R Markdown

R Markdown

A framework for integrating code, text and results.

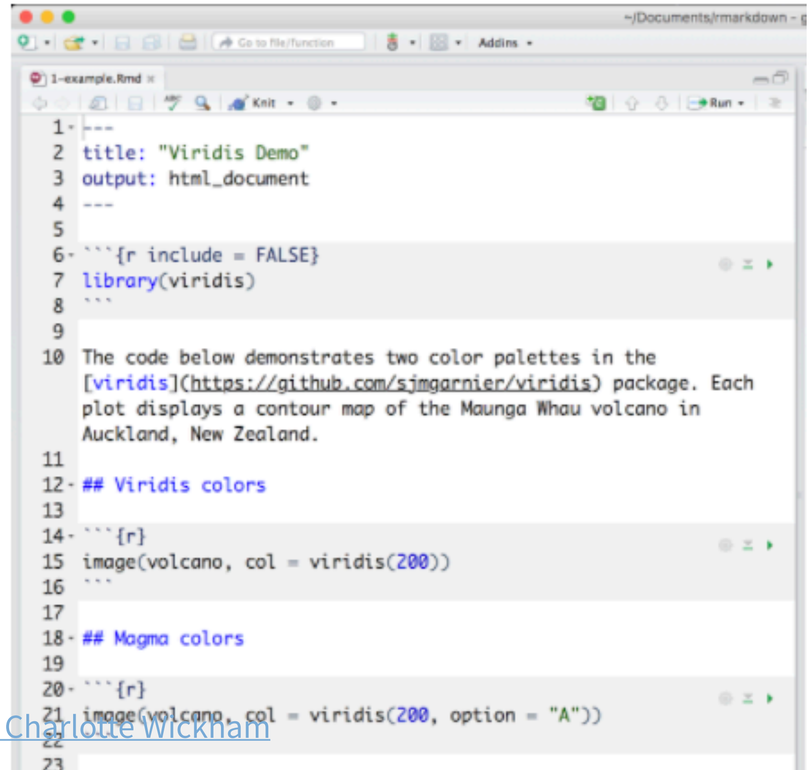
You have already been using it! R Notebooks

How it works

Figure from http://rmarkdown.rstudio.com/lesson-2.html

How it works

.Rmd

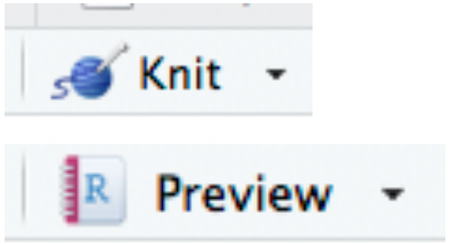


A plain text file,
starts with a header,
contains R in code
chunks, and text.

Figure from http://rmarkdown.rstudio.com/lesson-2.html

How it works

Document is rendered:



Cmd/Ctrl + Shift + r

`rmarkdown::render()`

Figure from http://rmarkdown.rstudio.com/lesson-2.html

How it works

Final document opened in Viewer (if html)

Can be found in same directory as .Rmd (default)

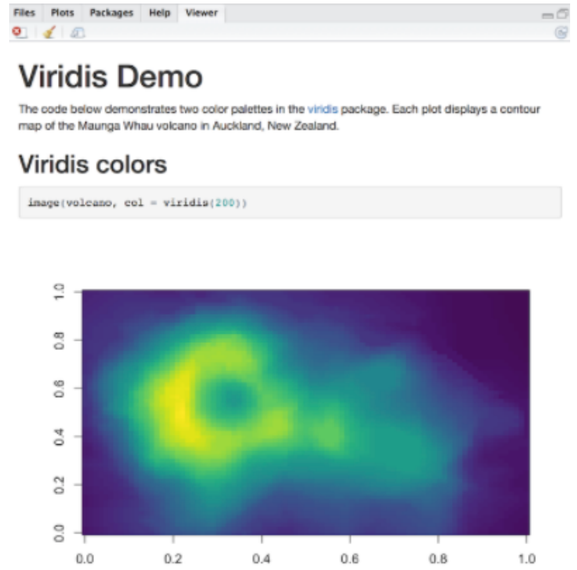


Figure from http://rmarkdown.rstudio.com/lesson-2.html

Structure of a R Markdown file

.Rmd extension

default.Rmd

Knit | Insert | Run

```
---
title: "My Title"
output: html_document
---

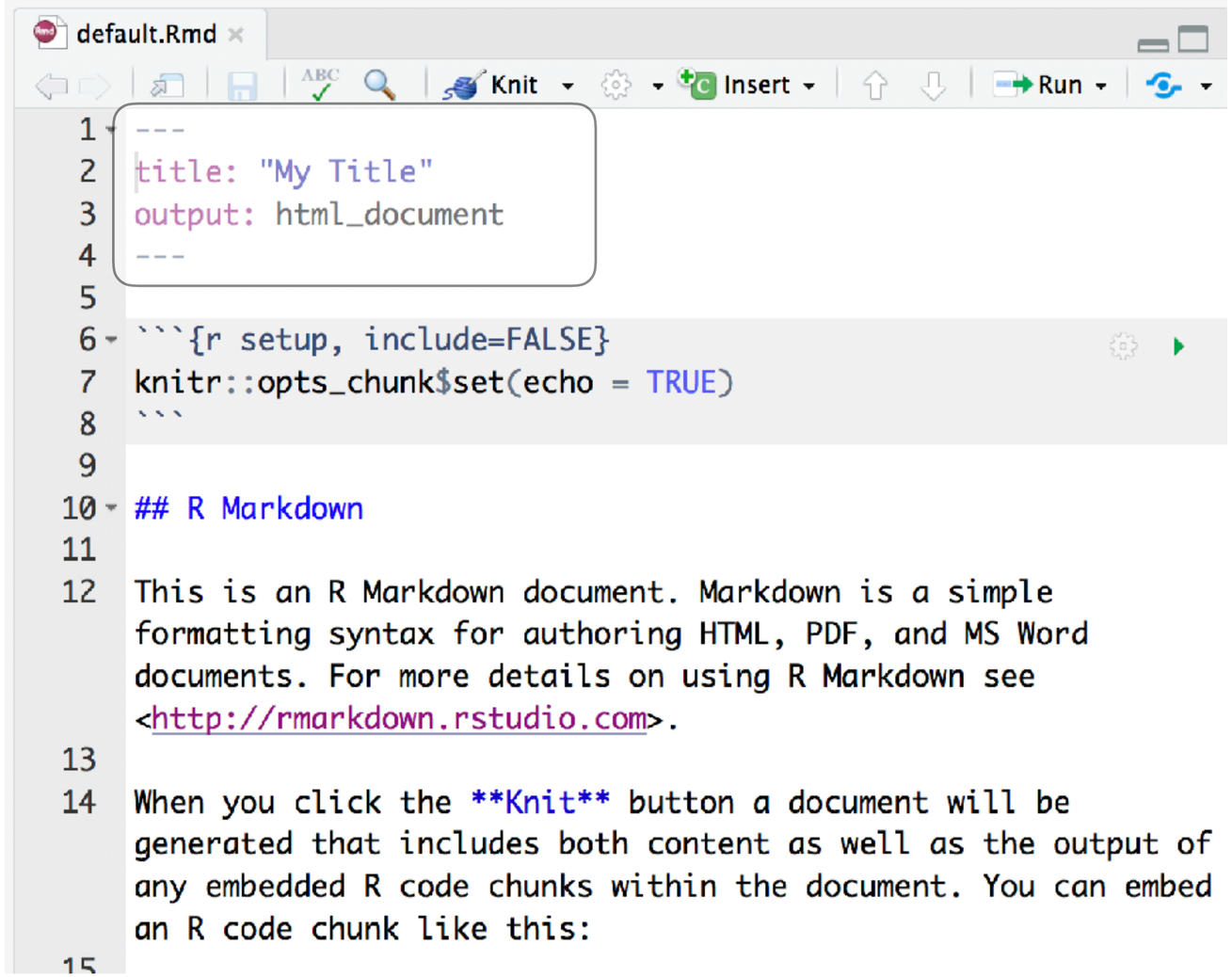
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple
formatting syntax for authoring HTML, PDF, and MS Word
documents. For more details on using R Markdown see
<http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be
generated that includes both content as well as the output of
any embedded R code chunks within the document. You can embed
an R code chunk like this:
```

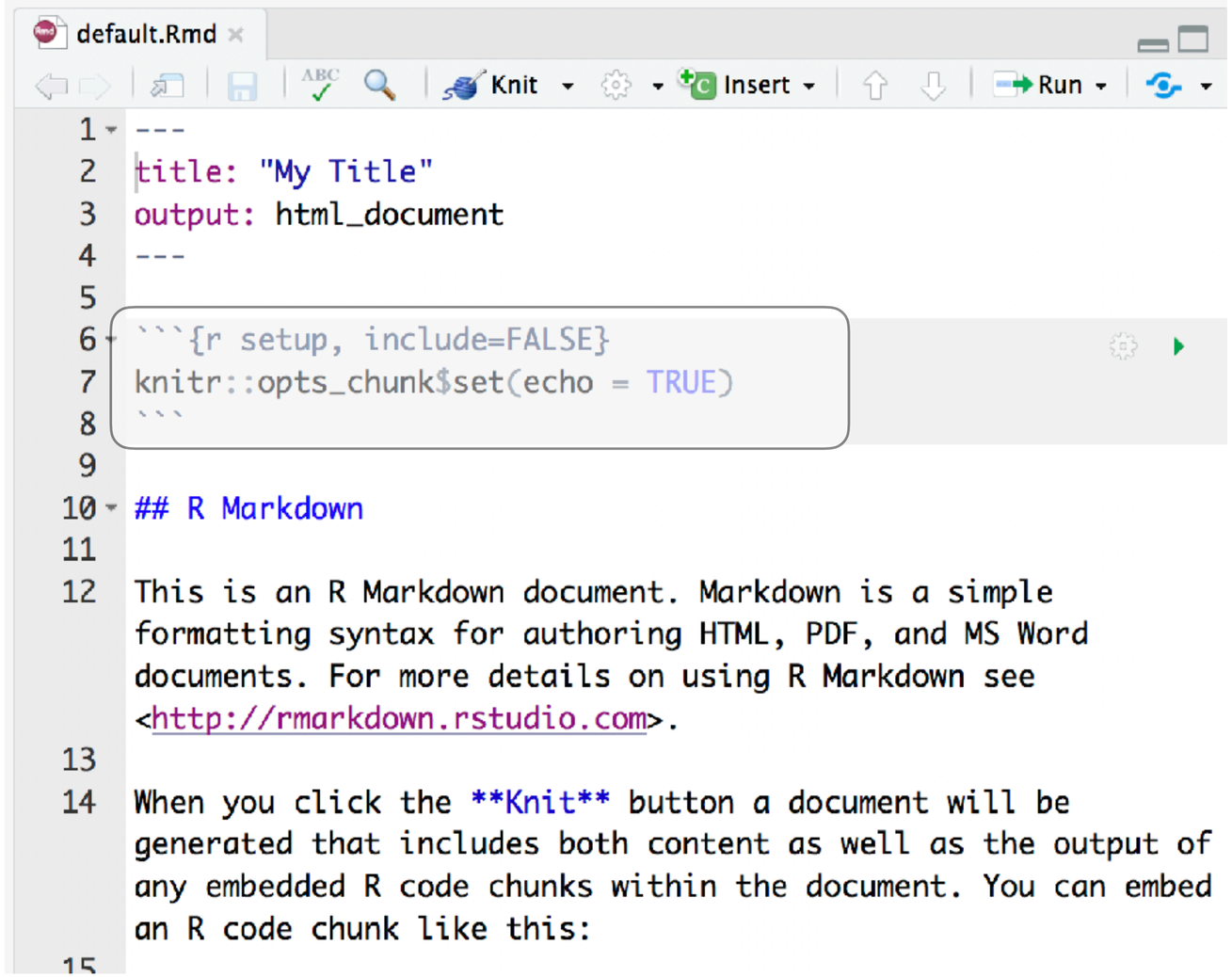
Structure of a R Markdown file



Header `---`

`---`

Structure of a R Markdown file



Code chunks ```{r}
```
```

Structure of a R Markdown file

default.Rmd

Knit | Insert | Run

```
---
title: "My Title"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple
formatting syntax for authoring HTML, PDF, and MS Word
documents. For more details on using R Markdown see
<http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be
generated that includes both content as well as the output of
any embedded R code chunks within the document. You can embed
an R code chunk like this:
```

Text

Code Chunks

````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
````

Code Chunks

Language for code in chunk

````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
````

Optional chunk name

Comma separated set of chunk options

Some common chunk options

`include = FALSE`	Code should be run, but nothing should appear in document
`echo = FALSE`	Code should not appear in document
`results = "hide"`	Results should not be included in document

Your Turn 1

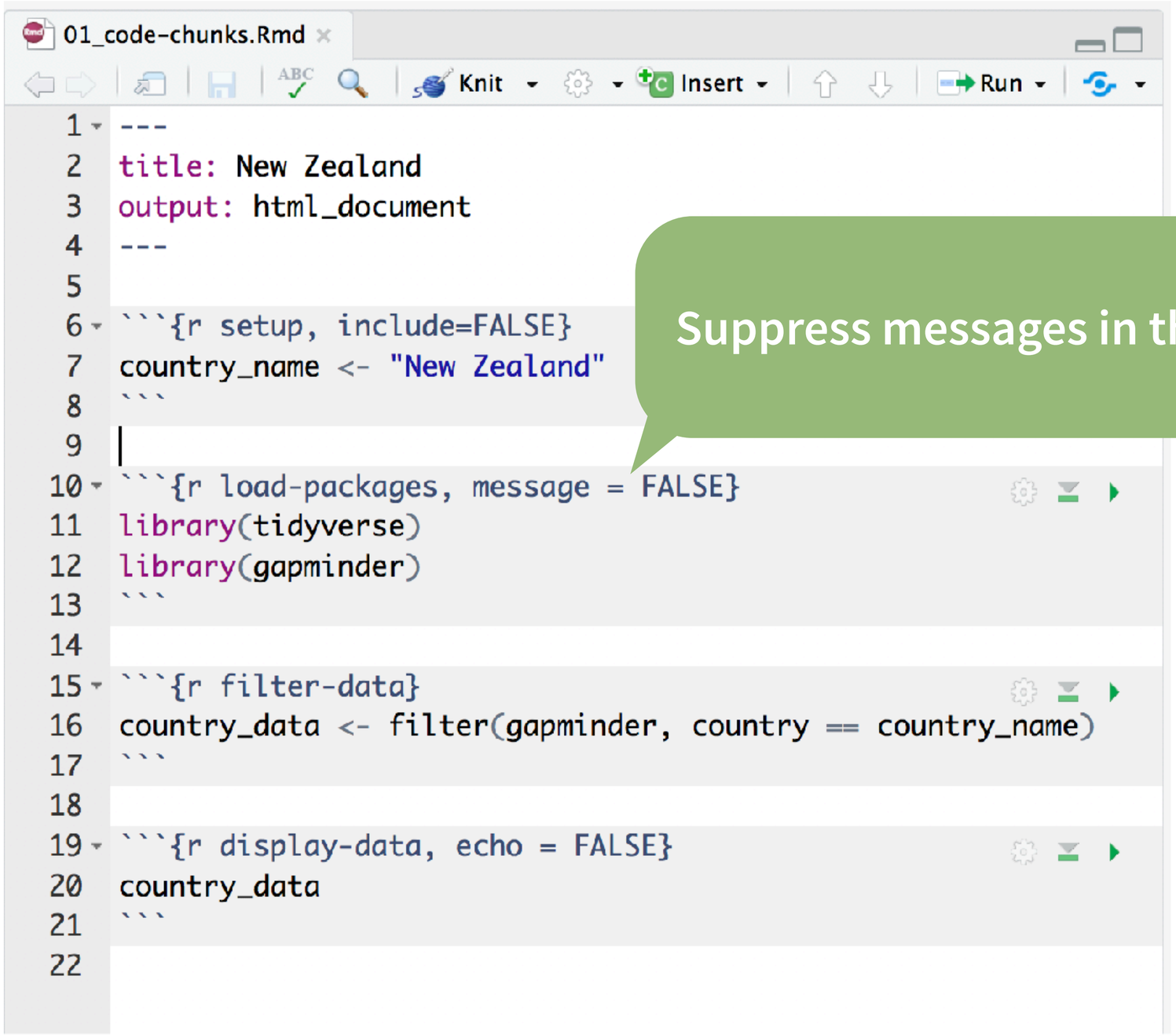
Open 07-rmarkdown/01_code-chunks.Rmd

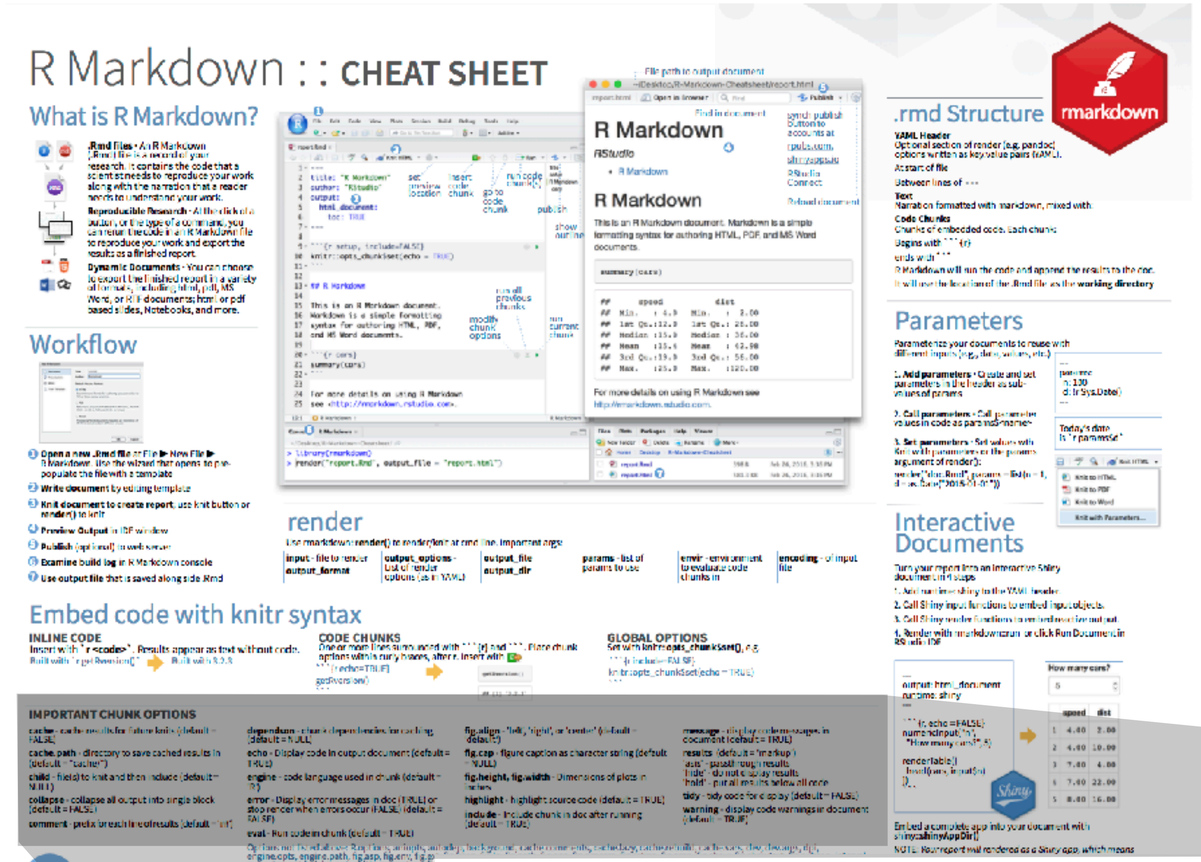
Knit it and take a look.

Edit:

1. Give the three unnamed chunks names
2. Add an option to the last chunk to avoid displaying the code
3. Add message = FALSE to the options for the chunk that loads the packages

Knit again





More options:

`error`

`fig.height, fig.width,`

`warning`

`eval`

IMPORTANT CHUNK OPTIONS

cache - cache results for future knits (default = FALSE)

cache.path - directory to save cached results in (default = "cache/")

child - file(s) to knit and then include (default = NULL)

collapse - collapse all output into single block (default = FALSE)

comment - prefix for each line of results (default = '##')

dependson - chunk dependencies for caching (default = NULL)

echo - Display code in output document (default = TRUE)

engine - code language used in chunk (default = 'R')

error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)

eval - Run code in chunk (default = TRUE)

fig.align - 'left', 'right', or 'center' (default = 'default')

fig.cap - figure caption as character string (default = NULL)

fig.height, fig.width - Dimensions of plots in inches

highlight - highlight source code (default = TRUE)

include - Include chunk in doc after running (default = TRUE)

message - display code messages in document (default = TRUE)

results (default = 'markup')
'asis' - passthrough results
'hide' - do not display results
'hold' - put all results below all code

tidy - tidy code for display (default = FALSE)

warning - display code warnings in document (default = TRUE)

Options not listed above: R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts

Text

Interpreted as markdown, a simple syntax for formatting

```
## Section heading

Normal text

### Sub-section heading

**Bold** text and *italic* text
```

Section heading

Normal text

Sub-section heading

Bold text and *italic* text

R inline

Allows calculated values in text blocks

`` `r  <<code to run here>>` ``

```
Today is `r Sys.Date()`
2 + 2 is `r 2 + 2`
```

Today is 2017-12-06

2 + 2 is 4

Pandoc's Markdown

Write with syntax on the left to create effect on right (after render)

Plain text
End a line with two spaces
to start a new paragraph.
italics and **bold**
`verbatim code`
sub/superscript^2^~2~
~~strikethrough~~
escaped: * _ \\
endash: --, emdash: ---
equation: $A = \pi * r^{2}$
equation block:

$$E = mc^{2}$$

> block quote

Header1 {#anchor}

Header 2 {#css_id}

Header 3 {.css_class}

Header 4

Header 5

Header 6

<!--Text comment-->

\textbf{Tex ignored in HTML}
HTML ignored in pdfs

<http://www.rstudio.com>
[link](www.rstudio.com)
Jump to [Header 1](#anchor)
image:

![Caption](smallorb.png)

* unordered list
  + sub-item 1
  + sub-item 2
    - sub-sub-item 1

* item 2

  Continued (indent 4 spaces)

1. ordered list
2. item 2
   i) sub-item 1
      A. sub-sub-item 1

(@) A list whose numbering

continues after

(@) an interruption

Term 1

: Definition 1

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

- slide bullet 1
- slide bullet 2

(>- to have bullets appear on click)

horizontal rule/slide break:

***

A footnote [^1]

[^1]: Here is the footnote.

Plain text
End a line with two spaces
to start a new paragraph.
italics and **bold**
`verbatim code`
sub/superscript^2
~~strikethrough~~
escaped: * _ \
endash: –, emdash: —
equation: $A = \pi * r^2$
equation block:

$$E = mc^2$$

> block quote

Header1

Header 2

Header 3

Header 4

Header 5

Header 6

HTML ignored in pdfs

http://www.rstudio.com
link
Jump to Header 1
image:

Caption

- unordered list
  - sub-item 1
  - sub-item 2
    - sub-sub-item 1
- item 2

  Continued (indent 4 spaces)

1. ordered list
2. item 2
   i. sub-item 1
      A. sub-sub-item 1

(@) A list whose numbering

continues after

(@) an interruption

Term 1

Definition 1

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

- slide bullet 1
- slide bullet 2

(>- to have bullets appear on click)

horizontal rule/slide break:

A footnote [1]

1. Here is the footnote. ↩

Your Turn 2

1. Open rmarkdown/ 02_text-blocks.Rmd
2. Knit
3. Use markdown syntax to match this
4. Knit

New Zealand

```
library(tidyverse)
library(gapminder)
```

This report examines a subset of the `gapminder` data set. In particular the data for New Zealand.

The report will examine:

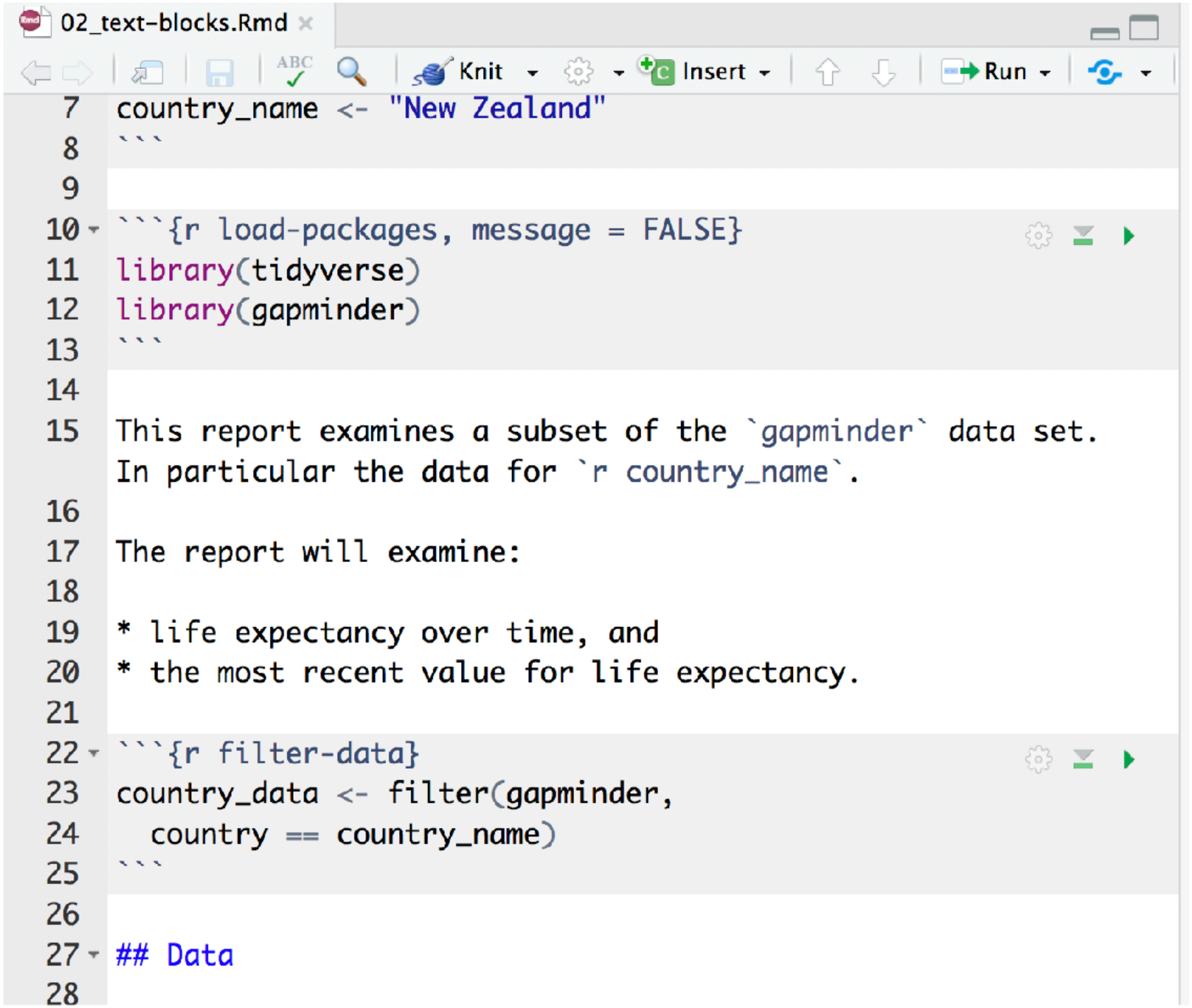
- life expectancy over time, and
- the most recent value for life expectancy.

```
country_data <- filter(gapminder,
  country == country_name)
```

Data

Your Turn 2

1. Open rmarkdown/ 02_text-blocks.Rmd
2. Knit
3. Use markdown syntax to match this
4. Knit



Your Turn 3

1. Open rmarkdown/03_yaml-header.Rmd
2. Knit
3. Knit -> Knit to Word
4. What changes in the .Rmd file?

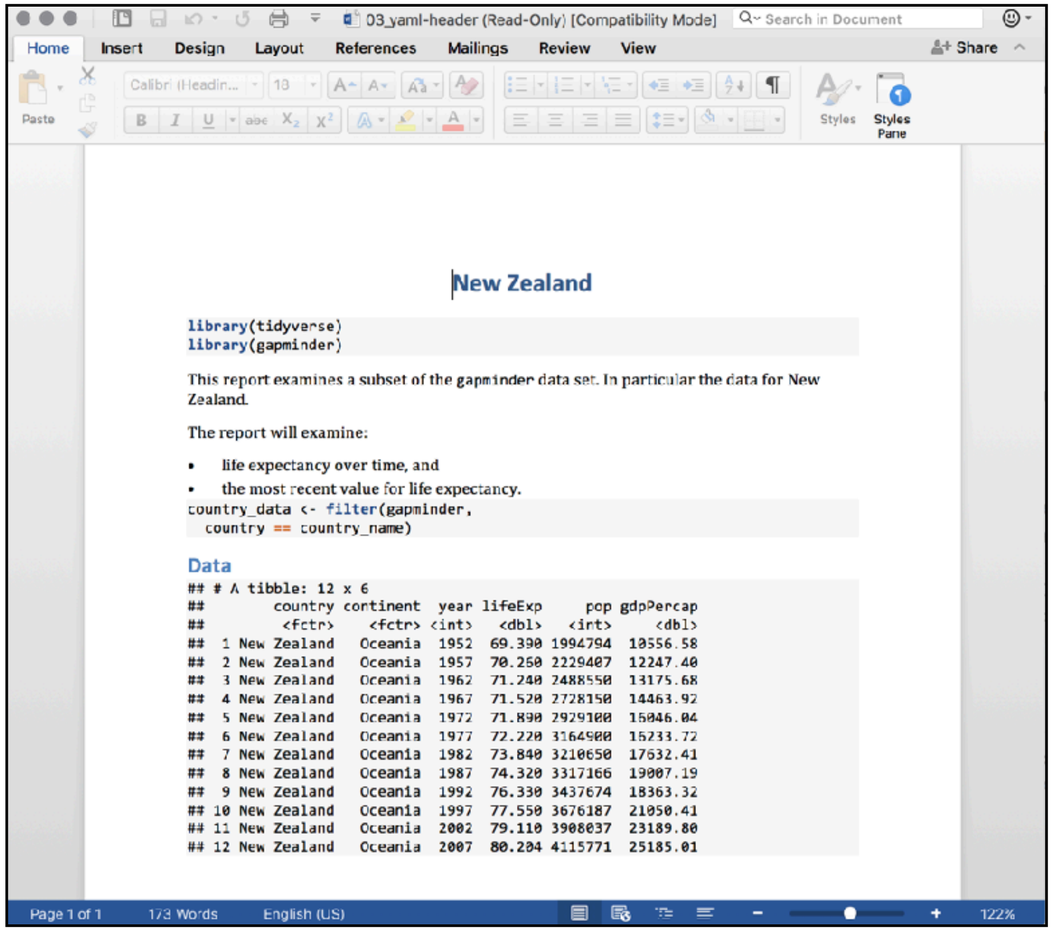
03_yaml-header.Rmd

Knit

```
1  ---
2  title: "New Zealand"
3  output:
4    word_document: default
5    html_document: default
6  ---
7
```

First output will determine output for "Knit" button

03_yaml-header.docx



in same directory as .Rmd

YAML header

Some document options, and controls rendering process

```
---
title: New Zealand
date: 2017-12-07
author: Charlotte Wickham
output: html_document
---
```

YAML header

Some document options, and controls rendering process

```
---
title: New Zealand
date: 2017-12-07
author: Charlotte Wickham
output:
  html_document: default
---
```

Output is html_document with default options

Two spaces

YAML header

Some document options, and controls rendering process

```
---
title: New Zealand
date: 2017-12-07
author: Charlotte Wickham
output:
  html_document:
    toc: true
---
```

Output is html_document with toc option set to true

Your Turn 4

In rmarkdown/04_yaml-header.Rmd

1. Remove `word_document: default`
2. Add `html_document` options:

    `toc: true`

    `df_print: paged`

3. Knit

```
---
title: New Zealand
date: 2017-12-07
author: Charlotte Wickham
output:
  html_document:
    toc: true
---
```

Automation

Your Turn 5

1. Open 05_nz-report.Rmd
2. Take a look at the file, and try to predict the output, then Knit.
3. Edit the report to make it for the "Canada".

````
8
9  ```{r setup, include=FALSE}
10 country_name <- "New Zealand"
11 knitr::opts_chunk$set(echo = FALSE, message = FALSE)
12 ```
13
````

Setting options for all chunks

````
34
35 ```{r, plot-lifeExp}
36 ggplot(country_data) +
37   geom_line(aes(x = year, y = lifeExp)) +
38   labs(title = paste("Life expectancy in", country_name),
39     x = "Year",
40     y = "Life Expectancy") +
41   theme_bw()
42 ```
43
````

Figures in code chunks are included by default

```
---
title: Canada
output:
  html_document:
    toc: true
    df_print: paged
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE, message = FALSE)
country_name <- "Canada"
```

This report examines a subset of the `gapminder` data set. In particular the data for `r country_name`.

```

## solutions/05_canada-report.Rmd

````
---
title: Canada
output:
  html_document:
    toc: true
    df_print: paged

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE, message = FALSE)
country_name <- "Canada"
```
````

06-any-report.Rmd

````
---
title: "`r params$country`"
output:
  html_document:
    toc: true
    df_print: paged
params:
  country: Canada
---

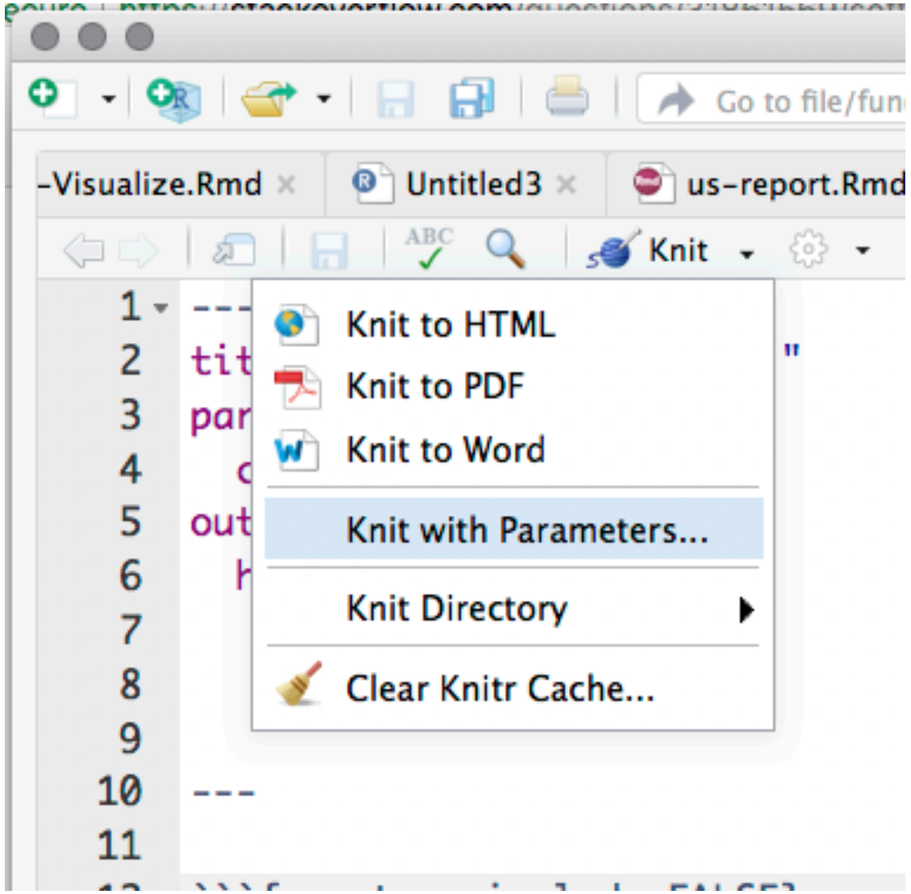

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE, message = FALSE)
country_name <- params$country
```
````

Add your own parameters under the params option

Access the parameter values with `params$name`

Knitting a parameterized report

Knit with Parameters under Knit menu



Or use render() function

```
rmarkdown::render("07-rmarkdown/06_any-report.Rmd",
  params = list(country = "Canada"))
```

```
rmarkdown::render("07-rmarkdown/06_any-report.Rmd",
  output_file = "Canada.html",
  params = list(country = "Canada"))
```

Your Turn 6

Knit 06_any-report.Rmd with

```
country = "Canada"
```

Files | Plots | Packages | Help | Viewer

Canada

- Life Expectancy
- Data

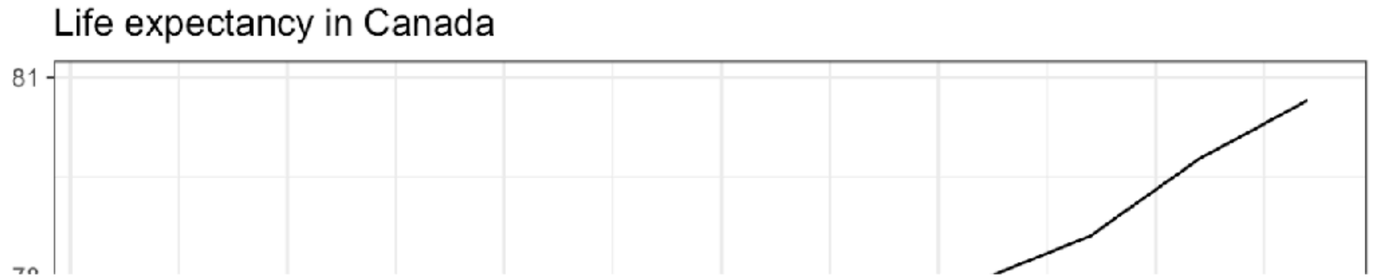
This report examines a subset of the `gapminder` data set. In particular the data for Canada.

The report will examine:

- life expectancy over time, and
- the most recent value for life expectancy.

Life Expectancy

Canada had a life expectancy of 80.7 in 2007.



Makes automating easy

```
countries <- c("New Zealand", "United States", "Canada")

# For each country
walk(countries,
  ~ rmarkdown::render("07-rmarkdown/06_any-report.Rmd",
      output_file = file.path("reports", paste0(.x, ".html")),
      params = list(country = .x))
)
```

`walk()` is in purrr, see http://r4ds.had.co.nz/iteration.html

Workflow

Two common uses:

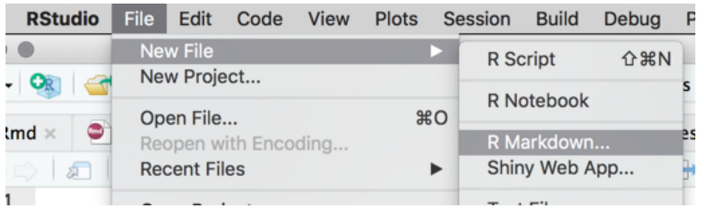
1. Use Rmarkdown like a data analysis log

    Build up text and chunks as you work though a problem

2. At the end of a complicated analysis to communicate

    Usually have other R scripts that perform a analyses and save appropriate results. Rmarkdown reads in results and formats nicely.

To start: File -> New File -> New Rmarkdown...



Add packages for prettier results

Balance between generalizability and fine control over appearance

E.g. pander, basic tables in any output format

E.g. stargazer, pretty tables for models in html or pdf

Other things to look into

HTML output provides interactive components, e.g.: http://www.htmlwidgets.org/showcase_leaflet.html

You can have other languages in code chunks, e.g. SQL, python:

http://rmarkdown.rstudio.com/lesson-5.html

PDF output requires a LaTeX install, but offers the best control over static content.

(Applied) **Data Science**