

Dynamic document generation using Stata

Zhao Xu

StataCorp LLC

June 16, 2019

Creating documents

Common problems in creating a new document:

- Needing to copy Stata generated results back and forth.
 - Error prone and time-consuming

Creating documents

Common problems in creating a new document:

- Needing to copy Stata generated results back and forth.
 - Error prone and time-consuming
- Needing to update document due to new or improved data
 - Redo step 1

Creating documents

Common problems in creating a new document:

- Needing to copy Stata generated results back and forth.
 - Error prone and time-consuming
- Needing to update document due to new or improved data
 - Redo step 1
- Needing to create similar reports with similar formats frequently
 - Redo step 1

Creating documents

Common problems in creating a new document:

- Needing to copy Stata generated results back and forth.
 - Error prone and time-consuming
- Needing to update document due to new or improved data
 - Redo step 1
- Needing to create similar reports with similar formats frequently
 - Redo step 1
- All in all, document maintenance costs exist and matter

Dynamic documents

Dynamic documents are created on the fly and in real time which try to eliminate manual steps and adapt to new data sources easily.

- Document creation process driven by Stata commands

Dynamic documents

Dynamic documents are created on the fly and in real time which try to eliminate manual steps and adapt to new data sources easily.

- Document creation process driven by Stata commands
- Mixed formatted text and Stata output

Dynamic documents

Dynamic documents are created on the fly and in real time which try to eliminate manual steps and adapt to new data sources easily.

- Document creation process driven by Stata commands
- Mixed formatted text and Stata output
- Include inline Stata results

Dynamic documents

Dynamic documents are created on the fly and in real time which try to eliminate manual steps and adapt to new data sources easily.

- Document creation process driven by Stata commands
- Mixed formatted text and Stata output
- Include inline Stata results
- Embed Stata graphs

Dynamic documents

Dynamic documents are created on the fly and in real time which try to eliminate manual steps and adapt to new data sources easily.

- Document creation process driven by Stata commands
- Mixed formatted text and Stata output
- Include inline Stata results
- Embed Stata graphs
- Produce tables containing output from Stata commands

Benefits

- Easy to reproduce from the scratch at the push of a button

Benefits

- Easy to reproduce from the scratch at the push of a button
- Reusable and easy to share with others

Benefits

- Easy to reproduce from the scratch at the push of a button
- Reusable and easy to share with others
- Easy to maintain and update

Benefits

- Easy to reproduce from the scratch at the push of a button
- Reusable and easy to share with others
- Easy to maintain and update
- Save and convert to a variety of formats

Overview of official Stata commands

New commands in Stata 15

- **putdocx** - create docx documents
- **putpdf** - create PDF documents
- **dyndoc** - convert dynamic Markdown documents to HTML

putdocx: Creating docx documents

With **putdocx** command, you can:

- Write paragraph, Stata graphs, and tables to Word(.docx) files
- Various formats can be applied to paragraphs, texts, and tables
- Embed Stata results into Word paragraphs and tables
- Append multiple .docx files

Workflow

- Create a new document with **putdocx begin**
- Add new paragraphs with **putdocx paragraph**
- Include text with **putdocx text**
- Include graphs with **putdocx image**
- Add new tables with **putdocx table**
- Close and save the document with **putdocx save**

A quick example

`putdocx` can add formatted text to a paragraph. You can *italicize*, ~~strikeout~~, underline, sub/super script, and `shade`.

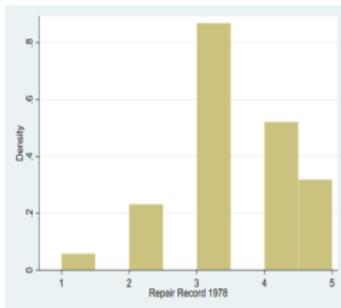


Figure 1: A Scatter Plot

Embed the output from a regression command.

mpg	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
price	-.0009192	.0002042	-4.50	0.000	-.0013263	-.0005121
_cons	26.96417	1.393952	19.34	0.000	24.18538	29.74297

```

quickdo x
1 sysuse auto, replace
2
3 putdocx begin
4
5 // Create a paragraph
6 putdocx paragraph
7 putdocx text ("putdocx "), bold
8 putdocx text ("can add formatted text to a paragraph. You can ")
9 putdocx text ("italicize, "), italic
10 putdocx text ("strikeout"), strikeout
11 putdocx text ("underline"), underline
12 putdocx text ("22"), sub/super script
13 putdocx text ("22"), script(sub)
14 putdocx text (" and ")
15 putdocx text ("shade."), shading("blue")
16
17 // Embed a graph
18 histogram rep
19 graph export hist.png, replace
20 putdocx paragraph, halign(center)
21 putdocx image hist.png, width(4in) linebreak
22 putdocx text ("Figure 1: A Scatter Plot"), font(, 14) bold
23
24 // Embed Stata output
25 putdocx paragraph
26 putdocx text ("Embed the output from a regression command.")
27 regress mpg price
28 putdocx table mytable = etable
29
30 putdocx save quick.docx, replace
31

```

Comments

- Document generation commands are mixed with other Stata commands
- Access Stata results directly, such as local/global macros, dataset, `r()`, `e()`, etc., and write them into the document
- Paragraphs, images, and tables can be programmably customized
- One point click and no manual intervention

Add tables to a Word document

- Add the table generated by Stata commands
 - Coefficient table of the last estimation command
 - Table of margins after the **margins** command
 - Table of results from one or more models displayed by **estimates table**.
- Add the current Stata dataset in memory as a table
- Add a Stata or Mata matrix as a table
- Create an empty table with specified dimensions
 - Build table from scratch
 - Need extra programming
 - Flexible and generic
 - Good for writing user-written commands

Add coefficient tables

```
. sysuse auto, clear  
(1978 Automobile Data)  
  
. reg mpg weight i.rep78, noheader
```

```
-----  
      mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]  
-----+-----  
      weight |   -.005503   .000601    -9.16   0.000    - .006704   - .004302  
      rep78 |  
        2 |   -.4786043  2.765035    -0.17   0.863    -6.004085   5.046877  
        3 |   -.4715623  2.553145    -0.18   0.854    -5.573614   4.63049  
        4 |   -.5990319  2.606599    -0.23   0.819    -5.807905   4.609841  
        5 |   2.086276   2.724817     0.77   0.447    -3.358836   7.531388  
      _cons |   38.05941   3.093361    12.30   0.000    31.87783    44.241  
-----
```

```
. putdocx table a = etable
```

Add marginal results

```
. webuse margex, clear  
(Artificial data for margins)  
  
. logistic outcome i.sex i.group sex#group  
  
. margins sex group
```

```
-----  
          |                Delta-method  
          |      Margin   Std. Err.      z    P>|z|    [95% Conf. Interval]  
-----+-----  
      sex |  
    male |   .1561738   .0132774   11.76   0.000   .1301506   .182197  
  female |   .1983749   .0101546   19.54   0.000   .1784723   .2182776  
          |  
      group |  
         1 |   .3211001   .0176403   18.20   0.000   .2865257   .3556744  
         2 |   .1152127   .0099854   11.54   0.000   .0956417   .1347838  
         3 |   .0265018   .0109802    2.41   0.016   .0049811   .0480226  
-----
```

```
. putdocx table b = etable
```

Add table of results by estimates table

```
. sysuse auto, clear  
(1978 Automobile Data)  
  
. regress mpg foreign weight headroom trunk length turn displacement  
. estimates store Model1  
. regress mpg foreign weight headroom  
. estimates store Model2  
. regress mpg foreign weight  
. estimates store Model3  
. estimates table Model1 Model2 Model3, stats(N r2) star b(%9.3f) stfmt(%9.3f)
```

Variable	Model1	Model2	Model3
foreign	-1.967	-1.655	-1.650
weight	-0.004*	-0.006***	-0.007***
headroom	-0.059	-0.219	
trunk	-0.012		
length	-0.063		
turn	-0.165		
displacement	0.001		
_cons	53.138***	41.993***	41.680***
N	74	74	74
r2	0.677	0.663	0.663

Legend: * p<0.05; ** p<0.01; *** p<0.001

```
. putdocx table c = etable
```

Add current dataset in memory as a table

- Order of table columns are determined by order of variables specified so that it is very flexible to change
- **if** and **in** qualifiers may be applied to export only observations that meet the specified conditions
- Useful for commands like **bootstrap**, **jackknife**, **rolling**, **statsby**, **collapse**, **contract**, etc

Add current dataset in memory as a table

```
. sysuse auto, clear  
(1978 Automobile Data)  
  
. statsby Total=r(N) Average=r(mean) Max=r(max) Min=r(min), by(foreign): summarize mpg  
. list, noobs
```

```
+-----+  
| foreign  Total    Average  Max  Min |  
+-----+  
| Domestic    52   19.82692   34   12 |  
| Foreign     22   24.77273   41   14 |  
+-----+
```

```
. putdocx table tbl1 = data("foreign Total Average Max Min"), varnames border(insideV, nil)  
. putdocx table tbl1(., .), halign(right)
```

Add Stata log files to a Word document

General idea:

- Open a log file and create a record of what you type and any output that produces using **log using** command
- Use **file** command to read the log line by line
- Write each line to the .docx file using **putdocx paragraph** and **putdocx text**

Add Stata log files to a Word document

```
log using log2docx.txt, text replace nomsg

sysuse auto, clear
regress mpg weight foreign

webuse byssin1
table workplace smokes race [fw=pop], by(sex) c(mean prob) format(%9.3f) sc col row

log close

putdocx clear
putdocx begin

file open fh using log2docx.txt, read
file read fh line

putdocx paragraph, font("Courier", 9.5)
while r(eof)==0 {
    putdocx text ("'"line'"), linebreak
    file read fh line
}
file close fh

putdocx save log2docx, replace
```

Put everything together

- Append contents in memory to an existing document

```
. putdocx save exist.docx, append
```

- Merge multiple existing documents to a single document

```
. putdocx append f1.docx f2.docx [...], saving(final.docx, replace)
```

putpdf: Creating PDF documents

- **putpdf** creates PDF documents directly
- No other dependency needed
- Similar syntax in kind to **putdocx**

Workflow

- Create a new document with **putpdf begin**
- Add new paragraphs with **putpdf paragraph**
- Include text with **putpdf text**
- Include graphs with **putpdf image**
- Add new tables with **putpdf table**
- Close and save the document with **putpdf save**

A quick example

```
sysuse auto, replace
putpdf begin

putpdf paragraph
putpdf text ("putpdf "), bold
putpdf text ("can add formatted text to a paragraph. You can ")
putpdf text ("italicize, "), italic
putpdf text ("strikeout, "), strikeout
putpdf text ("underline"), underline
putpdf text ("", sub/super script")
putpdf text ("2 "), script(sub)
putpdf text ("", and ")
putpdf text ("shade."), bgcolor("blue")

histogram rep
graph export hist.png, replace
putpdf paragraph, halign(center)
putpdf image hist.png, width(4in) linebreak
putpdf text ("Figure 1: A Scatter Plot"), font(, 14) bold

putpdf paragraph
putpdf text ("Embed the output from a regression command.")
regress mpg price
putpdf table mytable = etable

putpdf save quick.pdf, replace
```

dyndoc: Converting dynamic Markdown documents to HTML

- **dyndoc** converts a dynamic Markdown document containing formatted text and Stata commands to a HTML file

dyndoc: Converting dynamic Markdown documents to HTML

- **dyndoc** converts a dynamic Markdown document containing formatted text and Stata commands to a HTML file
- Markdown is a simple markup language based on plain text and is a simple way to make a structured document

dyndoc: Converting dynamic Markdown documents to HTML

- **dyndoc** converts a dynamic Markdown document containing formatted text and Stata commands to a HTML file
- Markdown is a simple markup language based on plain text and is a simple way to make a structured document
- Stata processes the Markdown text and Stata dynamic tags and creates the output HTML file

dyndoc: Converting dynamic Markdown documents to HTML

- **dyndoc** converts a dynamic Markdown document containing formatted text and Stata commands to a HTML file
- Markdown is a simple markup language based on plain text and is a simple way to make a structured document
- Stata processes the Markdown text and Stata dynamic tags and creates the output HTML file
- Stata dynamic tags allow Stata commands, output, and graphs to be interleaved with Markdown text

A quick example

```
. dyndoc dyndoc_ex.md, replace
```

- .md is the extension for Markdown files
- The creates the web page dyndoc_ex.html

Quick view of Stata dynamic tags

- `<<dd_do>>` executes a block of Stata code and optionally include its output
- `<</dd_do>>` ends `<<dd_do>>`

```
<<dd_do>>  
sysuse auto  
regress weight displacement  
<</dd_do>>
```

Quick view of Stata dynamic tags

- `<<dd_display [fmt] exp>>` includes output of Stata expression as shown by Stata's **display** command

```
<<dd_do>>  
summarize weight  
<</dd_do>>
```

The average weight of those cars is `<<dd_display %9.2f 'r(mean)'.>>`

Quick view of Stata dynamic tags

- `<<dd_graph>>` exports a Stata graph and include a link to the file

```
<<dd_do>>  
scatter mpg weight, mcolor(blue%50)  
<</dd_do>>
```

```
<<dd_graph: sav("graph.svg") alt("scatter mpg price") replace  
height(400)>>
```

dyntext: a tool for processing dynamic tags

- **dyntext** converts a dynamic text file containing both plain text and Stata commands to an output file in text format
- Stata process the dynamic tags in the dynamic text file
- It is useful for interacting with LaTeX and JavaScript files

dyntext: working with LaTeX

General idea:

- The original .tex file, say **sample.tex**, contains both LaTeX text and Stata commands embedded within the dynamic tags
- **dyntext** processes the dynamic tags, runs Stata commands and writes Stata results into a new .tex file, say **sample_res.tex**
- Use the new .tex file to generate the final PDF file

```
. dyntext sample.tex, saving(sample_res.tex) replace
```

Comments on dyndoc

- Results in the document come from executing Stata commands instead of being copied and pasted
- One single document and easy to use
- Any changes in data or in Stata are instantly reflected in the final document

Conclusion

- Three official commands to generate three kinds of documents
- Documents generated are reproducible and reusable
- There are plenty of user-written commands out there to generate dynamic documents
- You should really give them a try

Thank You!