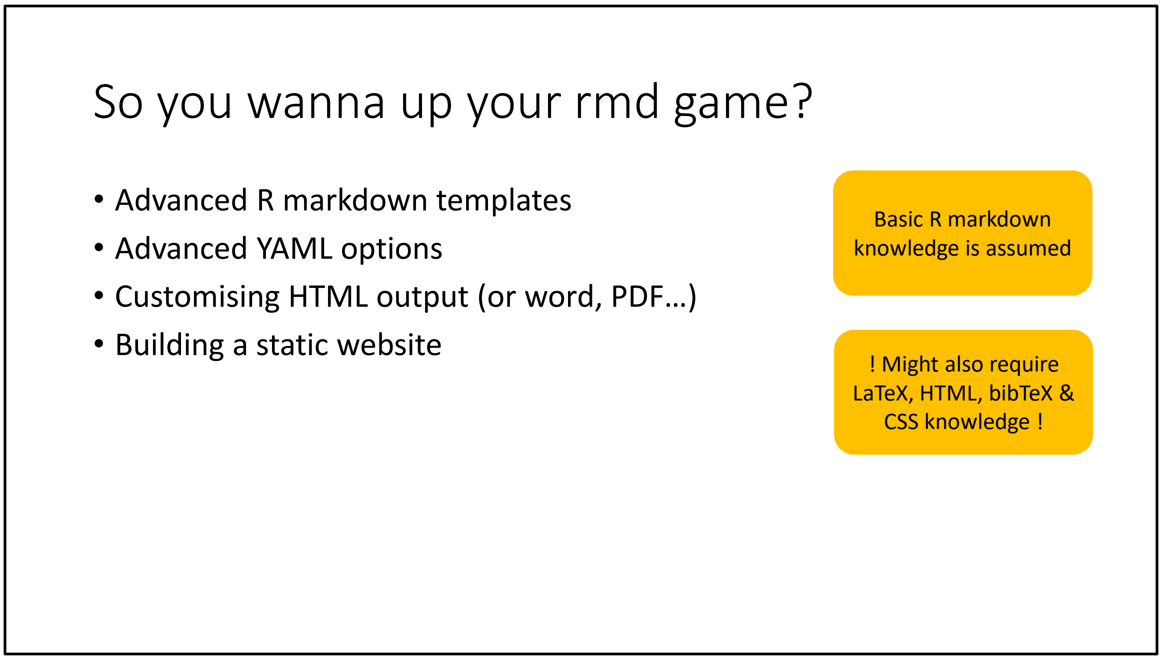
Advanced R Markdown

NERD R Club

24th March 2021



So you wanna up your rmd game?

- Advanced R markdown templates
- Advanced YAML options
- Customising HTML output (or word, PDF…)
- Building a static website

Basic R markdown knowledge is assumed

! Might also require LaTeX, HTML, bibTeX & CSS knowledge !

Here we will cover some common features to extend the basic R markdown experience. We will focus on HTML outputs because they offer a lot of accessible customisation but LaTeX would be the most customisable – at which point why bother with markdown?

I will assume you are comfortable with basic R markdown like using markup, knitting to different file types or using code chunks.

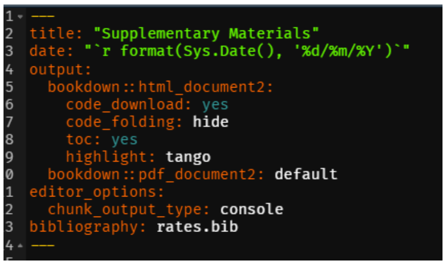
I will also assume that you are comfortable with R projects.

You will see that the more advanced features require other computing science knowledge such as HTML, LaTeX, CSS or bibTeX.

Default template is ~basic~

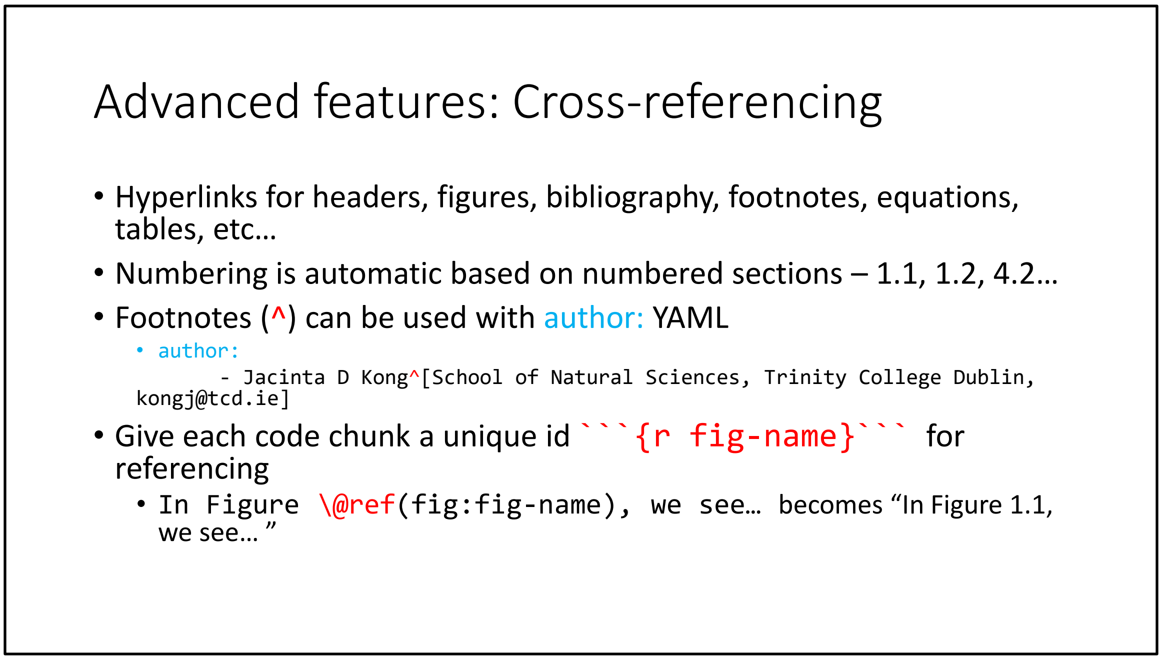
Assumed knowledge: you are familiar with the default html_document template

- Other templates have more features:
  - Tufte style (with margin text)
  - **Bookdown::html_document2** (or word_document2, pdf_document2)
- Advantages:
  - Cross-referencing
  - Figure numbering
    - With `cap = “<caption>”` in chunk option
  - Equation labelling & numbering



```yaml
---
title: "Supplementary Materials"
date: "`r format(Sys.Date(), '%d/%m/%Y')`"
output:
  bookdown::html_document2:
    code_download: yes
    code_folding: hide
    toc: yes
    highlight: tango
  bookdown::pdf_document2: default
editor_options:
  chunk_output_type: console
bibliography: rates.bib
---
```

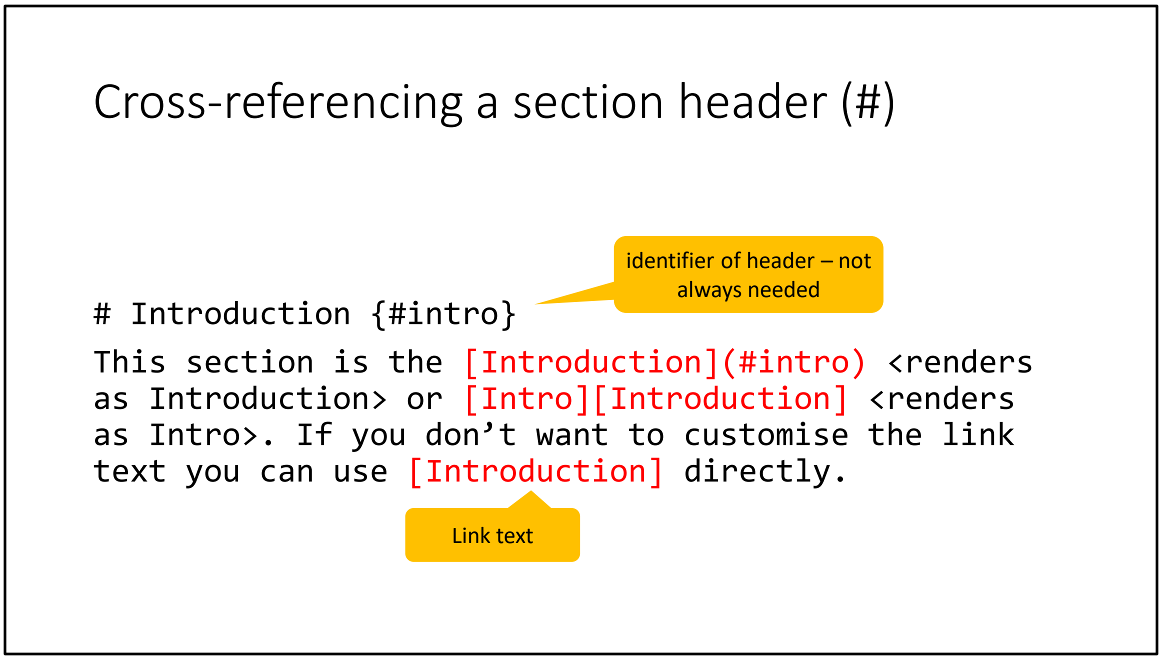
The default templates in the R markdown package are fine for simple reports but aren’t designed for technical documents. The main features missing from the default templates that are useful for technical documents are cross-referencing, figure or table numbering and equation numbering. We can use templates from other packages that support the extra features we want. The bookdown package is for authoring technical documents, including chapter books. Another example is the Tufte style that uses margin text or figures. Accessing these templates is simple. Install the packages and add the template to the output section in the YAML like you would for the default templates.



Advanced features: Cross-referencing

- Hyperlinks for headers, figures, bibliography, footnotes, equations, tables, etc…
- Numbering is automatic based on numbered sections – 1.1, 1.2, 4.2…
- Footnotes (^) can be used with author: YAML
  - author:
    ```
            - Jacinta D Kong^[School of Natural Sciences, Trinity College Dublin,
    kongj@tcd.ie]
    ```
- Give each code chunk a unique id ```{r fig-name}``` for referencing
  - In Figure \@ref(fig:fig-name), we see… becomes “In Figure 1.1, we see… ”

Cross-referencing is adding hyperlinks within the document. For example, linking to another section or a table. By default the headers are numbered. A good habit is to always give your code chunk a unique name. The name cannot have spaces. The markup \@ref() is used to link figures, tables or equations. Cross-referencing works the same way as in Word.



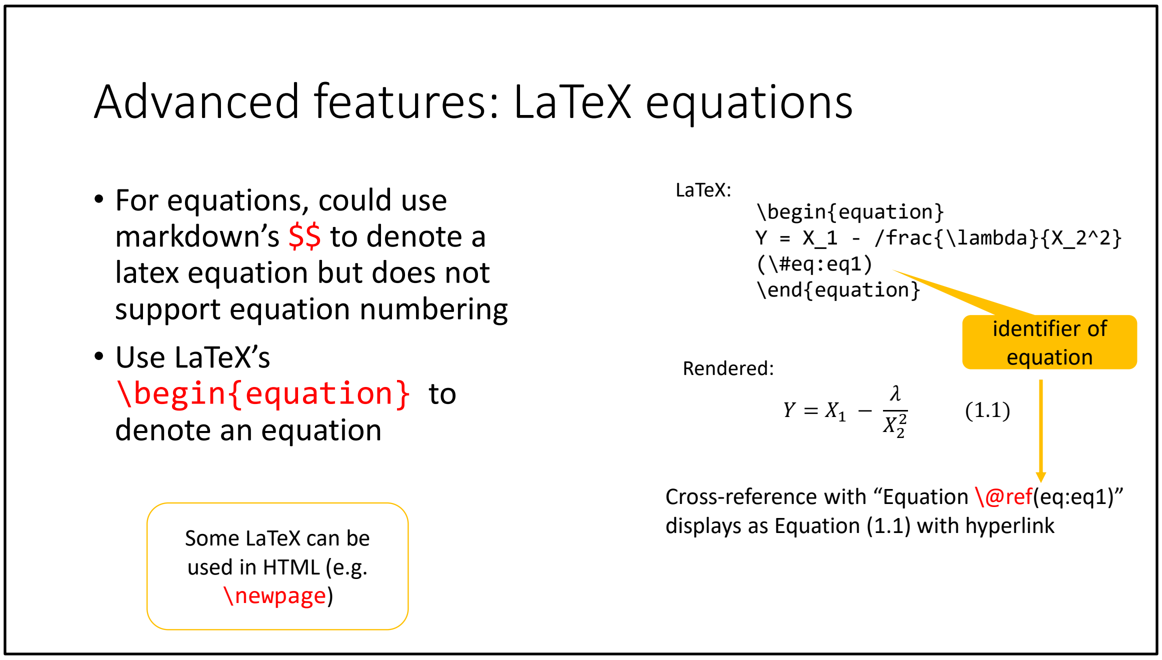
Cross-referencing a section header (#)

```
# Introduction {#intro}
This section is the [Introduction](#intro) <renders
as Introduction> or [Intro][Introduction] <renders
as Intro>. If you don’t want to customise the link
text you can use [Introduction] directly.
```

identifier of header – not always needed
Link text

There are three ways of cross-referencing headers. You can:

1. Give the header an identifier and link the identifier
2. Link via the name of the section with a custom link text – may get clunky if your header name is long, hence the identifier
3. Link using only the name of the section – the link text is the entire header



Advanced features: LaTeX equations

- For equations, could use markdown's $$ to denote a latex equation but does not support equation numbering
- Use LaTeX's `\begin{equation}` to denote an equation

Some LaTeX can be used in HTML (e.g. \newpage)

LaTeX:

```
\begin{equation}
Y = X_1 - /frac{\lambda}{X_2^2}
(\#eq:eq1)
\end{equation}
```

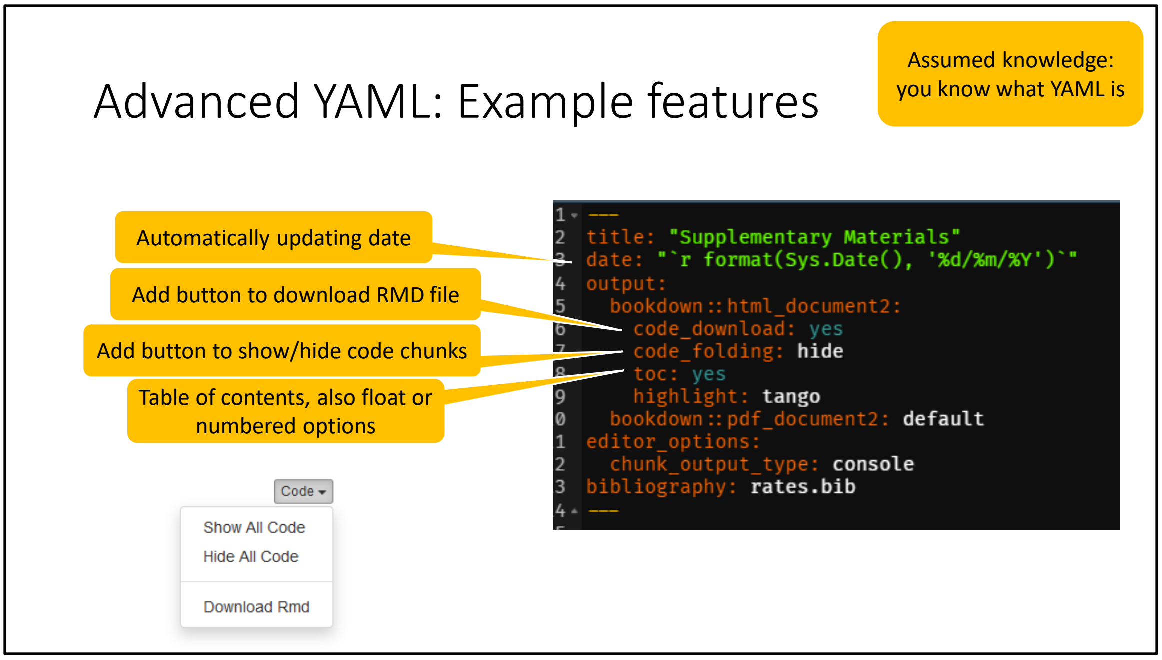
identifier of equation

Rendered:

$$Y = X_1 - \frac{\lambda}{X_2^2} \qquad (1.1)$$

Cross-reference with "Equation \@ref(eq:eq1)" displays as Equation (1.1) with hyperlink

LaTeX equations are supported but to allow equation numbering you must use LaTex formatting, not markdown. Like how $$ denotes the start and end of an equation, \begin{equation} and \end{equation} denote the start and end using LaTeX markup. Then each equation must have a unique name (\#eq:name) that you use to create the cross-reference. Only the equation number is cross-referenced so you will need to write "Equation".



Advanced YAML: Example features

Assumed knowledge: you know what YAML is

Automatically updating date

Add button to download RMD file

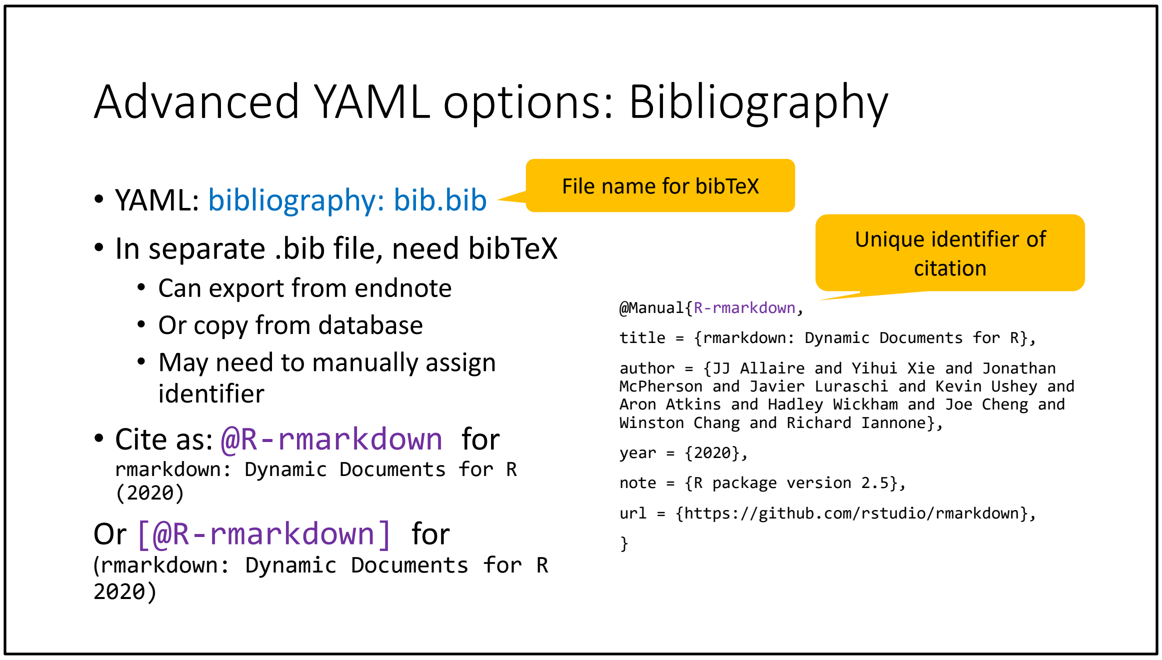
Add button to show/hide code chunks

Table of contents, also float or numbered options

```yaml
---
title: "Supplementary Materials"
date: "`r format(Sys.Date(), '%d/%m/%Y')`"
output:
  bookdown::html_document2:
    code_download: yes
    code_folding: hide
    toc: yes
    highlight: tango
  bookdown::pdf_document2: default
editor_options:
  chunk_output_type: console
bibliography: rates.bib
---
```

These features work with the default template. They may make the document easier to share and read:

1. Adding R code to YAML requires “, ‘ and `. ` r` to tell markdown that this is R code, “ to tell YAML that this isn’t YAML and ‘ for use within the R code so that the string isn’t completed. “ and ‘ can be used interchangeably.
2. Code download adds a drop down menu that lets a reader download the original Rmd from the HTML which means one less file to share
3. Code folding shows or hides all the code chunks which may break up the flow of the document. The chunks can be hidden or shown by default – here the default is hide
4. Table of contents are always useful for long documents. The content can be numbered sections or can float. Floating ToCs will scroll with the user so that it’s always visible.



Advanced YAML options: Bibliography

- YAML: bibliography: bib.bib

File name for bibTeX

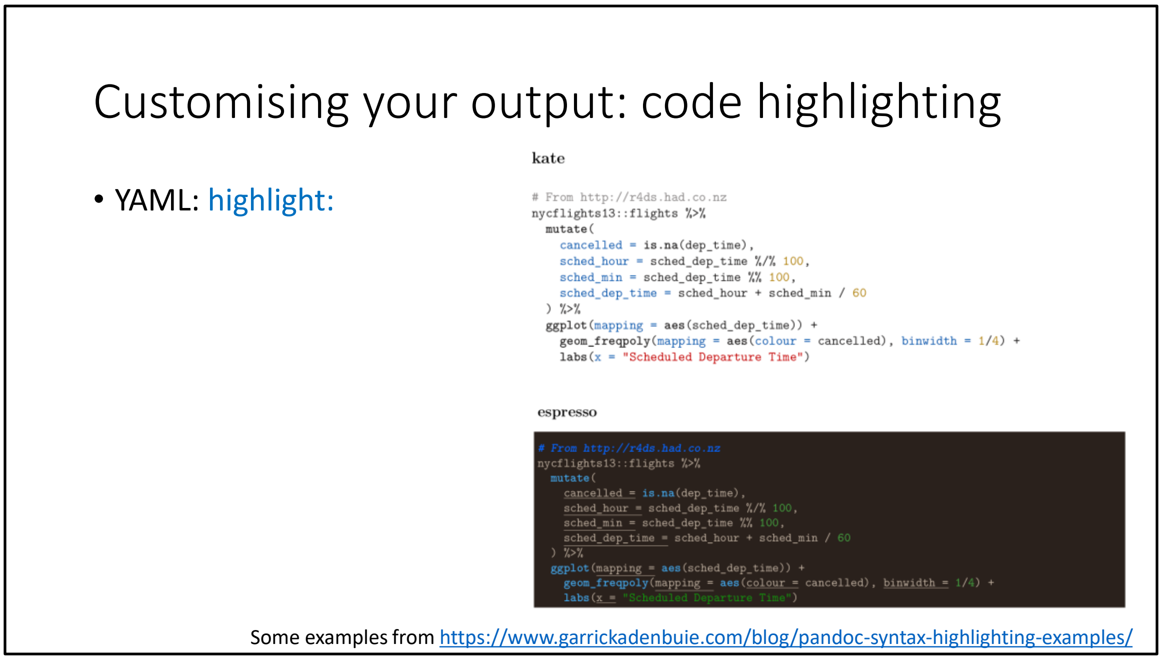
- In separate .bib file, need bibTeX
  - Can export from endnote
  - Or copy from database
  - May need to manually assign identifier
- Cite as: `@R-rmarkdown` for
  `rmarkdown: Dynamic Documents for R (2020)`

Or `[@R-rmarkdown]` for
`(rmarkdown: Dynamic Documents for R 2020)`

Unique identifier of citation

```
@Manual{R-rmarkdown,
title = {rmarkdown: Dynamic Documents for R},
author = {JJ Allaire and Yihui Xie and Jonathan
McPherson and Javier Luraschi and Kevin Ushey and
Aron Atkins and Hadley Wickham and Joe Cheng and
Winston Chang and Richard Iannone},
year = {2020},
note = {R package version 2.5},
url = {https://github.com/rstudio/rmarkdown},
}
```

Bibliographies work the same as LaTeX using a separate .bib file containing the references and their fields. Each reference needs a unique identifier that is used in the in-line citation. You can cite as author (date) or (author date). The bib file needs to be named in the YAML.



Customising your output: code highlighting

- YAML: highlight:

kate

```
# From http://r4ds.had.co.nz
nycflights13::flights %>%
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %/% 100,
    sched_min = sched_dep_time %% 100,
    sched_dep_time = sched_hour + sched_min / 60
  ) %>%
  ggplot(mapping = aes(sched_dep_time)) +
    geom_freqpoly(mapping = aes(colour = cancelled), binwidth = 1/4) +
    labs(x = "Scheduled Departure Time")
```

espresso

```
# From http://r4ds.had.co.nz
nycflights13::flights %>%
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %/% 100,
    sched_min = sched_dep_time %% 100,
    sched_dep_time = sched_hour + sched_min / 60
  ) %>%
  ggplot(mapping = aes(sched_dep_time)) +
    geom_freqpoly(mapping = aes(colour = cancelled), binwidth = 1/4) +
    labs(x = "Scheduled Departure Time")
```

Some examples from https://www.garrickadenbuie.com/blog/pandoc-syntax-highlighting-examples/

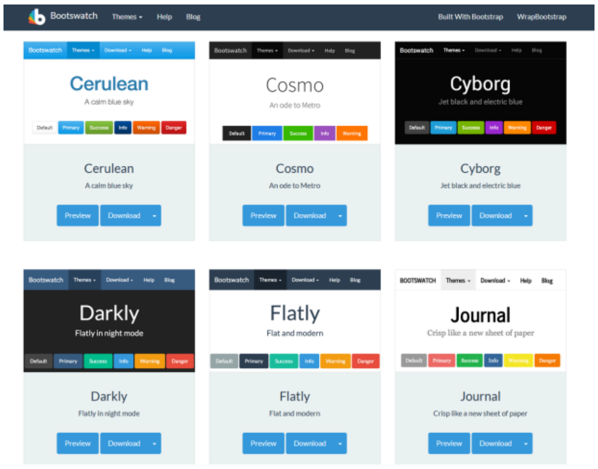
Code highlight is a matter of preference. It changes the colour of the code chunk and which terms are colour coded. pygments is default – it is based off Python's colour scheme. Other colours are kate, monochrome, breezeDark, espresso, zenburn, haddock, and tango.

Customising your output: themes

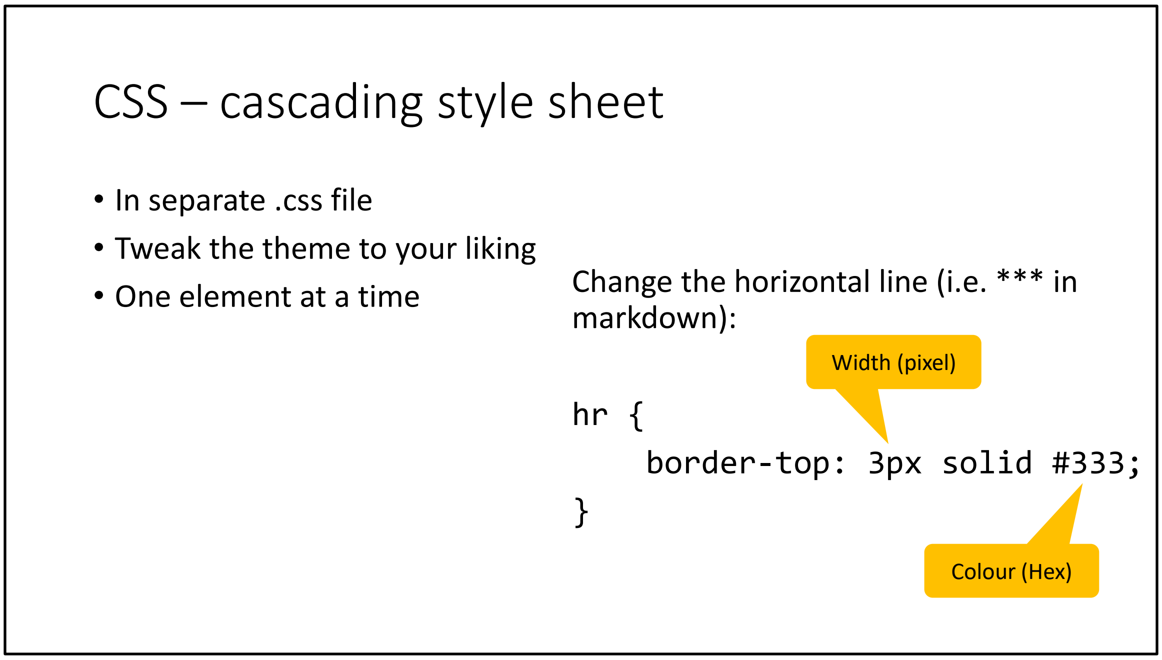
- YAML: theme:
- From Bootswatch

https://bootswatch.com/3/

- Can customise individual elements with CSS
  - YAML: css: filename.css



Defining the theme will change the entire colour scheme and font. You can look at the available themes online.



CSS – cascading style sheet

- In separate .css file
- Tweak the theme to your liking
- One element at a time

Change the horizontal line (i.e. *** in markdown):

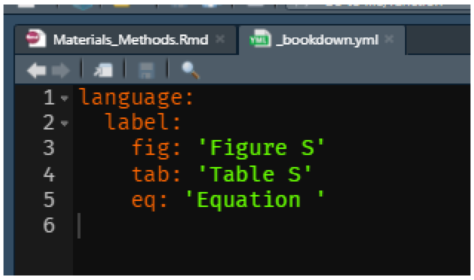
```
hr {
    border-top: 3px solid #333;
}
```

Width (pixel)
Colour (Hex)

If you want to tweak the theme you can manually add a CSS file to the YAML (css). Whatever element you want to change.

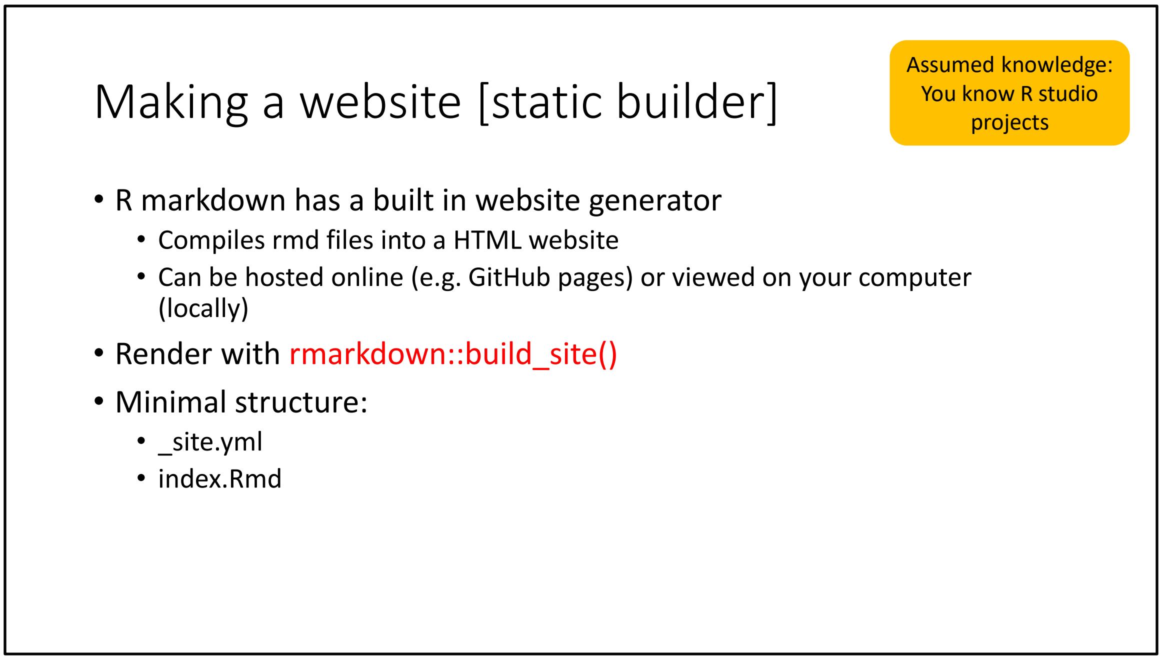
Customising your output: Labels

- Default is Figure, Table, Equation etc…
- What if you want Figure S, Table S, Eqn?
- Have separate file called "_bookdown.yml" in same directory as main rmd
- Define YAML for language and label
- Will automatically be included in captions when knitted
  - E.g. Figure S1.2



```
language:
  label:
    fig: 'Figure S'
    tab: 'Table S'
    eq: 'Equation '
```

What if you want to change the labelling on your figure captions, like to Fig or Figure S for supplementary. You need to have a separate YAML file called _bookdown.yml. No exceptions. This doesn't work if the YAML is in the main document. To change the language you use the YAML lang – e.g. lang: de for German.



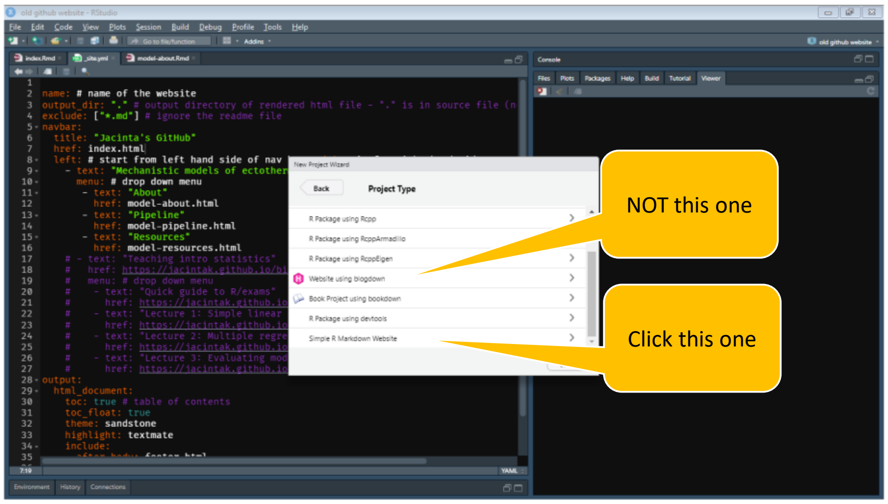
Making a website [static builder]

Assumed knowledge: You know R studio projects

- R markdown has a built in website generator
  - Compiles rmd files into a HTML website
  - Can be hosted online (e.g. GitHub pages) or viewed on your computer (locally)
- Render with rmarkdown::build_site()
- Minimal structure:
  - _site.yml
  - index.Rmd

You can make a website in Rmarkdown by having a site YAML configuration file and a Rmd file called index.

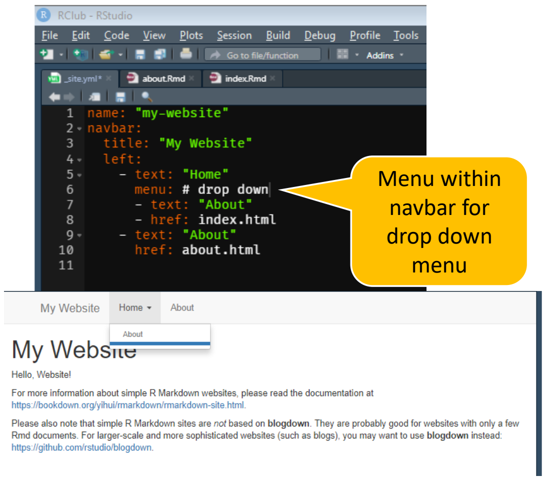
Starting a website via New Project



```
name: # name of the website
output_dir: "." # output directory of rendered html file - "." is in source file (n
exclude: ["*.md"] # ignore the readme file
navbar:
  title: "Jacinta's GitHub"
  href: index.html
  left: # start from left hand side of nav
    - text: "Mechanistic models of ectother
      menu: # drop down menu
       - text: "About"
         href: model-about.html
       - text: "Pipeline"
         href: model-pipeline.html
       - text: "Resources"
         href: model-resources.html
    # - text: "Teaching intro statistics"
    #   href: https://jacintak.github.io/bi
    #   menu: # drop down menu
    #     - text: "Quick guide to R/exams"
    #       href: https://jacintak.github.io
    #     - text: "Lecture 1: Simple linear
    #       href: https://jacintak.github.io
    #     - text: "Lecture 2: Multiple regre
    #       href: https://jacintak.github.io
    #     - text: "Lecture 3: Evaluating mod
    #       href: https://jacintak.github.io
output:
  html_document:
    toc: true # table of contents
    toc_float: true
    theme: sandstone
    highlight: textmate
    include:
      after_body: footer.html
```

Choose simple R markdown website

Website components: _site.yml


RClub - RStudio
File Edit Code View Plots Session Build Debug Profile Tools
Go to file/function
Addins
_site.yml*
about.Rmd
index.Rmd
Menu within navbar for drop down menu
My Website
Home
About
About

```yaml
name: "my-website"
navbar:
  title: "My Website"
  left:
    - text: "Home"
      menu: # drop down
      - text: "About"
      - href: index.html
    - text: "About"
      href: about.html
```

My Website

Hello, Website!

For more information about simple R Markdown websites, please read the documentation at https://bookdown.org/yihui/rmarkdown/rmarkdown-site.html.

Please also note that simple R Markdown sites are *not* based on **blogdown**. They are probably good for websites with only a few Rmd documents. For larger-scale and more sophisticated websites (such as blogs), you may want to use **blogdown** instead: https://github.com/rstudio/blogdown.

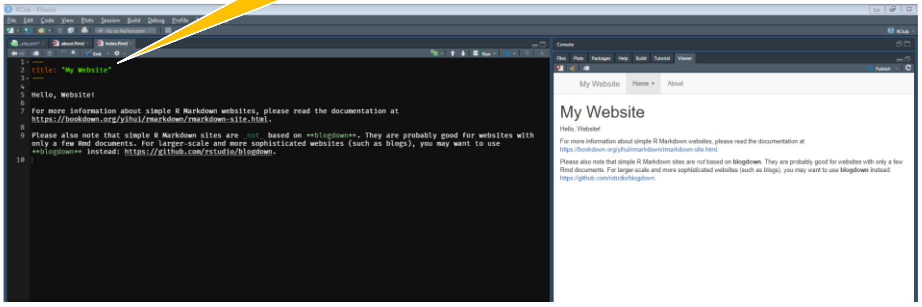
- Website settings for all pages
- "_" in file name tells rmarkdown to ignore the file when compiling
- Drop down Navbar
  - Left: navbar reads left to right. "right" will read from right to left
  - Text: label
  - Href: URL or html file name

The navbar can read left to right or right to left. You can add a drop down option using menu and add lower level pages. Otherwise only higher level pages are shown.

Website components: index.rmd

No other information in YAML needed

Homepage of website. File name MUST be “index”


RClub - RStudio
File Edit Code View Plots Session Build Debug Profile
_site.yml* about.Rmd index.Rmd
Knit
Run
Console
Files Plots Packages Help Build Tutorial Viewer
Publish
My Website Home About

```
---
title: "My Website"
---

Hello, Website!

For more information about simple R Markdown websites, please read the documentation at
https://bookdown.org/yihui/rmarkdown/rmarkdown-site.html.

Please also note that simple R Markdown sites are _not_ based on **blogdown**. They are probably good for websites with
only a few Rmd documents. For larger-scale and more sophisticated websites (such as blogs), you may want to use
**blogdown** instead: https://github.com/rstudio/blogdown.
```

My Website

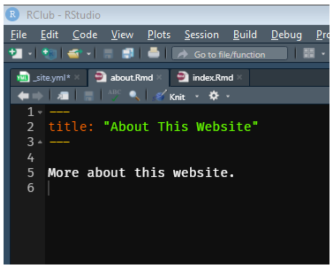
Hello, Website!

For more information about simple R Markdown websites, please read the documentation at https://bookdown.org/yihui/rmarkdown/rmarkdown-site.html.

Please also note that simple R Markdown sites are *not* based on **blogdown**. They are probably good for websites with only a few Rmd documents. For larger-scale and more sophisticated websites (such as blogs), you may want to use **blogdown** instead: https://github.com/rstudio/blogdown.

Could put author or date in YAML, but only a title is needed.

Website components: any other pages

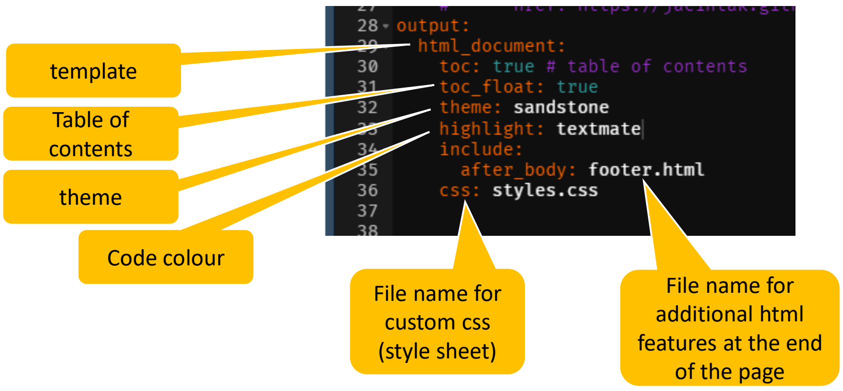


```
---
title: "About This Website"
---

More about this website.

```

- Rmd file
- YAML just needs page title
- Body of website uses regular r markdown including code chunks etc
- Make as many as you want
- Remember to link them to the navbar
- Cross-link with:
  [page name](page.html)
  - e.g. [homepage](index.html)

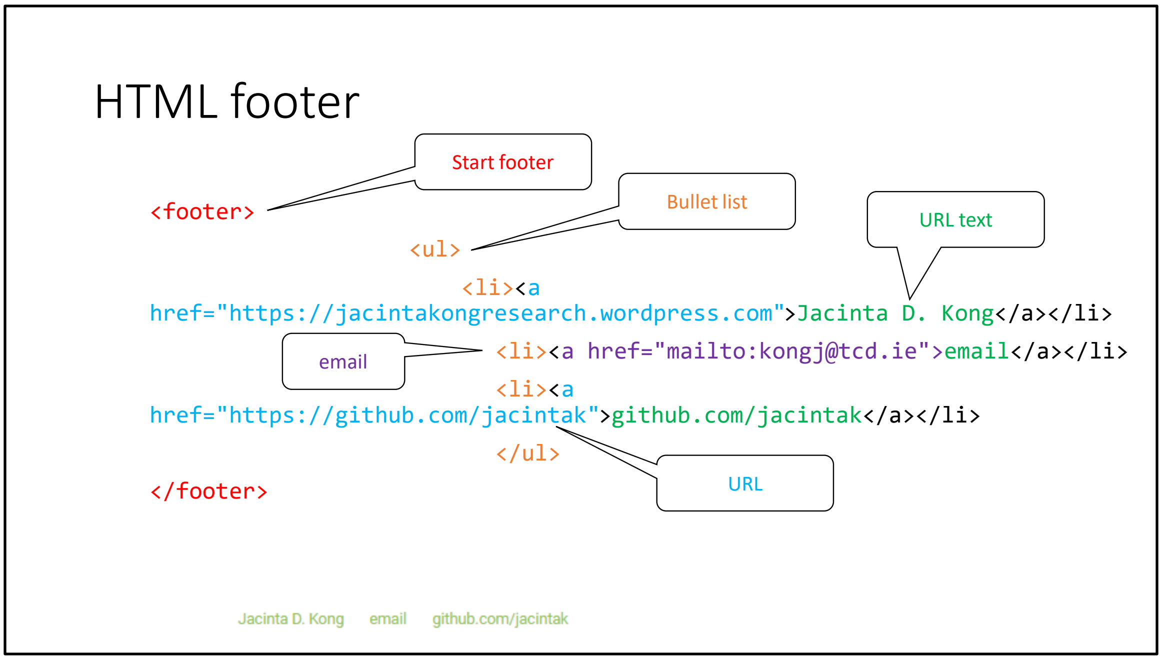
Customising your website appearance

- Use YAML like you would for a regular HTML output



```yaml
output:
  html_document:
    toc: true # table of contents
    toc_float: true
    theme: sandstone
    highlight: textmate
    include:
      after_body: footer.html
    css: styles.css
```

template
Table of contents
theme
Code colour
File name for custom css (style sheet)
File name for additional html features at the end of the page



HTML footer

Start footer

Bullet list

URL text

```
<footer>
                    <ul>
                        <li><a
href="https://jacintakongresearch.wordpress.com">Jacinta D. Kong</a></li>
                           <li><a href="mailto:kongj@tcd.ie">email</a></li>
                           <li><a
href="https://github.com/jacintak">github.com/jacintak</a></li>
                           </ul>
</footer>
```

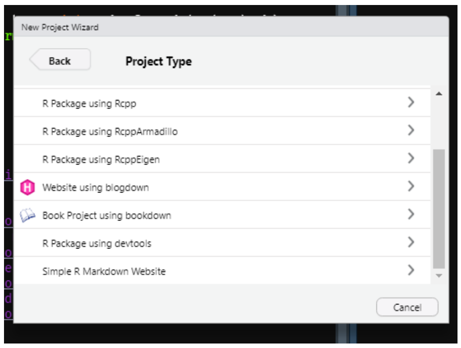
email

URL

Jacinta D. Kong   email   github.com/jacintak

Even more outputs…

- Chapter books like textbooks (bookdown)
- Presentations (not this one!)
- Websites with Hugo (blogdown)


New Project Wizard
Back
Project Type
R Package using Rcpp
R Package using RcppArmadillo
R Package using RcppEigen
Website using blogdown
Book Project using bookdown
R Package using devtools
Simple R Markdown Website
Cancel

The built-in website generator is great for simple sites. A more customisable and powerful engine is Hugo which is accessible via the package blogdown. Markdown also lets you make presentations (e.g. beamer style). Bookdown is meant for writing books so we have only scratched the surface here.