# sttex: A new dynamic document command for Stata and LaTeX

Ben Jann

University of Bern

2022 UK Stata Conference
London, September 8–9, 2022

# Outline

# Introduction

- Various utilities exist to weave Stata results into other documents.
- Examples for such dynamic document commands are official Stata's `dyndoc` and `dyntext` as well as user contributions such as
  - `texdoc` (Jann 2016) and `webdoc` (Jann 2017),
  - `log2markup` by Bruun (2016),
  - `markdoc` by Haghish (2016, 2020),
  - `stmd` by Henken (2019), or
  - `markstat` by Rodríguez (2017).
- As useful these tools are, I personally find them convenient mostly for small tasks.
- Main obstacle for productive use of the existing tools, in my opinion, is that all Stata commands are executed each time, making the workflow clumsy and slow.

  (Although there are some exceptions, such as the `do`/`nodo` option in `texdoc`/`webdoc`.)

# Introduction

- New command `sttex` solves the problem by keeping track of all pieces of Stata code in the document.
- Stata commands are only executed if there are changes in the commands (or in settings that require rerunning the commands). This makes it possible to work on the document in an efficient way.
- A single change in the code will still lead to execution of all commands. This can be avoided by partitioning the document into sections that can be run independently.
- `sttex` provides such functionality, including possible declaration of dependencies between sections.

# History

- 2009: first version of `texdoc`
- 2015/2016: major rewrite; introduction of `do`/`nodo` option
- summer 2016: development of `webdoc`
- fall 2016: not really satisfied with `textdoc`/`webdoc`; still too clumsy
- Friday morning, November 18, 2016, Bern: Discussion with Bill Gould on how to write a better program
- January/February 2018: development of `sttex`
- 2018–2022: use for own purposes (teaching, SJ papers, ...)
- September 2022: revision of `sttex`, finally wrote documentation; published on GitHub

# Syntax

- The syntax of the `sttex` command is

  `sttex` *sourcefile* [ , *options* ]

  where *sourcefile* is the LaTeX source file containing pieces of Stata code (default suffix is `.sttex`) and *options* are:
  - options for general behavior and typesetting (requires LaTeX)
  - *stlog-options* setting the defaults for Stata sections
  - `gropts(`*graph-options*`)` setting defaults for graphs

- `sttex` will parse the source file, run the Stata commands if necessary, and then weave the results into a target file with default suffix `.tex`.

- Various dynamic tags can be used within the source file. These tags are either dedicated LaTeX commands used by `sttex` or LaTeX comments that will be interpreted by `sttex`.

# Dedicated LaTeX commands

- To create a section of Stata or Mata output, type

  `\begin{`*keyword*`}[`*id*`] [`*stlog-options*`]`

  *Stata or Mata commands*

  `\end{`*keyword*`}`

  where *id* provides an optional name (an automatic name is used if *id* is omitted) and *keyword* is one of the following:

  | | |
  |---|---|
  | `stata` | run Stata commands and display the output |
  | `stata*` | run Stata commands without displaying the output |
  | `mata` | run Mata commands and display the output |
  | `mata*` | run Mata commands without displaying the output |

  *stlog-options* take precedence over log options specified at an upper level.

# Dedicated LaTeX commands

- To include a graph created by prior commands, type

  `\stgraph[`*id*`]{`*graph-options*`}`

  where *id* provides an optional name and where *graph-options* take precedence over graph options specified at an upper level.

  `\stgraph{}` must start at the beginning of a line.

- Inline expressions: To add strings and values of scalar expressions in the text, type

  `\stres[`*id*`]{`*display-directive*`}`

  Stata's `display` command will be applied to *display-directive* and the result will be inserted into the text.

  `\stres{}` can be specified anywhere inside a line of text, and it can also be specified multiple times in the same line. *display-directive* can span multiple lines.

# Interpreted LaTeX comments

- Initialization

  `%STinit` [ *targetfile* ] [ , *options* ]

  where *targetfile* specifies the name of the target file and *options* are as above; these options take precedence over the options specified with `sttex`.

  `%STinit` makes it possible to specify all relevant settings directly in the source file, such that `sttex` can be called without options.

  `%STinit` will only be recognized if specified at the beginning of one of the first 50 lines of the source file. In this case, lines before `%STinit` will be ignored.

# Interpreted LaTeX comments

- To create a new part, type

  `%STpart` [ *id* [ *parent* ] ] [ , *stlog-opts* `gropts(`*graph-opts*`)` ]

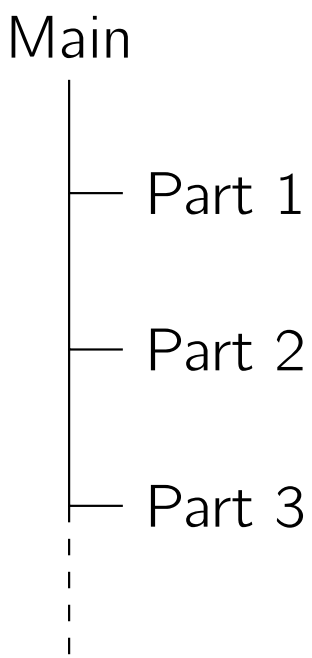
  where *id* provides an optional name. Specify *stlog-options* and *graph-options* to change default settings between parts.

  *parent* specifies the name of an optional parent part. A change in the parent part will cause execution of the code in the child, and vice-versa (i.e., a change in a specific part will cause the code in all its ancestors and all its descendants to be run).

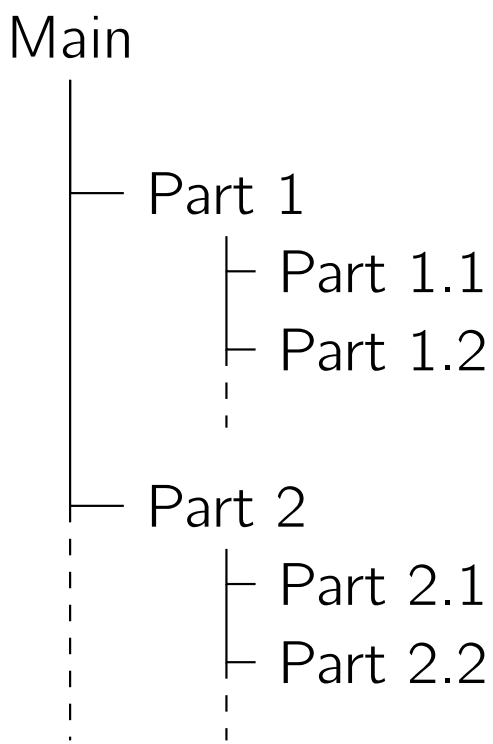
  By default, the (unnamed) main part is parent to all other parts. Specify *parent* as . (missing) to create a part that does not depend on the main part.

  `%STpart` must start at the beginning of a line.

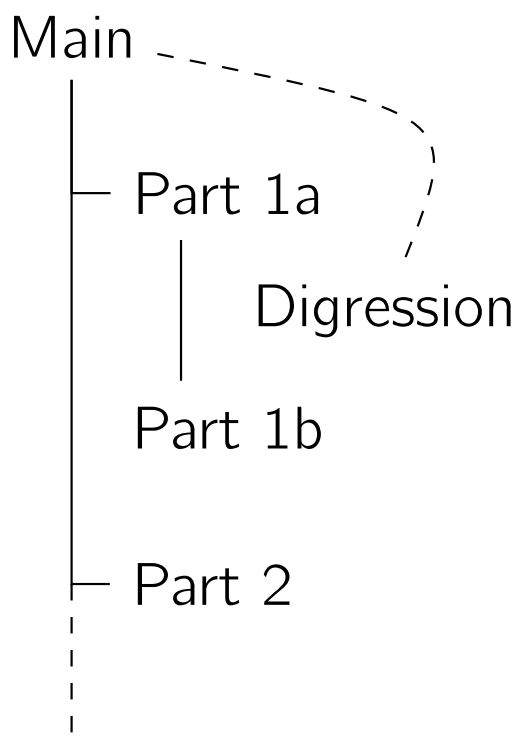
**Default**




**Nested**




**Digression**



# Stata output

```
%STpart data

In a first step we have a look at the data. Here it is:

\begin{stata}
    use ex/Monarchs-of-England, clear
    describe
    notes
\end{stata}

\medskip
Oh, interesting.
```

In a first step we have a look at the data. Here it is:

```
. use ex/Monarchs-of-England, clear
(Kings & Queens of England)
. describe
Contains data from ex/Monarchs-of-England.dta
 Observations:            65                  Kings & Queens of England
    Variables:             4                  9 Sep 2022 00:15
                                              (_dta has notes)
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
Name            str43   %43s
StartYear       int     %8.0g
EndYear         int     %8.0g
Reign           byte    %9.0g
-------------------------------------------------------------------------------
Sorted by:
. notes
_dta:
  1.  source: https://www.101computing.net/kings-queens-of-england/
```

Oh, interesting.

# New part, quiet execution, inline expression

```
%STpart descriptives

\begin{stata*}
    // quietly load data so that part can be run on its own
    use ex/Monarchs-of-England, clear
\end{stata*}

In a second step take a look at some descriptives.

\begin{stata}
    local x "Reign"
    summarize `x'
\end{stata}

\medskip
There are \stres{r(N)} observations, the mean of \stres{"`x'"} is
\stres{%9.1f r(mean)} and the standard deviation is \stres{%9.2f r(sd)}.
```

In a second step take a look at some descriptives.

```
. local x "Reign"
. summarize `x'
```

| Variable | Obs | Mean | Std. dev. | Min | Max |
|---|---|---|---|---|---|
| Reign | 65 | 19.33846 | 16.94317 | 0 | 70 |

There are 65 observations, the mean of Reign is 19.3 and the standard deviation is 16.94.

# Mata output

```
Lets check whether Mata computes the same results as \stcmd{summarize}:

\begin{mata}
    X = st_data(., "`x'")
    rows(X)
    mean(X)
    sqrt(variance(X))
\end{mata}

\medskip
Seems it does.
```

Lets check whether Mata computes the same results as `summarize`:

```
: X = st_data(., "`x'")
: rows(X)
  65
: mean(X)
  19.33846154
: sqrt(variance(X))
  16.94317426
```

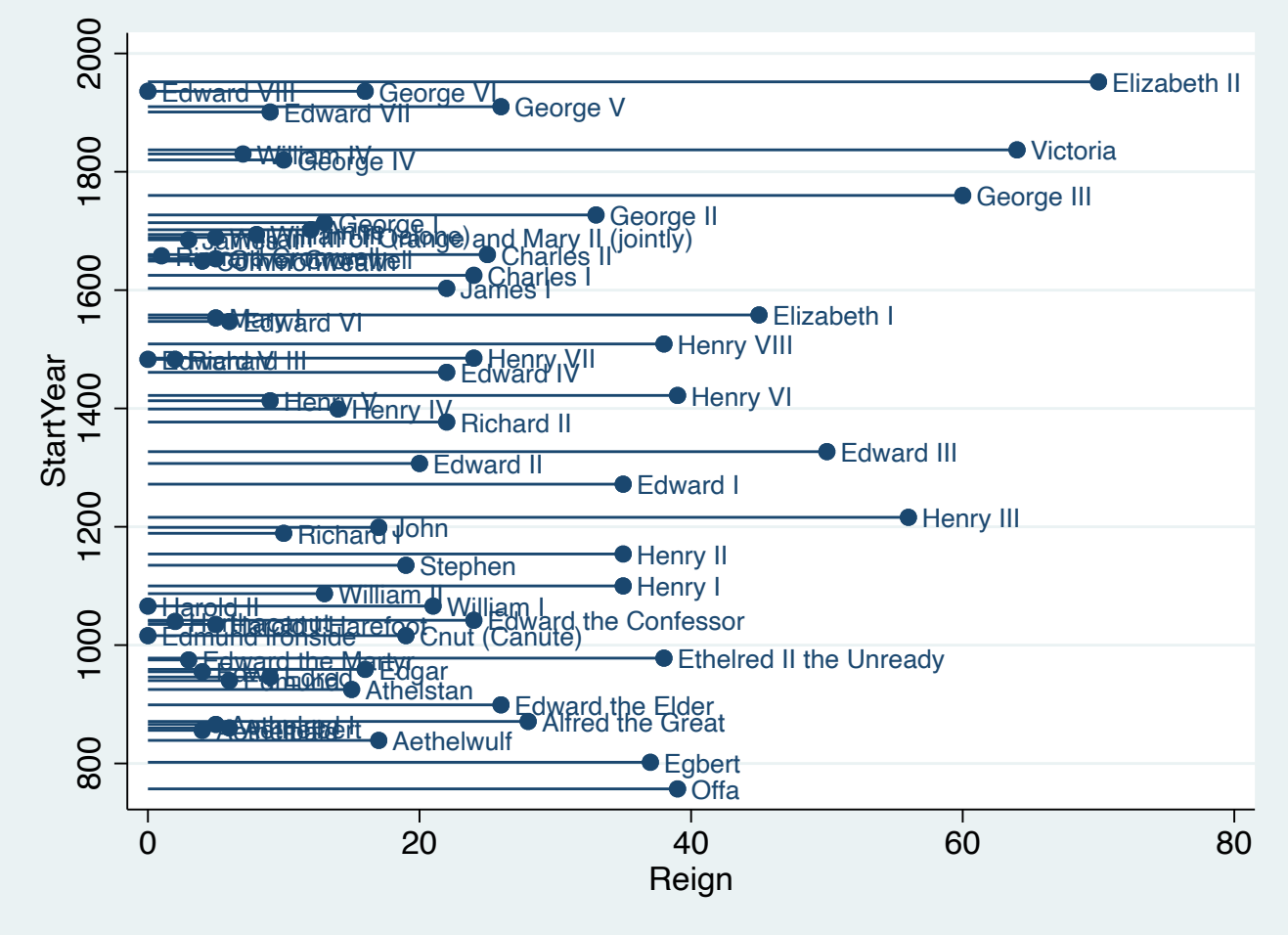
Seems it does.

# Graph

```
%STpart graph

\begin{stata*}
    // quietly load data so that part can be run on its own
    use ex/Monarchs-of-England, clear
\end{stata*}

Here is a scatterplot of \stcmd{Reign} by \stcmd{StartYear}.

\begin{stata}
    twoway dropline Reign StartYear, horizontal mlabel(Name) ylabels(#7)
\end{stata}
\stgraph{}
```

# Here is a scatterplot of `Reign` by `StartYear`.

```
. twoway dropline Reign StartYear, horizontal mlabel(Name) ylabels(#7)
```



# More information

- `sttex` is now available from GitHub, see https://github.com/benjann/sttex/.
- It can be installed by typing

```
. net from https://github.com/benjann/sttex/
. net install sttex
```

- After that, to view the documentation, type

```
. help sttex
```

- `sttex` requires Stata 11 or newer.

# References


- Bruun, N.H. (2016). LOG2MARKUP: Stata module to transform a Stata text log into a markup document. Available from https://ideas.repec.org/c/boc/bocode/s458147.html.
- Haghish, E. F (2016). MarkDoc: Literate Programming in Stata. The Stata Journal 16(4):964–988.
- Haghish, E.F. (2020). Software documentation with markdoc 5.0. Stata Journal 20(2), 336-362.
- Henken, D. (2019). STMD: Stata module to convert dynamic Markdown to HTML format, using Stata dyndoc. Available from https://ideas.repec.org/c/boc/bocode/s458606.html.
- Jann, B. (2016). Creating LaTeX documents from within Stata using texdoc. The Stata Journal 16(2): 245–263.
- Jann, B. (2017). Creating HTML or Markdown documents from within Stata using webdoc. The Stata Journal 17(1):3–38.
- Rodríguez, G. (2017). Literate data analysis with Stata and Markdown. The Stata Journal 17(3):600–618.