# Stata Markdown

DIME Analytics RA Continuing Education

11 Feb 2021

# 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?

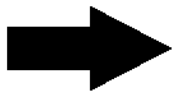
In the context of this presentation, what we call a *dynamic document* is a document that contain both text and Stata outputs, where the outputs are updated automatically every time the script is run.

# 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?

- Include text and outputs in the same document – no more having to copy and paste figures, tables or plots to a document editor!
- Nice option for simple documents that don't require a lot of formatting
- Can include code snippets or code examples in a document too
- 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 the text part in many editors (like the dofile editor), which makes it harder to debug text formatting

# 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 I believe it's the most general and user friendly
- 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 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 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 session OSF page
2. On the Files section, select `OSF Storage > 2021_02 - RA Continuing Education (Stata Markdown)`
3. Download the file `stata-markdown-CE-session.zip`
4. Unzip the file in a location you can remember

# Installation

Install the necessary programs

1. pandoc
   - If you use a WB computer, please use the Windows `.zip` installation file instead of the `.msi` installer and take note of the location where you unzip the Pandoc folder
2. TeX/LaTeX

# Installation

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:
    - WB computers: use the path where you unzipped Pandoc
    - Non-WB computers: open the command line and type `where pandoc`
  - On Mac or Linux: on the terminal, type `which pandoc`

# Stata setup

1. Go to the location where you unzipped the file `stata-markdown-CE-session.zip`
2. Open `Master.do`

# Stata setup

```
/*******************************************************************************
    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"
```

# Stata setup

1. Paste the location of pdflatex to the line that starts with `whereis pdflatex`
   - Example: *"C:/Program Files/MiKTeX 2.9/miktex/bin/x64/pdflatex.exe"*
   - **On Windows:** Make sure to include the `.exe` file extension
2. Paste the location of Pandoc to the line that starts with `whereis pandoc`
   - Example: *"C:/WBG/pandoc-2.11.4/pandoc.exe"*
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

In our case, we used:

```
markstat using stata-markdown-template, pdf
```

`stata-markdown-template` is the file with the extension `.stmd` you will find in the materials folder.

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

## Exercise 1:

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

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 on Exercise 1:

- 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 using `markstat`, 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`, for example
- Check the file called `Stata markdown template.stmd` on your Do-file editor to see how it works
- To write text, write markdown without indentation
- Check `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

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

# Including Stata output

````
```{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)
````

# Including Stata graphs

## Exercise 3

1. Go to `Stata markdown template.stmd`
2. Create and export a graph using the loaded data (you can use the command `scatter`, for example)
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)
```

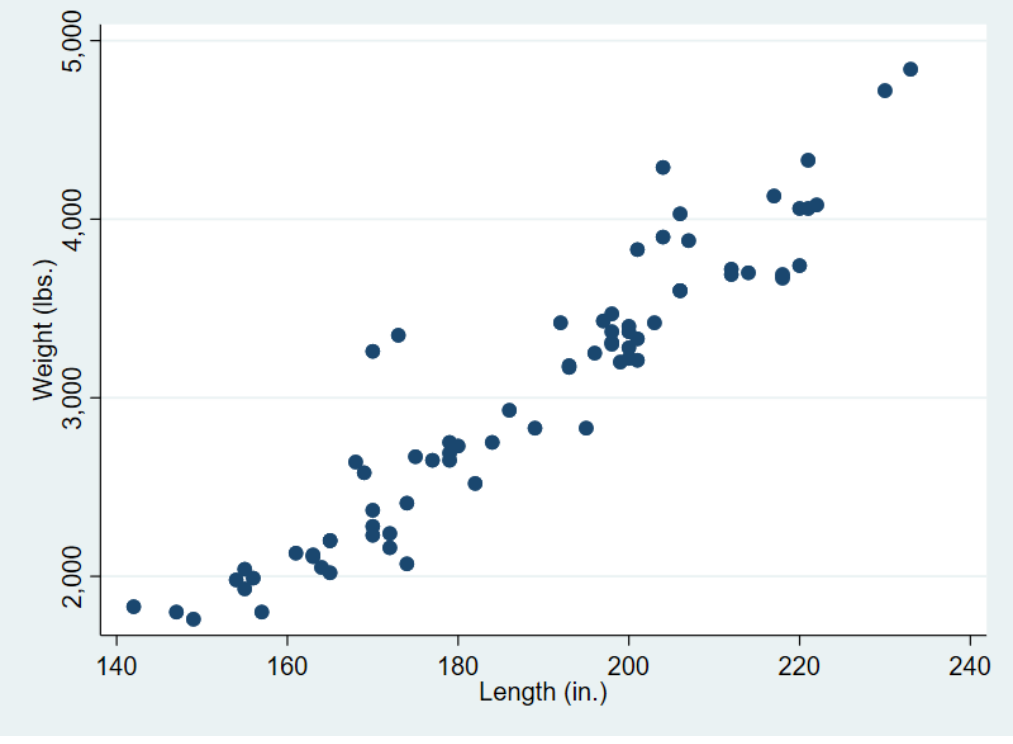



Figure 1: Correlation between weight and length

# 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)
```

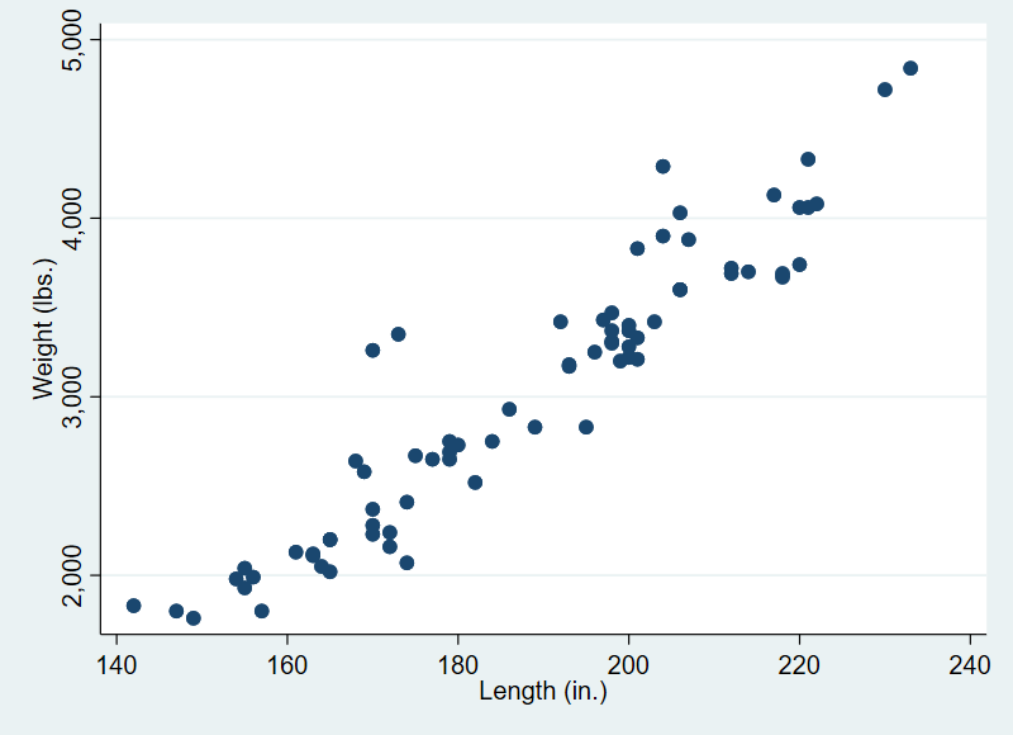



Figure 2: Correlation between weight and length

# Omitting Stata output

- Now, the last slide probably wasn't exactly what you were expecting, right?
- Using `{s/}` will 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

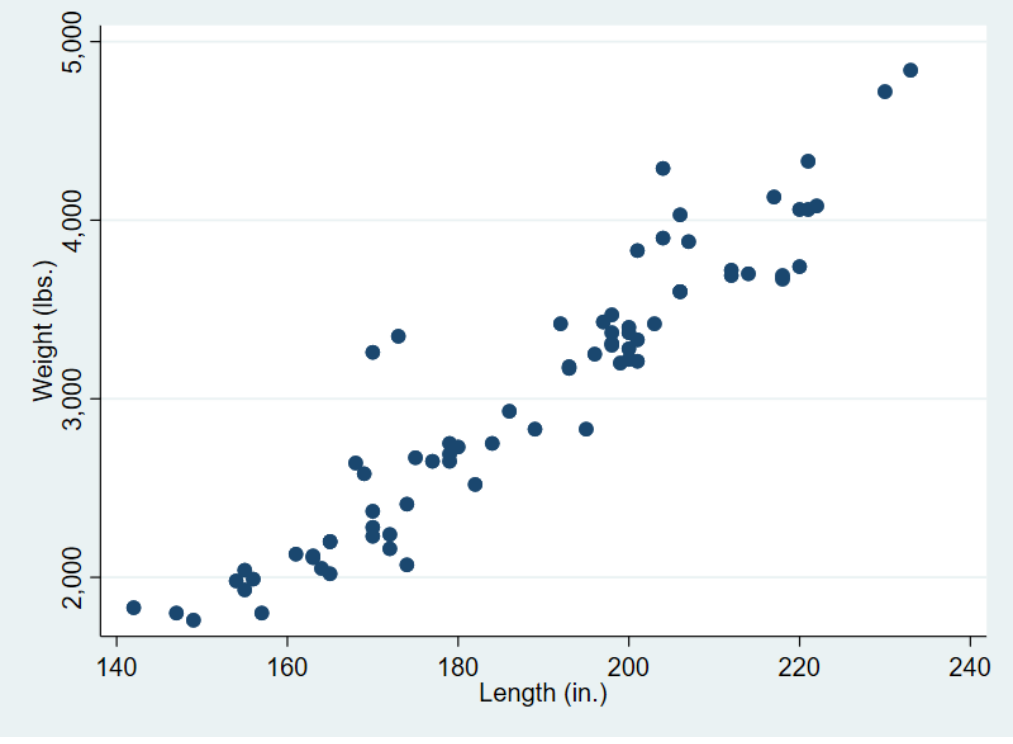



Figure 3: Correlation between weight and length

# 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 10 Feb 2021.

# 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
- 22 are foreign models, and 52 are domestic

# 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 |

# Annex: 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
```

# Annex: Other tools for dynamic documents in Stata

**texdoc**

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

# Annex: Other tools for dynamic documents in Stata

**Stata Markdown**

- Tools built-in Stata 15 or newer versions
- `dyndoc`: create Word or HTML files from Markdown
- `putpdf`: create PDF files
- Syntax is different for different output formats
- Syntax is unique to these commands
- The markdown file is not as easy to read

# Annex: Other tools for dynamic documents in Stata

- E.F Haghish has a few different commands for dynamic documents in Stata.
- This presentation is great introduction to them

**MarkDoc**

- Translates log files to Markdown, doc, LaTeX, HTML, epub and a number of other formats

**weaver**

- Real-time preview of the document

# Annex: 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