

# Introduction to Stata

Training Workshop on the  
Commitment to Equity Methodology  
CEQ Institute and The Ministry of Finance

Accra

February 7-10, 2017

# What is Stata?

- A programming language to do statistics
- Strongly influenced by economists
- Open source, sort of
  - You can see how Stata codes many of its commands
  - You can add your own commands to Stata
  - You can publish commands for others to use
- An acceptable way to manage data

# How Stata Works

- You can work interactively, through a user interface
- For serious work, it is much better to write programs
  - Stata calls these “do files”
  - Allows reproduction of your results
  - Allows identification and rectification of errors
  - We still run these “do files” interactively, mostly
- Stata does everything in RAM, except reading data (from other places) and saving data (to other places)
  - To do anything, you must load data into RAM

# Where Does Your Computer Store “Data” (Stuff)?

- RAM (random access memory)
  - very fast
  - “forgets” what it had when the power goes out
- Disks (hard drives, usb drives, dvd’s, etc)
  - Much slower
  - But stable – they remember what’s recorded on them with the power goes out
- Internet (“the cloud”, file servers, etc)
  - Slower still
  - But vast
- To work in Stata, you must “load” or “read” data stored on a disk or the internet into the RAM

# Three Topics for Today

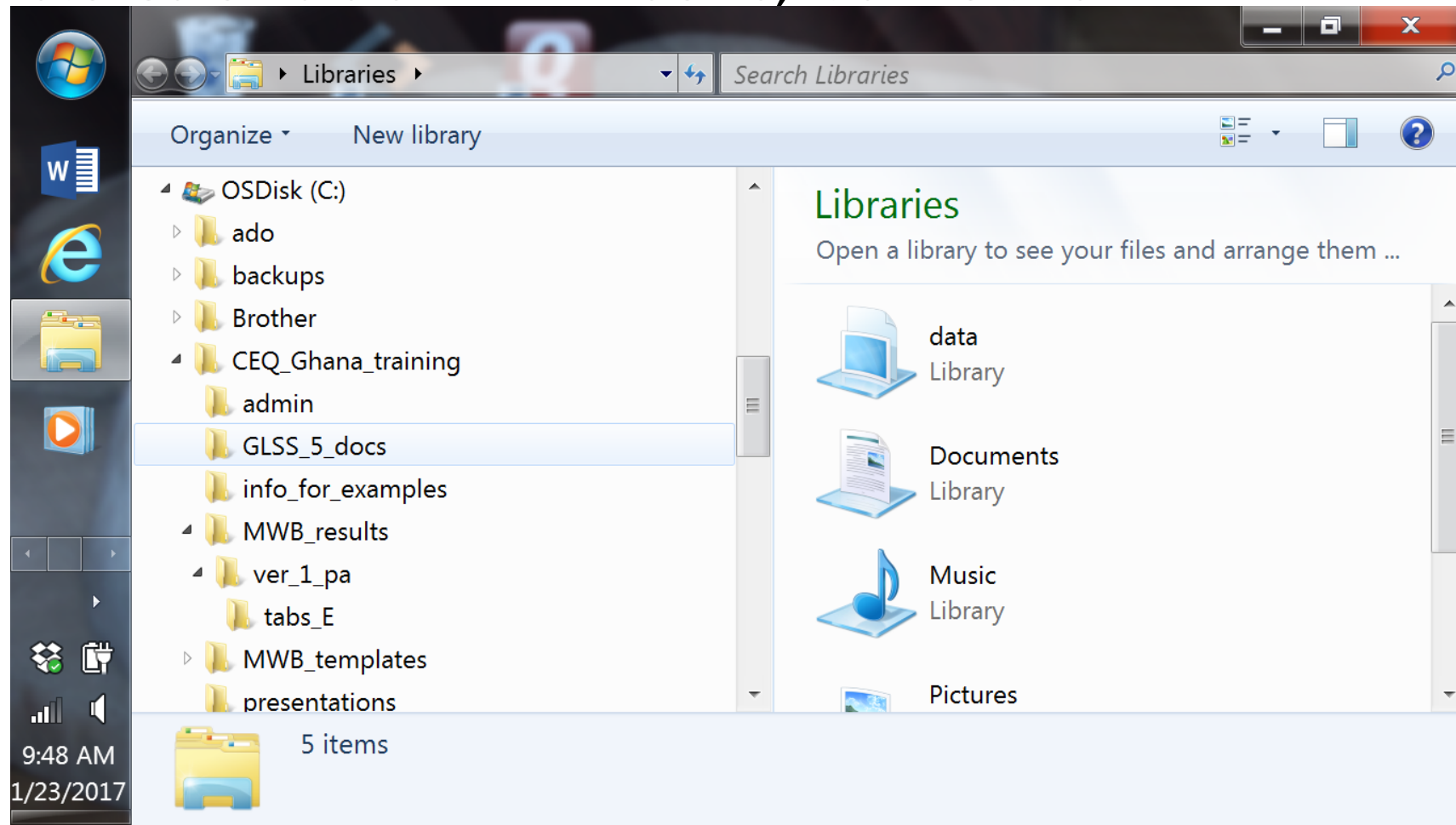
- Statistics
  - Stata is really good at generating sophisticated statistical analyses
  - But we will mostly skip this
- Data management
  - This is about getting data into the RAM (and Stata) and then getting (other) data out to permanent storage
  - And about manipulating data – creating new variables, modifying existing ones
  - And about manipulating *datasets*
    - Mostly, merging two different datasets
- Programming
  - How to keep a permanent record of what you're doing
  - How to manipulate data efficiently

# How to Load Data into Stata

- Many options
  - Type them in by hand (ugh)
  - Manually copy from Excel or Word or a text processor and paste into Stata's data editor window
  - Read them from a comma- or space-delimited file
  - Read them from an Excel spreadsheet
  - Read them from a Stata dataset (Stata extension is .dta)
- We will focus on the last two, which are the most common
- Stata has commands for each
  - use <dataset path and filename> for Stata datasets
  - Import excel <spreadsheet path and filename>

# An Aside on (Sub-)Directories and Paths

- Windows (or Mac OS) must organize its many files stored on a disk. In Windows, it's like this:



## An Aside on (Sub-)Directories and Paths

- The sub-directories, or folders, help you (and Windows) keep files organized
- To read a file, you need to tell Stata where it is
  - Requires a path (subdirectory) ...
  - ... and filename
- For example:
  - use `c:\CEQ_Ghana_training\stata\data\GSS_vars.dta`
    - This is a Stata command to read the Stata dataset GSS\_vars.dta into the RAM so Stata can work on it
  - import excel using "c:\CEQ\_Ghana\_training\info\_for\_examples\inc\_dist.xlsx", sheet("Gini") cellrange(B4:F15) firstrow
    - This is a Stata command to read part of an Excel spreadsheet



## Loading Data from Excel – Let’s Try It

- First, check that the Excel file is on your disk:
  - `dir "c:\CEQ_Ghana_training\info_for_examples\"`
- Now go look at that spreadsheet (with Excel)
- Import the data:
  - import excel using "c:  
  \CEQ\_Ghana\_training\info\_for\_examples\small\_data.xlsx"  
  , sheet("HH\_1") cellrange(A3:E8) firstrow
- See what you imported:
  - `list *`, `clean`
  - `describe`
- Put some labels on the variables
  - label var hhid "Unique household id"
  - etc

# Labeling and Saving Data

- Label the dataset:
  - label data “Practice dataset #1, household data”
- Sort the data:
  - important for us to be able to merge later
  - sort hhid
- Describe the data again
- Save the data:
  - first, check the default (sub)directory:
    - pwd
  - now save:
    - save “c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_1”
- And load the data again (now a Stata dataset)
  - use “c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_1”

# Manipulating Data

- Create a new variable, income per capita
  - generate `income_pc = income/hhsize`
  - label var `income_pc` “HH Income per capita”
- Create a new variable, conditional on some criterion
  - generate `income_pa = income/hhsize if income>500`
  - list what you got
  - generate `poor = (income_pc<500)`
  - list what you got
- Label the values of a variable
  - label define `poorstatus 0 “Poor” 1 “Not Poor”`
  - label values `poor poorstatus`
- Save the data again ...

## Structure of (Almost) All Stata Commands

- verb variable(s) <if ...> [weights], options
- verb is the command
- variable(s) are the variables to operate on
- if ... is to subset the command to only some observations
- [weights] are to apply different weight to each observation
  - Stata has several types of weighting schemes
- options are command-specific, and always come after a comma, at the end

# Merging Data

- Load and save the data in tabs HH\_1 and HH\_2
  - import excel using "c:\CEQ\_Ghana\_training\info\_for\_examples\small\_data.xlsx", sheet("HH\_1") cellrange(A3:E8) firstrow
  - save "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_1"
  - import excel using "c:\CEQ\_Ghana\_training\info\_for\_examples\small\_data.xlsx", sheet("HH\_2") cellrange(A3:C8) firstrow
  - save "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_2"
  - use "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_1", clear
- Merge the data
  - merge using "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_2"
  - oops
  - merge hhid using "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_2"

# Merging Data

- Load the second dataset again
  - use "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_2", clear
  - sort hhid
  - save "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_2", replace
- Now load the first dataset again
  - use "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_1", clear
  - merge hhid using "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_2"
- Check results
  - list, clean
  - desc

## Merging Data: one-to-one and n-to-one

- Some datasets may have multiple observations for each unique observation in another dataset
  - household-level data vs. individual-level data
  - Stata will do an “n-to-one” or “one-to-n” merge here
- Load the individual-level data from the spreadsheet
  - import excel using "c:  
  \CEQ\_Ghana\_training\info\_for\_examples\small\_data.xlsx",  
  sheet("indiv") cellrange(A3:D20) firstrow
  - sort hhid pid
  - save "c:\CEQ\_Ghana\_training\info\_for\_examples\indiv"
- Now merge in household data
  - merge n:1 hhid using "c:\CEQ\_Ghana\_training\info\_for\_examples\HH\_1"

## Programming – Writing “do files”

- It is very bad form to do research with interactive or point-and-click commands
- Programs (do files):
  - keep a record of what you have done
  - allow you (and others) to cross-check your work
  - make it very easy to make small changes to your research
- Let’s write a do-file to:
  - read all the data, both HH and individual, in the spreadsheet
  - clean the error in hhid
  - merge them together
  - create HH income per capita and per adult equivalent
  - tabulate average HH income per capita by area of residence



## Programming – Writing “do files”

- Stata has an internal text editor, like a word processor
  - start it with ctrl-9, or the “window” menu
- Enter the commands we have learned, in order
- run them: ctrl-D or the “tools” menu
- Add comments
  - very important for good programming
  - help you remember what you are doing
- Locals and globals – place-keepers
  - for example, use a global for the path
  - or a local for a specific value
  - or a local with a list of variables

# Programming – Writing “do files”

- Looping with foreach
  - foreach <local> in <list> {
    - ... do something to every item in the list ...
    - }

For example:

```
foreach nn in hhscale eqscale {  
    generate income_`nn' = income/`nn'  
}
```

Or:

```
local namelist " hhscale eqscale "  
foreach nn of local namelist {  
    generate income_`nn' = income/`nn'  
}
```

# Exercise

- Write a do-file to:
  - read and merge all the data, both HH and individual, in the spreadsheet
  - clean the error in hhid
  - merge them together
  - allow adding an arbitrary value to HH income *if the HH is rural*
  - create HH income per capita and per adult equivalent
  - tabulate average HH income per capita by area of residence
  - find out how many secondary graduates there are per HH
- Comment it nicely
- Use a local and a global
- Use a loop when you can