

ECON 326: Economics of Developing Countries

TA Session 1

Vaidehi Parameswaran (Northwestern Econ)

(Adapted from Sebastian Sardon's slides)

January 2025

Today

- ▶ Introductions
- ▶ Stata:
 1. Stata basics
 2. Data management
 3. Data visualisation
 4. Data analysis (OLS, Binary variables)

Introductions

- ▶ Me: a second-year grad student in the Econ department
 - ▷ I hope to study labour markets, monopsony power in India
 - ▷ Email: vaidehiparameswaran2029@u.northwestern.edu
 - ▷ Office hours: Tuesday 12:30-1:30pm, Friday 1:30-3pm

Stata Basics

Why Stata?

Advantages of Stata:

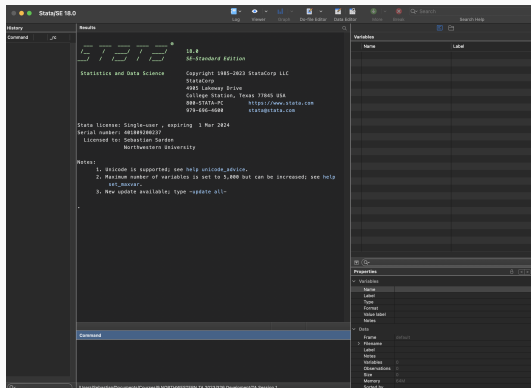
- ▶ **Simple:** you won't need to spend much time updating it, installing packages, or debugging your code (relative to, e.g., R and Python)
- ▶ **Well documented:** you can figure out how any command works writing `help` (or just `h`) into the prompt, followed by the command's name
- ▶ **Widely used:** Stata has a lot of built-in commands for the kind of econometrics you'll be doing in this class and is widely used in applied micro research

Disadvantages of Stata:

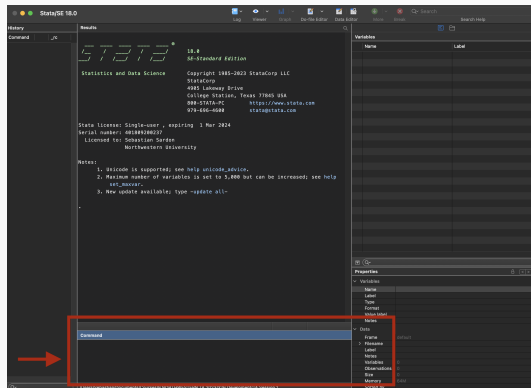
- ▶ **Cost:** Stata is not free, but you can access it through the university
- ▶ **Not open source:** Needs a license, can't see the source code
- ▶ **Limited for tasks like ML:** If you're interested in machine learning, you might want to learn R or Python

The Interface

- Open Stata to see the **Results Window**.

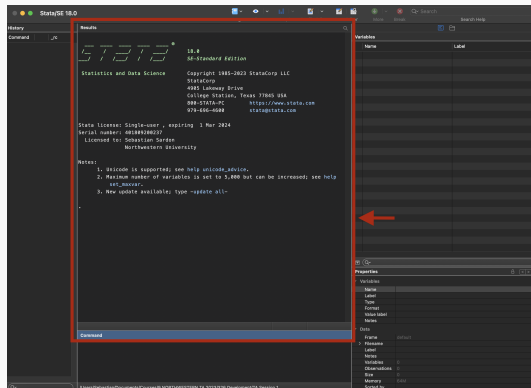


The Interface



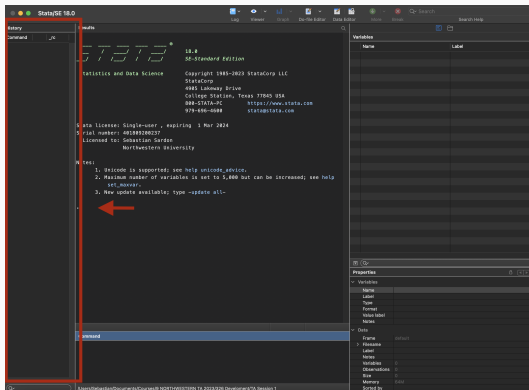
The **Command Prompt** allows you to type and run commands (try “di” followed by what you want Stata to display).

The Interface



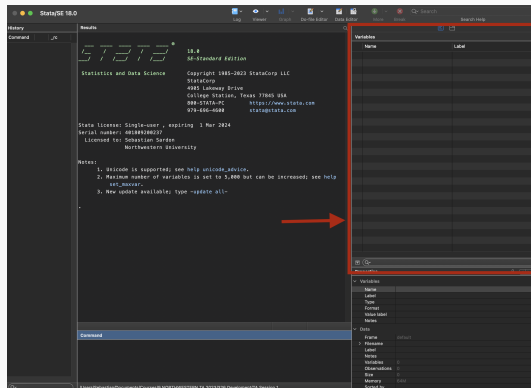
The **Results** panel will contain the output of all executed commands. Right-click it to clear it every once in a while.

The Interface



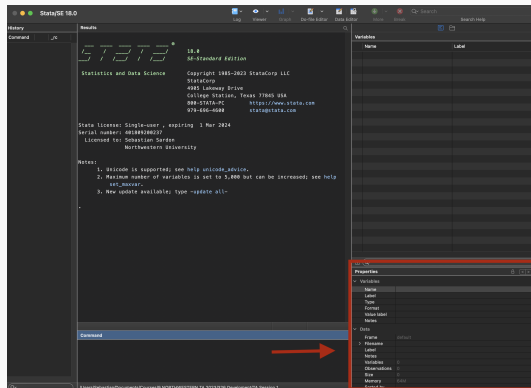
The **History** panel will list all commands sent to the prompt.

The Interface



The **Variables** panel will show you all variables in the dataset that's currently open. Typically in Stata, you can only have one dataset open at once (a drawback).

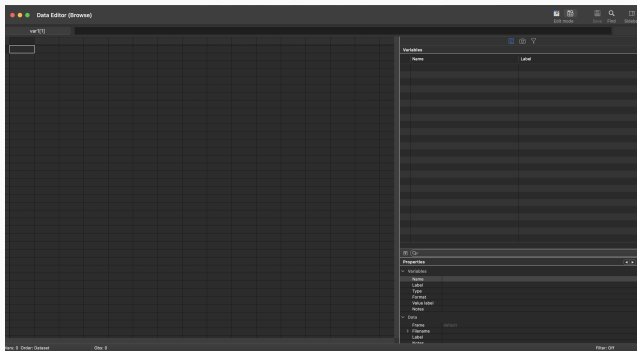
The Interface



Finally, the **Properties** panel will show additional information about the currently open dataset. Be careful with how much **Memory** is being used.

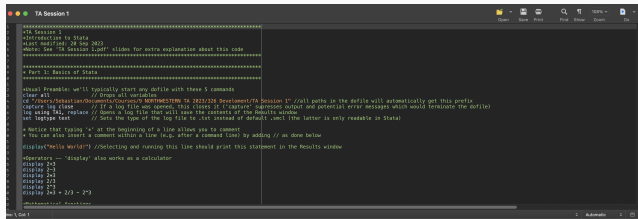
The Interface

- ▶ There's also a **Data Editor** window that allows you to visualize the currently open dataset
- ▶ To open it, send the `br` command to the prompt



The Interface

- ▶ Finally, there's the **Do-File Editor**, where we will do most of our work
- ▶ It's a highly optimized text editor for writing Stata code
- ▶ You open it by opening any .do file, or in the menu bar via **File**→ **New** → **Do-file**.
- ▶ You can select the lines you want to run, then run them with **Shift**+**CMD**+**D** (Mac) or **CTRL**+**D** (Windows)
- ▶ Or use those same shortcuts without selecting any lines to run the entire do-file at once



```
1 *****
2 *TA Session 1
3 *Introduction to Stata
4 *last modified: 28 Sep 2022
5 *where: See "TA Session 1.pdf" slides for extra explanation about this code
6 *****
7 * Part 1: Basics of Stata
8 *****
9 *usual preamble: we'll typically start any dofile with these 5 commands
10 *clear all // drops all variables
11 cd "/Users/Sebastian/Documents/Courses/TA 2022/2022 Development/TA Session 1/" //all paths in the dofile will automatically get this prefix
12 capture log close // If a log file was opened, this closes it. (Captures "success" output and potential error messages which would terminate the dofile)
13 log using TA1, replace // Opens a log file that will save the contents of the Results window
14 *set logtype text // Sets the type of the log file to .txt instead of default .smcl (the latter is only readable in Stata)
15
16 * a notice that typing "*" at the beginning of a line allows you to comment
17 * You can also insert a comment within a line (e.g. after a command line) by adding // as done below
18
19 display("Hello World!") //Selecting and running this line should print this statement in the Results window
20
21 *Operators -- "display" also works as a calculator:
22 display 2+3
23 display 1-3
24 display 2*3
25 display 2/3
26 display 2^3
27 display 2+3 = 2/3 - 2^3
28
29 *****end*****
30
31 TA1.do:1
```

Part 1: Basics of Stata

- Usual Preamble: typically start any dofile with these 5 commands

1. `clear all` → Drops all variables

2. `cd`

`"/Users/vaidehiparameswaran/Desktop/second-year-coursework/econ326/"`
→ Change Directory: this indicates where you will open from and save your files. All paths in the dofile will automatically get this prefix.

3. `capture log close` → **If** a log file was opened, this closes it. `capture` suppresses output and potential error messages which would terminate the dofile. Here, we'd get an error the first time we run the dofile: no log would be open.

4. `log using TA1, replace` → Opens a log file (called **TA1**) that will record the contents of the Results window. This lets others verify your code actually worked.

5. `set logtype text` → By default, log files are only readable in Stata. This converts your log file to a text file that's readable with any text editor.

Part 1: Basics of Stata

- ▶ Commenting your code: this means writing notes that won't run as commands, useful to annotate dofile and divide into sections
- ▶ Two ways to comment your code:
 - ▷ Typing `*` at the beginning of a line allows you to comment
 - ▷ You can also insert a comment after a command line by adding `//` after the command
 - ▷ Example:

```
display("Hello World!") //This line prints statement in results window
```

- ▶ Operators: `display` (or equivalently, `di`) also works as a calculator
 - ▷ `display 2*3 + 3/2 - 2^3` → This will give you -0.5. Notice Stata follows the standard order of operations so you don't need brackets unless you want to change their order.
- ▶ Mathematical functions
 - ▷ `display ln(sqrt(abs(-2)))` → This will give you 0.34657359, the result of $\log(\sqrt{|-2|})$.

Part 1: Basics of Stata

- ▶ String Functions: **string variables** are those containing letters (and other characters) instead of numbers (e.g. country names). You use string functions to manipulate them:
 - ▷ `display substr("abc",1,2)` → This tells Stata that, starting from position 1, in the string "abc", i.e. starting from "a", it should keep 2 characters. The result would be "ab".
 - ▷ `display subinstr("abc","b","X",1)` → This tells Stata to replace a substring ("b") within a string ("abc") with another substring ("X"). The last argument, "1", indicates how many times such replacements are going to be made. Here "b" only occurs once within the string so this does not matter. The result would be "aXb".
 - ▷ If we replace the "1" from above with a period, we are telling Stata to do the replacement as many times as possible. For example: `display subinstr("abcb", "b", "X", .)` yields "aXcXX".

Part 2: Data Management

- ▶ We will work with NLS data on wage and education.

- ▶ To begin, run the preamble commands.

- ▶ Then open the data:

```
use "rawdata.dta", clear
```

- ▶ The `use` command opens the data set from the directory you set earlier in the preamble with `cd`.

- ▶ `clear` is an option that specifies that it is okay to replace the dataset in memory. You need to include this to avoid errors in case you already had a dataset loaded into Stata.

- ▶ **In general, a comma will separate a Stata command and its options.**

Part 2: Data Management – Understanding the data

- ▶ `describe` → Overview of all your variables (types, formats and labels)
- ▶ `codebook` → This prints the codebook of your data with a description of each variable
- ▶ `browse` or `br` → See and navigate dataset as if it was an Excel spreadsheet
- ▶ `count` → Shows you the total number of observations in your data.

Part 2: Data Management – Understanding the data

- ▶ `tabulate x` → This command gives you a frequency table for the variable specified (`africa`).
- ▶ `tabulate x y` → If you want a 2 by 2 frequency table. The first variable will be displayed in the rows and the second in columns.
- ▶ `sum x` → summarize statistics of the variable `x` (short version: number of observations, mean, standard deviation, min, max)
- ▶ `sum x, d` → detailed version of `sum`, useful to inspect the distribution of a variable

Part 2: Data Management – Generating and manipulating variables

- ▶ `gen lwage = ln (wage)` → The `gen` command generates new variables.
- ▶ `drop lwage` → The `drop` command deletes variables.
- ▶ `gen lwage = ln(wage) if collgrad == 1`
- ▶ `replace lwage = ln(wage) if collgrad == 0`
- ▶ `label var lwage "log wages"` → The `label var` command labels variables.

Part 2: Data Management – Saving our work

- ▶ `cap mkdir output` → create a folder (directory) called `output` – this goes inside the folder we are working within as set with `cd` above
- ▶ `save "output/nls88.dta", replace` → Save the modified dataset; `replace` tells Stata to overwrite its previous version (if any)

Part 3: Data Visualization – Scatter Plots

- ▶ Suppose we want to create a scatterplot of hourly wages and total experience.
- ▶ This is easy to do with the command `twoway scatter wage ttl_exp`

Part 3: Data Visualization – Other Graphs

- ▶ `hist X` → histogram to visualize the distribution of a variable
- ▶ `kdensity X` → estimate probability density function of a variable

- ▶ Suppose we want to estimate the linear regression model relating wage with education, age, and experience.

$$\log lwage_i = \alpha + \beta_1 educ_i + \beta_2 age_i + \beta_3 exp_i + \mathbf{X}_i' \gamma + \epsilon_i$$

- ▶ \mathbf{X}_i is a vector of controls (ignore for now)



$$\log lwage_i = \alpha + \beta_1 educ_i + \beta_2 age_i + \beta_3 exp_i + \mathbf{X}_i' \gamma + \epsilon_i$$

- ▶ We want to estimate β_1 : the effect of education on wages
- ▶ We can estimate α with the ordinary least squares (OLS) estimator as follows

```
reg lwage educ age ttl_exp
```

- ▶ Additional controls (vector \mathbf{X}) can be added. For example,

```
reg lwage educ age ttl_exp tenure industry
```

Part 4: Data Analysis – OLS

- By running the simple regression command `reg lwage educ age ttl_exp` we get:

```
. reg lwage educ age ttl_exp
```

Source	SS	df	MS	Number of obs	=	2,244
Model	200.836661	3	66.9455538	F(3, 2240)	=	277.64
Residual	540.126138	2,240	.24112774	Prob > F	=	0.0000
Total	740.962799	2,243	.330344538	R-squared	=	0.2710
				Adj R-squared	=	0.2701
				Root MSE	=	.49105

lwage	Coefficient	Std. err.	t	P> t	[95% conf. interval]
educ	.0798886	.004203	19.01	0.000	.0716384 .0881228
age	-.009497	.00342	-2.75	0.006	-.0162837 -.0027093
ttl_exp	.0401031	.0023148	17.35	0.000	.0356238 .0447025
_cons	.6909684	.1452393	4.75	0.000	.4061588 .975786

- We also get
 - ▷ The OLS estimator's standard error (square root of its variance, SE)
 - ▷ Its t -statistic (ratio of coefficient and SE)
 - ▷ Its p -value (Probability of obtaining an estimate at least as big as ours, and with the same sign, in a world where $\beta_1 = 0$ (i.e., “under the null hypothesis”))
 - ▷ A confidence interval where the true value of β_1 lies with 95% probability
 - ▷ An estimate of the intercept α (see the `_cons` statistic below the highlighted one)

Part 4: Data Analysis – OLS

- By running the simple regression command `reg lwage educ age ttl_exp` we get:

```
. reg lwage educ age ttl_exp
```

Source	SS	df	MS	Number of obs	=	2,244
Model	200.836661	3	66.9455538	F(3, 2240)	=	277.64
Residual	540.126138	2,240	.24112774	Prob > F	=	0.0000
Total	740.962799	2,243	.330344538	R-squared	=	0.2710
				Adj R-squared	=	0.2701
				Root MSE	=	.49105

	lwage	Coefficient	Std. err.	t	P> t	[95% conf. interval]
educ		.0798806	.004203	19.01	0.000	.0716384 .0881228
age		-.009497	.00342	-2.78	0.006	-.0162837 -.0027903
ttl_exp		.0401631	.0023148	17.35	0.000	.0356238 .0447025
_cons		.6909684	.1452393	4.76	0.000	.4061508 .975786

- In addition, Stata reports the regression's R^2
- This indicates the share of the variation in `lwage` that is explained by `educ`
- But correlation \neq causation.

Part 4: Data Analysis – Aside on Binary Variables

- ▶ Binary variables (that take only two values, for e.g., 0 or 1) are common in economics
- ▶ Regressing a dependent variable on a binary variable gives us the average difference in the dependent variable between the two groups
- ▶ To see this, let's regress the log of wages on a binary variable indicating whether the individual is a college graduate
- ▶ We can do this with the command `reg lwage collgrad`
- ▶ Now let's also get the mean wages for college graduates and non-college graduates

See you next time!