Introduction to Stata

Chelsea Goforth
Statistical Consulting Associate

StatLab: Statistics and Data Analysis
University of Virginia Library

Fall 2015

What is Stata?

- a full-featured statistical programming language
- traditionally command-line driven syntax, but now also features point-and-click interface (will focus less on this today though)
- available for Windows, Mac OS X, Unix/Linux
- generally cheaper than either SPSS or SAS and you can purchase perpetual or annual licenses of different flavors:
  - Stata/MP: the fastest version of Stata (for dual-core and multicore/multiprocessor computers)
  - Stata/SE: for large datasets
  - Stata/IC: for moderate-sized datasets
  - Small Stata: handles small datasets (for students only)
- all versions provide the full set of features and commands (no special add-ons or 'toolboxes' to purchase)
- thousands of pages of fantastic documentation
- see the full sales pitch: http://www.stata.com/why-use-stata/

Getting Stata at UVa

1. Stata GradPlan: the software is purchased online and is delivered directly to you (electronically or via snail mail)
2. The Hive: provides access to Stata/IC
3. Computer Labs: ITS computer lab in Clark Hall and the Scholars' Lab in Alderman Library

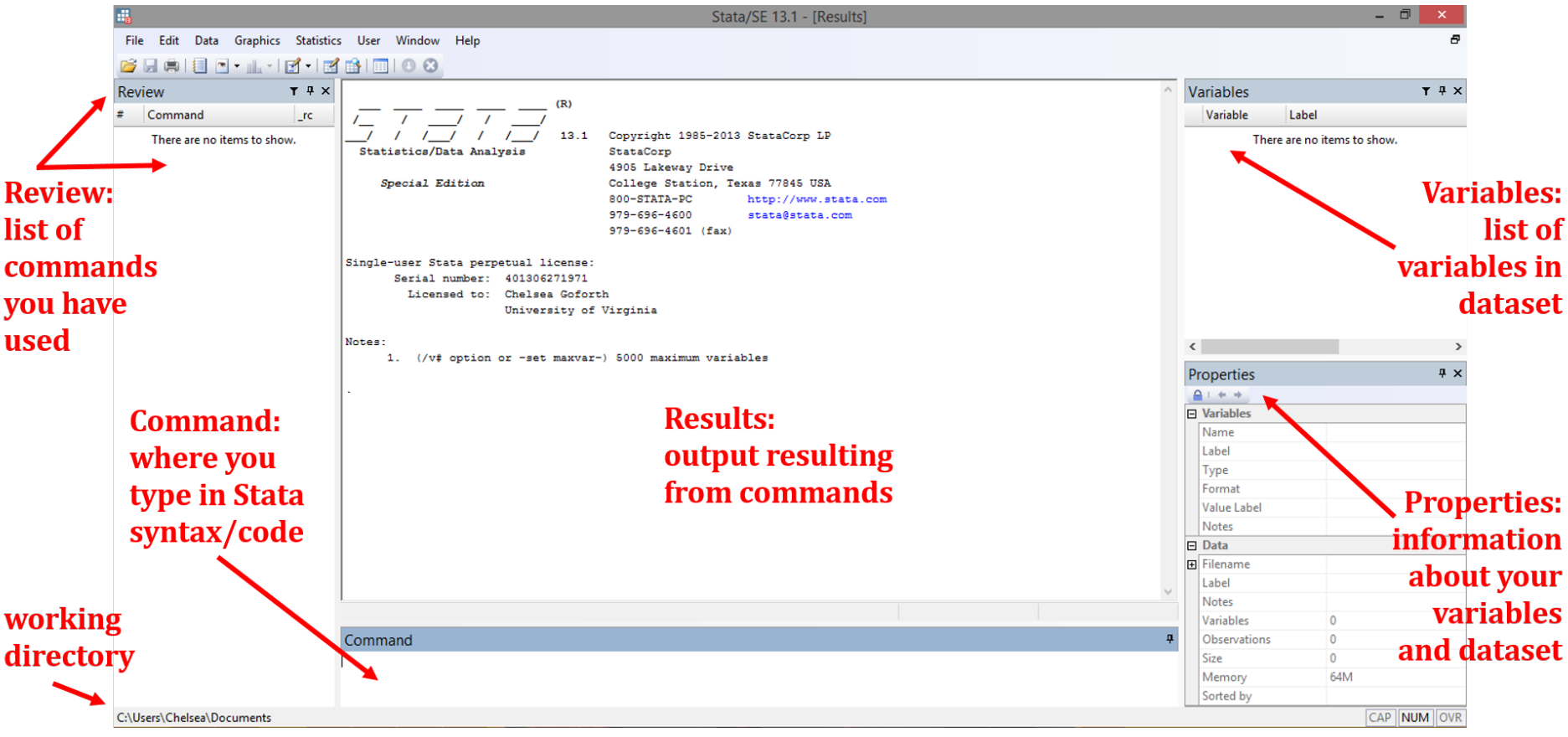
Note: The University owns 45 concurrent licenses. These apply to both computer labs and the Hive. You may have to wait for licenses to open if all are in use.

Navigating Stata: The User-Interface

The default Stata user-interface consists of five windows (and you can resize or close some of them depending on your preferences):

1. **Command**: where commands are entered for execution
2. **Results**: where all output/results are displayed (excluding graphics, which open in a separate window)
3. **Review**: a running list of all commands you've used in the order in which you've used them (a single click on one will transfer it back to the command window, and double-clicking will re-run the command)
   - Note that if you run a command that produces an error, this command will be highlighted in red in the review window, and the `_rc` column will contain a nonzero number that indicates the error code
4. **Variables**: displays a list of all variables in your dataset
5. **Properties**: displays properties of your variables and dataset, incl. variable names, labels, value labels, notes, formats, storage types

Navigating Stata: The User-Interface



Review: list of commands you have used

Variables: list of variables in dataset

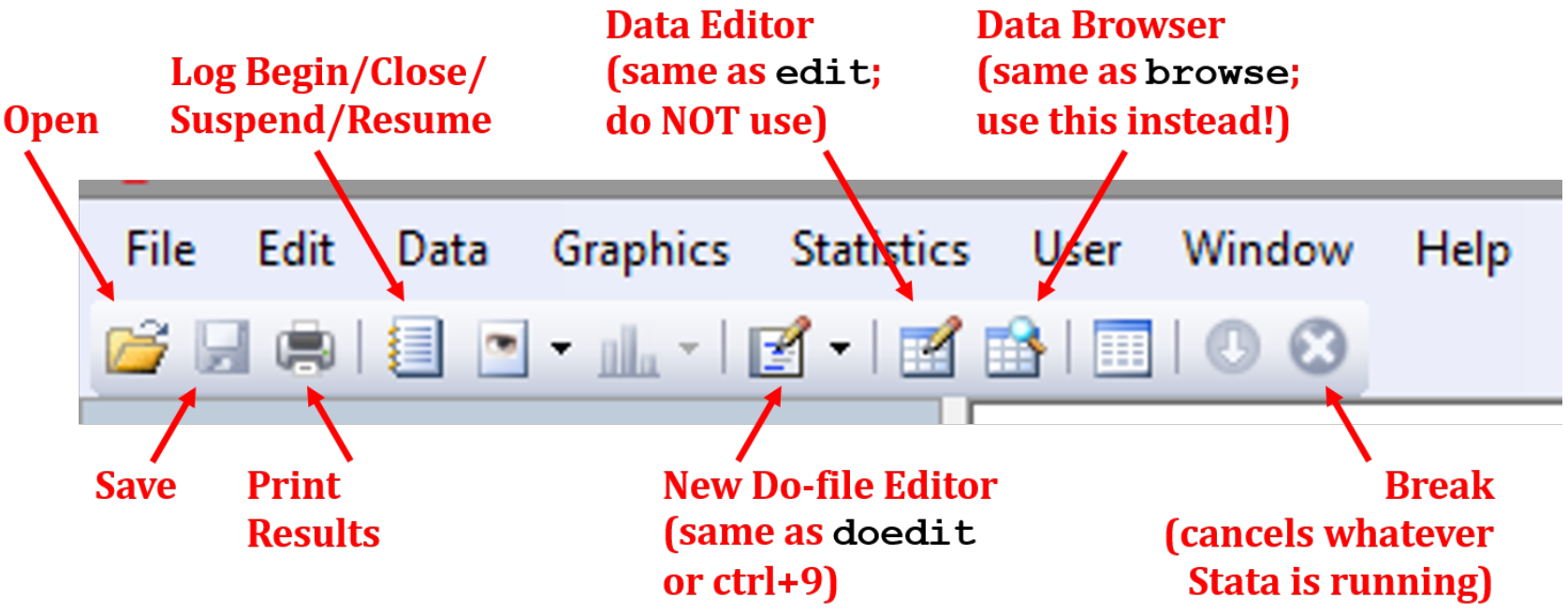
Command: where you type in Stata syntax/code

Results: output resulting from commands

Properties: information about your variables and dataset

working directory

Navigating Stata: The User-Interface



Using Stata Syntax

Some Terminology:

1. **Log File**: a full record of your Stata session; logs capture all the commands, output, and other text printed in the results window (excludes graphics)
   - Logs can be recorded in two different formats; if no file extension is specified, the default is .smcl (Stata Markup and Control Language – preserves fonts and colors) or plain text (specify either .log or .txt). You can also use the `-translate-` command to convert a .smcl log into text or other formats.
2. **Do-file**: a text file of user-written Stata commands; do-files allow you to reproduce all manipulations and analyses of the data (including graphics), allowing for easy replication. We will work from a do-file prepared in advance today.
   - The only difference between a do-file and a notepad text file is that there is a button to run commands that are typed here.
   - If nothing is highlighted, this button runs everything in the do-file; if some lines are highlighted (or partially highlighted), only those are run.

Using Stata Syntax

Stata provides a point-and-click interface, but I strongly recommended you use syntax (more specifically, a do-file). Why?

- Scientific research requires documented, reproducible research findings; using and saving syntax makes the research reproducible and collaboration easier (makes life easier for your future self as well!).
- Helps you return to original data quickly and safely; makes it easier to fix any errors and useful for version control.
- Facilitates performing alternate analyses more efficiently.
- When you Google for Stata help, the results will almost always be syntax, and the help files present all examples using syntax as well.

Syntax can be entered interactively one line at a time (via the Command window) or submitted in batch from a "do-file" (i.e., a text file of Stata commands). We'll be focusing on the latter today to emphasize the importance of record-keeping.

Using Stata Syntax

Most Stata commands have the following structure:

`command` [*variables*] [`if`] [`in`] [`weight`] [`, options`]

...where everything in square brackets is optional:

- `if` : to subset the data using variable expressions
  - see next slide for operators & do-file for examples
- `in` : to specify particular observations using row numbers
  - e.g., `-browse in 1/10-` to browse the first 10 observations
- `weight` : to account for frequency, sampling, analytic, and/or importance weights (not covered today)
- `, options` : everything following the comma after the main command syntax is an optional command (which options are available varies by command)

Again, commands are entered either interactively, one line at a time (via the Command window) or submitted in batch from a do-file.

Using Stata Syntax

Operators for subsetting with `if`:

Arithmetic		Logical		Relational	
+	addition	&	and	>	greater than
−	subtraction	|	or	<	less than
*	multiplication	!	not	>=	> or equal to
/	division	~	not	<=	< or equal to
^	power			==	equal to
−	negation			!=	not equal to
+	string concatenation			~=	not equal to

Using Stata Syntax

All commands typed interactively are lost once Stata is closed (unless you manually start a command log) – use a do-file instead!

Creating a Do-file: A Standard Preamble

```
clear
version 14
set more off
cd "C:\Users\Chelsea\Documents"
capture log close
log using "example.log", replace
```

This is the beginning of your do-file (before even reading in any data!).

Using Stata Syntax

- `clear` – clears out any existing data from Stata's working memory
- `version 14` – specifies the version of Stata you're using

  When Stata releases new versions, do-files written for older versions might stop working. This command prevents that from happening.
- `set more off` – tells Stata to continue issuing commands, even if the results window is full

  Every time the results window fills up, Stata stops, even if in the middle of a calculation, and writes `-more-` at the bottom of the screen; you have to click to see further output. This is very annoying, but if you use `-set more off-` you can let Stata run while you do other things. Note, though, that this is a temporary command. It must be run within a block of commands from a do-file for more to be set off. If you are working in the command window or if you run something in the middle of a do-file, `-more-` will still appear.

Using Stata Syntax

- `cd "C:\Users\Chelsea\Documents"` – changes the working directory to the folder specified in between the quotation marks

  It's good practice to put all of your files for one project in one folder on your computer. Then, type that folder's address or directory pathway here so that Stata will look here by default for any files specified and automatically save files to this folder as well.

- `capture log close` – closes any open log files

  Note that `-log close-` often also works to this end; however, Stata will get stuck and issue an error message if there isn't a log open to close. Adding `-capture-` to that command tells Stata to close a log if one is open, but to ignore the command and move on if not.

- `log using "example.log", replace` – opens a new log called "example" that will allow you to document a full record of your Stata session (and replaces any log in the same folder with the same name)

Using Stata Syntax

Three ways of using comments to annotate your do-file (include any notes that will explain your analyses to make replication even easier):

- use an asterisk ($*$) at the beginning of a line (single line comment)
- use two forward slashes ($//$) at the end of any command line (single line comment)
- use two forward slashes and two asterisks ($/*$ comment $*/$) to bookend your comment (multi-line comment such that anything in between the two asterisks will constitute a comment)

Syntax Highlighting (in do-files):

- `blue` $\rightarrow$ denotes built-in commands/functions
- `red` $\rightarrow$ denotes string text (not commands/functions or other non-numeric information)
- `green` $\rightarrow$ denotes comments (information not processed by Stata when executed)

Using Stata Syntax

Other tips for using Stata's syntax/do-files:

- Stata is case sensitive; commands are always in lower case
- Many commands can be abbreviated (e.g., `-tab-` instead of `-tabulate-` or `-reg-` instead of `-regress-`); the help file for each command will underline the shortest possible abbreviation
- When working via the Command window, previous commands can be recalled with the "Page Up" and "Page Down" buttons
- After the preamble, read in your data, type out all of your commands to clean data and/or run analyses, then always end your do-file by saving your data *using a different file name*. **Always preserve your original data this way!**

Computing Workflow

But why do all of this?
To prevent you from committing the cardinal sin of data management:

overwriting your original data

When writing a paper, you probably open a new Word document, type some stuff, and save often. You may even have multiple versions of the same document:

paper-20150922
paper-20150922_cgedits
paper-20150922_final
paper-20150922_final-20150925edits
paper-20150922_final-20150925_FINAL

Setting aside the fact that this isn't a terrific workflow for paper-writing, it's even more problematic for data management.

Computing Workflow

A better workflow for data management:

- Think of the original data, with all of its flaws, as sacred; and, think of a do-file as a list of instructions that will transform your original data into clean, edited data, ready for analyses.
- You will make mistakes, and when you do, you will need to go back to the original data and start over. You can change the "instructions" in the do-file, but only if you preserve your starting point.

Principles for a computing workflow:

1. Dual workflow: separate your data management from analysis and make sure each is a self-containing file
2. Run order naming: name files so that if they are re-run in alphabetical order, they will reproduce exactly the same results

Computing Workflow

Example of dual workflow and run order naming:

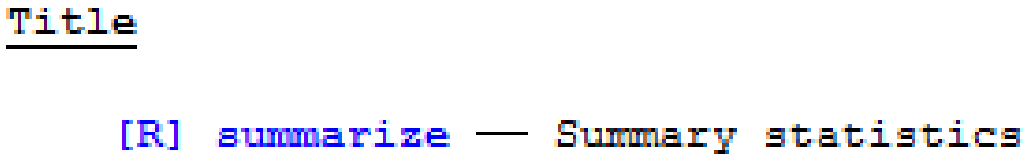
data management	**data analysis**	
data01.do	stat01a.do	stat03aV2.do
data02V2.do	stat01b.do	stat03b.do
data03.do	stat01cV2.do	stat03c.do
data03-1.do		stat03c1.do
data03-2.do	stat02a.do	stat03c2V2.do
data04.do	stat02a1.do	stat03d.do
	stat02b.do	

Getting Help with Stata

With hundreds of commands available to you, you're bound to need help at some point while using Stata. Luckily, Stata comes with over 12,000 pages of fantastic documentation; these manuals all come in PDF format. There are many ways of accessing this documentation:

1. `help` *command*

   ...will open a new viewer window and display the command's help file, which is an abbreviated version of the information found in the PDF manual. These help files also provide hyperlinks (in blue) to additional, relevant help pages, and you can click on the hyperlink in the title (see example below), which will open up to the appropriate page in the full PDF documentation.



Getting Help with Stata

Stata's help files contain the following sections (in addition to title):

- **Syntax**: the bare-bones version of the command
  - the command required to perform that operation is in black (with the shortest abbreviation underlined), you fill in the italicized parts as necessary, and all bracketed sections are optional
- **Options (table)**: a list of the words you can type after a comma to alter the behavior of a command. Short descriptions are listed here; more detailed descriptions are provided in the options section below.
- **Menu**: how to use this command via point-and-click (but don't!)
- **Description**: a short explanation of the command
- **Options (details)**: more detail for every entry in the options table
- **Examples**: examples that use Stata's example datasets
- **Stored results**: values you can pull from the results
- **Also see**: in the top-right corner; links to help pages for similar commands in case this one isn't exactly what you need

Getting Help with Stata

What if you don't know the appropriate command name?

2. `search` *`topic`* OR `findit` *`topic`*

   ...similar to the `help` command, both `search` and `findit` will open a new viewer window; however, instead of opening to a specific command, it will search the Stata documentation and other resources for the key words (each command may be followed by one or several key words).

   The `findit` command used to be an expanded version of `search`, but as of Stata 13, the two are synonymous.

Getting Help with Stata

What if you don't know the appropriate command name?

3. Help → PDF Documentation (point-and-click)

   ...will open the full PDF documentation, which will allow you to browse or click through until you find what you need.
   In addition, new with Stata 14, Quick Starts are examples of the usage of each Stata command that appear at the beginning of the documentation for each command in Stata's PDF documentation. For basic commands, quick starts help new users get started and highlight useful but less well-known options. For complex commands, they simplify the syntax diagram and how options and combinations of options for common tasks.

Getting Help with Stata

Other Online Resources for Learning Stata:

- Stata FAQs: http://www.stata.com/support/faqs/
- Video Tutorials: http://www.stata.com/links/video-tutorials/
- Stata NetCourses: http://www.stata.com/netcourse/
- Statlist: The Stata Forum (http://www.statalist.org/) and Searchable Archive (http://www.stata.com/statalist/archive/)
  - Advice on how to post: http://www.statalist.org/forums/help#howto
- UCLA IDRE Resources: http://www.ats.ucla.edu/stat/stata/
- SSCC Articles: http://www.ssc.wisc.edu/sscc/pubs/stat.htm

Getting Help with Stata

Printed Materials:

- Long, J. Scott. 2009. *The Workflow of Data Analysis Using Stata*. College Station, TX: Stata Press.
- Mitchell, Michael N. 2010. *Data Management Using Stata: A Practical Handbook*. College Station, TX: Stata Press.
- Cameron, A. Colin and Trivedi, Pravin K. 2010. *Microeconometrics Using Stata*, Revised Edition. College Station, TX: Stata Press.
- Acock, Alan C. 2012. *A Gentle Introduction to Stata*, Revised 3rd ed. College Station, TX: Stata Press.
- Mitchell, Michael N. 2012. *A Visual Guide to Stata Graphics*. College Station, TX: Stata Press.

Example Data

The Chronicle of Higher Education College Completion[1]

- Includes data for 3,800 degree-granting institutions.
- Key variables:
  - **chronname**: institution name
  - **level**: level of institution (4-year, 2-year)
  - **control**: control of institution (public, private not-for-profit, private for-profit)
  - **grad_100_value** and **grad_150_value**: percentage of first-time, full-time, degree-seeking undergraduates who complete a degree or certificate program within 100% or 150% of expected time, respectively (e.g., bachelor's-seeking group at 4-year institutions)
  - **student_count**: total number of undergraduates in 2010
  - **med_sat_value**: median estimated SAT value for incoming students
  - **aid_value**: avg amount of student aid going to undergrad recipients
  - **endow_value**: end-of-year endowment per full-time equivalent student
  - **pell_value**: percentage of undergraduates receiving a Pell Grant

[1] Supported by the Bill & Melinda Gates Foundation

Example Data

To the do-file! Where we'll learn...

- how to read in and save data,
- how to browse and edit data,
- the basics of data management,
- how to run preliminary descriptive statistics, and
- introductory graphics using Stata.

Thank you!

Thanks for attending today!

For help and advice with your data analysis, contact the StatLab to set up an appointment: statlab@virginia.edu

Sign up for more workshops or download past workshop materials: http://data.library.virginia.edu/statlab/

Register for the Research Data Services newsletter to stay up-to-date on StatLab events and resources: http://data.library.virginia.edu/newsletters/