

# **Dynamic documents in R**

**reproducible research with R  
Markdown**

**2021-03-04**

# Rmarkdown

TEXT. CODE. OUTPUT.  
(GET IT TOGETHER, PEOPLE.)



Artwork by @allison\_horst

# R Markdown



**Authoring framework: code and text in same document**

**Reproducible: re-run your analysis**

**Flexible: Output to different formats easily**



## Your turn 1

Create a new R Markdown file. Go to File > New File > R Markdown. Press OK. Save the file and press the "Knit" button above.

The screenshot shows an RStudio interface with the following components:

- Left Panel (Code Editor):** Displays an R Markdown file named "1-example.Rmd". The code includes YAML metadata at the top, followed by plain text describing a plot, and a code chunk for generating a contour map.
- Middle Panel (Text View):** Shows the plain text content from the R Markdown file.
- Right Panel (Viewer):** Shows the generated output, which is a contour map of the Maunga Whau volcano in Auckland, New Zealand.

Annotations with orange curly braces highlight specific parts of the code:

- A brace on the left side of the YAML metadata block groups it together.
- A brace on the right side of the plain text block groups it together.
- A brace on the right side of the code chunk block groups it together.

```
1 ---  
2 title: "Viridis Demo"  
3 output: html_document  
4 ---  
5  
6 ```{r include = FALSE}  
7 library(viridis)  
8 ```  
9  
10 The code below demonstrates two color palettes in the  
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each  
12 plot displays a contour map of the Maunga Whau volcano in  
13 Auckland, New Zealand.  
14 ## Viridis colors  
15 image(volcano, col = viridis(200))  
16 ```  
17  
18 ## Magma colors  
19  
20 image(volcano, col = viridis(200, option = "A"))  
21 ```  
22  
23
```

# R Markdown

Prose

Code

Metadata



# R Markdown

**Prose = Markdown**

Code

Metadata



# Visual R Markdown

The screenshot shows an R Markdown document titled "relational-data.Rmd". The title is "Filtering joins". The text explains that filtering joins match observations in the same way as [mutating joins](#), but affect the observations, not the variables<sup>1</sup>. There are two types:

<code>semi_join(x, y)</code>	$x \ltimes y$	Keeps all observations in $x$ that have a match in $y$
<code>anti_join(x, y)</code>	$x \triangleright y$	Drops all observations in $x$ that have a match in $y$

Graphically, a semi-join looks like this:

```
{r, echo = FALSE, out.width = NULL}
knitr::include_graphics("diagrams/join-semi.png")
```

The diagram illustrates a semi-join operation between two tables,  $x$  and  $y$ . Table  $x$  has columns  $x_1, x_2, x_3$  and rows 1, 2, 3. Table  $y$  has columns  $y_1, y_2, y_3$  and rows 1, 2, 3. A green dot at  $(x_1, 1)$  connects to a purple dot at  $(y_1, 1)$ . A yellow dot at  $(x_2, 2)$  connects to a purple dot at  $(y_2, 2)$ . An orange dot at  $(x_3, 3)$  has no connection. An arrow points from the joined tables to a resulting table:

key	val_x
1	x1
2	x2

Only the existence of a match is important; it doesn't matter which observation is matched. This means that filtering joins never duplicate rows like mutating joins do:

# Basic Markdown Syntax

\*italic\*    \*\*bold\*\*

\_italic\_    \_\_bold\_\_

# Basic Markdown Syntax

```
# Header 1  
## Header 2  
### Header 3
```

# Basic Markdown Syntax

`http://example.com`

`[linked phrase](http://example.com)`

**Learn more about Markdown Syntax  
with the ten-twenty minute tutorial on  
markdown at  
[https://commonmark.org/help/tutorial.](https://commonmark.org/help/tutorial)**

## Your turn 2 (open `exercises.Rmd`)

Read this short introduction to Visual R Markdown:

<https://rstudio.github.io/visual-markdown-editing/#/intro?id=getting-started>

Use Visual R Markdown to stylize the text from the Gapminder website below. Experiment with bolding, italicizing, making lists, etc.

# R Markdown

Prose

**Code = R code chunks**

Metadata



## Code chunks

```
```{r select_example, echo = FALSE}
gapminder %>%
  select(year, country)
````
```

## Code chunks

fences (3  
backticks)

```
{r select_example, echo = FALSE}  
gapminder %>%  
  select(year, country)
```

## Code chunks

chunk name

```
```{r select_example, echo = FALSE}  
gapminder %>%  
  select(year, country)  
```
```

chunk  
arguments

# Chunk options

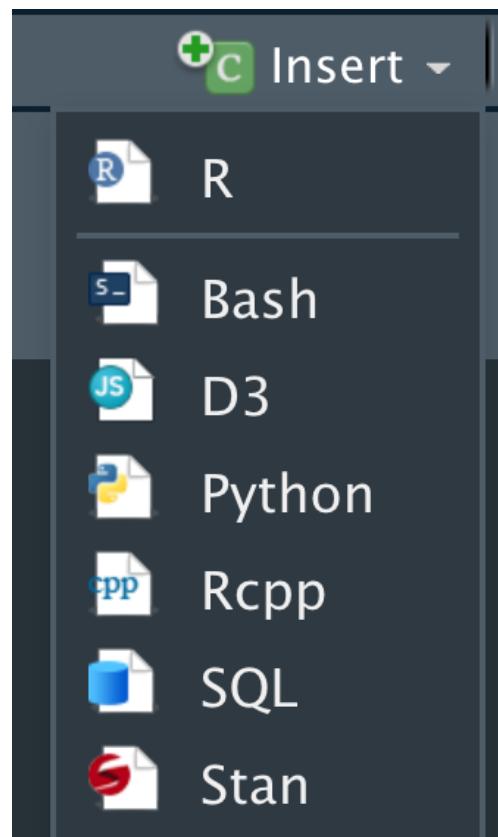
| Option                            | Effect                                       |
|-----------------------------------|--|
| <code>include = FALSE</code>      | run the code but don't print it or results   |
| <code>eval = FALSE</code>         | don't evaluate the code                      |
| <code>echo = FALSE</code>         | run the code and output but don't print code |
| <code>message = FALSE</code>      | don't print messages (e.g. from a function)  |
| <code>warning = FALSE</code>      | don't print warnings                         |
| <code>fig.cap = "Figure 1"</code> | caption output plot with "Figure 1"          |

See the [knitr web page](#)

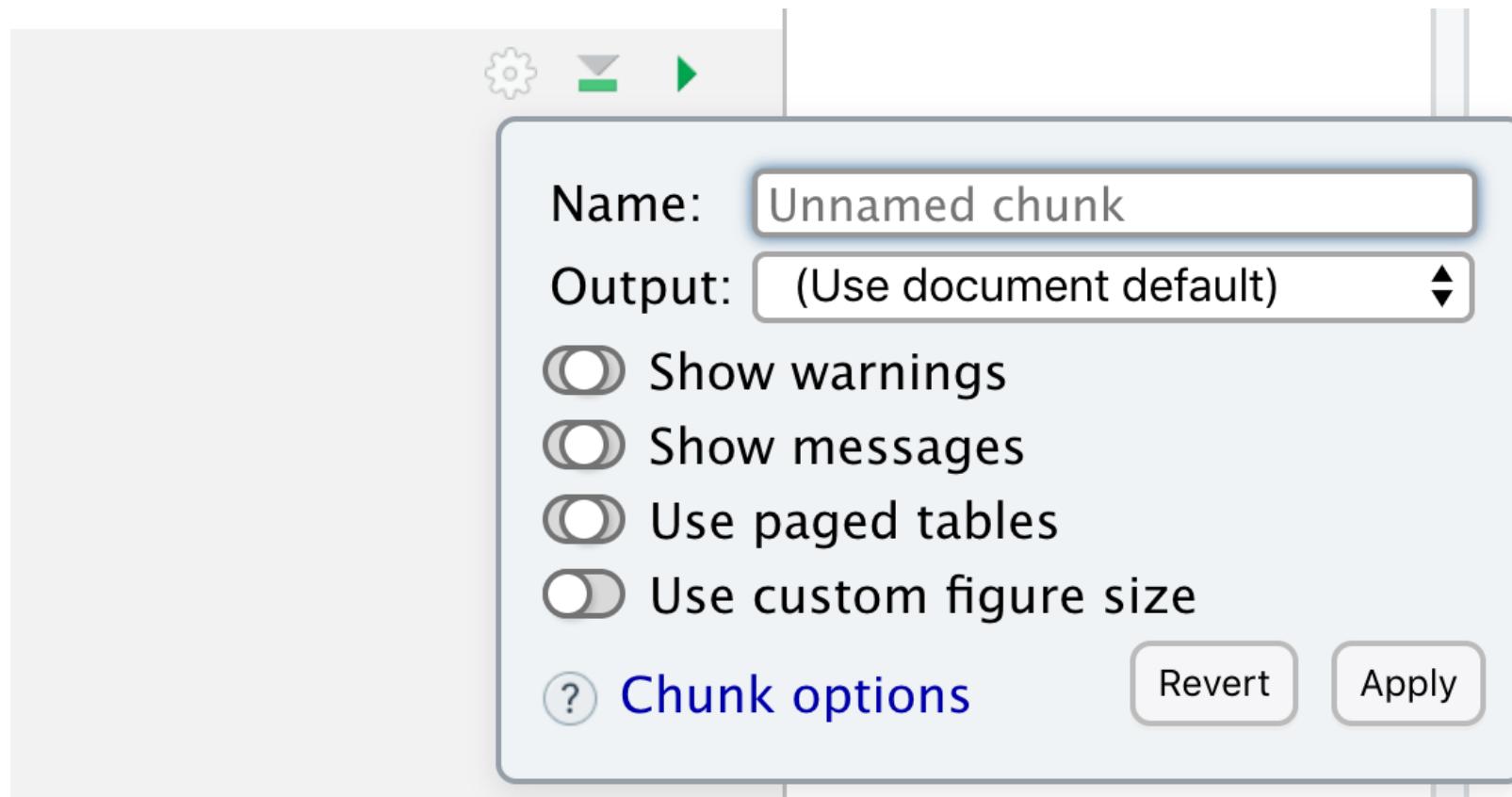
# Engines

52! Including **Python, Julia, C++, SQL, SAS, and Stata**

# Insert code chunks with cmd/ctrl + alt/option + I



# Edit code chunk options



## Your turn 3

**Create a code chunk. You can type it in manually, use the keyboard short-cut (Cmd/Ctrl + Option/Alt + I), or use the "Insert" button above. Put the following code in it:**

```
gapminder %>%
  slice(1:5) %>%
  knitr::kable()
```

**Knit the document**

## Your turn 4

**Add echo = FALSE to the code chunk above and re-knit**

**Remove echo = FALSE from the code chunk and move it to knitr::opts\_chunk\$set() in the setup code chunk. Re-knit. What's different about this?**

**Make sure to remove knitr::opts\_chunk\$set(echo = FALSE)**

# Inline Code

Lore*m ipsum dolor sit  
amet, consetetur  
sadipscing  
`r max(gapminder\$year)`  
elitr, sed diam nonumy  
eirmod tempor invidunt*

# Inline Code

backticks  
+ r  
`r max(gapminder\$year)`  
any R code

## Your turn 5

**Remove eval = FALSE so that R Markdown evaluates the code.**

**Use summarize() and n\_distinct() to get the the number of unique years in gapminder and save the results as n\_years.**

**Use inline code to describe the data set in the text below the code chunk and re-knit.**

# R Markdown

Prose

Code

**Metadata = YAML**



# YAML Metadata

```
---
author: Malcolm Barrett
title: Quarterly Report
output:
  html_document: default
  pdf_document:
    toc: true
---
```

```
title: "Annual report"  
author: Malcolm Barrett  
date: "`r Sys.Date()`"  
output:  
  pdf_document:  
    toc: true
```

```
title: "Annual report"  
author: Malcolm Barrett  
date: "r Sys.Date()"
```

**key**      **value**

output:

pdf\_document:

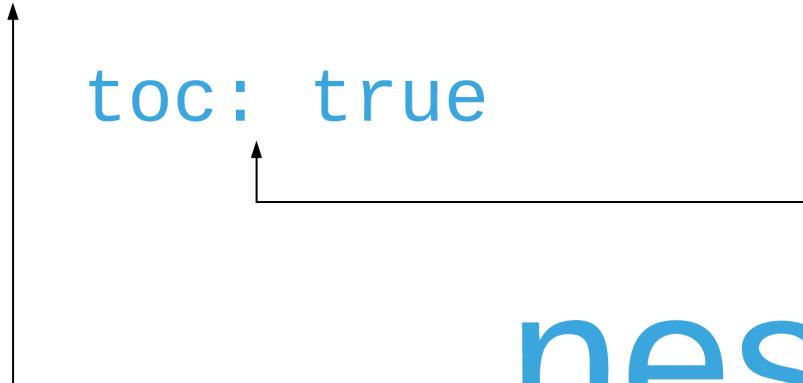
    toc: true

```
title: "Annual report"  
author: Malcolm Barrett  
date: "`r Sys.Date()`"
```

output: ← **top level**

`pdf_document:`

`toc: true`



**nested**

```
title: "Annual report"
```

```
author: Malcolm Barron
```

```
date: " `r Sys.Date() ` "
```

```
output:
```

```
pdf_document:
```

```
  toc: true
```

# output function

```
output
```

# arguments

```
title: "Annual report"  
author: Malcolm Barrett  
date: "`r Sys.Date()`"  
output:  
  pdf_document:  
    toc: true
```

**pdf\_document(toc = TRUE)**

# Output formats

| Function                               | Outputs             |
|--|---------------------|
| <code>html_document()</code>           | HTML                |
| <code>pdf_document()</code>            | PDF                 |
| <code>word_document()</code>           | Word .docx          |
| <code>odt_document()</code>            | .odt                |
| <code>rtf_document()</code>            | .rtf                |
| <code>md_document()</code>             | Markdown            |
| <code>slidy_presentation()</code>      | Slidy Slides (HTML) |
| <code>beamer_presentation()</code>     | Beamer Slides (PDF) |
| <code>ioslides_presentation()</code>   | ioslides (HTML)     |
| <code>powerpoint_presentation()</code> | Powerpoint Slides   |

# Your turn 6

**Set figure chunk options to the code chunk below, such as dpi, fig.width, and fig.height. Run knitr::opts\_chunk\$get() in the console to see the defaults.**

**Add your name to the YAML header using author: Your Name.**

**Change the YAML header above from output: html\_document to output: distill::distill\_article.**

**Set distill::distill\_article to use the toc and code\_folding options and re-knit**

# ymlthis

check out the `ymlthis` package for tools  
and documentation for working with  
`YAML`

<https://r-lib.github.io/ymlthis/>

# Parameters

```
---  
params:  
  param1: x  
  param2: y  
  data: df  
---
```

```
params$param1  
params$param2  
params$data
```

## Your turn 7

**Change the params option in the YAML header to use a different continent. Re-knit**

```
gapminder %>%
  filter(continent == params$continent) %>%
  ggplot(aes(x = year, y = lifeExp, group = country, color = country)
  geom_line(lwd = 1, show.legend = FALSE) +
  scale_color_manual(values = country_colors) +
  theme_minimal(14) +
  theme(strip.text = element_text(size = rel(1.1))) +
  ggtitle(paste("Continent:", params$continent))
```

# Bibliographies and citations

# Bibliographies and citations

**Bibliography files: .bib, Zotero, others**

# Bibliographies and citations

Bibliography files: .bib, Zotero, others

Citation styles: .csl

# Bibliographies and citations

Bibliography files: .bib, Zotero, others

Citation styles: .csl

[@citation-label]

Or just use Visual R Markdown's citation wizard!

# Including bibliography files in YAML

```
---
```

```
bibliography: file.bib
```

```
csl: file.csl
```

```
---
```

**Visual R Markdown can also manage this for you.**

## Your turn 8

**Cite the Causal Inference book in text below. Using the citation wizard, find the right citation under My sources > Bibliography.**

**Add the American Journal of Epidemiology CSL to the YAML using csl: aje.csl**

**Re-knit**

# **Make cool stuff in R Markdown!**

**bookdown**

**blogdown**

**these slides!**

# Resources

**R Markdown:** A comprehensive but friendly introduction to R Markdown and friends. Free online.

**R for Data Science:** A comprehensive but friendly introduction to the tidyverse. Free online.

**R Markdown for Scientists:** R Markdown for Scientists workshop material.