



<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).

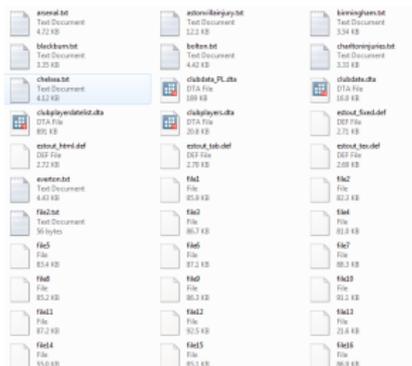
- ★ Once it works, no need to keep it organised (until revision time).

- ▶ **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.**

▼ do	Today 7:44 am	--	Folder
analysis.do	Today 7:44 am	559 bytes	Stata Do-file
describe.do	Today 7:42 am	366 bytes	Stata Do-file
prepare.do	Today 7:42 am	528 bytes	Stata Do-file
read_data.do	Today 7:42 am	490 bytes	Stata Do-file
▼ dta	Today 6:54 am	--	Folder
_DS_Store	Today 6:29 am	6 KB	Document
auto_merged.dta	Today 6:54 am	9 KB	Stata Data File
auto.dta	Today 6:54 am	6 KB	Stata Data File
autornd.dta	Today 6:54 am	4 KB	Stata Data File
▼ log	Today 7:26 am	--	Folder
analysis.smcl	Today 7:26 am	4 KB	Stata SMCL File
describe.smcl	Today 6:56 am	2 KB	Stata SMCL File
prepare.smcl	Today 6:54 am	1 KB	Stata SMCL File
read_data.smcl	Today 6:54 am	824 bytes	Stata SMCL File
▶ output	Today 7:32 am	--	Folder
▶ statdoc	Today 7:32 am	--	Folder
statdoc-0...encies.jar	Today 7:08 am	1.5 MB	Java JAR file



Manual Solution

- Look at all data files: documents, script files **one by one**.
- **Work back from output tables/graphs** \implies which datafile, which variables, what transformations, what selections?
- **Copy, merge, rerun, move ...** give up and start over.



image source: www.learningVideo.com

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.

Files summary
Data
auto.dta
auto_merged.dta
automd.dta
Scripts
analysis.do
describe.do
prepare.do
read_data.do
Images
price_histogram.png



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**

```
. statdoc
Executing statdoc
with Stata in /Applications/Stata/
in directory /Users/mas/Dropbox/Stata_conference_2015/example

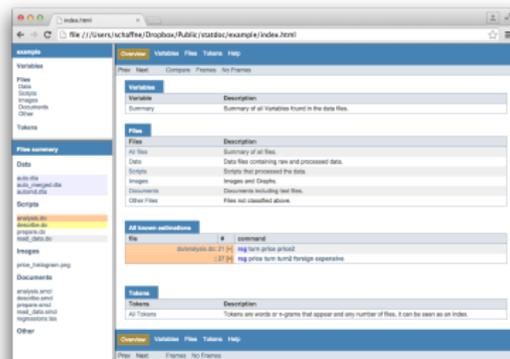
Statdoc generates autogical documentation for
input: /Users/mas/Dropbox/Stata_conference_2015/example
output: /Users/mas/Dropbox/Stata_conference_2015/example/statdoc
Version v0.9.1-beta.snapshot
Please be patient...

STATDOC: Copyright 2014-2015, Markus Scheffner
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: 4 remaining: 4
Stage 3 (templates): Thread active: 4 remaining: 7
Process complete in: 1 seconds.

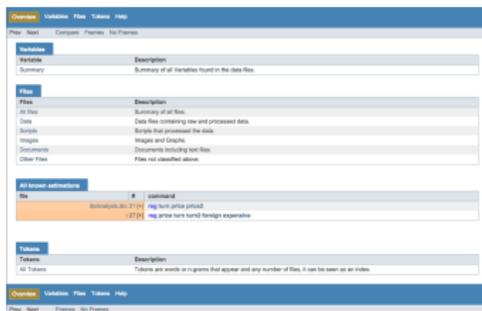
Variables: 16 | Files: 13 | Tokens: 61

All done, copy the following URL into your browser:
file:///Users/mas/Dropbox/Stata_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"`
-

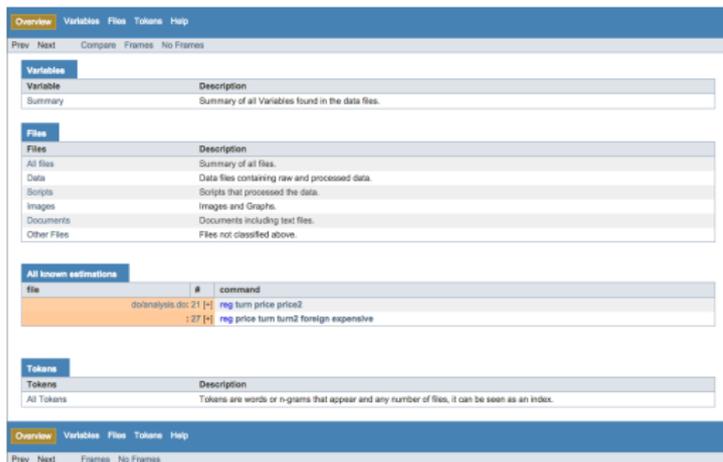


The screenshot displays the StataDoc interface with the following sections:

- Variables:** A table with columns 'Variable' and 'Description'. The 'Summary' row indicates a summary of all variables found in the data files.
- File:** A table with columns 'File' and 'Description'. Rows include 'All files' (summary of all files), 'Data' (data files containing raw and processed data), 'Scripts' (scripts that processed the data), 'Images and Outputs', 'Documents' (documents including text files), and 'Other files' (files not specified above).
- All known definitions:** A table with columns 'File', 'Label', and 'Description'. It lists 'BANKIND' (1-1) as 'log for price period' and 'PRICE' (1-1) as 'log price for bank foreign expenses'.
- Tokens:** A table with columns 'Token' and 'Description'. The 'All Tokens' row indicates that tokens are words or programs that appear and any number of files, and can be seen as an index.

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).



The screenshot displays a software interface with a blue header bar containing the menu items: Overview, Variables, Files, Tokens, and Help. Below the header, there are navigation options: Prev, Next, Compare, Frames, and No Frames.

The main content area is divided into several sections:

- Variables**: A table with two columns: Variable and Description. The first row shows 'Summary' with the description 'Summary of all Variables found in the data files.'
- Files**: A table with two columns: Files and Description. It lists categories: All files (Summary of all files), Data (Data files containing raw and processed data), Scripts (Scripts that processed the data), Images (Images and Graphs), Documents (Documents including text files), and Other Files (Files not classified above).
- All known estimations**: A table with three columns: file, #, and command. It shows two entries:

file	#	command
datanalys.doc 21 [H]		reg turn price price2
: 27 [H]		reg price turn turn2 foreign expensive
- Tokens**: A table with two columns: Tokens and Description. The first row shows 'All Tokens' with the description 'Tokens are words or n-grams that appear in any number of files, it can be seen as an index.'

At the bottom, there is another blue header bar with the same menu items: Overview, Variables, Files, Tokens, and Help, and navigation options: Prev, Next, Frames, and No Frames.

Key Inputs: Scripts (do files)

Links to other files, parsed and categorise commands

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

All commands		
#	content	log
8 [+]	<code>clear</code>	
	<code>// switch to the working directory</code>	
11 [+]	<code>cd "~/Dropbox/Stata_conference_2015/example"</code>	
13 [+]	<code>log using "log/analysis.smcl" , smcl replace</code>	
15 [+]	<code>use "dta/auto_merged.dta"</code>	
	<code>The main regression</code>	
21 [+]	<code>reg turn price price2</code>	
22 [+]	<code>estimates store r1</code>	

- Find other files used (use, import), produced (save, graph export) and called (do).

do/read_data.do:18	[16]	dta/auto.dta	
do/read_data.do:24	[18]	dta/autormd.dta	
	→	do/prepare.do	\
		log/prepare.smcl	[14]
		dta/auto_merged.dta	[29]
			do/analysis.do:15 do/describe.do:14



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.

All variables in auto_merged.dta					
Variable	Graph	Type	N	Descriptives	Label
[+] displacement @auto_merged.dta		other (int)	74 (31)	(\bar{x} =197.297, 91.837) [79,425]	Displacement (cu. in.)
[+] expensive @auto_merged.dta		dummy (float)	74 (2)	"0", "1"	
[+] foreign @auto_merged.dta		dummy (byte)	74 (2)	"Domestic", "Foreign"	Car type
[+] gear_ratio @auto_merged.dta		other (float)	74 (36)	(\bar{x} =3.015, 0.456) [2.190,3.890]	Gear Ratio
[+] headroom @auto_merged.dta		category (float)	74 (8)	"1.5", "2", "2.5", "3", "3.5", "4", "4.5", "5"	Headroom (in.)
[+] length @auto_merged.dta		other (int)	74 (47)	(\bar{x} =187.932, 22.266) [142,233]	Length (in.)
[+] make @auto_merged.dta		identifier (str18)	74 (74)		Make and Model

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					
Variable	Graph	Type	N	Descriptives	Label
[+] expensive @auto_merged.dta		dummy (float)	74 (2)	"0", "1"	

Estimations and Descriptive Statistics with expensive		
file	#	command
do/analysis.do: 27	[+]	reg price turn turn2 foreign expensive

Data Manipulations with expensive		
file	#	command
do/prepare.do: 26	[+]	gen expensive = 0
: 27	[+]	replace expensive = 1 if price > 7500

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.
-

 analyse-dta.do.vm	18/09/2015 7:27 AM	VM File	8 KB
 compare.vm	18/09/2015 7:27 AM	VM File	3 KB
 dumpdata.vm	18/09/2015 7:27 AM	VM File	1 KB
 file-item.vm	18/09/2015 7:27 AM	VM File	9 KB
 files-frame.vm	18/09/2015 7:27 AM	VM File	3 KB
 files-summary.vm	18/09/2015 7:27 AM	VM File	1 KB
 help-doc.vm	18/09/2015 7:27 AM	VM File	10 KB
 index.vm	18/09/2015 7:27 AM	VM File	3 KB
 item-footer.vm	18/09/2015 7:27 AM	VM File	3 KB
 item-header.vm	18/09/2015 7:27 AM	VM File	4 KB
 overview-frame.vm	18/09/2015 7:27 AM	VM File	2 KB
 overview-summary.vm	18/09/2015 7:27 AM	VM File	3 KB
 token-item.vm	18/09/2015 7:27 AM	VM File	2 KB
 tokens-frame.vm	18/09/2015 7:27 AM	VM File	2 KB
 tokens-summary.vm	18/09/2015 7:27 AM	VM File	1 KB
 variable-item.vm	18/09/2015 7:27 AM	VM File	4 KB
 variables-frame.vm	18/09/2015 7:27 AM	VM File	2 KB
 variables-summary.vm	18/09/2015 7:27 AM	VM File	5 KB
 VM_global_library.vm	18/09/2015 7:27 AM	VM File	14 KB

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 stored 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.

	KFZA_F1	KFZA_F2	KFZA_F3	KFZA_F4	KFZA_F5	KFZA_F6	KFZA_F7	KFZA_F8	KFZA_F9	KFZA_F10	KFZA_F11
stressindex	-0.462	-0.361	-0.327	-0.434	-0.449	-0.091	-0.106	-0.278	-0.400	-0.494	-0.327
doing_acting	0.339	0.242	0.292	0.200	0.205	0.337	0.205	0.188	0.230	0.335	0.184
thinking_analysing	0.251	0.122	0.223	0.256	0.400	0.021	0.291	0.173	-0.097	0.266	0.177
sensing_understanding	-0.312	-0.338	-0.238	-0.203	-0.162	-0.021	-0.088	-0.162	-0.469	-0.329	-0.370
SDnn	-0.253	-0.252	-0.233	-0.029	-0.110	0.078	0.034	-0.041	-0.387	-0.235	-0.113
rMSSD	-0.344	-0.337	-0.295	-0.105	-0.168	0.057	-0.003	-0.041	-0.418	-0.291	-0.201
log_LF_HF	0.430	0.400	0.327	0.299	0.325	0.060	0.176	0.188	0.357	0.398	0.341
pNN50	-0.372	-0.355	-0.333	-0.144	-0.206	-0.000	-0.035	-0.087	-0.449	-0.322	-0.212
total_power	-0.260	-0.275	-0.235	-0.012	-0.110	0.096	0.049	-0.022	-0.375	-0.231	-0.091
avg_BMP	0.113	0.023	0.099	-0.052	0.124	-0.145	0.025	0.013	0.048	0.088	-0.034
difference_day_night	-0.028	-0.057	0.072	-0.065	0.040	-0.055	0.192	-0.040	-0.134	-0.106	-0.246

<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/itasp/>

- ▶ 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**”

