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 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** 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 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 incline Stata results
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 incline Stata results
- Embed Stata graphs
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 incline 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
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

```plaintext
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| -.0013266 to -.0005121|
| cons | 26.96417  | 1.393952  | 19.34 | 0.000| 24.18538 to 29.74297 |
```

Zhao Xu

Dynamic document generation using Stata
Comments

- Document generation commands are mixed with other Stata commands
- Access Stata results directly, such as local/global macros, dataset, \texttt{r()}, \texttt{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 | I
  2 |  -0.4786043  2.765035  -0.17  0.863  -6.004085   5.046877
  3 |  -0.4715623  2.553145  -0.18  0.854  -5.573614   4.63049
  4 |  -0.5990319  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

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

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>sex</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>male</td>
<td>.1561738</td>
<td>.0132774</td>
<td>11.76</td>
<td>0.000</td>
</tr>
<tr>
<td>female</td>
<td>.1983749</td>
<td>.0101546</td>
<td>19.54</td>
<td>0.000</td>
</tr>
<tr>
<td>group</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>.3211001</td>
<td>.0176403</td>
<td>18.20</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>.1152127</td>
<td>.0099854</td>
<td>11.54</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>.0265018</td>
<td>.0109802</td>
<td>2.41</td>
<td>0.016</td>
</tr>
</tbody>
</table>

.putdocx table b = etable
```
Add table of results by estimates table

```stata
.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)

-----------------------------------------------------------
<table>
<thead>
<tr>
<th>Variable</th>
<th>Model1</th>
<th>Model2</th>
<th>Model3</th>
</tr>
</thead>
<tbody>
<tr>
<td>foreign</td>
<td>-1.967</td>
<td>-1.655</td>
<td>-1.650</td>
</tr>
<tr>
<td>weight</td>
<td>-0.004*</td>
<td>-0.006***</td>
<td>-0.007***</td>
</tr>
<tr>
<td>headroom</td>
<td>-0.059</td>
<td>-0.219</td>
<td></td>
</tr>
<tr>
<td>trunk</td>
<td>-0.012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>-0.063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>turn</td>
<td>-0.165</td>
<td></td>
<td></td>
</tr>
<tr>
<td>displacement</td>
<td>0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_cons</td>
<td>53.138***</td>
<td>41.993***</td>
<td>41.680***</td>
</tr>
</tbody>
</table>

-----------------------------------------------------------
| 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
```

Zhao Xu  Dynamic document generation using Stata
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

+-----------------------------------------+
<table>
<thead>
<tr>
<th>foreign  Total   Average   Max   Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domestic  52  19.82692  34   12</td>
</tr>
<tr>
<td>Foreign   22  24.77273  41   14</td>
</tr>
</tbody>
</table>
+-----------------------------------------+

.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 \texttt{log using} command
- Use \texttt{file} command to read the log line by line
- Write each line to the .docx file using \texttt{putdocx paragraph} and \texttt{putdocx text}
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"")
    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>>`

```stata
<<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

```plaintext
<<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 processes 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
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!