

# Dynamic documents in Stata

Luiza Andrade

December 18, 2023



# 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 document is compiled.

# Introduction

## ► Why use dynamic documents?

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

```
1
2 # My Analysis
3
4 Here I tell what I'll do.
5 Now I'll show some code:
6
7 ```{s}
8 sysuse auto
9 summarize price
10 ```
11
12 And here I can explain what I just showed,
13 all in the same document.
14
15 This is how dynamic documents work.
```



### My Analysis

Here I tell what I'll do. Now I'll show some code:

```
. sysuse auto
(1978 Automobile Data)
. summarize price
```

Variable	Obs	Mean	Std. Dev.	Min	Max
price	74	6165.257	2949.496	3291	15906

And here I can explain what I just showed, all in the same document.

This is how dynamic documents work.

# Introduction

When to use dynamic documents (in Stata)?

- ▶ To include text and outputs in the same document – no more copying and pasting figures, tables or plots to a document editor!
- ▶ To create simple documents that don't require a lot of formatting
- ▶ To include code snippets or code examples in a document
- ▶ To quickly visualize formatted tables

# Introduction

## **Pros:**

- ▶ Save time spent on copying and pasting or switching software
- ▶ Avoid having outdated exhibits in outputs
- ▶ Include (and run) code in a document

## **Cons:**

- ▶ Error messages may not be super clear (specially when using LaTeX)
- ▶ Harder to fine-tune formatting
- ▶ Incomplete syntax highlighting

# Introduction

- ▶ There are a handful of options for dynamic documents in Stata
  - ▶ Built-in: `dyndoc`
  - ▶ RMarkdown extension: `Statamarkdown`
  - ▶ Haghish's `markdoc`
  - ▶ Ben Jann's `webdoc` and `texdoc`
- ▶ For this presentation, we will use `markstat`, as I believe it's the most general and user-friendly
  - ▶ Syntax is simple
  - ▶ Can be used through the do-file editor
  - ▶ Can create all the file formats I like to use



# Introduction

## `markstat`

- ▶ Stata command created by German Rodriguez
- ▶ Allows users to create and compile Stata markdown files by 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 this session's material includes a cheatsheet with everything you need to know to for basic formatting

## Get the material

1. Go to <https://github.com/luizaandrade/stata-markdown>
2. Click on the green <> Code button
  - ▶ Clone the repository; or
  - ▶ Download files in a ZIP and unzip them
1. Today's demo uses two files: `Main.do` and `stata-markdown-template.stmd`

# Dependencies

1. pandoc
2. TeX/LaTeX: if saving to PDF

# Setup

Find out where the programs are installed

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

# Setup

- ▶ To compile the Stata markdown file into a document, Stata needs to know where to find the programs that compile the document
- ▶ This is done by using the Stata command `whereis` to point to the installations of `pdflatex` and `pandoc` that you found using the code in the previous slide
- ▶ The file `Main.do` in this session's repository includes an example of how to do this in Part 2

# Stata setup

```
22  /*****
23  PART 1: Install necessary packages
24  *****/
25
26  if `packages' {
27
28      * Install markstat to use Stata markdown
29      ssc install markstat, replace
30
31      * Install whereis to make markstat work
32      ssc install whereis, replace
33
34      * Install other commands we will use for this session
35      ssc install estout, replace
36      ssc install unique, replace
37
38  }
39
40  /*****
41  PART 2: Set folder paths
42  *****/
43
44  if `paths' {
45
46      * Tell Stata where to find the relevant programs
47      whereis pdflatex      "C:\texlive\2023\bin\windows\pdflatex.exe"
48      whereis pandoc        "C:\Users\luizaandrade\AppData\Local\Pandoc\pandoc.exe"
49      global mdfolder      "C:\Users\luizaandrade\Documents\GitHub\stata-markdown"
50
51  }
```



## markstat

The command that creates the final document is markstat

markstat using filename,

```
[pdf docx slides beamer mathjax  
bibliography strict nodo nor keep]
```

## markstat

- ▶ In Main.do, we used

markstat using stata-markdown-template, pdf

- ▶ stata-markdown-template is the file with the extension .stmd you will find in the root folder we just cloned/downloaded.
  - ▶ It contains the Markdown and Stata code we just rendered to a pdf file using markstat.
  - ▶ Try opening it in the do-file editor to see how its content relates to the file generated.

## markstat

Save output to different file formats for Stata markdown template by specifying the desired format

1. markstat using  
"`{mdfolder}/stata-markdown-template`", pdf
2. markstat using  
"`{mdfolder}/stata-markdown-template`", docx
3. markstat using  
"`{mdfolder}/stata-markdown-template`", slides
4. markstat using  
"`{mdfolder}/stata-markdown-template`", beamer

## markstat

Here are some notes in case Stata returned error messages when running the code in the previous slide:

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

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

- ▶ You might have noticed that the materials folder contains a file named `stata.sty`. This file is necessary to render Stata Markdown code in PDF, and it should be placed in the same folder as your `.stmd` file

## 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
- ▶ To write text, write markdown without indentation
  - ▶ See `Resources/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:

```
This is regular text  
Hello world!
```

```
*This is Stata:  
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

- By default, the code inside the chunks, will be printed to the document, and so will its outputs

```
```{s}
```

```
* Summary of miles per gallon
```

```
sysuse auto,clear
```

```
sum mpg
```

```
```
```

```
. * Summary of miles per gallon
```

```
. sysuse auto, clear
```

```
(1978 automobile data)
```

```
. 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)
```

## 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 saved as PNG format

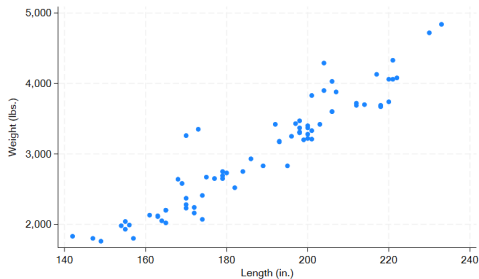


Figure 1: Correlation between weight and length

## Omitting Stata output

- ▶ Now, the last slide probably wasn't exactly what you were expecting, right?
- ▶ Using `{s/}` will omit the code you used, **but not the messages it generated**
- ▶ To omit any message, 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

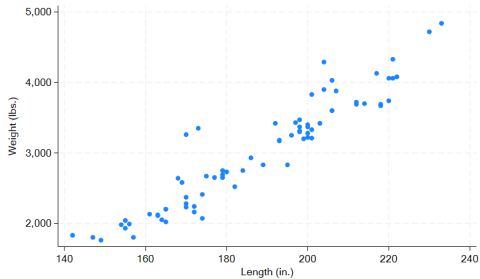


Figure 2: Correlation between weight and length

## Including math notation

- ▶ To include math notation, such as equations, you can use LaTeX notation
- ▶ `markstat` will read anything inside dollar signs (\$) as LaTeX math notation
- ▶ You can find resources on LaTeX math notation here

\$\$

`Y = \beta_{0} + \beta_{1} x_{1} + \ldots + \beta_{n} x_{n} + \epsilon`

\$\$

$$Y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n + \epsilon$$

## 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
- ▶ For a demo of how to use `esttab` to customize tables, see the [worldbank/stata-tables](https://worldbank/stata-tables) repository

## 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<br>(408.2)      | -580.8<br>(519.5)    | -519.7<br>(516.9)   |
| Trunk space (.. ft.) |                       | 292.8**<br>(102.8)   | 328.4**<br>(104.7)  |
| Car origin           |                       |                      | 1128.8<br>(763.2)   |
| Constant             | 4970.3***<br>(1269.0) | 3875.9**<br>(1270.0) | 2866.9*<br>(1432.4) |
| Observations         | 74                    | 74                   | 74                  |

Standard errors in parentheses

\* p<0.05, \*\* p<0.01, \*\*\* p<0.001

## 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 16 Dec 2023.

## Including code inline

```
```{s}  
qui count  
local models `r(N)'  
  
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
- + ``s `foreign'`` are foreign models, and ``s `domestic'`` are domestic

This results in:

- ▶ The sample includes 74 different car models
- ▶ 22 are foreign models, and 52 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
```

## Appendix: Additional 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