texdoc 2.0

An update on creating LaTeX documents from within Stata

Ben Jann

University of Bern, ben.jann@soz.unibe.ch

2016 Belgian Stata Users Group Meeting
Brussels, September 6, 2016

Outline

- Motivation
- The `texdoc` command
- Examples
- Limitations

Motivation

- As Stata users, we create many documents that include pieces of Stata output, graphs, or other Stata results in one way or the other.
- Manual inclusion of such elements in documents can be tedious and error prone.
- Good—and efficient—practice is to automate such tasks.
- Some candidates for automation:
  - Yearly reports with a given structure but changing results
  - Research articles containing tables and graphs
  - Documentations of datasets or data analyses
  - Stata Journal articles illustrating the use of Stata commands
  - Stata Press books or other textbooks
  - Solutions to Stata exercises
  - Presentations and class notes

Motivation

- There are two main reasons for automation.

1. Efficiency

- Do manual work only once.

2. Reproducibility

- As scientists, we want complete documentation of data production and data analysis.
- Automation makes errors less likely (and makes the detection of errors more likely).
- As a side effect, automation leads to standardization, which is usually a good idea for high quality and reliable science.

The texdoc command

- `texdoc` is a command that supports such automation.
- With `texdoc` you can maintain a single do-file that contains
  - the Stata code of your data analysis and
  - the text for your report/article/book etc.
- Processing the do-file with `texdoc` will run the analysis and create the source file of your document, containing text and results.
- `texdoc` is for use with LaTeX.
  - LaTeX has a somewhat steep learning curve, but is very flexible once you master it.
  - The end-product usually is a PDF. Hence, `texdoc` is not a tool, for example, for producing websites.
- `texdoc` has been around for some time.
  - Earlier versions, however, were only useful for small/simple documents.
  - The new version has many improvements and additional features.
  - The most important new feature is the possibility to turn Stata code on an off.

The texdoc do command

- The basic procedure is to write a do-file including Stata commands and sections of LaTeX code and then process the do-file by:

  `texdoc do` *filename* [, *options*]

- The output of `texdoc do` will be a source file that can then be processed by a LaTeX compiler to generate the final document.
- To facilitate the workflow, a good idea is to set up a keyboard shortcut in your text editor, say Ctrl+R, that grabs the current do-file and processes it by `texdoc do`.
- `texdoc do` can be nested. In complex documents it may be desirable to include parts of the code in separates files. Use `texdoc do` to call these files within your master do-file. This also works if the master do-file itself is processed by `texdoc do`.

Structure of a texdoc do-file

- The basic structure of a do-file to be processed by `texdoc do` is

```
texdoc init [docname] [, options]
... Stata commands ...
/***
... LaTeX section ...
***/
... Stata commands ...
/***
... LaTeX section ...
***/
etc.
texdoc close
```

Structure of a texdoc do-file

- The command

```
texdoc init [docname] [, options]
```

  initializes the LaTeX document and specifies general settings.
  - *docname* is the name of the LaTeX file be written to
  - *options* may be used, e.g., to specify folders for log files and graphs and determine the rules for naming the files. Furthermore, the default behavior of the `texdoc stlog` (see below) can be set.
  - `texdoc init` can be applied repeatedly within a do-file (omitting *docname*) to change the settings between different sections of the do-file.

- The command

```
texdoc close
```

  closes the LaTeX document. As `texdoc do` automatically closes the LaTeX document, `texdoc close` is usually not needed.

Structure of a texdoc do-file

- Use

```
/***
... LaTeX section ...
***/
```

to included a section of text and LaTeX code in the document. You may also type

```
/*tex
... LaTeX section ...
tex*/
```

The text within such a section will not be interpreted by Stata. That is, you cannot use Stata macros within such a section.

Including output from Stata commands

- The syntax to include output from Stata commands in the LaTeX document is

```
texdoc init [docname] [, options]
...
texdoc stlog [name] [, options]
... Stata commands ...
texdoc stlog close
...
texdoc close
```

  - All output form the commands between `texdoc stlog` and `texdoc stlog close` will be written to a separate log file that is then included, with proper formatting, in the LaTeX document.
  - You may provide a stable *name* for the output section or have `texdoc` make a name up on the fly.

Including output from Stata commands

- The *options* of `texdoc stlog` determine what exactly is done with the commands in the output section.
- Some options are:
  - `nodo` to skip executing the commands. This is an extremely useful option as it allows you to skip rerunning the commands once an output section is all set.
  - `cmdstrip` to remove the command lines form the output (i.e. only print the output without commands).
  - `cmdlog` to print the Stata code instead of a Stata log.
  - etc.
- All options can also be specified with `texdoc init` to set the default behavior. Each option has a complementary form so that the chosen defaults can be overridden.
  - For example, specify option `nodo` with `texdoc init` to turn all commands off, but then specify option `do` with `texdoc stlog` to turn the commands back on in a specific output section.

The logall option

- Alternatively, if you want to automatically include all Stata output in the LaTeX document, you can use the `logall` option:

```
texdoc init [docname], logall [options]
/***
... LaTeX section ...
***/
... Stata commands ...
/***
... LaTeX section ...
***/
... Stata commands ...
etc.
texdoc close
```

Including graphs

- Graphs created within a `texdoc stlog` section can be included in the document as follows:

```
texdoc stlog [name] [, options]
... Stata commands creating a graph ...
texdoc stlog close
texdoc graph [name] [, graph_options]
```

  - By default, `texdoc graph` exports the graph from the topmost graph window and includes code in the LaTeX document to display the graph.
  - `texdoc graph` takes account of the settings of `texdoc stlog`. For example, if the `nodo` option has been specified (and, hence, no graph was created), `texdoc graph` only includes appropriate code in the LaTeX document without trying to export the graph.

Including graphs

- *graph_options* determine how the graph is exported and how it is embedded in the LaTeX document. Default graph options can also be specified with `texdoc init`.
- Some options are:
  - `as(`*fileformats*`)` to set the output format(s). The default is `as(pdf)`.
  - `name(`*name*`)` to specify the name of the graph window to be exported.
  - `optargs(`*args*`)` to pass optional arguments through to the LaTeX graph command.
  - `figure[(`*args*`)]` to include the graph in a (floating) figure environment.
  - `caption(`*string*`)` to provide a caption for the figure.
  - `label(`*string*`)` to provide a cross-reference label for the figure.
  - etc.

Some further commands

- LaTeX:
  - `texdoc write` *textline* to write a single line of LaTeX code. Stata macros within *textline* will be interpreted.
  - `texdoc append` *filename* to include LaTeX code from an external file.
- Output sections:
  - `texdoc stlog` [*name*] `using` *do-file* [`,` *options*] to include Stata output from an external do-file.
  - `texdoc stlog oom` *command* to suppress output from a command and include an output-omitted tag.
  - `texdoc stlog cnp` to include a continued-on-next-page tag.
- Other:
  - `// texdoc exit` to exit a texdoc do-file.
  - `texdoc strip` *filename newname* to remove all `texdoc` elements from a do-file.

Examples

Some limitations

- Much effort has been put into making `texdoc` general and robust (for example, inline comments or commands such as `cd` or `clear all` do not disturb `texdoc`).
- Nonetheless, there are a number of limitations. Some of these limitations are:
  - `texdoc` commands should always start on a new line, with `texdoc` being the first (non-comment) word on the line.
  - `texdoc` only provides limited support for the semicolon command delimiter. Do not use semicolons to delimit `texdoc` commands.
  - `texdoc` does not parse the contents of a do-file that is called from the main do-file using the `do` command. Use `texdoc do` to include nested do-files.
  - `texdoc` closes the default log if it is on. Use a named log to log a Stata session in which `texdoc` is applied.

Paper and software

- Jann, Ben (2016). Creating LaTeX documents from within Stata using texdoc. The Stata Journal 16(2): 245-263.
  - `http://www.stata-journal.com/article.html?article=pr0062`
  - `http://ideas.repec.org/p/bss/wpaper/14.html` (working paper)
- texdoc website
  - `http://repec.sowi.unibe.ch/stata/texdoc`
- Installation:
  - In Stata type:

```
. ssc install texdoc
. net install sjlatex, from(http://www.stata-journal.com/production)
```

  - To compile a LaTeX document containing Stata output you also need to install the Stata LaTeX files on your system and load the `stata` package in your LaTeX document (`\usepackage{stata}`).
  - In Stata, use the `sjlatex install` command to download and install the Stata LaTeX files (either to the working directory or to the local search tree of your LaTeX installation).

New webdoc command

- texdoc clone for creating HTML pages
- provides various additional features relevant for HTML (headers, automatic TOC, Base64 images, ...)
- see `http://repec.sowi.unibe.ch/stata/webdoc`