

R Markdown

R Markdown

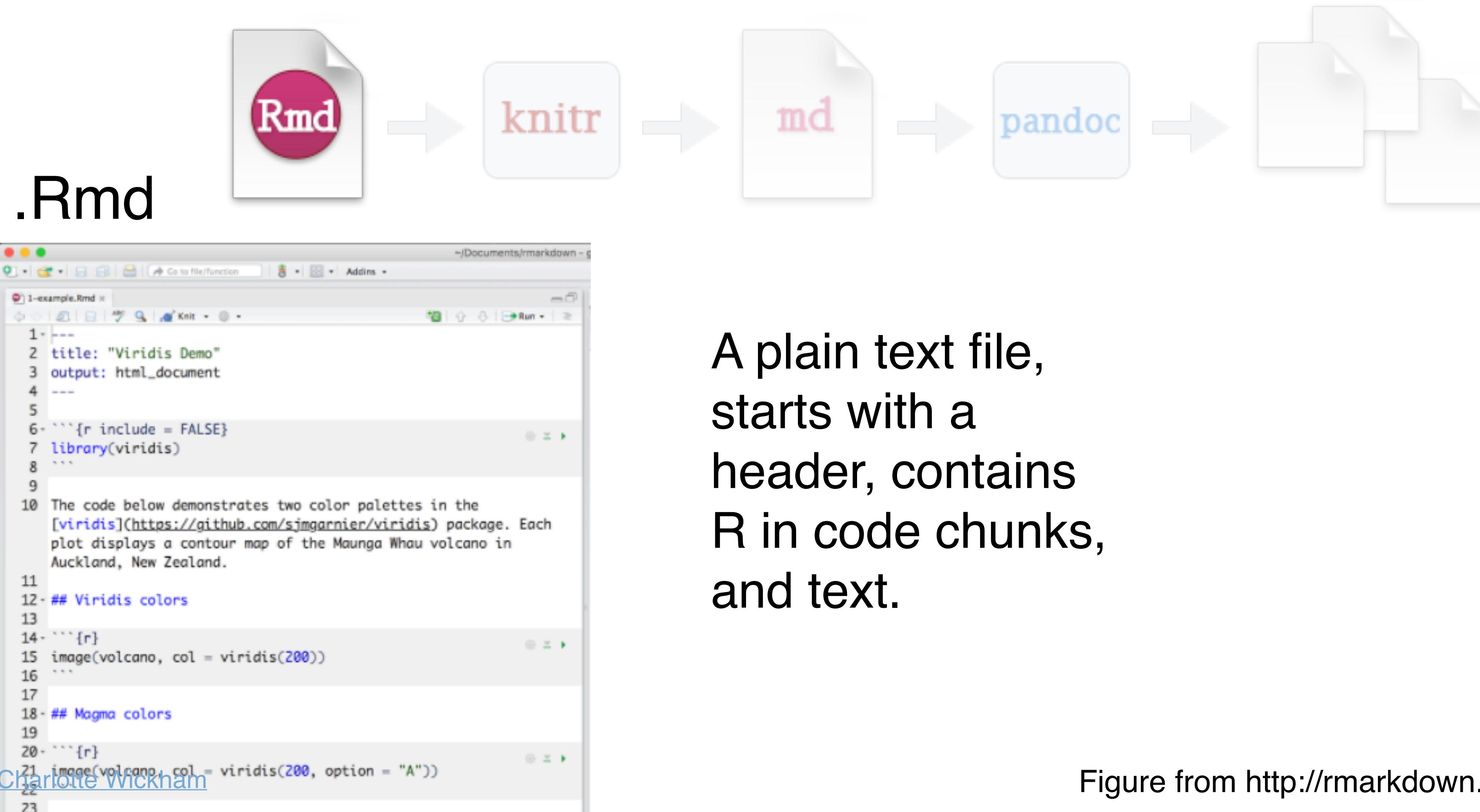
A framework for integrating code, text and results.

You have already been using it! R Notebooks

How it works



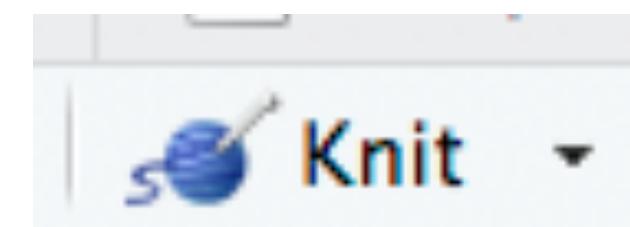
How it works



How it works



Document is rendered:



Cmd/Ctrl + Shift + r

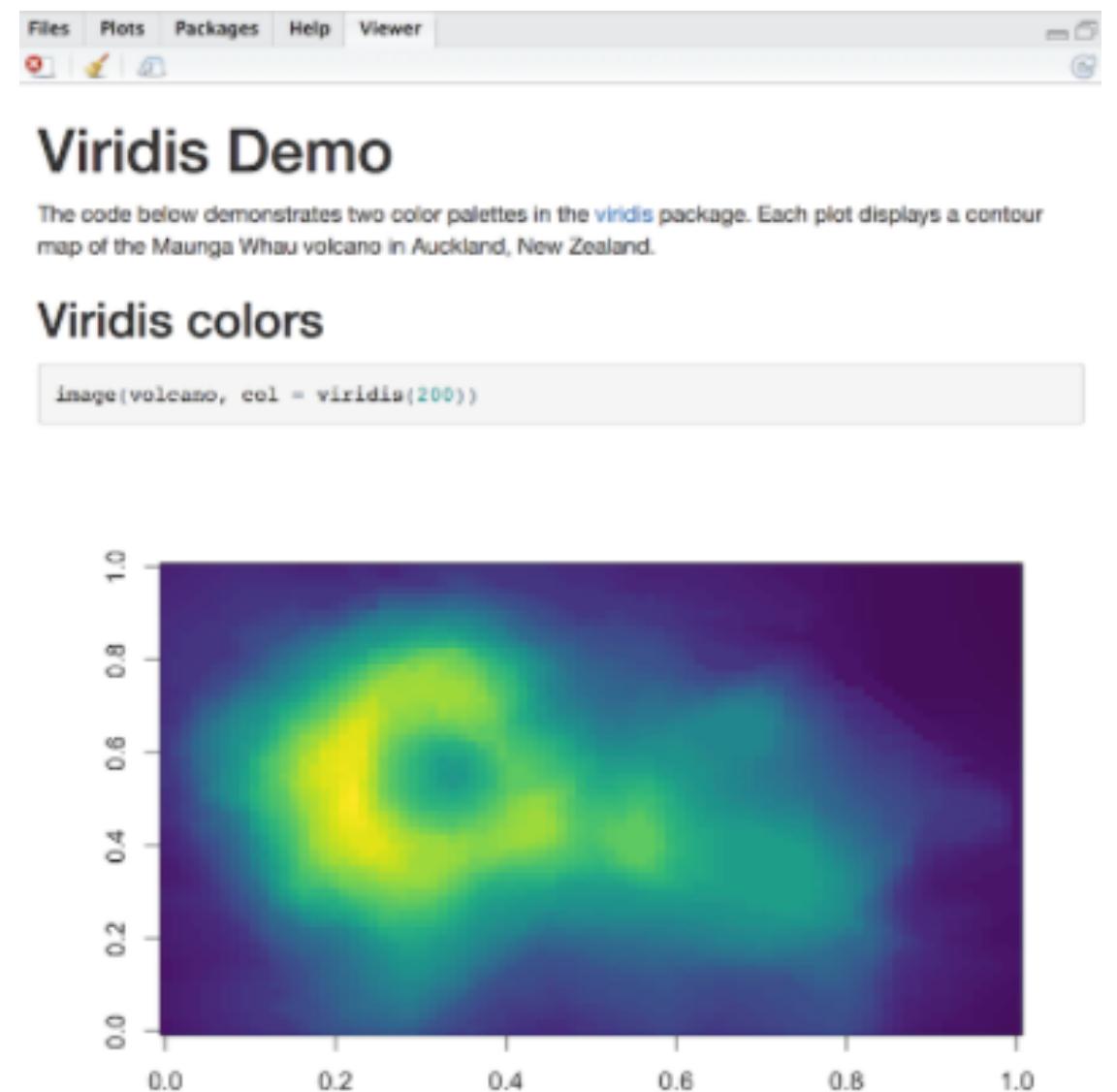
`rmarkdown::render()`

How it works



Final document opened in Viewer (if html)

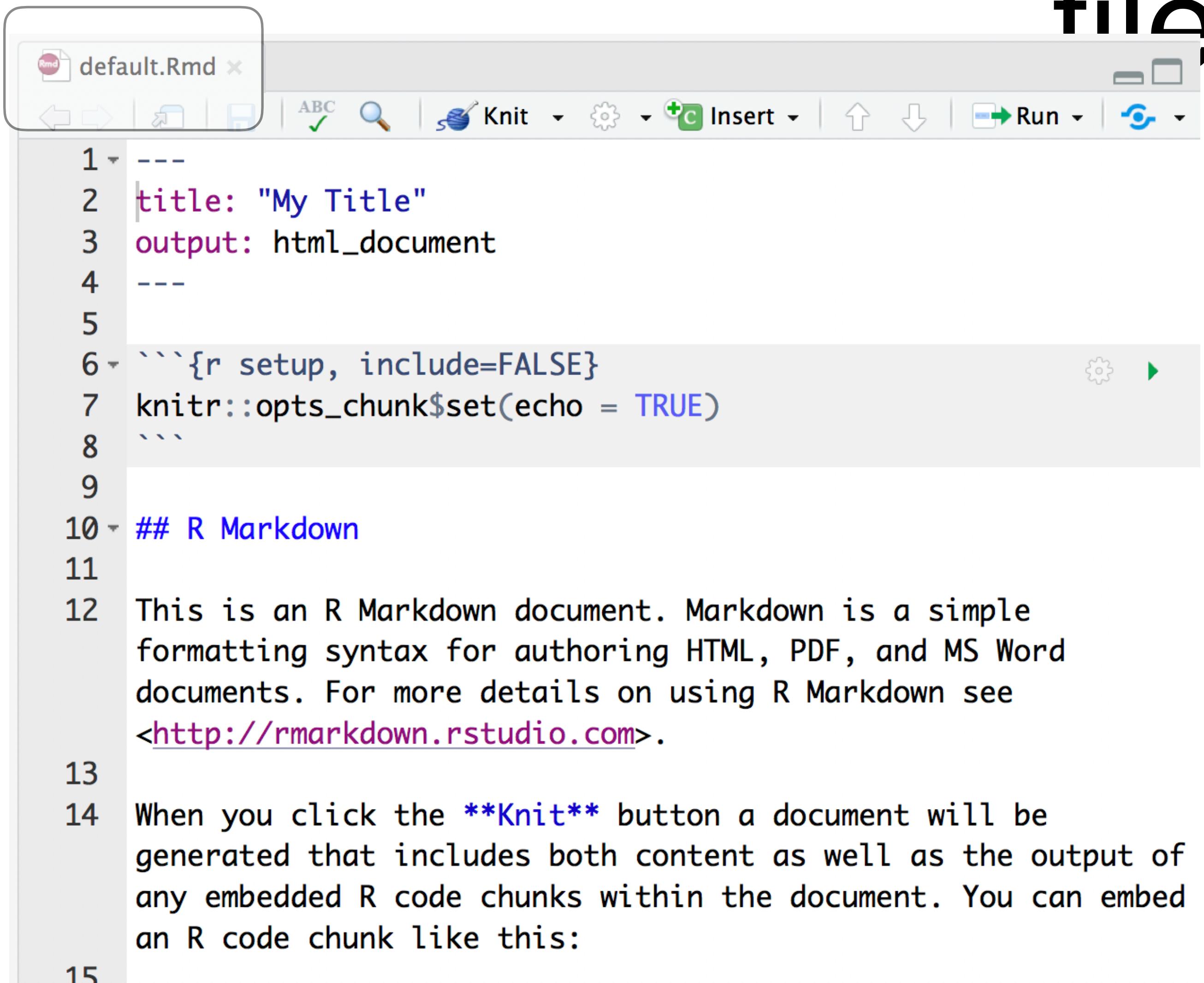
Can be found in same directory as .Rmd (default)



Structure of a R Markdown

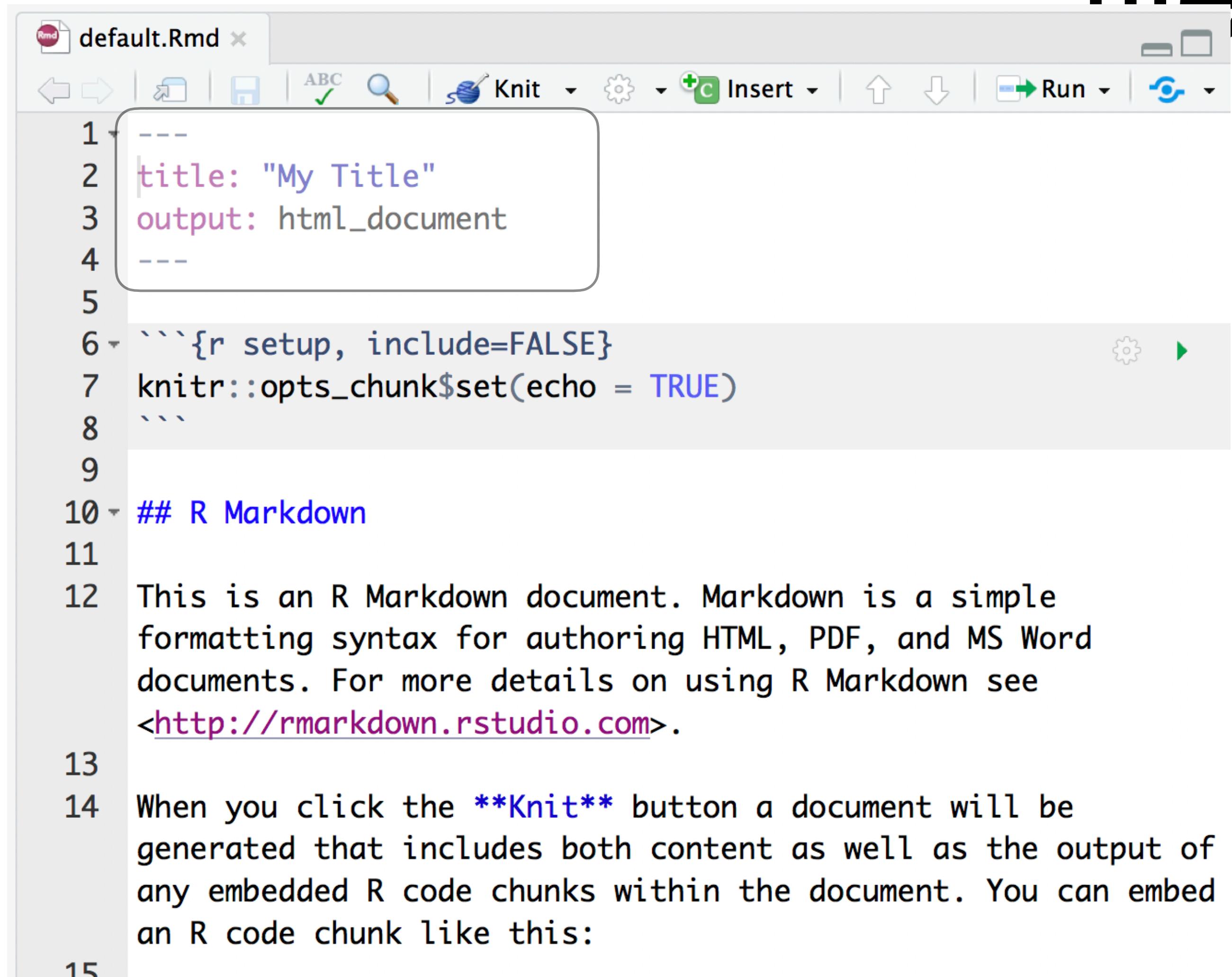
file

.Rmd extension



```
1 ---  
2 title: "My Title"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8 ```  
9  
10 ## R Markdown  
11  
12 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.  
13  
14 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:  
15
```

Structure of a R Markdown file



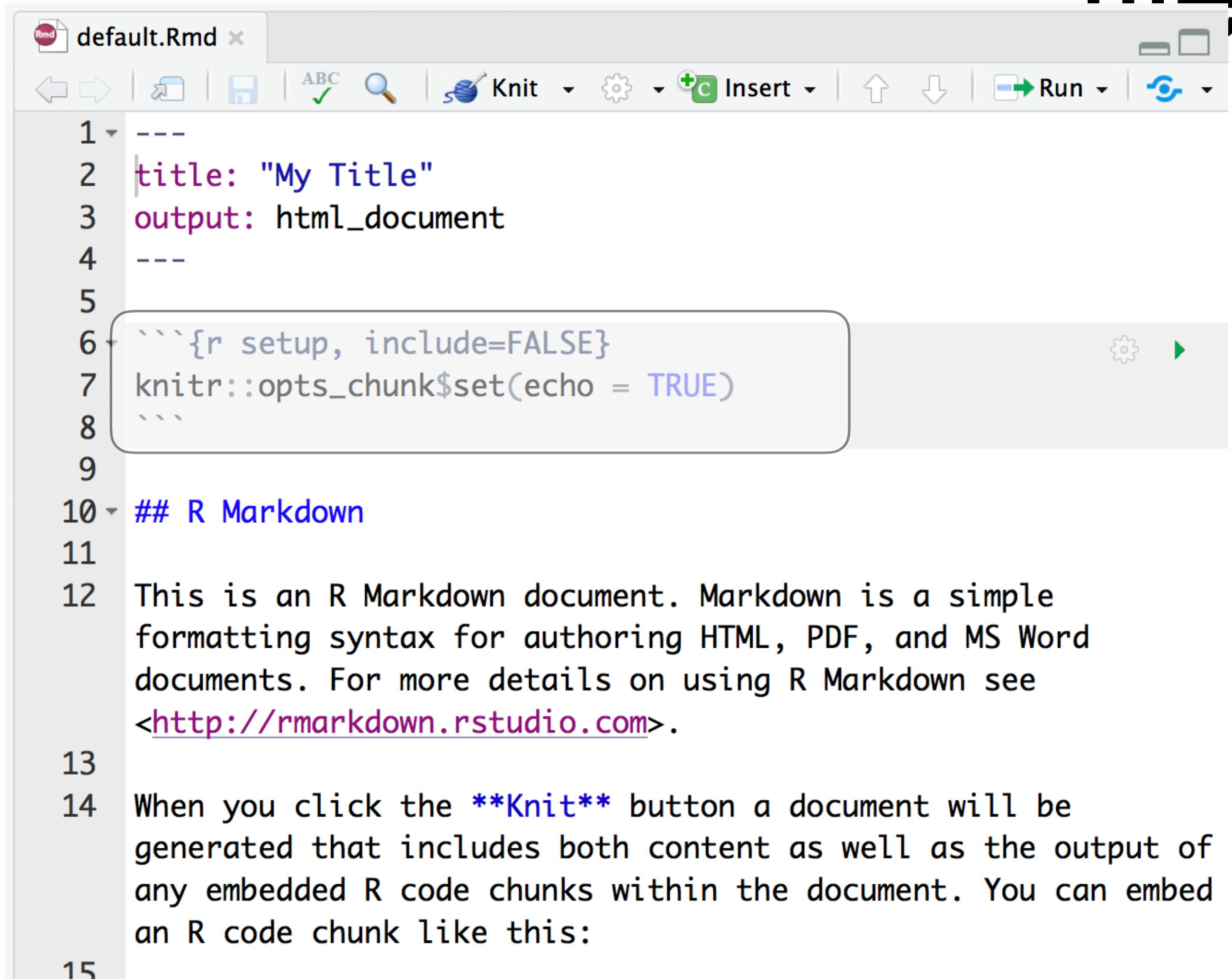
The screenshot shows the RStudio interface with a document titled "default.Rmd". The code editor displays the following content:

```
1 ---  
2 title: "My Title"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8 ```  
9  
10 ## R Markdown  
11  
12 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.  
13  
14 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:  
15
```

The "Knit" button in the toolbar is highlighted with a green checkmark.

Header ---

Structure of a R Markdown file



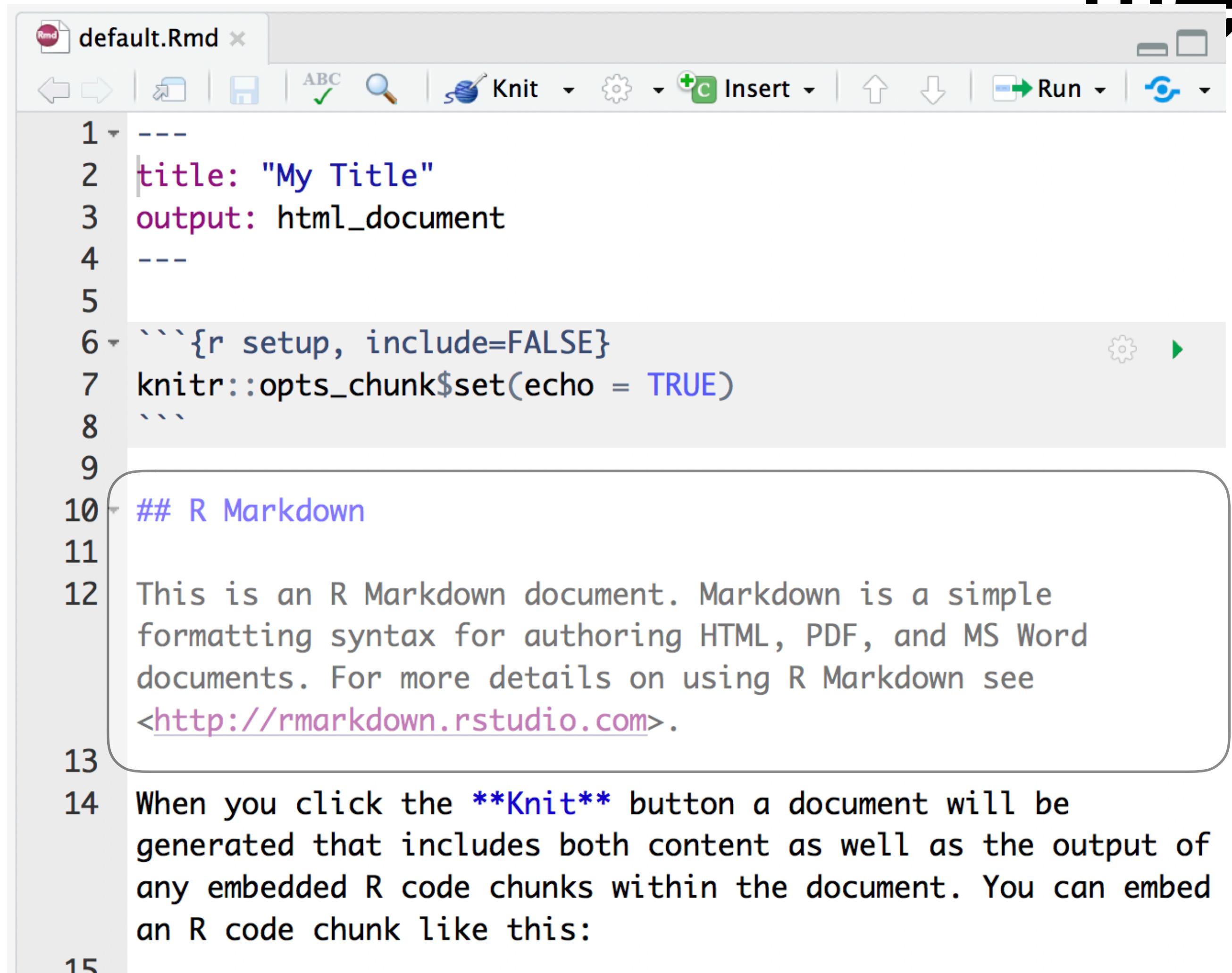
```
default.Rmd x
ABC Knit Insert Run

1 ---  
2 title: "My Title"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8 ```  
9  
10 ## R Markdown  
11  
12 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.  
13  
14 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:  
15
```

Code chunks ````{r}`

...

Structure of a R Markdown file



The screenshot shows the RStudio interface with a file named "default.Rmd" open. The code editor displays the following R Markdown structure:

```
1 ---  
2 title: "My Title"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8 ```  
9  
10 ## R Markdown  
11  
12 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.  
13  
14 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:  
15
```

The code editor has a toolbar with various icons for file operations, search, and knit/run. A tooltip or callout box highlights the "Text" section (lines 10-13) with the text "Text".

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

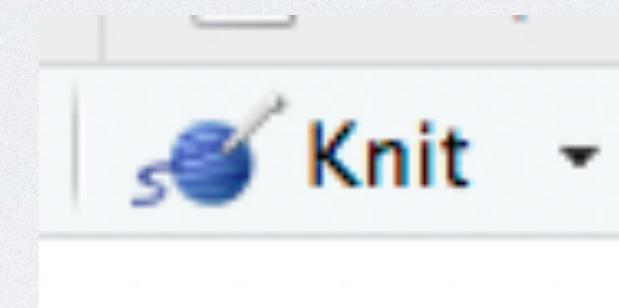
`results = "hide"`

Results should not be included in document

Your Turn 1

Open 06-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

01_code-chunks.Rmd

ABC Knit Insert Run

```
1 ---  
2 title: New Zealand  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 country_name <- "New Zealand"  
8 ```  
9  
10 ```{r load-packages, message = FALSE}  
11 library(tidyverse)  
12 library(gapminder)  
13 ```  
14  
15 ```{r filter-data}  
16 country_data <- filter(gapminder, country == country_name)  
17 ```  
18  
19 ```{r display-data, echo = FALSE}  
20 country_data  
21 ```  
22
```

Suppress messages in the output

R Markdown :: CHEAT SHEET

What is R Markdown?

.Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce what you along with the narration that a reader needs to understand your work.

Reproducible Research - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow



- Open a new .Rmd file at File ► New File ► R Markdown. Use the wizard that opens to pre-populate the file with a template
- Write document by editing template
- Knit document to create report; use knit button or render() to knit
- Preview Output in IDE window
- Publish (optional) to web server
- Examine build log in R Markdown console
- Use output file that is saved along side .Rmd

Embed code with knitr syntax

INLINE CODE
Insert with `r <code>`. Results appear as text without code.
Built with `r getRVersion()`

CODE CHUNKS
One or more lines surrounded with `r` and `r`. Place chunk options within curly braces, after r. Insert with
`r echo=TRUE`
`r getRVersion()`

GLOBAL OPTIONS
Set with knitr::opts_chunk\$set(), e.g.
`r include=FALSE`
`knitr::opts_chunk\$set(echo = TRUE)`

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)



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)



.rmd Structure

YAML Header
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML). At start of file Between lines of ---

Text
Narration formatted with markdown, mixed with:

Code Chunks
Chunks of embedded code. Each chunk: Begins with `r` ends with `r` R Markdown will run the code and append the results to the doc. It will use the location of the .Rmd file as the working directory

Parameters
Parameterize your documents to reuse with different inputs (e.g., data, values, etc.)

Interactive Documents
Turn your report into an interactive Shiny document in 4 steps

- Add runtime shiny to the YAML header.
- Call Shiny input functions to embed input objects.
- Call Shiny render functions to embed reactive output.
- Render with markdown::run or click Run Document in RStudio IDE

render

Use rmarkdown::render() to render/knit at cmd line. Important args:

| | | | | | |
|------------------------|--|-------------|--------------------------------|--|--------------------------|
| Input - file to render | output_options - List of render options (as in YAML) | output_file | params - list of params to use | envir - environment to evaluate code chunks in | encoding - of input file |
|------------------------|--|-------------|--------------------------------|--|--------------------------|

GLOBAL OPTIONS
Set with knitr::opts_chunk\$set(), e.g.
`r include=FALSE`
`knitr::opts_chunk\$set(echo = TRUE)`

How many cars?

```
output: html_document
runtime: shiny
...
<table>
<tr><td>5</td><td>speed dist</td></tr>
<tr><td>1</td><td>4.00 2.00</td></tr>
<tr><td>2</td><td>4.00 10.00</td></tr>
<tr><td>3</td><td>7.00 4.00</td></tr>
<tr><td>4</td><td>7.00 22.00</td></tr>
<tr><td>5</td><td>8.00 16.00</td></tr>
</table>
```

Shiny

Embed a complete app into your document with shiny::shinyAppDir()

NOTE: Your report will be rendered as a Shiny app, which means

More options:

error

fig.height, fig.width,
warning

eval

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/superscript2~2
~~strikethrough~~
escaped: `^_\\`|_
endash: --, emdash: —
equation: $A = \pi r^2$S
equation block:
E = mc2
```

block quote

```
# Header1 #anchor
## Header 2 {css_id}
### Header 3 {css_class}
#### Header 4
##### Header 5
#### Text comment-->
<!--Text ignored in HTML>
<em>HTML ignored in pdfs</em>
<http://www.rstudio.com>
<link>(www.rstudio.com)
Jump to [Header 1](#anchor)
image:
![(Caption)](smallorb.png)
```

list

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

Continued (indent 4 spaces)

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

1. Here is the footnote.[↔]

R Studio

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at [rmarkdown.rstudio.com](#) • rmarkdown 1.6

Set render options with YAML

When you render, R Markdown

1. runs the R code, embeds results and text into .md file with knitr
2. then converts the .md file into the finished format with pandoc

Set a document's default output format in the YAML header:

```
... output: html_document
... # Body
```

output value creates

output value	creates
html_document	html
pdf_document	pdf (requires Tex)
word_document	Microsoft Word (.docx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	Github compatible markdown
ioslides_presentation	ioslides HTML slides
slidy_presentation	slidy HTML slides
beamer_presentation	Beamer pdf slides (requires Tex)

Customize output with sub-options (listed to the right):

```
... output: html_document
... code_folding: hide
... toc_float: TRUE
... # Body
```

html tabs

Use tablet.css class to place sub-headers into tabs

```
# Tabset .tabset.tabset-fade.tabset-pills
## Tab 1
text 1
## Tab 2
text 2
### End tabset
```

Create a Reusable Template

1. Create a new package with a inst/markdown/templates directory
2. In the directory, Place a folder that contains: template.yaml (see below) skeleton.Rmd (contents of the template) any supporting files
3. Install the package
4. Access template in wizard at File ► New File ► R Markdown template.yaml

name: My Template

Table Suggestions

Several functions format R data into tables

Table with knitr	Table with xtable	Table with stargazer	Table with kable
knitr::kable(data, caption = "Table with kable")	print(xtable::xtable(data, caption = "Table with xtable"), type = "html", html.table.attributes = "border=0")	stargazer::stargazer(data, type = "html", title = "Table with stargazer")	knitr::kable(data, caption = "Table with stargazer")

Citations and Bibliographies

Create citations with .bib, .bibtex, .cops, .enl, .medline, .mods, .ns, .wos, and .xml files

1. Set bibliography file and CSL 1.0 Style file (optional) in the YAML header
2. Use citation keys in text

3. Render. Bibliography will be added to end of document

R

Caption

list

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

Continued (indent 4 spaces)

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

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.]

1. Here is the footnote.[↔]

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/superscript²~₂

strikethrough~~

escaped: `^_\\`|_

endash: --, emdash: —

equation: \$A = \pi r^2\$S

equation block:

$$E = mc^2$$

Plain text

End a line with two spaces to start a new paragraph.

italics and **bold**.

verbatim code

sub/superscript²~₂

strikethrough~~

escaped: `^_\\`|_

endash: --, emdash: —

equation: \$A = \pi r^2\$S

equation block:

$$E = mc^2$$

block quote

Header1

Header 2

Header 3

Header 4

Header 5

Header 6

<!--Text comment-->

<texbf{Tex ignored in HTML}>

HTML ignored in pdfs

<http://www.rstudio.com>

link

Jump to **Header 1** **(#anchor)**

image:

Caption

list

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

Continued (indent 4 spaces)

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

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.]

1. Here is the footnote.[↔]

CC BY Charlotte Wickham

Your Turn 2

1. Open 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



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

02_text-blocks.Rmd x

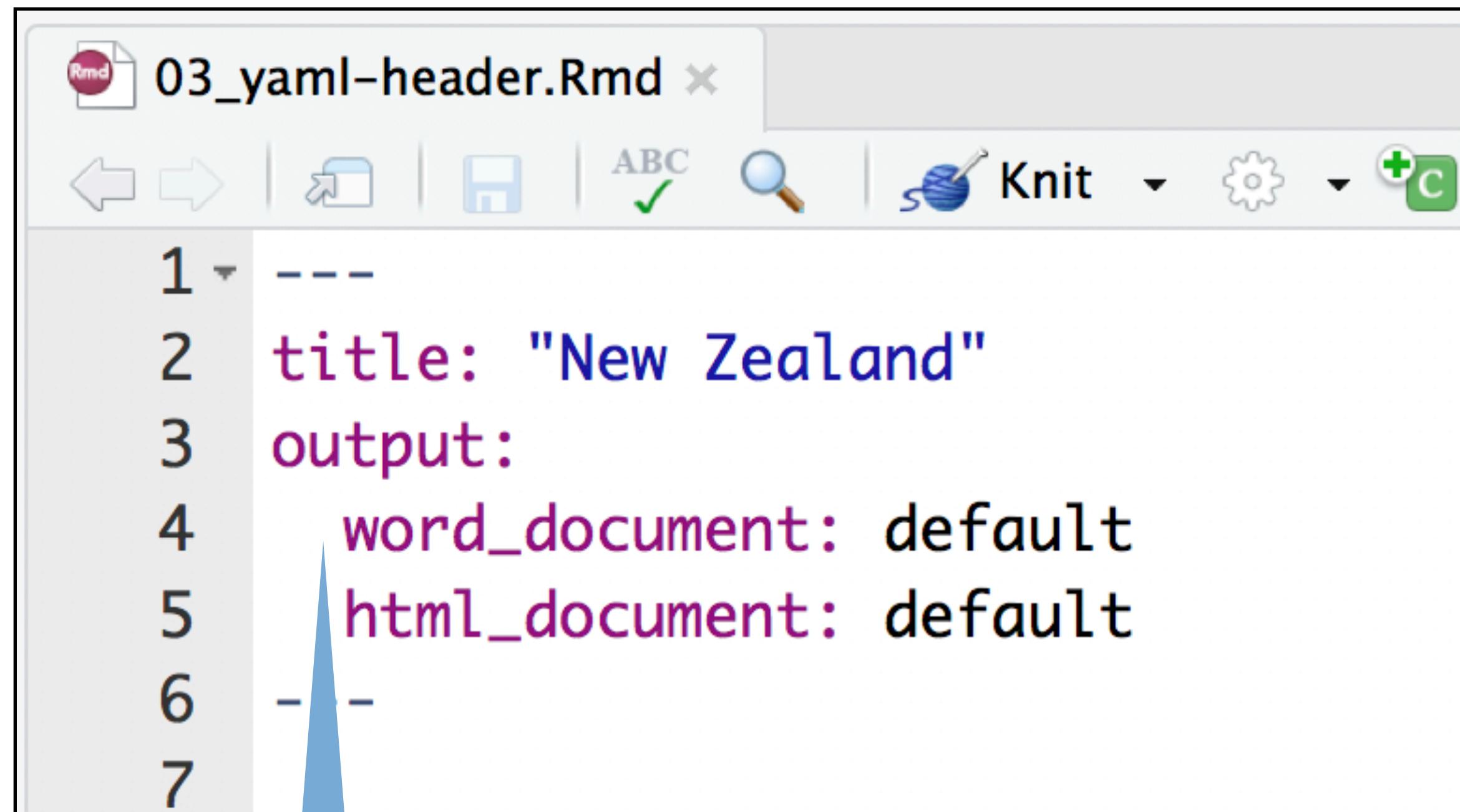
ABC Knit Insert Run

```
7 country_name <- "New Zealand"
8 ...
9
10```{r load-packages, message = FALSE}
11library(tidyverse)
12library(gapminder)
13...
14
15This report examines a subset of the `gapminder` data set.
In particular the data for `r country_name`.
16
17The report will examine:
18
19* life expectancy over time, and
20* the most recent value for life expectancy.
21
22```{r filter-data}
23country_data <- filter(gapminder,
24  country == country_name)
25...
26
27## Data
28
```

Your Turn 3

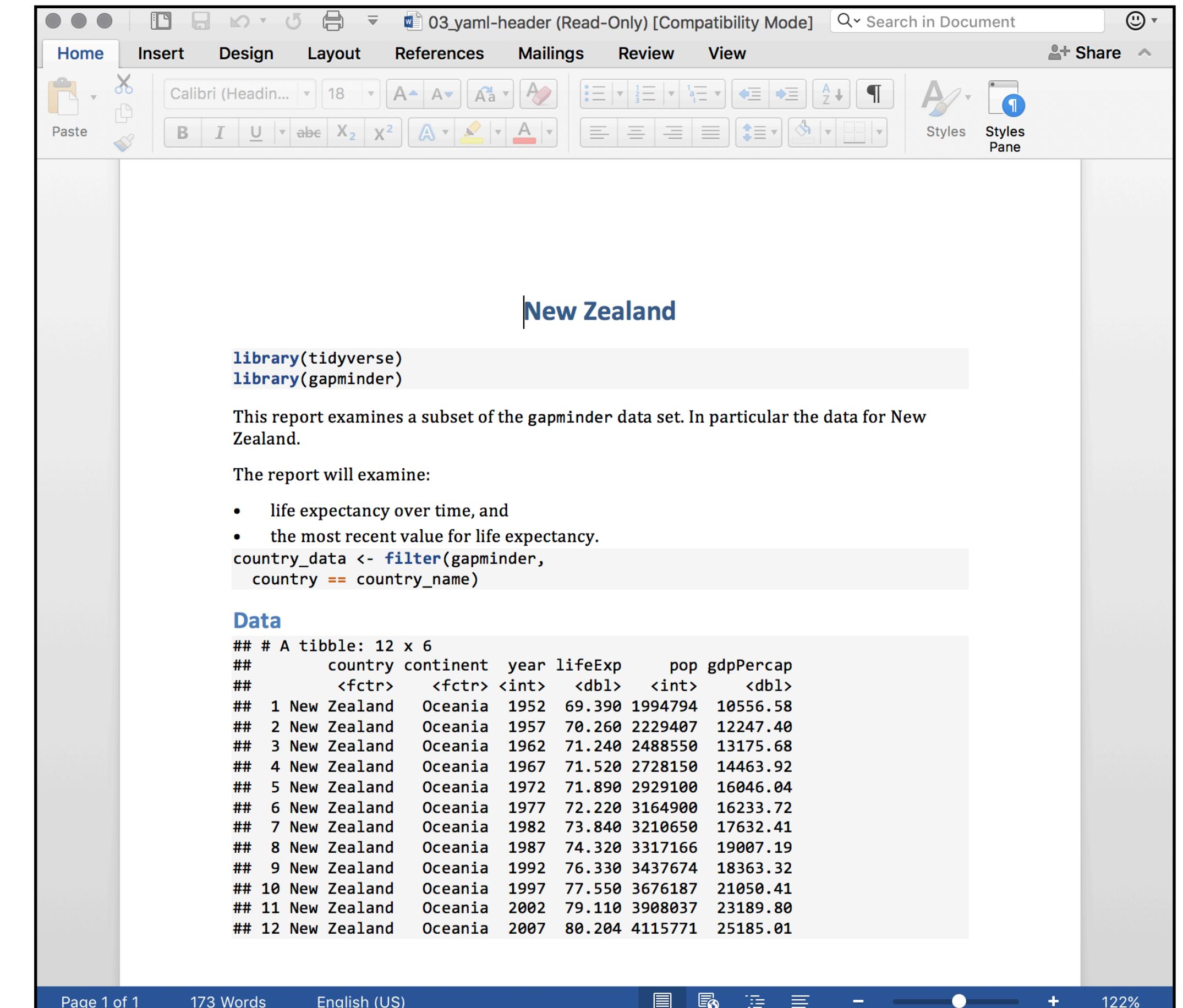
1. Open 03_yaml-header.Rmd
2. Knit
3. Knit -> Knit to Word
4. What changes in the .Rmd file?

03_yaml-header.docx



```
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



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

```
## # A tibble: 12 x 6  
##   country continent year lifeExp     pop gdpPercap  
##   <fctr>    <fctr> <int>   <dbl>   <dbl>      <dbl>  
## 1 New Zealand  Oceania  1952  69.390 1994794  10556.58  
## 2 New Zealand  Oceania  1957  70.260 2229407  12247.40  
## 3 New Zealand  Oceania  1962  71.240 2488550  13175.68  
## 4 New Zealand  Oceania  1967  71.520 2728150  14463.92  
## 5 New Zealand  Oceania  1972  71.890 2929100  16046.04  
## 6 New Zealand  Oceania  1977  72.220 3164900  16233.72  
## 7 New Zealand  Oceania  1982  73.840 3210650  17632.41  
## 8 New Zealand  Oceania  1987  74.320 3317166  19007.19  
## 9 New Zealand  Oceania  1992  76.330 3437674  18363.32  
## 10 New Zealand Oceania  1997  77.550 3676187  21050.41  
## 11 New Zealand Oceania  2002  79.110 3908037  23189.80  
## 12 New Zealand Oceania  2007  80.204 4115771  25185.01
```

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
```

```
---
```



Output is
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
```

```
---
```

Two spaces

Output is
html_document with
default options

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

```
1 ---  
2 title: Canada  
3 output:  
4   html_document:  
5     toc: true  
6     df_print: paged  
7 ---  
8  
9 r setup, include=FALSE  
10 knitr::opts_chunk$set(echo = FALSE, message = FALSE)  
11 country_name <- "Canada"  
12 ````  
13  
14 This report examines a subset of the `gapminder` data set. In particular the data  
for `r country_name`.  
15
```



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
```

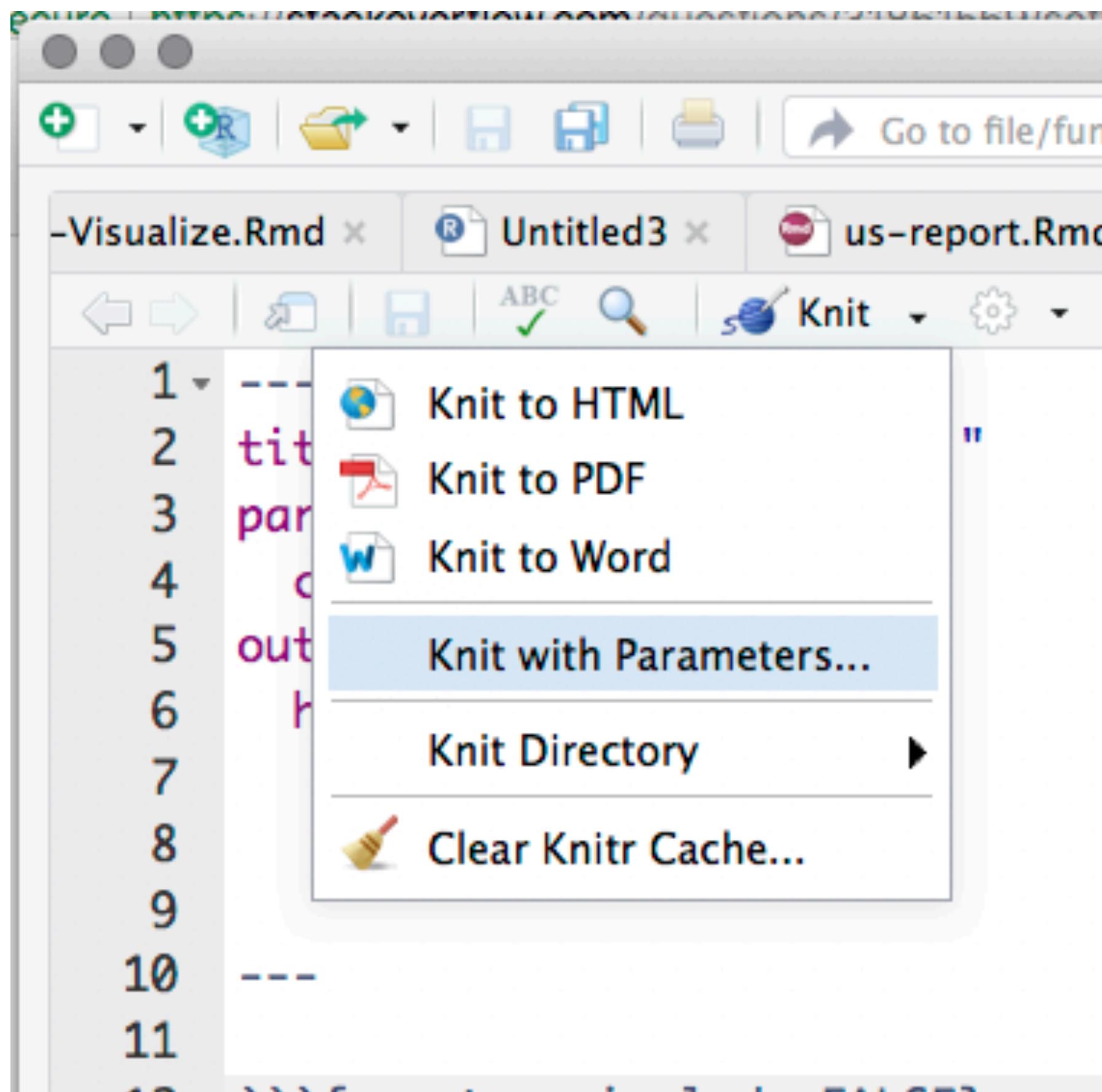
```
```
```

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

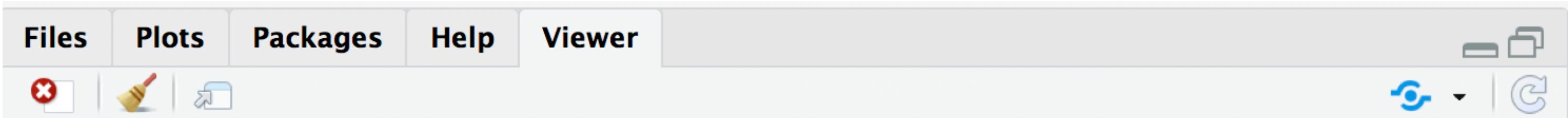


```
rmarkdown::render("06-rmarkdown/06_any-report.Rmd",  
  params = list(country = "Canada"))
```

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

Your Turn 6

Knit 06_any-report.Rmd with
country = "Spain"



Canada

- [Life Expectancy](#)
- [Data](#)

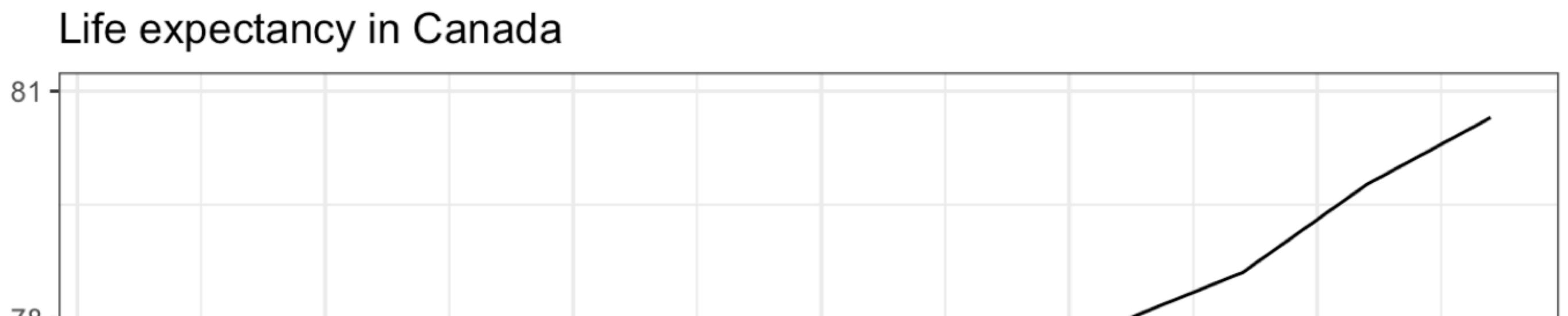
This report examines a subset of the `gapminder` data set. In particular the data for Canada.

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("06-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>

The image shows a stack of three overlapping browser tabs. The top tab is titled "New Zealand.html" and contains the text "New Zealand". Below it is a list: "• Life Expectancy" and "• Data". The middle tab is titled "Canada.html" and contains the text "Canada". Below it is a list: "• Life Expectancy" and "• Data". The bottom tab is titled "United States.html" and contains the text "United States". Below it is a list: "• Life Expectancy" and "• Data". Each tab also has a "Life Expectancy" section below it, which includes a chart showing life expectancy over time.

New Zealand.html

New Zealand

- Life Expectancy
- Data

Canada.html

Canada

- Life Expectancy
- Data

United States.html

United States

- Life Expectancy
- Data

Life Expectancy

United States had a life expectancy of 78.2 in 2010

Life expectancy in United States

Workflow

Two common uses:

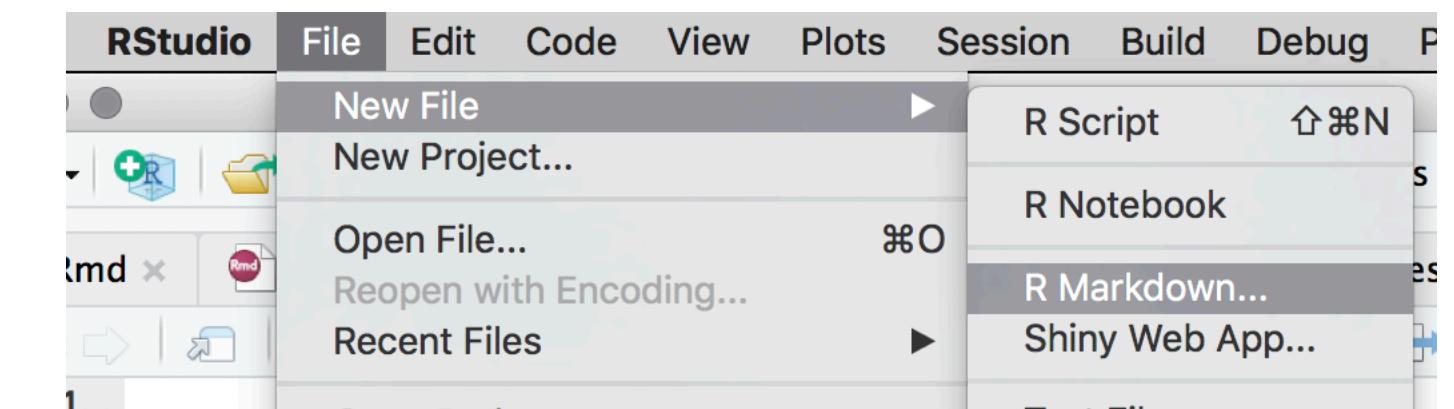
1. Use Rmarkdown like a data analysis log

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 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](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

