Introduction to use of Stata in VHM 802/812

Jenny Yu
January 11, 2021

1. This session assumes you have the following knowledge of Stata:

Understand and/or know how to:

- Interact with Stata; organize your files
- Load non-Stata format data into Stata (mainly Excel files and csv files)
- Variable naming conventions; variable types, data types; missing values
- Convert between numeric variables and string variables
- Generate a categorical/an indicator variable from a continuous/string variable
- Label a variable; define a value label and attach it to a variable
- Explore Stata Menus, Toolbar and dialog boxes
- Use Stata functions
- Create reproducible do-files and log files in your working folder

2. Steps in the statistical analysis:

- Plan and organize your files and directories
  - systematically think ahead - **simple and consistent** principles
    - Directories on your hard drive – where to store your course and/or research materials
    - One (sub-)directory only for one project – so it's **easy to find** your project files
    - Names for projects, folders and files – all **mnemonic**
  - **Separate do-files for data management and statistical analysis**
    - Individual do-files: typically (but not always) start with the letters cr or an
    - Or create your own rules for your do-files' names
- Data management (vhm811/801): well documented do-files
  - Verify your data are accurate and no duplicate observations; missing values are properly handled
  - Generate the variables that are required for the next step of statistical analysis; make sure all variables are **well named and properly labeled**
  - …
- **Running statistical analysis (vhm8120)**
- Presenting results – interpret your analysis results
- Protecting files – **documentation** to keep track of what you have done and thought

3. A sample do-file for statistical analysis: 1aL.do

- Robust
  - Replace line 4 in 1aL.do with your own working directory, then click 'Execute(do)' on the Toolbar
- Legible
  - Comments & blank lines
    - Different styles of comment indicators, e.g.: use three forward slashes (///) where you want to split a long line into several lines to make long lines more readable

. help comments

4. The workflow of the do-file: 1aL.do

- Understand the data
  - Understand the data structure

- Assure the outcome variable and all relevant predictors are numeric variables

- Describe variables
  - Understand the distribution of each of the variables on its own
  - Understand the nature and strength of relationships among variables
  - Create summaries or distribution tables
    - Data > Describe data > Describe data contents (codebook)
    - Statistics > Summaries, tables, and tests > Summary and descriptive statistics (summarize)
    - Statistics > Summaries, tables, and tests > Frequency tables
    - Statistics > Summaries, tables, and tests > Other tables > Compact table of summary statistics (tabstat)
      - (The following is new to Stata 17)
    - Statistics > Summaries, tables, and tests > Tables and collections > Build and style table
  - Create graphs
    - Graphics Menu
    - Statistics > Nonparametric analysis > Lowess smoothing
- Statistical model
  - Statistics Menu/commands
- Model evaluation/postestimation:
  - After the statistical model, click the menu Statistics > Postestimation
    - Acquire model predicted values
      - Generate variables/values required for model estimation and/or forecasting
      - Create graphs if needed
      - Display/list/investigate selected values
    - Evaluate the model validity: compute model residuals
  - Individual residual checking
    - List the extreme residuals: list command
  - Overall residual checking
    - Can use summary statistics commands: such as
      - . list /.display
      - . summarize
      - . tabstat
    - Assessing normality and homoscedasticity of residuals
    - Other assessment: residuals versus fitted value – linearity
  - Is the model valid?
    - Yes – interpretation and reporting
    - No – go back to modify the previous model
      - Build a different model by power-transformation outcome
      - Repeat the model building and evaluation process
      - Backtransform outcome

5. Review Stata language syntax:

- . h language

  [prefix:] command [varlist] [=exp] [if] [in] [weight] [using filename] *[, options]*
  - Square brackets are optional (as opposed to the required), fill them in only if needed
  - Each item should be typed exactly as they appear in the order of the diagram
  - *Options,* are specified at the end of the command; a comma must precede the first option
  - Click the links in the language help file to find out each of the items in the language syntax
    - . version
    - . pwd
    - . count

- . h set /. h more

  set more {on|off} [, permanently]
  - Must select one item inside the curly brackets {}

6. Create Stata graphs:

6.1 Create and edit graphs using menus

- . twoway (scatter milk120 parity) (lfit milk120 parity) /*order of scatter & lfit matters*/
  - Graphics > Twoway graph (scatter, line, etc.)
    - Click Create button under the Plots tab to create Plot 1 for scatter plot
    - Select Basic plots & Scatter, select Y & X variables, click Accept
    - Click Create button again to create Plot 2 for twoway linear prediction plot (lfit)
    - Select Fit plots & Scatter, select Y & X variables, click Accept/Submit
- . twoway (scatter milk120 parity, msymbol(circle_hollow)) ///
  (lfit milk120 parity, range(0 8)), ///
  ytitle("milk120") xtitle("parity") name(a, replace)
  - Edit Plot 1 of scatter plot:
    - Click Plot 1 under Plot definitions, then click Eidt button on the right
    - Click Marker properties, click dropdown arrow for symbol, then select Hollow circle, then click Accept, click Accept again to finish editing Plot 1
  - Edit Plot 2 of linear prediction plot - lfit:
    - Click Plot 2 under Plot definitions, then click Eidt button to edit Plot 2 of lfit
    - Click Options button, check the box of "Override range of X for which predictions are to be calculated", fin the box with 0 for min value and 8 for max value, then click Accept, click Accept again to finish editing Plot 2

6.2 Use Graph Editor and Graph Play*

- . h graph editor
  - Use graph editor to change the look of the graphs; record the changes and replay them again
- Create a Graph Recording file:
  - Click **Start Graph Editor** on your Stata graph
  - Click **Start recording,** then start editing your graph….once finish editing your graph
  - Click **End recording**
  - Click **browse** and navigate to your project folder, **type a name** (myScatter) to give your Recording File a name, then click Save button
  - Click **Stop Graph Editor.**
- To play the recording by point and click:
  - Click **Start Graph Editor** on your Stata graph
  - Click on the **Play Recording** button. You will be presented with a list of your recordings
  - Select the recording name you want to play, the edits will be applied to your graph
  - Click **Stop Graph Editor**
- To play the recording in your do-file (recording file is saved in your current working folder):
  - . twoway (scatter milk120 parity) (lfit milk120 parity), play(myscatterlfit)

7. Stata estimation commands – Linear regression

- Explore the menu and remember the commands
- . h regress
- . h regress_postestimation
  - Click predict to understand the statistics (or the options) of predict command for regress: xb, re, rsta, rstu, stdp, stdf…

- Or from the menu to find: rvfplot
  - Statistics > Linear models and related > Regression diagnostics > Residual-versus-fitted plot

8. Stata operators:

- Stata has four different classes of operators:
  - **Arithmetic operators:** + (addition), - (subtraction), * (multiplication), / (division), ^ (raise to a power), and the prefix - (negation). Any arithmetic operation on a missing value or an impossible arithmetic operation (such as division by zero) yields a missing value.
  - **String operators:** The + and * signs are also used as string operators.
  - **Relational operators:** > (greater than), < (less than), >= (greater than or equal), <= (less than or equal), == (equal), != (not equal), ~=(also not equal)
  - **Logical operators:** & (and), | (or), and ! (not).
    - The logical operators interpret any nonzero value (including missing) as true and zero as false.

9. Stata functions:

- Functions are used in expressions, which are abbreviated **exp** in Stata language syntax diagrams. e.g.
  - . generate *newvar = exp*
  - . gen log_milk = ln(milk120) //ln() is a function
- Functions may be specified in any expression. The arguments of a function may be any expression, including other functions.
- A function's arguments are enclosed in parentheses and, if there are multiple arguments, separated by commas.
  - . display exp(ln(2)*.1725382)
- Seek help from Stata documentation for commonly used functions: date and time, mathematical, random-number, statistical, string, and trigonometric functions
  - . h functions
  - . h invttail

10. Debugging do-files: determining the source of an error

- Read the error messages or click error code:
  - If you understand the error message, there is no need to explore the error code
  - . h error
- Common errors – fix them
  - no; dataset in memory has changed since last saved
    - ask yourself whether you want to overwrite the current dataset in memory
  - log file already exists
    - ask yourself whether you want to overwrite the log file
  - incorrect command name:
    - . codbook // unrecognized command:
  - incorrect option of the command:
    - . codbook, b //option b not found
  - incorrect variable name:
    - . codebook milkl20 //variable milkl20 not found
  - missing comma before option:
    - . use daisy2red clear //invalid 'clear'
    - . predict se_mean stdp //too many variables specified
  - A few more error messages:
    - not sorted; variable xxxx already defined; too many values, varlist required, = exp required, using required, by() option required …

- Tolerate errors* - allowing the command/do-file to continue despite errors
  - . capture: suppressing all its output (including error messages, if any) and issues a return code of zero
    - . h capture
    - . capture log close
    - . capture drop rawres stdres delres
      (note: Stata commands do either exactly what you say or nothing at all. If one of the three variables did not exist in the data, drop would then do nothing. It would not drop the other two. To achieve the desired result, we must give three commands:
      . capture drop rawres
      . capture drop stdres
      . capture drop delres
- Other methods to resolve errors
  - Drop all derived/generated variables, trying an alternative approach
  - Run the do-file in steps
  - Reload the data set
  - Restart Stata

11. Do-files (recap): Robust & Legible

11.1 Robust means:

- A do-file produces exactly the same result when rerunning the do-file at a later time or on different computers (only need to change the working directory)
- A do-file is self-contained:
  - Should not rely on something left in memory by a prior do-file or commands run from the Command Window
  - Should not use a dataset unless it loads the dataset itself
  - Should not compute a test of coefficient unless it estimates those coefficients in the same do-file
- Explanation for the commands:
  - Use version control - Stata might change the command names or computational methods
  - Change directory – exclude directory information or keep minimum directory location in commands that read or write files
  - Include seeds if random numbers required
    - If try to replicate results that use random numbers, you need to use the same random numbers; otherwise you will obtain different results

11.2 Legible means:

- Internally documented and carefully formatted
  - Make the content/logic clear
  - Help you and your supervisor(s)/collaborator(s) easier to debug and understand what you did
- Several practical steps:
  - Use lots of comments
  - Use blank lines
  - Use alignment and indentation if you prefer
  - Split a long line into several lines