Output** (Wednesday, August 7, 2:00 pm-3:00 pm in Rm. 3030)

Stata 100 Training Handout

7/8/2019

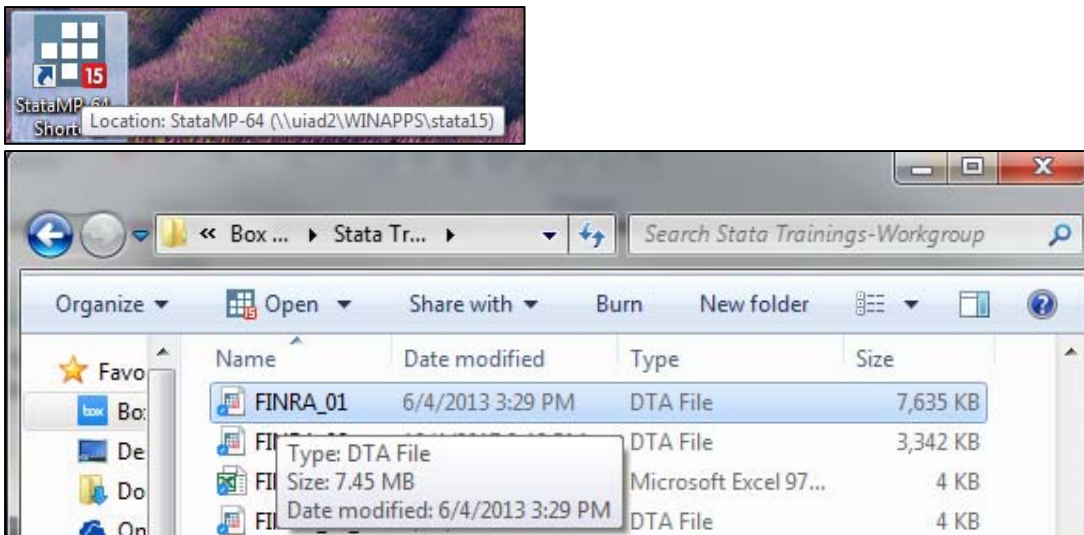
Agenda:

1. Introductions
2. What is Stata? Why use it?
3. Goal: Set up a file, have it do everything, output the things you want.
4. How to actually use Stata. Our focus today is to orient ourselves with the program and how to write commands.
 - a. In general the steps for an analysis are: Get data, clean/rework data, analyze data, and output results.

****Please feel free to ask questions at any point of the session.****

How do I open this?

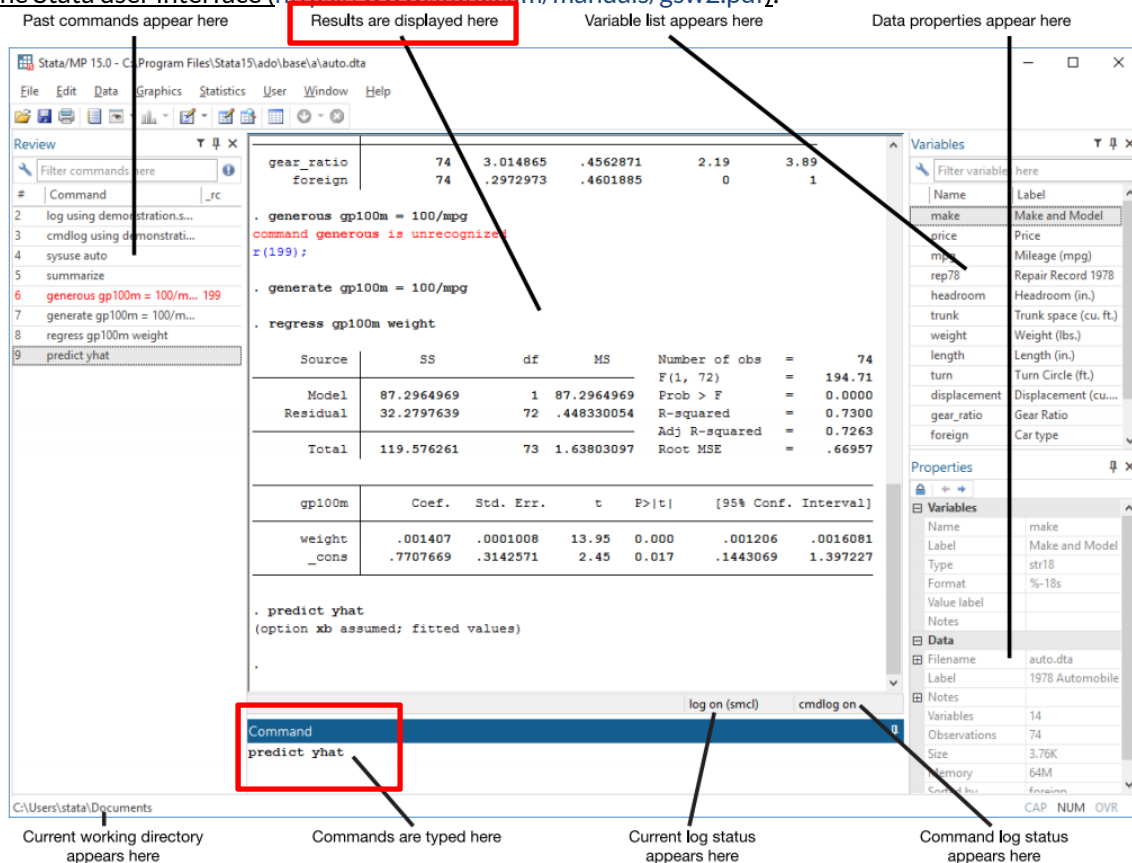
Click to open a shortcut to the Stata program (make sure IT has installed) or a Stata data (.dta), script (.do), or other file type (.log, .smcl, .ado, etc.) associated with it. It's just like Microsoft Word in that you can open it by clicking the icon or opening a word document or template.



What am I looking at?

The main Stata console that opens up is your home base. Anything that you “do” will show up here. There’s a lot happening here, so for now focus on the **Results window** (center) and **command line** (bottom).

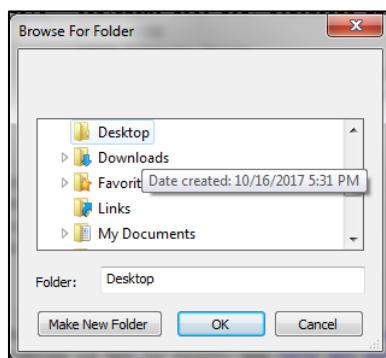
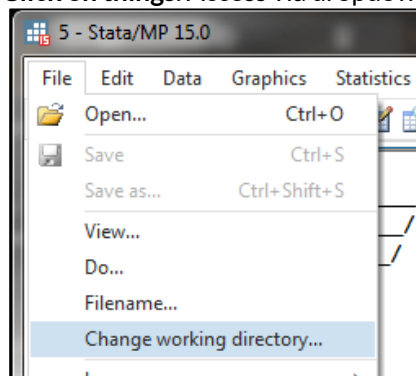
The Stata user interface (<https://www.stata.com/manuals/gsw2.pdf>):



In this handout: Courier text is Stata code, *italicized* text signifies variable names, and underlined text denotes menu selections.

Three ways to do the same thing

1. Click on things. Access via dropdown menus

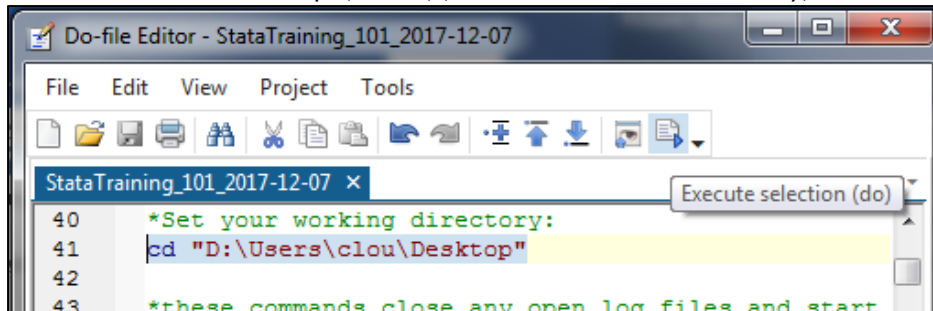


2. Type in a command in the command window:
e.g. `cd "D:\Users\clou\Desktop"`

Basic command structure

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

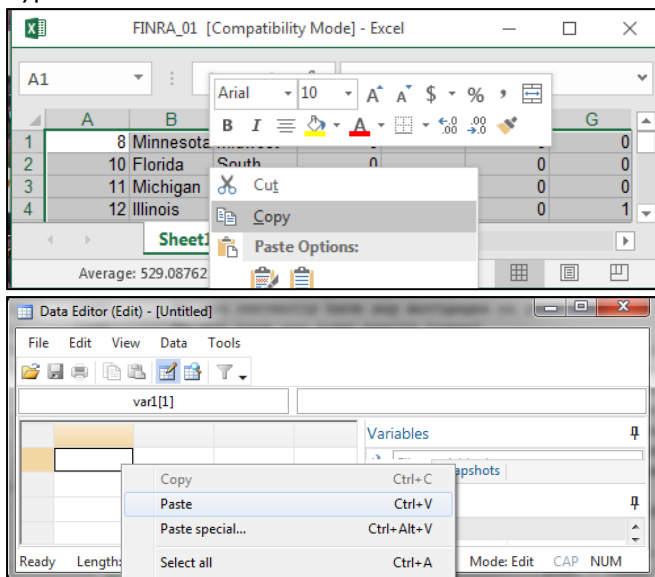
3. Run commands from a script (do-file) (what we want to do eventually)



How do I get data in here?

1. Manually enter data or copy/paste from a spreadsheet into browser in edit mode.

- Click Data>Data Editor> Data Editor (Edit) or 
- Type: `edit -`




2. Import from a non-stata file type (like how Microsoft word needs to convert a PDF before it can open it as a word doc)

- file>import>Excel Spreadsheet [or your file type]
- `import excel "FINRA_01.xls", sheet("Sheet1") clear`

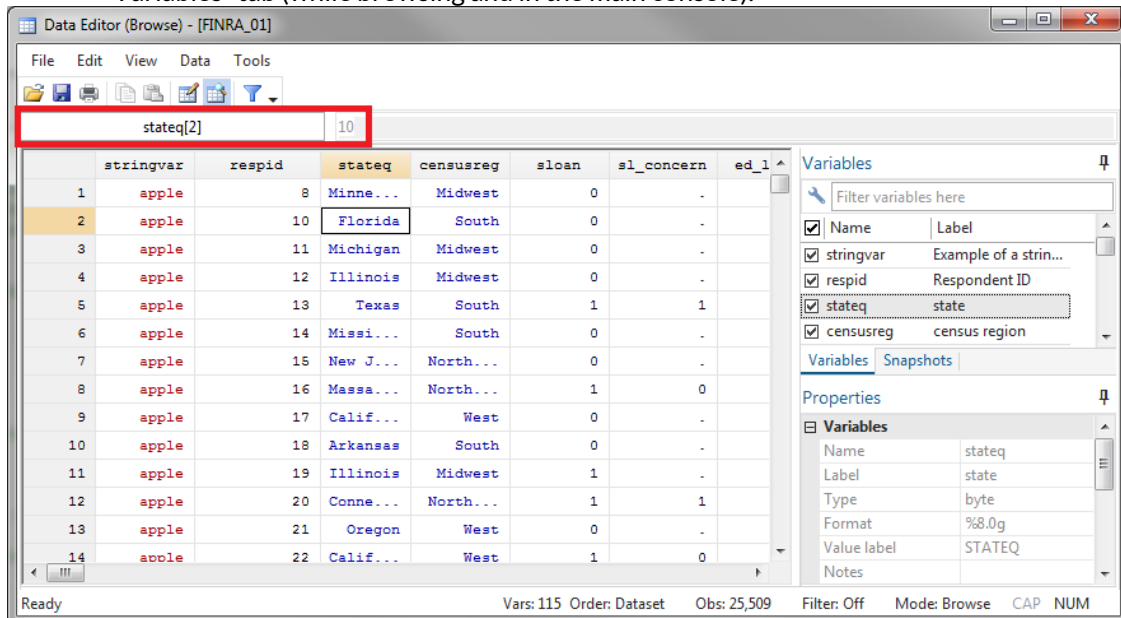
3. Open existing dataset (already a stata file)

- file>open
- `use "FINRA_01.dta", clear`

How do I look at my data?

Using the data editor in browse mode (`Data>Data Editor>Data Editor (Browse)` or  or `browse`), you can see that data stored in Stata basically looks like a spreadsheet.

- Rows = records or observations (e.g. respondents of a survey)
 - Sample size (N) is the total of all the rows.
- Columns = variables or characteristics (e.g. age, state)
 - One advantage of Stata is you can really easily search and look at groups of your variables in the “Variables” tab (while browsing and in the main console).



There are several different data types. This is important to know because Stata is rigid in how it stores data, and you will run into errors/issues if your variables are not in the right type.

- **Numeric** variables such as *respid* above contain discrete (integer) or continuous numeric data and appear in black (8, 10, etc.). You can manipulate this data like regular numbers (i.e. add values, multiply them, etc.).
 - **Dummy variables:** Numeric variables with the values 0 or 1 are a specific type of numeric variable called a dummy, binary, or indicator variable. A value of 1 means a record has the quality the variable’s name indicates.
- **String** variables contain text and appear in red (*apple*). Always use double quotes for these values in your commands and code ("*apple*"). Adding strings concatenates text and “string functions” are used to manipulate them.
- **Labeled numeric** variables: Variables whose data appears in blue like *stateq* above are labeled numeric variables. They appear as text (*Florida*) in the browser and in the output for most commands but actually have a numeric value underlying them (10 for *Florida* in the above example) which must be referenced in commands/code.
 - Labeled numeric variables often signify categorical or ordinal variables where the underlying numeric value does not contain useful information beyond degree (i.e. a red car is not “2x” a blue car).

Note: there are special values for missing data for numeric (. [period]) and string (" " [empty string]) variables. The missing value for numeric data (.) is the highest numeric value and empty strings (" ") are the lowest string value in Stata, which is important for subsetting and recoding variables (more on this below). Other special values such as “don’t know,” “refused,” etc. for numeric variables are also often coded as negative or extremely high values—refer to the data dictionary or codebook for your particular dataset.

We can see differences between these different types of data using the **display** function (basically a calculator). Add `2+2` vs. `stringvar+"2"` and display the value of the `stateq` variable for the 1st record:

```
. display 2+2           . display stringvar +"2"           . display stateq
4                        apple2                          24
```

How can I get to know my data?

One advantage of using Stata versus Excel: it's relatively easy to run diagnostics or descriptive statistics for all or part of your data set. A crucial first step in an analysis is becoming familiar with your data. Are there missing data? Do some variables have special values? Do some records look weird? Are the variables the expected format (e.g. is age a numeric variable, not a string)? There are many issues that could arise when becoming familiar with a new dataset and it's important to refer to their documentation for help.

1. **describe** provides basic information about the dataset and/or its variables including **name**, **data type**, and **label** (usually a description) if one exists. This is an easy way to see if all the variables you want are included, and if they're in the right format.

. describe				
Contains data from D:\Users\clou\Desktop\FINRA_01.dta				
obs:	25,509			
vars:	115		6 Dec 2017 12:13	
size:	7,907,790			
variable name	storage type	display format	value label	variable label
stringvar	str5	%9s		Example of a string variable
respid	long	%12.0g		Respondent ID
stateq	byte	%8.0g	STATEQ	state
censusreg	byte	%8.0g	CENSUSRE	census region
sloan	float	%9.0g		R currently has student loans
sl_concern	float	%9.0g		R concerned that s/he cannot pay back student loans
ed_lths	float	%9.0g		Education is less than High School

2. **codebook** provides more information on variables adding in **range**, **number missing**, and **number of unique values** as well as **example values** for categorical variables and **distribution** for numeric variables

respid Respondent ID				
type: numeric (long)				
range: [0,75001]				
unique values: 25,509				
missing .: 0/25,509				
mean: 30698.5				
std. dev: 21087.2				
percentiles:				
	10%	25%	50%	75%
	4034	12808	27066	48130
				62813

stateq state				
type: numeric (byte)				
label: STATEQ				
range: [1,51]				
unique values: 51				
missing .: 0/25,509				
examples:				
	11	Georgia		
	21	Maryland		
	31	New Jersey		
	41	South Carolina		

3. **list** will print a part of your data which can be useful to spot missing/special values & other issues.

```
. list stringvar respid stateq censusreg sloan sl_concern ed_lths in 1/10
```

	string~r	respid	stateq	census~g	sloan	sl_con~n	ed_lths
1.	apple	8	Minnesot	Midwest	0	.	0
2.	apple	10	Florida	South	0	.	0
3.	apple	11	Michigan	Midwest	0	.	0
4.	apple	12	Illinois	Midwest	0	.	0
5.	apple	13	Texas	South	1	1	0
6.	apple	14	Mississi	South	0	.	0
7.	apple	15	New Jers	Northeas	0	.	0
8.	apple	16	Massachu	Northeas	1	0	0
9.	apple	17	Californ	West	0	.	0
10.	apple	18	Arkansas	South	0	.	0

How can I begin to see patterns and relationships in my data?

Stata can provide summary and descriptive statistics of your data faster than in Excel. The main relationships that tell you about your data are measures of central tendency (mean, median, mode) and spread/variability (range, standard deviation, variance).

1. Run descriptive statistics of your variables using
 - o **tabulate** for categorical or ordinal (e.g. gender, educational level)
 - Takes all a variable's observations and gives you the frequency and percent of each value (among the total observations)
 - o **summarize** for discrete or continuous numeric (e.g. age, wage)
 - Get the number of observations, mean, standard deviation, and range
 - Mean of a binary variable is the share of the total with that characteristic
 - o either **tabulate** or **summarize** for dummies/binary/indicators:

Summarizing the age and binary white variables

```
. summarize A3A r_white
```

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	25,509	47.00588	16.07551	18	101
r_white	25,509	.7336626	.4420514	0	1

Tabulating the education category variable

```
. tabulate ed_catvar
```

ed_catvar	Freq.	Percent	Cum.
Less than High School	1,903	7.46	7.46
High School or equivalent	6,561	25.72	33.18
Some College	8,419	33.00	66.18
College	5,343	20.95	87.13
Postgraduate Degree	3,283	12.87	100.00
Total	25,509	100.00	

Tabulating student loans with missing option to show . values

```
. tabulate sloan, missing
```

R currently has student loans	Freq.	Percent	Cum.
0	20,049	78.60	78.60
1	5,141	20.15	98.75
.	319	1.25	100.00
Total	25,509	100.00	

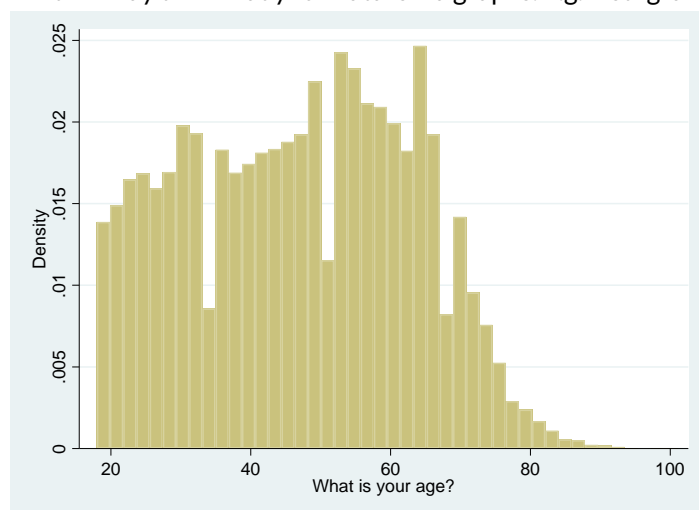
2. You can also crosstab two categorical/dummy variables with tabulate:

Below, we can easily see that most people who have student loans are in the categories “Some college” or “College.”

```
. tabulate ed_catvar Sloan, missing
```

ed_catvar	R currently has student loans			Total
	0	1	.	
Less than High School	1,765	94	44	1,903
High School or equiva	5,849	612	100	6,561
Some College	6,311	2,013	95	8,419
College	3,739	1,551	53	5,343
Postgraduate Degree	2,385	871	27	3,283
Total	20,049	5,141	319	25,509

3. Another way to look at your data is via graphs. E.g. histogram showing the distribution of age:



What if I only want to see descriptives for a subset of all the observations?

Stata can also work with a subset of your data more easily than Excel. What if I want to know the mean of everyone's age, but only for people with postgraduate degrees? What if I want the educational attainment of only college-aged students?

Use an **if** conditional statement (always before the comma for options) to specify the particular observations you want a command to operate on. **if** expressions use common comparator operators to specify one or more conditions observations must meet for the observations to be included in the operation:

- equals (==)
- not equals (!= or ~=)
- greater than (>)
- less than (<)
- greater than or equal to (>=),
- less than or equal to (<=)

You can combine operators with the following Booleans:

- and (&)
- or (|)

E.g. provide statistics of age or education observations below age 50:

```
. summarize A3A if A3A < 50
```

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	13,507	34.19094	9.127357	18	49

```
. tabulate ed_catvar if A3A < 50
```

ed_catvar	Freq.	Percent	Cum.
Less than High School	1,313	9.72	9.72
High School or equivalent	3,350	24.80	34.52
Some College	4,307	31.89	66.41
College	3,091	22.88	89.29
Postgraduate Degree	1,446	10.71	100.00
Total	13,507	100.00	

Creating an expression with **in** instead of **if** will specify a subset of observations based on their record number/order rather than a set of conditions. E.g. we already used **list** to print out the values of a few variables for the first ten observations by including an expression with **in** (**in 1/10**):

```
. list stringvar respid stateq censusreg sloan sl_concern ed_lths in 1/10
```

	string~r	respid	stateq	census~g	sloan	sl_con~n	ed_lths
1.	apple	8	Minnesot	Midwest	0	.	0
2.	apple	10	Florida	South	0	.	0
3.	apple	11	Michigan	Midwest	0	.	0
4.	apple	12	Illinois	Midwest	0	.	0
5.	apple	13	Texas	South	1	1	0
6.	apple	14	Mississi	South	0	.	0
7.	apple	15	New Jers	Northeas	0	.	0
8.	apple	16	Massachu	Northeas	1	0	0
9.	apple	17	Californ	West	0	.	0
10.	apple	18	Arkansas	South	0	.	0

How can I add or change variables?

So far, we've only looked at the variables already in the dataset. However, data rarely come perfectly. We often want to create new variables based on the variables in the dataset. It's easy to create new variables (columns) with whatever value you would like to assign, based on the values of one or more existing variables, or more complex expressions as well as to update values. There are two main commands here:

1. **generate:** Create new variables with the `generate` command, a new variable name, and assign it (=) to some initial value. Command format:
`generate varname = value`

Create a new binary variable called `a1824` indicating respondents ages 18-24:

```
. generate a1824 = 0
```

	A3A	a1824	stringvar	respid
1	62	0	apple	8
2	55	0	apple	10
3	72	0	apple	11
4	69	0	apple	12
5	27	0	apple	13
6	33	0	apple	14
7	68	0	apple	15
8	18	0	apple	16
9	44	0	apple	17

2. **replace:** The **replace** command will update the value(s) of records for an existing variable and has similar syntax to `generate`: `replace varname = newvalue`. It is often combined with `if` or `in` to update the values of only a subset of records:
`. replace a1824 = 1 if A3A >= 18 & A3A < 25`
 (2,581 real changes made)

Data Editor (Browse) - [FINRA_01.dta]

File Edit View Data Tools

a1824[1] 0

	A3A	a1824	stringvar	respid
1	62	0	apple	8
2	55	0	apple	10
3	72	0	apple	11
4	69	0	apple	12
5	27	0	apple	13
6	33	0	apple	14
7	68	0	apple	15
8	18	1	apple	16
9	44	0	apple	17

3. When you generate or replace variables, confirm whether they were created as expected by crosstabbing vs. other (original) variable(s) or summarizing.

```
. summarize A3A if a1824 == 0
```

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	22,928	49.91787	14.25653	25	101

```
. summarize A3A if a1824 == 1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	2,581	21.13754	2.006788	18	24

****Most mistakes are human errors, and most of these are simple typos. Checking that the new variables you created have the correct range (e.g. there are no negative age values) and "look right" can save you a lot of time and trouble in the long run.****

4. You can also create categorical variables:

```
. gen age_cat = .
(25,509 missing values generated)

. replace age_cat = 1 if A3A > 17 & A3A < 25
(2,581 real changes made)

. replace age_cat = 2 if A3A > 24 & A3A < 61
(16,843 real changes made)

. replace age_cat = 3 if A3A > 60
(6,085 real changes made)
```

```
. tab A3A age_cat , m
```

What is your age?	age_cat			Total
	1	2	3	
18	374	0	0	374
19	294	0	0	294
20	322	0	0	322
21	395	0	0	395
22	408	0	0	408
23	385	0	0	385
24	403	0	0	403
25	0	408	0	408
26	0	361	0	361
27	0	406	0	406
28	0	380	0	380

5. A variable whose value is a transformation of the value of another variable. Create age in months from age in years:

```
. gen age_in_months = A3A*12

. sum age_in_months A3A
```

Variable	Obs	Mean	Std. Dev.	Min	Max
age_in_mon~s	25,509	564.0706	192.9061	216	1212
A3A	25,509	47.00588	16.07551	18	101

6. Variables based on more than one other variable. Below we create a general has bank account dummy variable based on separate checking and savings account dummies:

```
. gen bankacct = .
(25,509 missing values generated)

. replace bankacct = 1 if B1 == 1
(22,948 real changes made)

. replace bankacct = 1 if B2 == 1
(598 real changes made)

. replace bankacct = 0 if B1 == 2 & B2 == 2
(1,558 real changes made)
```

. tab bankacct B1 if B2 != 1, m					
bankacct	Do you [Does your household] have a checking account?				Total
	Yes	No	Don't kno	Prefer no	
0	0	1,558	0	0	1,558
1	4,522	0	0	0	4,522
.	0	20	95	290	405
Total	4,522	1,578	95	290	6,485

. tab bankacct B2 if B1 != 1, m					
bankacct	Do you [Does your household] have a savings account, money market account, or CD				Total
	Yes	No	Don't kno	Prefer no	
0	0	1,558	0	0	1,558
1	598	0	0	0	598
.	0	65	90	250	405
Total	598	1,623	90	250	2,561

Per the above example: it is often best practice to start by setting a new variable to missing so that missing (.), the highest value, and other special values are not accidentally coded to a valid value.

A word on variable names

Good variable names are concise, yet descriptive.

Bad name	Reason it's bad	Better name
A2BSUS	Not descriptive and hard to remember	edu_cat
ckorsavebankaccountnumber	Too long	bankacct
YearsAlive_2018	Roundabout way of saying age; mixes upper and lower case letters, which is annoying to type every time	age

How do I save my work?

Now that you've added the variables you want, you want to save your dataset so that you don't have to recreate the variables each time you want to analyze the data. To save a Stata dataset, type:

```
save "Finra_02.dta", replace
```

This will save whatever is in the memory (you can check `browse` to see). Be sure not to save it as a new name so you don't overwrite the original data. You can also save other types of files, which we'll get to later in a later session.

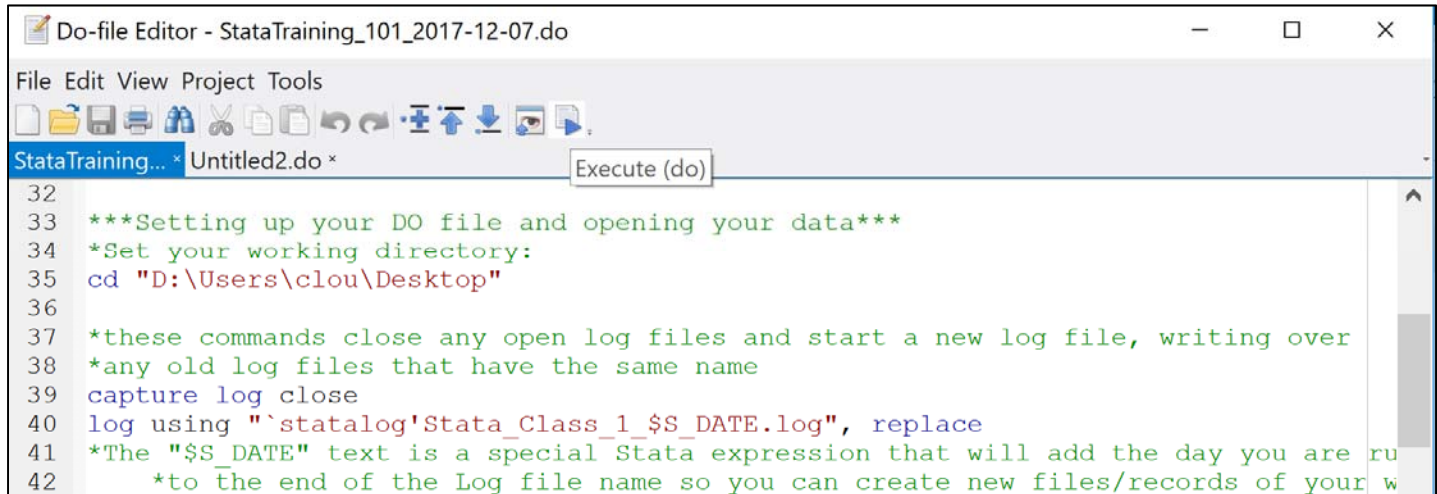
I can do all of this in Excel. Why shouldn't I?

Putting what you've just learned into a script will allow you to save, record, and replicate your work. The biggest advantage of using Stata or a similar statistical programming language— even more than statistical modeling, I think— is to allow you or others to easily save work, record results, and reproduce or modify an analysis.

Goal: set up a do file that does everything you need to do, run it, and examine the output.

1. When you move all the commands from the command line to Stata scripts called `do` files (text documents containing a series of commands) you can modify, save, and run through your entire program without typing in each line.

```
. doedit "StataTraining_101_2017-12-07.do"
```

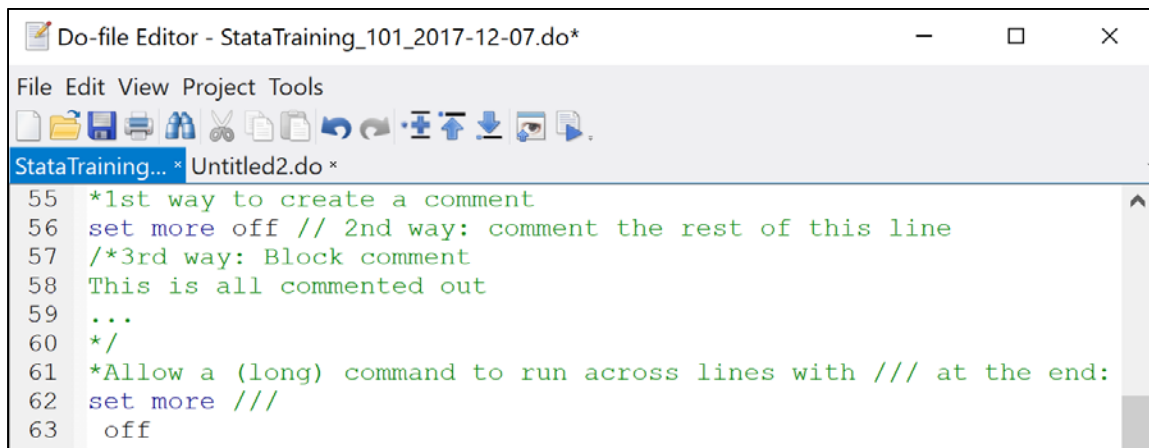


The screenshot shows the Stata Do-file Editor window titled "Do-file Editor - StataTraining_101_2017-12-07.do". The menu bar includes File, Edit, View, Project, and Tools. The toolbar contains icons for file operations and execution. The editor shows a script with the following content:

```
32
33 ***Setting up your DO file and opening your data***
34 *Set your working directory:
35 cd "D:\Users\clou\Desktop"
36
37 *these commands close any open log files and start a new log file, writing over
38 *any old log files that have the same name
39 capture log close
40 log using "`statalog'Stata_Class_1_$$_DATE.log", replace
41 *The "$$_DATE" text is a special Stata expression that will add the day you are ru
42 *to the end of the Log file name so you can create new files/records of your w
```

Run commands in `do` file by clicking the **Execute (do)** button that looks like a paper with a play sign (can run the whole thing or just selected lines by highlighting them) or by using the `do` command. You should concentrate on using `do` files going forward as they allow you to save and reproduce your analysis; the one above will run through this entire training and more supplementary material on top of it.

- Your `do` file code should include comments (the text in **green**) which will help guide you the next time you work on a project or someone new to the project or taking over your work. Specify comments with a single star (*) at the beginning of a line, double forward slash (//) to comment out the rest of a line, or /* */ for a block that will comment out everything between the stars and can go across one or more lines.
 - Stata commands usually have to live a single line, but you can use block comments or triple-slash (///) at the end of a line to continue a command to the next line.



The screenshot shows the Stata Do-file Editor window titled "Do-file Editor - StataTraining_101_2017-12-07.do*". The menu bar and toolbar are the same as in the previous screenshot. The editor shows a script with the following content:

```
55 *1st way to create a comment
56 set more off // 2nd way: comment the rest of this line
57 /*3rd way: Block comment
58 This is all commented out
59 ...
60 */
61 *Allow a (long) command to run across lines with /// at the end:
62 set more ///
63 off
```

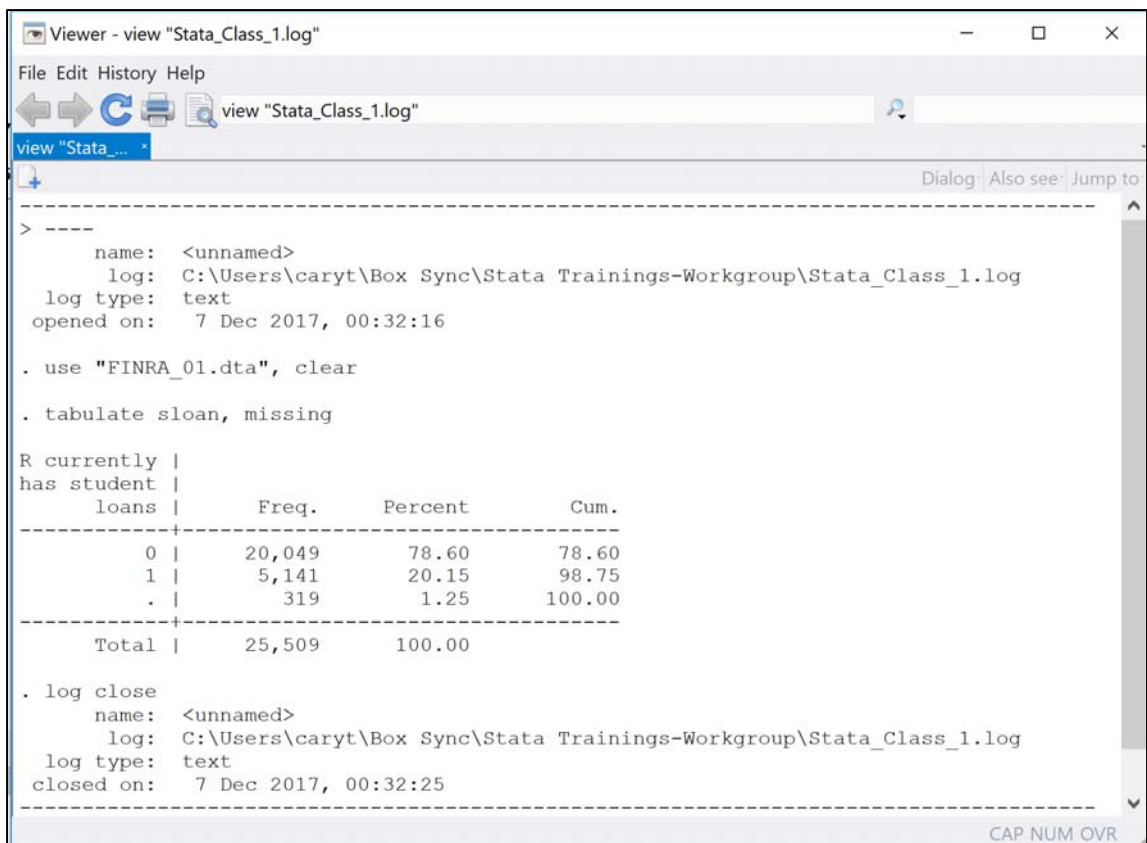
How do I keep track of everything that happens in my do file?

You've set up your do-file with all your commands (your code). You run your commands. How do you check what's happening? How can you present it to someone without them having to open and run the program?

Start a log file at the beginning of your do file. The log file will capture whatever appears in the main results window (both the commands and their output) until you close it in a separate file under the name you specify providing a record of your work.

. log using "Stata_Class_1.log", replace			
name: <unnamed>			
log: C:\Users\caryt\Box Sync\Stata Trainings-Workgroup\Stata_Class_1.log			
log type: text			
opened on: 7 Dec 2017, 00:32:16			
. use "FINRA_01.dta", clear			
. tabulate sloan, missing			
R currently has student loans	Freq.	Percent	Cum.
0	20,049	78.60	78.60
1	5,141	20.15	98.75
.	319	1.25	100.00
Total	25,509	100.00	
. log close			
name: <unnamed>			
log: C:\Users\caryt\Box Sync\Stata Trainings-Workgroup\Stata_Class_1.log			
log type: text			
closed on: 7 Dec 2017, 00:32:25			

(output)



The screenshot shows a Stata log viewer window titled "Viewer - view 'Stata_Class_1.log'". The window contains the following text:

```
> ----
      name: <unnamed>
      log: C:\Users\caryt\Box Sync\Stata Trainings-Workgroup\Stata_Class_1.log
      log type: text
      opened on: 7 Dec 2017, 00:32:16

. use "FINRA_01.dta", clear

. tabulate sloan, missing

R currently |
has student |
  loans |      Freq.    Percent    Cum.
-----+-----
      0 |      20,049      78.60     78.60
      1 |       5,141     20.15     98.75
      . |       319      1.25    100.00
-----+-----
    Total |      25,509    100.00

. log close
      name: <unnamed>
      log: C:\Users\caryt\Box Sync\Stata Trainings-Workgroup\Stata_Class_1.log
      log type: text
      closed on: 7 Dec 2017, 00:32:25
-----
```

At the bottom right of the window, the text "CAP NUM OVR" is visible.

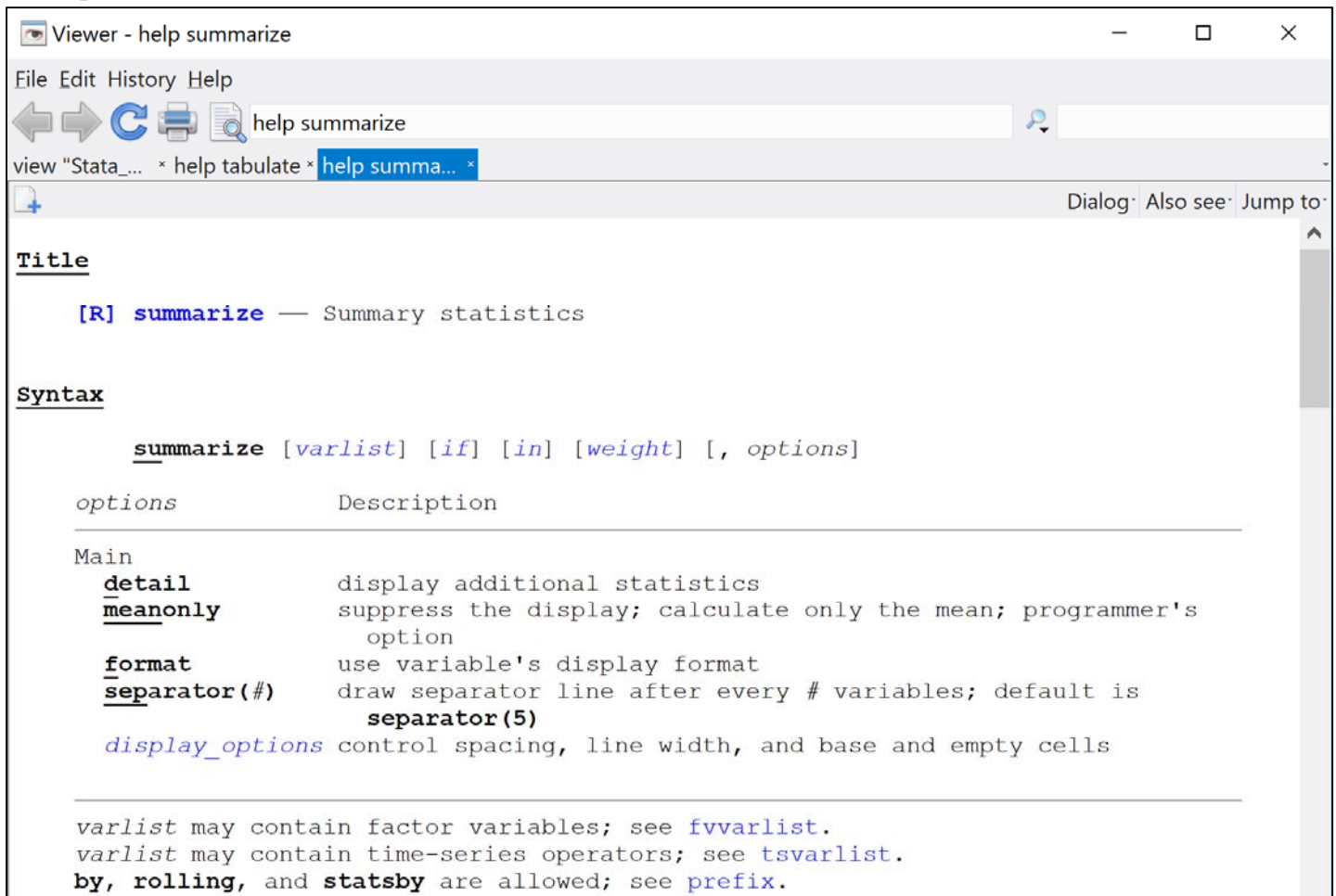
(captured in log)

Generally run your entire *do* file through once the analysis setup is final to create a "clean" log file. It is also best practice to save a new version of your data under a different file name, often at the beginning or end of your program, so that you do not accidentally overwrite your original data source.

What if I run into errors or want to learn a new command?

Help files can teach you more about basic and more advanced Stata commands and how to use them. Once you understand the syntax and setup of a help file, you can learn almost any Stata command or concept:

```
. help summarize
```



The screenshot shows a Stata Viewer window titled "Viewer - help summarize". The window has a menu bar with "File", "Edit", "History", and "Help". Below the menu bar is a toolbar with icons for back, forward, search, and other functions. The main content area displays the help text for the `summarize` command. The text is formatted with bold and italicized keywords. The window also shows a tab bar at the top with "view 'Stata_...' x help tabulate x help summa... x".

Title

[R] summarize — Summary statistics

Syntax

summarize [*varlist*] [*if*] [*in*] [*weight*] [, *options*]

<i>options</i>	Description
Main	
<u>detail</u>	display additional statistics
<u>meanonly</u>	suppress the display; calculate only the mean; programmer's option
<u>format</u>	use variable's display format
<u>separator</u>(#)	draw separator line after every # variables; default is separator(5)
<i>display_options</i>	control spacing, line width, and base and empty cells

varlist may contain factor variables; see [fvvarlist](#).
varlist may contain time-series operators; see [tsvarlist](#).
by, **rolling**, and **statsby** are allowed; see [prefix](#).

There's much more you can "do" with Stata including more advanced data cleaning as well as manipulating your dataset by merging with other data, reshaping, etc. There is more next week on these more advanced operations.

- In the meantime, play around with this and your own `do` file and try the exercises at the end on your own without looking at solutions.
- Printouts of the `do` and `log` files from this training are on the following pages.


```

1 ** LOCATION: D:\Users\clou\Desktop\Do
2 ** CREATED BY: Emma Kalish
3 ** CREATED ON: 7/20/15
4 ** LAST EDITED: 6/29/18 by Hannah Hassani
5 ** LAST RUN:
6 ** DESCRIPTION: Example do file for Stata class
7 ** NOTES: Uses Stata15
8 ****
9
10 *Review Stata interface/windows
11     *Results (main) window: shows the commands that are run and the resulting output (errors show up in red text)
12     *Variables window: shows variables in current dataset including any labels
13     *Properties window: detailed characteristics of dataset and its variables as well as variable(s) selected in the Variables
    window
14     *Review window: shows previous commands entered and if they resulted in an error (if text is in red)
15     *Command window: allows users to type in commands to be executed directly/interactively
16
17 *Basic structure of command -> [commandname][what (e.g., variable(s), file, etc.), [options]
18
19 *Although you can run commands by typing them into the command window and using the dropdown menus,
20 *The real power of Stata lies in using it as a statistical programming language.
21 *That is creating scripts/programs in the form of DO files that allow you to save and reproduce your analysis.
22
23 *The purpose of using DO/Log files = recording and replicating your work (you can also steal from old DO files for new analyses)
24 *To this end, create new versions of DO, LOG, and DATA files rather than saving over old ones.
25
26 **3 Main Types of Stata Files:
27 *Data files (end in .dta) look like spreadsheets and contain the information you want to analyze
28 *DO files (end in .do) are essentially Stata programs that allow you to save and re-run/reproduce your analysis from scratch
29 *LOG files (end in .log [unformatted and can open with any text editor] or .smcl [formatted, but can only open with Stata])
30     *capture the results (what appears in the main results window) of your commands/DO file and contain a record of the
    steps taken in your analysis.
31
32
33 ***Setting up your DO file and opening your data***
34 *Set your working directory:
35 cd "D:\Users\hhassani\Desktop"
36
37 *This command allows do file to run continuously rather than having to click "more" on the screen
38 set more off, permanently
39
40 **Commenting your code
41 *Your DO file should contain comments to document your work allowing others to follow your work and
42 *reminding yourself of what you were doing later. There are a few ways to create comments,
43 *you can start a line with an * (star/asterix) to make the line a comment.
44 *You can also follow a command with a // (double forward slash) to make the REST of the line a comment
45 *To create a multi-line comment block start with /* and end with */
46 * /// (three forward slashes allow you to continue a command across multiple lines
47
48 *1st way to create a comment
49 set more off // 2nd way: comment the rest of this line
50 /*3rd way: Block comment
51 This is all commented out
52 ...
53 */
54 *Allow a (long) command to run across lines with /// at the end:

```

```

55 set more ///
56 off
57
58 *these commands close any open log files (capture log close) and
59 *start a new log file (log using) to capture your output and results,
60 *the replace option (options always follow a comma) writes over
61 *any old log files that have the same name in your working directory
62 capture log close
63 log using "Stata_Class_1.log", replace
64
65
66 ***Loading in your data***
67 *1) Copy and paste data from another program into the data editor or manually enter it:
68 edit
69 *2) Use the import command or wizard to directly bring in data from a file in another format
70 import excel "FINRA_o1.xls", sheet("Sheet1") clear
71 *3) load the data you want to use with the Stata "use" command
72     *add clear as an option at the end after a comma to empty any data that was in before
73 use "FINRA_o1.dta", clear
74 * This is a file containing survey data on individuals background and financial situation
75
76
77 ***Getting to know your data***
78 *Inspect/get a detailed look at your data using the data editor in browse mode:
79 browse
80     *rows are observations/records; columns are variables/characteristics/features
81     *Red data is coded as string, black is numeric, and blue is numeric with text labels.
82
83 *We can see this by using the 'display' command to turn Stata into a calculator:
84 *Adding 2+2 with numeric values
85 display 2+2
86 *Adding 'stringvar'+"2" with string values
87 display stringvar+"2"
88 *Show the value of the 'stateq' variable (will show the value of the 1st record)
89 display stateq
90
91 *Describe will provide basic information on your data set and its variables:
92 describe
93 *Codebook provides more details on specific variables including missingness, range, example values or distribution
94 codebook stringvar respid stateq censusreg sloan sl_concern ed_lths
95 *List will print out values for the variables and records/observations specified
96 list stringvar respid stateq censusreg sloan sl_concern ed_lths in 1/10
97
98
99 ***Descriptive statistics***
100 *You may want to examine variables of interest for your analysis more closely
101 *or include descriptive statistics for them in your study
102
103 *Use summarize for discrete or continuous numeric variables,
104 *tabulate for ordinal or categorical variables,
105 *and either for dummies/binary/indicators:
106
107 *summarize the age variable
108 summarize A3A
109 *or we can summarize more than one variable at a time
110 *(age and whether observation is white):

```

```

111 summarize A3A r_white
112 *tabulate education category variable
113 tabulate ed_catvar
114 *tabulate student loan reciprocity variable with the missing option (, missing)
115     *at the end of the command after a comma to show any missing values
116     *(coded as . for numeric data and "" for string data)
117 tabulate sloan, missing
118 *You usually don't have to type out the whole expression,
119 *Stata will know what you mean if you abbreviate as long as
120 *there is no ambiguity with other commands, variables, options, etc.
121 tab sloan, m
122 *Another option: see a labeled numeric variable without its value labels
123     *(i.e. see the underlying numeric values)
124 tab B1
125 tab B1, nolabel
126 *Also, use tabulate with two categorical variables to show their crosstab:
127 tab ed_catvar sloan , missing
128 tab ed_catvar sloan, column //the column option reports the % within each row that are in each column category (%s in
    columns sum to 100%)
129 tab ed_catvar sloan, row //the row option reports the % within each column that are in each row category (%s in rows sum to
    100%)
130 tab ed_catvar sloan, cell //the cell option reports the % in the entire table total that are in each cell (%s in cells sum to 100%)
131
132 *another way to look at your data - graphs
133 *Histogram showing the distribution of age:
134 histogram A3A
135
136
137 ***Subsetting your data***
138 *IF and IN statements allow you to operate on subsets of your data
139 *IN statements define subsets based on records' index (observation) number
140 *IF statements define subsets based on conditional statements
141     *& is AND
142     *| is OR
143     *Use == to compare the equality of two values
144     * != is not equal to (! is NOT in general)
145     * > is greater than
146     * < is less than
147     * >= is greater than or equal to (= must come after < or >, not before or it will not work, i.e. >= is CORRECT; => is
    INCORRECT)
148     * <= is less than or equal to
149 *Summarize age for just observations less than age 50:
150 summarize A3A if A3A < 50
151 *Tabulate education for just observations less than age 50:
152 tabulate ed_catvar if A3A < 50
153 *or just those whose age is equal to 50:
154 tabulate ed_catvar if A3A == 50
155
156 *We already used an 'in' expression with 'list' above to show the values of
157 *some of our variables for the first ten observations:
158 list stringvar respid stateq censusreg sloan sl_concern ed_lths in 1/10
159
160
161 ***Creating new variables and updating existing variables***
162 *Be mindful of missing ("" if string or . if numeric ) values as well as special values codes
163 *Special values are often negative or high values like 9998, 9999, etc.

```

```

164 *and can indicate "don't know," "refused," etc. in survey data.
165
166 *always inspect your data initially via tab, summ, etc. to look for these and deal with them appropriately
167 *BEFORE constructing variables or starting your actual analysis
168
169 *binary (0/1) age variable
170 *The 'generate' creates a new variable with the name specified and set to the value after the equals sign
171 generate a1824 = 0
172 *The 'replace' command updates the values of an existing variable;
173     *often you'll combine it with if statements to change the values of just a subset of observations
174 replace a1824 = 1 if A3A >= 18 & A3A < 25
175 *look at your variable after you make it
176 tab a1824, m
177 *And compare it to the original
178 summarize A3A if a1824 == 0
179 summarize A3A if a1824 == 1
180 *Finally, label your variable
181 label variable a1824 "age between 18 and 24"
182
183 *You can also create categorical variables
184 gen age_cat = .
185 replace age_cat = 1 if A3A > 17 & A3A < 25
186 replace age_cat = 2 if A3A > 24 & A3A < 61
187 replace age_cat = 3 if A3A > 60
188 *label your variable
189 label variable age_cat "age categories"
190 *label the values of your variable
191 label define age 1 "18-24" 2 "25-60" 3 "more than 60"
192 label values age_cat age
193 tab age_cat, m
194 *Confirm that you created variable correctly by crosstabbing vs. original variable(s)
195 tab A3A age_cat, m
196
197 **Other kinds of variables to create
198 *Scaling/transformations
199 *Create an age in months variable based on the age in years:
200 gen age_in_months = A3A*12
201 sum age_in_months A3A
202 *Create age-squared
203 gen age_squared = A3A*A3A
204 sum age_squared A3A
205 *Create logged-age
206 gen age_logged = ln(A3A)
207 sum age_logged A3A
208
209 *Create a new variable based on the value of multiple other variables:
210 *Create a new dummy variable based on a few other dummies rather than just one
211 *B1 = checking acct; B2 = savings acct
212 gen bankacct = .
213 replace bankacct = 1 if B1 == 1
214 replace bankacct = 1 if B2 == 1
215 replace bankacct = 0 if B1 == 2 & B2 == 2
216 *could also have used an OR statement instead of an AND statement - depends on how you want to define things
217 replace bankacct = 0 if (B1 == 2 | B2 == 2) & bankacct ==.
218 *Confirm that you created variable correctly by crosstabbing vs. original variable(s)
219 tab bankacct B1 if B2 != 1, m

```

```

220 tab bankacct B2 if B1 != 1, m
221
222 *Interaction variable: multiple the values of the female indicator and age
223     *to create a variable that captures females age but is 0 for males
224 gen female_age = g_female*A3A
225 *Compare the original and new variable for females and non-females:
226 sum female_age A3A if g_female == 1
227 sum female_age A3A if g_female == 0
228
229
230 ***Stata help: once you can use a Stata help file, you should be able to figure out almost any command!
231 help summarize
232 *Basic syntax of a Stata command:
233 *commandname expression if/in expression , options
234 *The 1st expression can contain variable names, assignment clauses, subcommands, etc. and depend on the particular
    command
235 *The if/in statement is followed by a 2nd expression defining the subset of the data set you want the command to work on
236 *Options always follow a single comma and are also command specific/dependent.
237 *Reference the help file for more on options, syntax, etc. for specific commands
238
239 *More on syntax is available here: http://www.stata.com/manuals13/gsw10.pdf
240
241 *A list of basic Stata commands is available at:
242 * http://www.stata.com/manuals13/u27.pdf
243 * and
244 * https://people.ucsc.edu/~aspearot/Econ113W13%20/basic\_tutorial\_stata.pdf
245
246
247 *always save with a new name, do not overwrite your data.
248 save "FINRA_o2.dta", replace
249 log close
250
251
252 *****
253 **EXERCISES**
254 *****
255 *1. Start a separate, new log file and open up the original FINRA_o1 data set
256
257 *Start new Log file
258 capture log close
259 log using "Stata_Class_1_EXERCISES_$S_DATE.log", replace
260 *The "$S_DATE" text is a special Stata expression that will add the day you are running the program
261     *to the end of the Log file name so you can create new files/records of your work automatically every day.
262
263 *Open base data set
264 use "FINRA_o1.dta", clear
265
266 *2. Provide descriptive statistics of variable G22.
267     *How many and what percentage of records have the value "Don't know", "Prefer not to say", and missing?
268     *How are "Don't know" and "Prefer not to say" coded in the data?
269
270 *Determine what type of variable G22 is (dummy/categorical/continous)
271 codebook G22
272 *Show descriptive statistics of G22 using tabulate since it looks to be categorical
273     *(use summarize to describe continous variables usually; could use either command for dummies)
274 tab G22, m //don't forget the missing option to show what % of all observations have a missing value for this variable: Share

```

```

" Don't know" = 0.93% , Share "Prefer not to say" = 0.05% , and Share missing (.) = 79.85%
275 *Add the "nolabel" option to see how "Don't know" and "Prefer not to say" are coded numerically
276 tab G22, m nolabel // "Don't know" is coded as 98; "Prefer not to say" is coded as 99
277
278
279 *3. Create a new version of this variable called G22_clean that recodes "Don't know" and "Prefer not to say" to missing
280     *Crosstab the new and old versions of the variable so that you can confirm you created it correctly.
281
282 *Start by setting the new variable to missing so that "Don't know" and "Prefer not to say" are recoded to missing automatically
283 gen G22_clean = .
284 *Then update the values of the new variable using replace to the value of the old variable only when they are "valid"
285 replace G22_clean = 1 if G22 == 1
286 replace G22_clean = 2 if G22 == 2
287 *I can also create labels to describe the new variable and its new values
288 *label the variable
289 label var G22_clean "Clean version of G22"
290 *label its values
291 label define G22_clean_label 1 "Yes" 2 "No"
292 label value G22_clean G22_clean_label
293 *Finally check the new vs. old variable using tabulate to confirm it was created correctly
294 tab G22 G22_clean, m // looks good
295
296
297 *4. What is the average number of dependent children (depchild)? What is the 25th percentile? 75th percentile?
298
299 *You can get "standard" descriptive statistic percentiles by adding the detail option to the summarize command:
300 summ depchild, d //the 25th %tile is 0; the 75th %tile is 1.
301
302
303 *5. Use HELP to figure out how to use the "centile" command to produce the 20th, 40th, 60th, and 80th percentile
304     *for the "wgt_n2" variable.
305
306 *Pull up the Stata help file
307 help centile
308 *Use centile to get the 20th, 40th, 60th, and 80th percentile of "wgt_n2" since summarize, detail does not provide these
309 centile wgt_n2, centile(20 40 60 80) //the syntax for centile is a little tricky as the option needed to specify the specific
    percentile cuts to show repeats the command name
310 *The 20th %tile is .4376673; the 40th %tile is .6453935; 60th %tile is 1.083546; 80th %tile is 1.534317
311
312
313 *6. Create a new categorical version of the weight variable called "wgt_n2_quintile" that contains information on
314     *which quintile each record's weight "wgt_n2" is in using the results of the centile command from question 5.
315
316 *Create this new variable and set to missing initially
317 gen wgt_n2_quintile = .
318 *Update the value with the percentile number; make sure the ranges you use in the "if" statements reflect what you really
    want and do not overlap
319 replace wgt_n2_quintile = 1 if wgt_n2 < .4376673
320 replace wgt_n2_quintile = 2 if wgt_n2 >= .4376673 & wgt_n2 < .6453935
321 replace wgt_n2_quintile = 3 if wgt_n2 >= .6453935 & wgt_n2 < 1.083546
322 replace wgt_n2_quintile = 4 if wgt_n2 >= 1.083546 & wgt_n2 < 1.534317
323 replace wgt_n2_quintile = 5 if wgt_n2 >= 1.534317 & wgt_n2 < . // Missing (.) is the highest numeric value in Stata,
324                                     *so specifying that the variable range should be
    less than missing here
325
326                                     *will make sure that if any missing values exist,
    *they will not get accidentally coded into the

```

```
5th quintile (we only want to count valid values)
327 *Tab to describe this new variable
328 tab wgt_n2 Quintile, m
329 *Confirm it was created correctly by summarizing the original variable by their value in the the new version fo the variable
330 summ wgt_n2 if wgt_n2_Quintile == 1
331 summ wgt_n2 if wgt_n2_Quintile == 2
332 summ wgt_n2 if wgt_n2_Quintile == 3
333 summ wgt_n2 if wgt_n2_Quintile == 4
334 summ wgt_n2 if wgt_n2_Quintile == 5
335 *The min and max value of the original variable for each group indicate that the quintile variable was created correctly
336
337 *7. Save a new version of the data file called FINRA_o3.dta,
338 *close your log,
339 *and then inspect your log by navigating to where you saved it and opening it with notepad.
340
341 *save new version of the data
342 save "FINRA_o3.dta", replace
343 *close log
344 log close
345 *view log by navigating to it and opening it with Stata or notepad.
346 *Mine is in my working directory "D:\Users\hhassani\Desktop"
347
348
349
```

Stata_Class_1

```

name: <unnamed>
log: D:\Users\hhassani\Desktop\Stata_Class_1.log
log type: text
opened on: 29 Jun 2018, 09:59:02

```

```

.
.
. ***Loading in your data***
. *1) Copy and paste data from another program into the data editor or manually enter it:
. edit

. *2) Use the import command or wizard to directly bring in data from a file in another
format
. import excel "FINRA_01.xls", sheet("Sheet1") clear

. *3) load the data you want to use with the Stata "use" command
.      *add clear as an option at the end after a comma to empty any data that was in
before
. use "FINRA_01.dta", clear

. * This is a file containing survey data on individuals background and financial situation
.
.
. ***Getting to know your data***
. *Inspect/get a detailed look at your data using the data editor in browse mode:
. browse

.      *rows are observations/records; columns are variables/characteristics/features
.      *Red data is coded as string, black is numeric, and blue is numeric with text
labels.

. *We can see this by using the 'display' command to turn Stata into a calculator:
. *Adding 2+2 with numeric values
. display 2+2
4

. *Adding 'stringvar'+"2" with string values
. display stringvar +"2"
apple2

. *Show the value of the 'stateq' variable (will show the value of the 1st record)
. display stateq
24

.
. *Describe will provide basic information on your data set and its variables:
. describe

```

Contains data from FINRA_01.dta

```

obs:      25,509
vars:      115
size:      7,907,790
6 Dec 2017 12:13

```

variable name	storage type	display format	value label	variable label
stringvar	str5	%9s		Example of a string variable
respid	long	%12.0g		Respondent ID
stateq	byte	%8.0g	STATEQ	state
censusreg	byte	%8.0g	CENSUSRE	census region
sloan	float	%9.0g		R currently has student loans
sl_concern	float	%9.0g		R concerned that s/he cannot pay back student loans
ed_lths	float	%9.0g		Education is less than High School
ed_hs	float	%9.0g		Education is High School or equivalent
ed_somcol	float	%9.0g		Education is some college

			Stata_Class_1	
ed_col1	float	%9.0g		Education is a college degree
ed_postcol1	float	%9.0g		Education is a post college degree
ed_al_somcol1	float	%9.0g		At least some college
ed_catvar	float	%25.0g	education	
ed_lteqhs	float	%9.0g		Education is High School, Equivalent or less
ed2_catvar	float	%19.0g	education2	
a_2029	float	%9.0g		Between ages of 20 and 29
a_3039	float	%9.0g		Between ages of 30 and 39
a_4049	float	%9.0g		Between ages of 40 and 49
a_5059	float	%9.0g		Between ages of 50 and 59
a_60plus	float	%9.0g		Age 60 or older
a_catvar	float	%12.0g	agegrp	
r_white	float	%9.0g		White, non-Hispanic
r_black	float	%9.0g		Black, non-Hispanic
r_hisp	float	%9.0g		Hispanic, any race
r_asian	float	%9.0g		Asian, non-Hispanic
r_other	float	%9.0g		Native American or Other, non-Hispanic
reg_ne	float	%9.0g		Northeast Census Region
reg_mw	float	%9.0g		Midwest Census Region
reg_south	float	%9.0g		South Census Region
reg_west	float	%9.0g		West Census Region
g_male	float	%9.0g		Male
g_female	float	%9.0g		Female
i_group1	byte	%8.0g		Less than \$15,000
i_group2	byte	%8.0g		At least \$15,000 but less than \$25,000
i_group3	byte	%8.0g		At least \$25,000 but less than \$35,000
i_group4	byte	%8.0g		At least \$35,000 but less than \$50,000
i_group5	byte	%8.0g		At least \$50,000 but less than \$75,000
i_group6	byte	%8.0g		At least \$75,000 but less than \$100,000
i_group7	byte	%8.0g		At least \$100,000
i_catvar	float	%11.0g	inc	Income Categories
i2_group1	float	%9.0g		Less than \$25,000
i2_group2	float	%9.0g		At least \$25,000 but less than \$50,000
i2_group3	float	%9.0g		At least \$50,000 but less than \$100,000
i2_group4	float	%9.0g		At least \$100,000
i2_catvar	float	%17.0g	inc2	All Income Categories
la_marr	float	%9.0g		Married
la_cohab	float	%9.0g		Cohabiting
la_nomarr	float	%9.0g		Never married, not cohabiting
la_separate	float	%9.0g		Separated, divorced, widowed; not cohabiting
la_catvar	float	%44.0g	larrg	
depchild	float	%9.0g		Number of dependent children
dc_nokid	float	%9.0g		One Dep Children
dc_1kid	float	%9.0g		
dc_2kid	float	%9.0g		Two Dep Children
dc_3kid	float	%9.0g		Three Dep Children
dc_4kid	float	%9.0g		4 or more Dep Children
dc_1or2kid	float	%9.0g		One or Two Dep Children
dc_3morekid	float	%9.0g		Three or more Dep Children
dc_anykid	float	%9.0g		Has Dep Children
dc_catvar	float	%22.0g	dc1	
dc2_catvar	float	%26.0g	dc2	
dc3_catvar	float	%16.0g	dc3	
emp_self	float	%9.0g		Respondent is Self Employed
emp_full	float	%9.0g		Respondent is Employed Full time
emp_part	float	%9.0g		Respondent is Employed Part time
emp_notLF	float	%9.0g		Respondent is Not in Labor Force
emp_sick	float	%9.0g		Respondent is Disabled/Sick
emp_unemp	float	%9.0g		Respondent is Unemployed
emp_catvar	float	%18.0g	empstat	
wgt_n2	double	%10.0g		National weight by age/gen, eth, ed, censusdiv
A3	byte	%8.0g	A3	What is your gender?
A3A	int	%8.0g	A3A	What is your age?
A4A	byte	%8.0g	A4A	Ethnicity
A5	byte	%8.0g	A5	What was the last year of education that you
completed?				
A6	byte	%8.0g	A6	What is your marital status?
A7	byte	%8.0g	A7	Which of the following describes your current

Stata_Class_1

Living arrangements?				
A11	byte	%8.0g	A11	* How many children do you have who are
financially dependent on you [or your spouse]				
A8	byte	%8.0g	A8	* What is your [household's] approximate annual
income, including wages, tips, and other income?				
A9	byte	%8.0g	LABA	Which of the following best describes your
current employment or work status?				
A10	byte	%8.0g	LABA	* Which of the following best describes your
[spouse's/partner's] current employment or work status?				
A21	byte	%8.0g	LABB	Are you a part-time student taking courses for
credit?				
A22	byte	%8.0g	A22	Which of the following best describes the
school you are attending?				
J1	byte	%8.0g	J1	* Overall, thinking of your assets, debts and
savings, how satisfied are you with your				
J4	byte	%8.0g	J4	* In a typical month, how difficult is it for you
to cover your expenses and pay a				
J5	byte	%8.0g	LABB	* Have you set aside emergency or rainy day funds
that would cover your expenses for				
J20	byte	%8.0g	J20	* How confident are you that you could come up
with \$2,000 if an unexpected need arises?				
B1	byte	%8.0g	LABB	Do you [Does your household] have a checking
account?				
B2	byte	%8.0g	LABB	* Do you [Does your household] have a savings
account, money market account, or CD?				
B4	byte	%8.0g	LABB	Do you [or your spouse/partner] overdraw your
checking account occasionally?				
B14	byte	%8.0g	LABB	* Not including retirement accounts, do you [does
your household] have any investments?				
C1	byte	%8.0g	LABB	* Do you [or your spouse/partner] have any
retirement plans through a current or past				
C4	byte	%8.0g	LABB	* Do you [or your spouse/partner] have any other
retirement accounts NOT through a				
C11	byte	%8.0g	LABB	* In the last 12 months, have you [or your
spouse/partner] taken a hardship withdrawal?				
D20_1	byte	%8.0g	LABB	* Over the past 12 months, did you [your
household] receive any of the following				
D20_5	byte	%8.0g	LABB	* Over the past 12 months, did you [your
household] receive any of the following				
D20_6	byte	%8.0g	LABB	* Over the past 12 months, did you [your
household] receive any of the following				
EA_1	byte	%8.0g	LABB	Do you [or your spouse/partner] currently own
any of the following? - Your home				
EA_2	byte	%8.0g	LABB	* Do you [or your spouse/partner] currently own
any of the following? - Other real				
E7	byte	%8.0g	LABB	Do you currently have any mortgages on your
home?				
E8	byte	%8.0g	LABB	Do you have any home equity loans?
E15	byte	%8.0g	E15	* How many times have you been late with your
mortgage payments in the last 2 years?				
E16	byte	%8.0g	LABB	* Have you been involved in a foreclosure process
on your home in the last 2 years?				
F2_1	byte	%8.0g	LABB	* In the past 12 months, which of the following
describes your experience with credit?				
F2_3	byte	%8.0g	LABB	* In the past 12 months, which of the following
describes your experience with credit?				
F2_4	byte	%8.0g	LABB	* In the past 12 months, which of the following
describes your experience with credit?				
F2_5	byte	%8.0g	LABB	* In the past 12 months, which of the following
describes your experience with credit?				
F2_6	byte	%8.0g	LABB	* In the past 12 months, which of the following
describes your experience with credit?				
G21	byte	%8.0g	LABB	Do you currently have any student loans?
G22	byte	%8.0g	LABB	Are you concerned that you might not be able to
pay off your student loans?				
G4	byte	%8.0g	LABB	Have you declared bankruptcy in the last two
years?				
G5_1	byte	%8.0g	LABD	* In the past 5 years, how many times have you... -
Taken out an auto title loan? Auto				
G5_2	byte	%8.0g	LABD	* In the past 5 years, how many times have you... -

Stata_Class_1

Taken out a short term 'payday'
G23 byte %8.0g
following statement? - I have too
probpop float %9.0g
coll no BA
probpop2 float %9.0g
no BA

LABEL

* How strongly do you agree or disagree with the
Problematic Population; Older than 25, Some
Problematic Population 2; Ages 25-40, Some coll
* indicated variables have notes

Sorted by:

. *Codebook provides more details on specific variables including missingness, range, example
values or distribution
. codebook stringvar respid stateq censusreg sloan sl_concern ed_lths

stringvar

Example of a string variable

type: string (str5)
unique values: 1 missing "": 0/25,509
tabulation: Freq. Value
25,509 "apple"

respid

Respondent ID

type: numeric (long)
range: [8,75001]
unique values: 25,509 units: 1
missing: 0/25,509
mean: 30698.5
std. dev: 21087.2
percentiles: 10% 25% 50% 75% 90%
4034 12808 27066 48130 62813

stateq

state

type: numeric (byte)
label: STATEQ
range: [1,51]
unique values: 51 units: 1
missing: 0/25,509
examples: 11 Georgia
21 Maryland
31 New Jersey
41 South Carolina

censusreg

census region
Page 4

Stata_Class_1

```

type:   numeric (byte)
label:  CENSUSRE

range:   [1, 4]
unique values: 4

units: 1
missing: 0/25,509

tabulation:  Freq.   Numeric   Label
              4,501       1   Northeast
              6,004       2   Midwest
              8,501       3   South
              6,503       4   West

```

```

sloan
R currently has student loans

```

```

type:   numeric (float)

range:   [0, 1]
unique values: 2

units: 1
missing: 319/25,509

tabulation:  Freq.   Value
              20,049   0
              5,141   1
              319     .

```

```

sl_concern
R concerned that s/he cannot pay back student loans

```

```

type:   numeric (float)

range:   [0, 1]
unique values: 2

units: 1
missing: 20,619/25,509

tabulation:  Freq.   Value
              2,145   0
              2,745   1
              20,619   .

```

```

ed_lths
Education is less than High School

```

```

type:   numeric (float)

range:   [0, 1]
unique values: 2

units: 1
missing: 0/25,509

tabulation:  Freq.   Value
              23,606   0
              1,903   1

```

```

. *List will print out values for the variables and records/observations specified
. list stringvar respid stateq censusreg sloan sl_concern ed_lths in 1/10

```

```

+-----+-----+-----+-----+-----+-----+
| string-r  respid    stateq  census-g  sloan   sl_con-n  ed_lths |
+-----+-----+-----+-----+-----+-----+

```

Stata_Class_1

1.	apple	8	Minnesota	Midwest	0	.	0
2.	apple	10	Florida	South	0	.	0
3.	apple	11	Michigan	Midwest	0	.	0
4.	apple	12	Illinois	Midwest	0	.	0
5.	apple	13	Texas	South	1	1	0
6.	apple	14	Mississippi	South	0	.	0
7.	apple	15	New Jersey	Northeast	0	.	0
8.	apple	16	Massachusetts	Northeast	1	0	0
9.	apple	17	California	West	0	.	0
10.	apple	18	Arkansas	South	0	.	0

Descriptive statistics

*You may want to examine variables of interest for your analysis more closely
*or include descriptive statistics for them in your study

*Use summarize for discrete or continuous numeric variables,
*tabulate for ordinal or categorical variables,
*and either for dummies/binary/indicators:

*summarize the age variable
summarize A3A

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	25,509	47.00588	16.07551	18	101

*or we can summarize more than one variable at a time
*(age and whether observation is white):

summarize A3A r_white

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	25,509	47.00588	16.07551	18	101
r_white	25,509	.7336626	.4420514	0	1

*tabulate education category variable
tabulate ed_catvar

ed_catvar	Freq.	Percent	Cum.
Less than High School	1,903	7.46	7.46
High School or equivalent	6,561	25.72	33.18
Some College	8,419	33.00	66.18
College	5,343	20.95	87.13
Postgraduate Degree	3,283	12.87	100.00
Total	25,509	100.00	

*tabulate student loan reciprocity variable with the missing option (, missing)
*at the end of the command after a comma to show any missing values
*(coded as . for numeric data and "" for string data)

tabulate sloan, missing

R currently has student loans	Freq.	Percent	Cum.
0	20,049	78.60	78.60
1	5,141	20.15	98.75
.	319	1.25	100.00
Total	25,509	100.00	

*You usually don't have to type out the whole expression,
*Stata will know what you mean if you abbreviate as long as

Stata_Class_1

. *there is no ambiguity with other commands, variables, options, etc.
 . tab sloan, m

R currently has student loans	Freq.	Percent	Cum.
0	20,049	78.60	78.60
1	5,141	20.15	98.75
.	319	1.25	100.00
Total	25,509	100.00	

. *Another option: see a labeled numeric variable without its value labels
 . *(i.e. see the underlying numeric values)
 . tab B1

Do you [Does your household] have a checking account?	Freq.	Percent	Cum.
Yes	22,948	89.96	89.96
No	2,151	8.43	98.39
Don't know	107	0.42	98.81
Prefer not to say	303	1.19	100.00
Total	25,509	100.00	

. tab B1, nolabel

Do you [Does your household] have a checking account?	Freq.	Percent	Cum.
1	22,948	89.96	89.96
2	2,151	8.43	98.39
98	107	0.42	98.81
99	303	1.19	100.00
Total	25,509	100.00	

. *Also, use tabulate with two categorical variables to show their crosstab:
 . tab ed_catvar sloan, missing

ed_catvar	R currently has student loans			Total
	0	1	.	
Less than High School	1,765	94	44	1,903
High School or equivalent	5,849	612	100	6,561
Some College	6,311	2,013	95	8,419
College	3,739	1,551	53	5,343
Postgraduate Degree	2,385	871	27	3,283
Total	20,049	5,141	319	25,509

. tab ed_catvar sloan, column //the column option reports the % within each row that are in each column category (%s in columns sum to 100%)

Key
frequency
column percentage

Stata_Class_1			
ed_catvar	student loans		Total
	0	1	
Less than High School	1,765 8.80	94 1.83	1,859 7.38
High School or equiva	5,849 29.17	612 11.90	6,461 25.65
Some College	6,311 31.48	2,013 39.16	8,324 33.04
College	3,739 18.65	1,551 30.17	5,290 21.00
Postgraduate Degree	2,385 11.90	871 16.94	3,256 12.93
Total	20,049 100.00	5,141 100.00	25,190 100.00

. tab ed_catvar Sloan, row //the row option reports the % within each column that are in each row category (%s in rows sum to 100%)

Key
frequency
row percentage

ed_catvar	R currently has student loans		Total
	0	1	
Less than High School	1,765 94.94	94 5.06	1,859 100.00
High School or equiva	5,849 90.53	612 9.47	6,461 100.00
Some College	6,311 75.82	2,013 24.18	8,324 100.00
College	3,739 70.68	1,551 29.32	5,290 100.00
Postgraduate Degree	2,385 73.25	871 26.75	3,256 100.00
Total	20,049 79.59	5,141 20.41	25,190 100.00

. tab ed_catvar Sloan, cell //the cell option reports the % in the entire table total that are in each cell (%s in cells sum to 100%)

Key
frequency
cell percentage

ed_catvar	R currently has student loans		Total
	0	1	
Less than High School	1,765	94	1,859

		Stata	Class_1
	7.01	0.37	7.38
High School or equivalent	5,849 23.22	612 2.43	6,461 25.65
Some College	6,311 25.05	2,013 7.99	8,324 33.04
College	3,739 14.84	1,551 6.16	5,290 21.00
Postgraduate Degree	2,385 9.47	871 3.46	3,256 12.93
Total	20,049 79.59	5,141 20.41	25,190 100.00

```
. *another way to look at your data - graphs
. *Histogram showing the distribution of age:
. histogram A3A
(bin=44, start=18, width=1.8863636)
```

```
.
. ***Subsetting your data***
. *IF and IN statements allow you to operate on subsets of your data
. *IN statements define subsets based on records' index (observation) number
. *IF statements define subsets based on conditional statements
.   *& is AND
.   *| is OR
.   *Use == to compare the equality of two values
.   *!= is not equal to (! is NOT in general)
.   * > is greater than
.   * < is less than
.   * >= is greater than or equal to (= must come after < or >, not before or it will
not work, i.e. >= is CORRECT; => is INCORRECT)
.   * <= is less than or equal to
. *Summarize age for just observations less than age 50:
. summarize A3A if A3A < 50
```

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	13,507	34.19094	9.127357	18	49

```
. *Tabulate education for just observations less than age 50:
. tabulate ed_catvar if A3A < 50
```

ed_catvar	Freq.	Percent	Cum.
Less than High School	1,313	9.72	9.72
High School or equivalent	3,350	24.80	34.52
Some College	4,307	31.89	66.41
College	3,091	22.88	89.29
Postgraduate Degree	1,446	10.71	100.00
Total	13,507	100.00	

```
. *or just those whose age is equal to 50:
. tabulate ed_catvar if A3A == 50
```

ed_catvar	Freq.	Percent	Cum.
Less than High School	42	7.53	7.53
High School or equivalent	163	29.21	36.74
Some College	166	29.75	66.49
College	130	23.30	89.78
Postgraduate Degree	57	10.22	100.00

Total | 558 Stata_Class_1 100.00

. *We already used an 'in' expression with 'list' above to show the values of
 . *some of our variables for the first ten observations:
 . list stringvar respid stateq censusreg sloan sl_concern ed_lths in 1/10

	string-r	respid	stateq	census-g	sloan	sl_con-n	ed_lths
1.	apple	8	Minnesot	Midwest	0	.	0
2.	apple	10	Florida	South	0	.	0
3.	apple	11	Michigan	Midwest	0	.	0
4.	apple	12	Illinois	Midwest	0	.	0
5.	apple	13	Texas	South	1	1	0
6.	apple	14	Mississi	South	0	.	0
7.	apple	15	New Jers	Northeas	0	.	0
8.	apple	16	Massachu	Northeas	1	0	0
9.	apple	17	Californ	West	0	.	0
10.	apple	18	Arkansas	South	0	.	0

.
 . ***Creating new variables and updating existing variables***
 . *Be mindful of missing (" " if string or . if numeric) values as well as special values
 codes
 . *Special values are often negative or high values like 9998, 9999, etc.
 . *and can indicate "don't know," "refused," etc. in survey data.
 . *always inspect your data initially via tab, summ, etc. to look for these and deal with
 them appropriately
 . *BEFORE constructing variables or starting your actual analysis
 .
 . *binary (0/1) age variable
 . *The 'generate' creates a new variable with the name specified and set to the value after
 the equals sign
 . generate a1824 = 0
 .
 . *The 'replace' command updates the values of an existing variable;
 . *often you'll combine it with if statements to change the values of just a subset
 of observations
 . replace a1824 = 1 if A3A >= 18 & A3A < 25
 (2,581 real changes made)

. *Look at your variable after you make it
 . tab a1824, m

a1824	Freq.	Percent	Cum.
0	22,928	89.88	89.88
1	2,581	10.12	100.00
Total	25,509	100.00	

. *And compare it to the original
 . summarize A3A if a1824 == 0

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	22,928	49.91787	14.25653	25	101

. summarize A3A if a1824 == 1

Variable	Obs	Mean	Std. Dev.	Min	Max
A3A	2,581	21.13754	2.006788	18	24

. *Finally, label your variable

```

. label variable a1824 "age between 18 and 24"

. *You can also create categorical variables
. gen age_cat = .
(25,509 missing values generated)

. replace age_cat = 1 if A3A > 17 & A3A < 25
(2,581 real changes made)

. replace age_cat = 2 if A3A > 24 & A3A < 61
(16,843 real changes made)

. replace age_cat = 3 if A3A > 60
(6,085 real changes made)

. *Label your variable
. label variable age_cat "age categories"

. *Label the values of your variable
. label define age 1 "18-24" 2 "25-60" 3 "more than 60"

. label values age_cat age

. tab age_cat, m

```

age categories	Freq.	Percent	Cum.
18-24	2,581	10.12	10.12
25-60	16,843	66.03	76.15
more than 60	6,085	23.85	100.00
Total	25,509	100.00	

```

. *Confirm that you created variable correctly by crosstabbing vs. original variable(s)
. tab A3A age_cat, m

```

What is your age?	age categories			Total
	18-24	25-60	more than	
18	374	0	0	374
19	294	0	0	294
20	322	0	0	322
21	395	0	0	395
22	408	0	0	408
23	385	0	0	385
24	403	0	0	403
25	0	408	0	408
26	0	361	0	361
27	0	406	0	406
28	0	380	0	380
29	0	435	0	435
30	0	502	0	502
31	0	450	0	450
32	0	496	0	496
33	0	433	0	433
34	0	413	0	413
35	0	469	0	469
36	0	412	0	412
37	0	430	0	430
38	0	383	0	383
39	0	390	0	390
40	0	449	0	449
41	0	415	0	415
42	0	457	0	457
43	0	464	0	464
44	0	419	0	419
45	0	475	0	475
46	0	429	0	429

		Stata_Class_1		
47	0	466	0	466
48	0	460	0	460
49	0	524	0	524
50	0	558	0	558
51	0	555	0	555
52	0	597	0	597
53	0	571	0	571
54	0	582	0	582
55	0	538	0	538
56	0	536	0	536
57	0	481	0	481
58	0	484	0	484
59	0	523	0	523
60	0	492	0	492
61	0	0	467	467
62	0	0	460	460
63	0	0	418	418
64	0	0	449	449
65	0	0	738	738
66	0	0	502	502
67	0	0	424	424
68	0	0	396	396
69	0	0	367	367
70	0	0	316	316
71	0	0	232	232
72	0	0	228	228
73	0	0	199	199
74	0	0	166	166
75	0	0	139	139
76	0	0	114	114
77	0	0	80	80
78	0	0	60	60
79	0	0	58	58
80	0	0	57	57
81	0	0	42	42
82	0	0	39	39
83	0	0	27	27
84	0	0	26	26
85	0	0	27	27
86	0	0	13	13
87	0	0	12	12
88	0	0	7	7
89	0	0	4	4
90	0	0	7	7
91	0	0	2	2
92	0	0	3	3
93	0	0	2	2
94	0	0	2	2
99	0	0	1	1
101 or older	0	0	1	1
Total	2,581	16,843	6,085	25,509

```

. **Other kinds of variables to create
. *Scaling/transformations
. *Create an age in months variable based on the age in years:
. gen age_in_months = A3A*12

```

```

. sum age_in_months A3A

```

Variable	Obs	Mean	Std. Dev.	Min	Max
age_in_mon~s	25,509	564.0706	192.9061	216	1212
A3A	25,509	47.00588	16.07551	18	101

```

. *Create age-squared
. gen age_squared = A3A*A3A

```

Stata_Class_1

```
. sum age_squared A3A
```

Variable	Obs	Mean	Std. Dev.	Min	Max
age_squared	25,509	2467.965	1533.861	324	10201
A3A	25,509	47.00588	16.07551	18	101

```
. *Create logged-age
. gen age_logged = ln(A3A)
```

```
. sum age_logged A3A
```

Variable	Obs	Mean	Std. Dev.	Min	Max
age_logged	25,509	3.783343	.3809941	2.890372	4.61512
A3A	25,509	47.00588	16.07551	18	101

```
. *Create a new variable based on the value of multiple other variables:
. *Create a new dummy variable based on a few other dummies rather than just one
. *B1 = checking acct; B2 = savings acct
. gen bankacct = .
(25,509 missing values generated)
```

```
. replace bankacct = 1 if B1 == 1
(22,948 real changes made)
```

```
. replace bankacct = 1 if B2 == 1
(598 real changes made)
```

```
. replace bankacct = 0 if B1 == 2 & B2 == 2
(1,558 real changes made)
```

```
. *could also have used an OR statement instead of an AND statement - depends on how you want
to define things
. replace bankacct = 0 if (B1 == 2 | B2 == 2) & bankacct == .
(85 real changes made)
```

```
. *Confirm that you created variable correctly by crosstabbing vs. original variable(s)
. tab bankacct B1 if B2 != 1, m
```

bankacct	Do you [Does your household] have a checking account?				Total
	Yes	No	Don't know	Prefer no	
0	0	1,578	21	44	1,643
1	4,522	0	0	0	4,522
.	0	0	74	246	320
Total	4,522	1,578	95	290	6,485

```
. tab bankacct B2 if B1 != 1, m
```

bankacct	Do you [Does your household] have a savings account, money market account, or CD				Total
	Yes	No	Don't know	Prefer no	
0	0	1,623	16	4	1,643
1	598	0	0	0	598
.	0	0	74	246	320
Total	598	1,623	90	250	2,561

```
. *Interaction variable: multiple the values of the female indicator and age
. *to create a variable that captures females age but is 0 for males
. gen female_age = g_female*A3A
```

Stata_Class_1

. *Compare the original and new variable for females and non-females:
 . sum female_age A3A if g_female == 1

Variable	Obs	Mean	Std. Dev.	Min	Max
female_age	14,127	46.11814	16.08758	18	101
A3A	14,127	46.11814	16.08758	18	101

. sum female_age A3A if g_female == 0

Variable	Obs	Mean	Std. Dev.	Min	Max
female_age	11,382	0	0	0	0
A3A	11,382	48.10771	15.99282	18	99

.
 .
 . ***Stata help: once you can use a Stata help file, you should be able to figure out almost any command!
 . help summarize

. *Basic syntax of a Stata command:
 . *commandname expression if/in expression , options
 . *The 1st expression can contain variable names, assignment clauses, subcommands, etc. and depend on the particular command
 . *The if/in statement is followed by a 2nd expression defining the subset of the data set you want the command to work on
 . *Options always follow a single comma and are also command specific/dependent.
 . *Reference the help file for more on options, syntax, etc. for specific commands
 .
 . *More on syntax is available here: <http://www.stata.com/manuals13/gsw10.pdf>
 .
 . *A list of basic Stata commands is available at:
 . * <http://www.stata.com/manuals13/u27.pdf>
 . * and
 . * https://people.ucsc.edu/~aspearot/Econ113W13%20/basic_tutorial_stata.pdf
 .

. *always save with a new name, do not overwrite your data.
 . save "FINRA_02.dta", replace
 file FINRA_02.dta saved

. log close
 name: <unnamed>
 log: D:\Users\hhassani\Desktop\Stata_Class_1.log
 log type: text
 closed on: 29 Jun 2018, 09:59:19