

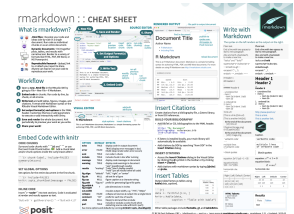
# rmarkdown :: Cheatsheet

## 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.



Download PDF



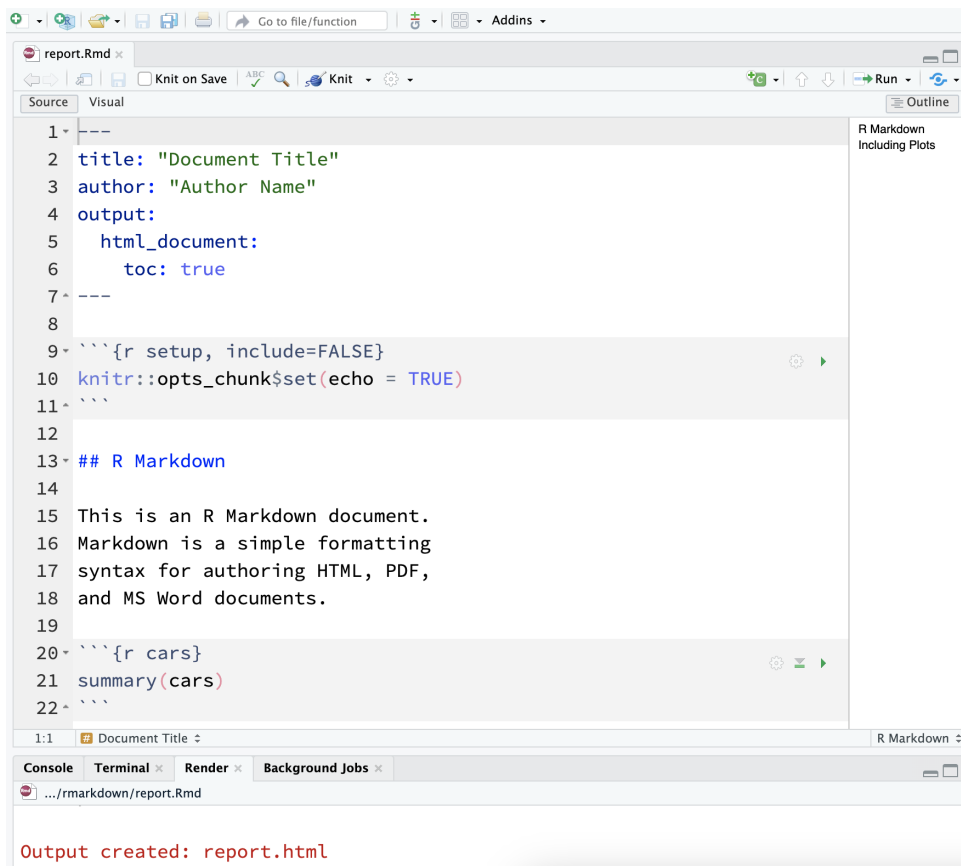
## Workflow

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 Markdown syntax or the RStudio Visual Markdown Editor.
4. **Set output format(s) and options** in the YAML header. Customize themes or add parameters to execute or add interactivity with Shiny.
5. **Save and render** the whole document. Knit periodically to preview your work as you write.
6. **Share your work!**

Translations (PDF)

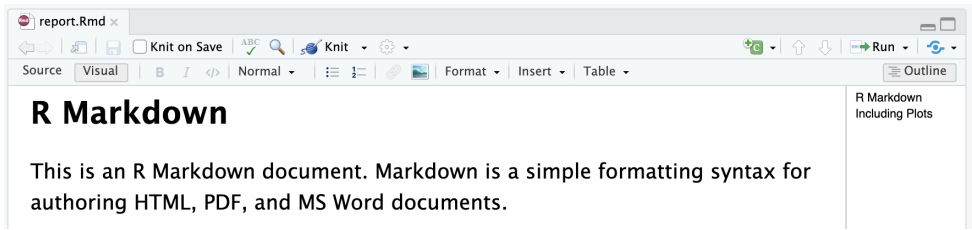
- Dutch
- German
- Italian
- Japanese
- Korean
- Spanish
- Turkish
- Vietnamese

## Source Editor



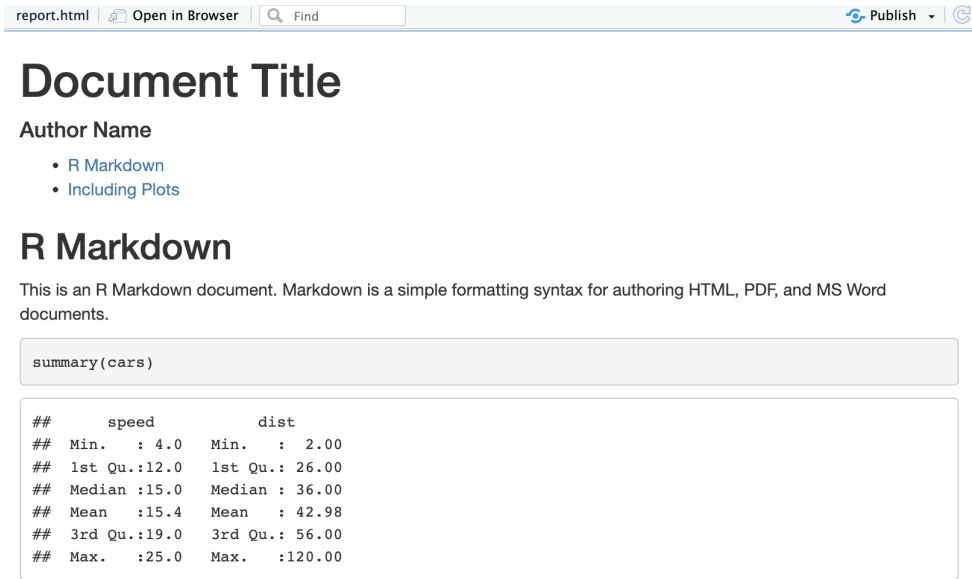
Expand to read about the features in the Source Editor

# Visual Editor



Expand to read about the features in the Visual Editor

# Rendered Output



Expand to read about the features in the Rendered Output Window

# Embed Code With knitr

## Code Chunks

Surround code chunks with ``r`` and ``r`` or use the Insert Code Chunk button. Add a chunk label and/or chunk options inside the curly braces after `r``.

```
{{r chunk-label, include = FALSE}}
```

## Set Global Options

Set options for the entire document in the first chunk.

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

## Inline Code

Insert ``r <code>`` into text sections. Code is evaluated at render and results appear as text.

The markdown text

Built with ``r getRversion()``

will render as “Built with 4.3.0” in the output file.

## Chunk Options

Table of chunk options. The first column is the option name, the second column is the option’s default value, the third column describes what the option does.

Option	Default	Effects
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" (pass through results), "hide" (don't display results), "hold" (put all results below all code)
fig.align	"default"	"left", "right", or "center"
fig.alt	NULL	alt text for a figure
fig.cap	NULL	figure caption as a character string
fig.path	"figure/"	prefix for generating file paths
fig.width & fig.height	7	plot dimensions in inches
out.width		rescales output width, e.g. "75%", "300px"
collapse	FALSE	collapse all sources & output into a single block
comment	"##"	prefix for each line of results
child	NULL	file(s) to knit and then include
purl	TRUE	include or exclude a code chunk when extracting source code with <code>knitr::purl()</code>

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.

### Build Your Bibliography

- Add BibTex or CSL bibliographies to the YAML header.

```
---
title: "My Document"
bibliography: references.bib
link-citations: TRUE
---
```

- If Zotero is installed locally, your main library will automatically be available.
- Add citations by DOI by searching “from DOI” in the **Insert Citation** dialog.

### Insert Citations

- 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

Output data frames as tables using `kable(data, caption)`.

```
` ` {{r}}
data <- faithful[1:4,]
knitr::kable(data, caption = "Tables with kable")
` `
```

Other table packages include **flextable**, **gt**, and **kableExtra**.

# Write With Markdown

The syntax on the left renders as the output on the right.

Table of markdown syntax and rendered examples. The syntax in the first column renders to the output in the second column.

Markdown Syntax	Rendered Output
Plain text.	Plain text.
End a line with two spaces to start a new paragraph.	End a line with two spaces to start a new paragraph.
Also end a line with a backslash\ to make a new line.	Also end a line with a backslash to make a new line.
<i>*italics*</i> and <b>**bold**</b>	<i>italics</i> and <b>bold</b>
superscript <sup>2</sup> /subscript <sub>2</sub>	superscript <sup>2</sup> /subscript <sub>2</sub>
~strike through~	<del>strike through</del>
escaped: \* \_ \\	escaped: * _ \
en dash: --, em dash: ---	en dash: –, em dash: —
# Header 1	Header 1
## Header 2	Header 2
...	...
##### Header 6	Header 6
- unordered list - item 2 - item 2a (indent 1 tab) - item 2b	<ul style="list-style-type: none"><li>• unordered list</li><li>• item 2<ul style="list-style-type: none"><li>◦ item 2a (indent 1 tab)</li><li>◦ item 2b</li></ul></li></ul>
1. ordered list 2. item 2 - item 2a (indent 1 tab) - item 2b	<ol style="list-style-type: none"><li>1. ordered list</li><li>2. item 2<ul style="list-style-type: none"><li>◦ item 2a</li></ul></li></ol>

<code>&lt;link url&gt;</code>	<code>https://posit.co/</code>																
<code>[This is a link.](link url)</code>	This is a link.																
<code>[This is another link.][id].</code>	This is another link.																
At the end of the document: <code>[id]: link url</code>																	
<code>![Caption](image.png)</code>  or  <code>![Caption](id2)</code>  At the end of the document include: <code>[id2]: image.png</code>																	
<code>`verbatim code`</code>	<code>verbatim code</code>																
<code>```\nmultiple lines\nof verbatim code\n```</code>	multiple lines of verbatim code																
<code>&gt; block quotes</code>	<div>block quotes</div>																
equation: <code>\$e^{i \pi} + 1 = 0\$</code>	equation: $e^{i\pi} + 1 = 0$																
equation block: <code>\$\$E = mc^2\$\$</code>	equation block: $E = mc^2$																
horizontal rule: <code>---</code>	horizontal rule:																
<code> Right Left Default Center \n ----: :--- :----- :----: \n 12   12   12   12   \n 123  123  123  123  \n 1    1    1    1    \n\nTable: Caption text, example rendered table</code>	<table><caption>Caption text, example rendered table.</caption><tr><th>Right</th><th>Left</th><th>Default</th><th>Center</th></tr><tr><td>12</td><td>12</td><td>12</td><td>12</td></tr><tr><td>123</td><td>123</td><td>123</td><td>123</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	Right	Left	Default	Center	12	12	12	12	123	123	123	123	1	1	1	1
Right	Left	Default	Center														
12	12	12	12														
123	123	123	123														
1	1	1	1														
HTML Tabsets  <code>::: panel-tabset\n## Plots text\ntext\n\n## Tables\nmore text\n:::</code>	<div>Plots</div> <div>Tables</div> <div>text</div>																

## Set Output Formats and Their Options in YAML

Use the document's YAML header to set an **output format** and customize it with **output options**. Indent format 2 characters, indent options 4 characters.

```
---
title: "My Document"
author: "Author Name"
output:
  html_document:
    toc: true
    toc-location: left
---
```

## Output Format Table

Table of output formats. The output format in the first column creates the file type in the second column.

Output Format	Creates
html_document	.html
pdf_document <sup>1</sup>	.pdf
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_presentations	ioslides HTML slides
slidy_presentation	Slidy HTML slides
beamer_presentation <sup>2</sup>	Beamer slides

Also see **flexdashboard**, **bookdown**, **distill**, and **blogdown**.

## Output Options Table

Table of output options. The first column is the option name, the second column is the description and possible values, and then remaining columns show what file types each option can be applied to.

Important Options	Description				
		HTML	PDF	MS Word	MS PPT
anchor_sections	Show section anchors on mouse hover (TRUE or FALSE)	X			
citation_package	The LaTeX package to process citations ("default", "natbib", biblatex")				
code_download	Give readers an option to download the .Rmd source code (TRUE or FALSE)	X			
code_folding	Let readers toggle the display of R code ("none", "hide", or "show")	X			
css	CSS or SCSS file to use to style the document (e.g. "style.css")	X			
dev	Graphics device to use for figure output (e.g. "png", "pdf")	X	X		

Important Options	Description	HTML	PDF	MS	MS
				Word	PPT
df_print	Method for printing data frames (“default”, “kable”, “tibble”, “paged”)	X	X	X	X
fig_caption	Should figures be rendered with captions (TRUE or FALSE)	X	X	X	X
highlight	Syntax highlighting (“tango”, “pygments”, “kate”, “zenburn”, “textmate”)	X	X	X	
includes	File of content to place in doc (“in_header”, “before_body”, “after_body”)	X	X		
keep_md	Keep the Markdown .md file generated by knitting (TRUE or FALSE)	X	X	X	X
keep_tex	Keep the intermediate TEX file used to convert to PDF (TRUE or FALSE)		X		
latex_engine	LaTeX engine for producing PDF output (“pdflatex”, “xelatex”, or “lualatex”)		X		
reference_docx/_doc	docx/pptx file containing styles to copy in the output (e.g. “file.docx”, “file.pptx”)			X	X
theme	Theme options (see Bootswatch and Custom Themes below)	X			
toc	Add a table of contents at start of document (TRUE or FALSE)	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
toc_float	Float the table of contents to the left of the main document content (TRUE or FALSE)	X			

Use `?<output_format>` to see all of a format’s options, e.g. `?html_document`

## 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$state]
summary(data)
```
```

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

### Reusable Templates

1. **Create a new package** with an `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**.

## Bootstrap Themes

Customize HTML documents with Bootstrap themes from the **bslib** package using the theme output option. Use `bslib::bootstrap_themes()` to list available themes.

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

## Custom Themes

Customize individual HTML elements using bslib variables. Use `?bs_theme` to see more variables.

```
---
output:
  html_document:
    theme:
      bg: "#121212"
      fg: "#E4E4E4"
      base_font:
        google: "Prompt"
---
```

More on **bslib** at <https://pkgs.rstudio.com/bslib/>.

## Styling With CSS and SCSS

Add CSS and SCSS to your documents by adding a path to a file with the **css** option in the YAML header.

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

Apply CSS styling by writing HTML tags directly or:

- Use markdown to apply style attributes inline.
  - Bracketed Span  
A `[green]{.my-color}` word. will render as “A **green** word.”

- Fenced Div

```
:::{.my-color}
All of these words
are green
:::
```

will render as

**All of these words**

**are green.**

- Use the Visual Editor. Go to **Format > Div/Span** and 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.

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



```
runtime: shiny
```

2. Call Shiny input functions to embed input objects.

3. Call Shiny output functions to embed reactive output.

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

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

4. Render with `rmarkdown::run()` or click **Run Document** in RStudio IDE.

Also see Shiny Prerendered for better performance.

[https://rmarkdown.rstudio.com/authoring\\_shiny\\_prerendered](https://rmarkdown.rstudio.com/authoring_shiny_prerendered).

Embed a complete Shiny app into your document with `shiny::shinyAppDir()`. More at

<https://bookdown.org/yihui/rmarkdown/shiny-embedded.html>.

## Render

When you render a document, rmarkdown:

1. Runs the code and embeds results and text into an .md file with knitr.
2. Converts the .md file into the output 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 in the R console. See `?render` for available options.

## Share

**Publish on Posit Connect** to share R Markdown documents securely, schedule automatic updates, and interact with parameters in real-time. <https://posit.co/products/enterprise/connect/>.

CC BY SA Posit Software, PBC • [info@posit.co](mailto:info@posit.co) • [posit.co](https://posit.co)

Learn more at [rmarkdown.rstudio.com](https://rmarkdown.rstudio.com).

Updated: 2023-06.

```
packageVersion("rmarkdown")
```

```
[1] '2.21'
```

### Footnotes

1. PDFs and Beamer slides require LaTeX, use `tinytex::install_tinytex()`.↵
2. PDFs and Beamer slides require LaTeX, use `tinytex::install_tinytex()`.↵