STATA WORSHOP SERIES

Introduction to Stata

Jin Chen

Indiana Statistical Consulting Center

Feb4, 2012

Outline of the Workshop

- The Basics of Stata
  - The Stata software
  - The Stata interface
  - The Stata files and language
- Data Management
  - Data loading
  - Data exploration
  - Data management

The Basics of Stata

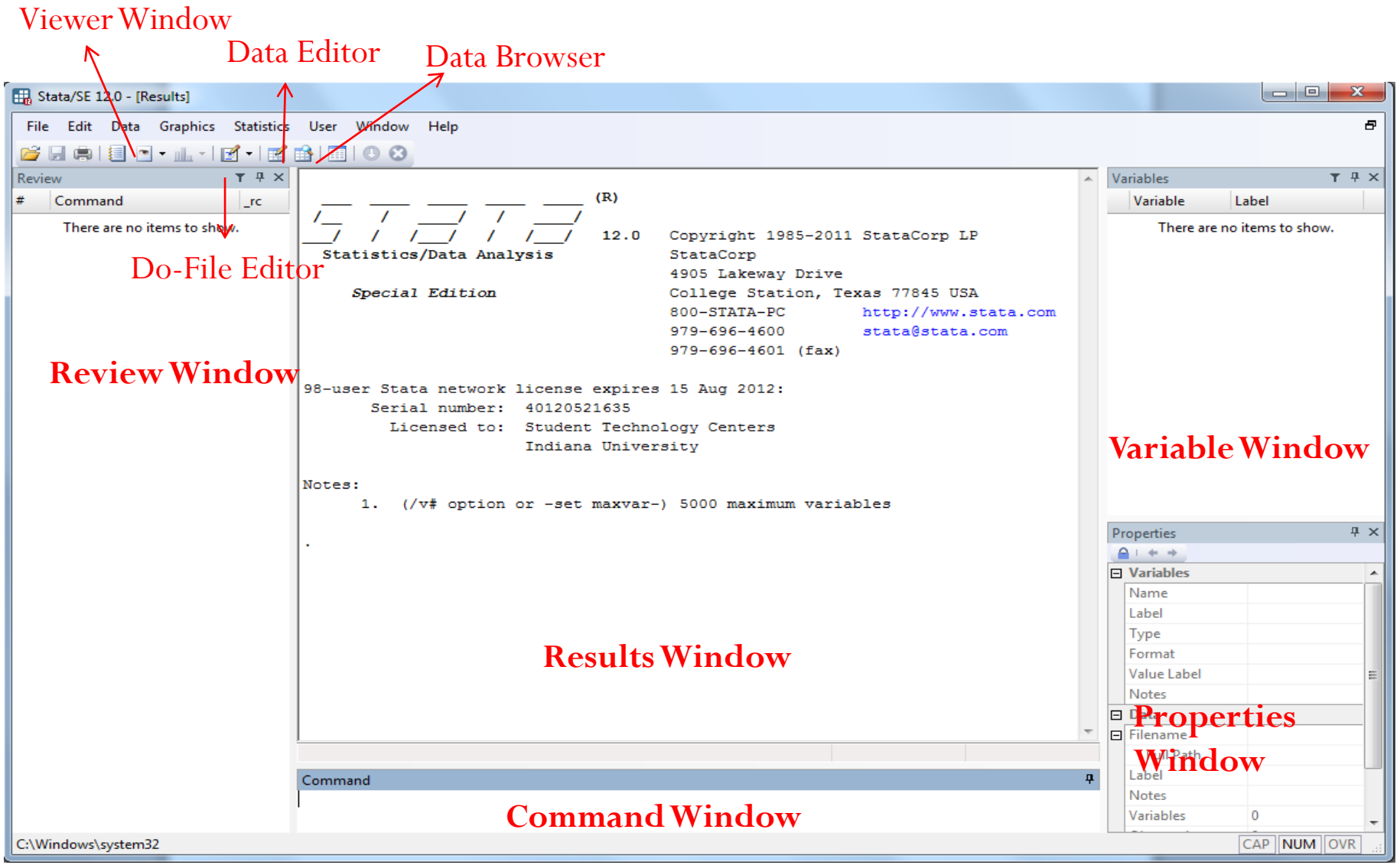
- The Stata software
  - A powerful statistical package widely used for managing, analyzing and visualizing data across social science disciplines such as economics, sociology, and political science.
  - Access to Stata on campus
    - Student Technology Centers (STCs) labs (PC) and central shared computing systems (Quarry)
  - Purchase Stata
    - SE vs. IC vs. SM
    - Perpetual license vs. annual license
    - Campus Pickup Gradplan http://www.stata.com/order/new/edu/gradplan.html
    - Contact Stat/Math Center for more detailed information

Compare Features across Stata Packages

Package	Max. no. of variables	Max. no. of right-hand variables	Max. no. of observations	64-bit version available?	Fastest: designed for parallel processing?	Platforms
Stata/SE	32,767	10,998	unlimited*	Yes	No	Windows, Mac, or Unix
Stata/IC	2,047	798	unlimited*	Yes	No	Windows, Mac, or Unix
Small Stata	99	99	1,200	Yes	No	Windows, Mac, or Unix

See http://www.stata.com/products/whichstata.html

The Stata Interface



Menus vs. Commands

- Stata operates either via menus or via the command window
- Commands are typically faster for often-used commands
- Menus are faster for complex commands, such as graphs
- We will use commands for most part of the workshop

Speaking Stata

- The Stata Files
  - Data file: *flilename*.dta
  - Do file: *filename*.do
  - Log file: *filename*.smcl (default) or *filename*.log
  - Ado file: *filename.ado*
- The Stata *Language*
  - Command is very intuitive
  - The basic language syntax:

```
[prefix:]command[varlist][=exp][if][in][weight]
                        [using filename][,option]
```

Update Stata

- Update official Stata packages (on your own machine)
  - update query
  - update all
  - help whatsnew
- Find and install user-written packages
  - findit *packagename*
  - ssc install *packagename*
- Update user-written packages
  - adoupdate

Getting Help

- Get help on a particular command
  - In command window type: **help *commandname***
- Obtain all references to a topic
  - Type: **search *commandname***
- If you don't know the command name
  - Click *help* menu→ Stata manual
- Statalist

Data Management

- Keep track of what you are doing and be able to replicate your results
  - command log file
    - cmdlog using *filename/path*
    - cmdlog close
  - do file
    - type doedit or click on icon for new do-file editor
  - log file
    - log using *filename/path*
    - log close
  - comments (use in do file)
    - * at the beginning of the line, the entire line as comments
    - /* comments */
    - // comments
    - /// join the next line, often used for long command lines

- Some to-dos before loading your data
  - Clear everything in memory: **clear**
  - (Allocate memory for your data: **set mem 50m**)
  - Set working directory
    - Check current working directory: **cd**
    - Change directory: **cd** ***"directorypath"***
  - If you are using a university computer and you would like to apply a user-written package
    - **sysdir**
    - **sysdir set PLUS "*directorypath*"**
    - **net set ado** ***"directorypath"***
  - Let's try these steps in Stata

A Do File Template (Long, 2009)

```
capture log close
log using NAMEOFDOFILE, replace text

// program: NAMEOFDOFILE
// task:
// project:
// author:

version 12
clear all
macro drop _all
set linesize 80
set more off

// #1
 // Describe step

log close
exit
```

Data Loading

- Loading your data
  - List all data available in Stata: sysuse dir
  - Use data shipped with Stata : sysuse *dataname*
  - Save data in the working directory: save *dataname* [, replace]
  - Reload data from your working directory:

    use *dataname* [, clear]
  - Read and add notes to data

    note

    note: *yournotes*

Exercise 1

- Load auto.dta from Stata and use log file to keep track of the commands as well as results
- Save a copy of the data, named auto2, to your working directory ; and then reload data from your working directory
- Add note “this is fun” to auto2

Alternative Ways of Importing and Exporting data

- Loading/saving your data from/to user-defined sources and formats
  - See insheet and outsheet, infile and outfile, import excel and export excel
  - Use dropdown menu: File→import/export
  - Copy and paste your data into data editor
  - Use StatTransfer

Data Exploration

- Exploring your data file
  - codebook [*varlist]* *[*, compact]
    - Returns the variable name, type, range, number of unique values, number of missing values, and summary stats
  - describe [*varlist]*
    - Returns variable names, types and labels
  - *summarize [varlist] [, detail]*
    - *Returns summary statistics such as number of observations, mean, standard deviation, minimum and maximum*
  - list [*varlist]* [in] [if]
    - returns all observations and variables in the data file, use in and if to list only a subset of the dataset

- Qualifiers and operators
  - Specify the range of observations: *command* in *range*
    - e.g. list make mpg in 2

      list make mpg in 1/10

      list make mpg in -10/-1
  - Specify the conditions: *command* if *exp*
    - e.g. list make mpg if mpg>25

      list make mpg if mpg>=25 & mpg<30

      list make mpg if mpg>25 | mpg<10

      sum price if foreign==1 & rep78!=1

Exercise 2

- Generate a *codebook* for all variables in auto2
- Compare results by *codebook* and *describe*
- Produce summary statistics for all (numeric) variables
- Obtain summary statistics for price mpg rep78, with detailed info on the distributions
- List the repair record in 1978 (rep78) for the 10th observation, the first 10 observations, and all observations that has fewer than 2 repair records

Data Manipulation

- Variables
  - Generate new variables
    - generate *newvar=exp [if] [in]*
    - egen *newvar=fcn(argument)*
      - *e.g. mean, max, min, median, sd, std*
  - Recode old variables
    - replace *oldvar=exp [if] [in]*
    - recode *varlist (rule) ,[, gen(newvar)]*
  - Rename variables
    - rename *oldvar newvar*
    - rename (oldvarlist) (newvarlist)
  - Caution: missing values denoted as . → infinity

Exercise 3

- Generate a new variable (price2) indicating car price in thousands
- Generate a new variable (pmean1) whose values are mean prices across all observations
- Generate a new variable (pmean2) representing average price for each car type (foreign)
- Recode rep78 into a new variable rep2, where missing values (.) are coded as zeros, 1-3 are coded as 1, and 4 and beyond are coded as 2.
- Generate rep3, equal to rep78; then use *replace* to replicate the values in rep2.
- Rename price2 to be price1000

- Labels
  - Label dataset
    - label data "*datalabels*"
  - Variable labels
    - label variable *varname* "*varlabel*"
  - Value labels
    - label define *lbname* #"*label*"[#"*label*"],[*add modify replace*]
  - Attach a value label to a variable
    - label values *varname lbname*
  - Check current labels
    - label dir
    - label list *lbname*
    - label list
    - label drop *lbname*

Exercise 4

- Label dataset auto2 as "Sample data for Stata Workshop"
- Label variable rep2 as "recoded repair record"
- Create and attach a value label (repcat) that denotes three levels of repair frequency, namely, none, low frequency, and high frequency, to variable rep2.
- Use *label list lbname* to check the value label you created

- Ordering of variables or observations
  - order *varlist*, [last][before(varname)]
  - sort *varlist*
  - gsort *varlist*
- Keep/drop variables or observations
  - keep/drop *varlist*
  - keep/drop [if] [in]

Exercise 5

```
order make rep78
order make, last
order make, before(rep78)
sort rep78
gsort foreign –rep78
drop rep3
tab rep2, m
keep if rep2>0
tab rep2, m
```

Preview of Stata Workshop Part II

- Statistics
  - Descriptive statistics
  - Inferential statistics
- Data visualizations
- Automation (if time permits)
  - Macros
  - Loops

Questions?

- Contact US

chen92@umail.iu.edu

iscc@indiana.iu.edu

410 N Park Ave, Rm 202

Bloomington, IN 47408

Tel: 812-855-8526