Stata Workshop

At MINAGRI

Roshni Khincha and Sakina Shibuya
DIME, World Bank

August, 2018

Section 1

Section 2

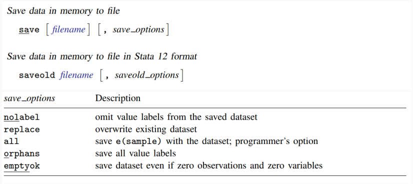
Edit data in Stata

```
use "$data\cs_s0_s5_household.dta", clear
```

Lab Task 7: Saving Stata datasets

- The command for saving a Stata dataset is ***save***.
- *save* saves your data in memory in a file format called dta. This is a file that can only be read with Stata.
- The command for saving a dataset in excel and csv is ***export***.
- ***export*** is the opposite of import, and is very versatile. It lets you save data in excel, csv, sas and others. Please refer to the help file on ***export***.

Lab Task 7: Saving Stata datasets



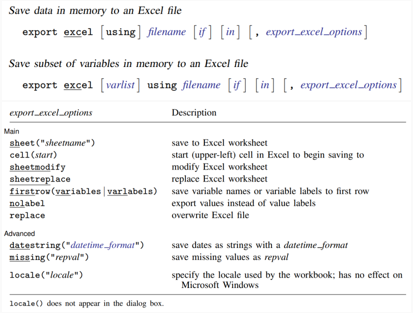
Save data in memory to file

```
save [filename] [, save_options]
```

Save data in memory to file in Stata 12 format

```
saveold filename [, saveold_options]
```

save_options	Description
`nolabel`	omit value labels from the saved dataset
`replace`	overwrite existing dataset
`all`	save `e(sample)` with the dataset; programmer's option
`orphans`	save all value labels
`emptyok`	save dataset even if zero observations and zero variables



Save data in memory to an Excel file

```
export excel [using] filename [if] [in] [, export_excel_options]
```

Save subset of variables in memory to an Excel file

```
export excel [varlist] using filename [if] [in] [, export_excel_options]
```

export_excel_options	Description
Main	
`sheet("sheetname")`	save to Excel worksheet
`cell(start)`	start (upper-left) cell in Excel to begin saving to
`sheetmodify`	modify Excel worksheet
`sheetreplace`	replace Excel worksheet
`firstrow(variables|varlabels)`	save variable names or variable labels to first row
`nolabel`	export values instead of value labels
`replace`	overwrite Excel file
Advanced	
`datestring("datetime_format")`	save dates as strings with a *datetime_format*
`missing("repval")`	save missing values as *repval*
`locale("locale")`	specify the locale used by the workbook; has no effect on Microsoft Windows

`locale()` does not appear in the dialog box.

Lab Task 7: Saving Stata datasets

Let's save the modified data as a dta file. Type...

```
save "$data\cs_s0_s5_household_modified.dta", replace
```

Lab Task 7: Saving Stata datasets

Let's save the modified data as a dta file. Type...

```
save "$data\cs_s0_s5_household_modified.dta", replace
```

Notice that we use the ***replace*** option. This overwrites the existing file. Type the same command without ***, replace***, and see what error you get!

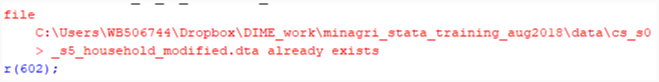
Lab Task 7: Saving Stata datasets

Let's save the modified data as a dta file. Type...

```
save "$data\cs_s0_s5_household_modified.dta", replace
```

Notice that we use the ***replace*** option. This overwrites the existing file. Type the same command without ***, replace***, and see what error you get!

Did you get an error like this?



```
file
    C:\Users\WB506744\Dropbox\DIME_work\minagri_stata_training_aug2018\data\cs_s0
    > _s5_household_modified.dta already exists
r(602);
```

Lab Task 7: Saving Stata datasets

Now, let's save the modified data as a excel. This is helpful if you are sending the dataset to someone who does not use or have Stata. Type...

```
export excel using "$data\cs_s0_s5_household_modified.xls", replace
```

Lab Task 7: Saving Stata datasets

Now, let's save the modified data as a excel. This is helpful if you are sending the dataset to someone who does not use or have Stata. Type...

```
export excel using "$data\cs_s0_s5_household_modified.xls", replace
```

Open the output file. Notice that it doesn't have variable names as column names. This is very inconvenient!

Lab Task 7: Saving Stata datasets

Now, let's save the modified data as a excel. This is helpful if you are sending the dataset to someone who does not use or have Stata. Type...

```
export excel using "$data\cs_s0_s5_household_modified.xls", replace
```

Open the output file. Notice that it doesn't have variable names as column names. This is very inconvenient! Use an optional command, ***firstrow(variables)***.

```
export excel using "$data\cs_s0_s5_household_modified.xls", ///
replace firstrow(variables)
```

Lab Task 7: Saving Stata datasets

Now, let's save the modified data as a excel. This is helpful if you are sending the dataset to someone who does not use or have Stata. Type...

```
export excel using "$data\cs_s0_s5_household_modified.xls", replace
```

Open the output file. Notice that it doesn't have variable names as column names. This is very inconvenient! Use an optional command, ***firstrow(variables)***.

```
export excel using "$data\cs_s0_s5_household_modified.xls", ///
replace firstrow(variables)
```

Notice ///. This is a way to let Stata now that multiple lines constitute a single command. It's helpful when your command is getting too long on your do file.

Lab Task 7: Saving Stata datasets

Now, let's save the modified data as a excel. This is helpful if you are sending the dataset to someone who does not use or have Stata. Type...

```
export excel using "$data\cs_s0_s5_household_modified.xls", replace
```

Open the output file. Notice that it doesn't have variable names as column names. This is very inconvenient! Use an optional command, ***firstrow(variables)***.

```
export excel using "$data\cs_s0_s5_household_modified.xls", ///
replace firstrow(variables)
```

Notice ///. This is a way to let Stata now that multiple lines constitute a single command. It's helpful when your command is getting too long on your do file. Open the newly saved excel file. You will find column names!

Section 3:
Introduction to Stata Graphics

Table gives all the details.

What's happening in this regression table? What's important?

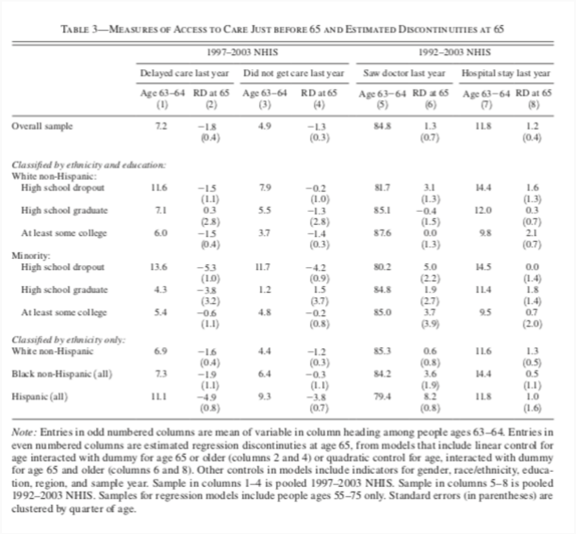


TABLE 3—MEASURES OF ACCESS TO CARE JUST BEFORE 65 AND ESTIMATED DISCONTINUITIES AT 65

	1997–2003 NHIS				1992–2003 NHIS			
	Delayed care last year		Did not get care last year		Saw doctor last year		Hospital stay last year	
	Age 63–64 (1)	RD at 65 (2)	Age 63–64 (3)	RD at 65 (4)	Age 63–64 (5)	RD at 65 (6)	Age 63–64 (7)	RD at 65 (8)
Overall sample	7.2	−1.8 (0.4)	4.9	−1.3 (0.3)	84.8	1.3 (0.7)	11.8	1.2 (0.4)
Classified by ethnicity and education:								
White non-Hispanic:								
High school dropout	11.6	−1.5 (1.1)	7.9	−0.2 (1.0)	81.7	3.1 (1.3)	14.4	1.6 (1.3)
High school graduate	7.1	0.3 (2.8)	5.5	−1.3 (2.8)	85.1	−0.4 (1.5)	12.0	0.3 (0.7)
At least some college	6.0	−1.5 (0.4)	3.7	−1.4 (0.3)	87.6	0.0 (1.3)	9.8	2.1 (0.7)
Minority:								
High school dropout	13.6	−5.3 (1.0)	11.7	−4.2 (0.9)	80.2	5.0 (2.2)	14.5	0.0 (1.4)
High school graduate	4.3	−3.8 (3.2)	1.2	1.5 (3.7)	84.8	1.9 (2.7)	11.4	1.8 (1.4)
At least some college	5.4	−0.6 (1.1)	4.8	−0.2 (0.8)	85.0	3.7 (3.9)	9.5	0.7 (2.0)
Classified by ethnicity only:								
White non-Hispanic	6.9	−1.6 (0.4)	4.4	−1.2 (0.3)	85.3	0.6 (0.8)	11.6	1.3 (0.5)
Black non-Hispanic (all)	7.3	−1.9 (1.1)	6.4	−0.3 (1.1)	84.2	3.6 (1.9)	14.4	0.5 (1.1)
Hispanic (all)	11.1	−4.9 (0.8)	9.3	−3.8 (0.7)	79.4	8.2 (0.8)	11.8	1.0 (1.6)

Note: Entries in odd numbered columns are mean of variable in column heading among people ages 63–64. Entries in even numbered columns are estimated regression discontinuties at age 65, from models that include linear control for age interacted with dummy for age 65 or older (columns 2 and 4) or quadratic control for age, interacted with dummy for age 65 and older (columns 6 and 8). Other controls in models include indicators for gender, race/ethnicity, education, region, and sample year. Sample in columns 1–4 is pooled 1997–2003 NHIS. Sample in columns 5–8 is pooled 1992–2003 NHIS. Samples for regression models include people ages 55–75 only. Standard errors (in parentheses) are clustered by quarter of age.

But figures *tell the story*.

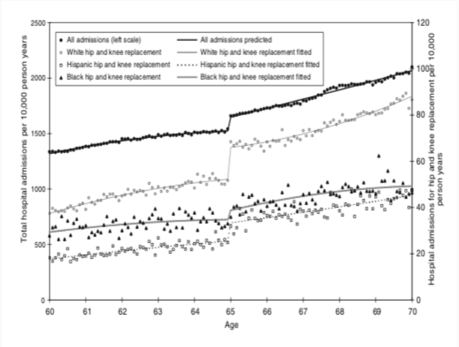
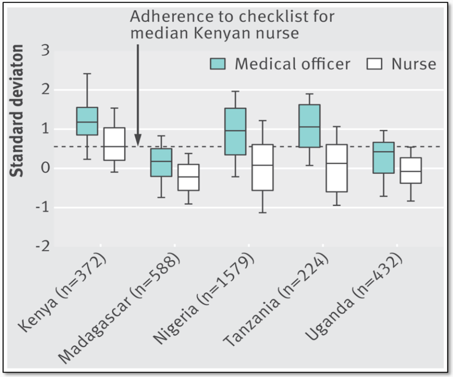


FIGURE 3. HOSPITAL ADMISSION RATES BY RACE/ETHNICITY

- This is the data that generates those estimates.
- You can see exactly what is happening very quickly!
- Even more importantly: **Your eyes are naturally drawn to the story!**

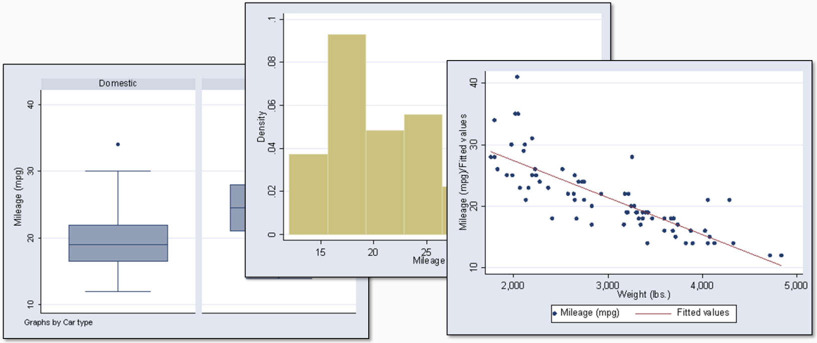
Example: comparing means



- What is the main story in this graph?
- We need more context to say something detailed about this, but what has the person creating the graph highlighted for us?

Stata default graphs

- This is what a Stata graph looks like with very minimal customizing using optional commands.
- Notice that there is no graph title. They are still informative, but need much improvement.
- We will not go too deep to editing a Stata graph today, but I'll show you have to make a graph and make some edits for effective data visualization.

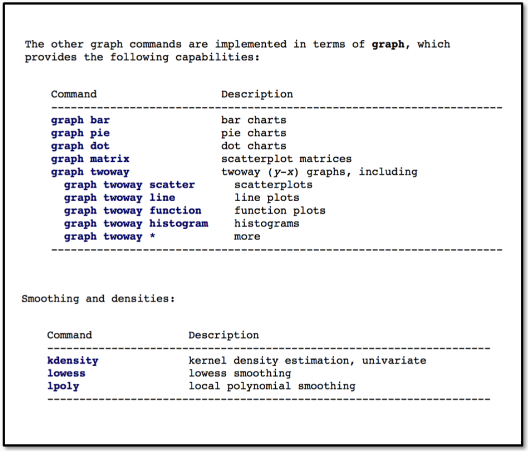


Stata has three core built-in graph functions.

[graph *graphtype*]
graphs which plot one or more variables on one axis

[twoway *graphtype*]
graphs which plot two variables together on an x and y axis

[histogram], [kdensity], [lowess]
Essential distributional commands



```
The other graph commands are implemented in terms of graph, which
provides the following capabilities:

    Command                   Description
    ---------------------------------------------------------------------
    graph bar                 bar charts
    graph pie                 pie charts
    graph dot                 dot charts
    graph matrix              scatterplot matrices
    graph twoway              twoway (y-x) graphs, including
      graph twoway scatter      scatterplots
      graph twoway line         line plots
      graph twoway function     function plots
      graph twoway histogram    histograms
      graph twoway *            more
    ---------------------------------------------------------------------


Smoothing and densities:

    Command                   Description
    ---------------------------------------------------------------------
    kdensity                  kernel density estimation, univariate
    lowess                    lowess smoothing
    lpoly                     local polynomial smoothing
    ---------------------------------------------------------------------
```

Oneway vs Twoway

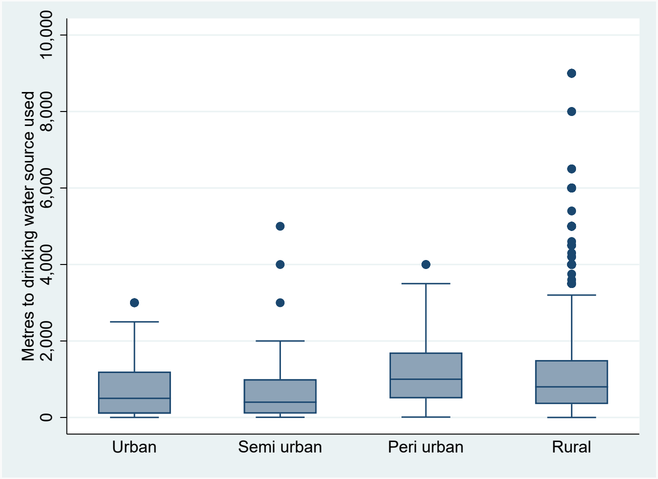
super quick explanation about what they mean and twoway_options

Stata graph exercise 1

Box plot

Box plot

Let's make a a box plot like the one below using the variable, **m_drink_ws**. Notice a box plot is an example of a oneway graph.



Box plot

Let's make a box plot from your do file.

1. Type ***search box plots*** in the command window to find out what command to be used. ***search*** is a more general search through help files and other Stata resources.

Box plot

Let's make a box plot from your do file.

1. Type ***search box plots*** in the command window to find out what command to be used. ***search*** is a more general search through help files and other Stata resources.
2. The command should look like the following. Run from the do file.

```
graph box m_drink_ws
```

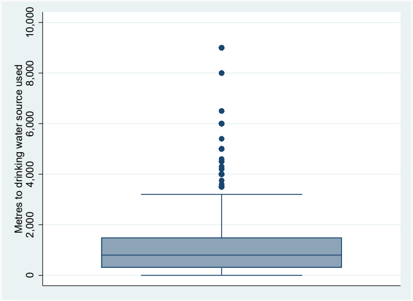
Box plot

Let's make a box plot from your do file.

1. Type ***search box plots*** in the command window to find out what command to be used. ***search*** is a more general search through help files and other Stata resources.
2. The command should look like the following. Run from the do file.

```
graph box m_drink_ws
```

3. Notice the difference from earlier?



Box plot

Now, let's make multiple box plots by the residential environment, **urban_2012**.

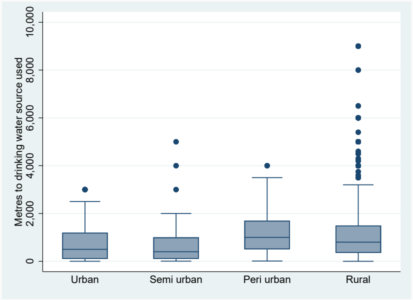
1. Type ***search box plots*** to see how to achieve this.

Box plot

Now, let's make multiple box plots by the residential environment, **urban_2012**.

1. Type ***search box plots*** to see how to achieve this.
2. The optional commant, ***over()*** can do this. Run the new ***graph box*** command with the ***over*** option.

```
graph box m_drink_ws, over(urban_2012)
```

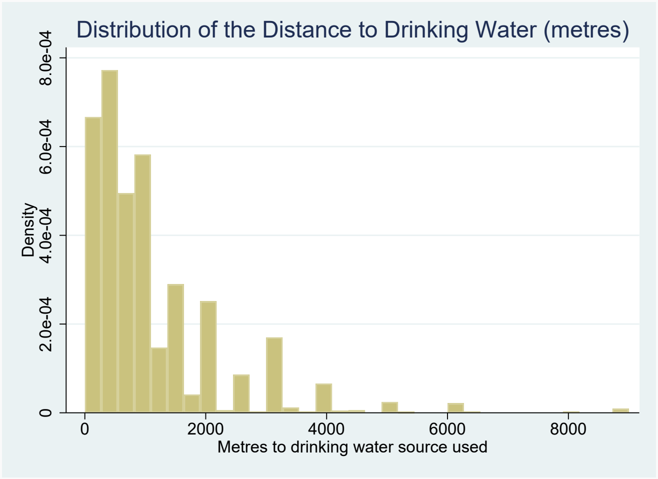


Stata graph exercise 2

Histogram

Histogram

Let's make a a histogram like the one below using the variable, **m_drink_ws**. Notice that a histogram is an example of a twoway graph.



Histrogram

Let's make a histogram from your do file.

1. Type ***help histogram*** in the command window to find out what command to be used.

Histrogram

Let's make a histogram from your do file.

1. Type ***help histogram*** in the command window to find out what command to be used.
2. The command should look like the following. Run from the do file.

```
histogram m_drink_ws
```

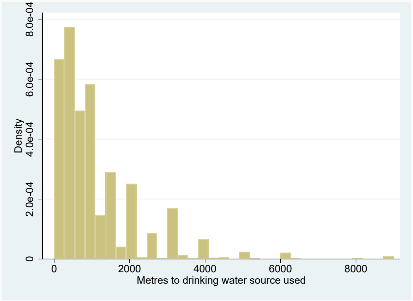
Histrogram

Let's make a histogram from your do file.

1. Type ***help histogram*** in the command window to find out what command to be used.
2. The command should look like the following. Run from the do file.

```
histogram m_drink_ws
```

3. Notice the difference from earlier? We need the title.



Histrogram

Now, let's add the title. You can also choose your own title that is informative. Notice in general a good title is informative but short.

Histrogram

Now, let's add the title. You can also choose your own title that is informative. Notice in general a good title is informative but short.

1. The optional command, ***title()*** can do this. Run the new ***histogram*** command with the ***title*** option.

```
histogram m_drink_ws, ///
title("Distribution of the Distance to Drinking Water (metres)")
```

Histrogram

Now, let's add the title. You can also choose your own title that is informative. Notice in general a good title is informative but short.

1. The optional command, ***title()*** can do this. Run the new ***histogram*** command with the ***title*** option.

```
histogram m_drink_ws, ///
title("Distribution of the Distance to Drinking Water (metres)")
```

Histrogram

Now, let's add the title. You can also choose your own title that is informative. Notice in general a good title is informative but short.

1. The optional command, ***title()*** can do this. Run the new ***histogram*** command with the ***title*** option.

   ```
   histogram m_drink_ws, ///
   title("Distribution of the Distance to Drinking Water (metres)")
   ```

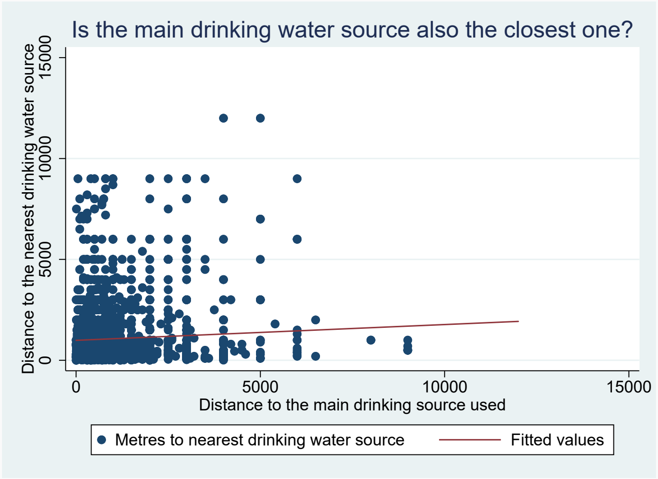
2. ***help twoway_options*** to find out more about the ***title*** option and more.

Stata graph exercise 3

Scatter plot

Scatter plot

Let's make a a scatter plot with a fitted line like the one below using the variable, **m_drink_ws** and **m_used_ws**. Notice that a scatter plot with a fitted line is an example of a twoway graph.



Scatter plot

Let's make a scatter plot from your do file.

1. Type ***help scatter*** in the command window to find out what command to be used.

Scatter plot

Let’s make a scatter plot from your do file.

1. Type ***help scatter*** in the command window to find out what command to be used.
2. The command should look like the following. Run from the do file.

```
scatter m_used_ws m_drink_ws
```

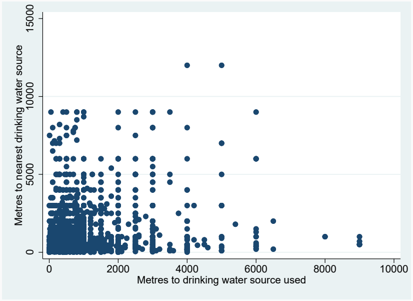
Scatter plot

Let's make a scatter plot from your do file.

1. Type ***help scatter*** in the command window to find out what command to be used.
2. The command should look like the following. Run from the do file.

```
scatter m_used_ws m_drink_ws
```

3. Notice the difference from earlier? No fitted line!



Scatter plot

Let's add a fitted line. Type ***help lfit*** to learn how to do this.

Scatter plot

Let's add a fitted line. Type ***help lfit*** to learn how to do this.

1. You may notice that this is an entirely different command. Stata can actually overlay multiple twoway graphs. To do this, run the following command. Notice that || is a way to overlay the graphs.

```
scatter m_used_ws m_drink_ws  || ///
lfit m_drink_ws m_used_ws
```

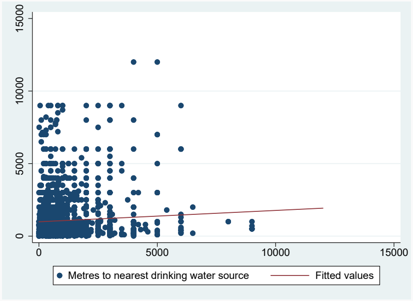
Scatter plot

Let's add a fitted line. Type ***help lfit*** to learn how to do this.

1. You may notice that this is an entirely different command. Stata can actually overlay multiple twoway graphs. To do this, run the following command. Notice that || is a way to overlay the graphs.

```
scatter m_used_ws m_drink_ws  || ///
lfit m_drink_ws m_used_ws
```

2. Notice the difference from earlier? No main title and y and y titles!



Scatter plot

This is because the fitted line is a linear prediction and no longer represents the raw distance values. But we can simply add on titles that can be helpful for the graph's intended audience.

Scatter plot

This is because the fitted line is a linear prediction and no longer represents the raw distance values. But we can simply add on titles that can be helpful for the graph's intended audience.

1. Recall the ***twoway_options***, and the ***title()*** option. You can use the same option and very similar options called ***xtitle()*** and ***ytitle()***.

```
scatter m_used_ws m_drink_ws  || ///
lfit m_drink_ws m_used_ws, ///
        ytitle("Distance to the nearest drinking water source") ///
        xtitle("Distance to the main drinking source used") ///
        title("Is the main drinking water also the closest source?")
```

Stata graph exercise 3

Saving and combining graphs

Saving a Stata graph

Let's save all 3 graphs we made today.

1. To do so, add ***graph save*** after each of your graphs like the following.
2. Notice that you need to specify where you want save it, and how you want to name it.

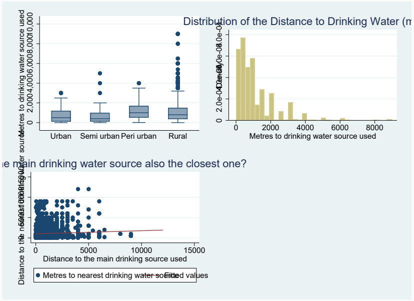
Combining Stata graphs

Let's combine all 3 graphs we made today.

1. To do so, add ***graph combine*** after each of your graphs like the following.
2. Notice that you need to specify where you want save it, and how you want to name it.

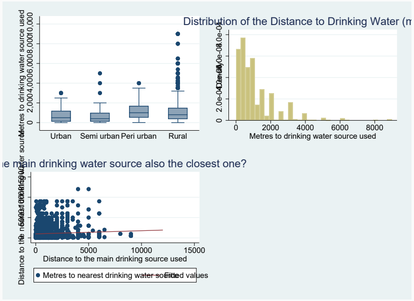
Combining Stata graphs

Does yours look like this?



Combining Stata graphs

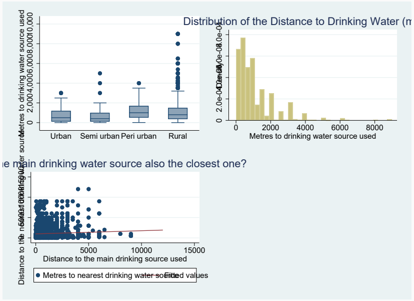
Does yours look like this?



- Why look so ugly?

Combining Stata graphs

Does yours look like this?



- Why look so ugly?
- Luckily, there are ways to make this nice like this.

Add the clean version of this and tell that the code is in the solution do file.

Extra section: More about Stata

Other features of Stata

Using macros
(globals, locals and scalars)

Macros

- You need to be at least familiar with this topic for the resources you will be introduced to this week
- This technique is critical as projects grow in size. But even the smallest DIME project absolutely needs this.
- Macros (globals, locals, scalar) save some information (text or number) that you can reference later.
    - Example: we want to access files in the folder multiple times. We can store the folder location in a global and use it multiple times

Defining macros - local

Type in your dofile the following and *run all the lines at once*.

- Note that ‘ is not the same as ’.

```
local numberA 3
local numberB 5
local result = (`numberA´ * `numberB´) - `numberA´
display "The result is `result´."
```

Defining macros - local

Type in your dofile the following and *run all the lines at once*.

- Note that ‘ is not the same as ’.

```
local numberA 3
local numberB 5
local result = (`numberA´ * `numberB´) - `numberA´
display "The result is `result´."
```

- What did the result say?

Defining macros - local

Type in your dofile the following and *run all the lines at once*.

- Note that ‘ is not the same as ’.

```
local numberA 3
local numberB 5
local result = (`numberA´ * `numberB´) - `numberA´
display "The result is `result´."
```

- What did the result say?

```
The result is 12.
```

Defining macros - local

Type in your dofile the following and *run all the lines at once*.

- Note that ‘ is not the same as ’.

```
local numberA 3
local numberB 5
local result = (`numberA´ * `numberB´) - `numberA´
display "The result is `result´."
```

- What did the result say?

```
The result is 12.
```

- Try running them one by one, and see what happens?

Defining macros - local

Type in your dofile the following and *run all the lines at once*.

- Note that ‘ is not the same as ’.

```
local numberA 3
local numberB 5
local result = (`numberA´ * `numberB´) - `numberA´
display "The result is `result´."
```

- What did the result say?

```
The result is 12.
```

- Try running them one by one, and see what happens?
    - It probably didn’t run. This is one of the major differences between global and local. Local is really local and only last within a single run. For more please refer to the help file on *macro*.

Other features of Stata

Missing values

Missing values

- String variables can be empty, but numeric variables canâĂŹt be empty. Instead numeric variables have something called âĂIJmissing valuesâĂİ.
    - Missing values are represented in Stata with a period as in " **.** ".
    - You can also use .a or .b etc. to .z for missing values and you will learn later how these can be used
- Stata canâĂŹt use missing values in computations (averages, regressions etc.) so it skips observations with missing values .
- Missing values changes the analysis as observations with missing values are excluded from commands like summarize and regress.
- Good practice to always check for missing values when tabulating variables.

One more useful command

tabstat:
another command of summary statistics

One more useful command

tabstat

- While ***summarize*** and ***tabulate*** provide useful fixed format output, ***tabstat*** gives you the ability specify exactly what statistics you want in your input.
- By default, ***tabstat*** only disply the mean.
- We can add a whole range of statistics using the option ***statistics()***. See ***help tabstat***, for a list of the statistics you can add.

One more useful command

Here are some examples.

- This is the very basic command.

```
. tabstat m_main_ws

    variable |      mean
-------------+----------
   m_main_ws |  791.2462
------------------------
```

- You can add multiple variable at a time.

```
. tabstat m_main_ws m_used_ws

   stats |  m_main~s  m_used~s
---------+--------------------
    mean |  791.2462   863.863
------------------------------
```

One more useful command

Lastly...

- You choose what types of statistics you want it to display.

```
. tabstat m_main_ws m_used_ws, statistics(mean sd median)

   stats |  m_main~s  m_used~s
---------+--------------------
    mean |  791.2462   863.863
      sd |    853.42  1005.191
     p50 |       500       500
------------------------------
```