

R Markdown :: CHEAT SHEET



What is R Markdown?



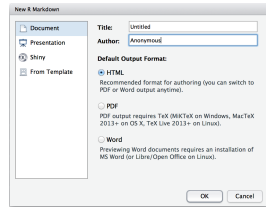
.Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

Reproducible Research - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

The screenshots illustrate the RStudio workflow for R Markdown. The main editor shows a .Rmd file with YAML metadata (title, author, output), R code chunks (setup, cars), and text. Annotations point to various actions: 'set preview location', 'insert code chunk', 'go to code chunk', 'run code chunk(s)', 'publish', 'show outline', 'run all previous chunks', 'modify chunk options', and 'run current chunk'. The preview window shows the rendered HTML output, including a table of car statistics. The console shows the execution of `library(rmarkdown)` and `render("report.Rmd", output_file = "report.html")`.

Workflow



- 1 **Open a new .Rmd file** at File ► New File ► R Markdown. Use the wizard that opens to pre-populate the file with a template
- 2 **Write document** by editing template
- 3 **Knit document to create report**; use knit button or `render()` to knit
- 4 **Preview Output** in IDE window
- 5 **Publish** (optional) to web server
- 6 **Examine build log** in R Markdown console
- 7 **Use output file** that is saved along side .Rmd

render

Use `rmarkdown::render()` to render/knit at cmd line. Important args:

input - file to render	output_options - List of render options (as in YAML)	output_file	output_dir	params - list of params to use	envir - environment to evaluate code chunks in	encoding - of input file
-------------------------------	---	--------------------	-------------------	---------------------------------------	---	---------------------------------

Embed code with knitr syntax

INLINE CODE
Insert with ``r <code>``. Results appear as text without code.
Built with ``r getRversion()`` → Built with 3.2.3

CODE CHUNKS
One or more lines surrounded with ````${code}````. Place chunk options within curly braces, after `r`. Insert with ````${code}````

GLOBAL OPTIONS
Set with `knitr::opts_chunk$set()`, e.g.
````${code}````  
`knitr::opts_chunk$set(echo = TRUE)`

### IMPORTANT CHUNK OPTIONS

**cache** - cache results for future knits (default = FALSE)  
**cache.path** - directory to save cached results in (default = "cache/")  
**child** - file(s) to knit and then include (default = NULL)  
**collapse** - collapse all output into single block (default = FALSE)  
**comment** - prefix for each line of results (default = '###')

**dependson** - chunk dependencies for caching (default = NULL)  
**echo** - Display code in output document (default = TRUE)  
**engine** - code language used in chunk (default = 'R')  
**error** - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)  
**eval** - Run code in chunk (default = TRUE)

**fig.align** - 'left', 'right', or 'center' (default = 'default')  
**fig.cap** - figure caption as character string (default = NULL)  
**fig.height, fig.width** - Dimensions of plots in inches  
**highlight** - highlight source code (default = TRUE)  
**include** - Include chunk in doc after running (default = TRUE)

**message** - display code messages in document (default = TRUE)  
**results** (default = 'markup')  
**'asis'** - passthrough results  
**'hide'** - do not display results  
**'hold'** - put all results below all code  
**tidy** - tidy code for display (default = FALSE)  
**warning** - display code warnings in document (default = TRUE)

Options not listed above: `R.options`, `aniopts`, `autodep`, `background`, `cache.comments`, `cache.lazy`, `cache.rebuild`, `cache.vars`, `dev`, `dev.args`, `dpi`, `engine.opts`, `engine.path`, `fig.asp`, `fig.env`, `fig.ext`, `fig.keep`, `fig.lp`, `fig.path`, `fig.pos`, `fig.process`, `fig.retina`, `fig.scap`, `fig.show`, `fig.showtext`, `fig.subcap`, `interval`, `out.extra`, `out.height`, `out.width`, `prompt`, `purl`, `ref.label`, `render`, `size`, `split`, `tidy.opts`

## .rmd Structure

**YAML Header**  
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).  
At start of file  
Between lines of ---  
**Text**  
Narration formatted with markdown, mixed with:  
**Code Chunks**  
Chunks of embedded code. Each chunk:  
Begins with ````${code}````  
ends with ``````  
R Markdown will run the code and append the results to the doc.  
It will use the location of the .Rmd file as the **working directory**

## Parameters

Parameterize your documents to reuse with different inputs (e.g., data, values, etc.)

1. **Add parameters** - Create and set parameters in the header as sub-values of params

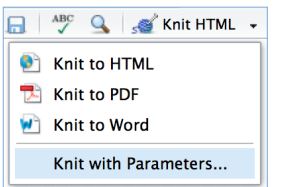
```
params:
 n: 100
 d: !r Sys.Date()
```

2. **Call parameters** - Call parameter values in code as `params$<name>`

```
Today's date
is !r params$d`
```

3. **Set parameters** - Set values with Knit with parameters or the params argument of render():

```
render("doc.Rmd", params = list(n = 1,
d = as.Date("2015-01-01")))
```



## Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

1. Add runtime: shiny to the YAML header.
2. Call Shiny input functions to embed input objects.
3. Call Shiny render functions to embed reactive output.
4. Render with `rmarkdown::run` or click Run Document in RStudio IDE

The screenshot shows a .Rmd file with the following code: `--- output: html_document runtime: shiny ---` followed by a Shiny input function `numericInput("n", "How many cars?", 5)` and a `renderTable` function that uses `cars` data and the input `n`. The rendered output shows a table with 5 rows of car data, where the first row is highlighted in blue.

Embed a complete app into your document with `shiny::shinyAppDir()`

NOTE: Your report will be rendered as a Shiny app, which means you must choose an html output format, like `html_document`, and serve it with an active R Session.





# Pandoc's Markdown

Write with syntax on the left to create effect on right (after render)

Plain text  
End a line with two spaces to start a new paragraph.  
**italics** and **bold**  
`verbatim code`  
sub/superscript<sup>2</sup>~  
~strikethrough~  
escaped: \\* \\_ \\  
endash: --, emdash: ---  
equation:  $A = \pi * r^2$   
equation block:

Plain text  
End a line with two spaces to start a new paragraph.  
*italics* and **bold**  
`verbatim code`  
sub/superscript<sup>2</sup>  
~strikethrough~  
escaped: \* \_ \  
endash: --, emdash: ---  
equation:  $A = \pi * r^2$   
equation block:

$E = mc^2$

> block quote

# Header1 {#anchor}

## Header 2 {#css\_id}

### Header 3 {#css\_class}

#### Header 4

##### Header 5

##### Header 6

<!--Text comment-->

\textbf{Text ignored in HTML}

<em>HTML ignored in pdfs</em>

<http://www.rstudio.com>

[link](www.rstudio.com)

Jump to [Header 1](#anchor)

image:

block quote

## Header1

## Header 2

### Header 3

#### Header 4

#### Header 5

#### Header 6

HTML ignored in pdfs

<http://www.rstudio.com>

link

Jump to Header 1

image:

Caption

\* unordered list

- + sub-item 1
- + sub-item 2
- sub-sub-item 1

\* item 2

Continued (indent 4 spaces)

1. ordered list
2. item 2
  - i) sub-item 1
    - A. sub-sub-item 1

(@) A list whose numbering continues after

(@) an interruption

Term 1

- unordered list
  - sub-item 1
  - sub-item 2
    - sub-sub-item 1
- item 2

Continued (indent 4 spaces)

1. ordered list
2. item 2
  - i. sub-item 1
    - A. sub-sub-item 1

1. A list whose numbering continues after

2. an interruption

Term 1

: Definition 1

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

- slide bullet 1
- slide bullet 2

(-> to have bullets appear on click)

horizontal rule/slide break:

\*\*\*

Definition 1

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

- slide bullet 1
- slide bullet 2

(-> to have bullets appear on click)

horizontal rule/slide break:

A footnote <sup>1</sup>

A footnote [<sup>1</sup>]

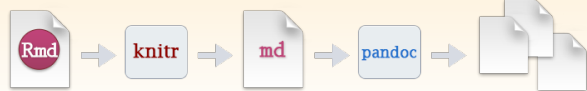
[<sup>1</sup>]: Here is the footnote.

A footnote <sup>1</sup>

1. Here is the footnote. ↩

# Set render options with YAML

1. runs the R code, embeds results and text into .md file with knitr
2. then converts the .md file into the finished format with pandoc



Set a document's default output format in the YAML header:

```

output: html_document

```

output value	creates
html_document	html
pdf_document	pdf (requires Tex)
word_document	Microsoft Word (.docx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	Github compatible markdown
ioslides_presentation	ioslides HTML slides
slidy_presentation	slidy HTML slides
beamer_presentation	Beamer pdf slides (requires Tex)

Customize output with sub-options (listed to the right):

```

output: html_document:
 code_folding: hide
 toc_float: TRUE

```

Indent 2 spaces

Indent 4 spaces

**html tabsets**

Use tabset css class to place sub-headers into tabs

```
Tabset {.tabset .tabset-fade .tabset-pills}
Tab 1
text 1
Tab 2
text 2
End tabset
```

## Create a Reusable Template

1. Create a new package with an inst/rmarkdown/templates directory
2. In the directory, Place a folder that contains: **template.yaml** (see below) **skeleton.Rmd** (contents of the template) any supporting files
3. Install the package
4. Access **template** in wizard at File ► New File ► R Markdown template.yaml

```

name: My Template

```

sub-option	description	html	pdf	word	odt	rtf	md	github	ioslides	slidy	beamer
<b>citation_package</b>	The LaTeX package to process citations, natbib, biblatex or none		X				X				X
<b>code_folding</b>	Let readers to toggle the display of R code, "none", "hide", or "show"	X									
<b>colortheme</b>	Beamer color theme to use										X
<b>css</b>	CSS file to use to style document	X							X	X	
<b>dev</b>	Graphics device to use for figure output (e.g. "png")	X	X				X	X	X	X	X
<b>duration</b>	Add a countdown timer (in minutes) to footer of slides										X
<b>fig_caption</b>	Should figures be rendered with captions?	X	X	X	X				X	X	X
<b>fig_height, fig_width</b>	Default figure height and width (in inches) for document	X	X	X	X	X	X	X	X	X	X
<b>highlight</b>	Syntax highlighting: "tango", "pygments", "kate", "zenburn", "textmate"	X	X	X							X
<b>includes</b>	File of content to place in document (in_header, before_body, after_body)	X	X		X		X	X	X	X	X
<b>incremental</b>	Should bullets appear one at a time (on presenter mouse clicks)?									X	X
<b>keep_md</b>	Save a copy of .md file that contains knitr output	X		X	X	X				X	X
<b>keep_tex</b>	Save a copy of .tex file that contains knitr output	X									X
<b>latex_engine</b>	Engine to render latex, "pdflatex", "xelatex", or "lualatex"		X								X
<b>lib_dir</b>	Directory of dependency files to use (Bootstrap, MathJax, etc.)	X								X	X
<b>mathjax</b>	Set to local or a URL to use a local/URL version of MathJax to render equations	X								X	X
<b>md_extensions</b>	Markdown extensions to add to default definition or R Markdown	X	X	X	X	X	X	X	X	X	X
<b>number_sections</b>	Add section numbering to headers	X	X								
<b>pandoc_args</b>	Additional arguments to pass to Pandoc	X	X	X	X	X	X	X	X	X	X
<b>preserve_yaml</b>	Preserve YAML front matter in final document?								X		
<b>reference_docx</b>	docx file whose styles should be copied when producing docx output			X							
<b>self_contained</b>	Embed dependencies into the doc	X								X	X
<b>slide_level</b>	The lowest heading level that defines individual slides										X
<b>smaller</b>	Use the smaller font size in the presentation?										X
<b>smart</b>	Convert straight quotes to curly, dashes to em-dashes, ... to ellipses, etc.	X									X
<b>template</b>	Pandoc template to use when rendering file quarterly_report.html).	X	X		X						X
<b>theme</b>	Bootswatch or Beamer theme to use for page	X									X
<b>toc</b>	Add a table of contents at start of document	X	X	X		X	X	X			X
<b>toc_depth</b>	The lowest level of headings to add to table of contents	X	X	X		X	X	X			
<b>toc_float</b>	Float the table of contents to the left of the main content	X									

## Table Suggestions

Several functions format R data into tables

eruptions	waiting
3.600	79
1.800	54
3.333	74
2.283	62

eruptions	waiting
1	3.600 79.00
2	1.80 54.00
3	3.33 74.00
4	2.28 62.00

```
data <- faithful[1:4,]
knitr::kable(data, caption = "Table with kable")

knitr::stargazer(data, caption = "Table with stargazer",
 type = "html", html.table.attributes = "border=0")

stargazer::stargazer(data, type = "html", title = "Table with stargazer")
```

## Citations and Bibliographies

Create citations with .bib, .bibtex, .copac, .enl, .json, .medline, .mods, .ris, .wos, and .xml files

1. **Set bibliography file** and CSL 1.0 Style file (optional) in the YAML header
2. **Use citation keys in text**
3. **Render.** Bibliography will be added to end of document

```

bibliography: refs.bib
cs1: style.csl

```

Smith cited [@smith04].  
Smith cited without author [-@smith04].  
@smith04 cited in line.

Smith cited (Joe Smith 2004).  
Smith cited without author (2004).  
Joe Smith (2004) cited in line.

