

R Markdown Workshop

R Markdown Basics with rmarkdown & knitr



Presented by Emi Tanaka

School of Mathematics and Statistics



THE UNIVERSITY OF
SYDNEY



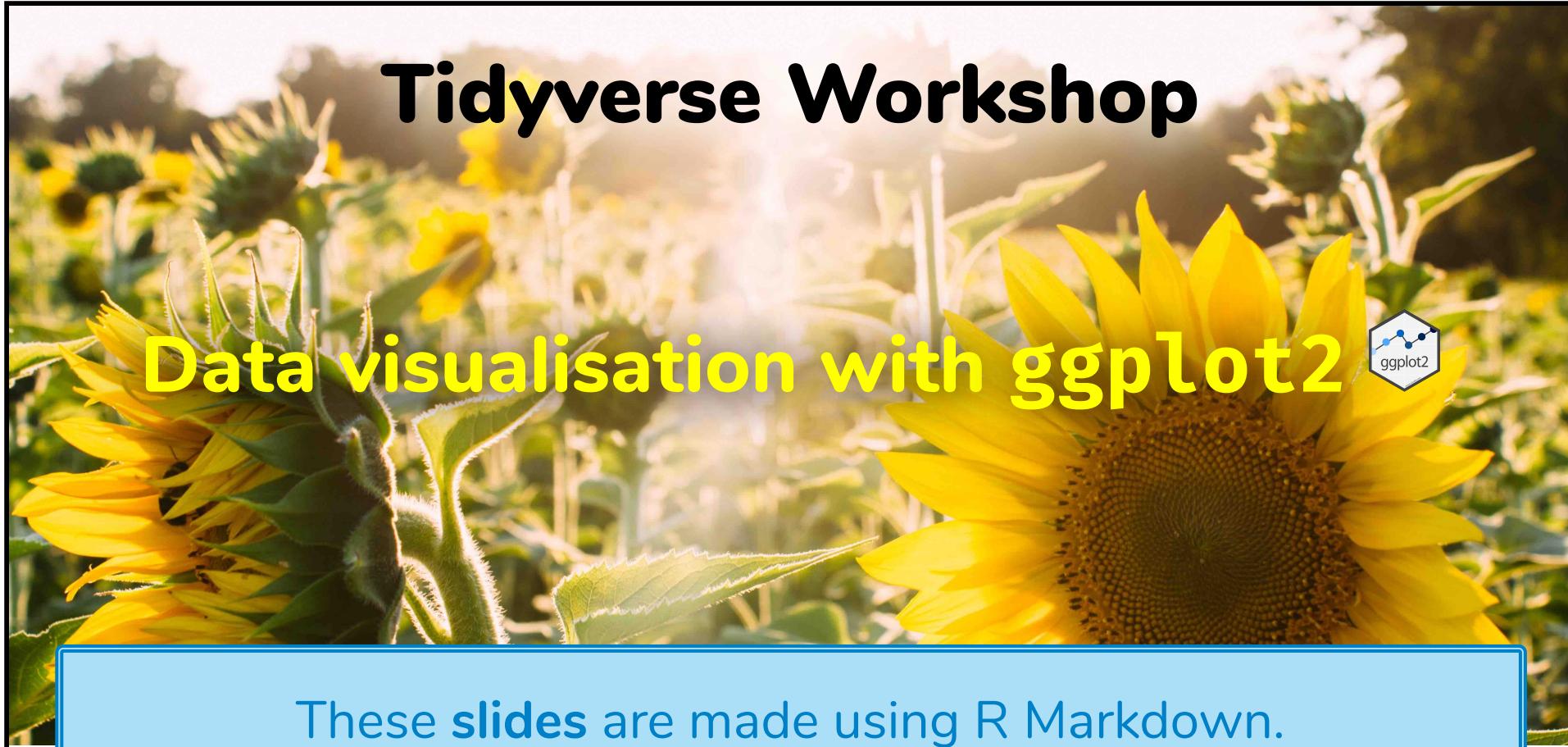
dr.emi.tanaka@gmail.com



@statsgen

1st Dec 2019 @ Biometrics by the Botanic Gardens | Adelaide, Australia

What you can do with R Markdown?



Tidyverse Workshop

Data visualisation with ggplot2

The ggplot2 logo is located in the top right corner of the slide.

These slides are made using R Markdown.

What you can do with R Markdown?

R Markdown Workshop

Reproducible Reports

Emi Tanaka

19th November 2019

The following are texts from Xie (2013) Dynamic Graphics and Reporting for Statistics. *PhD Thesis*.

Dynamic reporting

Statistics as a scientific discipline has a dynamic nature, which can be observed in many statistical algorithms and theories as well as in data analysis. For example, asymptotic theories in statistics are inherently dynamic: they describe how a statistic or an estimator behaves as the sample size increases. Data analysis is almost never a static process. Instead, it is an iterative process involving cleaning, describing, modeling, and re-cleaning the data. Reports may end up being re-written due to changes in the data and analysis.

A
e

This **dynamic report** is made using R Markdown.

What you can do with R Markdown?

This manuscript is made using R Markdown.

What you can do with R Markdown?

Forecasting: Principles and Practice

Rob J Hyndman and George Athanasopoulos

Monash University, Australia

Preface

W
te
T

This **online book** is made using R Markdown.

Available at <https://otexts.com/fpp2/>

The **pdf book** version is also made using R Markdown.

What you can do with R Markdown?

Tidy tools for
supporting
fluent workflow
in temporal
data analysis

Earo Wang

This thesis (online and pdf) is made using R Markdown.
Available at <https://thesis.earo.me/>



MONASH University

**Tidy tools for supporting fluent
workflow in temporal data
analysis**

Yiru (Earo) Wang
B.Comm. (Hons), Monash University

What you can do with R Markdown?

Scientific and Technical Blogging: Radix vs. Blogdown

Individuals can use R Markdown documents for blogging to quickly share their knowledge but what framework should you be using?

AUTHOR

Emi Tanaka

AFFILIATION

School of Mathematics and Statistics,
The University of Sydney

PUBLISHED

Dec. 14, 2018

≡ HOME POSTS ABOUT

Scientific and Technical Blogging: Radix vs Blogdown (Remix)

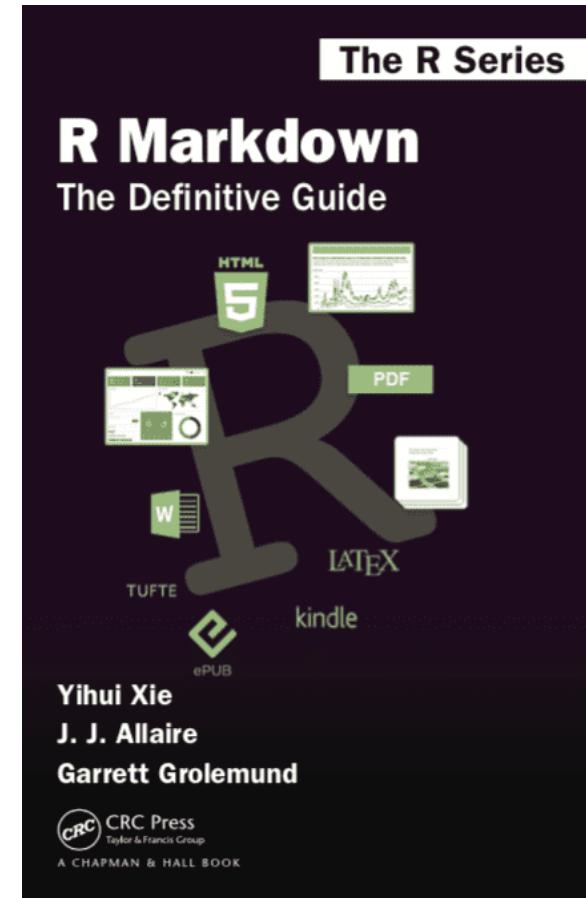
⌚ 2018, Sep 20 ⓘ 8 mins read

These blog posts are made using R Markdown.

What you can do with R Markdown?

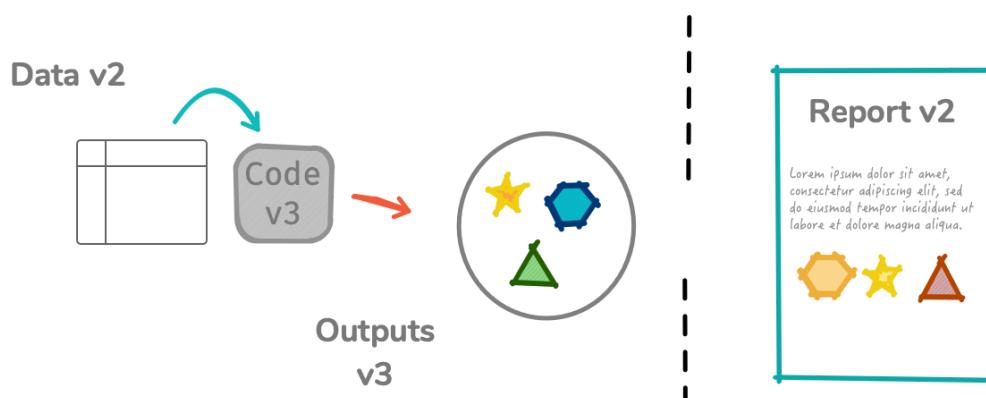
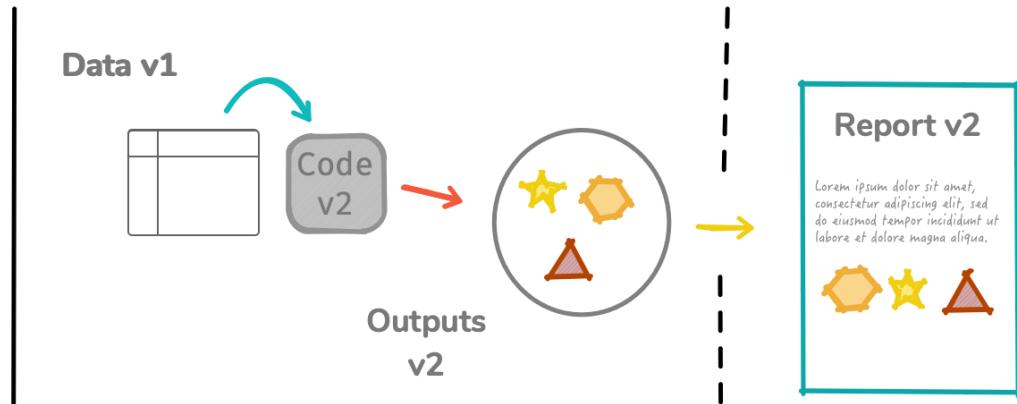
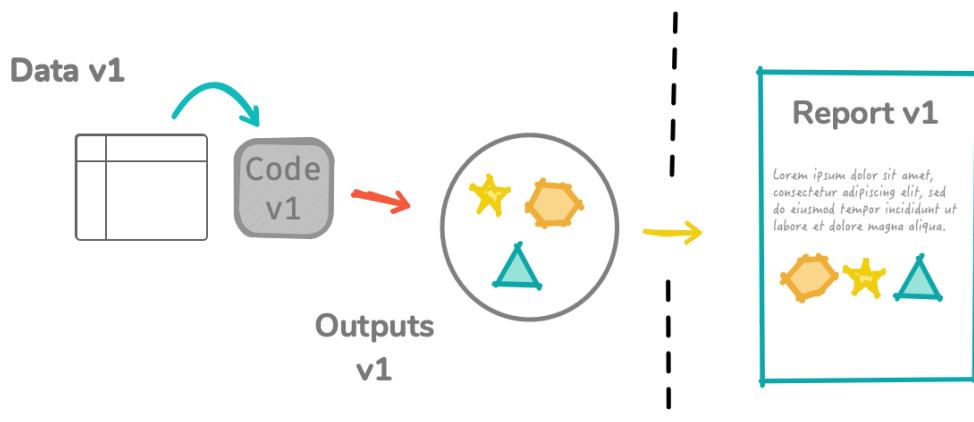
Possibilities are endless...

- Microsoft Word document (.doc, .docx)
- Open Document Text (.odt)
- Rich text format (.rtf)
- Power point slides (.ppt, .pptx)
- Posters (as .html or .pdf)
- Resume (as .html or .pdf)
- Interactive R Notebooks (.html)
- Markdown documents (.md)
- Dashboard (.html)



Check out Xie et al. (2018) R Markdown: The Definite Guide.

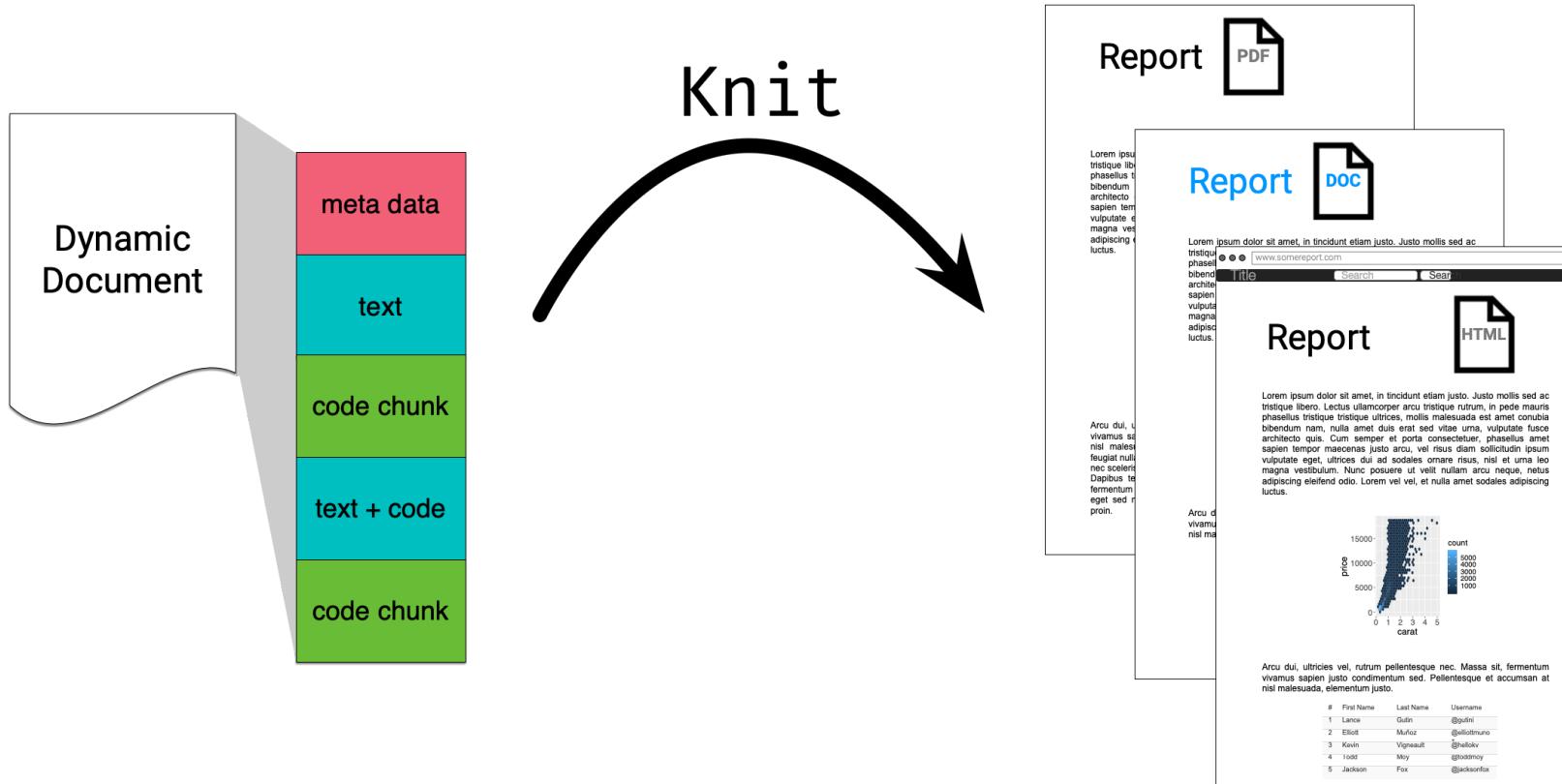
Before R Markdown (and Sweave)



R Markdown in a nutshell



R Markdown integrates **text + code** in one source document with ability to knit *to many output formats* (via Pandoc).

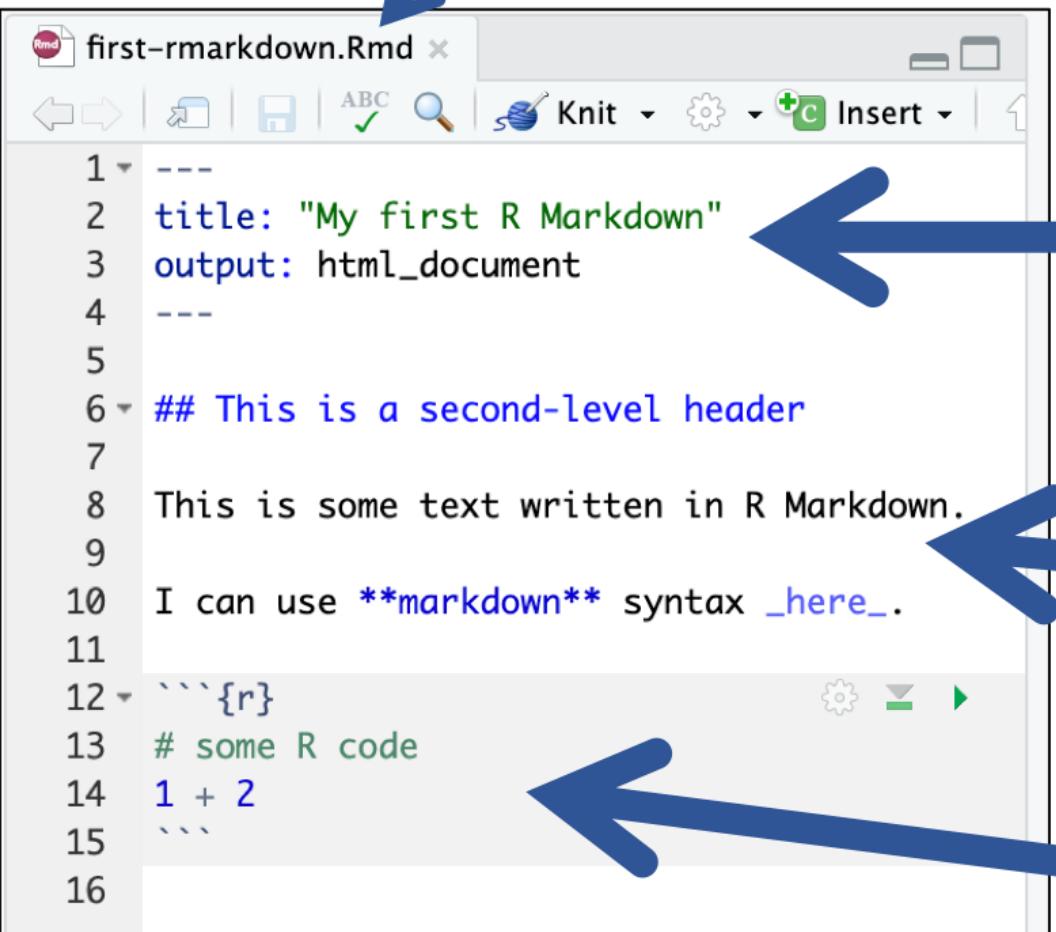




Open and inspect the file
`first-rmarkdown.Rmd`

R Markdown Basics

File extension is Rmd



```
first-rmarkdown.Rmd x
[File] [New] [Open] [Save] [ABC] [Knit] [Insert]
1 ---  
2 title: "My first R Markdown"  
3 output: html_document  
4 ---  
5  
6 ## This is a second-level header  
7  
8 This is some text written in R Markdown.  
9  
10 I can use **markdown** syntax _here_.  
11  
12 ```{r}  
13 # some R code  
14 1 + 2  
15```
```

YAML: meta
data fenced by

Text can use
markdown

Code chunk:
fenced by ```

Knitting: Rmd → md → html



```
first-rmarkdown.Rmd x
Rmd ABC Knit Insert 1

1 ---
2 title: "My first R Markdown"
3 output: html_document
4 ---
5
6 ## This is a second-level header
7
8 This is some text written in R Markdown.
9
10 I can use **markdown** syntax here.
11
12 `r
13 # some R code
14 1 + 2
15 `r
16
```

Output file name matches
Rmd file name

first-rmarkdown.html

My first R Markdown

This is a second-level header

This is some text written in R Markdown.

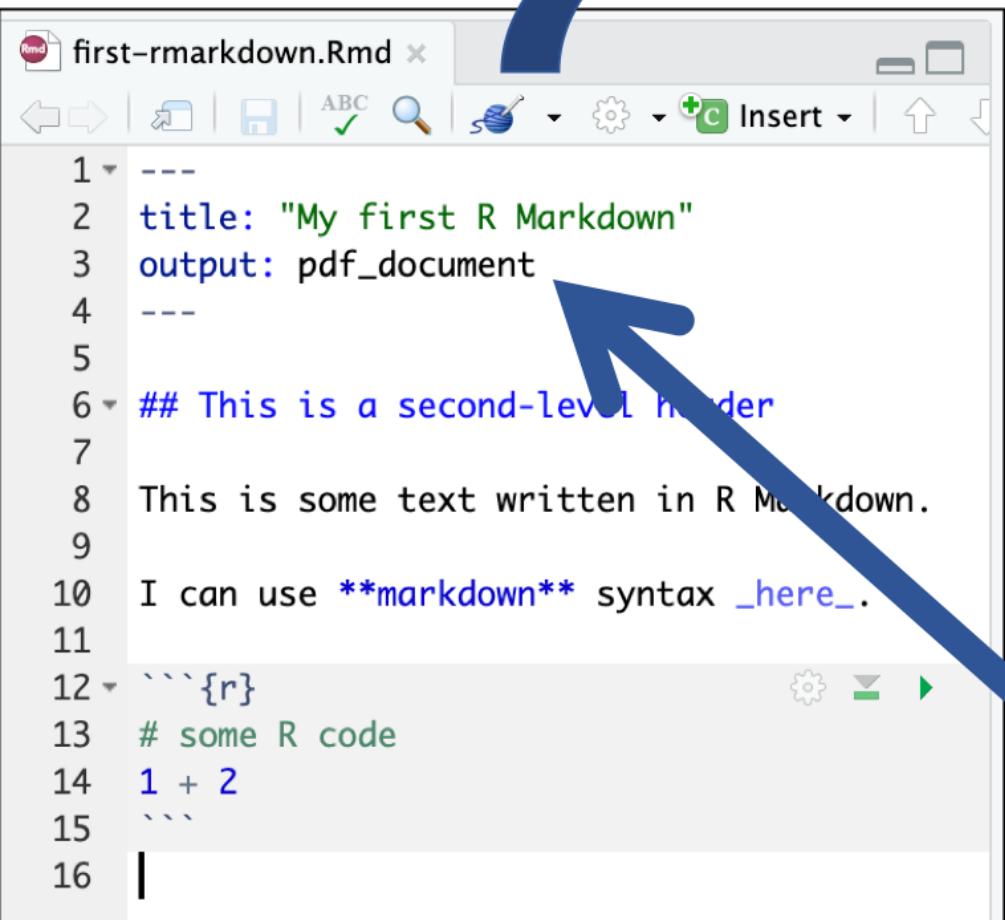
I can use **markdown** syntax *here*.

```
# some R code
```

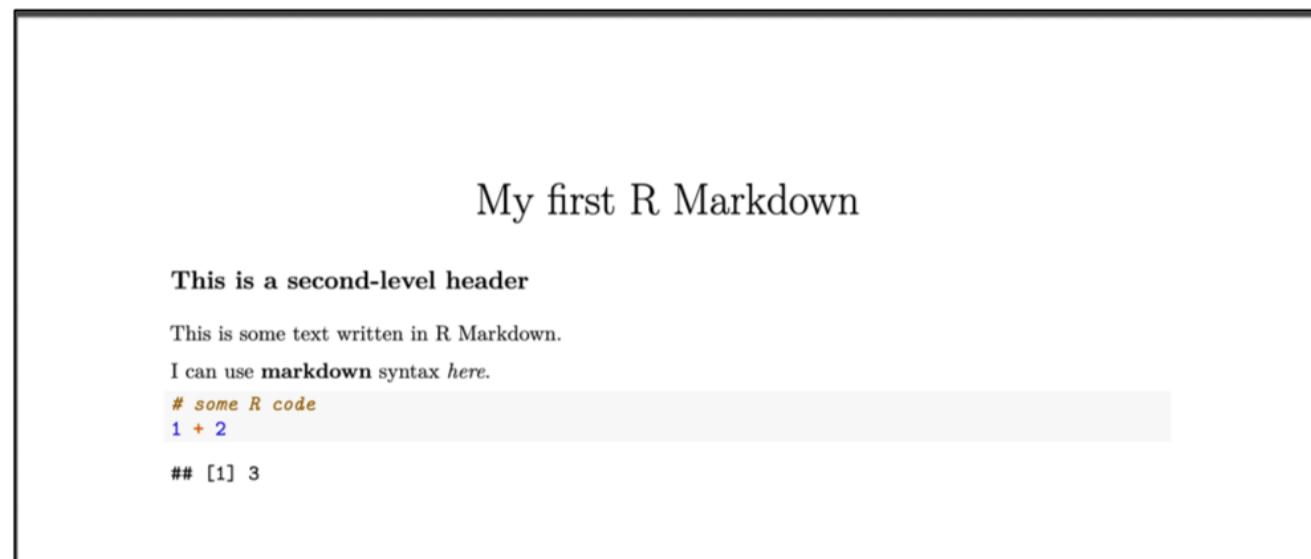
```
1 + 2
```

```
## [1] 3
```

Knitting: Rmd → md → pdf



```
1 ---  
2 title: "My first R Markdown"  
3 output: pdf_document  
4 ---  
5  
6 ## This is a second-level header  
7  
8 This is some text written in R Markdown.  
9  
10 I can use **markdown** syntax _here_.  
11  
12 ```{r}  
13 # some R code  
14 1 + 2  
15 ...  
16
```



My first R Markdown

This is a second-level header

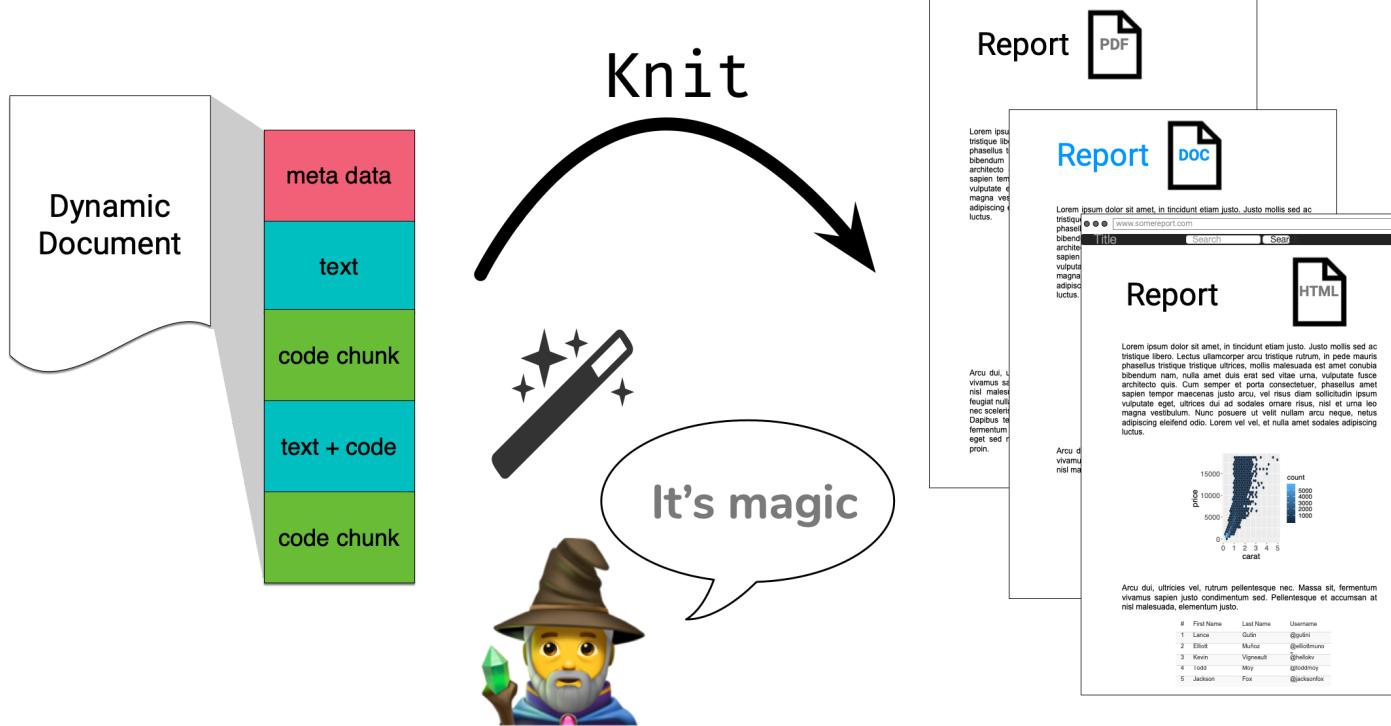
This is some text written in R Markdown.

I can use markdown syntax *here*.

```
# some R code  
1 + 2  
## [1] 3
```

pdf output

How does it all work?



R Markdown under the hood: html_document

knitr 📦 does the hard work of converting Rmd 📄 → md 📄

---	Rmd 📄	---	md 📄
	title: "My Report"		title: "My Report"
	output: html_document		output: html_document
---		---	
	# Hello		# Hello
	```{r}		``` r
	1 + 1		1 + 1
	```		```
			...
			...
			## [1] 2
			```

# R Markdown under the hood: html_document

Then `rmarkdown` 📦 prepares md 📄 for `pandoc` to convert to html 📄.

```
--- md 📄 html 📄
title: "My Report"
output: html_document

Hello
```r
1 + 1
```
```
## [1] 2
```
<body>
<div class="container-fluid main-container">
<div class="fluid-row" id="header">
<h1 class="title toc-ignore">My Report</h1>
</div>
<div id="hello" class="section level1">
<h1>Hello</h1>
<pre class="r"><code>1 + 1</code></pre>
<pre><code>## [1] 2</code></pre>
</div>
</div>
</body>
```

⚠ This html file is missing  
some elements due to  
restriction of space.

# R Markdown under the hood: pdf_document

Same as before: knitr 📦 converts Rmd 📄 → md 📄

---	Rmd 📄	---	md 📄
	title: "My Report"		title: "My Report"
	output: pdf_document		output: pdf_document
---		---	
# Hello		# Hello	
```{r}		``` r	
1 + 1		1 + 1	
```		```	
			## [1] 2
			```

R Markdown under the hood: pdf_document

Then rmarkdown 📦 prepares md 📄 for pandoc to convert to tex 📄 → pdf 📄.

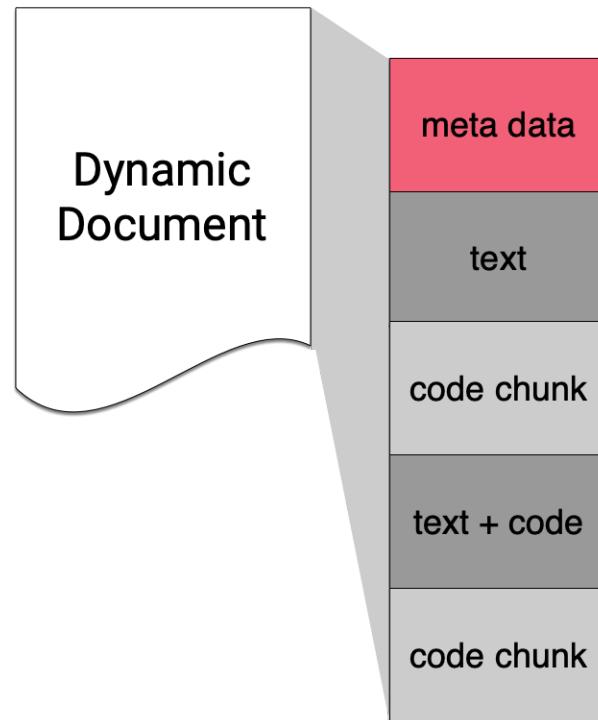
```
---      md   
title: "My Report"  
output: html_document  
---  
  
# Hello  
```r  
1 + 1
...
...
[1] 2
...
[1] 2
...

\begin{document} tex 
\maketitle
\hypertarget{hello}{%
\section{Hello}\label{hello}}
\begin{Shaded}
\begin{Highlighting}[]
\DecValTok{1} \OperatorTok{+}\StringTok{} \DecValTok{1}
\end{Highlighting}
\end{Shaded}
\begin{verbatim}
[1] 2
\end{verbatim}
\end{document}
```

 This tex document is missing elements before `\begin{document}` due to limitations of

**⚠ This tex document is missing elements before \begin{document} due to restriction of space.**

# Meta data with YAML



# YAML - YAML Ain't Markup Language

Basic format

```

```

```
key: value
```

```

```

Example

```

```

```
title: "R Markdown Workshop"
subtitle: "The Basics"
author: "Emi Tanaka"
date: "`r Sys.Date()`"
output: html_document
```

```

```

There must be a space after ":"!

# Default (minimal) html output

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta name="author" content="Emi Tanaka" />
<meta name="date" content="2019-11-19" />
<title>R Markdown Workshop</title>
</head>

<body>
<h1 class="title toc-ignore">R Markdown Workshop</h1>
<h3 class="subtitle">The Basics</h3>
<h4 class="author">Emi Tanaka</h4>
<h4 class="date">2019-11-19</h4>
</body>
</html>
```



html meta data

output

## R Markdown Workshop

The Basics

Emi Tanaka

2019-11-19



Default html template add special  
YAML key values to file automatically

# YAML structure

---

- White spaces indicate structure in YAML - don't use tabs though!
- Same as R, you can comment lines by starting with #.
- YAML is case sensitive.
- A key can hold multiple values.

```
key:
 - value1
 - value2
```

```
key: [value1, value2]
```

# YAML with multiple key values

```

```

```
title: "R Package & R Markdown Workshop"
author:
 - "Damjan Vukcevic"
 - "Emi Tanaka"
output: html_document

```

output

R Package & R Markdown Workshop

Damjan Vukcevic

Emi Tanaka

output

html raw file

```
<body>
<h1 class="title toc-ignore">R Package & R Markdown Workshop</h1>
<h4 class="author">Damjan Vukcevic</h4>
<h4 class="author">Emi Tanaka</h4>
</body>
```

# Values spanning multiple lines

```

title: >
 this is a
 single line
```

```
abstract: |
 this value spans
 many lines and
 appears as it is
```

```
output: pdf_document
```

```

```

output

this is a single line

Abstract

this value spans  
many lines and  
appears as it is

Note that there are **two spaces** after each line in the above YAML values. This is necessary to create a new line for markdown.



Open and inspect the file  
second-rmarkdown.Rmd

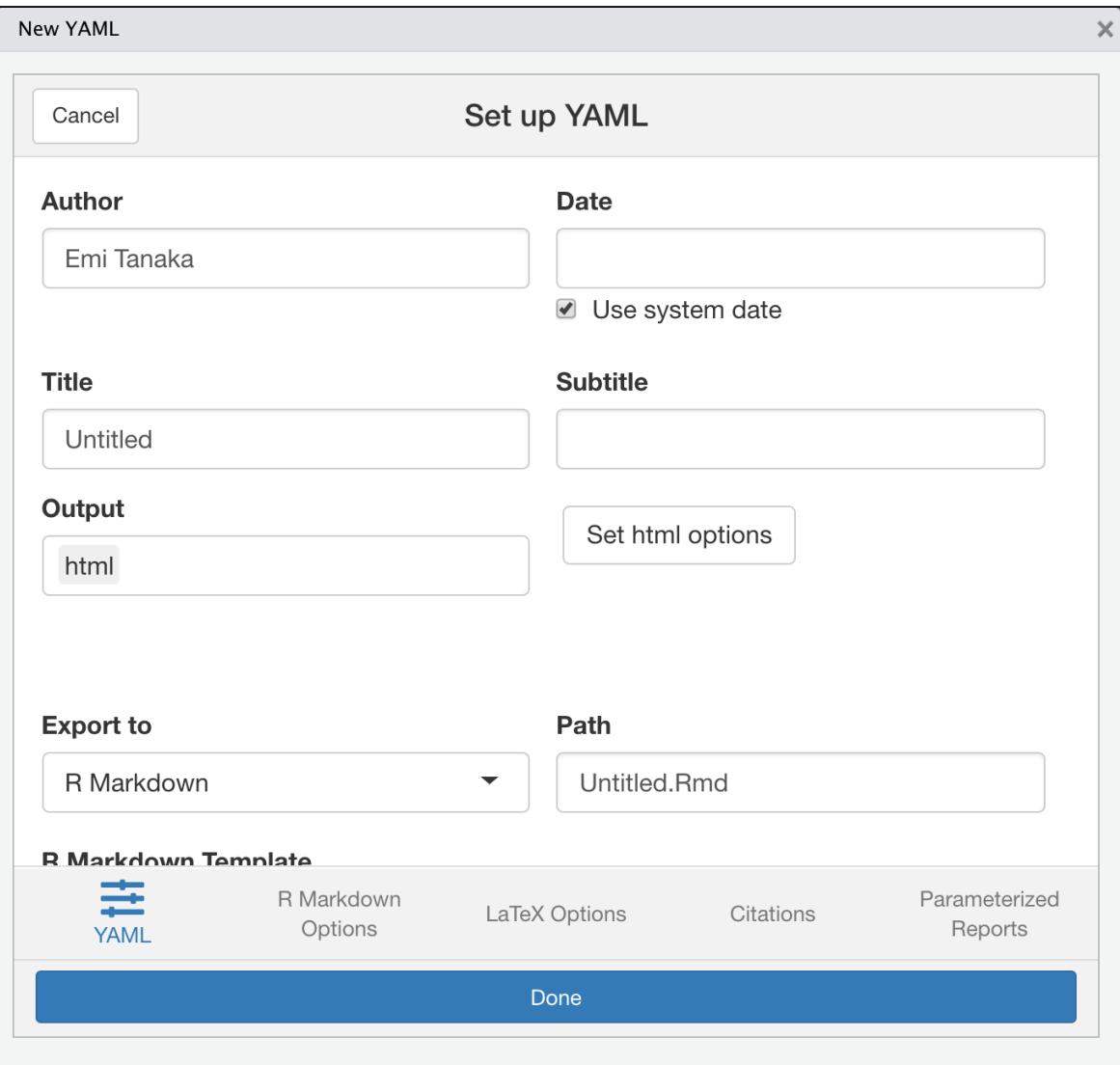
# key can contain keys

---

```
output:
 html_document:
 toc: true
 toc_float: true
 code_download: true
 code_folding: "hide"
```

---

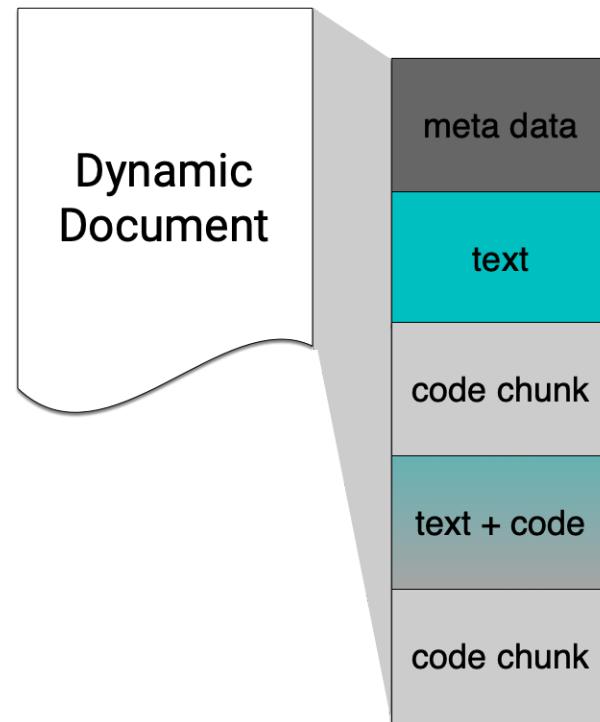
What does each of the above keys do?



- If you installed `ymlthis` then go to RStudio > Addins > YMLTHIS
- It will launch a shiny app with various YAML options
- See also helper functions at <https://ymlthis.r-lib.org/>

 Open and inspect the file  
challenge-05-yaml.Rmd

# Text



# Text in Markdown: Headers

text

```
Header 1
Header 2
Header 3
Header 4
Header 5
Header 6
```

output

**Header 1**  
**Header 2**  
**Header 3**  
**Header 4**  
**Header 5**  
**Header 6**

# Text in Markdown: Formatting

text

**\*\*This text is bold\*\***

**\_\_This text is also bold\_\_**

*\*This text is italic\**

*\_This text is also italic\_*

**\*\*\_You can combine both\_\*\***

output

This text is bold

This text is also bold

*This text is italic*

*This text is also italic*

**You can combine both**

# Text in Markdown: Lists

text

## 1. Breakfast

- \* Cereal

- \* Milk

## 1. Lunch

- \* Pizza

## \* Tissue box

- \* Soft

- \* Hard

## \* Soy sauce

output

## 1. Breakfast

- Cereal

- Milk

## 2. Lunch

- Pizza

## • Tissue box

- Soft

- Hard

## • Soy sauce

# Markdown: lightweight markup language

*Focus on writing - add flair later*

markdown

```
Section
```

Bla bla.

Then **bold**.

html

```
<h1>Section</h1>
```

Bla bla.

Then **bold**.

latex

```
\section{Section}
```

Bla bla.

Then \textbf{bold}.

docx

## Section

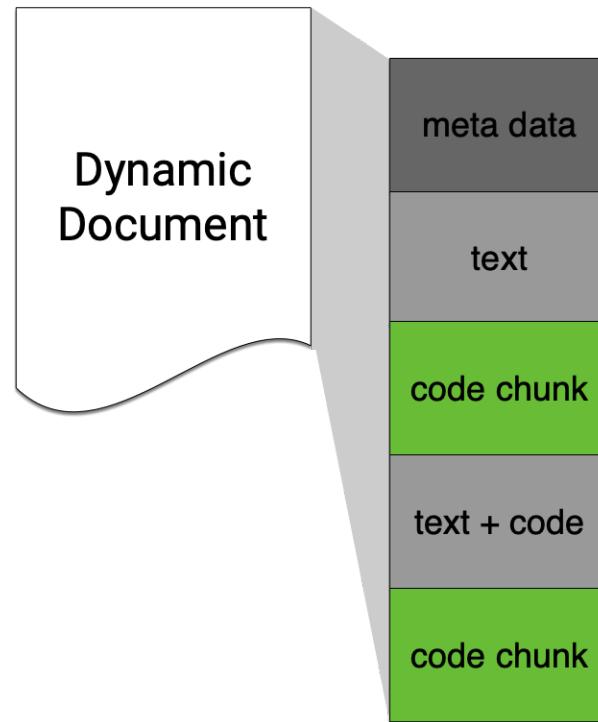
Bla bla.

Then bold.



Open and go through:  
challenge-06-table.Rmd

# Code chunk



# Keyboard shortcut for inserting code chunk

In RStudio .Rmd  press

- Mac:  +  + 
- PC:  +  + 

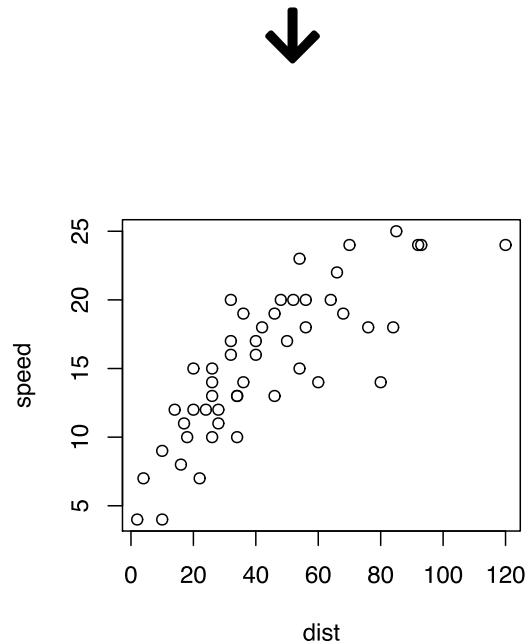
to insert a chunk of R code

```
```{r}
```

```
```
```

# Chunk options: echo & eval

```
```{r, echo = FALSE}  
plot(speed ~ dist, cars)  
```
```



```
```{r, eval = FALSE}  
plot(speed ~ dist, cars)  
```
```

↓

```
plot(speed ~ dist, cars)
```

# Valid chunk options

- Chunk options must be written in **one line**, i.e. no line break.
- All option values must be **valid R expressions**. Exception is the chunk name. E.g.
  - `fig.path = figures/` is not valid but  
`fig.path = "figures/"` is valid
  - `eval = true` is not valid but  
`eval = runif(1) > 0.5` is valid

# Chunk names (or labels)

The chunk below is called `plot1`.

```
```{r plot1}
ggplot(cars, aes(dist, speed)) + geom_poin-
````
```

All chunks have a label regardless of whether it is explicitly supplied or not.

**⚠** Do not include spaces, `"_"` or punctuation marks in your chunk name!



## Pick your challenge:

challenge-07-include.Rmd

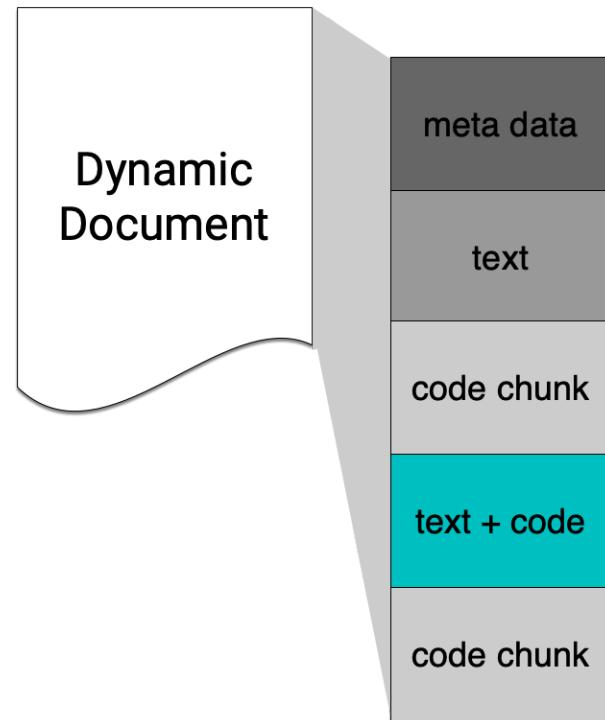
challenge-08-global-option.Rmd

challenge-09-figures.Rmd

challenge-10-cache.Rmd

challenge-11-names.Rmd

# Inline code



# Inline R Commands

Text can contain inline R commands where output is evaluated but code is not echoed.

Text

Updated last: `r Sys.Date()`

Output

Updated last: 2019-12-03

Text

The value of `\pi` is `r pi`.

Output

The value of  $\pi$  is 3.1415927.

# Accessing YAML meta data inline

All YAML meta data are stored in `rmarkdown::metadata` as a list.

---

```
title: "R Markdown Workshop"
output: html_document
```

---

```
rmarkdown::metadata$title
[1] "R Markdown Workshop"
```

All key values of `params` is stored as a list in `params`.

---

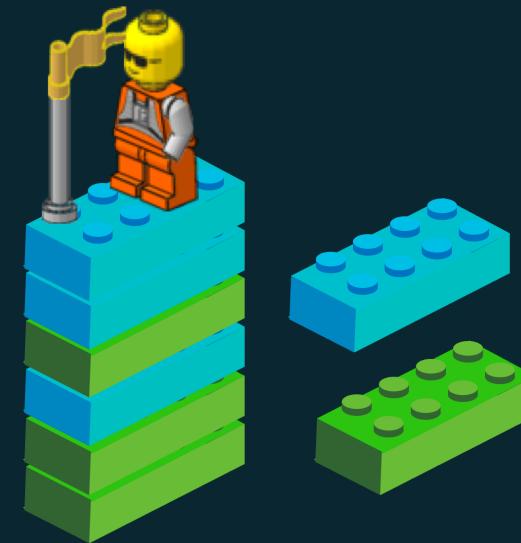
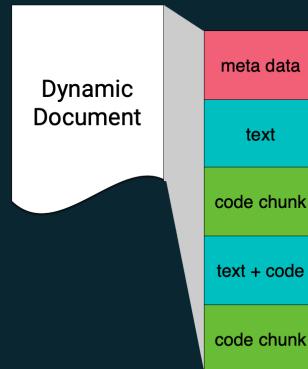
```
params:
 data: datav1.csv
 output: pdf_document
```

---

```
params$data
```

```
[1] "datav1.csv"
```

# Stack your Rmd components as you want but only one YAML piece and it needs to be on the top!



For workflow, you'd probably want to keep your tower's height manageable.

# R Markdown is not just for R

```
```{python, echo = FALSE}
a = [1, 2, 3]
a[0]
```
```

```
```{bash, echo = FALSE}
date +%B
```
```

1



December

...



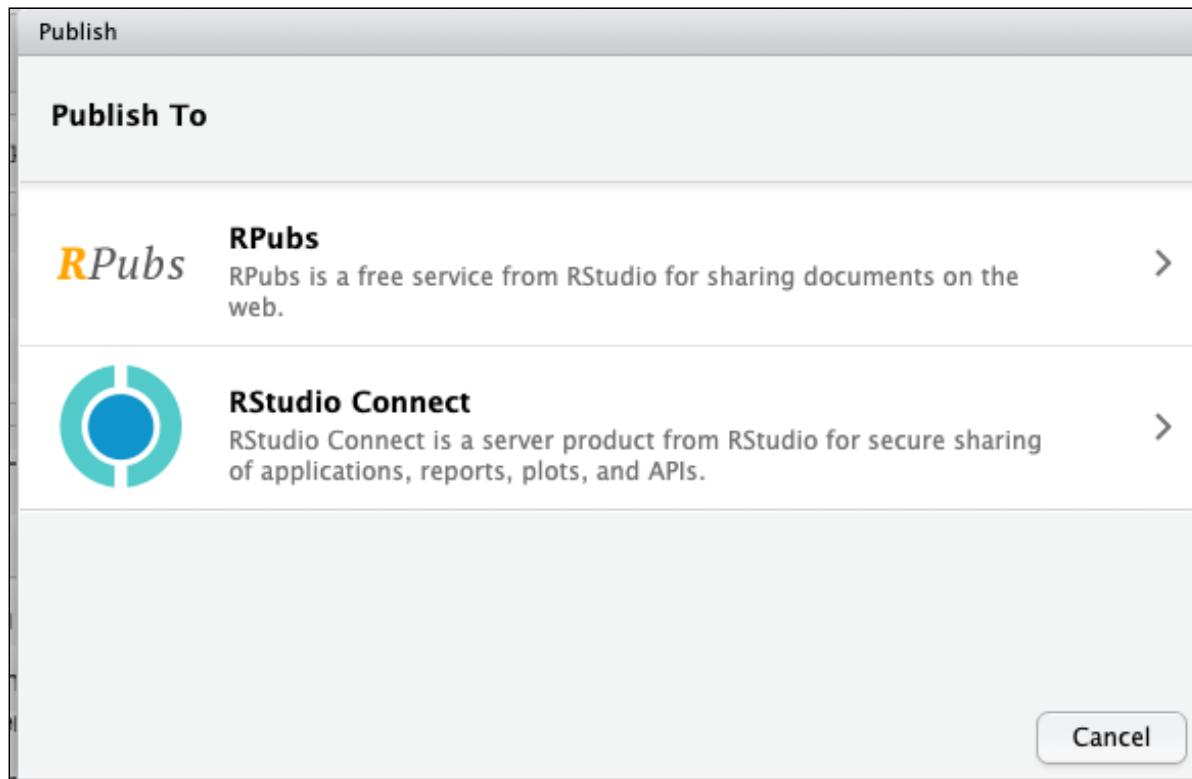
# Sharing your html documents

# RPubs

After knitting, find this Publish button:



and select RPubs if there is a choice:



# Netlify: without Git

- Rename your output html file as `index.html`
- Go to <https://app.netlify.com/>
- If you are logged in, go to the bottom of Sites to find

Want to deploy a new site without connecting to Git?  
Drag and drop your site folder here

- Drag and drop the folder that contains the `index.html`
-  it's alive!
- You probably want to change the site name - to do this go to site > site settings > change site name.

# Netlify: with Git

- Rename your output html file as `index.html`
- Git commit your folder (or html file alone if no dependency)
- Host it on GitHub (my preference), GitLab or Bitbucket
- Go to <https://app.netlify.com/> and create a new site from the repo
- 🎉
- Again, you probably want to change the site name - to do this go to site > site settings > change site name.

Happy sharing!  
That's it for the basics!

# Session Information

---

```
devtools::session_info()
```

– Session info

---

setting value

version R version 3.6.0 (2019-04-26)

os macOS Mojave 10.14.6

system x86\_64, darwin15.6.0

ui X11

language (EN)

collate en\_AU.UTF-8

These slides are licensed under