rmarkdown : : CHEAT SHEET

What is rmarkdown?

.Rmd files • Develop your code and ideas side by side in a single document. Run code as individual chunks or as an entire document.

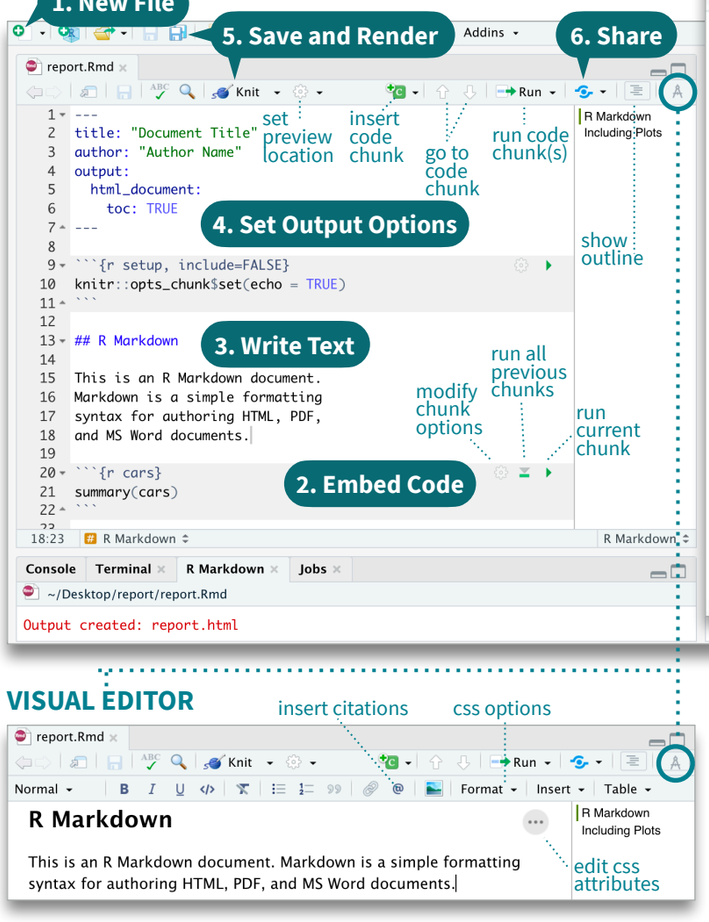
Dynamic Documents • Knit together plots, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS Powerpoint.

Reproducible Research • Upload, link to, or attach your report to share. Anyone can read or run your code to reproduce your work.

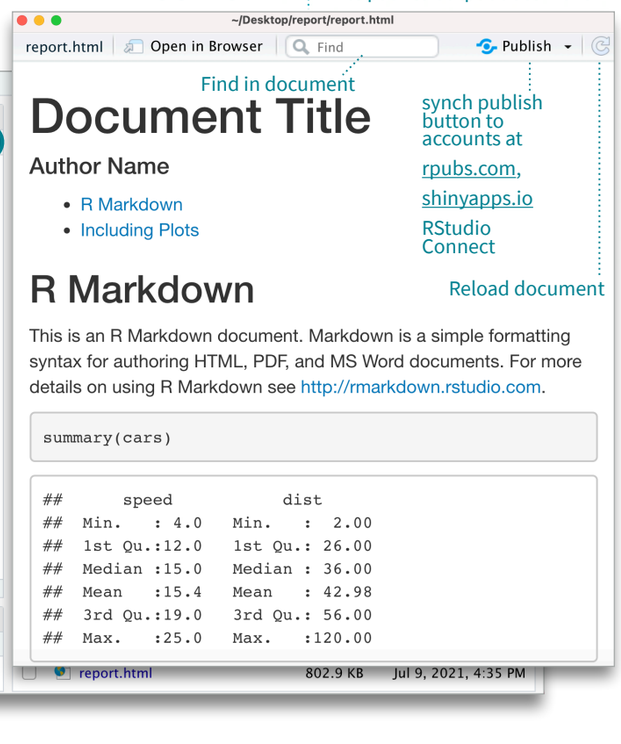
Workflow

1. Open a **new .rmd file** in the RStudio IDE by going to File > New File > R Markdown.
2. **Embed code** in chunks. Run code by line, by chunk, or all at once.
3. **Write text** and add tables, figures, images, and citations. Format with Pandoc syntax or the RStudio IDE Visual Editor.
4. **Set output format(s) and options** in the YAML header. Add parameters to execute or add interactivity with Shiny.
5. **Save and knit** to render the whole document. Knit periodically to preview your work as you write.
6. **Share your work!**

SOURCE EDITOR



RENDERED OUTPUT



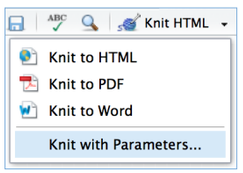
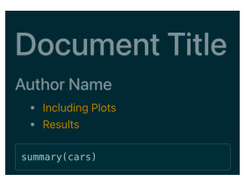
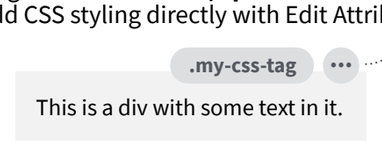
VISUAL EDITOR

Embed Code with knitr

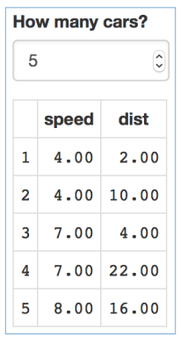
CODE CHUNKS

Surround code chunks with ```` ```{r} ```` and ```` ``` ````. Also insert with the Insert Code Chunk button.

Add chunk options or a chunk label after **r**, within the curly braces.

```
```{r chunk-label, include = FALSE}
#your code goes here
summary(mtcars)
```
```

SET GLOBAL OPTIONS

Use knitr::opts_chunk$set(<chunk options>) to set options for the entire document.

```
```{r include = FALSE}
knitr::opts_chunk$set(message = FALSE)
```
```

INLINE CODE

Insert **`r <code>`** into text sections. Results appear as text without code.

"Built with `r getRversion()`" --> "Built with 4.1.0"

IMPORTANT CODE CHUNK OPTIONS

OPTION	DEFAULT	DESCRIPTION
echo	TRUE	display code in output document
error	FALSE	TRUE (display error messages in doc) FALSE (stop render when error occurs)
eval	TRUE	run code in chunk
include	TRUE	include chunk in doc after running
message	TRUE	display code messages in document
warning	TRUE	display code warnings in document
results	"markup"	"asis" (passthrough results) "hide" (don't display results) "hold" (put all results below all code)
fig.align	"default"	"left", "right", or "center"
fig.cap	NULL	figure caption as a character string
fig.path	"figure/"	prefix for generating figure file paths
out.width & out.height		output width, e.g. "75%", "300px"
highlight	TRUE	highlight source code
collapse	FALSE	collapse all output into a single block
comment	"##"	prefix for each line of results
child	NULL	files(s) to knit and then include
purl	TRUE	include or exclude a code chunk when extracting source code
ref.label	NULL	a character vector of chunk names that the current chunk inherits

See more options and defaults by running str(knitr::opts_chunk$get())

Insert Citations

Create citations from a bibliography file, a Zotero library, or from DOI references.

1. Add BibTeX or CSL bibliographies to the YAML header.
```
---
title: "My Document"
bibliography: references.bib
link-citations: TRUE
---
```
2. If Zotero is installed locally, your main library will automatically be available.
3. Add citations by DOI by searching "from DOI" in the **Insert Citation** dialog.

Access the **Insert Citations** dialog in the Visual Editor by clicking the **@** symbol in the toolbar or by clicking **Insert > Citation**. Add citations with markdown syntax by typing **[@cite]** or **@cite**.

Insert Tables

Convert data frames to output as tables using **kable(**data, caption**)**.

Table with kable

eruptions	waiting
3.600	79
1.800	54
3.333	74
2.283	62

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

Other table packages include flextable, gt, and kableExtra.

Write Text with pandoc

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. Also end with a backslash\ to make a new line.	Plain text. End a line with two spaces to start a new paragraph. Also end with a backslash to make a new line.
italics and **bold** superscript^2^/subscript~2~ ~~strikethrough~~ escaped: * _ \\ endash: --, emdash: ---	*italics* and **bold** superscript^2/subscript$_2$ ~~strikethrough~~ escaped: * _ \ endash: –, emdash: —
# Header 1 ## Header 2 ... ###### Header 6	**Header 1** **Header 2** ... **Header 6**
- unordered list - item 2 - item 2a (indent 2 tabs) - item 2b 1. ordered list 2. item 2 - item 2a (indent 2 tabs) - item 2b	• unordered list • item 2 • item 2a (indent 2 tabs) • item 2b 1. ordered list 2. item 2 • item 2a (indent 2 tabs) • item 2b
<link url> [This is a link.](link url) [This is another link][id]. At the end of the document: [id]: link url ![Caption](image.png) or ![Caption][id2] At the end of the document: [id2]: image.png "Alt text"	http://www.rstudio.com/ This is a link. This is another link.  Caption.
`verbatim code` ``` multiple lines of verbatim code ```	`verbatim code` ``` multiple lines of verbatim code ```
> block quotes	> block quotes
equation: $A = \pi*r^2{2}$ equation block: $$E = mc^{2}$$	equation: $A = \pi * r^2$ equation block: $$E = mc^2$$
horizontal rules: *** ---	horizontal rules: ---
HTML Tabsets: # Results {.tabset} ## Plots text ## Tables more text	HTML Tabsets: **Results** Plots | Tables text
Bracketed Span: A [green]{.my-color} word. Fenced Div: ::: {.my-color} All of these words are green. :::	Bracketed Span: A green word. Fenced Div: All of these words are green.

Define .my-color in an attached .css/.sass/.scss file. (see back).

Set Output Formats and their Options with YAML

Set the output format and options in the document YAML header. Choose an **output value** and further customize it with **output options**.

```
---
title: "My Document"
author: "Author Name"
output:
 html_document:
  toc: TRUE
---
```

Indent values 1 tab, options 2 tabs

Output Value	Creates
html_document	html
pdf_document	pdf (requires Tex)
word_document	Microsoft Word (.docx)
powerpoint_presentation	Microsoft Powerpoint (.pptx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	markdown for Github
ioslides_presentation	ioslides HTML slides
slidy_presentation	slidy HTML slides
beamer_presentation	Beamer slides (requires Tex)

IMPORTANT OPTIONS	DESCRIPTION	HTML	PDF	MS Word	MS PPT
dev	Graphics device to use for figure output (e.g. "png", "pdf")	X	X		
includes	File of content to place in doc ("in_header", "before_body", "after_body")	X	X		
toc	Add a table of contents at start of document?	X	X	X	X
toc_depth	The lowest level of headings to add to table of contents (e.g. 2, 3)	X	X	X	X
fig_caption	Should figures be rendered with captions?	X	X	X	X
theme	Theme options (see Bootswatch and Custom Themes below)	X			
toc_float	Float the table of contents next to the left of the document?	X			
code_download	Give readers an option to download the .Rmd source code?	X			
code_folding	Let readers to toggle the display of R code ("none", "hide", or "show")	X			
css	CSS file to use to style document (e.g. "style.css")	X			
keep_md	Save a copy of .md file that contains knitr output?	X	X	X	X
highlight	Syntax highlighting ("tango", "pygments", "kate", "zenburn", "textmate")	X	X	X	
anchor_sections	Show section anchors on mouse hover?	X			
df_print	Method for printing data frames ("default", "kable", "tibble", "paged")	X	X	X	X
reference_docx/_doc	docx/pptx file containing styles to copy in the output (e.g. "file.docx", "file.pptx")			X	X
keep_tex*	Save a copy of .tex file that contains knitr output?		X		
latex_engine*	Engine to render latex ("pdflatex", "xelatex", or "lualatex")		X		
citation_package*	The LaTeX package to process citations ("default", "natbib", "biblatex")		X		

*for options that require Tex, we recommend the **tinytex** package

To get a list of all options for an output type, read the format's help page, e.g. **?html_document**
Other packages add additional formats, see **flexdashboard**, **bookdown**, **distill**, and **blogdown**.

Render

When you render, R Markdown:

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

Save, then **Knit** to preview the document output. The resulting HTML/PDF/MS Word/etc document will be created and saved in the same directory as the .Rmd file.

Use rmarkdown::**render()** to render/knit at cmd line. ? render for args

Share

Publish on RStudio Connect, to share R Markdown documents securely, schedule automatic updates, and interact with parameters in real time.
www.rstudio.com/products/connect/

More Header Options

PARAMETERS

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

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

```
---
params:
  state: "hawaii"
---
```

2. **Call parameters** in code using params$<name>

```
```{r}
data <- df[, params$data]
summary(data)
```
```

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



REUSABLE TEMPLATES

1. **Create a new package** with a inst/rmarkdown/templates directory
2. **Add a folder** containing **template.yaml** (below) and **skeleton.Rmd** (template contents)

```
---
name: "My Template"
---
```

3. **Install** the package to access template by going to **File > New R Markdown > From Template**

BOOTSWATCH THEMES

Customize documents with the **bslib** library. Add the "theme:" option underneath the output type in the YAML header.

bslib comes with many built in bootswatch themes. Get a list of themes with **bootswatch_themes()**.




```
---
title: "My Document"
author: "Author Name"
output:
 html_document:
  theme:
   bootswatch: solar
---
```

CUSTOM THEMES

Also specify exactly which elements to customize with with theme options.

```
---
output:
 html_document:
  theme:
   bg: "#121212"
   fg: "#E4E4E4"
   base_font: !expr
      bslib::font_google("Open Sans")
---
```

More on **bslib** **rstudio.github.io/bslib/**.

STYLING WITH CSS and SCSS

Add CSS and SCSS to your document by adding a file to the header options using the **css** option, or by adding a CSS chunk.

```
---
title: "My Document"
author: "Author Name"
output:
 html_document:
  css: "style.css"
---
```

```
```{css, echo=FALSE}
.my-color {
  color: green;
}
```
```

Apply CSS styling by writing HTML tags directly or

1. Add bracketed spans or fenced divs in the Source Editor.

```
Bracketed Span:
A [green]{.my-color} word.
Fenced Div:
::: {.my-color}
All of these words
are green.
:::
```

A green word.

All of these words are green.

2. Or go to **Format > Div/Span** in the Visual Editor. Add CSS styling directly with Edit Attributes.



INTERACTIVITY

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

```
---
output: html_document
runtime: shiny
---
```

```
```{r, echo = FALSE}
numericInput("n",
  "How many cars?", 5)

renderTable({
  head(cars, input$n)
})
```
```



Call **bslib:bs_themer()** in any code chunk to customize theme options.

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