Tips N' Tricks:
Reproducible Research:
Literate Programming and Dynamic Documents in Stata

Building on the last Tips N' Tricks document about Stata terminal or batch mode, and integration of external programming languages into Stata code and vice versa, this topic is about how to produce "dynamic documents", aimed at "reproducible research", using "literate programming" in Stata. Essentially, the idea is that you can have one main file (with possible sub-files) that includes text, code, images (charts/graphs/tables), and when run, produces a final, beautifully formatted document (.html, .doc, .pdf, .tex, etc.) which is publication quality, or nearly-so. Actually writing articles that integrate analyses and write-up is one possible use. Another possible use is simply to better integrate code and comments in a presentable way, especially for non-programmers like collaborators and advisors, who may review your notes. In this way, you can display only the underlying code you wish, hiding the more gory bits, and wrap it in well-formatted text. This type of dynamic document/reproducible research is quite common and popular in hard sciences, but less so in the social sciences.

Definitions

These three concepts (literate programming, dynamic documents, and reproducible research) are closely related. Here are some definitions:

> **Literate programming** is an approach to programming introduced by Donald Knuth in which a program is given as an explanation of the program logic in a natural language, such as English, interspersed with snippets of macros and traditional source code, from which a compilable source code can be generated.[1]
>
> **A living document or dynamic document** is a document that is continually edited and updated. A simple example of a living document is an article in Wikipedia, an online encyclopedia that permits anyone to freely edit its articles, in contrast to "dead" or "static" documents, such as an article in a single edition of the Encyclopdia Britannica.[2]
>
> **Reproducibility** is the ability of an entire experiment or study to be duplicated, either by the same researcher or by someone else working independently. Reproducing an experiment is called replicating it. Reproducibility is one of the main principles of the scientific method.
>
> The term **reproducible research** refers to the idea that the ultimate product of academic research is the paper along with the full computational environment used to produce the results in the paper such as the code, data, etc. that can be used to reproduce the results and create new work based on the research.[3]

[1] `https://en.wikipedia.org/wiki/Literate_programming`
[2] `https://en.wikipedia.org/wiki/Living_document`
[3] `https://en.wikipedia.org/wiki/Reproducibility`

Advantages

So what are some of the advantages?

1. All-in-one: Clean, readable annotated, analyses with integrated graphics
2. Updates: Automatically updated text/graphics/tables when underlying analyses change
3. Workflow: Keeps text formatting/fiddling to a minimum
4. Reproducibility: From dirty data to final draft
5. Correctness: Re-run analyses to ensure consistency
6. Transparency: Openness in quantitative research.
7. Open-Source Ethic: Leads to more innovative research.
8. Extensibility: Easier to modify, extend, reuse, mash-up: the data/analyses for new research
9. Time: Big investment up front saves time on the back-end
10. Record: Most importantly, a clear record of every step performed

Some Resources

http://www.stata.com/meeting/germany14/abstracts/materials/de14_rising.pdf
http://www.stata.com/meeting/italy08/rising_2008.pdf
http://www.haghish.com/talk/reproducible_analysis_using_stata.php
http://www.haghish.com/talk/reproducible_report.php

Some Additional Notes

Stata, R, Python

It should be noted that "reproducible research" is still quite limited in Stata, and a bit finicky. But it can still be done. It is currently much more advanced in languages like Python (iPython/Jupyter Notebooks) and R (RMarkdown, Knitr, and Weave).

LaTeX, HTML, Markdown

No matter whether you are working in Stata, Python, or R, you will need to master an additional coding language in order to produce "dynamic documents". This is because Stata/Python/R are programming languages used to perform data manipulations and statistical analyses. But with the proper tools, you can use Markdown/HTML/LaTeXto wrap nicely formatted text around your code, and insert images. Markdown is the easiest to learn, as it is essentially plain text with a few symbols for bold, underline, headings, etc. HTML is a tad more complex, and more flexible. LaTeXis a a beast in terms of capabilities and complexity, but you shouldn't be too scared of it. Even though the learning curve is steep, there are many useful templates, and once you have the formatting set, you are pretty much just writing plain text.

What are you waiting for? Check out the sample .do file for an example of a reproducible / replicable document produced with Stata.