

sttex: A new dynamic document command for Stata and L^AT_EX

Ben Jann

University of Bern

2022 Northern European Stata Conference
Oslo, October 12, 2022

Outline

1 Introduction

2 Syntax

3 Example

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.

1 Introduction

2 Syntax

3 Example

Syntax

- The syntax of the `sttex` command is

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

where *sourcefile* is the $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ source file containing pieces of Stata code (default suffix is `.sttex`) and *options* are:

- ▶ options for general behavior and typesetting (requires $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$)
 - ▶ options setting the defaults for Stata sections
 - ▶ gropts(*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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ commands or $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ comments that will be interpreted by `sttex`. I will only present the most important ones below.

Dedicated L^AT_EX commands

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

```
\begin{keyword} [id] [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:

<code>stata</code>	run Stata commands and display the output
<code>stata*</code>	run Stata commands without displaying the output
<code>mata</code>	run Mata commands and display the output
<code>mata*</code>	run Mata commands without displaying the output

options specify the details about how to process the Stata section and format the log. These options take precedence over options specified when calling `sttex`.

Dedicated L^AT_EX commands

- To include a graph created by prior commands, type

```
\stgraph[id]{options}
```

where *id* provides an optional name and where *options* specify the details of how to handle the graph. These options take precedence over options specified when calling `sttex`.

`\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 L^AT_EX comments

- Initialization

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

where *targetfile* specifies the name of the target file and *options* are as above. These options take precedence over options specified when calling `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 L^AT_EX comments

- To create a new part, type

```
%STpart [id [parent]] [, options]
```

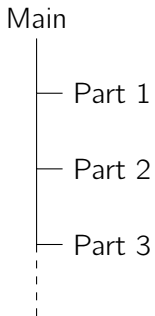
where *id* provides an optional name. Specify *options* to change default settings between parts.

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).

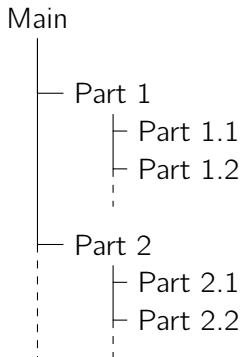
By default, the (unnamed) main part is parent to all other parts. Specify *parent* as . (missing) to create a part that does not depend on the main part.

%STpart must start at the beginning of a line.

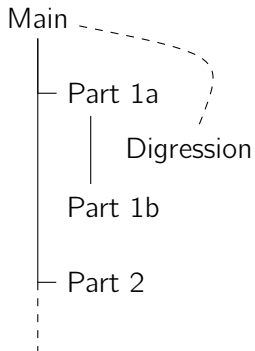
Default



Nested



Digression



1 Introduction

2 Syntax

3 Example

Stata output

%STpart data

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

```
\begin{stata}
  use ex/Norwegian-monarchs, clear
  describe
  notes
\end{stata}
```

```
\medskip
Oh, interesting.
```

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

```
. use ex/Norwegian-monarchs, clear  
(Norwegian monarchs)
```

```
. describe
```

```
Contains data from ex/Norwegian-monarchs.dta
```

```
Observations:      70      Norwegian monarchs  
Variables:         5      11 Oct 2022 18:34  
                    (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
Dynasty	str50	%50s		
Name	str21	%21s		
StartYear	int	%8.0g		
EndYear	int	%8.0g		
ReignDur	byte	%9.0g		

```
Sorted by:
```

```
. notes
```

```
_dta:  
1. source: https://en.wikipedia.org/wiki/List\_of\_Norwegian\_monarchs
```

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/Norwegian-monarchs, clear
\end{stata*}
```

In a second step take a look at some descriptives.

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

\medskip

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

In a second step take a look at some descriptives.

```
. local x "ReignDur"  
. summarize `x`
```

Variable	Obs	Mean	Std. dev.	Min	Max
ReignDur	70	19.11429	14.33405	0	60

There are 70 observations, the mean of ReignDur is 19.1 and the standard deviation is 14.33.

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)  
  70  
: mean(X)  
 19.11428571  
: sqrt(variance(X))  
 14.33405073
```

Seems it does.

Graph

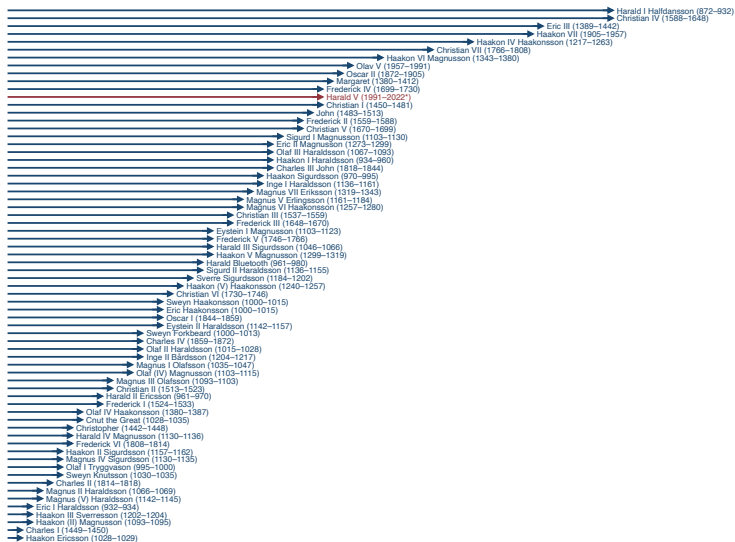
```
%STpart graph
```

```
\begin{stata*}  
    use ex/Norwegian-monarchs, clear  
\end{stata*}
```

Here is the hit list of duration of rule:

```
\begin{stata*}  
    keep if ReignDur>0  
    sort ReignDur  
    generate id = _n  
    generate str Lbl = Name + " (" + string(Start) + "-" + string(End) + ///  
        cond(End==2022,"*", "") + ")"  
    local opts horizontal mlabel(Lbl) mlabsize(tiny) msymbol(A) msangle(-90)  
    twoway (dropline ReignDur id if (Name!="Harald V"), 'opts') ///  
        (dropline ReignDur id if (Name=="Harald V"), 'opts') ///  
        , ylabel(none) yscale(off) xscale(off range(075)) xtitle("") legend(off)  
\end{stata*}  
\stgraph{  
\note{The following code was use to create the graph: \stres{{log}}}
```

Here is the hit list of duration of rule:





The following code was use to create the graph:

```
. keep if ReignDur>0
(2 observations deleted)

. sort ReignDur

. generate id = _n

. generate str Lbl = Name + " (" + string(Start) + "-" + string(End) + ///  
>     cond(End==2022,"*","" ) + ")"

. local opts horizontal mlabel(Lbl) mlabsize(tiny) msymbol(A) msangle(-90)

. twoway (dropline ReignDur id if (Name!="Harald V"), `opts´) ///  
>     (dropline ReignDur id if (Name=="Harald V"), `opts´) ///  
>     , ylabel(none) yscale(off) xscale(off range(075)) xtitle("") legend(off)
```

More information

- There are some further dynamic tags for specific purposes, as well as a wide range of options providing additional functionality. See the help file.
- `sttex` is available from SSC. It can be installed by typing

```
. ssc install sttex, replace
```
- To view the documentation, type

```
. help sttex
```
- `sttex` requires Stata 11 or newer.
- `sttex` is also on GitHub, see <https://github.com/benjann/sttex/>.

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.