

# 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

.Rmd



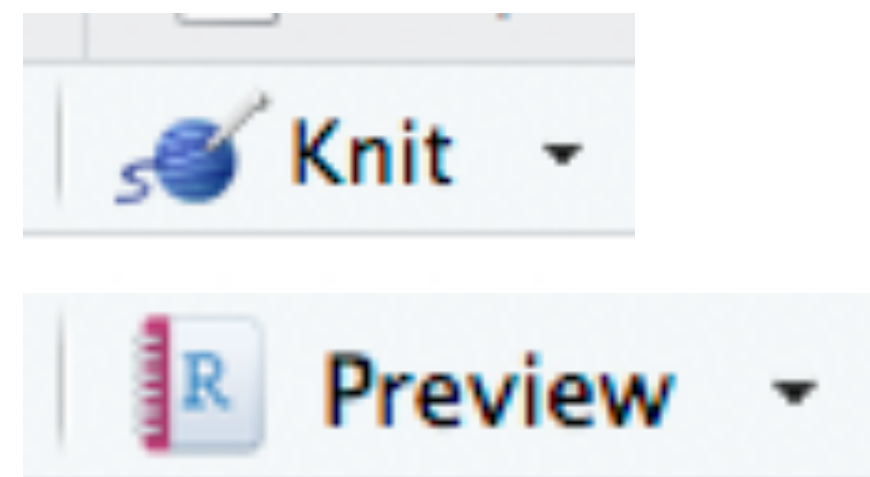
A plain text file, starts with a header, contains R in code chunks, and text.

```
1- ---
2- title: "Viridis Demo"
3- output: html_document
4- ---
5-
6- ```{r include = FALSE}
7- library(viridis)
8- ```
9-
10- The code below demonstrates two color palettes in the
11- [viridis](https://github.com/sjmgarnier/viridis) package. Each
12- plot displays a contour map of the Maunga Whau volcano in
13- Auckland, New Zealand.
14-
15- ## Viridis colors
16-
17- ```{r}
18- image(volcano, col = viridis(200))
19- ```
20-
21- ## Magma colors
22-
23- ```{r}
24- image(volcano, col = viridis(200, option = "A"))
25- ```
```

# 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)



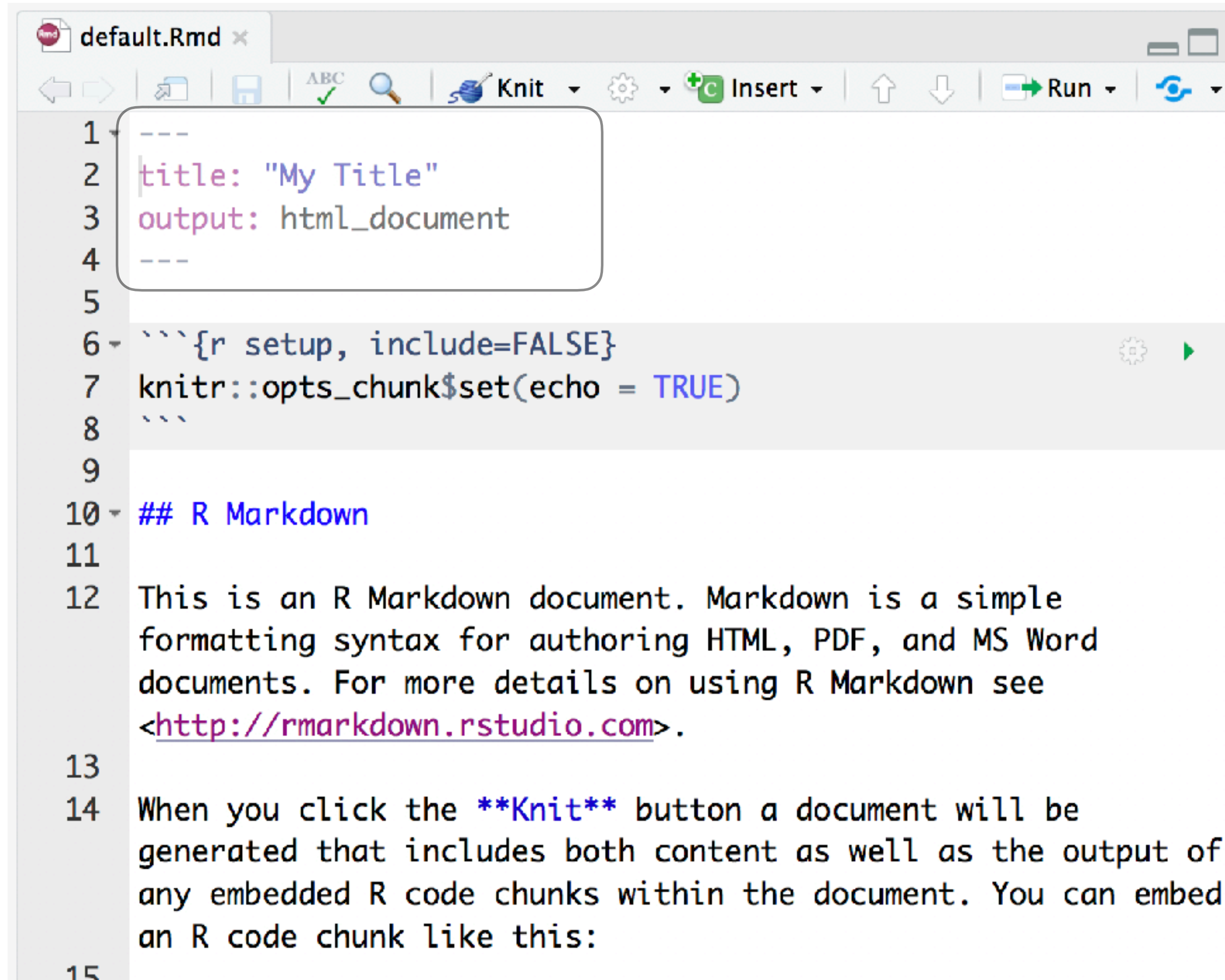


# Structure of a R Markdown file

.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
13 formatting syntax for authoring HTML, PDF, and MS Word
14 documents. For more details on using R Markdown see
15 <http://rmarkdown.rstudio.com>.
16
17 When you click the Knit button a document will be
18 generated that includes both content as well as the output of
19 any embedded R code chunks within the document. You can embed
20 an R code chunk like this:
```

# Structure of a R Markdown file

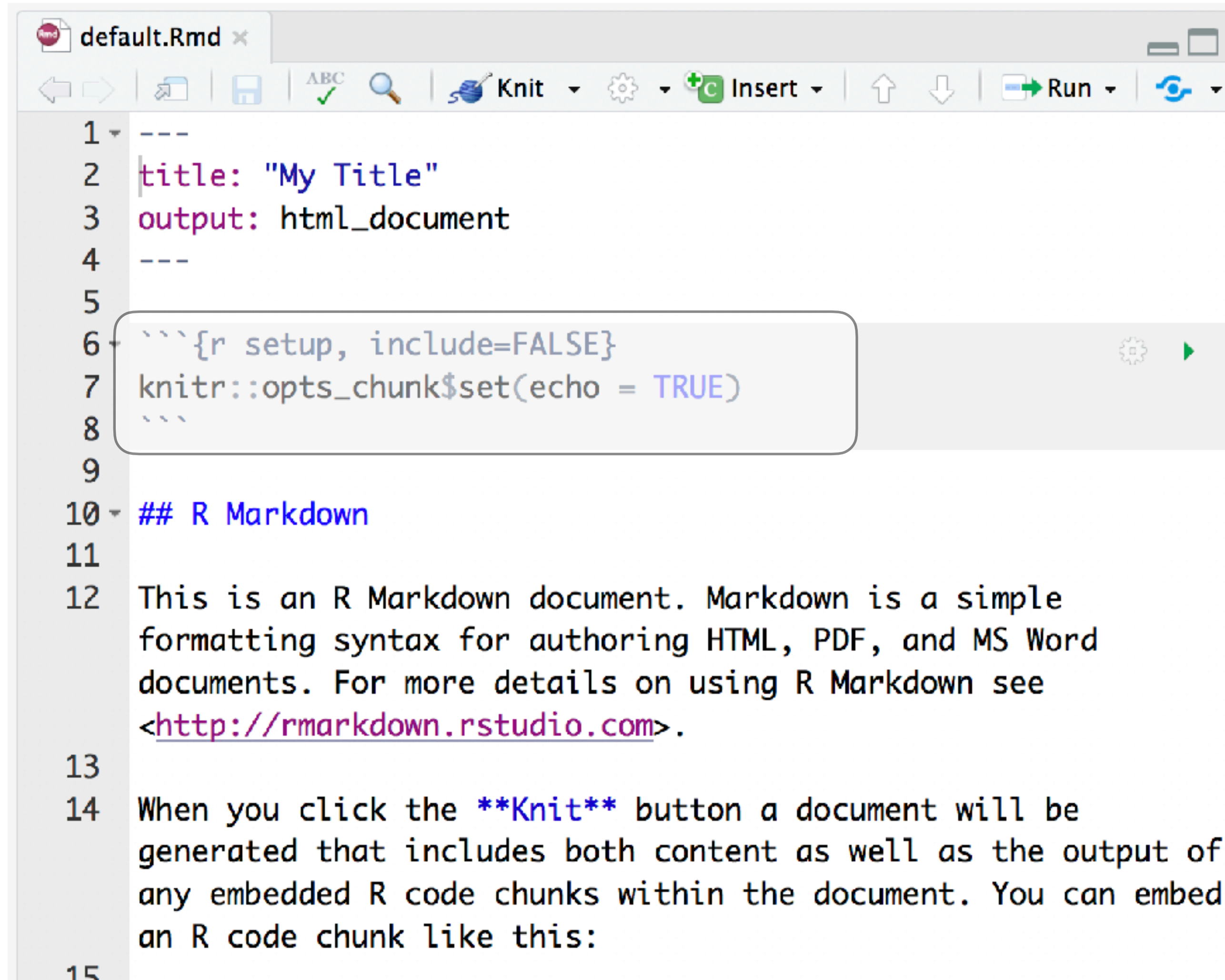


```
1 ---  
2 title: "My Title"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8 ```  
9  
10 ## R Markdown  
11  
12 This is an R Markdown document. Markdown is a simple  
13 formatting syntax for authoring HTML, PDF, and MS Word  
14 documents. For more details on using R Markdown see  
15 <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:
```

Header ---  
---



# Structure of a R Markdown file



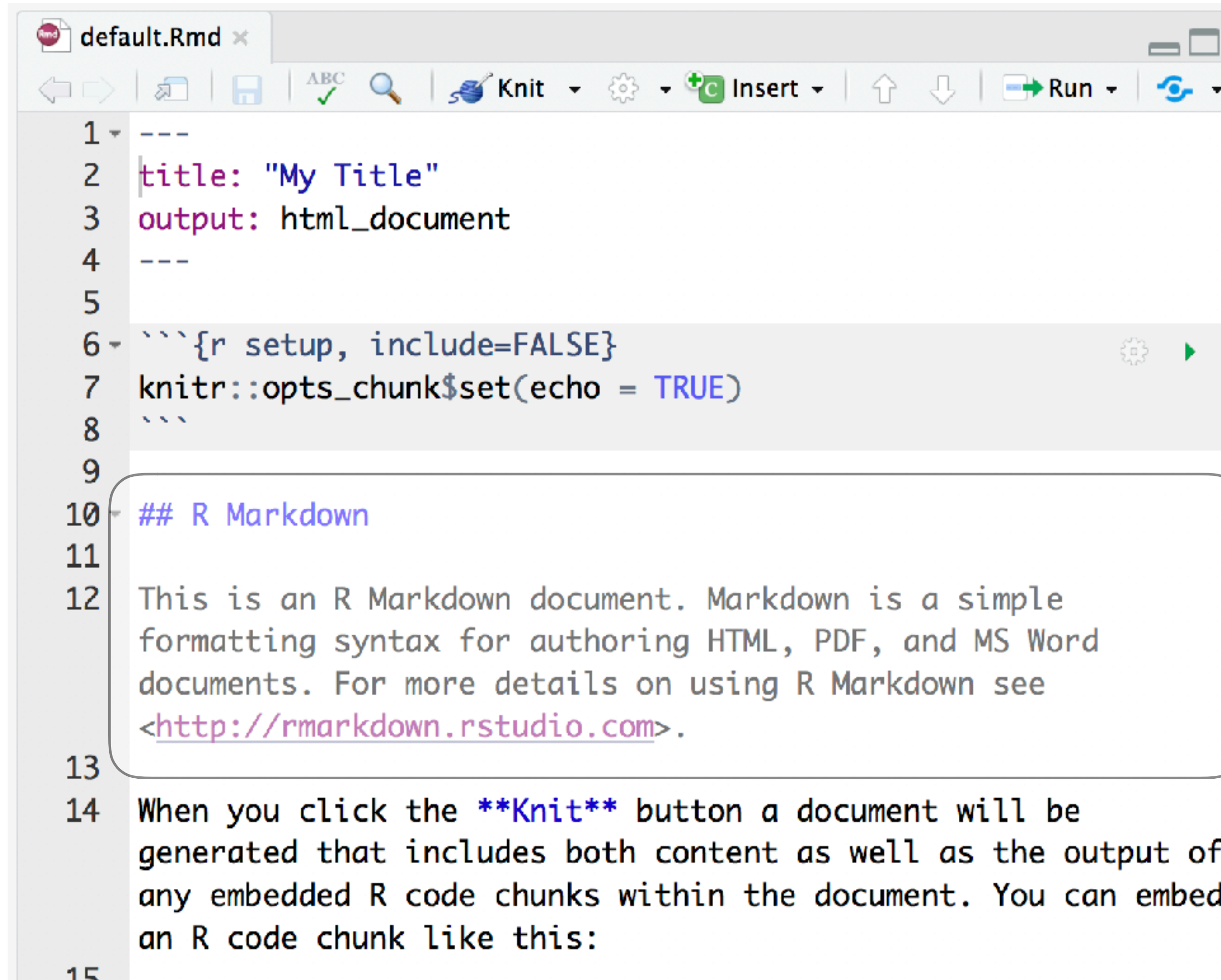
The screenshot shows a text editor window titled 'default.Rmd'. The editor contains the following content:

```
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  
13 formatting syntax for authoring HTML, PDF, and MS Word  
14 documents. For more details on using R Markdown see  
15 <http://rmarkdown.rstudio.com>.
```

A callout box highlights the R code chunk starting at line 6, showing the syntax for setting options like `knitr::opts_chunk$set(echo = TRUE)`.

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

# Structure of a R Markdown file



The screenshot shows a text editor window titled 'default.Rmd'. The editor contains 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  
13 formatting syntax for authoring HTML, PDF, and MS Word  
14 documents. For more details on using R Markdown see  
15 <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

01\_code-chunks.Rmd

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 your work 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 document, you can reuse 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 R Markdown documents, HTML or PDF based slides, Notebooks, and more.

## Workflow

- 
1. Open a new Rmd file in File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.
  2. Write document by editing template.
  3. Knit document to create report, use knit button or render() in knitr.
  4. Preview Output in PDF window.
  5. Publish (optional) in web viewer.
  6. Examine build log in R Markdown console.
  7. Use output file that is saved along side Rmd.

## render

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

|                               |                                                    |                                  |                                |                                       |                                                       |                                 |
|-------------------------------|----------------------------------------------------|----------------------------------|--------------------------------|---------------------------------------|-------------------------------------------------------|---------------------------------|
| <b>input</b> - file to render | <b>output_format</b> - output options (as in YAML) | <b>output_file</b> - output file | <b>output_dir</b> - output dir | <b>params</b> - list of params to use | <b>envir</b> - environment to evaluate code chunks in | <b>encoding</b> - of input file |
|-------------------------------|----------------------------------------------------|----------------------------------|--------------------------------|---------------------------------------|-------------------------------------------------------|---------------------------------|

## Embed code with knitr syntax

**INLINE CODE**  
Insert with `<code>`. Results appear as text without code.  
Build with `<pre>Rversion()</pre>` → Build with `R.2.8`

**CODE CHUNKS**  
One or more lines surrounded with `***` (x) and `***`. Place chunk options within curly braces, after `x`. Insert with `<pre>*** {r echo=TRUE} getRversion()</pre>` → `<pre>getRversion()</pre>`

**GLOBAL OPTIONS**  
Set with `knitr::opts_knit$set()`, e.g. `*** {r echo=FALSE} knitr::opts_chunk$set(echo = TRUE)</pre>`

**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)

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

**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 header (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 `***` (x) ends with `***`

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.)
1. Add parameters - Create and set parameters in the header as sub-values of params
  2. Call parameters - Call parameter values in code as `params$parameter`
  3. Set parameters - Set values with `knitr::opts_knit$set()` with parameters as the params argument of render()
- render("doc.Rmd", params = list(x = 1, d = as.Date("2015-01-01")))

## Interactive Documents

Turn your report into an interactive Shiny document (HTML app)

1. Add `runShinyApp()` to the YAML header.
2. Call Shiny input functions to embed input objects.
3. Call Shiny render functions to embed results for output.
4. Render with `rmarkdown::render()` or click Run Document in RStudio IDE

output: html\_document  
runtime: shiny

```
*** {r echo=FALSE}  
numericInput("n",  
  "How many cars?", 5)  
  
renderTable({  
  head(mtcars, input$n)  
})
```

How many cars?

5

| speed | dist  |
|-------|-------|
| 1 4.9 | 2.00  |
| 2 4.9 | 10.00 |
| 3 7.0 | 4.00  |
| 4 7.0 | 22.00 |
| 5 8.0 | 16.00 |

Embed a complete app into your document with `runShinyAppDir()`

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



## 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
```







# 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

## 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

KnitInsertRun

country\_name <- "New Zealand"

library(tidyverse)  
library(gapminder)

This report examines a subset of the `gapminder` data set.  
In particular the data for `country\_name`.

The report will examine:  
  
\* life expectancy over time, and  
\* the most recent value for life expectancy.

country\_data <- filter(gapminder,  
 country == country\_name)

## Data

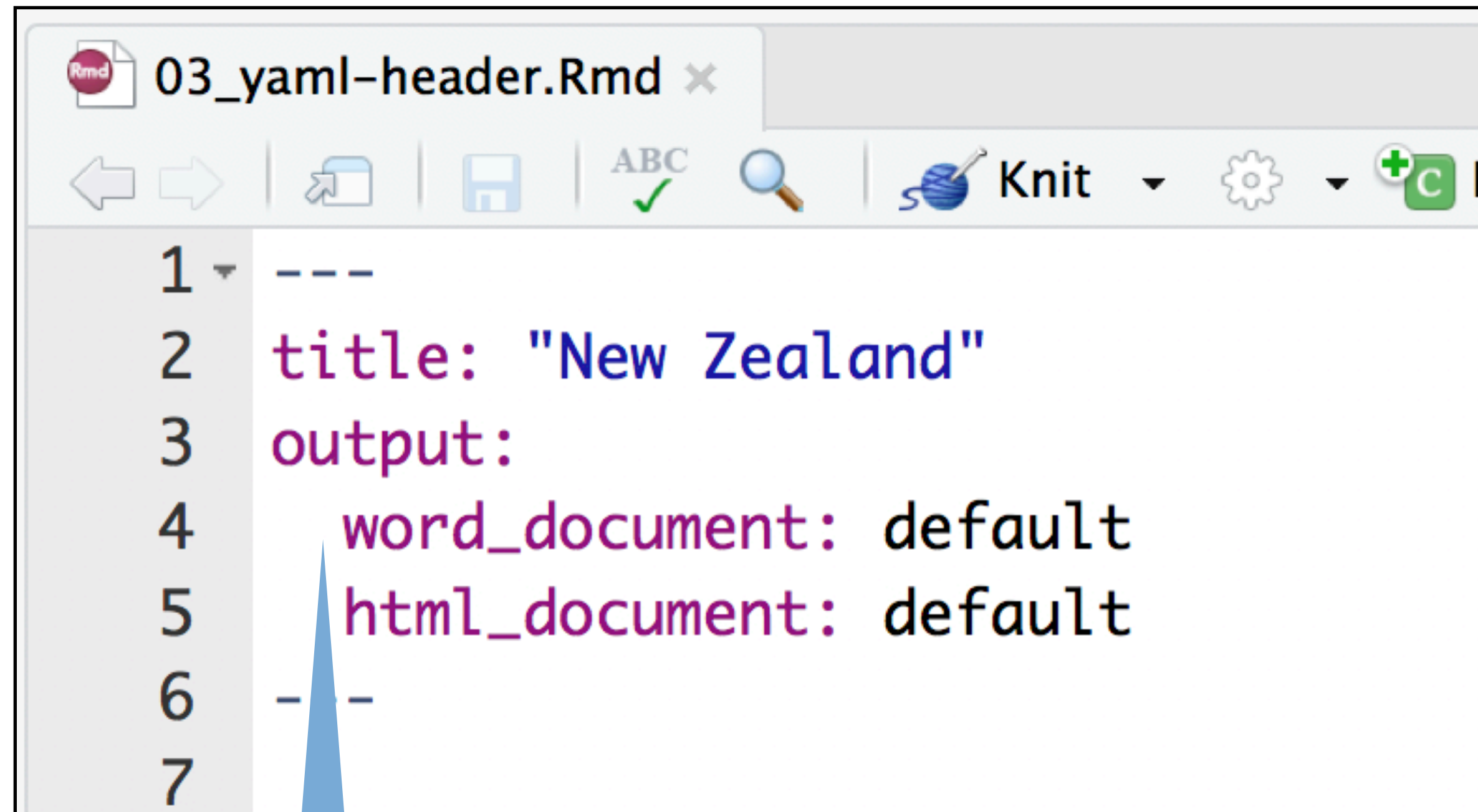


# Your Turn 3

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

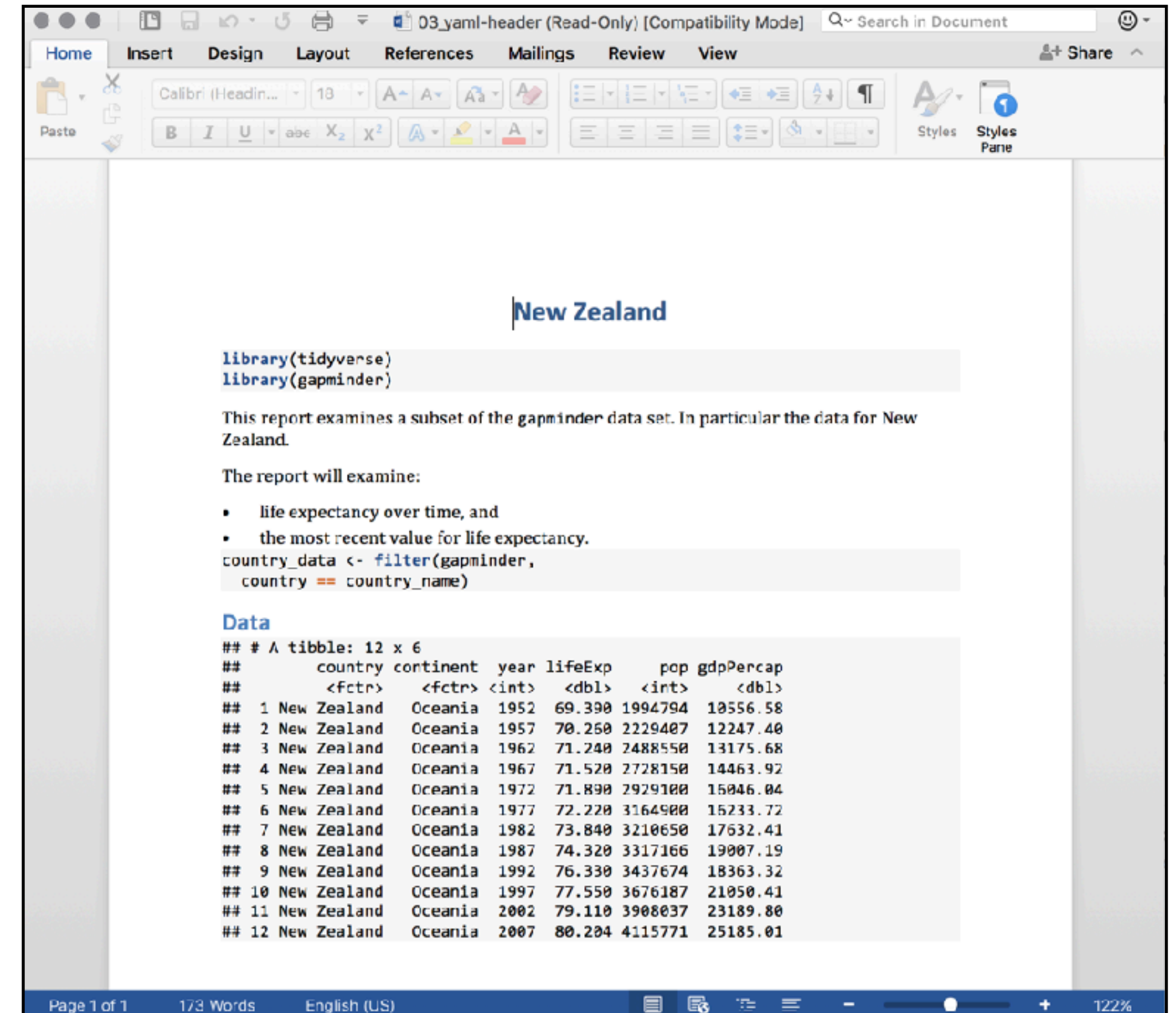


## 03\_yaml-header.docx



```
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> <int>    <dbl>
## 1 New Zealand Oceania 1952  69.390 1994794 10556.58
## 2 New Zealand Oceania 1957  70.250 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 15046.04
## 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  
---
```

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



```
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
15 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
```
```

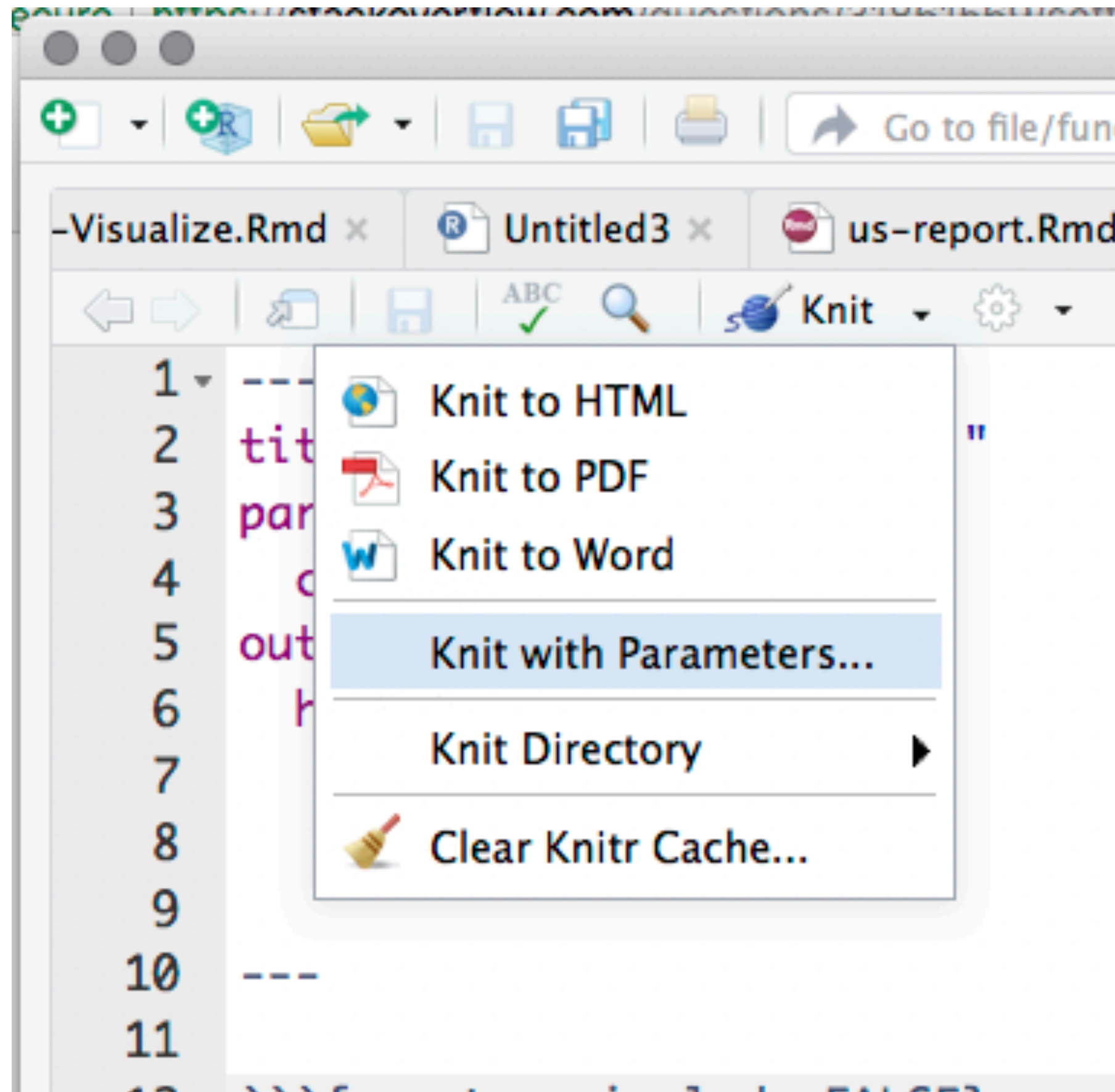
**Add your own  
parameters under the  
params option**

**Access the parameter values with  
params\$name**



# Knitting a parameterized report

Knit with Parameters under Knit menu



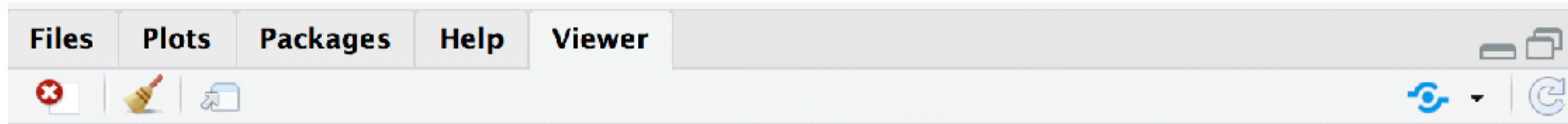
Or use render() function

```
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"`



# Canada

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

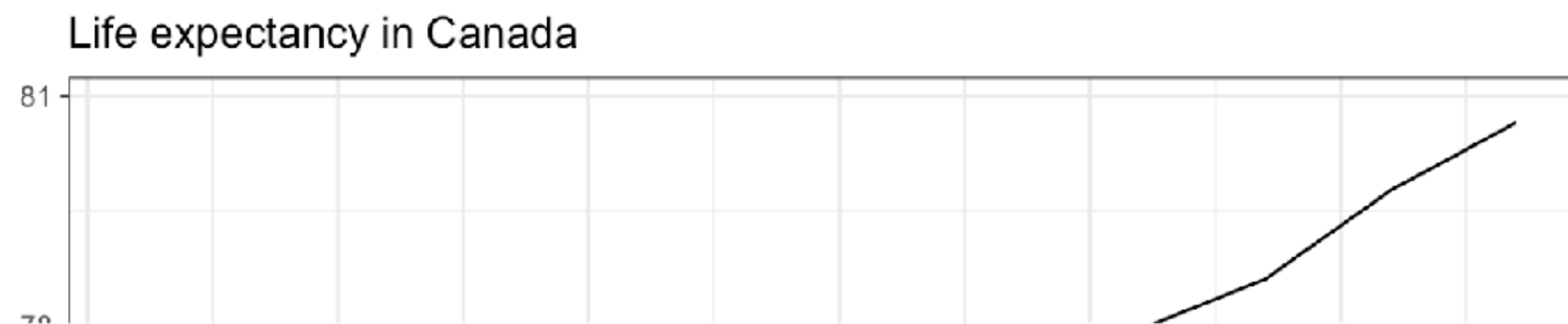
This report examines a subset of the `gapminder` data set. In particular the data for Canada.

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>

New Zealand.html

## New Zealand

- Life Expectancy
- Data

Life Canada.html

## Canada

- Life Expectancy
- Data

United States.html

Life

Canac

## United States

- Life Expectancy
- Data

### Life Expectancy

United States had a life expectancy of 78.2 in 20

Life expectancy in United States

|  |  |  |  |  |  |
|--|--|--|--|--|--|
|  |  |  |  |  |  |
|--|--|--|--|--|--|

# Workflow

Two common uses:

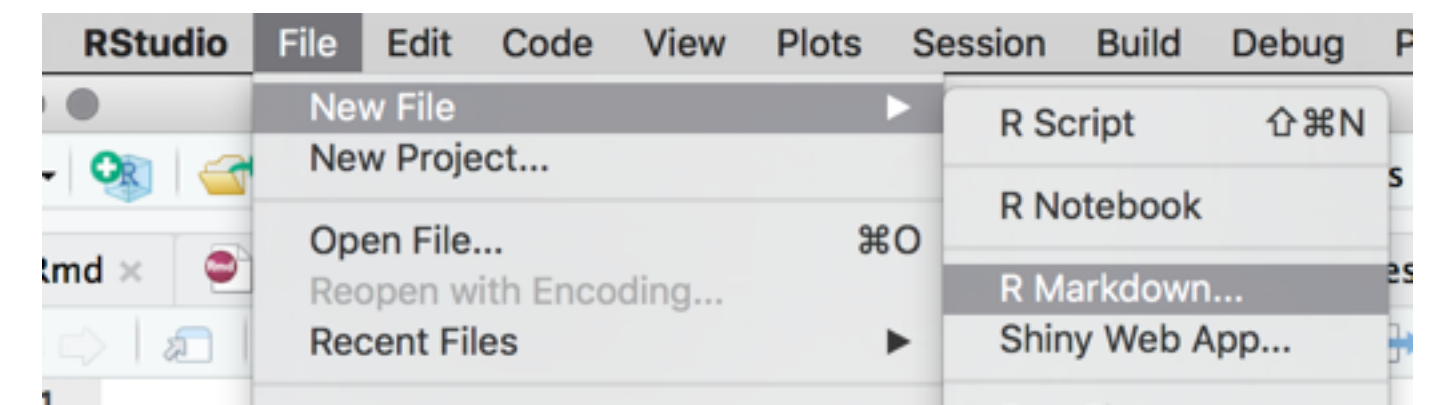
1. Use Rmarkdown like a data analysis log

Build up text and chunks as you work through 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)

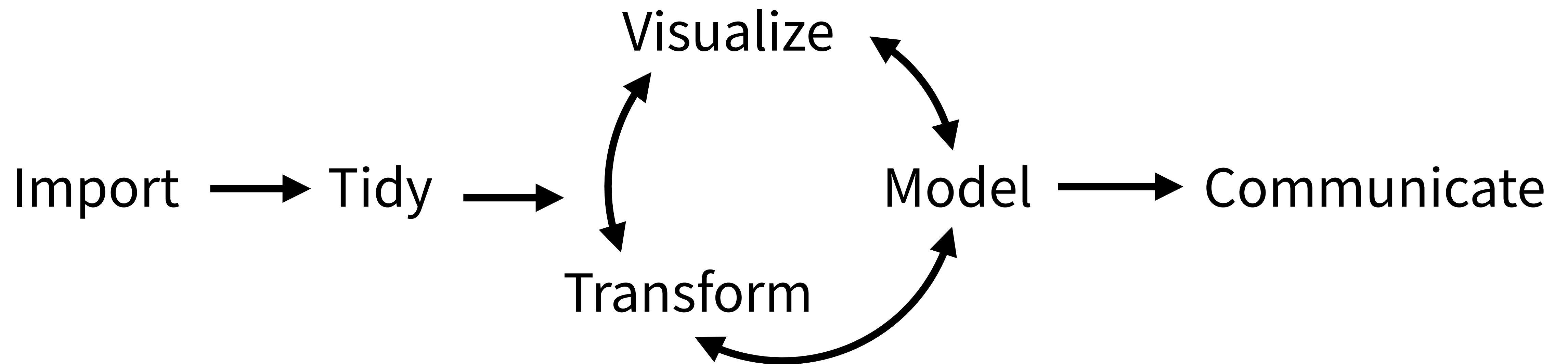
You can have other languages in code chunks, e.g.

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



Program

