Dynamic Documents in Stata

Bill Rising

StataCorp LP

2016 Oceania Stata Users Group Meeting
University of Sydney
30 Sep 2016
The Good and Bad of Creating Documents

Think of documents you’ve made in the past, good and bad

Good:
- Reused ideas from one project for another
- Reused lessons for teaching
  - Better: polished lessons to shining perfection

Bad:
- Questions on methods for reaching particular numerical results
- Updating analyses because of new or improved data
- Producing repetitive reports
General Idea

- What gets done once often gets done twice
  - Similar projects
  - Updated datasets
  - Datasets arriving over time or from various sources
  - Teaching

- The second and later repetitions should not start from scratch
Dynamic Documents

- Needed: reproducible and reusable documents, aka dynamic documents
  - Documents should be reproducible
    - No magic required or desired
  - Documents should be reusable
    - This is especially necessary for teaching
- Both of these are easy for pure narratives
- Including computational results is trickier
- Making this nice for all collaborative parties is even trickier
One underlying file for producing a final document, including narrative and stats
- If not a single document, a single folder with easily-related files
- The final document can be reliably reproduced from scratch
- Drafts of the final document can be passed around to all collaborators
  - Topic experts as well as statistical experts as well as writers
  - Those comfortable with programmerish work and those who are not
- The final document could be in a variety of forms
  - As Ian Watson pointed out, there are different audiences who use different types of files
What We’ll See Here

- Several tools for producing dynamic documents
- Some way of deciding between complexity, completeness, and comprehension
Bare Necessities for Teaching

- Commands
- Results
- Graphs
Bare Necessities for Reports

- Results without commands
- Inline results
  - Results often show up within the narrative
- Invisible commands
We will look at several pieces of available (and soon-to-be available) software:

- `texdoc` for making documents which are like Stata Journal articles
- `StatWeave` for making general-purpose documents
- `Markdoc` for creating general-purpose documents in many formats
- `StatTag` for better collaborative documents
- A suite for producing lessons with handouts
### Terminology

- It will help to have some defined jargon here to refer to files
  - A *base* file gets processed by the software
  - The result of the processing is an *interim* file, if that file needs more processing
  - The document as it would be viewed will be called a *final* file
    - This is not final as in “final draft”
texdoc Basics

- texdoc was written and is maintained by Ben Jann
- texdoc produces PDF documents which look like Stata Journal articles
- Narrative and Stata code all go into a base do-file
- Narrative and \LaTeX\ code are in special /*** ... ****/ comments
- Code used in the document are in texdoc stlog and texdoc stlog close blocks
- Invisible code is typical Stata code
texdoc Process

- texdoc do does the do-file, scoops up results, and creates an interim \texttt{\LaTeX} file
  - It also creates a large series of small files, one for each block of code included in the document
- The final document is made by typesetting the \texttt{\LaTeX} file using \texttt{pdflatex}
texdoc Advantages

- Nice, clean output
  - Thanks to using the `stata` style and `stlog`

- Can refresh the `tex` document without running the underlying commands
  - This is done by adding the `nodo` option to the starting `texdoc init` command
  - This is made for fixing typos in an otherwise-completed document
**texdoc Disadvantages**

- Cannot have in-line results
- Base file is a bit ugly and can be hard to read unless the narrative is long
texdoc Installation

- Simple, because it can be done from within Stata
  
  . ssc install texdoc
texdoc Dependencies

- Requires the sjlatex package
  - Which is not bad, because it can be installed from within Stata
    . net from http://www.stata-journal.com/production
    . net install sjlatex
StatWeave Basics

- StatWeave was written by Russ Lenth
- StatWeave produces PDF documents
  - It once worked nicely from within Open Office (.odt) documents
  - This has been broken by updates to Open Office
- Narrative and code go in one \LaTeX\ or HTML base file
- Narrative and \LaTeX\ (or HTML) code is simply written
- All Stata code is in \begin{Statacode} \ldots \end{Statacode} blocks
StatWeave Process

- Statweave processes the \LaTeX\ file to make an interim \LaTeX\ file, and then runs \texttt{pdf\LaTeX} to make the final file.
- For HTML, it makes the final HTML in one step.
StatWeave Advantages

- StatWeave is document-centered, so it can mix code and results from Stata, SAS, R, Unix shells, and DOS
- It can produce both \LaTeX\ and HTML files
- Inline expressions are simple
- It can split commands and output
  - This is useful for producing slides and handouts from a single file
StatWeave Disadvantages

- The base file is a bit ugly
- While it can produce both \texttt{\LaTeX} and HTML, it can do this only with separate base files
- The output is not fancy-formatted
- The user must know \texttt{\LaTeX} and HTML well to produce documents
StatWeave Installation

- StatWeave can currently be downloaded from http://homepage.divms.uiowa.edu/~rlenth/StatWeave/
- This will change soon to a github site
- The installation is a little unix-like, including a configuration file
None yet—it is written in Java, so it is platform independent

As soon as it gets posted to github, it will need a package for manipulating log files

`. ssc install smcl2do`

Aside: this can create do-files from log files, stripping commands resulting in errors
MarkDoc Basics

- `texdoc` was written and is maintained by Haghish.
- MarkDoc produces pretty much any document you like from one base file.
- Narrative and code all go into a single do-file.
- Narrative and `LaTeX` code are in special `/***/` comments.
- Stata code is simply written.
- Special comments squelch either commands or output.
- MarkDoc uses Markdown for formatting.
  - Though equations can be included.
MarkDoc Process

- Simply do the special do-file
- Then feed the log file to `markdoc`
- Use the `export()` option to say what format you would like for the final document
MarkDoc Advantages

- It is possible to get either PDF or HTML files without knowing much of either \LaTeX\ or HTML
  - You do need to know Markdown, however, but it is easy
- Inline expressions are possible
- It produces many types of files
  - This comes from using Pandoc
MarkDoc Disadvantages

- Its author makes many small changes, meaning that the software for reproducible documents itself is not reproducible.
- Because it uses Pandoc, it's PDF output is not as carefully controlled.
- It really likes PNG files over all other graphics formats.
MarkDoc Installation

- The software can be installed from the SSC
  `. ssc install markdoc`
- Because of the dependencies, it takes a little effort
- MarkDoc can download the needed files if you like, but that can cause confusion if you have them installed elsewhere
MarkDoc Dependencies

- It requires a few pieces of open-source software
  - Pandoc
  - wkhtmltopdf
- It also requires one Stata package
  . ssc install weaver
  - There is an optional package which can also be installed to allow Stata syntax highlighting in the output file
    . ssc install statax
StatTag Basics

- `texdoc` was written and is maintained by Leah Welty, Luke Rasmussen, and Abigail Baldridge.
- StatTag is made for having a dynamic MS Word document which allows better collaboration by having a separate do-file and MS Word document.
- Narrative goes in a normal MS Word document.
- Code goes in a separate do-file.
- Results from the do-file need to be from display, a table, or a graph.
- There is a dialog box for linking the do-file to locations in the MS Word document.
StatTag Process

- Write the MS Word document as you normally would
- Link in results from your do-file
- Refresh
StatTag Advantages

- Collaborators see a simple MS Word document
  - Can comment and make changes, as with any other document
  - Changes get tracked
  - Document is easy to read because it doesn’t have Stata code sprinkled throughout
- Tables need to be reformatted but the formatting sticks even if the results change
- Is visually more natural for non-programmerish people
StatTag Disadvantages

- Only output from `display`, a highlighted table or a graph can be included
  - No good for teaching Stata or explicitly showing commands from Stata
- MS Windows-only
  - For now; Mac version due late in 2016
I have a nice, but strange, workflow for producing lessons and handouts

- It has no name yet, because it is not ready for sharing

- It is made for producing slides (such as these) and handouts for teaching training sessions

- All “narrative” is an outline item

- All code (invisible or not) is a comment

- Slides typically have just commands; handouts have commands and output

- Indexing and conditional materials are also supported
Producing Lessons Process

- Run a unix script to create a folder with all the proper template files
- Change some info files for the lesson
- Start typing in the outline, including code wherever
- Run an applescript to convert the outline into a beamer-friendly intermediate \LaTeX{} file
Producing Lessons Advantages

- Very fast lesson development
- Do not need to remember much \LaTeX
- Easy to update lessons after each Stata release
- Handouts which complement the slides well
- Can reorder lessons without worrying about redundant materials
Producing Lessons Disadvantages

- Cannot use for writing papers or reports
- Painful installation
  - Working on fixing this
- Requires expensive Mac-only software
Not publicly available yet

Partly because the installation needs to be smoothed out
Producing Lessons Dependencies

- OmniOutliner Pro from omnigroup.com
  - This is Mac-only and commercial
- A bunch of \LaTeX\ style files
  - Needed for conditional material, fancy slide numbering etc.
- Several Unix scripts
  - Needed mostly for combining several lessons into one training file
There are plenty of packages out there for making dynamic documents
  This must be something users are interested in
Most have quirks in their behavior
Some show great promise