https://github.com/mas802/statdoc

Markus Schaffner

Queensland Behavioural Economics Group, QUT

SUGM Oceania, 2015
The problem

- **Data/Research projects evolve over time.**
  - Working to deadlines,
    - Just get it done, clean up later (never).
    - Once it works, no need to keep it organised (until revision time).
  - **Collaboration:** different styles of working in teams. E.g. Inconsistency across commenting, abbreviation of commands,
  - **Multiple versions** of the same or similar files and folders . . .

⇒ Keeping projects organised requires a lot of effort.
Manual Solution

- Look at all data files: documents, script files one by one.
- Work back from output tables/graphs ⇒ which datafile, which variables, what transformations, what selections?
- Copy, merge, rerun, move . . . give up and start over.
Statdoc solution
Automagically document entire folders

- Inspired **professional programming** tools e.g. Javadoc.
- Scans **all files** similar to the manual approach to categorise, **visualise/digest content** and find the **links**.
- Can run **standalone or from within Stata** and produces a set of static html pages.
How to run Statdoc

A contained example

1. Stata ado to **install** from “https://mas802.github.io/statdoc/ado”
   - Output
2. **Restart** (as it is written in Java)
   - HTML open in browser (browse)
3. cd “project folder”
4. Run with the command: **statdoc**

```bash
. statdoc
Executing statdoc
with Stata in /Applications/Stats/
in directory /Users/mas/Dropbox/Stats_conference_2015/example

Statdoc generates automatical documentation for
input: /Users/mas/Dropbox/Stats_conference_2015/example
output: /Users/mas/Dropbox/Stats_conference_2015/example/statdoc
Version v8.1.1-beta.snapshot
Please be patient...

STATDOC: Copyright 2014-2015, Markus Schaffner
Apache License, Version 2.0

Stage 1 (reading files and data): Threads active: 1 remaining: 0
Stage 2b (resolve matching): Threads active: 0 remaining: 1
Stage 3 (templates): Threads active: 0 remaining: 4
Process complete in: 1 seconds.

Variables: 16 | Files: 13 | Tokens: 61

All done, copy the following URL into your browser:
file:///Users/mas/Dropbox/Stats_conference_2015/example/statdoc/index.html
```
What does Stadoc process?

- **Automatically discovered**
  - File (type):
    - Scripts, (load/save, log)
  - Data (variables),
  - Variables (descriptive statistics),
  - Tokens (index).

- **Additional manual documentation**
  - **data-files**: labels, notes, ...
  - **do-files**: Document comment /** */ , key Value outputs “@key value”
Key Inputs: Files

- **Customisable Typology** into Data, Scripts, Images, Documents and Others. With custom processors for subtypes e.g. Documents, Images, Log files (smcl).

- Scheduled for **further processing** (e.g. parsing scripts and reading data).
Key Inputs: Scripts (do files)
Links to other files, parsed and categorise commands

- Categorise all commands: descriptive, estimation, manipulation, input, output, system (color coded):

- Find other files used (use, import), produced (save, graph export) and called (do).
Key Inputs: Data Files
Overview statistics, details

- Produce static **descriptive overview** of all variables in each data file.
- **Smart classification** and **efficient processing**.
- Subsamples if necessary to run efficiently.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Graph</th>
<th>Type</th>
<th>N</th>
<th>Descriptives</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>[+ ] displacement</td>
<td><img src="image" alt="Graph" /></td>
<td>other (int)</td>
<td>74 (31)</td>
<td>$(\bar{x}=197.297, 91.837) [79, 425]$</td>
<td>Displacement (cu. in.)</td>
</tr>
<tr>
<td>[+ ] expensive</td>
<td><img src="image" alt="Graph" /></td>
<td>dummy (float)</td>
<td>74 (2)</td>
<td>&quot;0&quot;, &quot;1&quot;</td>
<td></td>
</tr>
<tr>
<td>[+ ] foreign</td>
<td><img src="image" alt="Graph" /></td>
<td>dummy (byte)</td>
<td>74 (2)</td>
<td>&quot;Domestic&quot;, &quot;Foreign&quot;</td>
<td>Car type</td>
</tr>
<tr>
<td>[+ ] gear_ratio</td>
<td><img src="image" alt="Graph" /></td>
<td>other (float)</td>
<td>74 (36)</td>
<td>$(\bar{x}=3.015, 0.456) [2.190, 3.890]$</td>
<td>Gear Ratio</td>
</tr>
<tr>
<td>[+ ] headroom</td>
<td><img src="image" alt="Graph" /></td>
<td>category (float)</td>
<td>74 (8)</td>
<td>&quot;1.5&quot;, &quot;2&quot;, &quot;2.5&quot;, &quot;3&quot;, &quot;3.5&quot;, &quot;4&quot;, &quot;4.5&quot;, &quot;5&quot;</td>
<td>Headroom (in.)</td>
</tr>
<tr>
<td>[+ ] length</td>
<td><img src="image" alt="Graph" /></td>
<td>other (int)</td>
<td>74 (47)</td>
<td>$(\bar{x}=187.932, 22.256) [142, 233]$</td>
<td>Length (in.)</td>
</tr>
<tr>
<td>[+ ] make</td>
<td><img src="image" alt="Graph" /></td>
<td>identifier (str18)</td>
<td>74 (74)</td>
<td></td>
<td>Make and Model</td>
</tr>
</tbody>
</table>
Key Inputs: Variables

Descriptive statistics, origin, usage

- Overview of descriptive statistics for variables with the same name in different dataset (compare).
- All usage of variable (incl. limited wildcard use) in script files.
- \(\implies\) Complete history/story of variables

### All variables with the name expensive

<table>
<thead>
<tr>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>[+] expensive @auto_merged.dta</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Graph Image]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>N</th>
<th>Descriptives</th>
</tr>
</thead>
<tbody>
<tr>
<td>dummy (float)</td>
<td>74</td>
<td>&quot;0&quot;, &quot;1&quot;</td>
</tr>
</tbody>
</table>

### Estimations and Descriptive Statistics with expensive

<table>
<thead>
<tr>
<th>file</th>
</tr>
</thead>
<tbody>
<tr>
<td>do/analysis.do: 27 [+] reg price turn turn2 foreign expensive</td>
</tr>
</tbody>
</table>

### Data Manipulations with expensive

<table>
<thead>
<tr>
<th>file</th>
</tr>
</thead>
<tbody>
<tr>
<td>do/prepare.do: 26 [+] gen expensive = 0</td>
</tr>
<tr>
<td>27 [+] replace expensive = 1 if price &gt; 7500</td>
</tr>
</tbody>
</table>
Customise
Open source, all files editable

- Most of the magic happens in **template files** that can be fully adjusted.
- **statdoc.properties** allows to customise almost everything.
Latest Feature: Dynamic Outputs
Leverage the statdoc templating engine with @statdocrun

- **@statdocrun** allows statdoc to run self-contained do files for you.
- Information can be stared in **key-value pairs**.
- This information can then be used in templates to produce **txt, html, tex, do files, anything text-based**.
- Very **powerful and flexible** on top of estout and others.

https://github.com/mas802/statdoc/wiki/Create-custom-output-files-using-@statdocrun
Statdoc empowered research project life-cycle

- Project start: EXPLORE
  - Find **all variables** and existing documentation.
  - Find **irregularities and documentation gaps**.
  - Inspect script files of **others**.

- Production phase: QUALITY CONTROL and ASSISTING
  - Find **irregularities and documentation gaps**.
  - Produce outputs with **@statdocrun**.
  - Make sure **documentation is kept uptodate**.
  - Facilitate **communication in the team**.

- Post-production/Revision: DOCUMENT and COMMUNICATE
  - Store **snapshots**.
    - Make sure **source and output** are **transparently** linked.
    - Easily **publish full documentation** for others to follow (citations!).
    - Easily **re-discover features** of the project.
Examples

- **EXAMPLE:**
  - Example used for presentation.
    - http://mas802.github.io/statdoc/example/

- **EXPLORE:**
  - Introduction to Stata Programming (Baum)
    - http://mas802.github.io/statdoc/itsp/
  - Merging multiple Micro and Macro datasets.
    - http://mas802.github.io/statdoc/merging/
  - ado files: http://mas802.github.io/statdoc/ado/

- **DOCUMENT:**
  - Cameron http://mas802.github.io/statdoc/cameron/
  - Allcott and Taubinsky, AER 2015
    - http://mas802.github.io/statdoc/allcott/

- more: http://mas802.github.io/statdoc/examples/
Next Steps

- **Dynamic** instead of static?
- **Deeper links** (html files, tex, full text).
- **Versioning**, integrate with git to document changes.
- More output generation with `@statdocrun`.
- Automatic **estimation analysis**, one page per regression (-table).
- Hints to **improve project quality**.
- ...
Thanks for your attention.

Questions?

https://github.com/mas802/statdoc

or Google: “statdoc stata”