Creating LaTeX and HTML documents from within Stata using textdoc and webdoc

Ben Jann

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

Swiss Stata Users Group meeting
Bern, November 17, 2016

Outline

- Motivation

- The `texdoc` and `webdoc` commands
    - Usage of `texdoc`
    - Examples
    - Additional info on `webdoc`

- Limitations

- Documentation and software

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.

Motivation

- 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
    - ▶ Websites . . .
        - ★ . . . reporting results computed by Stata
        - ★ . . . documenting datasets or data analyses
        - ★ . . . documenting the use of Stata commands

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 and webdoc commands

- `texdoc` and `webdoc` are commands that support such automation.
  - `texdoc` is for LaTeX, the final product usually being a PDF
  - `webdoc` is for HTML (or Markdown)

- With `texdoc`/`webdoc` you can maintain a single do-file that contains
  - the Stata code of your data analysis and
  - the text for your report/article/book/website etc.

- Processing the do-file with `texdoc`/`webdoc` will run the analysis and create the source file of your document, containing text and results.

Usage of texdoc

- 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 <u>init</u> [*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.
  - ▸ If texdoc init is omitted, texdoc do will automatically initialize the output document using the name of the do-file.

Structure of a texdoc do-file

- Inserts such as

  ```
  /***
  ... LATEX section ...
  ***/
  ```

  are used to included sections of text and LATEX code in the
  document. The sections will be copied to the output document as is
  (without expanding Stata macros).

- The command

  ```
  texdoc close
  ```

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

Including output from Stata commands

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

  ```
  ...
  texdoc stlog [name] [, options]
  ... Stata commands ...
  texdoc stlog 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 in the LaTeX document with proper formatting.
  - ▶ 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 substitute` *from to* `...` to define substitutions that will be applied within /*** ***/ blocks.
  - `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` [*name*][, *options*]: *command* to include the output from a single Stata command.
  - `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.

Some further commands

- Dynamic text:
  - `texdoc local` *name definition* to define a local macro that will be backed up on disk. Macros defined by `texdoc local` ...
    - ⋆ will be restored from disk if necessary (i.e. if the `nodo` has been applied)
    - ⋆ will be expanded within subsequent `/*** ***/` blocks
- Other:
  - `// texdoc exit` to exit a texdoc do-file.
  - `texdoc strip` *filename newname* to remove all texdoc commands and LATEX blocks from a do-file.

Examples

- Example 1: Writing an article

- Example 2: Documenting a data analysis

- Example 3: Literate programming and software certification

Usage of webdoc

- `webdoc` is very similar to `texdoc`, but the output is a HTML file instead of a LaTeX file.
- `webdoc` also has a number of useful extra features.
  - `webdoc init` provides a `header` option.
    - ⋆ Basic CSS settings.
    - ⋆ Support for Bootstrap (including Bootswatch themes).
    - ⋆ Color schemes for Stata output.
  - `webdoc toc` creates a table of contents.
  - `webdoc stlog` has an `sthlp` option to translate help files including clickable navigation.
  - `webdoc stlog` has a `dosave` option to create a do-file from an output section.
  - `webdoc graph` can embed graphs in the output documents (using Base64 encoding or SVG).
  - `webdoc set` can be used to define custom HTML settings.

Examples

- Formatting
  - ▸ Basic CSS header
  - ▸ Stata color schemes
  - ▸ Bootstrap/Bootswatch
  - ▸ Math

- Stata output
  - ▸ Contents of output sections
  - ▸ Highlighting selected output
  - ▸ Setting the screen width
  - ▸ Displaying Stata code
  - ▸ Saving Stata code
  - ▸ Displaying help files

- Table of contents
  - ▸ Basic usage
  - ▸ Adding section numbers
  - ▸ Formatting the TOC

- Graphs
  - ▸ The hardcode option
  - ▸ SVG format
  - ▸ Cross-referencing

- Tables
  - ▸ Using esttab (or similar)
  - ▸ Improved esttab tables
  - ▸ Creating custom tables

- Using Markdown
  - ▸ Basic procedure
  - ▸ Table of contents

Some limitations

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

Documentation and software

- Working papers:
  - ► http://ideas.repec.org/p/bss/wpaper/14.html (texdoc)
  - ► http://ideas.repec.org/p/bss/wpaper/22.html (webdoc)

- Stata Journal:
  - ► Jann, Ben (2016). Creating LaTeX documents from within Stata using texdoc. The Stata Journal 16(2): 245-263.
  - ► Paper on webdoc under review.

- Online documentation:
  - ► http://repec.sowi.unibe.ch/stata/texdoc
  - ► http://repec.sowi.unibe.ch/stata/webdoc

Documentation and software

- Installation:
  - ▸ texdoc:

    ```
    . ssc install texdoc
    ```

    - ★ 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}). To obtain the Stata LaTeX files, first type

      ```
      . net from http://www.stata-journal.com/production
      . net install sjlatex
      ```

      to install the sjlatex package. After that, use command sjlatex install to download and install the Stata LATEX files (either to the working directory or to the local search tree of your LATEX installation).

  - ▸ webdoc:

    ```
    . ssc install webdoc
    ```