Dynamic documents in Stata

February 21, 2019

1 Introduction

A dynamic document is baiscally a document that is updated whenever your results are updated. In the last few sessions, we've been using LaTeX to create documents that input files created by Stata code, and every time the Stata outputs are updated, we need to re-compile the LaTeX file and the results in the document are also updated.

In this session, we are going to cut one step in this process: we are going to write code in Stata that creates the outputs, writes them into a LaTeX file and compiles that file. So once you run the code, you get the results in the Stata window, but also a PDF document with them.

1.1 Options for dynamic documents in Stata

- Stata markdown
- `markstat`
- `texdoc`

1.2 When to use dynamic documents with Stata

If you are exporting outputs from Stata, you most definitely need dynamic documents. I mean, the other option is to just copy and paste results as many times as you update your document, and that's not an efficient use of your time. The question is: do you want to use a specific software to write the texdocument, or do you want to write it in Stata? Here are the things to weight when choosing:

1.2.1 Pros

- As I said earlier, it saves you steps in the process
- It allows you to include code in the document

1.2.2 Cons

- It's harder to debug LaTeX code when writing within Stata
- Harder to format text
- There's no LaTeX syntax highlighter in Stata

1.2.3 So when do you use them?

- Need to include code, text and outputs (like this document)
- Compile tables exported to LaTeX
- Simple documents showing results
- Writing complex documents, if you're a LaTeX genius

2 Setup

The first thing you need to create this documents is a TeX compiler, such as MiKTeX. Because we used them in previous sessions, I'm assuming you already have that. Here are the necessary Stata commands:

```
* Install the necessary packages
ssc install texdoc, replace             // Write Stata and tex in a do-file
ssc install texify, replace             // Compile LaTeX file from Stata
```

You will also need to download a Stata tex style for LaTeX to format the Stata code as it looks like in Stata. Here's how you do that:

```
* Change current directory to the directory where the .tex file will be saved
cd      "FOLDER/PATH/HERE"
* Copy the style to the folder path
copy    https://www.stata-journal.com/production/sjlatex/stata.sty      stata.sty
```

As you can see, this just download a file called `stata.sty` to whichever folder you pick. This file needs to be saved in the same folder as the .tex file you are creating. It can also be saved in your tex search tree, so you don't need to have it in the same folder as every dynamic document you write in Stata. But I haven't managed to figure out where that is exactly.

Now, here's how this work: the first command we'll use is `texdoc`. To use it, you will write a do file that contains both TeX code and Stata code. TeX code will be preceeded by `/***` and suceeded by `***/`. Stata code will be preceded and suceeded by some form of the `texdoc` command, which we will explore soon. Finally, the do file will start by initializing the texdocument. We'll see all the `texdoc` options very soon. For now, all you need to know is that you write and save a do file that looks like this:

```
texdoc init "${raw}/main_doc.tex", replace
/***
\documentclass[a4paper]{article}
\usepackage{stata}
\usepackage{graphicx}
\usepackage{adjustbox}
\usepackage{float}
%-----------------------------------------------------------
\begin{document}
\title{Dynamic documents in Stata}
\maketitle
Hello world
```

```
\end{document}
***/
```

After you write that do file, you will use yet another subcommand of `texdoc` to tell Stata to read such do file and use it to save a `.tex` file. Finally, you are going to use the `texify` command to compile the `.tex` file, which will make a PDF document pop up.

You can set up a master do file that does all this, and it will look like the one below. Just keep in mind that you need to close the PDF file before compiling it again.

```
* Set folder paths
global  project_folder  "GITHUB/FOLDER/HERE"
global  do                      "${project_folder}/do"
global  output  "${project_folder}/output"

* Create document
texdoc do              "${do}/main_doc.do", replace    // Creates a .texdocument
texify                  "${output}/main_doc.tex"        // Turns it into a PDF document
```

3 Printing code and output in the document

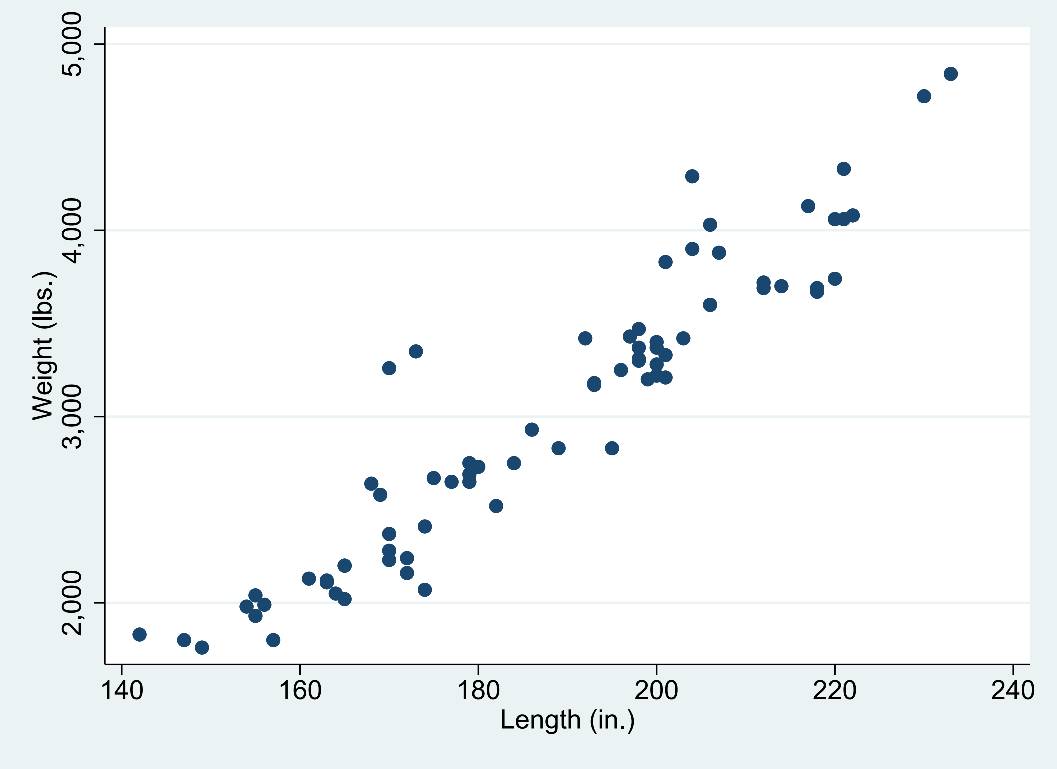
```
\begin{document}
\title{Dynamic documents in Stata}
\maketitle
Hello world
texdoc stlog
* Load the data set
sysuse auto, clear
* Summarize the variables
sum weight length

texdoc stlog close
\end{document}
```

Will give you the following results:

```
.
. * Load the data set
. sysuse auto, clear
(1978 Automobile Data)

.
. * Summarize the variables
. sum weight length

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      weight |         74    3019.459    777.1936       1760       4840
      length |         74    187.9324    22.26634        142        233

.
```

4 Adding graphs

doc stlog doc stlog close



```
.
. texdoc stlog

.
. * Create a graph in Stata
. scatter weight length

.
. texdoc stlog close

.
. * Print the graph in the document
.
```

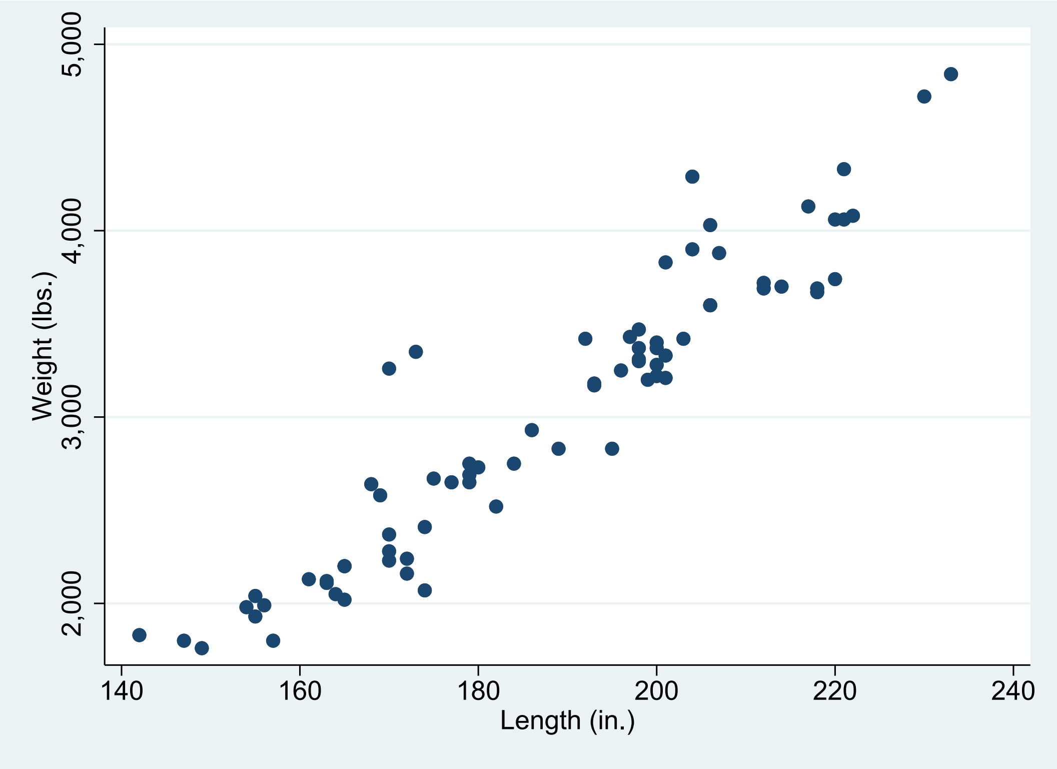
Here are some useful options to use with the graph:

- `figure(h!)`:
  hide command lines
- `label(f1)`:
  don't run the Stata commands
- `caption(A scatter plot)`:
  display a copy of the commands instead of an output log
- `optargs(width=0.8\textwidth)`:

The options can be specified for each chunk of code, or in the beginning of the do file to apply to all chunks.

5 Adding text

You can write and format text between chunks of code by adding
doc stlog doc stlog close



```
/***
\vspace{.5cm}
The graph below shows the correlation between weight of a car in pounds and its length in inches:
***/
texdoc stlog
* Create a graph in Stata
scatter weight length

texdoc stlog close
* Print the graph in the document
```

To format the text and create sections, write LaTeX code :
doc stlog doc stlog close doc graph

```
/***
\vspace{.5cm}
```

```
\section{Adding a graph}
The graph below shows the correlation between weight of a car in pounds and its length in inches:
***/

texdoc stlog

* Create a graph in Stata
scatter weight length

texdoc stlog close

* Print the graph in the document
texdoc graph
```

6 Stata output

The following options can be useful:

- `texdoc stlog,` *cmdstrip* :
  hide command lines
- `texdoc stlog,` *nodo* :
  don't run the Stata commands
- `texdoc stlog,` *cmdlog* :
  display a copy of the commands instead of an output log
- `texdoc stlog,` *nolog* :
  don't include the log in the LaTeX document. This is useful when you are doing things like changing variable labels
- `texdoc stlog,` *nooutput* :
  suppress command output in the log

The options can be specified for each chunk of code, or in the beginning of the do file to apply to all chunks.

7 Adding tables

To add tables to the document, it's necessary to export the table to tex format and then input them into the document. Like this:

```
texdoc stlog, nolog

reg price headroom
est sto reg1

regress price headroom trunk
est sto reg2

regress price headroom trunk foreign
est sto reg3

esttab reg1 reg2 reg3 using "${raw}/reg_table.tex", ///
        replace ///
        label se ///
        nomtitles

texdoc stlog close
texdoc write \input{reg_table.tex}
```

The result is

	(1)	(2)	(3)
Headroom (in.)	399.2	-580.8	-519.7
	(408.2)	(519.5)	(516.9)
Trunk space (cu. 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

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$