

# Package: Statamarkdown (via r-universe)

October 29, 2024

**Version** 0.9.2

**Date** 2023-12-03

**Title** 'Stata' Markdown

**Description** Settings and functions to extend the 'knitr' 'Stata' engine.

**Imports** knitr (>= 1.43), xfun (>= 0.39)

**Suggests** rmarkdown

**SystemRequirements** Stata

**Maintainer** Doug Hemken <dehemken@wisc.edu>

**License** MIT + file LICENSE

**BugReports** <https://github.com/Hemken/Statamarkdown/issues>

**Encoding** UTF-8

**Repository** <https://remlapmot.r-universe.dev>

**RemoteUrl** <https://github.com/Hemken/Statamarkdown>

**RemoteRef** HEAD

**RemoteSha** f72729d784891fd10cc0b67119346ec982bf477e

## Contents

Statamarkdown-package . . . . .	2
find_stata . . . . .	3
spinstata . . . . .	4
stata_engine . . . . .	6
<b>Index</b>	<b>8</b>

---

Statamarkdown-package *Settings and functions to extend the knitr Stata engine.*

---

## Description

To use these functions and settings, attach the **Statamarkdown** library from *within* the document to be knitr. A typical preliminary code check in a document would be

```
```{r setup, include=FALSE}
library(Statamarkdown)
```
```

Using the "Stata" language engine in **knitr** has a number of limitations. Each Stata code chunk is run as a separate batch file, and source code is part of the output returned to the document being knitr. This package provides a language engine with code chunk options to overcome these limitations.

If you render multiple documents from the same script or R session, you should detach ("Statamarkdown") in between documents.

## Code Block (Chunk) Options

### Statamarkdown Chunk Options:

*collectcode*: (logical): A function here sets up a chunk hook, that silently repeats selected code chunks at the beginning of later code chunks. This allows the code in one chunk to use the results of a previous chunk. The user marks code chunks to be silently repeated with the chunk option `collectcode=TRUE`.

*cleanlog*: (logical): A second function here sets up an output hook. This removes Stata code from the output by default. To leave Stata commands in the output, specify the chunk option `cleanlog=FALSE`.

*savedo*: (logical): To save the code from a code block (as a "do" file) and also to save the Stata log file produced by that code block, specify chunk option `savedo=TRUE`. The filenames are the same as the chunk label.

### Knitr Chunk Options:

*eval*: (logical): Whether or not to evaluate the code in the code block. Use `eval=FALSE` to show code to the reader without having it evaluated. Selective evaluation by specifying a numeric vector (an option for R code blocks) does *not* work for Stata.

*include*: (logical): Whether or not any trace of this code block appears in your document. Use `include=FALSE` to evaluate code but suppress the source code echo and all output (including error messages). This is equivalent to `eval=TRUE, echo=FALSE, results="hide", error=FALSE`.

*echo*: (logical, numeric vector): Whether or not to show the reader the source code. Use `echo=FALSE` to suppress the source code in your document. If this is specified as a numeric vector, it indicates which source lines to show or suppress. For example, `echo=c(1, 2)` shows only the first two lines of the code block in the document (while

still evaluating the entire code block). Likewise, `echo=-1` hides just the first line of code from the reader.

*results: (character):* To suppress normal output while still showing error messages use `results="hide"`.

*error: (logical):* Whether or not to show error messages in your document. To suppress error messages use `error=FALSE`.

Error messages that Stata writes to the log will appear as normal output - they are not "errors" in this context. This option affects error messages returned to/by the operating system.

*comment: (character):* A prefix to use before lines of output. The default for R output is `comment="###"`

*child: (character):* Filename to be run and input in the document.

### Author(s)

Doug Hemken

### References

More documentation and examples: <https://www.ssc.wisc.edu/~hemken/Stataworkshops/stata.html#stata-and-r-markdown>

### See Also

The package that this extends: [knitr-package](#).

---

|            |                                                                                                                                                |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| find_stata | <i>A helper function that seeks to locate your Stata executable. Ordinarily this is run automatically when <b>Statamarkdown</b> is loaded.</i> |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------|

---

### Description

This function searches for recent versions of Stata ( $\geq$  Stata 11), in some of the usual default installation locations.

If Stata is not found, you will have to specify its correct location yourself.

### Usage

```
find_stata(message=TRUE)
```

### Arguments

`message` (logical) Whether or not to print a message when Stata is found.

### Value

A character string with the path and name of the Stata executable.

**Author(s)**

Doug Hemken

**See Also**

[Statamarkdown-package](#)

**Examples**

```
indoc <- '
# An R console example
## In a first code chunk, set up with
```{r}
library(Statamarkdown)
```

## Then mark Stata code chunks with
```{stata}
sysuse auto, clear
generate gpm = 1/mpg
summarize price gpm
```
'

if (!is.null(Statamarkdown::find_stata())) {
  # To run this example, remove tempdir().
  fmd <- file.path(tempdir(), "test.md")
  fhtml <- file.path(tempdir(), "test.html")

  knitr::knit(text=indoc, output=fmd)
  rmarkdown::render(fmd, "html_document", fhtml)
}
```

---

spinstata

*Convert a specially marked up Stata "do" file to Markdown and HTML.*

---

**Description**

This function takes a Stata file containing special mark up in it's comments, and converts it to Markdown and HTML documents (or one of several other formats).

**Usage**

```
spinstata(statafile, text=NULL, keep=FALSE, ...)
```

**Arguments**

|           |                                                                                            |
|-----------|--------------------------------------------------------------------------------------------|
| statafile | A character string with the name of a Stata "do" file, containing markup in it's comments. |
| text      | A character string in place of a file.                                                     |
| keep      | Whether to save intermediate files.                                                        |
| ...       | options passed to <code>knitr::spin</code>                                                 |

**Details**

This function takes a Stata file containing special mark up in it's comments, and converts it into knitr's "spin" format. This is in turn sent to `knitr::spin`, and converted to Markdown and HTML (or one of several other formats).

Special Markup:

- `/*' "` - Begin document text, ends with `' */`
- `/*+ "` - Begin chunk header, ends with `"+*/`
- `/*R "` - Begin a chunk of R code, ends with `"R*/`
- `*/* "` - Dropped from document, ends with `*/*`

**Value**

The path to the output file.

If given text instead of a file, returns the compiled document as a character string.

**Author(s)**

Doug Hemken

**See Also**

[Statamarkdown-package](#)

**Examples**

```
## Not run:
indoc <- "/*'
# Statamarkdown Example
```

This is a special Stata script which can be used to generate a report.  
You can write normal text in command-style comments.

First we load Statamarkdown.

```
'*/
/*+  setup +*/
/*R
library(Statamarkdown)
R*/
```

```

/*' The report begins here. '*/

/*+ example1, engine='stata' +*/
sysuse auto
/* Stata comment */
summarize

/*' You can use the ***usual*** Markdown to mark up text.'*/
"
if (!is.null(Statamarkdown::find_stata())) {
  # To run this example, remove tempdir().
  fhtml <- file.path(tempdir(), "test.html")
  x<-Statamarkdown::spinstata(text=indoc)
  writeLines(x, fhtml)
}

## End(Not run)

```

---

stata\_engine

*Define a Stata engine for knitr*


---

## Description

This function creates a modified Stata engine.

Set up once per session (i.e. document). Ordinarily this is run automatically when **Statamarkdown** is loaded.

## Usage

```
stata_engine(options)
```

## Arguments

options            options are passed to the engine function when it is actually invoked within **knitr**.

## Details

This function is used as follows.

- `stata_engine(options)` is a language engine that returns Stata log output. The end user should not need to use the language engine function directly. This is the workhorse function that actually calls Stata and returns output.

## Value

The language engine function returns Stata code and output internally to **knitr**.

**Author(s)**

Doug Hemken

**See Also**

[knit\\_engines](#)

**Examples**

```
indoc <- '  
# An R console example  
## In a first code chunk, set up with  
```{r}  
library(Statamarkdown)  
```  
  
## Then mark Stata code chunks with  
```{stata}  
sysuse auto, clear  
generate gpm = 1/mpg  
summarize price gpm  
```  
,  
  
if (!is.null(Statamarkdown::find_stata())) {  
  # To run this example, remove tempdir().  
  fmd <- file.path(tempdir(), "test.md")  
  fhtml <- file.path(tempdir(), "test.html")  
  
  knitr::knit(text=indoc, output=fmd)  
  rmarkdown::render(fmd, "html_document", fhtml)  
}
```

# Index

[find\\_stata](#), [3](#)

[hook\\_orig](#) (Statamarkdown-package), [2](#)

[knit\\_engines](#), [7](#)

[knitr-package](#), [3](#)

[spinstata](#), [4](#)

[stata\\_engine](#), [6](#)

[Statamarkdown](#) (Statamarkdown-package), [2](#)

[Statamarkdown-package](#), [2](#)