Introduction to Stata

Introduction

In introductory biostatistics courses, you will use the Stata software to apply statistical concepts and practice analyses. Most of the commands you will need are available in two different ways, (1) typing commands and (2) using drop-down menus. Today, we will focus on providing instruction on (1) typing commands because it is quicker, more straightforward, and allows direct coding.

This session will follow as closely as possible to the Stata video modules posted online.

Note: To denote use of drop-down menus (2) and their sub-menus and items in this session, we will write: *Menu/Sub-menu/Menu Item*. The symbol ◊ indicates a command or task you should be performing in Stata.

1. Getting oriented with Stata

1.1 Opening Stata

◊ To get started today, log onto the VDI. If you are unsure how to access VDI, please visit this website.

◊ Navigate to the Start menu, select “All Programs” at the bottom, then open the “Stata 15” folder, and select the file “StataIC 15 (64-bit)” to open Stata.

1.2 Stata Windows

The Stata interface is divided into five windows. The first four windows appear when you open Stata. (Note: Older versions of Stata and different computing systems may be different from the screen shot below, but the same five windows should still show. If they are not all there click on *Edit/Preferences/Load Preference Set/Factory Settings*).

1. *Command* Window
   This is where you type commands into Stata in order to perform statistical analyses, plot graphs, etc. Commands are typed into this text box and then submitted by hitting the Enter key. [Many of the commands that you will use in this course are also available in the *Data*, *Statistics* and *Graphics* menus found on the toolbar at the top of the Stata screen.]

2. *Results* Window
   This is where the results of any command performed by Stata will appear.

3. *Variables* Window
   When you have a data set open, or if you have created a new data set, the variable names within the data set will be listed here.

4. *Review* Window

The *Review* window is useful because as you send commands to Stata they appear here even if you use the drop down menus to perform your analyses. You can also call back the commands [i.e. reenter them in the command line] from the *Review* window by clicking on them, editing them in the *Command* window, and then running them by hitting the Enter key.

5. *Graph* Window
   This window appears when you create a graph. The resulting graph can then be included in a word-processed document, which we will talk about later.

1.3 Stata Menus

As previously mentioned, most of the commands that you will use are available through the use of drop-down menus. The menus can be found on the Menu bar at the top of the Stata window. The following is a description of the Stata menus:

- *File* – Opens and saves Stata data files. Opens and closes log files. Saves or prints graphs. Imports and exports ASCII and Excel files. Exits Stata.
- *Edit* – Allows you to copy output from the *Results* or *Graph* windows to a word processor or other application. It also allows you to go back to Stata's default windowing or your own options as saved previously.
- *Data* – Open the Data Editor and Data Browser. Summarize data. Label data sets and variables. Create new variables. Sort data.
- *Graphics* – Contains all of Stata's graphing tools.
- *Statistics* – Data summaries and all statistical tests.
- *User* – Place to store any user-generated commands.
- *Window* – Controls the windows opened in Stata and allows you to open a Do-file Editor.
- *Help* – A good resource if you have questions about how to use Stata.

1.4 Data Editor

The *Data Editor* is where you can enter new data or make changes to the current dataset.

◊ Open the *Data Editor* by clicking the *Data Editor* button on the toolbar at the top of the screen. (The button that looks like a pencil writing on a spreadsheet.) Alternatively, you could choose *Data/Data Editor/Data Editor (Edit)* from the menu bar.

Note: The *Data Editor* is <u>different</u> from the *Data Browser*. The Browser only allows you to look at the data and does not allow you to make changes to the data set. The button for the *Data Browser* is located on the toolbar and looks like a spreadsheet with a magnifying glass in the corner. (From the menu bar: *Data/Data Editor/Data Editor (Browse)*.)

2. Loading Files in Stata

2.1 Loading a Stata file

Normally, you will want to open an existing data file instead of manually typing data into Stata as we did above. Stata data files have the extension `.dta`. Files can be opened in Stata by clicking on the *File menu*.

- ◊ We will now load a file into Stata. Navigate to the *File/Open* dropdown and then navigate to your P drive where the dataset should be saved. Select the `heart.dta` file, and it should open in Stata.
- ◊ What happened in the Variables window?
- ◊ Open the Data Browser to view the data.

2.2 Loading other file types

You can also load in non-Stata files into Stata. You can do this by the dropdown menus via *File/Import* and then select the file type you would like to convert to a Stata file.

3. Saving in Stata

In Stata, there is a distinction between saving the data file and your commands and results. We will discuss how to do both before proceeding.

3.1 Saving data files

To save a data file in Stata, we can use the dropdown menus. Note that this will only save the Stata file along with any changes made to it. This will save over the original data file you loaded into Stata. If you do not want to do this, you should save as a new version.

- ◊ Navigate to the *File/Save* dropdown. This will replace your current file, which is fine as we have not made any changes.

Notice that a command populates in the Results window. You could also save your files by writing this command instead of using the dropdowns.

3.2 Saving your work: Log-files

The log file is where you save the commands and results from your Stata session. Basically, as long as a log file is open everything that appears in the *Results* window is saved in the log file. Graphs, however, are not saved to the log file; they must be saved separately (more on this later).

- ◊ Create a log file by clicking on the button in the toolbar that looks like a spiral notebook. Alternatively, you could navigate the dropdowns *File/Log/Begin*.

- ◊ In the window that pops up, enter a file name. For example, you could call it "Stata_intro_2018" Choose the *.log* file type, and click on Save. The *.log* extension is used so that the log file can be opened in most word processing applications.

- ◊ Look at what appears in the *Results* window.

Once you have opened a log file, it will record everything you do until you suspend it or close it. You can do this by clicking on the same button you clicked on to create the file, then selecting the appropriate option. (From the menu bar: *File/Log* then select the appropriate option). When you choose the close option, you close the log file and have to open a new one before you can start recording again. However, with the suspend option you pause recording until you click the log button again, at which point Stata starts recording in the *same* log file. You can view the current contents of the log file by choosing the *view snapshot of log file* option.

Note: When turning in Stata output for your homework assignments, please, please, please only hand in only the relevant parts of your analyses. That is, instead of turning in the log file, copy & paste relevant parts of the code and output into a Word document, as we do here. This will help your TAs immensely!

4. Describing your dataset

The heart dataset contains information on one-year-old infants born with congenital heart disease who underwent reparative heart surgery in the first three months of life. There were two types of surgery: circulatory arrest (CA) and low-flow bypass (LF). This is encoded in the `trtment` variable. To assess the impact of each type of surgery, information was collected on two indices of the Bayley Scales of Infant Development: `mdi` and `pdi`.

4.1 The `describe` and `codebook` commands

◊ If you have not done so already, load the `heart.dta` file as described above.

◊ The describe command will report the number of observations and number of variables in your dataset

. describe ***Describes the variables in the current data set***

```
  obs:           157
 vars:             3
 size:         3,768
----------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
----------------------------------------------------------------------------
trtment         double  %9.0f      trtment    treatment group
pdi             double  %9.0f                 psychomotor development index
mdi             double  %9.0f                 mental development index
----------------------------------------------------------------------------
Sorted by:
```

◊ The codebook command will return information for a specific variable.

. codebook mdi ***Gives detailed information about a variable***

```
----------------------------------------------------------------------------
mdi                                               mental development index
----------------------------------------------------------------------------

                  type:  numeric (double)

                 range:  [50,142]                     units:  1
         unique values:  41                       missing .:  13/157
```

```
                  mean:   104.736
              std. dev:   15.6044

           percentiles:        10%        25%        50%        75%        90%
                                86         96      106.5        115        122
```

4.2 Missing values

Missing values in Stata are coded as a period. When you run commands on variables with missing values, it will ignore the missing values (i.e. if you are taking a mean of a variable, Stata will only compute the mean for non-missing values). In other instances, Stata interprets a missing value as a very large number; this may cause issues when using logic statements.

◊ Open the Data Browser and find an infant with a missing mdi score

4.3 Errors in Stata

When typing commands into the command window, spelling or punctuation mistakes often occur. In these cases, Stata will return an error message in red with a description of the error.

◊ Incorrectly enter one of the commands we performed above

```
. cdebook mdi
command cdebook is unrecognized
r(199);
```

If you receive an error in Stata, read the message it supplies. If you do not understand the error message, you can always try a quick google search as it is likely someone has encountered the same error in the past.

5. Variables

5.1 Generating a new variable with formulas

Sometimes you may need to create a new variable that is a function of one or more existing variables.

◊ Create a new variable called `mdisq` that is the square of gestational age (`mdi`^{2}).

```
. generate mdisq = mdi^2
(13 missing values generated)
```

5.2 Generating a new variable with a logic statement

A logic statement is a declarative sentence that will evaluate to either true or false. In Stata, these start with an `if` statement. They are very useful when creating new variables or only performing commands on a subset of the data.

The goal here is to create a new variable called `mdicat` that serves as an indicator for mdi score being above or below a score of 90.

◊ Create a new variable called `mdicat` that takes value 1 if mdi is less than 90.

```
. generate mdicat = 1 if mdi < 90
(137 missing values generated)
```

◊ Replace values of this variable with 0 if mdi is greater than or equal to 90.

```
. replace mdicat = 0 if mdi >= 90
(137 real changes made)
```

◊ Recall that missing values will be replaced with a 0 in the above statement (why?). You need to correct this by ensuring the missing values from `mdi` are also missing in `mdicat`.

```
. replace mdicat = . if mdi == .
(13 real changes made, 13 to missing)
```

5.3 Removing a variable

If you wish to remove a variable from your dataset, use the drop command

◊ Remove the mdisq variables as we do not need it for our analysis

```
. drop mdisq
```

5.4 Labels for Variables

For the sake of clarity you may want to label the variables in your dataset. This is helpful when you have given abbreviated names to your variables and want to keep a more detailed explanation of the variables' contents. These labels will appear in any tables or graphs that you make with these data.

◊ The labels for variables are in the *Variables* window. What is the label for the `mdi` variable?

◊ You can edit existing or add labels to variables using the following command:

```
label variable <variable_name> "<variable label>"
```

For example, we can enter:

```
label variable mdicat "moderate motor impairment"
```

6. Data Analysis

Now, that we have labeled and created necessary variables and have a basic understanding of what is contained in our dataset, we can use commands to give us summaries of our variables.

◊ We will go over a few commands here, so follow along in Stata.

```
. tabulate trtment
```

Tabulates the values of a categorical variable

```
  treatment |
      group |      Freq.     Percent        Cum.
------------+-----------------------------------
         CA |         81       51.59       51.59
         LF |         76       48.41      100.00
------------+-----------------------------------
      Total |        157      100.00
```

```
. tabulate trtment mdicat
```

Creates a table for two categorical variables

```
           |    moderate motor
 treatment |      impairment
     group |         0          1 |     Total
-----------+----------------------+----------
        CA |        62         12 |        74
        LF |        62          8 |        70
-----------+----------------------+----------
     Total |       124         20 |       144
```

```
. list trtment mdi in 1/10
```

Lists the designated variables for the first 10 observations in the data set

```
     +----------------+
     | trtment   mdi |
     |----------------|
  1. |      CA    74 |
  2. |      LF   124 |
  3. |      LF   109 |
  4. |      CA    78 |
  5. |      CA    91 |
     |----------------|
  6. |      CA   130 |
  7. |      LF   119 |
  8. |      LF     . |
  9. |      CA   115 |
 10. |      LF   112 |
     +----------------+
```

```
. summarize mdi
```

Displays summary statistics for a continuous variable

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         mdi |       144    104.7361    15.60437         50        142
```

```
. summarize mdi, detail
```

Displays additional summaries, including percentiles and the 4 largest and 4 smallest values

```
                  mental development index
-------------------------------------------------------------
      Percentiles      Smallest
 1%           56             50
 5%           78             56
10%           86             58       Obs                 144
25%           96             70       Sum of Wgt.         144

50%        106.5                      Mean           104.7361
                        Largest       Std. Dev.      15.60437
75%          115            131
90%          122            131       Variance       243.4963
95%          130            140       Skewness      -.6774051
99%          140            142       Kurtosis       4.119965
```

```
. sort trtment
```

Sorts data by the values of the specified variable *(<u>Note:</u> you will only see changes in Data Browser or Editor)*

Data must be sorted before using a "by" command (see next step)

```
. by trtment: summarize mdi
```

Summarizes a continuous variable, stratified by the categories of the "by" variable

```
-----------------------------------------------------------------------------
-> trtment = CA

    Variable |         Obs        Mean    Std. Dev.       Min        Max
-------------+-----------------------------------------------------------
         mdi |          74    103.1622    16.46501         56        142

-----------------------------------------------------------------------------
-> trtment = LF

    Variable |         Obs        Mean    Std. Dev.       Min        Max
-------------+-----------------------------------------------------------
         mdi |          70       106.4    14.57256         50        130
```

- ◊ We can use logic statements to summarize a continuous variable by another variable. To find the mean value of mdi among infants who received CA treatment,

```
. summarize mdi if trtment==0

    Variable |         Obs        Mean    Std. Dev.       Min        Max
-------------+-----------------------------------------------------------
         mdi |          74    103.1622    16.46501         56        142
```

7. Graphing

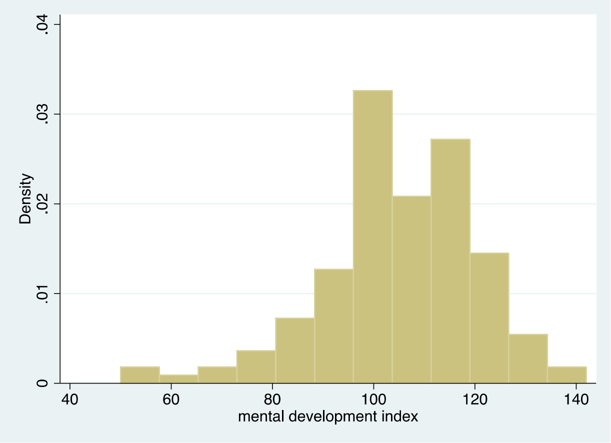
7.1 Creating Graphs

To create graphs in Stata it is often helpful to use the drop-down menus as these will provide more options for you to modify your graphs.

- ◊ Go to *Graphics/Histogram* and enter mdi into the variable box. Select "Okay". A graph window will appear with the histogram. Also, notice that the command is populated in the Results window.
- ◊ Close out of the Graph window and manually enter the following

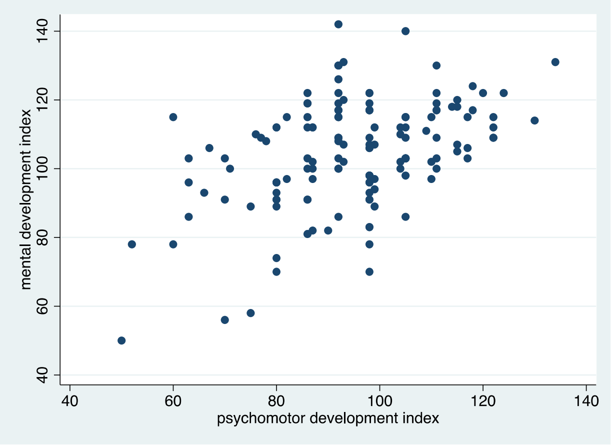
```
. histogram mdi
```

Constructs a basic histogram for a continuous variable



```
. scatter mdi pdi
```

Constructs a basic two-way scatter plot



7.2 Saving Graphs

While the graph window is active, under the *File* menu you have the option to either print the graph or save it (you can recall the saved graphs by *File/Open graph*). You can also copy and paste graphs to word processing programs by choosing *Copy Graph* from the *Edit* menu while the graph window is active and then pasting it into a Word document.

8. Wrap-up / Miscellaneous topics

8.1 Using Stata as a Calculator

You can use Stata as a calculator. For example, if you want to know what 2 + 2 is, you type:

```
. display 2+2
4
```

The `display` portion is very important – this lets Stata know you want to use it as a calculator. You will get an error if you do not type it.

8.2 Stopping your log-file

You can save your Stata file using the dropdowns *File/Save* or by clicking on the 'Save' button. Note that this does **not** save any information from the Command or Results window (this is what the log file is for). This will only save any changes that were made to the data file (i.e. new variable labels, new variables, etc.).

8.3 Saving your work in Stata: Do-files *(note: this is omitted from the online video)*

Unlike the log-file, the do-file will not track everything you do in the Results window. It instead provides a blank script for you to selectively write commands that you may want to save and re-run in the future. You can run the commands directly from the do-file. However, it will not save the output from the commands.

◊ Create a do file by selecting the button in the toolbar that looks like a blank notebook. Alternatively, you could navigate the dropdowns *File/New/Do-file*.

◊ In the blank window that appears, we will first write a comment starting with a star/asterisk. Type: `*INTRO TO STATA SESSION: IMPORTANT COMMANDS*`

◊ Choose two commands from section 6 to write in the script. Provide comments to remind yourself what they do.

◊ Run the all commands in the script by clicking the "Do" button in the do-file. You can also run individual commands by highlighting the command and clicking "Do".

◊ Save your do-file and close.

8.4 Closing Stata

Before exiting out of Stata, remember to save your updated heart dataset. I would recommend saving this as a new version called `heart_v2.dta`