Introduction to R Markdown

https://github.com/psboonstra/markdown-workshop

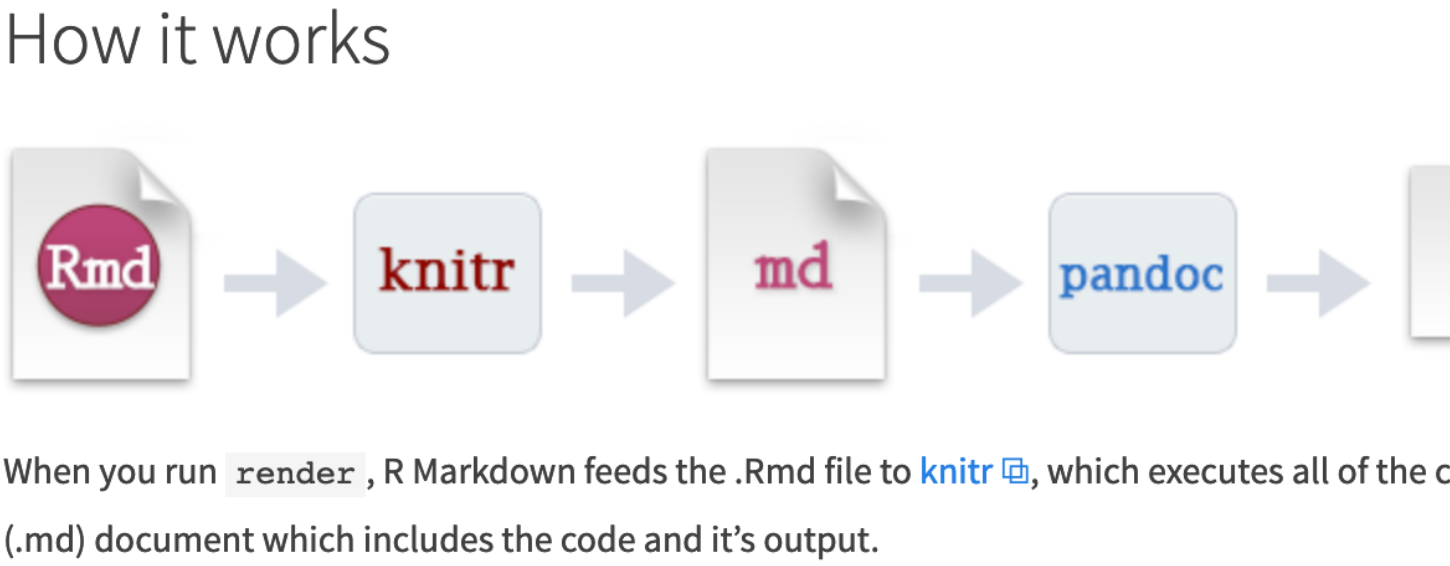
BDSI 2019; Univeristy of Michigan

When to use

- Reports
- Slides
- Manuscripts / books

Why to use

- `R` code and interpretations integrated into a single document
- Separate tasks of *reporting* the results from *formatting* the results:
  - decreases risk of copy-paste errors
  - decreases workload
- Quickly create the same document in different formats, e.g. slides to show and handouts for the audience
- Create websites

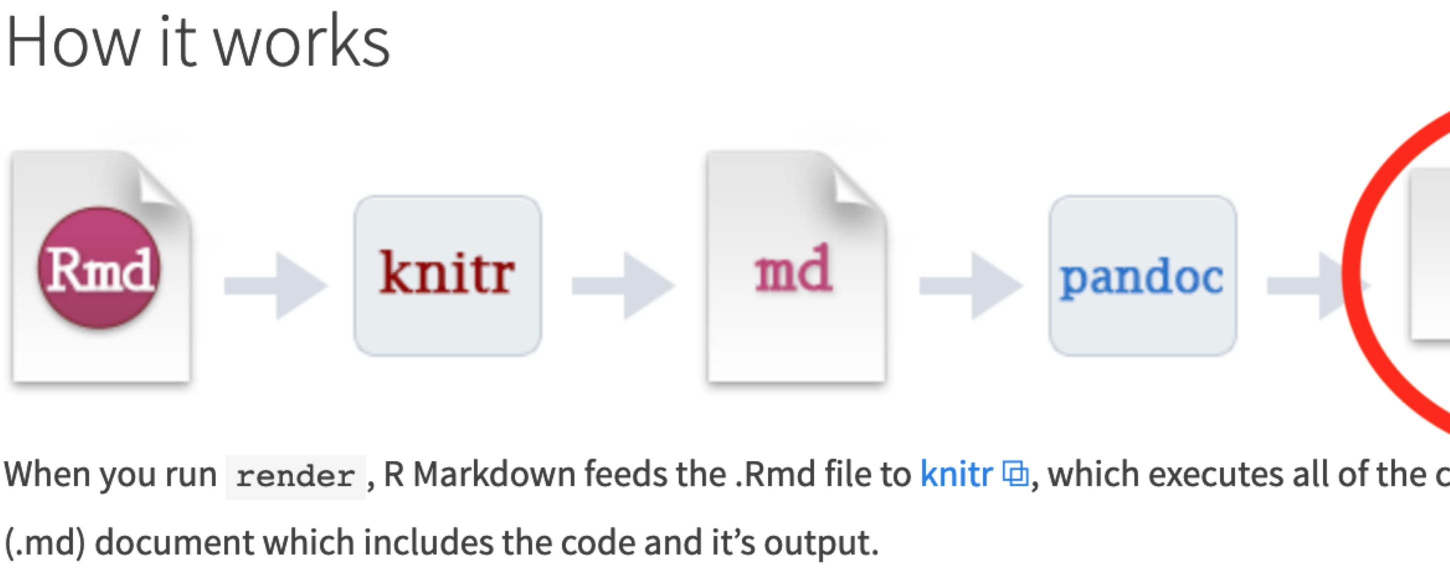


How it works

When you run `render`, R Markdown feeds the .Rmd file to knitr, which executes all of the c

(.md) document which includes the code and it's output.

source: rstudio.com



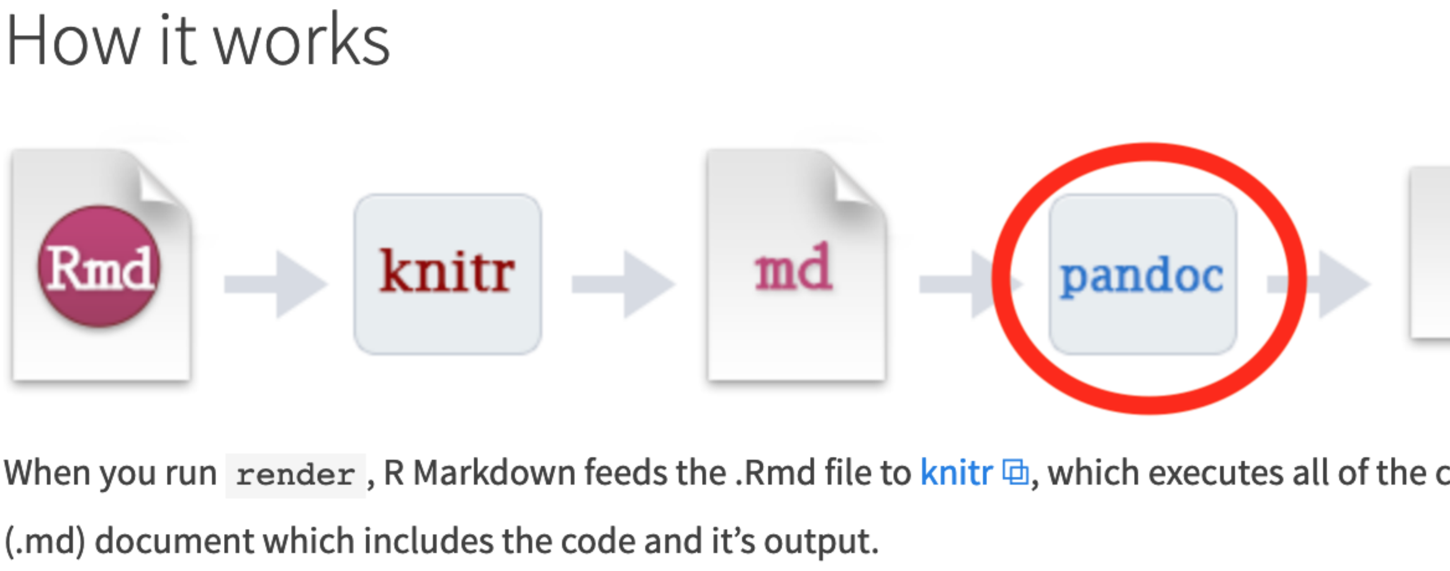
How it works

Rmd → knitr → md → pandoc →

When you run `render`, R Markdown feeds the .Rmd file to knitr, which executes all of the c

(.md) document which includes the code and it's output.

whatever format you want to create: html, pdf, docx, . . .



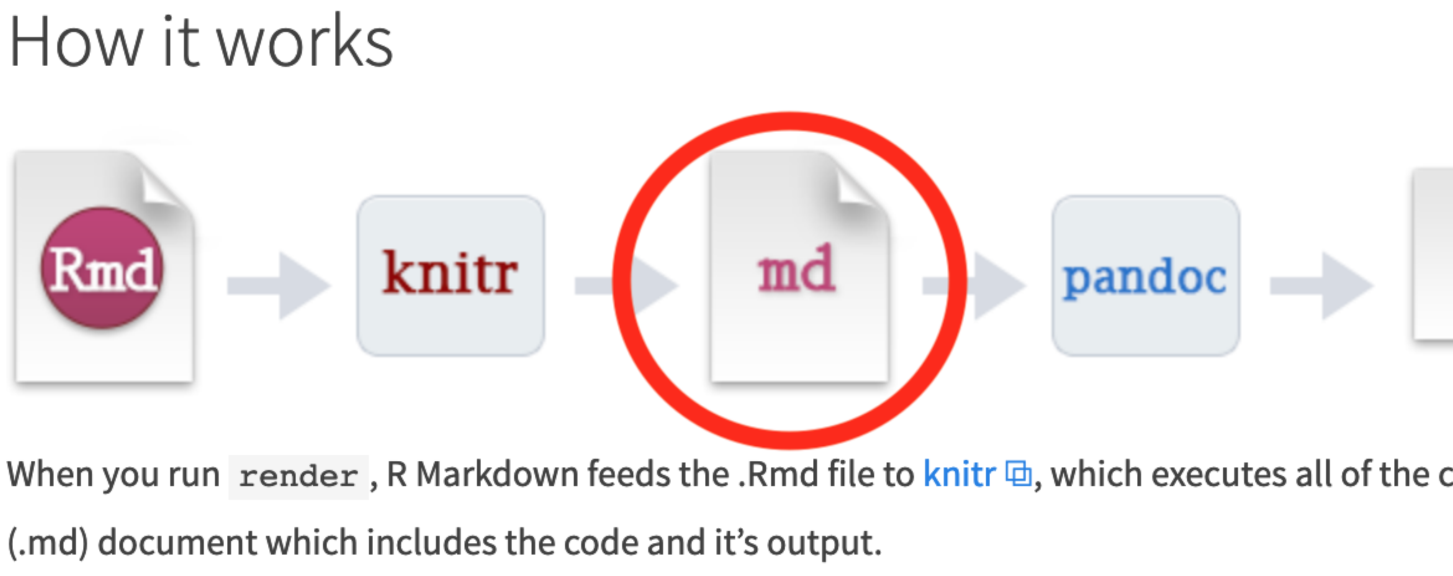
How it works

Rmd → knitr → md → pandoc →

When you run `render`, R Markdown feeds the .Rmd file to knitr, which executes all of the c

(.md) document which includes the code and it's output.

`pandoc`: "an open-source document converter" (wikipedia). Translates markup from one type of format, e.g. markdown, to another

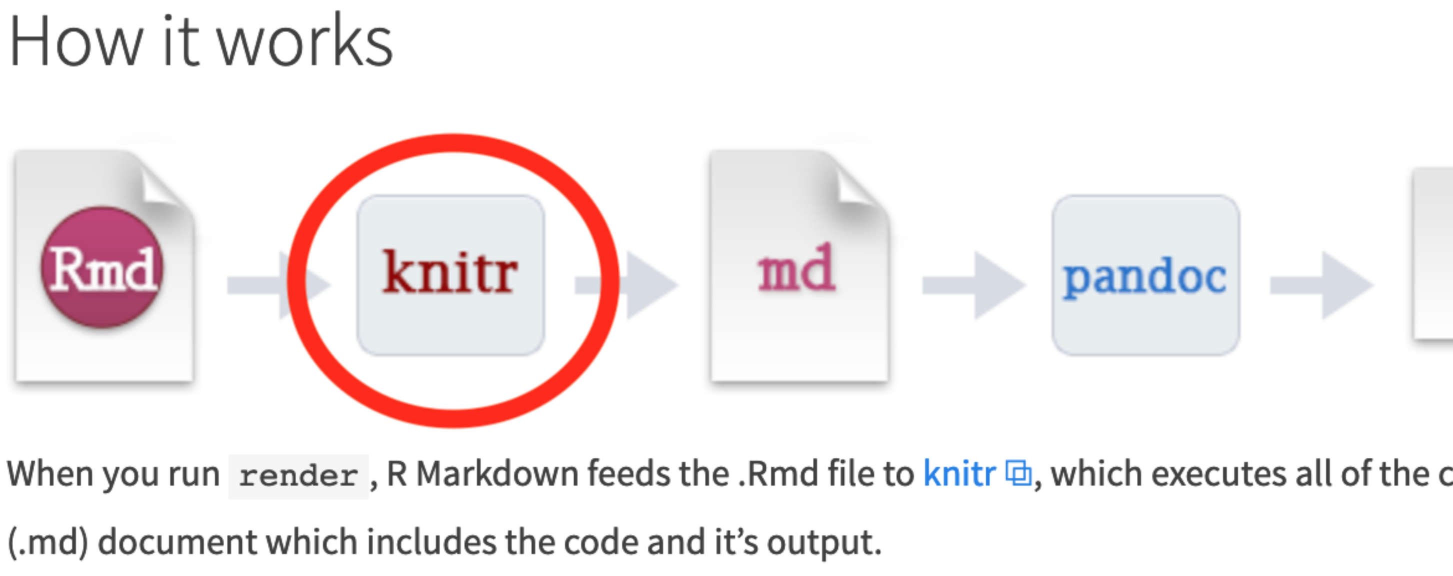


How it works

When you run `render`, R Markdown feeds the .Rmd file to knitr, which executes all of the c

(.md) document which includes the code and it's output.

`md`: a document written in markdown, "a lightweight markup language with plain text formatting syntax" (wikipedia). Github also uses markdown.

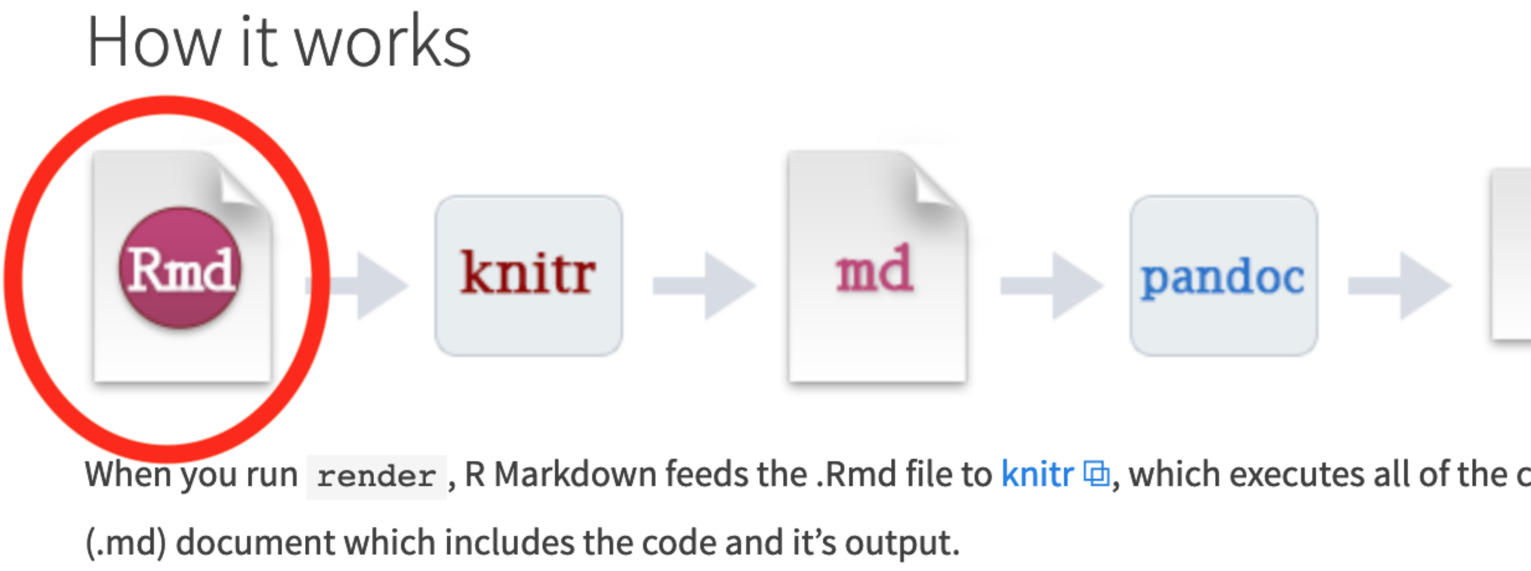


How it works

When you run `render`, R Markdown feeds the .Rmd file to knitr, which executes all of the c

(.md) document which includes the code and it's output.

`knitr`: an `R` package for creating reports directly in R. Will translate your R markdown document (`.Rmd`), including embedded `R` code, to a plain markdown document



How it works

When you run `render`, R Markdown feeds the .Rmd file to knitr, which executes all of the c

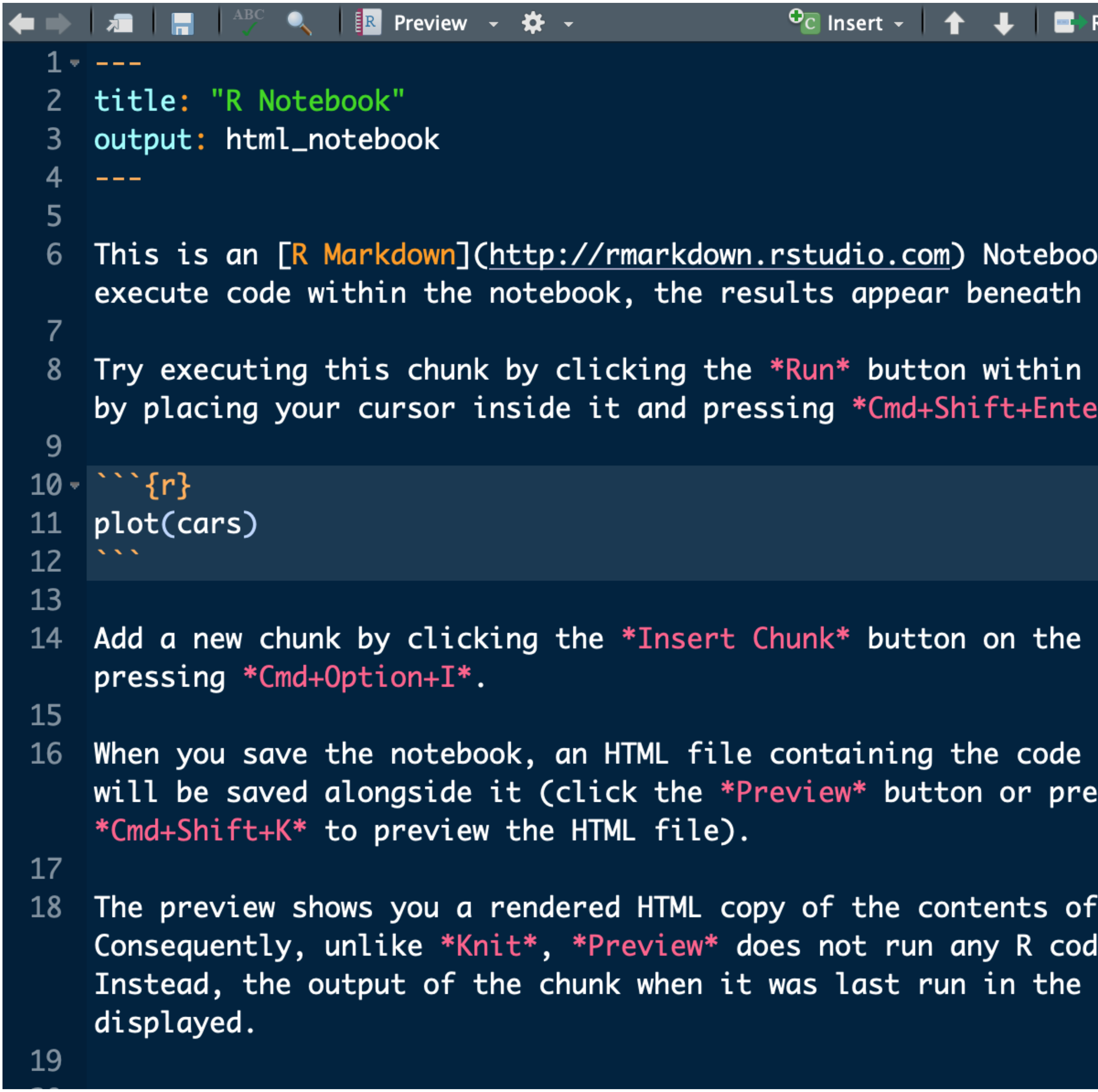
(.md) document which includes the code and it's output.

`.Rmd`: file type recognized by Rstudio. This is where everything goes: your header, `R` code chunks, and your content written in markdown

From R Studio, go to

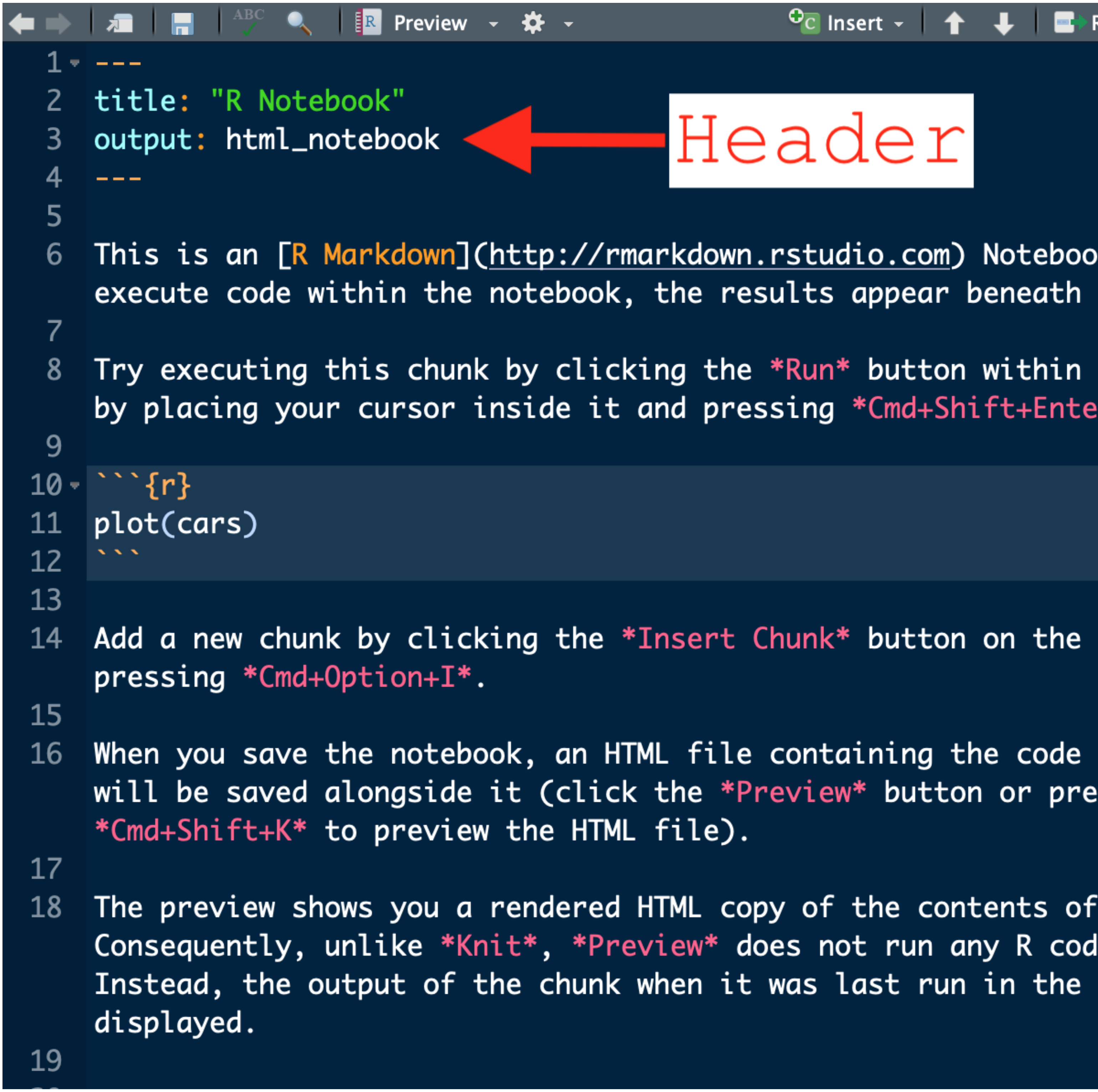
`File` > `New File` > `R Notebook`

Default R notebook



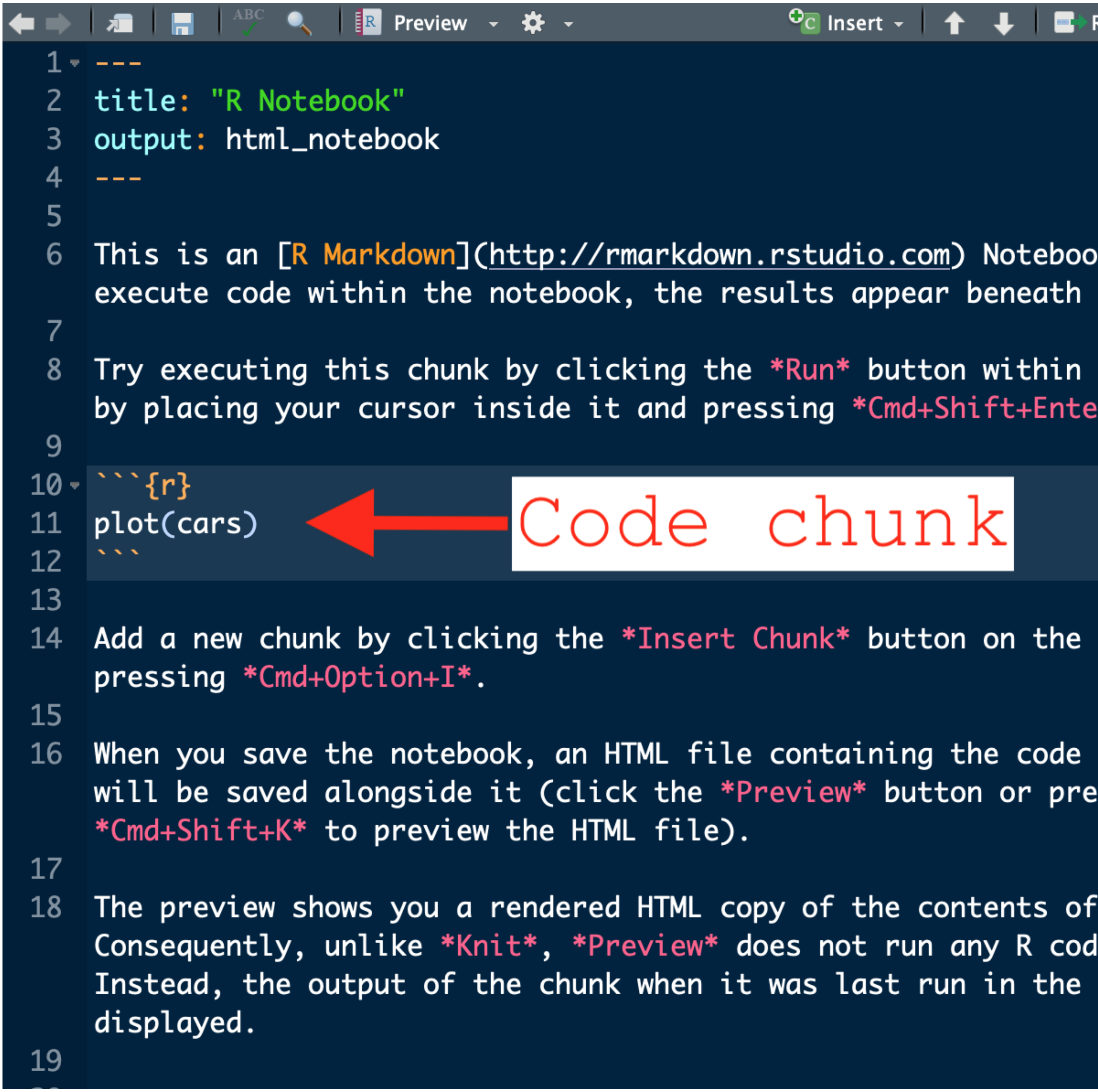
~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Noteboo
execute code within the notebook, the results appear beneath

Try executing this chunk by clicking the *Run* button within
by placing your cursor inside it and pressing *Cmd+Shift+Ente

```{r}
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the
pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code
will be saved alongside it (click the *Preview* button or pre
*Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of
Consequently, unlike *Knit*, *Preview* does not run any R cod
Instead, the output of the chunk when it was last run in the
displayed.

~~~

“YAML” Header



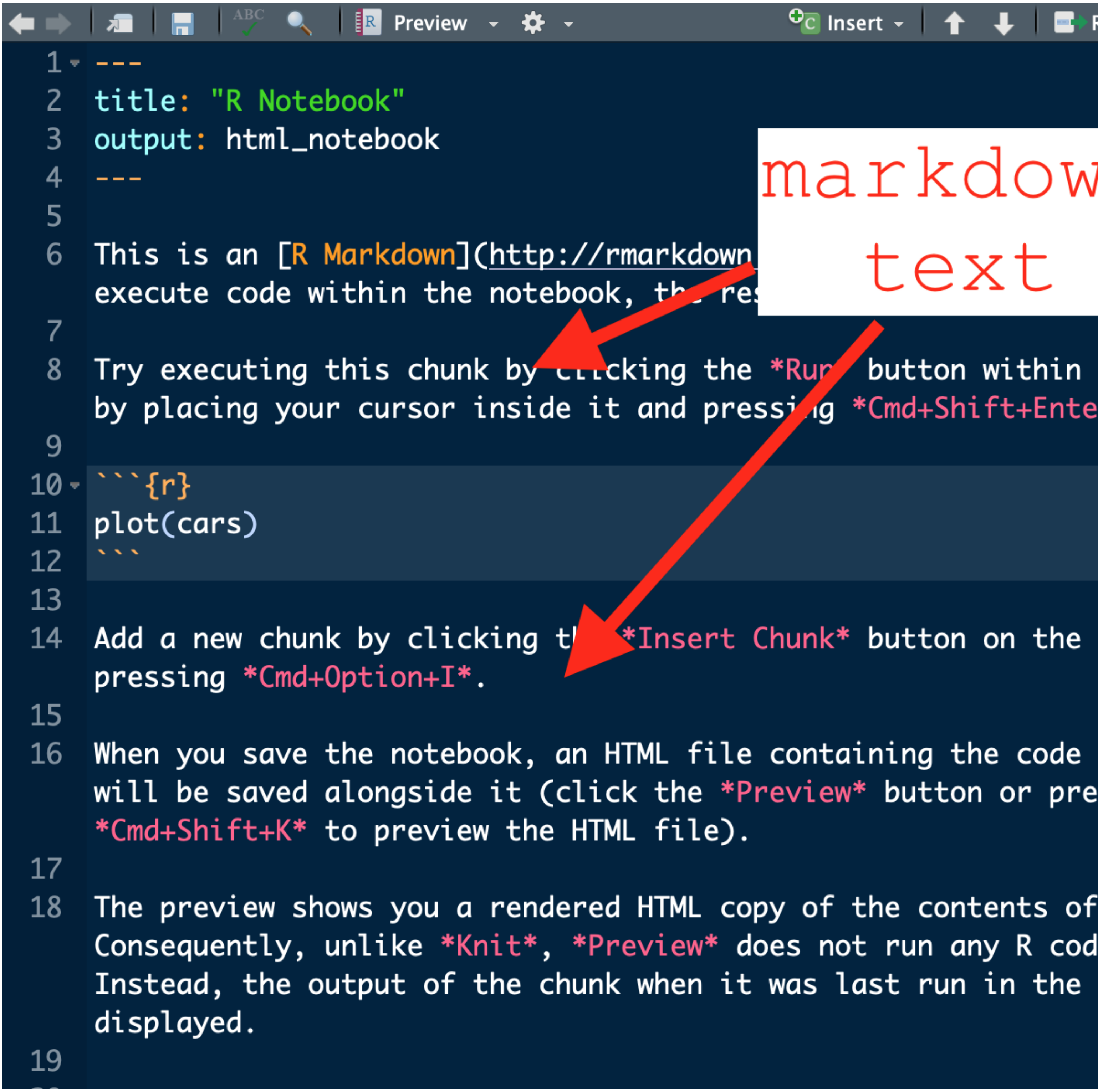
~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Noteboo
execute code within the notebook, the results appear beneath

Try executing this chunk by clicking the *Run* button within
by placing your cursor inside it and pressing *Cmd+Shift+Ente

```{r}
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the
pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code
will be saved alongside it (click the *Preview* button or pre
*Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of
Consequently, unlike *Knit*, *Preview* does not run any R cod
Instead, the output of the chunk when it was last run in the
displayed.

~~~

Text written in markdown



~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Noteboo
execute code within the notebook, the results appear beneath

Try executing this chunk by clicking the *Run* button within
by placing your cursor inside it and pressing *Cmd+Shift+Ente

```{r}
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the
pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code
will be saved alongside it (click the *Preview* button or pre
*Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of
Consequently, unlike *Knit*, *Preview* does not run any R cod
Instead, the output of the chunk when it was last run in the
displayed.

~~~

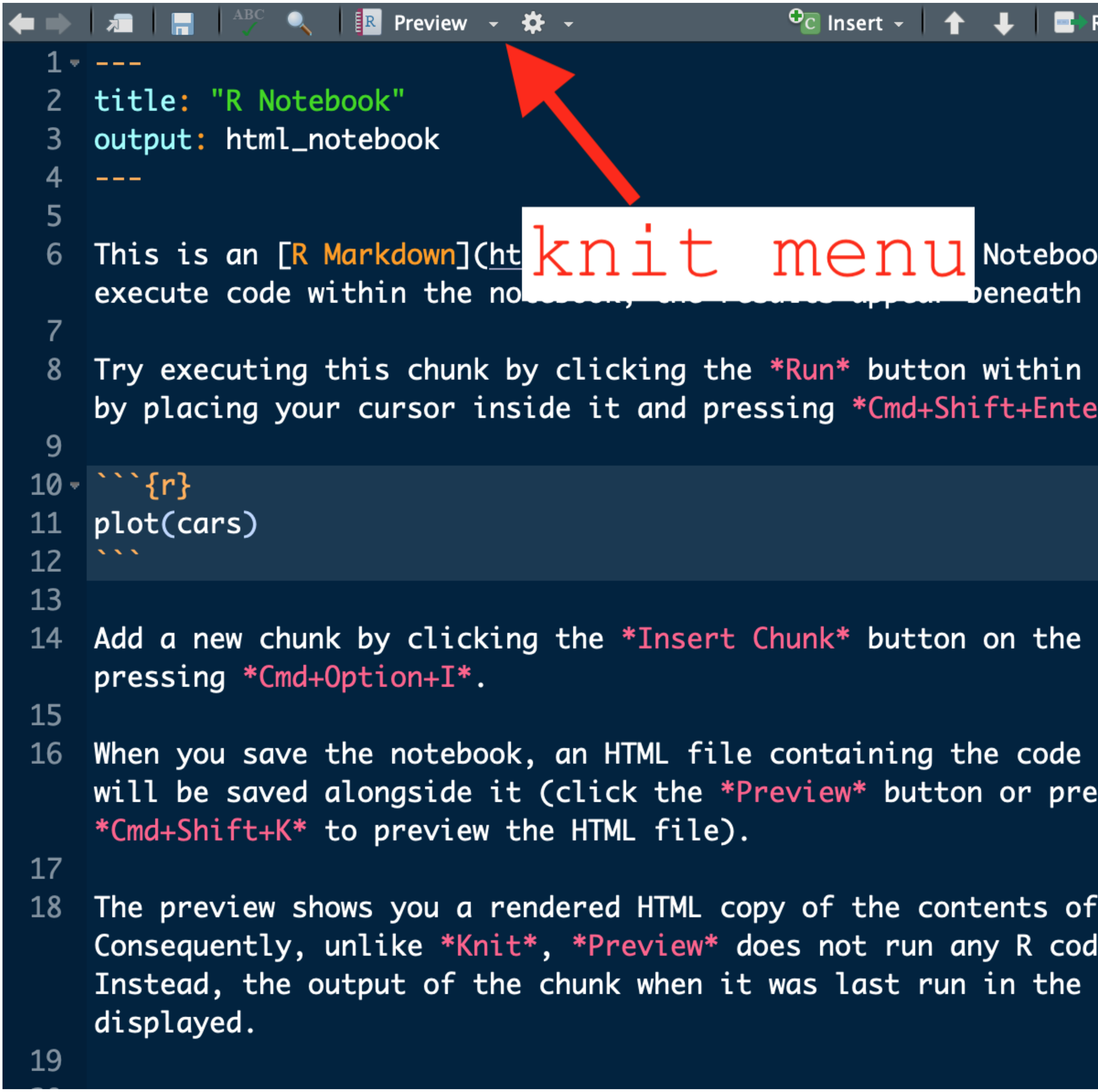
Code chunks



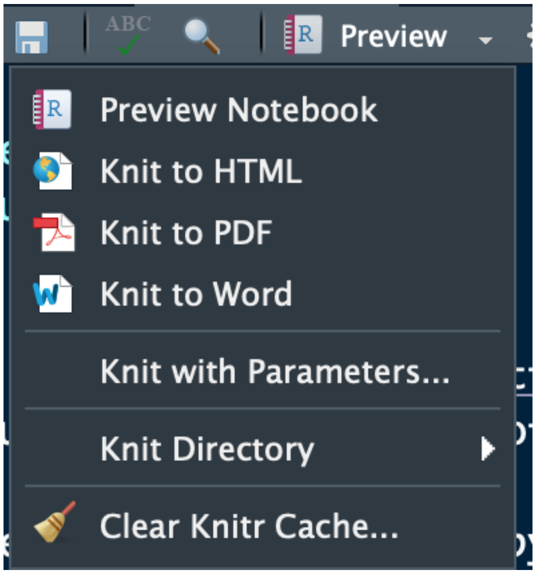
markdow
text

~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown
execute code within the notebook, the res

Try executing this chunk by clicking the *Run* button within
by placing your cursor inside it and pressing *Cmd+Shift+Ente

```{r}
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the
pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code
will be saved alongside it (click the *Preview* button or pre
*Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of
Consequently, unlike *Knit*, *Preview* does not run any R cod
Instead, the output of the chunk when it was last run in the
displayed.

~~~

Knitting your document

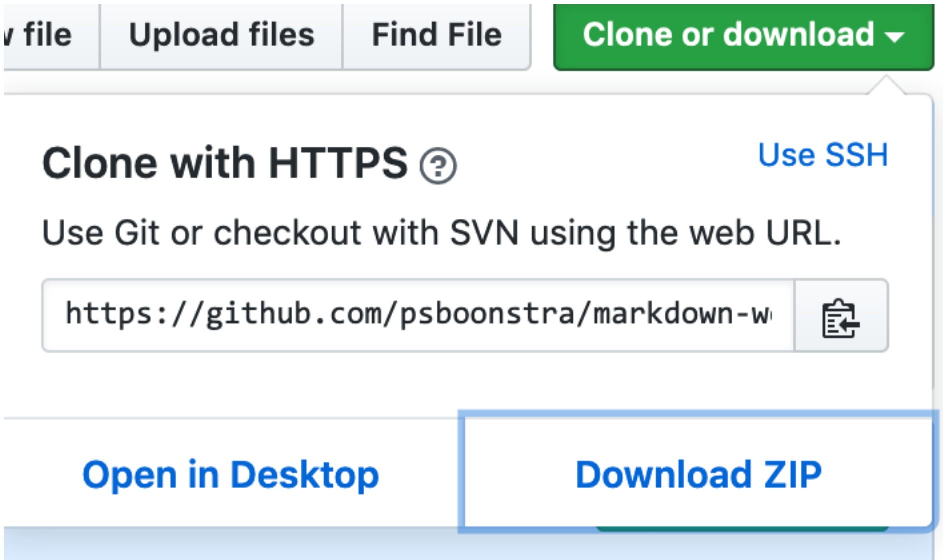


~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](ht Noteboo
execute code within the no beneath

Try executing this chunk by clicking the *Run* button within
by placing your cursor inside it and pressing *Cmd+Shift+Ente

```{r}
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the
pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code
will be saved alongside it (click the *Preview* button or pre
*Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of
Consequently, unlike *Knit*, *Preview* does not run any R cod
Instead, the output of the chunk when it was last run in the
displayed.

~~~



Try it out: Option 1

a. Download R (https://cran.r-project.org/)
b. Download RStudio to interface with R (https://www.rstudio.com/)
c. Go to https://github.com/psboonstra/umich-globalstatcore-R, then 'Clone or download', then 'Download ZIP'



Use Git or checkout with SVN using the web URL.

`https://github.com/psboonstra/markdown-w`

d. Unzip the folder, then open the .RProj file
e. In RStudio, click on 'Files' at the bottom, and pull up `01-exercise.Rmd`

Try it out: Option 2

a. Go to https://rstudio.cloud/ > Get Started
b. Create an account
c. Click the dropdown menu *next to* the New Project button, and enter the workshop URL of the workshop repository: https://github.com/psboonstra/markdown-workshop
d. Click on 'Files' at the bottom, and pull up `01-exercise.Rmd`

Your turn

08:00

Takeaways

- Chunk options control how the chunk is evaluated and used
- You can knit the same document to different formats (sometimes easy to do, sometimes requires a bit of finagling)
- Consider using in-line chunks instead of hard-coding results

Use Markdown to tell your story

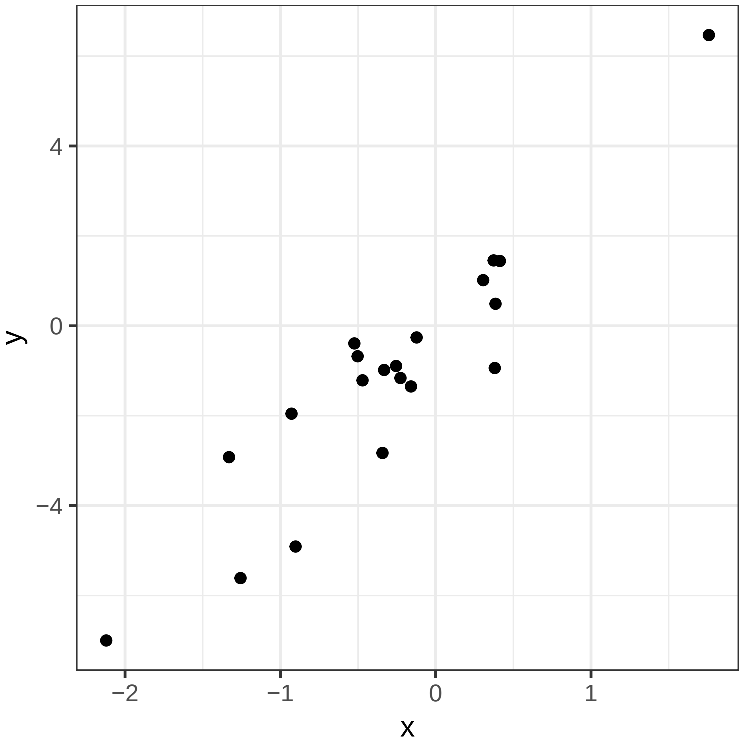
If you name a variable in an earlier code chunk, you can use it again in a later chunk.

early code chunk

```
x <- rnorm(20);
y <- 3 * x + rnorm(length(x));
foo = tibble(x = x, y = y);
```

later code chunk

```
library(ggplot2)
ggplot(data = foo) +
  geom_point(aes(x, y));
```



Tables

```
foo;
```

```
## # A tibble: 20 x 2
##         x      y
##     <dbl>  <dbl>
##  1 -0.902 -4.91
##  2  0.385  0.490
##  3 -0.524 -0.392
##  4  0.380 -0.938
##  5 -0.928 -1.95
##  6 -0.471 -1.21
##  7 -0.159 -1.35
##  8  0.413  1.44
##  9  0.306  1.02
## 10 -1.26  -5.61
## 11  0.373  1.45
## 12 -0.342 -2.83
## 13 -0.332 -0.982
## 14 -0.123 -0.259
## 15 -2.12  -7.00
## 16 -0.227 -1.16
## 17 -1.33  -2.92
## 18  1.76   6.47
## 19 -0.503 -0.677
## 20 -0.255 -0.894
```

Tables using 'kable'

x	y
-0.90217	-4.90855
0.38509	0.49021
-0.52358	-0.39152
0.38010	-0.93846
-0.92834	-1.95415
-0.47103	-1.21240
-0.15918	-1.34846
0.41299	1.44193
0.30577	1.01515
-1.25662	-5.61091
0.37305	1.45456
-0.34250	-2.82788
-0.33175	-0.98205
-0.12280	-0.25883
-2.12116	-6.99838
-0.22680	-1.16124
-1.33050	-2.92211
1.75864	6.46640
-0.50261	-0.67697
-0.25483	-0.89369

Other Markdown basics

- Use #, ##, ###, etc to indicate deeper layers of a header
- Use *, + for bulleted (unordered) lists
- Use (i), (a), or 1. for ordered lists
- Use *{text}* for *italics*, **{text}** for **bold**

Random other lessons I've learned

Markdown can be really, really finicky about horizontal and vertical spacing

If something (a new header option, a code chunk, etc) is not working as you expect, try adding an additional linebreak

If experimenting with a new feature, re-knit frequently

Caching

If, like me, you become a compulsive re-knitter, the code chunk option `cache = TRUE` is both useful and dangerous.

~~~
```{r, cache = TRUE}

(some intensive task)

```
~~~

As long as you don't change *anything* in the chunk, you won't need to re-run the intensive task upon re-knitting. However, things can go awry…

- Open the file `caching_mishap.Rmd` and make sure you understand the intended behavior (should be trivial!)
- Knit the document
- Now edit your first chunk, changing to `x <- rnorm(n = 1, mean = 100)`. *Leave the second chunk alone*
- Re-knit your document

That's how we get results like this:

```
x <- rnorm(n = 1, mean = 100);
```

```
x;
```

```
## [1] 0.47067
```

What happened

We triggered a recache of the first chunk without triggering a reache of the second

Possible solutions

- Don't split chunks if not necessary
- For chunks that may be susceptible, trigger a recache by adding a comment character (`#`) at the end of a line, or making some other innocuous change to your chunk
- Go to Knit > Clear Knitr Cache… or delete directly the folder ending in `[filename]_cache` in your working directory

`knitr` can run code in other languages

Including

- Python
- SQL
- Julia
- Stan
- Javascript

Use ` ```{python} ` to start a python code chunk, ` ```{julia} ` to start a bash code chunk, etc.

You may need external interpreters to successfully call other languages. I have not used this functionality before.

see Chapter 2.7, R Markdown: The Definitive Guide

You can knit `R` scripts!

You are not limited to using Markdown in `Rmd` files – you can knit `R` scripts using the same shortcut: *Cmd+Shift+K* / *Ctrl+Shift+K*

- Use `#'` to indicate a switch to markdown
- Use `#+` to indicate a new chunk

Your turn again

Open `02-exercise.R` and complete the 8 tasks. Indicate when you are done.

05:00

What to do next

https://rmarkdown.rstudio.com/

R Markdown: The definitive guide

- Free, online version of a book written by the `R`studio experts

R Markdown cheatsheet

- Helpful quick reference

Mastering markdown

- Reference site for markdown

References