

Using Stata Effectively

Zane Mokhiber
Economic Policy Institute

October 1, 2021
EARNCon 2021

Motivation

My goal is to teach you how to analyze microdata effectively and efficiently

- ▶ Allows you to answer questions you might not be able to otherwise answer using published data
 - ▶ ex: hourly wages by race and sex in a specific state
- ▶ Emphasis on file management and reproducibility
 - ▶ Analysis can be easily replicated by others (including future you)
 - ▶ Code/scripts are easily modified and tweaked, without re-doing everything

Overview

- ▶ Writing do-files in Stata using best practices and proper documentation
- ▶ Intermediate Stata operations: joining datasets, transforming data, macros, loops, exporting data, pooling data
- ▶ How to properly set up a project: directory structure, working directories, and storing raw data
- ▶ BONUS: Use EPI Stata data resources!

Example final product



Best practice: write do files

- ▶ Instead of typing commands in the command window, we can write them in a script, which stata calls a “do file”
- ▶ it's just a plain text file with the extension “.do”
- ▶ Why do we write do-files?
 - ▶ Your do-file is a fully documented record of the entire analysis
 - ▶ Your work is now easy to reproduce and much easier to update
 - ▶ It is much easier to spot mistakes and make improvements to code

Preamble and comments

- ▶ Always document what your do-file does
 - ▶ other people may need to know
 - ▶ future you will definitely forget

```
* File: earn_data_bootcamp.do  
* Desc: compare wages by race and sex in Ohio using the CPS  
* Auth: Zane Mokhiber
```

- ▶ Stata ignores comments or text after a * at the beginning of a line
 - ▶ use comments to explain clearly what you're doing
- ▶ Comment blocks are also useful

```
/* this is a comment  
and so is this  
these words will be ignored by Stata */
```

Preamble continued

Always put

```
set more off  
clear all
```

at the beginning of your do file

- ▶ Useful to remove “more” prompts and start with a fresh workspace
- ▶ Make sure the working directory is set properly
 - ▶ however, it is bad practice to include `cd` in any do file

Analysis from session 1

```
*load 2020 CPS ORG  
use epi_cpsorg_2019, clear  
*Create indicator variable for Ohio  
generate oh = 0  
replace oh = 1 if statefip == 39  
* age restriction  
keep if age >= 16  
* Ohio only  
keep if oh == 1  
*calculate avg wages by race and sex  
collapse (mean) wage [aw=orgwgt], by(wbho female)
```


Transforming data: Reshape

- ▶ In order to do some calculations on the data, we need to reshape the data
 - ▶ Our data is in “long” format: there is one value variable and two categorical variables
 - ▶ We want to reshape it to a “wide” format so values can be added or subtracted from each other

```
reshape wide wage, i(female) j(wbho)
* rename reshaped variables
rename wage1 white
rename wage2 black
rename wage3 hispanic
rename wage4 other
```

Helpful article on reshape:

<https://stats.idre.ucla.edu/stata/modules/reshaping-data-wide-to-long/>

Exporting the analysis

The collapsed and reshaped data is easily exported to excel using the export command

```
export excel using ohio_wages.xlsx, ///  
    replace firstrow(variables)
```

Adding more data to our analysis

- ▶ What if we want to look at multiple years of data
- ▶ maybe the sample we are looking at isn't large enough
- ▶ want to view changes over time

Join data together using `append`

- ▶ General rule of thumb for sample size concerns
 - ▶ sample > 1000 , no problems
 - ▶ sample < 500 , you may need to take a closer look
- ▶ Use `tabulate` or `count` to investigate

Best practice: store microdata files in one central location

- ▶ It's good practice to treat your raw data as “read only”
 - ▶ raw data never changes or moves
 - ▶ helps with reproducibility
 - ▶ saves space by not duplicating data files across multiple projects
- ▶ create a “data” folder somewhere on your computer
 - ▶ ex: C:\data\cps

```
cd C:\data\cps\  
unzipfile C:\Users\zmokhiber\Downloads\epi_cpsorg_1979_2021.z  
cd C:\Users\zmokhiber\Documents\data_bootcamp
```

Macros: store stuff for later

- ▶ with macros, you can store and refer to important things later
 - ▶ two types of macros, local and global
 - ▶ we'll just deal with local macros for now
 - ▶ syntax is `local {localname} {whatever you want to store}`
 - ▶ refer to the local after it is declared with `'`

** random example*

```
local currentyear 2020
```

```
display `currentyear'
```

**do some math*

```
display `currentyear'-1
```

Macros: store stuff for later

- ▶ to use the microdata, we have to type the full file path if it's not in our working directory
 - ▶ this is tedious
 - ▶ room for error if you have to type it a bunch of times
- ▶ Store the file path in a macro
 - ▶ in my case, the CPS files are in C:\data\cps

```
local datadir C:\data\cps\  
use `datadir'epi_cpsorg_2019.dta
```

Appending data

** Load 2018-2020 CPS ORG*

```
use `datadir'epi_cpsorg_2018.dta, clear  
append using `datadir'epi_cpsorg_2019.dta  
append using `datadir'epi_cpsorg_2020.dta
```

Merge in CPI for inflation adjustments

- ▶ Download the BLS CPI-U-RS from <https://www.bls.gov/cpi/research-series/r-cpi-u-rs-home.htm>.
- ▶ Use Excel to clean up and convert to .csv file
- ▶ import into stata

```
* CPI-U-RS from  
* https://www.bls.gov/cpi/research-series/allitems  
import delimited using bls_cpiurs.csv, clear  
keep year avg  
rename avg cpiurs  
keep if cpiurs ~= .  
save cpiurs.dta, replace
```


Merge in CPI for inflation adjustments

- ▶ The merge function matches two Stata datasets on variables (columns)
- ▶ The syntax is `{stata} merge {dataset structures} {matching variables} using {using data}`
- ▶ Some Stata vocabulary
 - ▶ Your “master” data is what you currently have in memory
 - ▶ Your “using” data is what you merge onto the master data

Merge in CPI for inflation adjustments

- ▶ in this case, our master dataset is the CPS data, since it's currently what is in memory
- ▶ using data is the CPI inflation adjustment
- ▶ many to one merge, matching variable between them is year

```
merge m:1 year using cpiurs.dta
```

Inflation adjustment

- ▶ To inflation adjust the wage we calculate
- ▶ $\text{inflation-adjusted wage} = \text{wage} * \text{CPI } 2020 / \text{CPI data year}$
- ▶ In Stata use the return macro `r(mean)` to grab the 2020 CPI

```
sum cpiurs if year == 2020  
display r(mean)
```

- ▶ Now we can inflation adjust wages in the CPS data:

```
* inflation adjust wages  
sum cpiurs if year == 2020  
replace wage = wage * (r(mean) / cpiurs)
```

Exporting the analysis

After collapsing and reshaping the data, the collapsed data is easily exported to excel using the export command

```
export excel using ohio_wages_pooled_years.xlsx, ///  
replace firstrow(variables)
```

Loops: program more efficiently

Say we wanted to look at more than three years of data? * Use foreach or forvalues loop for repeated actions + saves you from typing the same code over and over

```
* load one year of data
use `datadir'epi_cpsorg_2011.dta,clear
* append years 2012-2020
forvalues year = 2012/2020{
    append using `datadir'epi_cpsorg_`year'.dta
}
* display years now available in memory
tab year
```

Pool multiple years of data with load_epiextracts

Install the command

```
net from "https://microdata.epi.org/stata"\  
net install load_epiextracts
```

Load multiple years of EPI CPS:

```
load_epiextracts, begin(2018m1) end(2020m12) sample(org) ///  
sourcedir("C:\data\cps")
```

Limit your variable selection to save memory:

```
load_epiextracts, begin(2018m1) end(2020m12) sample(org) ///  
sourcedir("C:\data\cps") ///  
keep(year orgwgt wage statefips age wbho female mind03)
```

Resources/contact info

- ▶ All files associated with this presentation can be accessed at https://economic.github.io/data_bootcamp/
- ▶ EPI CPS data resources: <https://microdata.epi.org/>
- ▶ Additional stata resources
 - ▶ Princeton intro to stata: <https://data.princeton.edu/stata>
 - ▶ UCLA learning modules
<https://stats.idre.ucla.edu/other/mult-pkg/seminars/#Stata> and here
<https://stats.idre.ucla.edu/stata/modules/>
 - ▶ Stata also has a large library of video tutorials:
<https://www.stata.com/links/video-tutorials/> and webinars:
<https://www.stata.com/training/webinar/>
 - ▶ Stata cheat sheets:
<https://www.stata.com/bookstore/statacheatsheets.pdf>
- ▶ My contact info:
 - ▶ email: zmokhiber@epi.org
 - ▶ twitter: @zanemokhiber