Dynamic Presentations in Stata using Markstat

Luiza Andrade

March 1, 2019

Outline

1. Dynamic documents
2. `markdown` and `markstat`
3. `markstat` installation
4. Including console output in documents
5. Including graphs in documents
6. Including tables in documents
7. Including inline code in documents
8. References and resources

Introduction

- What is a dynamic document?

In the context of this presentation, what we call a *dynamic document* is a document that contain both text and Stata outputs, where the outputs are updated automatically every time the script runs

Introduction

- Why use dynamic documents?

Most tools for dynamic documents are created with *literate data analysis* in mind, where code and documentation being produced together to increase research transparency.

Introduction

When to use dynamic documents?

- Include text and outputs in the same document
- Better for simple documents, that don't require a lot of formatting
- Include code in the document
- Quickly visualize formatted tables

Introduction

Pros:

- Save time spent on copying and pasting or switching software
- Best option to include (and run) code in a document

Cons:

- Error messages may not be super clear (specially when using LaTeX)
- Harder to include detailed formatting
- No syntax highlighting for text (in Markdown or TeX)

Introduction

- There are a handful of options for dynamic documents in Stata
- You can find a review of different options in this link
- For this presentation, we will use `markstat`, as we believe it's the most general and user friendly
- texdoc is also a good option for LaTeX users
- At the end of this presentation, you can find some material on the different tools for dynamic documents in Stata

Introduction

`markstat`

- Stata command created by German Rodriguez
- Allows users to create and compile and Stata markdown file combining Stata code and markdown text
- Saves the outputs to PDF, word, HTML and beamer

Introduction

markstat

1. Reads the Stata markdown file
2. Separates (*tangles*) markdown and Stata code
3. Runs each of them separately
4. Puts their outputs back together (*weaves*) into a single document in the format you choose

Introduction

`markdown`

- Lightweight markup language
- Designed to be easily readable
- We won't go into details about markdown in this session, but some resources are listed in the end of this presentation
- This session's material includes a cheatsheet with everything you need to know to complete today's exercises

Get the material

1. Go to the workshop repository
2. Click **`Clone or download`**
3. If you have GitHub desktop installed, clone it to your GitHub folder
4. If you prefer to skip that step, just download it as a `.zip` file

Installation

Install the necessary programs

1. pandoc
2. TeX/LaTeX

Installation

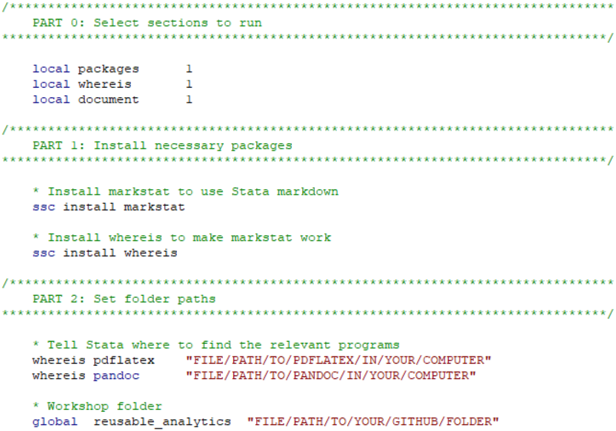
Find out where the programs are installed

- On Windows: type `where pdflatex` and `where pandoc` on the command line
- On Mac: open `terminal` and type `which pdflatex` and `which pandoc`

Installation

1. Go to the workshop folder
2. Go to the `Stata markdown` folder
3. Open `Master.do`

Installation



```
/*******************************************************************************
    PART 0: Select sections to run
*******************************************************************************/

    local packages      1
    local whereis       1
    local document      1

/*******************************************************************************
    PART 1: Install necessary packages
*******************************************************************************/

    * Install markstat to use Stata markdown
    ssc install markstat

    * Install whereis to make markstat work
    ssc install whereis

/*******************************************************************************
    PART 2: Set folder paths
*******************************************************************************/

    * Tell Stata where to find the relevant programs
    whereis pdflatex    "FILE/PATH/TO/PDFLATEX/IN/YOUR/COMPUTER"
    whereis pandoc      "FILE/PATH/TO/PANDOC/IN/YOUR/COMPUTER"

    * Workshop folder
    global  reusable_analytics  "FILE/PATH/TO/YOUR/GITHUB/FOLDER"
```

Installation

1. Paste the location of pdflatex to the line that starts with `whereis pdflatex`
2. Paste the location of pandoc to the line that starts with `whereis pandoc`
3. Make sure all the locals in `PART 0` are equal to 1
4. Run `Master.do`

markstat

The command that creates the final document is markstat

```
markstat using filename,
    [pdf docx slides beamer mathjax
     bibliography strict nodo nor keep]
```

markstat

Exercise 1:

Test different output formats for Stata markdown template by specifying on master:

1. `markstat using "${reusable_analytics}/Stata markdown template", pdf`
2. `markstat using "${reusable_analytics}/Stata markdown template", docx`
3. `markstat using "${reusable_analytics}/Stata markdown template", slides`
4. `markstat using "${reusable_analytics}/Stata markdown template", beamer`

markstat

Here are some notes on Exercise 1:

- Go to `markstat` website to see how to change the slides theme
- On beamer, slides with Stata code or ouput need to be in the fragile style. It can be set like this:

```
# Slide title {.fragile}
```

markstat

In some systems, you will not be able to replace the PDF if it's open. There are two possible solutions:

- Close the PDF file before running `markstat`; or
- Close the PDF file once you get an error message and press enter on the command window

`markstat`

- Writing markdown in Stata with `markstat` is simple, and similar to what would be done in `R`, for example
- Open the file called `Stata markdown template.stmd` to see how it works
- To write (and format) text, write markdown without indentation – use `Markdown cheatsheet.stmd` for examples of how to format text using markdown

Including Stata code

- The simplest way to write Stata code is start a line with four spaces or one tab:

```
Hello world!

    sysuse auto, clear
```

Including Stata code

- You can also use fenced code blocks (as the one below)
- They make the `.stmd` file harder to read, but allow you to use more advanced options – we'll see some examples soon

~~~
 Write text without indentation

```{s}
    * Write stata code inside chunks
    sum mpg
```
~~~

Including Stata output

Exercise 2:

1. Under the second title in `Stata markdown template.stmd`, add Stata code using a command that prints some output to the Stata window

- `summarize`, `keep`, `gen` and `tab` are good examples

1. Save the markdown file
2. If you have a PDF open, close it
3. Open `Master.do`
4. Set the `packages` and `paths` locals to `0`
5. Run `Master.do`

Including Stata output

~~~
```{s}
    * Summary of miles per gallon
    sum mpg
```
~~~

```
. * Summary of miles per gallon
. sum mpg
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         mpg |         74     21.2973    5.785503         12         41
```

Including Stata graphs

To include Stata graphs:

1. Create the graphs in Stata
2. Save it locally using `graph export`
3. Use the following markdown syntax to include the graph:
   `![figure caption](figure name.png)`

Including Stata graphs

~~~~
```{s}
    scatter weight length, ///
        legend(off)
    graph export scatter.png, width(800) replace
```

![Correlation between weight and length](scatter.png)
~~~~

Including Stata graphs

Exercise 3

1. Go to `Stata markdown template.stmd`
2. Create and export a graph using the loaded data
3. Include markdown code to display the graph you just saved
4. Save the markdown file
5. If you have a PDF open, close it
6. Run `Master.do`

Including Stata graphs

```
. scatter weight length, ///
>     legend(off)
. graph export scatter.png, width(800) replace
(file scatter.png written in PNG format)
```

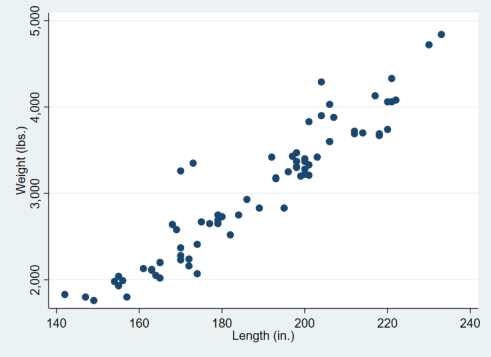


Figure 1: Correlation between weight and length

Omitting Stata code

- Depending on the type of document you are writing, you may want to only display the results of your code (tables, graphs, etc)
- This is when using strict code blocks is useful
- To omit the Stata code from the document, type `{s/}` on the opening of your code chunk

Omitting Stata code

~~~~
```{s/}
    scatter weight length, ///
        legend(off)
    graph export scatter.png, width(800) replace
```

![Correlation between weight and length](scatter.png)
~~~~

Omitting Stata code

```
(file scatter.png written in PNG format)
```

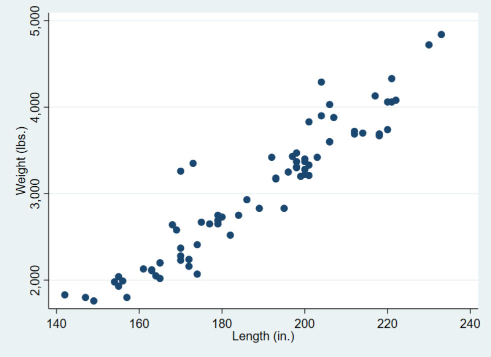


Figure 2: Correlation between weight and length

Omitting Stata output

- Now, the last slide probably wasn't exactly what you were expecting, right?
- Using `{s/}` will commit the code you used, but not it's output
- To omit the output, simply run your code `quietly`

Omitting Stata output

~~~~
```{s/}
    scatter weight length, ///
        legend(off)
    quietly graph export scatter.png, width(800) replace
```

![Correlation between weight and length](scatter.png)
~~~~

Omitting Stata output

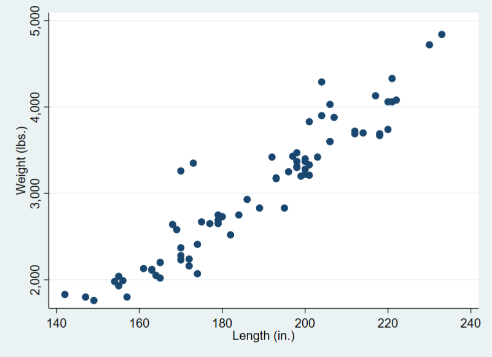


Figure 3: Correlation between weight and length

Including Stata tables

- To include estimation results, we recommend using `esttab`
- The window output of `esttab` is well-formatted, and including that output is the simplest way to display a table
- `esttab` also exports to HTML and TeX, but those are more advanced examples that are beyond the scope of this presentation
- You can find more detailed examples in the Stata Markdown website

Including Stata tables

~~~~
```{s/}
    qui reg price headroom
    est sto reg1

    qui regress price headroom trunk
    est sto reg2

    qui regress price headroom trunk foreign
    est sto reg3

    esttab reg1 reg2 reg3, ///
        replace ///
        label se ///
        nomtitles
```
~~~~

Including Stata tables

	(1)	(2)	(3)
Headroom (in.)	399.2	-580.8	-519.7
	(408.2)	(519.5)	(516.9)
Trunk space (.. ft.)		292.8**	328.4**
		(102.8)	(104.7)
Car type			1128.8
			(763.2)
Constant	4970.3***	3875.9**	2866.9*
	(1269.0)	(1270.0)	(1432.4)
Observations	74	74	74

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

Including Stata tables

Exercise 4

1. Go to `Stata markdown template.stmd`
2. Inside a Stata code block, run a few simple regressions on the loaded data
3. Use `esttab` to output the regression results
4. Save the markdown file
5. If you have a PDF open, close it
6. Run `Master.do`

Including code inline

- Sometimes we want to reference numbers in our text
- If the numbers change for any reason, it's better to have them automatically updated than review the whole presentation for adjustments
- Markdown lets you write code inline with your text

Including code inline

- Writing

```
Today is `s c(current_date)`.
```

- Will result in

Today is 28 Feb 2019.

Including code inline

Exercise 5

Using inline Stata code, try to include the following items to your current markdown file:

- The sample includes 74 different car models produced by 23 different companies
- 22 are foreign models, and 52 are domestic

Including code inline

~~~
```{s}
qui count
local models `r(N)'

cap drop make_*
qui split make, gen(make_)
qui unique make_1
local makes `r(unique)'

qui count if foreign == 1
local foreign `r(N)'

qui count if foreign == 0
local domestic `r(N)'
```
~~~

Including code inline

```
+ The sample includes `s `models'` different car
models produced by `s `makes'` different companies
+ `s `foreign'` are foreign models, and `s `domestic'`
are domestic
```

Including code inline

- Inline code is particularly useful when you want to display a custom table
- You can create the table using markdown, and add the numbers to the right columns using locals
- However, to create these you need to specify the `strict` option when compiling
- And they will not necessarily render will in all different formats

Including code inline

- Writing

```
| Car origin   | N obs        |
|:-------------|-------------:|
| Domestic     |`s `domestic'`|
| Foreign      |`s `foreign'` |
```

- Will result in

Car origin	N obs
Domestic	52
Foreign	22

Adding a title to your document

- There are three pieces of metadata that you can easily add to your document: title, author and date
- You can do this by adding the following code to the beginning of your document:

```
% Document Title
% Author
% Date
```

Other tools for dynamic documents in Stata

texdoc

- Stata package created by Ben Jann
- Write LaTeX code instead of markdown
- Resulting document is not as easy to read
- But can be easier to format, if you know TeX well
- Debugging LaTeX errors can be hard

Other resources

- This presentation was mostly based on German Rodriguez, 2017. "MARKSTAT: Stata module to support literate data analysis using Stata and Markdown," Statistical Software Components S458401, Boston College Department of Economics, revised 08 May 2018.
- The `markstat` website contains a lot of material, examples and FAQs
- texdoc is another option to create dynamic documents in Stata

datalibweb

- **`datalibweb`** is a data system developed by the Global Poverty Team for Statistical Development of the World Bank
- Allows users to access raw and harmonized data sets collected across Global Practices
- Collections are maintained by regional statistics teams and kept up to date
- Version control of data sets
- Explore data set documentation

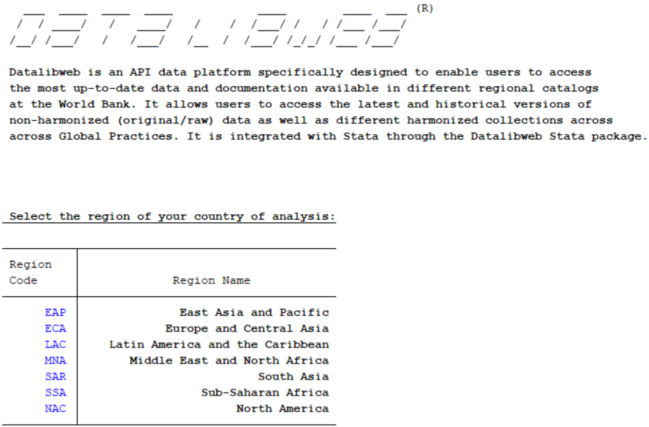
datalibweb installation

- In Stata, type

```
net install datalibweb, ///
    all replace force ///
    from("http://eca/povdata/datalibweb/_ado")

datalibweb, update(ado)
```

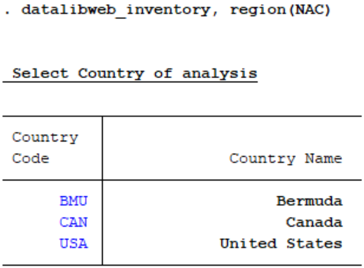
datalibweb usage

Type datalibweb to explore the data sets:



```
  ___  ____  ____  ____          ____        ____  ___ (R)
 /  / ____/   /   ____/   /    / /___/ /    / /___ /___/
/__/ /___/   /   /___/   /__  / /___/ /_/_/ /___ /___/

Datalibweb is an API data platform specifically designed to enable users to access
the most up-to-date data and documentation available in different regional catalogs
at the World Bank. It allows users to access the latest and historical versions of
non-harmonized (original/raw) data as well as different harmonized collections across
across Global Practices. It is integrated with Stata through the Datalibweb Stata package.



 Select the region of your country of analysis:
-----------------------------------------------

---------------------------------------------
 Region    |
 Code      |             Region Name
-----------+---------------------------------
      EAP  |          East Asia and Pacific
      ECA  |        Europe and Central Asia
      LAC  |   Latin America and the Caribbean
      MNA  |      Middle East and North Africa
      SAR  |                     South Asia
      SSA  |             Sub-Saharan Africa
      NAC  |                  North America
---------------------------------------------
```

datalibweb usage



```
. datalibweb_inventory, region(NAC)


 Select Country of analysis

 Country
 Code     |                Country Name
----------+-----------------------------
      BMU |                     Bermuda
      CAN |                      Canada
      USA |               United States
```

datalibweb

For more information, go to FURL datalibweb