Handling Large-Scale Unit Level Data Using STATA
Professor Pratap C. Mohanty
Department of Humanities and Social Sciences,
Indian Institute of Technology Roorkee
Lecture 15
Managing Data in Stata – III

Once again friends we are explaining the NPLEL module on Handling Large-Scale Data Using Stata. We are on the lecture explaining data or managing data in Stata number third. We are still continuing with the very basic commands of Stata with the original data. In the last lecture we discussed many important aspects of string, destringing of the variables and replace, recode and that is also important, especially when your variable is numeric or in string. So accordingly we suggested you go by recode for numeric variable and replace for string variable, then further operations the string to destring those we have already suggested earlier.

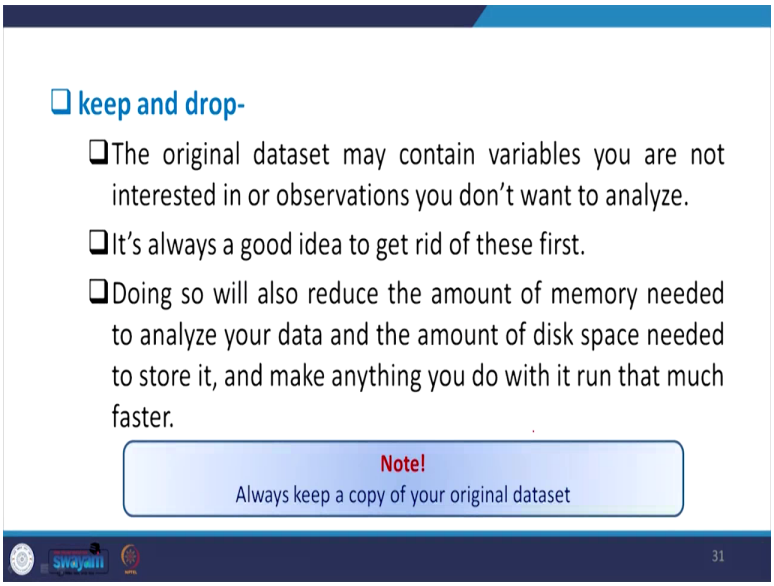
Now it is important to understand like in a large dataset, usually so many information is cropped up together. And so many information may not be important or useful for an individual researcher. Individual researcher may extract a very little portion out of it for his or her use or for publication or for writing a paper. So, it is very important for the researcher to understand what variable to keep and what variable to drop.

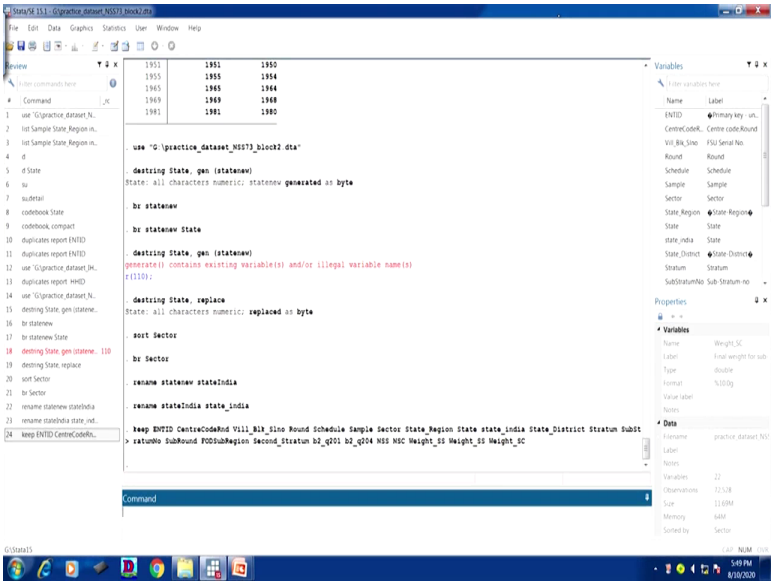
Though it seems that keep is opposite of drop and drop is obviously opposite of keep, but it is not obvious in Stata. Why am I saying? Keeping variable that does not mean it is dropping or dropping variable that does not mean it is keeping. Why? It is very important to note. Like I wanted to keep some variable. It will keep and automatically drop others that does not mean it is dropping, it drops others, but keeping, your focus here is to keep.

What do you mean by that? Your focus is to keep some variables, but if alternatively, if you just use drop in that case, drop other variables and keep that is not that so easy. If so many variables are to be dropped, so how many you will enter one by one? So in that case, keep is important. So just opposite, though in rough sense or in true sense even it is opposite, but in command the uses are different. You want keep sometimes and you want drop sometimes. Alternatively, it is not suggested. You please mark the difference, while I will tell you once again.

Like you have total 5,000 variables, you wanted to keep only 50 out of it. In that case, keep is very important. And reverse, again I am trying to use the reverse word, you want to drop a specific variable not the entire. So you just drop by taking the name of those variable and it will automatically keep. The opposite is true, but you have to appropriately choose the right command. Just keep is not enough. Just drop is not enough.

(Refer Slide Time: 04:56)





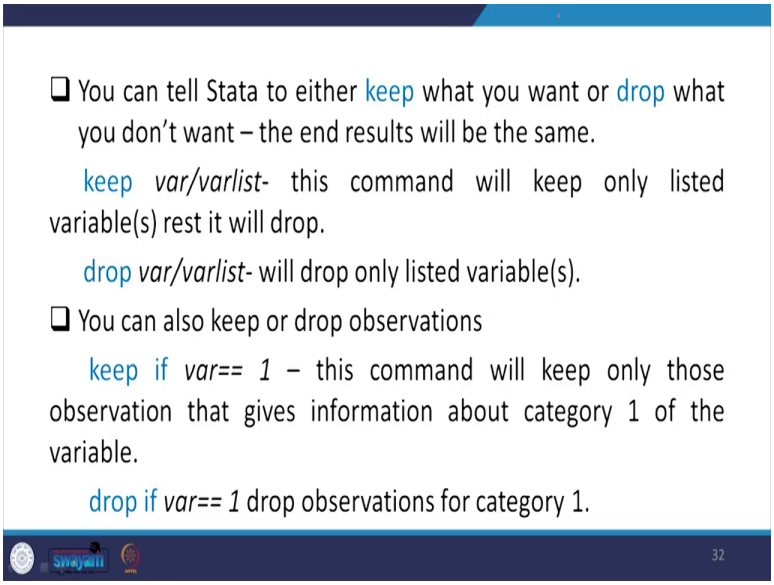
Let us understand the integrity of it further. The original dataset may contain variables you are not interested or may not want to analyze. It is always good to get rid of those at the first hand.

You need to get rid of those non-necessary variables. Doing so, we will also reduce the amount of memory or the byte space so that your operation will be faster and smooth and it will not hang also in between. So even disk space requirement will be going to be less when you have limited your data as per your requirement and make anything you do with it run that much faster.

Please take a note that, always keep a copy of your original dataset while you operate these two command, because if you dropped already, so those variables are not going to retrieved. So if you have already kept an additional file, by wrong attempts you have dropped, but you want those variables further that maybe very problematic, you cannot able to retrieve that. So it is always suggested that please keep a copy of it, and later on you can delete.

I can show you, type keep it is not so important, keep then these number of variables. Then if I can show you for your understanding, now it will drop. For example, here keep, if I just keep here like this, this, this, I have to keep more, because I will be reusing those later. So some numeric data I want. So if I do that, it will keep those variables where I have entered the keep command and rest of the variables are dropped. You can mark the difference what I have kept. Only those variables are visible. Similarly, if you drop, it will drop only those specific variable with the name, rest will be kept.

(Refer Slide Time: 07:34)



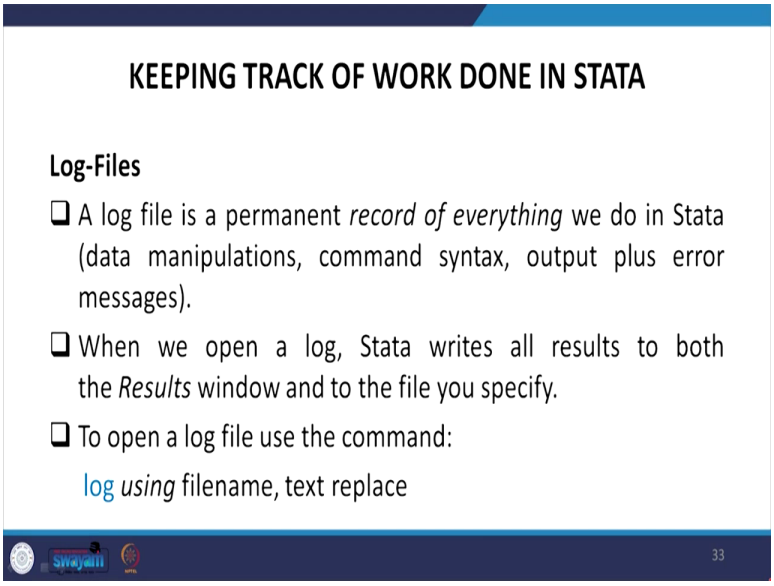
Let me proceed further. You can tell Stata to either keep what you want to drop, what you do not want and the end results will be the same. Same in the sense, keep, it will drop others; drop, it will keep others. Keep variable name, this command will keep only listed variables that I told you. Now it is interesting to note that you can also keep or drop observations, not just variables. Like keep if variable double equal to 1, like within the variable when there are different categories, one, two, three, four are there, you only wanted to continue with one, others are not going to be useful for you.

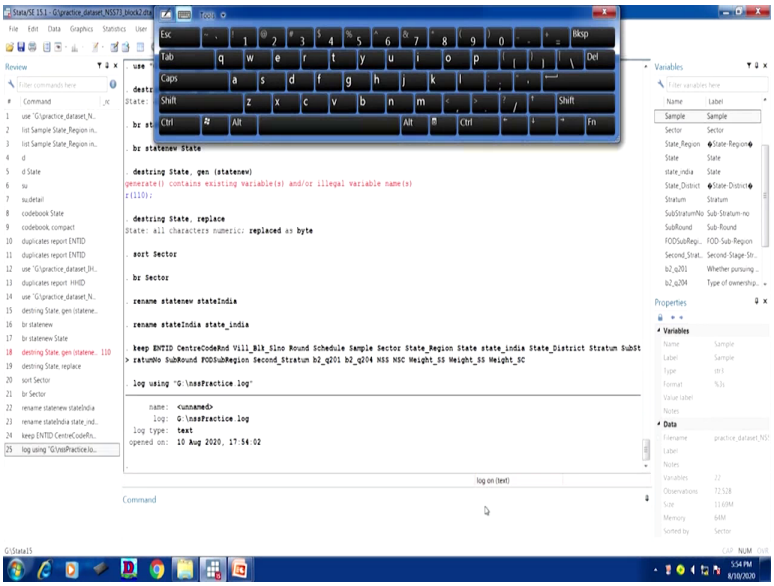
For example, in our dataset we have a variable called gender, male, female, then transgender. If I want to analyze for female entrepreneurship, this is the data on entrepreneurship. I wanted to analyze for women entrepreneurship, so why should I continue with other two categories? So in that case, I will simply give the command, keep if that variable gender, if it is there, I think we have already filtered, let me check.

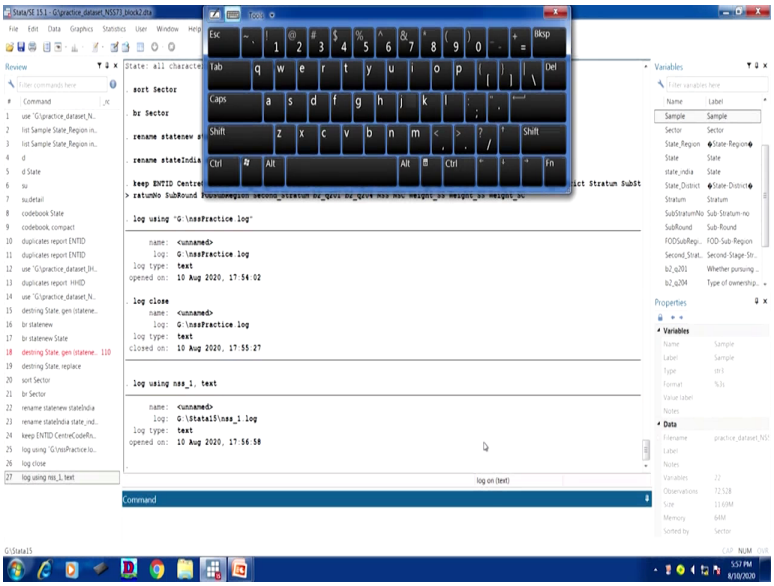
So we have already filtered for gender. We have taken one by entering the keep if format. So, in general, if you enter double equal to one or even any variable, you wanted to go by that particular category. I have already given you the example. So it will continue with that gender or that particular category.

Similarly, drop if one is not important, but other categories are important. There are education for example five, six entries are there, you only wanted to drop three as the category, others will be there. So only double equal to three if you make it drop, if education attainment (that is the variable name) double equal to three, it will only drop that three, others will be there.

(Refer Slide Time: 10:07)







Let me proceed further. So let us have a track of the work already done in Stata. We already mentioned a bit on log files. We are trying to have a track of the log files once again or track of our work through the log files. As I told you from the beginning that you please start with a log file, like here, let me open this then like this, what I will do, I will open this first.

Log file is very important for us to work. You simply click file then log begin, what it will give, you can save with a name. Any name you just type and the option you can pick-up from the save as type, if you go by the by default option it will give you the same window and the same type of result which is visible here.

If you go by the .log file, it gives in text format. I already guided you that text format. Let me do it for you with a name, let me type it. For example, it is NSS then I am deliberately doing capital letter, NSS then practice. I have followed a camel format, camel renaming format, nssPractice. If you simply save, it is there, already saved in our file. nssPractice is the file here.

If I just open it in our, it is already opened. look at here mark this carefully I am putting my cursor here. Log on is open, log on in text. It is visible. When you exit or there are various ways of understanding the log file. Either, what I have done, I opened through the pointer method. I clicked and open the log file. Another option for you is log using file name. Log, you just simply type log using file name, then text replace. For example, I can do it right now. Let me just come back. I will come back and clarify to you.

Log using, let me do another one. Log using file name, so here log close, the earlier replace, I will go by the replace also that is another option. That is fine. Now log, we have already closed the earlier log because that might have created problem. So I will go by log using then file name, earlier we have given a file name.

If I give another file name like nss, let me operate through underscore path, nss_1, any name, 1, 2, nss maybe 1. If I do by this then like here, one option smcl is important. Either you enter at smcl, if you do not enter text, it will by default take smcl. If you take text it will take you the option called log version, log on, the way I have shown you, then text replace, any replace is the word you do if there is an existing log file. If no existing log file is there, no need to go by, since we have already closed that log file, so need to take replace, only text.

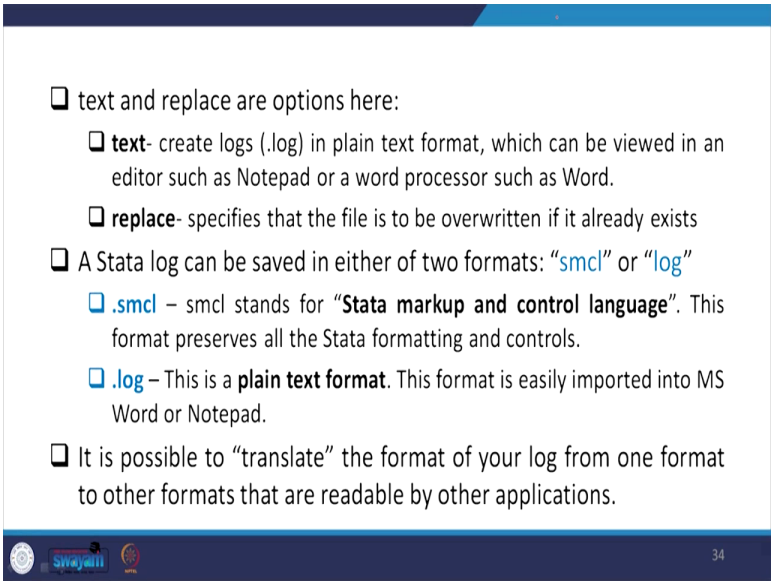
So what is opened, look at this log on text is opened, this is your log file. Since this is already on I will show you how this is, all our command in between is going to be saved except the graph. Graph usually opens in different window; we need to manually save those graphs. Let me go by our understanding. For your clarity, I am reading through if there is any important point missing you may mark it carefully.

So a log file is a permanent record of everything we do in Stata that is basically manipulating data, data syntax, output plus error messages. Error message, I told you, this generally pops up in the window, in the same window and with red color. And those errors, we will tell you how to handle those errors while to save those logs. But the log file saves everything, your errors as well as you write command, or correct command syntax also.

Why errors are important? Because in future or in another point of time if you wanted to check what kind of steps I have gone through, your errors or steps which were erroneous, helps the reader or the researcher very quickly to understand the right command or the right approach. So errors are also important to save. So the log file saves the errors as well.

When we open a log Stata writes all the results to both the results window and to the file you specify. To open a log, use the command we just already made, like replace is there if you go by overwriting another one you need to replace the first one and then you go for a new one.

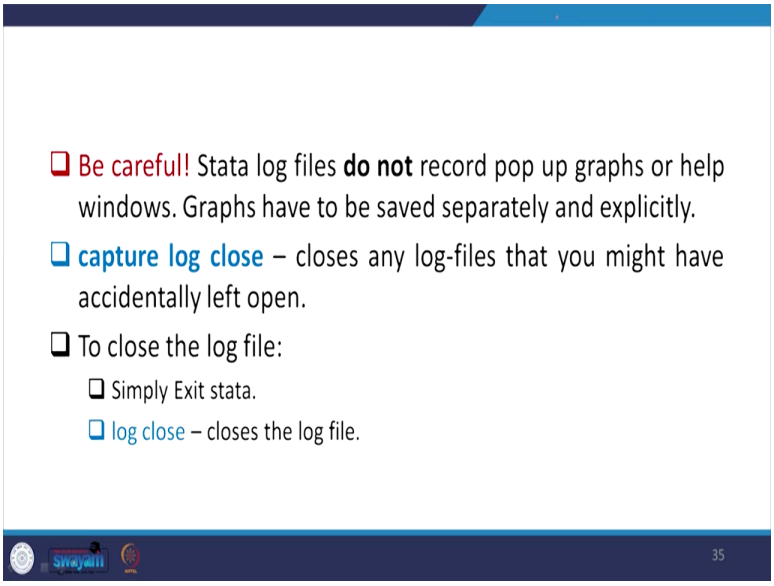
(Refer Slide Time: 17:22)

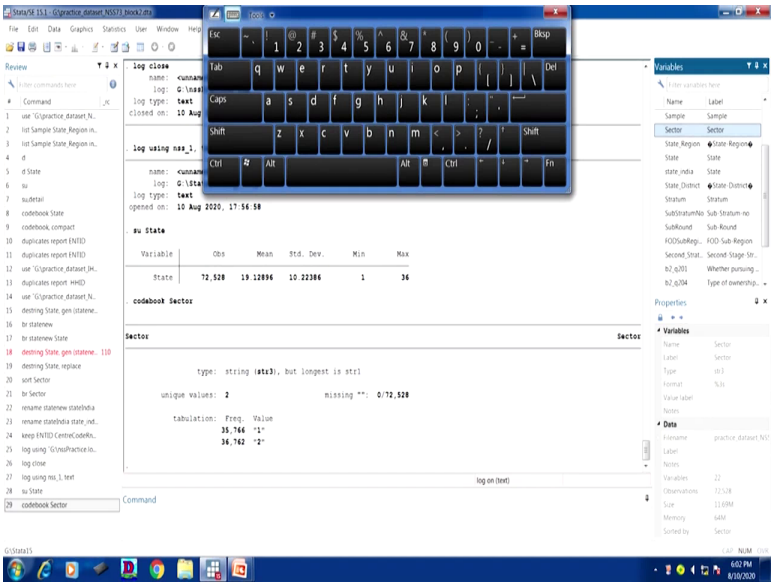


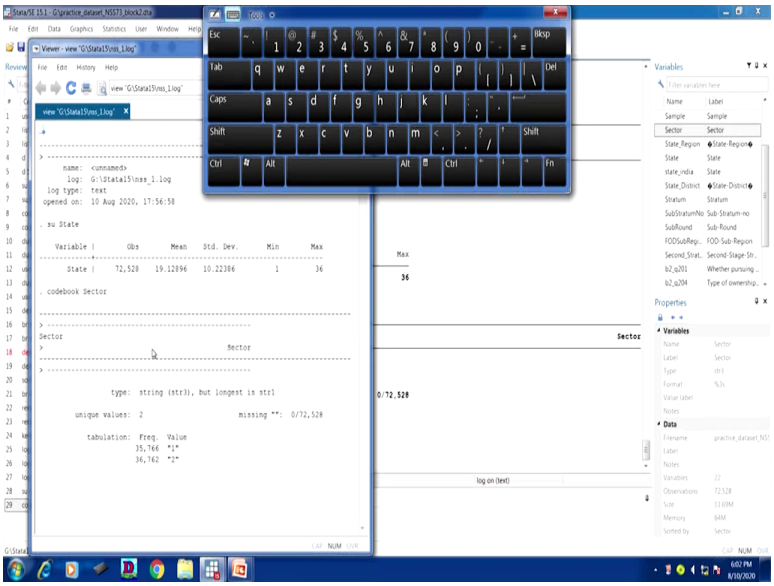
Text and replace are options here, like as I already told you, text, if you have given text or in the click based method .log format, otherwise in the syntax based command if you have entered text then you will be saved with a plain text format which can be viewed in an editor such as notepad or a word processor such as MS Word. So replace specifies that the file is to be overwritten if it already exists.

A Stata log can be saved in either of two formats that I told you smcl and .log. Let me proceed I already clarified. It is possible to translate the format to your log from one format to other format that are readable by other applications also.

(Refer Slide Time: 18:19)







So be careful that Stata log files do not record pop-up graphs, as I just mentioned. If you have any command on graphs, window comes separately and the log files do not save the graphs, you need to manually save your graph separately. And this do not record pop-up graphs or help windows, graphs have to be saved separately or explicitly.

So capture log close is also important. This closes any log files that you might have accidentally left open, so you need to go by this command. By any accidental reason if you have opened the log file that you need to close, you please go by capture log close it will automatically close the in between log files, otherwise simply exit the Stata, exit Stata, it will close or log close will close the log file.

So we have already saved a log file also. At the end, how other commands are in between saved. Let me just operate, I will show you how those have been saved automatically. Like state if I just enter it has given you summary statistics or codebook which we did earlier for our analysis. Codebook of a particular variable, let it be sector, if I simply enter it saved with other details. Let me open the log file, log file which we already created and since log is on, log text is on, it is not closed yet I am just highlighting here, let me open that.

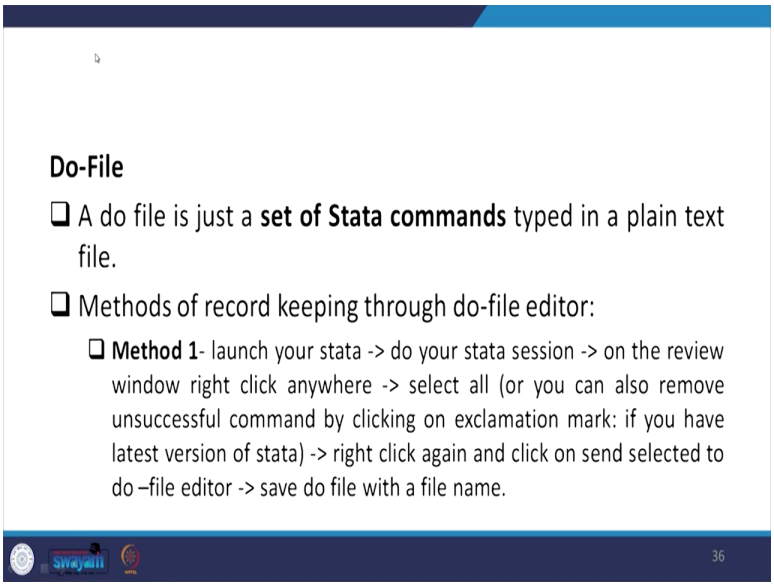
So the log file we created is nssPractice, what did I see nss_1. It has by default gets saved, it has saved here within Stata 15. Since Stata 15 opened so it by default took us to, consider the space within the Stata folder. And these are the operations we made just codebook we, summary

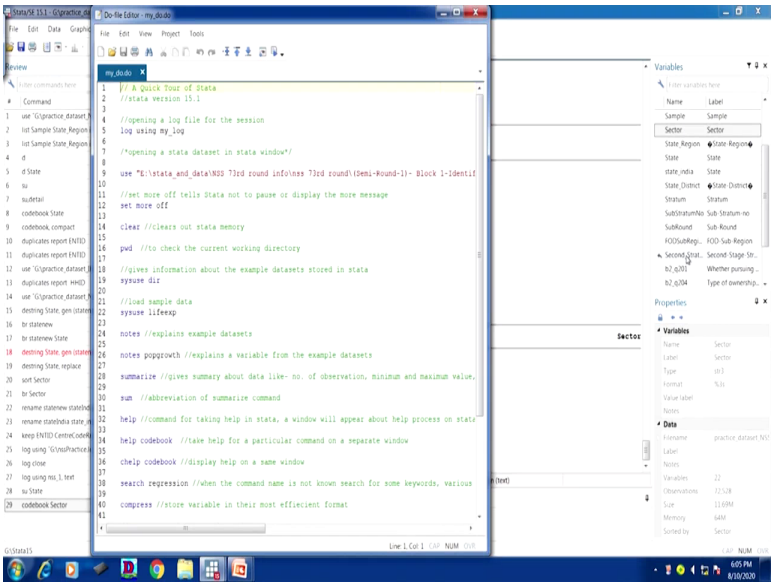
statistics we derived and also codebook we entered. So it has already saved for us and let us keep it open till it is continuing.

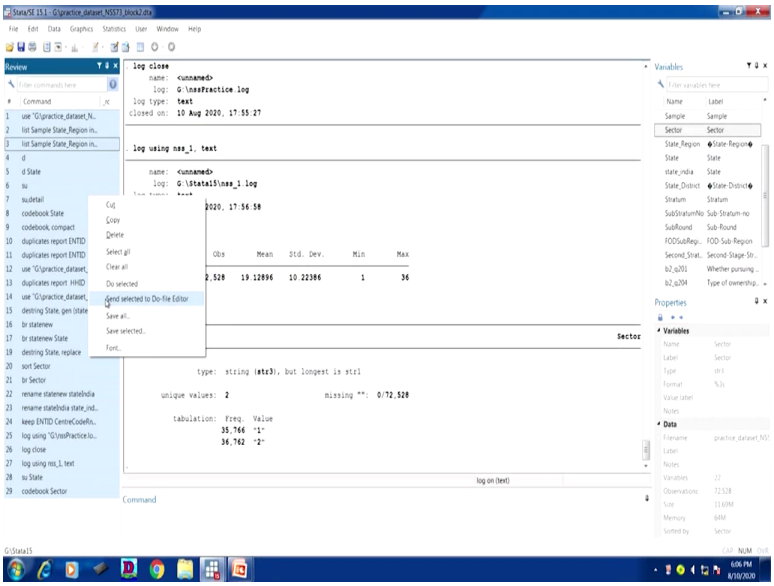
Another aspect as I told you like here in the left hand side review is there. I am going to use this, I want to use, these all are commands, experiment we made in between, we need to save those commands. So do-file is going to be useful. There are two approaches to do that: to go by the commands that is called do-file we need to save those commands. Log files also saves commands, but results are also automatically saves.

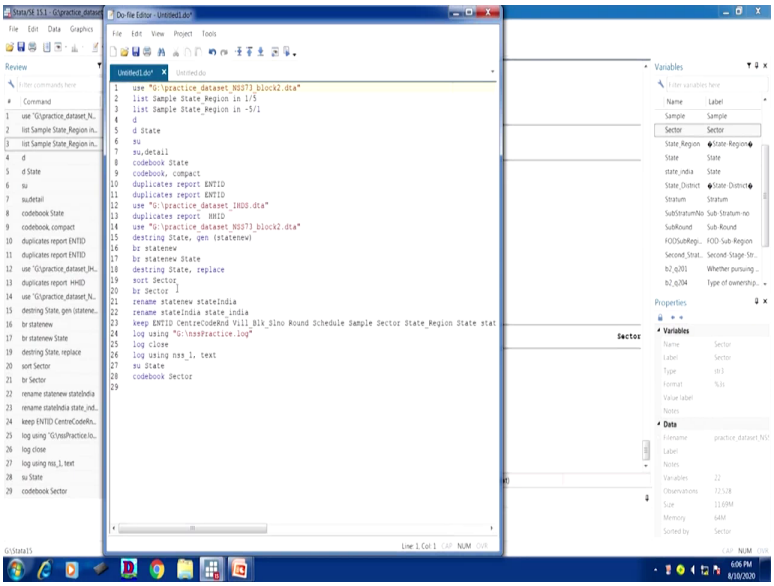
The number of pages are very huge but in case of do-file, you can save the commands in a very compact manner and that too you can also execute or run from the do-file. If you run from there, execute from there, you can derive the result but from the log file, you cannot run, you need to manually copy some of the commands and run it. So do-file is very important.

(Refer Slide Time: 22:14)




Do-file Editor - my_do.do
// A Quick Tour of Stata
//stata version 15.1
//opening a log file for the session
log using my_log
/*opening a stata dataset in stata window*/
use "E:\stata_and_data\NSS 73rd round info\nss 73rd round\(Semi-Round-1)- Block 1-Identif
//set more off tells Stata not to pause or display the more message
set more off
clear //clears out stata memory
pwd //to check the current working directory
//gives information about the example datasets stored in stata
sysuse dir
//load sample data
sysuse lifeexp
notes //explains example datasets
notes popgrowth //explains a variable from the example datasets
summarize //gives summary about data like- no. of observation, minimum and maximum value,
sum //abbreviation of summarize command
help //command for taking help in stata, a window will appear about help process on stata
help codebook //take help for a particular command on a separate window
chelp codebook //display help on a same window
search regression //when the command name is not known search for some keywords, various
compress //store variable in their most effiecient format
Line 1, Col 1
Variables
Properties
6:05 PM
8/10/2020


log close
name: <unnamed>
log: G:\nssPractice.log
log type: text
closed on: 10 Aug 2020, 17:55:27
. log using nss_1, text
name: <unnamed>
log: G:\Stata15\nss_1.log
2020, 17:56:58
Obs Mean Std. Dev. Min Max
2,528 19.12896 10.22386 1 36
Sector
type: string (str3), but longest is str1
unique values: 2 missing "": 0/72,528
tabulation: Freq. Value
35,766 "1"
36,762 "2"
Cut
Copy
Delete
Select all
Clear all
Do selected
Send selected to Do-file Editor
Save all...
Save selected...
Font...
log on (text)
Command
Variables
Properties
6:06 PM
8/10/2020



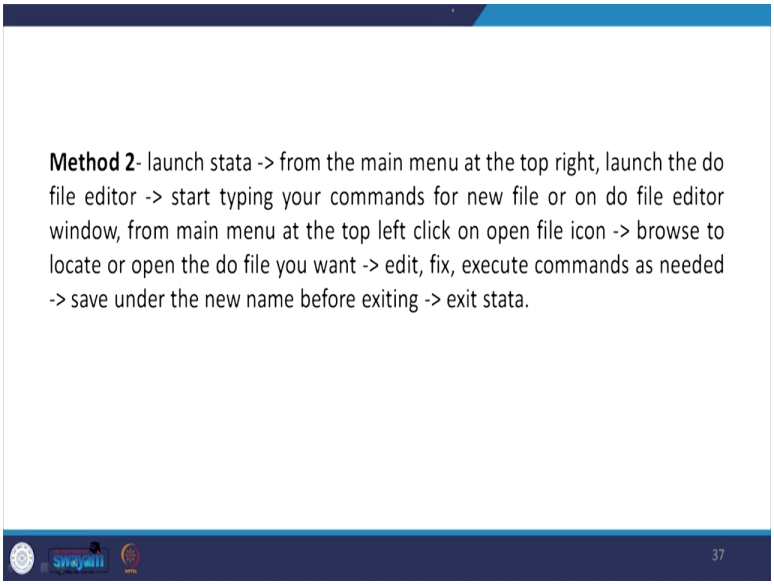
Do-file is just a set of Stata commands typed in a plain text file. There are two approaches to deal with the do-file. One is launch your Stata, like from the start button, as I told you, and on the review window, right click anywhere, select all, you can also remove unsuccessful command. Let me just start with this like you have to first click open a do-file. For your knowledge, we have already defined a do-file. Let me first open our do-file. It is here, my_do file. Now it has already been opened.

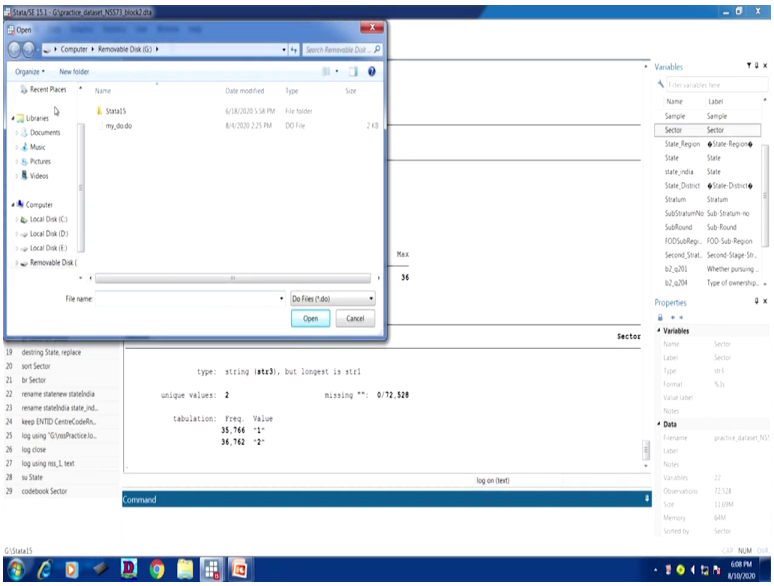
Look at so many commands are there. Look at the do-file we created for our use, for our clarity I will guide you in between what are those do-files and what it contains and how it works. So it is very clearly guiding you to execute the command, to understand the command, some annotation to our command. So many things are there I will guide you in between.

You can also open the do-file from here, by clicking here, new do-file. If you just click, it will also open. What I will do, I will tell you how to do it, once you have define a do-file, you can simply copy. What I will do, simply here will do it erase all the wrong, on the review side, if you click on this part here in the exclamation mark reverse, the errors symbol is there.

Once you click there, just put a click here, it has erased all the errors. You can copy entire commands and right click on it and paste it. If you right click on it, it will show you that copy all, like this. Select all and send selected to the do editor. If you send it, It automatically saves with a do-file whatever the command we already run that is saved.

(Refer Slide Time: 25:10)

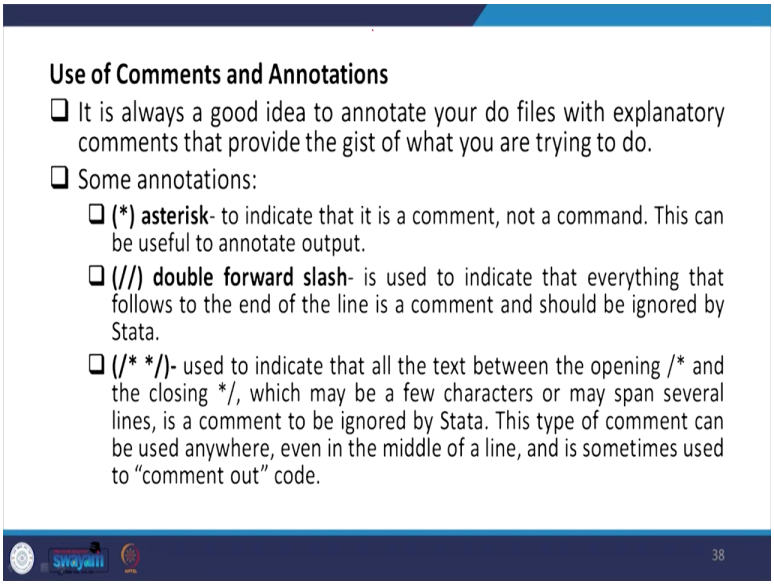


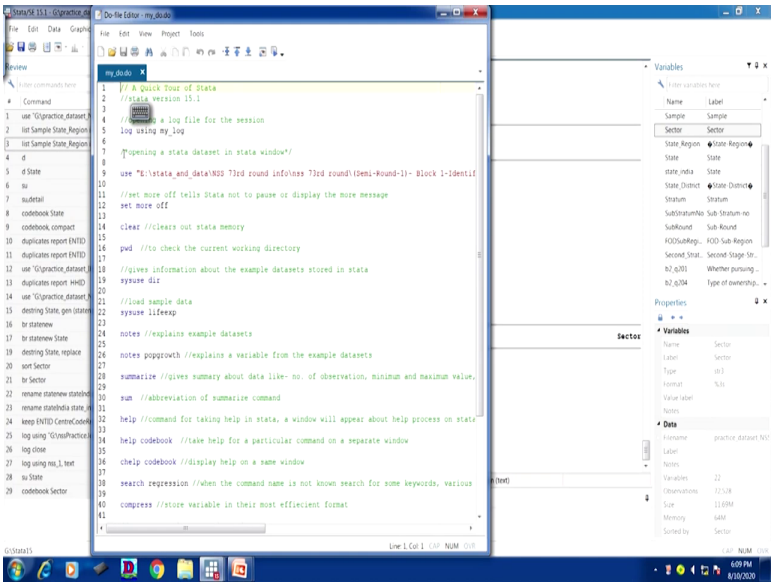


let me proceed further. Like, this is one approach that we have guided. Another approach is from the main menu at the top right, launch the do like file editor, start typing your commands. The way I have shown you we can type our command for new files or on do-file editor window from main menu at the top left click on the file icon browse to locate and okay. Like what I have just shown you, so here file then do. If you click on the do, it will prompt you to save with a name. Once you save it, another do-file will be created for you.

Once the do-file you created, you can copy those and paste it or you can manually enter as per your requirement. I am going to guide you how to enter in our successive slides, in another two, three slides we are going to complete the session of understanding do and other operations.

(Refer Slide Time: 26:30)





Use of comments and annotations especially in the do-file, which is very very important for advanced users. Even the basic users those are very handy with this, they can smartly operate Stata. It is always a good idea to annotate your do-files with explanatory comments that provide the gist of what you are trying to do. One thing I wanted to carefully guide you that use asterisks,

these annotations are going to be very very useful in Stata. Asterisks, the star mark, which we have shown in the first bullet point that indicates a comment, but not a command I will just show you. We have already defined that. Look at this, here is an asterisk, but stated with another one, I am going to guide you. Asterisk another one, I will just guide you. Asterisk opening a Stata dataset in like, I will tell you.

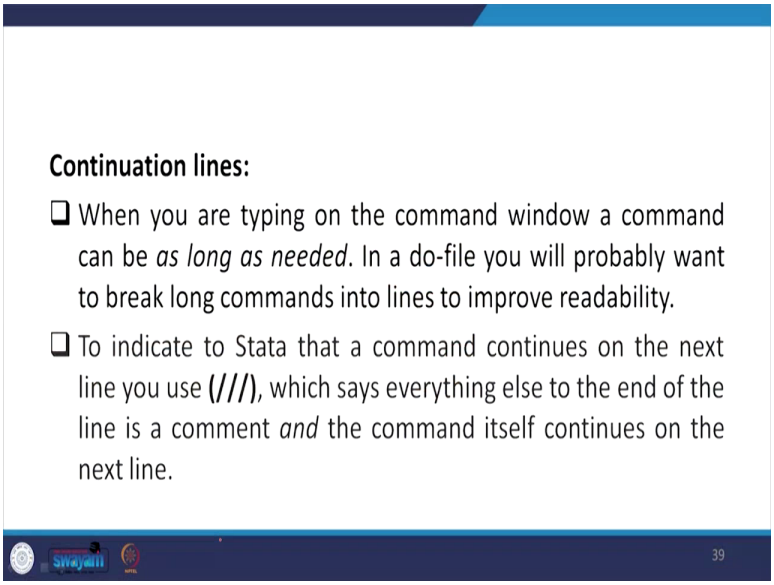
Let me first tell you what are others windows, asterisk, double slash and slash asterisks in between then asterisks slash what it indicates, let me first clarify, then we understand. Asterisks simply indicate comment, not a command. Whereas, if there is double forward slash, not the backward slash, this is used to indicate that everything that follows to the end of that line is a comment and should be ignored by Stata. If this is there, that should be ignored by Stata.

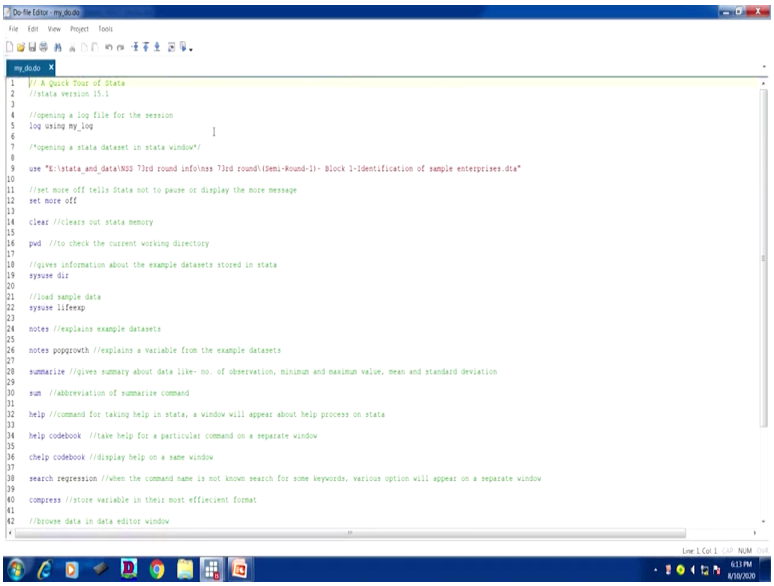
Let me first cover that then we will go by this. And if single forward slash with a star mark in between text this end with star and close with a forward slash, this is used to indicate that all the text between the opening and the closing which maybe a few characters or may span several lines is a comment to be also ignored by Stata.

This type of comment can be used anywhere, even in the middle of a line in your do-file. Do file that we created, if you have given this, within these entries, forward slash star your comment then ended with star and forward slash, the Stata is going to ignore it for sure. Similarly, the double forward slash is basically that is indicate everything that follows to the end of that line is a comment. This type of comment can be used anywhere in the middle of a line and sometimes used to comment out code.

Basically, you need to clarify about the code. You need to comment about the code you do not know, if a code is very specific, you do not know how to understand it you can comment out it.

(Refer Slide Time: 29:36)





Another one, let me just clarify, after that I will show you in the do-file. When you are typing on the command window, a command can be as long as needed. In a do-file, you will probably want to break long commands into lines for better readability in that case you may use the triple forward slash.

To indicate a Stata that a command continues on the next line you use, you have continued a command, but command is a pretty longer, it is very difficult to read, simply you enter that you start from enter a triple forward slash and you start with a next line. That basically breaking the

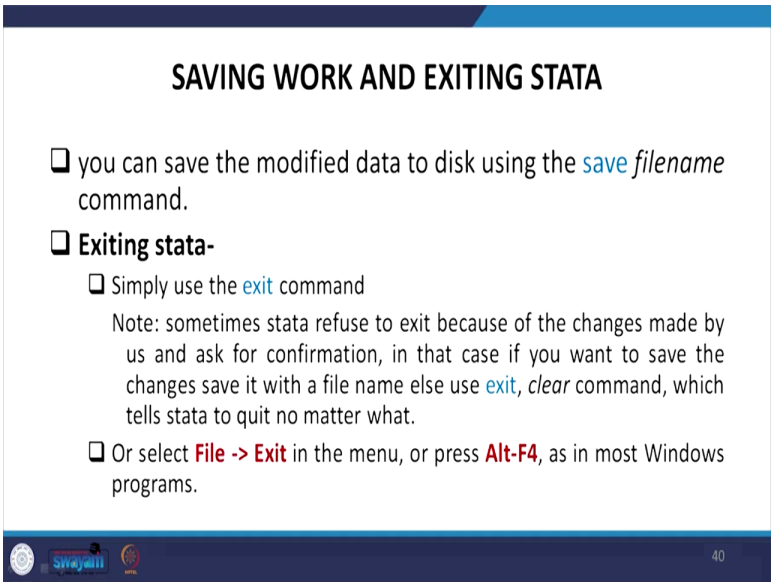
command and Stata reads in continuity. These three slash does mean that Stata reads in continuity.

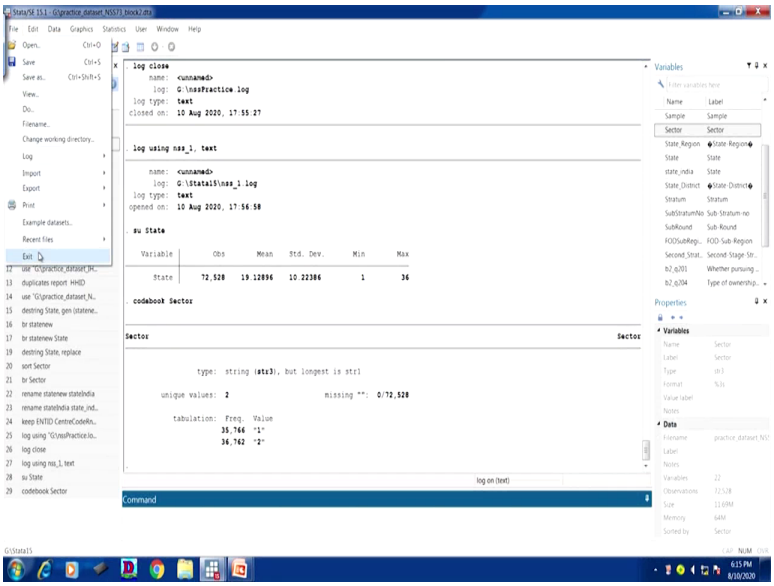
So only last slide is left let me show you once how we opened that and how that reads. Some minimum aspect of it, rest of the details we will clarify later. Other entries will certainly operate while we will be discussing. Look at this carefully, one forward slash here then star then we have written something then star then ended with a forward slash that means Stata is going to ignore it for sure, it is not a command.

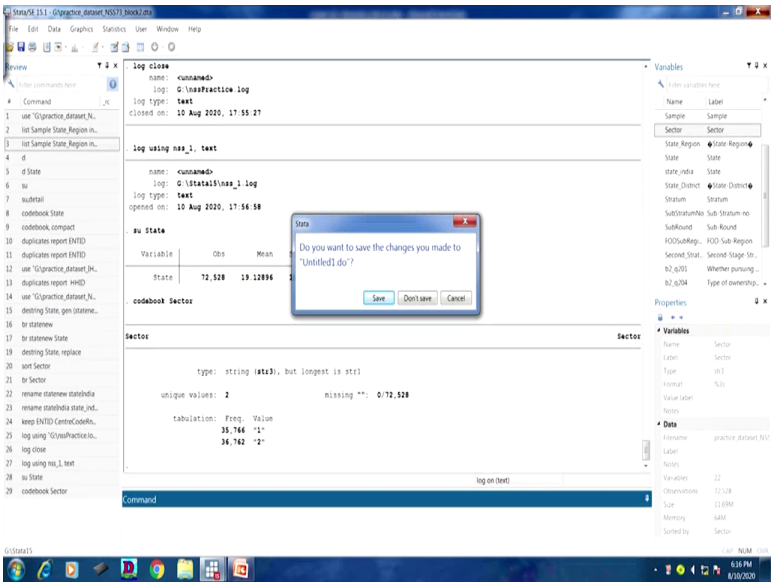
Similarly, starting with a double forward slash, Stata is also equally going to ignore. Triple slash we have not used here, we will use it certainly when we do require for a long command in our operation. Triple slash to break that if such a bigger page is there or command still continue, the same command. If a regression command is there, there are so many variables you are still continuing it, you need to break it somewhere, you need to break it to a next line, you just add triple forward triple slash.

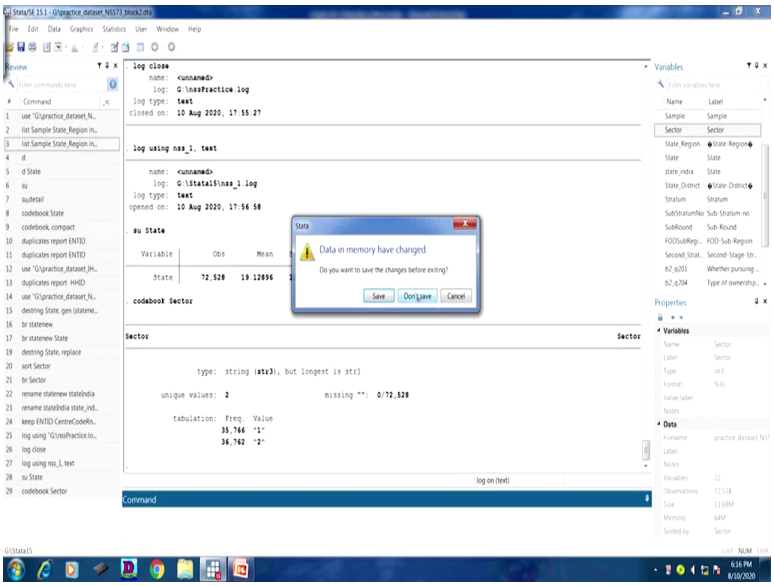
Stata is going to continue that particular command. If you select entire and execute, it will operate, we will clarify later on.

(Refer Slide Time: 32:00)




Stata/SE 15.1 - G:\practice_dataset_NSS73_block2.dta
File Edit Data Graphics Statistics User Window Help
Open... Ctrl+O
Save Ctrl+S
Save as... Ctrl+Shift+S
View...
Do...
Filename...
Change working directory...
Log
Import
Export
Print
Example datasets...
Recent files
Exit
. log close
name: <unnamed>
log: G:\nssPractice.log
log type: text
closed on: 10 Aug 2020, 17:55:27
. log using nss_1, text
name: <unnamed>
log: G:\Stata15\nss_1.log
log type: text
opened on: 10 Aug 2020, 17:56:58
. su State
Variable Obs Mean Std. Dev. Min Max
State 72,528 19.12896 10.22386 1 36
. codebook Sector
Sector Sector
type: string (str3), but longest is str1
unique values: 2 missing "": 0/72,528
tabulation: Freq. Value
35,766 "1"
36,762 "2"
log on (text)
Command
Variables
Properties
Data
G:\Stata15
6:15 PM
8/10/2020


Stata/SE 15.1 - G:\practice_dataset_NSS73_block2.dta
Review
Stata
Do you want to save the changes you made to "Untitled1.do"?
Save
Don't save
Cancel
. log close
name: <unnamed>
log: G:\nssPractice.log
log type: text
closed on: 10 Aug 2020, 17:55:27
. log using nss_1, text
name: <unnamed>
log: G:\Stata15\nss_1.log
log type: text
opened on: 10 Aug 2020, 17:56:58
. su State
. codebook Sector
Sector Sector
type: string (str3), but longest is str1
unique values: 2 missing "": 0/72,528
tabulation: Freq. Value
35,766 "1"
36,762 "2"
log on (text)
Command
Variables
Properties
Data
G:\Stata15
6:16 PM
8/10/2020



The last slide to be discussed at this moment is that, which is very important also, do not forget to save, otherwise Stata may not save the data. You can save the modified data to disk using the save with the filename command. Save then filename. If you do that, it will save. Otherwise, if you close it, it will also ask for save. You need to save with that filename. You create your file name.

And simultaneously exiting Stata is important. I am going to do that at the end. With that, we will close it, but simply exit is also going to exit your Stata. Note that, sometimes Stata refuse to exit because of the changes made in between or ask us for confirmation. In that case, if you want to save, which I just said, you can save with a new created file or if you do not want to save, then simply exit, it will clear all the command.

And otherwise through the pointer also, from the dropdown menus also from the file, there is also, here exit. If you go by that you can also close it. Another easy option, those advanced users usually do if they are very handy with the keyboard Alt F4. Usually we shutdown our computer Alt F4 is also going to close it.

So our lecture is going to be closed with the one if I just close my window. Thank you very much. It has popped up. I am going to close it. It asks for save. Since I do not want to save it, I will go by my own dataset, so I do not save. If you want to save, it will ask for a save. So do not save, and data memory have been changed then it is saved. With this, let me stop here. Thank you.