

Dynamic documents in Stata: MarkDoc, Ketchup, and Weaver

Summary

For Stata users who do not know LaTeX, writing a document that includes text, graphs, and Stata syntax and output has been a tedious and unreproducible manual process. To ease the process of creating dynamic documents in Stata, many Stata users have wished to see two additional features in Stata: literate programming and combining graphs with logfiles in a single document. MarkDoc, Ketchup, and Weaver are three user-written Stata packages that allow you to create a dynamic document that includes graphs, text, and Stata codes and outputs and export it in a variety of file formats, including PDF, Docx, HTML, LaTeX, OpenOffice/LibreOffice, EPUB, etc. I will also discuss further details about the specialties of these packages and their potential applications.

Limitations of Stata in producing dynamic documents

"The Stata dofile and logfile provide good-enough tools for reproducing the analysis. However, Stata logfile only includes text output and graphs are exported separately. Consequently, combining a graph to Stata output in a report has been a manual work. In addition, the possibility of adding text to explain the analysis results in Stata output is limited by the duplication problem. For example, a text paragraph that is printed using the display command gets duplicated in the logfile by appearing on the command and the command's output. Therefore, writing an analysis document that requires graphs and outputs, as well as text for describing the analysis results has become a manual work which is unreproducible, laborious, time consuming, prone to human error, and boring." [2]

```
. display "{p}Displaying a text paragraph..."  
Displaying a text paragraph...
```

Suggesting criteria for literate programming packages in Stata

The following points may be considered for developing Literate Statistical Practice [3] packages in Stata. To illustrate these points, imagine a large dofile.

- Human-language text as well as computer language (i.e. Stata) should be written in Do-file editor and executed from Stata to **keep the work-flow natural**. The use of external text processors should be eliminated, making the work flow natural for all people who are to some extent familiar with Stata.
- Clicking the Execute (Do) button in Stata dofile editor should be the only required click for generating the documents
- Human language and computer language should be **easily distinguishable** by human eyes.
- Avoid complex markup languages for writing text and integrating graphs in the document. Complex markup languages negatively influence the do-files aesthetics and reduce the readability of the syntax file.
- Simple is more! Keep the script file as clean as possible.

Importance of readability of the script file

Literate Programming [4] and Literate Statistical Practice [2] can facilitate learning, sharing, and making sense of analysis syntax [5] but it comes at a cost of mixing two different languages in a single text file. This by itself is only a "potential" problem i.e. if it is not done properly, it reduces the readability of the script file. In addition, using complex markup syntax for writing and styling text further decreases the readability of the script file because it makes human language to appear more similar to the programming language and less distinguishable.

Caring about the readability of the script file is the key feature of a successful literate program because it makes the practice of literate statistical programming fun rather than weird style of programming. Literate programming should make the codes easier to understand or follow.

What is proper practice in literate statistical programming?

It is very difficult to define what is proper practice of literate statistical practice. Some people will be very distracted if the script file is not clean and some other people may only care about the output document. The rule of thumb for judging proper practice of literate programming, in my opinion, is forgetting about the compiled outcome and focusing on the script file instead. Only because you have produced a clean PDF document that includes text, Stata (or any other statistical package) outputs, and graphs, it does not mean that your **script file is readable by other fellows or enthusiastic learners**.

Always consider that you will SHARE your script file rather than the weaved document. Do not write the script file for yourself (with your own rules) and instead, try to make it readable by everyone. If you think **your script file is clean and well-written**, then you probably have a reasonable style for weaving the document. In brief, write beautiful script files, programming is an art. [5]

Structure of the dynamic document

Given that the dynamic report includes computer language (i.e. Stata commands / programs) which generate results as well as human language which explains and discusses the results, it is clear that the structure of the dofile (or any script file) that create the dynamic document should be a sequence of human language and computer language.

The example dofile named **chunks.do** demonstrates this structure. The dofile is created by sequences of human text - written as comments - and chunks of stata codes which generate descriptive statistics and a regression model.

Markup languages for Weaving in Stata

The following markup languages are available for MarkDoc, Weaver, and Ketchup. This workshop only focuses on Markdown but gives quick tips how to get started with other markup languages as well. **Markdown, HTML, and LaTeX most be written as comments in the dofile.** This allows the human language to appear in green color which is easily distinguishable from the Stata commands, improving the readability of the dofile.

- **Markdown:** MarkDoc, Weaver, and Ketchup
- **HTML:** MarkDoc, Weaver, and Ketchup
- **Additional Markup Codes:** Weaver and Ketchup
- **LaTeX:** MarkDoc

Writing with Markdown in Stata

Markdown is an easy to read and easy-to-wrote text to HTML converter [7]. Markdown uses simple symbols to simplify styling text and as a result, has gained a remarkable popularity for writing in blogs, websites, and online forums. Markdown is an ideal markup language because not only it does not reduce the readability of the script file, but also it improves the readability by making headings and subheadings more distinguished from the text paragraphs. For a detailed explanation on using Markdown in Stata as well as Markdown syntax, see [Writing with Markdown in Stata](#).

Markdown syntax example

```
/*
This is heading 1
=====

This is heading 2
-----

This is a text paragraph which requires no syntax.

*/
/*
#This is heading 1
##This is heading 2
###This is heading 3
####This is heading 4
#####This is heading 5
#####This is heading 6
- hyphen creates a list
*italic* and italic make the text italic
**bold** and bold make the text bold
***italic & bold*** and italic & bold make the text italic & bold
[MyLink](http://haghish.com/stata "Optional Title") create a link named MyLink
![description](/path/to/graph.jpg "Optional Title") inserts graph.jpg to the
document
*/
```

Example for writing with HTML

```
/*
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<p>This is a text paragraph which begins with p tag</p>.
```

```
*/
```

Example for writing with LaTeX

```
/*
```

```
\documentclass{article}
\begin{document}
\section{This is heading 1}
\subsection{This is heading 2}
```

This is a text paragraph which requires no syntax.

```
\begin{verbatim}
```

```
*/
```

```
sysuse auto
regress price mpg
```

```
/*
```

```
\end{verbatim}
\end{document}
```

```
*/
```

Comparing Markdown, HTML, and LaTeX

The more complicated the markup language, the more flexible the text styling is. There is no doubt that HTML and LaTeX provide more flexibility for weaving the document. But the big advantage of Markdown is its simplicity and improved readability. Therefore, Markdown is recommended in general and the other markup languages should be used by advanced users who wish to produce a dynamic document with high details.

Overview of the dynamic document packages in Stata

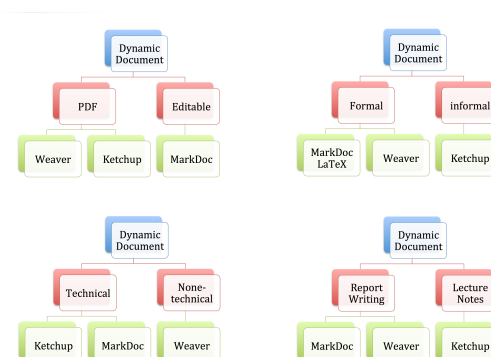
MarkDoc is a smcl logfile translator. You "prepare" a logfile and then translate it into an editable dynamic document. MarkDoc can translate smcl to Markdown, Docm, Docx, Doc, OpenOffice/LibreOffice ODT, LaTeX, ConTeXt, HTML, XHTML, OpenDocument XML, EPUB, ReStructuredText, AsciiDoc, Textile, and OPML.

Weaver creates an HTML logfile and stores only the information that you want to include in the logfile. It provides real-time preview of the dynamic document as you weave it. Weaver creates HTML and PDF documents only.

Ketchup is a mix of Weaver and MarkDoc. It creates HTML and PDF, but it's only a smcl translator. Ketchup provides syntax highlighting in the dynamic document identical to Stata syntax highlighter.

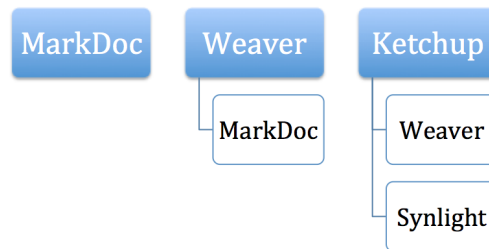
Comparing the packages

The figure below compares Weaver, MarkDoc, and Ketchup packages based on different criteria which are writing formal vs informal, technical vs non-technical, PDF vs editable, and writing lecture notes vs analysis report. **I add that working with Weaver requires learning more commands compared to Ketchup and MarkDoc**, with MarkDoc being the simplest. MarkDoc is the simplest both in terms of the learning curve as well as the appearance of the document.



Installing the packages

MarkDoc is an independent package. However, Weaver calls MarkDoc through **mp** and **md** command to convert Markdown syntax to HTML and appends it to the weaver document. Ketchup requires Weaver for making use of **img** and **knit** commands. In addition, Ketchup calls Synlight for highlighting syntax.



```
ssc install markdoc
```

```
ssc install ketchup
```

```
ssc install weaver
```

```
ssc install synlight
```

MarkDoc Package

MarkDoc is a Stata package that supports Literate Statistical Practice and compiles editable document formats such as Markdown, Docm, Docx, Doc, OpenOffice/LibreOffice ODT, LaTeX, ConTeXt, HTML, XHTML, OpenDocument XML, EPUB, ReStructuredText, AsciiDoc, Textile, and OPML. **MarkDoc** Supports 3 different languages for Weaving the dynamic document which are **Markdown**, **HTML**, and **LaTeX**. Although Markdown is recommended, HTML or LaTeX are much more flexible in terms of styling a sophisticated document and can be used by advanced users. MarkDoc is primarily a SMCL processor and converter and thus all it requires is a smcl logfile to process. MarkDoc includes one command and a few options only, making it very simple to practice. The `export(name)` option specifies the file format. Give the "file extension" to this option. If you use HTML or LaTeX to weave, the document should be exported in **html** or **tex** format.

```
markdoc smclfile [, replace erase export(name)]
```

Distinguished features of MarkDoc

- human language is written as a comment in the do-file and thus, it appears in green color by default
- MarkDoc supports Markdown, HTML, and LaTeX for writing text and inserting graphs or other materials
- MarkDoc is simple
- It creates a dynamic document in all popular editable formats

Try the example do-files

To save time, I have not printed any example do-files on the slides. However there are 3 different dofiles available for download at the bottom of this document which demonstrate how to weave a document in MarkDoc using Markdown, HTML, and LaTeX. [Click here to jump to the download links](#). The next 3 slides provide

Weaver

Weaver excels in weaving HTML and PDF dynamic documents but it has been written with a different philosophy. Weaver creates a HTML-based logfile and allows the users to decide what text, graphs, commands, and outputs to appear in the document. As a result, an exploratory analysis as well as the hypothesis testing can be written on a single file and the user decides what content should be printed on the HTML logfile. This degree of flexibility cannot be achieved by the current version of MarkDoc (1.6) and Ketchup (1.4). Weaver provides a handful of commands and learning it requires more effort compared to MarkDoc.

Weaver's engine

In contrast to MarkDoc, Weaver is not a translator. It creates an HTML logfile and saves all text, commands, and outputs into HTML rather than smcl. The HTML logfile provides a remarkable flexibility for writing a complex document. For example, scripts of any web-language can be inserted to Weaver's logfile including JavaScripts, CSS, etc. Perhaps the most notable feature of Weaver is that it allows live-view of the dynamic document in both HTML and PDF. To do so open the HTML logfile and refresh it in the web browser any time you weave something new or type **report** in Stata console to open the PDF preview.

Key features of Weaver

- HTML and PDF dynamic documents
- Weaver is picky. It provides much freedom in choosing what content to appear in the document
- HTML and PDF live-preview at any point
- Automate table of content plus links to the headings and subheadings
- Portrait and landscape modes
- Styling Stata codes and outputs

- Support for writing dynamic text (i.e. text that interprets Macro contents)
- Options for creative text writing through [Additional Markup Codes](#)
- Supporting web languages
- Supports writing with HTML, Markdown, and Additional Markup Codes
- Presentation mode for comparing several graphs in one page
- Allows appending smcl logfiles and HTML files to the document

Weaver's syntax

<code>weave using filename [, options]</code>	<code>/* begin weaving */</code>
<code>div command</code>	<code>/* include the command and outputs in the report */</code>
<code>codes command</code>	<code>/* only include the command in the report */</code>
<code>results command</code>	<code>/* only include the output in the report */</code>
<code>mp</code>	<code>/* begin registering comments */</code>
<code>md</code>	<code>/* end registering comments */</code>
<code>img filepath [, options]</code>	<code>/* insert a graph or image */</code>
<code>knit "string"</code>	<code>/* write dynamic text */</code>
<code>html "string"</code>	<code>/* write html or web scripts in the document */</code>
<code>report [, options]</code>	<code>/* create & open PDF preview */</code>
<code>linebreak</code>	<code>/* adds an empty line */</code>
<code>pagebreak</code>	<code>/* breaks the page and begins a new page */</code>
<code>weavend</code>	<code>/* ending the document */</code>

Weaver's applications

What distinguishes Weaver is that it allows you to include all of your codes in a single file, run all of the analysis, and Weave a document by selecting the content that you would like to include in the final document. When the dynamic document is meant to be shared with non-statisticians, Weaver becomes useful because it allows to keep the document simple by suppressing technical or unessential results. Particularly, use Weaver for weaving a document from a **large analysis** or writing a **formal analysis report**. Weaver has several predesigned document styles that can be used to generate a formal document without any need for further adjustment. Compared to MarkDoc and Ketchup, Weaver has a very simple engine which means it is less likely to have bugs.

Weaver example do-files

At the bottom of this document, there is an example do-file available for download. The example dofiles provides an example for each of the commands as well as the Additional Markup Codes.

Ketchup package

Ketchup is the offspring of MarkDoc and Weaver and therefore, has inherited several features of its parents. The engine of Ketchup package was written with the same logic as MarkDoc, i.e. processing and manipulating Stata SMCL logfile. However, the engine of Ketchup is HTML-based whereas the primary output of MarkDoc is a Markdown **md** file. Therefore both Ketchup and MarkDoc packages are simply smcl processors. Similar to MarkDoc, Ketchup supports writing text and importing graphs using **Markdown** and **HTML** formats and allow for combining these two markup languages as well. MarkDoc also inheres the simplicity of MarkDoc and includes one main command and two optional styling commands.

Ketchup also inherited many features of Weaver package. Similar to Weaver, Ketchup produces HTML and PDF documents, provides presentation mode for graphs, and many themes options for styling the document. Ketchup also borrows 2 optional commands from Weaver which are **img** used for inserting, resizing, and styling graphs and **knit** used for writing dynamic text. Ketchup also takes the **Additional Markup Codes** from Weaver and can highlight and change color of text and headings.

Distinguished features of Ketchup

What makes Ketchup distinguished from Weaver is its simplicity. Whereas Weaver requires learning several commands,

Ketchup is only one command that produces HTML and PDF documents from smcl logfile. In addition, Ketchup has a unique feature that both MarkDoc and Weaver lack which is Syntax highlighter. Borrowing functions from [Synlight package \[8\]](#), Ketchup highlights the syntax of Stata [Commands](#), [Functions](#), [Strings](#), [Local](#) and [Global Macros](#), [Numbers](#), [Comments](#), and [Braces](#) identical to do-file editor. The syntax highlighter can be turned off using **nolight** option which also speeds up the process of creating the dynamic documents.

Ketchup Syntax

```
ketchup smclfile [, options] /* the main command */  
img filepath [, options] /* Insert a graph or image */  
knit "string" /* write dynamic text */
```

Working with the `knit` and `img` commands is identical to the Weaver package. Weaver was updated to support Ketchup, which in turn, avoids defining new commands with the same functions.

Ketchup applications

Ketchup is most suitable for creating reasonably short dynamic reports. It is particularly beneficial for Stata learners and Stata educators who carry out workshops or teach statistics using Stata. Simple syntax and appealing themes make Ketchup a reasonably simple choice for taking notes inside do-file editor and follow a lecture.

Another reason for using ketchup for education purpose (or preparing Stata outputs for students) is that Ketchup preserves the macros in the commands and only interpret them in the output. In addition, the Weaver package can only print **one Stata command** at a time, using the `div` command. Therefore, Weaver is not able to print a code which is more than a single command (e.g. a loop) whereas Ketchup can process complex programs and loops and style them accordingly. Ketchup is an educative package and is not suitable for producing a formal document because all commands that appear in the logfile will be printed in the output document and such a feature is not ideal for writing formal analysis reports that are meant to be shared with colleagues that have limited understanding of statistics and no skill in Stata syntax.

Suggestions for MarkDoc, Weaver, and Ketchup

- Use MarkDoc for exporting editable formats (docx, doc, odt, LaTeX).
- Use MarkDoc for writing complex LaTeX document inside do-file edit and export them to LaTeX.
- Use MarkDoc for writing your next Stata Journal article! Even if you are not familiar with LaTeX, write it with Markdown and export it as **tex**.
- Use Weaver for writing long, formal, and less technical dynamic documents which are not written for nerds.
- Use Ketchup for teaching Stata, preparing workshop and classroom handout.
- Use Ketchup for learning Stata and writing lecture notes.
- Use Ketchup for creating technical dynamic documents.

Third-party software

MarkDoc requires installation of [Pandoc](#) on your machine. In addition to Pandoc, Weaver and Ketchup also require a PDF printer driver. Two drivers which are [Princexml](#) and [wkhtmltopdf](#) have been styled to work properly with Weaver and Ketchup. Princexml is the default printer.

Software installation

The required third-party software will be automatically downloaded on the user's machine, once you begin working with these packages. The software are stored in `~/ado/plus/Weaver` directory. You can see the full path of this directory on your machine by typing `cd "c:\sysdir_plus"`

Troubleshoot

The default software downloaded with Weaver, Ketchup, and markDoc have been tested in Microsoft Windows **XP**, Windows **7**, and Windows **8.1**, Macintosh **OSX 10.9.5**, Linux Mint **17** Cinnamon (32bit & 64bit), Ubuntu **14** (32bit & 64bit), and CentOS **7** (64bit). Linux users who are using a different linux distribution may need to manually install the required software and provide the file paths using `pandoc(str)` (for Pandoc) and `setpath(str)` for PDF printer's path.

Weaver interacts with the operating system for installing, creating, replacing, and removing files. Make sure that you run Stata as administrator. In addition, check permissions of the working directory. Stata should be allowed to remove or replace files in the working directory.

Conlusions

Using dynamic document packages are an ideal way for documenting the analysis as well as generating the analysis output to includes the results, graphs, and text.

MarkDoc, Weaver, and Ketchup are three different packages that produce dynamic documents in HTML, PDF, and many editable formats such as Microsoft Words, LaTeX, Open Office, etc...

All these packages support writing with Markdown and HTML. MarkDoc also supports writing with LaTeX.

The importance of a clean and readable script file should not be neglected by over emphasizing dynamic documents. Creating a beautiful dynamic document is not the only purpose of literate programming. The main purpose is to create a script file (dofile) that is readable by everyone. The script file should be very clean, written with a simple markup language

Example Do-files

 [Example Do-file for *sequence of chunks*, using *Ketchup* package.](#)

 [Example Do-file for *Markdown Syntax*](#)

 [Example Do-file for Weaving with Markdown in *MarkDoc*](#)

 [Example Do-file for Weaving with HTML in *MarkDoc*](#)

 [Example Do-file for Weaving with LaTeX in *MarkDoc*](#)

 [Example Do-file for *Weaver*](#)

 [Example Do-file for *Ketchup*](#)

References

[1] Baumer, B., M. Cetinkaya-Rundel, A. Bray, L. Loi, and N. J. Horton. 2014. R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics. *Technology Innovations in Statistics Education* 8(1).

[2] E. F. Haghish (Submitted). Editable dynamic documents with MarkDoc package.

[3] Rossini, A. J. 2001. Literate Statistical Practice. In *Proceedings of the 2nd International Workshop on Distributed Statistical Computing (DSC 2001)*.

[4] Knuth, D. E. (1992). *Literate programming*. CSLI Lecture Notes, Stanford, CA: Center for the Study of Language and Information (CSLI), 1992, 1.

[5] Baumer, B., M. Cetinkaya-Rundel, A. Bray, L. Loi, and N. J. Horton. 2014. R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics. *Technology Innovations in Statistics Education* 8(1).

[6] Knuth, D. E. (2007, January). *Computer programming as an art*. In *ACM Turing award lectures* (p. 1974). ACM.

[7] Gruber, J. 2004. *Markdown: Syntax*. URL <http://daringfireball.net/projects/markdown/syntax>.

[8] E. F. Haghish (2014). SYNLIGHT: Stata module to highlight syntax in SMCL and translate to HTML format. <https://ideas.repec.org/c/boc/bocode/s457894.html>.