

Reproducible Reports with R Markdown

Jessica Minnier, PhD & Meike Niederhausen, PhD
OCTRI Biostatistics, Epidemiology, Research & Design (BERD) Workshop

2019/07/18

 slides: bit.ly/berd_rmd
 pdf: bit.ly/berd_rmd_pdf

Load files for today's workshop

- Open slides bit.ly/berd_rmd
- Get project folder
 - Download zip folder at bit.ly/berd_rmd_zip
 - UNZIP completely
 - Open unzipped folder
 - Open `berd_rmarkdown_project.Rproj`
 - Inside RStudio, click on file `00-install.R` and click "Run"
 - Or open RStudio Cloud project (need to make an account) at bit.ly/berd_rmd_rc
 - click "Save a permanent copy"



Allison Horst

Learning objectives

- Understand how to use literate programming for reproducible research
- Basics of Markdown language
- Learn how to create R Markdown files with code and markdown text
- Turn R Markdown files into html, pdf, Word, or presentation files
- Learn about reproducible project workflows
- (If time allows) Learn some additional R Markdown tips

Why Reproducibility?

- Evidence your results are correct.
- Allow others to use our methods and results.

"An article about computational results is advertising, not scholarship. The actual scholarship is the full software environment, code and data, that produced the result."

-- (Claerbout and Karrenbach 1992)

Your closest collaborator is you six months ago, but you don't reply to emails.

-- @gonuke, quoting @mtholder

Types of Reproducibility

- **Computational reproducibility:** detailed information is provided about
 - *code, software, hardware and implementation details.*
- **Empirical reproducibility:** detailed information is provided about
 - *non-computational empirical scientific experiments and observations [data].*
- **Statistical reproducibility:** detailed information is provided about
 - *the choice of statistical tests, model parameters, threshold values, etc.*

[R OpenSci Reproducibility Guide](#)

Software tool for reproducibility: *Literate Programming*

"These tools enable writing and publishing **self-contained documents that include narrative and code used to generate both text and graphical results.**

In the R ecosystem, knitr [R markdown] and its ancestor Sweave used with RStudio are the main tools for literate computing. Markdown or LaTeX are used for writing the narrative, with chunks of R code sprinkled throughout the narrative. IPython is a popular related system for the Python language, providing an interactive notebook for browser-based literate computing."

[R OpenSci Reproducibility Guide](#)

R Markdown = .Rmd file = Code + text

`knitr` is a package that converts `.Rmd` files containing code + markdown syntax to a plain text `.md` markdown file, and then to other formats (html, pdf, Word, etc)

knitr converts .Rmd -> .md (behind the scenes)

The screenshot shows the RStudio Source Editor window. The title bar indicates the file is named 'slides_ex.Rmd'. The editor displays the following R Markdown code:

```
1 ---  
2 title: "Gapminder Report"  
3 author: "Your Name"  
4 date: "`r Sys.Date()`"  
5 output:  
6   html_document: default  
7   keep_md:true  
8 ---  
9  
10 ````{r setup, include=FALSE}  
11 knitr::opts_chunk$set(echo = TRUE)  
12 library(gapminder)  
13 library(nanar)  
14 library(tidyverse)  
15 ````  
16  
17 # Background  
18  
19 This is an analysis of the gapminder data set with `r nrow(gapminder)` observations.  
20  
21 ````{r datasummary}  
22 summary(gapminder)  
23 ````  
24  
25 # Analysis  
26  
27 ## GDP vs Life Expectancy  
28  
29 ````{r}  
30 ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +  
31   geom_point()  
32 ````  
33  
34
```

The status bar at the bottom shows the line number '34:1' and the text '# GDP vs Life Expectancy'. The 'R Markdown' tab is selected in the bottom right corner.

knitr converts .Rmd -> .md -> .html

The screenshot shows the RStudio Source Editor window. The title bar indicates the file is located at `~/Google Drive/BERD R Classes/berd_rmarkdown_project - master`. The tab bar shows the file is named `slides_ex.Rmd`. The toolbar includes standard options like Insert, Run, and Knit. The code editor displays an R Markdown document with the following content:

```
1 ---  
2 title: "Gapminder Report"  
3 author: "Your Name"  
4 date: "`r Sys.Date()`"  
5 output:  
6   html_document: default  
7 ---  
8  
9 ```{r setup, include=FALSE}  
10 knitr::opts_chunk$set(echo = TRUE)  
11 library(gapminder)  
12 library(naniar)  
13 library(tidyverse)  
14 ```  
15  
16 # Background  
17  
18 This is an analysis of the gapminder data set with `r nrow(gapminder)` observations.  
19  
20 ```{r datasummary}  
21 summary(gapminder)  
22 ```  
23  
24 # Analysis  
25  
26 ## GDP vs Life Expectancy  
27  
28 ```{r}  
29 ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +  
30   geom_point()  
31 ```  
32  
33
```

The status bar at the bottom shows the line number 6:7, the document title "# Gapminder Report", and the mode "R Markdown".

knitr converts .Rmd -> .md -> .pdf

```
~/Google Drive/BERD R Classes/berd_rmarkdown_project - master - RStudio Source Editor
(slides_ex.Rmd) | Insert | Run | Knit | ABC | Search | Help | RStudio Source Editor

1 ---  
2 title: "Gapminder Report"  
3 author: "Your Name"  
4 date: "r Sys.Date()"  
5 output:  
6   pdf_document: default  
---  
8  
9 ```{r setup, include=FALSE}  
10 knitr::opts_chunk$set(echo = TRUE)  
11 library(gapminder)  
12 library(nanar)  
13 library(tidyverse)  
14 ````  
15  
16 # Background  
17  
18 This is an analysis of the gapminder data set with `r nrow(gapminder)` observations.  
19  
20 ```{r datasummary}  
21 summary(gapminder)  
22 ````  
23  
24 # Analysis  
25  
26 ## GDP vs Life Expectancy  
27  
28 ```{r}  
29 ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +  
30   geom_point()  
31 ````  
32  
33
```

Gapminder Report
Your Name
2019-07-17

Background

Summary

This is an analysis of the gapminder data set with 1704 observations.

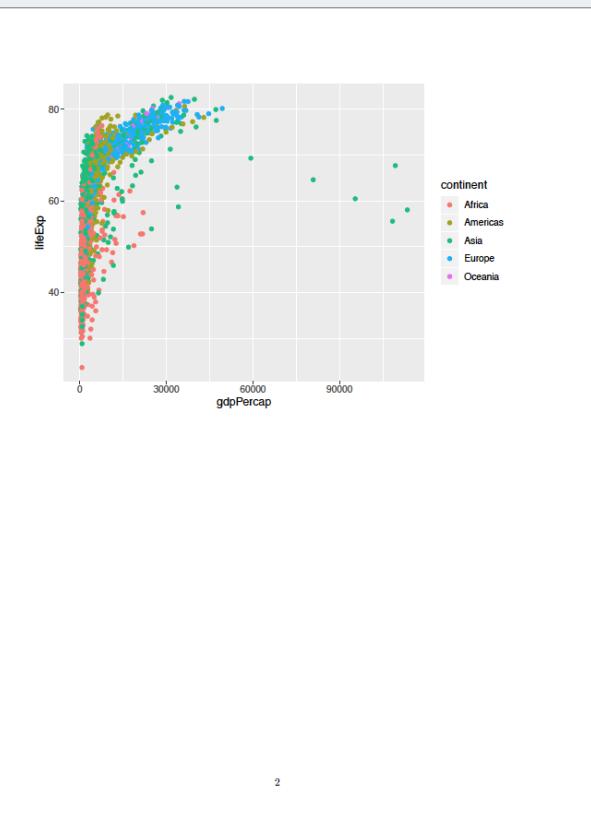
```
summary(gapminder)

##      country      continent       year     lifeExp
##  Afghanistan: 12 Africa :624 Min.  :1952   Min.  :23.60
##  Albania    : 12 Americas: 120 Median:1976   Median:34.70
##  Algeria    : 12 Asia   :396 Median:1950   Median:51.71
##  Angola     : 12 Europe  :360 Mean   :1980   Mean   :59.47
##  Argentina   : 12 Oceania: 24 3rd Qu.:1993  3rd Qu.:70.85
##  Australia  : 12 Max.   :2007   Max.   :82.60
##  (Other)    :1632
##      pop      gdpPercap
##  Min. :6.001e+04  Min.  : 241.2
##  1st Qu.:2.794e+06 1st Qu.: 1202.1
##  Median :7.026e+06  Median : 3551.1
##  3rd Qu.:2.465e+07 3rd Qu.: 9235.5
##  Max. :1.319e+09  Max.  :113523.1
##
```

Analysis

GDP vs Life Expectancy

```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point()
```



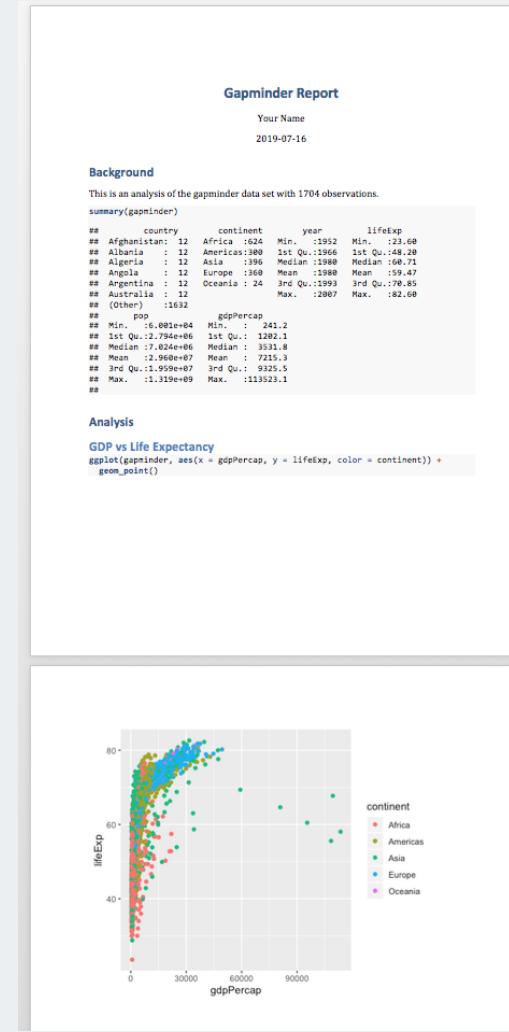
knitr converts .Rmd -> .md -> .doc

~/Google Drive/BERD R Classes/berd_rmarkdown_project - master - RStudio Source Editor

slides_ex.Rmd

```
1 ---
2 title: "Gapminder Report"
3 author: "Your Name"
4 date: `r Sys.Date()`
5 output:
6   word_document: default
7 ---
8
9 ```{r setup, include=FALSE}
10 knitr::opts_chunk$set(echo = TRUE)
11 library(gapminder)
12 library(naniar)
13 library(tidyverse)
14 ```
15
16 # Background
17
18 This is an analysis of the gapminder data set with `r nrow(gapminder)` observations.
19
20 ```{r datasummary}
21 summary(gapminder)
22 ```
23
24 # Analysis
25
26 ## GDP vs Life Expectancy
27
28 ```{r}
29 ggplot(gapminder, aes(x = gdpPerCap, y = lifeExp, color = continent)) +
30   geom_point()
31 ```
32
33
```

6:25 Gaptminder Report R Markdown



knitr converts .Rmd -> .md -> slides

```
~/Google Drive/BERD R Classes/berd_rmarkdown_project - master - RStudio Source Editor
(slides_ex.Rmd) [1] Insert | Run | Knit | ABC | Search | Help | Slides | R Markdown | Help | Help

1 ---
2 title: "Gapminder Report"
3 author: "Your Name"
4 date: "`r Sys.Date()`"
5 output:
6   ioslides_presentation
7 ---

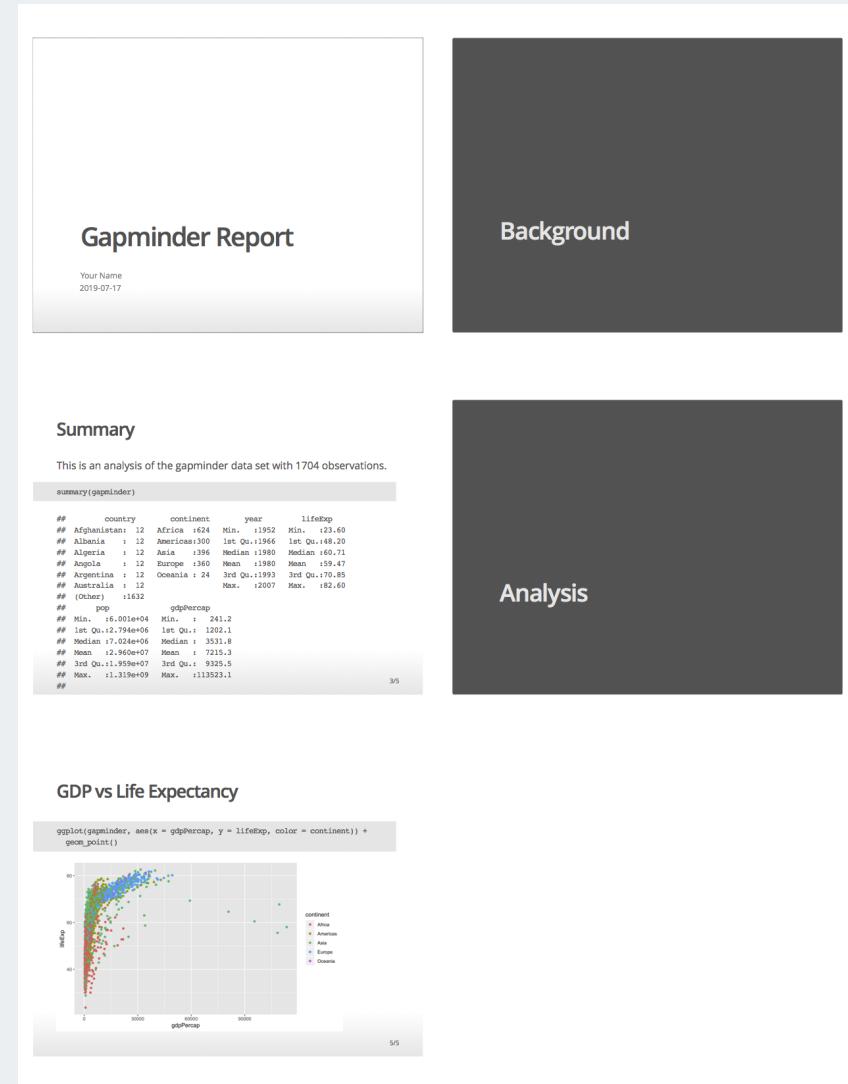
8
9 ````{r setup, include=FALSE}
10 knitr::opts_chunk$set(echo = TRUE)
11 library(gapminder)
12 library(nanar)
13 library(tidyverse)
14 ````

15
16 # Background
17
18 This is an analysis of the gapminder data set with `r nrow(gapminder)` observations.
19
20 ````{r datasummary}
21 summary(gapminder)
22 ````

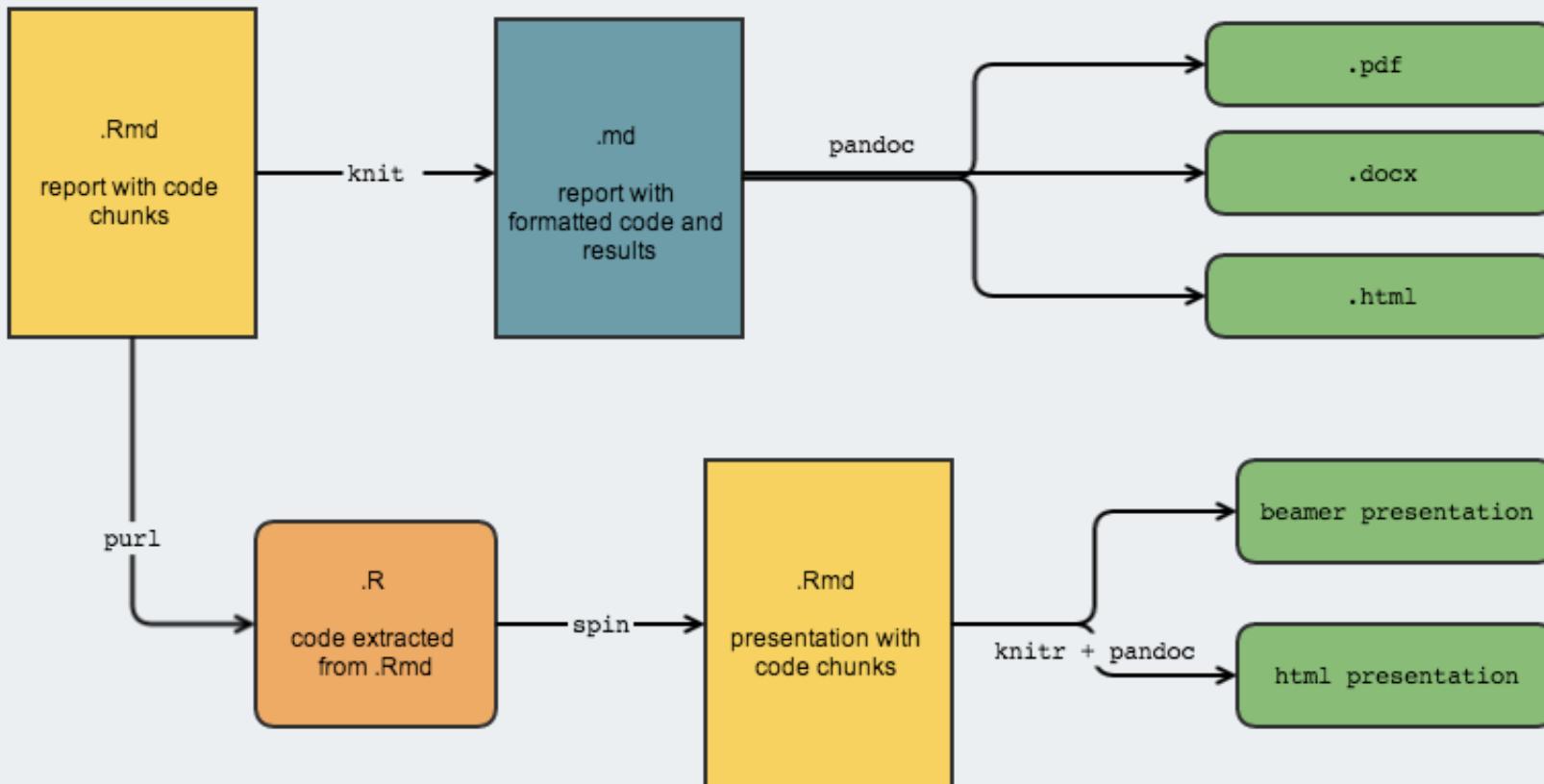
23
24 # Analysis
25
26 ## GDP vs Life Expectancy
27
28 ````{r}
29 ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +
30   geom_point()
31 ````

32
33
```

6:24 # Gapminder Report R Markdown



R Markdown vs. knitr::knit()



Michael Sachs

Good practices in RStudio

Use projects ([read this](#))

- Create an RStudio project for each data analysis project
- A project is associated with a directory folder
 - Sets *working directory*
 - Keep data files there
 - Keep scripts there; edit them, run them in bits or as a whole
 - Save your outputs (plots and cleaned data) there
- Only use relative paths, never absolute paths
 - relative (good): `read_csv("data/mydata.csv")`
 - absolute (bad): `read_csv("/home/yourname/Documents/stuff/mydata.csv")`

Advantages of using projects

- standardize file paths
- keep everything together
- a whole folder can be shared and run on another computer

Basic R Markdown example



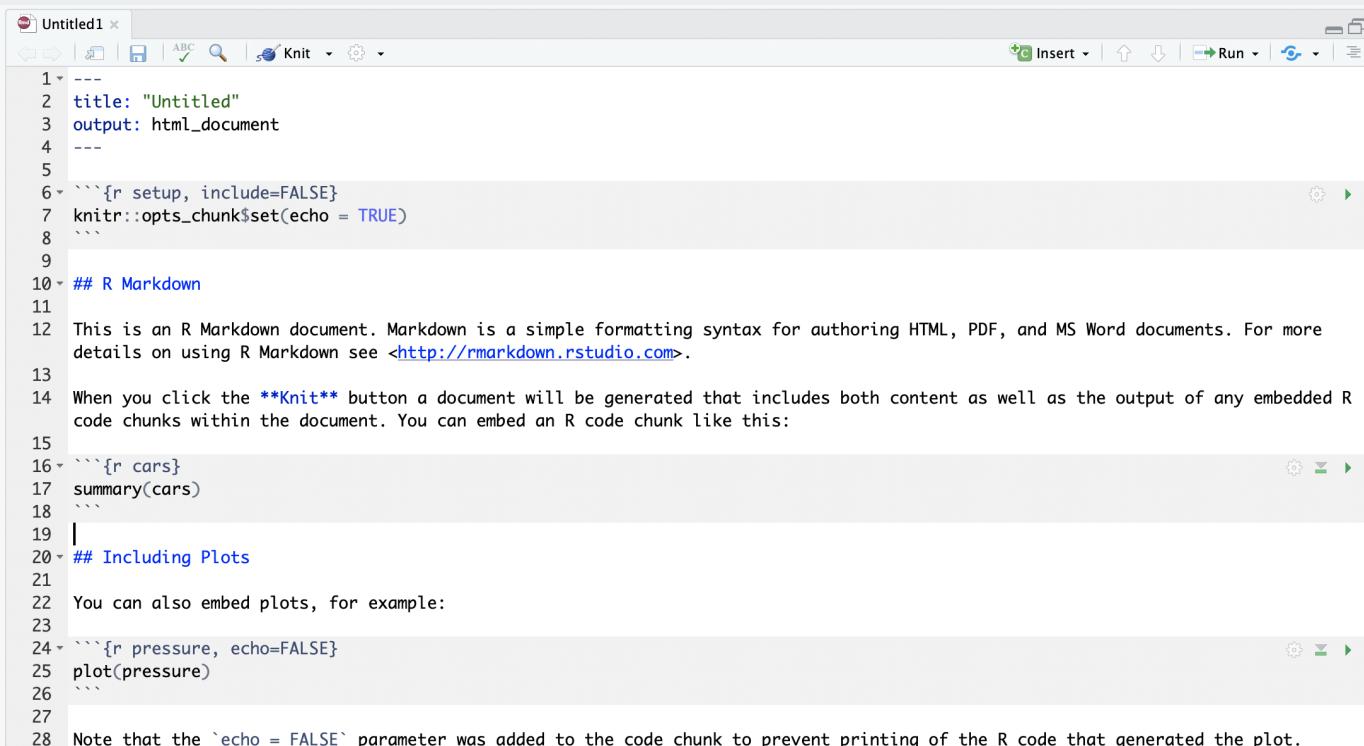
<https://www.rstudio.com/products/rpackages/>

Create an R Markdown file (.Rmd)

Two options:

1. click on File → New File → R Markdown..., or
2. in upper left corner of RStudio click on  →  R Markdown...

You should see the following text in your editor window:



```
1 ---  
2 title: "Untitled"  
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  
16 ```{r cars}  
17 summary(cars)  
18 ```  
19  
20 ## Including Plots  
21  
22 You can also embed plots, for example:  
23  
24 ```{r pressure, echo=FALSE}  
25 plot(pressure)  
26 ```  
27  
28 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
```

Knit the .Rmd file

Before knitting the .Rmd file, you must first **save it**.

To **knit** the .Rmd file, either

1. click on the knit icon  at the top of the editor window
2. or use keyboard shortcuts
 - Mac: *Command+Shift+K*
 - PC: *Ctrl+Shift+K*
3. or use the **render()** command in Console - See *Extensions section* for details

A new window will open with the html output.

Remark:

- The template .Rmd file that RStudio creates will knit to an html file by default
- Later we will go over knitting to other file types

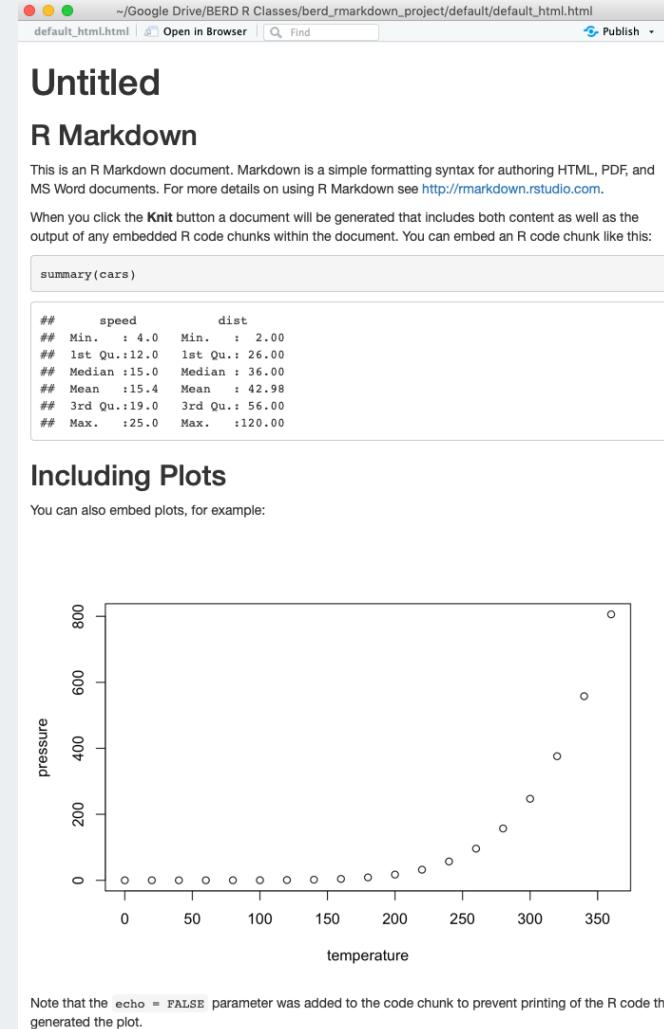
Compare the .Rmd file with its html output

.Rmd file

```
default_html.Rmd x
ABC Knit Insert Run

1 ---  
2 title: "Untitled"  
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  
16 ```{r cars}  
17 summary(cars)  
18 |||  
19  
20 ## Including Plots  
21  
22 You can also embed plots, for example:  
23  
24 ```{r pressure, echo=FALSE}  
25 plot(pressure)  
26 |||  
27  
28 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
```

html output



Compare the .Rmd file with its html output

.Rmd file

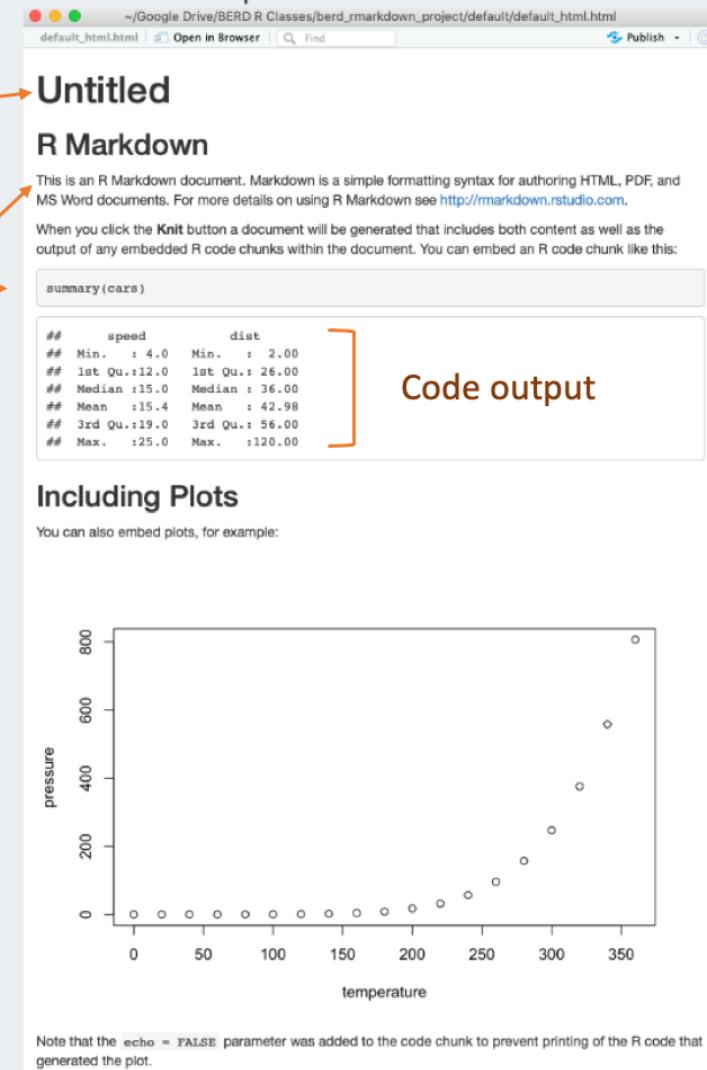
```
default_html.Rmd x
1 ---  
2 title: "Untitled"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8  
9  
10 ## R Markdown  
11  
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word 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  
16 ```{r cars}  
17 summary(cars)  
18 ...  
19  
20 ## Including Plots  
21  
22 You can also embed plots, for example:  
23  
24 ```{r pressure, echo=FALSE}  
25 plot(pressure)  
26 ...  
27  
28 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
```

YAML metadata

Code chunk

Text

html output



3 types of R Markdown content

1. *Text*
2. Code chunks
3. YAML metadata

Formatting text

- Markdown is a markup language similar to html or LaTeX
- All text formatting is specified via code

Text in editor:

```
Time to learn how to format text using R Markdown!  
  
If I put two spaces  
at the end of a line it will force a line break and start a new line.  
  
*This text is in italics*, but so is this text.  
  
**Bold** also has 2 options  
  
~~Should this be deleted?~~  
  
'Sometimes text needs to be verbatim'  
  
>or even a block quote.  
  
Needsuperscripts orsubscripts?
```

Output:

```
Time to learn how to format text using R Markdown!  
  
If I put two spaces  
at the end of a line it will force a line break and start a new line.  
  
This text is in italics, but so is this text.  
  
Bold also has 2 options  
  
Should this be deleted?  
  
Sometimes text needs to be verbatim  
  
or even a block quote.  
  
Needsuperscripts orsubscripts?
```

Headers

- Organize your documents using headers to create sections and subsections
- Later in the workshop we will cover
 - automatically numbering headers in output file for easy reference
 - easily creating a TOC based on the header names

Text in editor:

```
# 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

RStudio tip

You can easily navigating through your .Rmd file if you use headers to outline your text

The screenshot shows the RStudio interface with an R Markdown document titled "Untitled1". The code editor contains the following R Markdown code:

```
1 ---  
2 title: "Untitled"  
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  
16 `Untitled`  
17 |  
18 |  
19 |  
20 |  
21 |  
22 |  
23 |  
24 |  
25 |  
26 |  
27 |  
28 |  
29 |  
30 |  
31 |  
32 |  
33 |  
34 |  
35 |  
36 |  
37 |  
38 |  
39 |  
40 |  
41 |  
42 |  
43 |  
44 |  
45 |  
46 |  
47 |  
48 |  
49 |  
50 |  
51 |  
52 |  
53 |  
54 |  
55 |  
56 |  
57 |  
58 |  
59 |  
60 |  
61 |  
62 |  
63 |  
64 |  
65 |  
66 |  
67 |  
68 |  
69 |  
70 |  
71 |  
72 |  
73 |  
74 |  
75 |  
76 |  
77 |  
78 |  
79 |  
80 |  
81 |  
82 |  
83 |  
84 |  
85 |  
86 |  
87 |  
88 |  
89 |  
90 |  
91 |  
92 |  
93 |  
94 |  
95 |  
96 |  
97 |  
98 |  
99 |  
100 |  
101 |  
102 |  
103 |  
104 |  
105 |  
106 |  
107 |  
108 |  
109 |  
110 |  
111 |  
112 |  
113 |  
114 |  
115 |  
116 |  
117 |  
118 |  
119 |  
120 |  
121 |  
122 |  
123 |  
124 |  
125 |  
126 |  
127 |  
128 |  
129 |  
130 |  
131 |  
132 |  
133 |  
134 |  
135 |  
136 |  
137 |  
138 |  
139 |  
140 |  
141 |  
142 |  
143 |  
144 |  
145 |  
146 |  
147 |  
148 |  
149 |  
150 |  
151 |  
152 |  
153 |  
154 |  
155 |  
156 |  
157 |  
158 |  
159 |  
160 |  
161 |  
162 |  
163 |  
164 |  
165 |  
166 |  
167 |  
168 |  
169 |  
170 |  
171 |  
172 |  
173 |  
174 |  
175 |  
176 |  
177 |  
178 |  
179 |  
180 |  
181 |  
182 |  
183 |  
184 |  
185 |  
186 |  
187 |  
188 |  
189 |  
190 |  
191 |  
192 |  
193 |  
194 |  
195 |  
196 |  
197 |  
198 |  
199 |  
200 |  
201 |  
202 |  
203 |  
204 |  
205 |  
206 |  
207 |  
208 |  
209 |  
210 |  
211 |  
212 |  
213 |  
214 |  
215 |  
216 |  
217 |  
218 |  
219 |  
220 |  
221 |  
222 |  
223 |  
224 |  
225 |  
226 |  
227 |  
228 |  
229 |  
230 |  
231 |  
232 |  
233 |  
234 |  
235 |  
236 |  
237 |  
238 |  
239 |  
240 |  
241 |  
242 |  
243 |  
244 |  
245 |  
246 |  
247 |  
248 |  
249 |  
250 |  
251 |  
252 |  
253 |  
254 |  
255 |  
256 |  
257 |  
258 |  
259 |  
260 |  
261 |  
262 |  
263 |  
264 |  
265 |  
266 |  
267 |  
268 |  
269 |  
270 |  
271 |  
272 |  
273 |  
274 |  
275 |  
276 |  
277 |  
278 |  
279 |  
280 |  
281 |  
282 |  
283 |  
284 |  
285 |  
286 |  
287 |  
288 |  
289 |  
290 |  
291 |  
292 |  
293 |  
294 |  
295 |  
296 |  
297 |  
298 |  
299 |  
300 |  
301 |  
302 |  
303 |  
304 |  
305 |  
306 |  
307 |  
308 |  
309 |  
310 |  
311 |  
312 |  
313 |  
314 |  
315 |  
316 |  
317 |  
318 |  
319 |  
320 |  
321 |  
322 |  
323 |  
324 |  
325 |  
326 |  
327 |  
328 |  
329 |  
330 |  
331 |  
332 |  
333 |  
334 |  
335 |  
336 |  
337 |  
338 |  
339 |  
340 |  
341 |  
342 |  
343 |  
344 |  
345 |  
346 |  
347 |  
348 |  
349 |  
350 |  
351 |  
352 |  
353 |  
354 |  
355 |  
356 |  
357 |  
358 |  
359 |  
360 |  
361 |  
362 |  
363 |  
364 |  
365 |  
366 |  
367 |  
368 |  
369 |  
370 |  
371 |  
372 |  
373 |  
374 |  
375 |  
376 |  
377 |  
378 |  
379 |  
380 |  
381 |  
382 |  
383 |  
384 |  
385 |  
386 |  
387 |  
388 |  
389 |  
390 |  
391 |  
392 |  
393 |  
394 |  
395 |  
396 |  
397 |  
398 |  
399 |  
400 |  
401 |  
402 |  
403 |  
404 |  
405 |  
406 |  
407 |  
408 |  
409 |  
410 |  
411 |  
412 |  
413 |  
414 |  
415 |  
416 |  
417 |  
418 |  
419 |  
420 |  
421 |  
422 |  
423 |  
424 |  
425 |  
426 |  
427 |  
428 |  
429 |  
430 |  
431 |  
432 |  
433 |  
434 |  
435 |  
436 |  
437 |  
438 |  
439 |  
440 |  
441 |  
442 |  
443 |  
444 |  
445 |  
446 |  
447 |  
448 |  
449 |  
450 |  
451 |  
452 |  
453 |  
454 |  
455 |  
456 |  
457 |  
458 |  
459 |  
460 |  
461 |  
462 |  
463 |  
464 |  
465 |  
466 |  
467 |  
468 |  
469 |  
470 |  
471 |  
472 |  
473 |  
474 |  
475 |  
476 |  
477 |  
478 |  
479 |  
480 |  
481 |  
482 |  
483 |  
484 |  
485 |  
486 |  
487 |  
488 |  
489 |  
490 |  
491 |  
492 |  
493 |  
494 |  
495 |  
496 |  
497 |  
498 |  
499 |  
500 |  
501 |  
502 |  
503 |  
504 |  
505 |  
506 |  
507 |  
508 |  
509 |  
510 |  
511 |  
512 |  
513 |  
514 |  
515 |  
516 |  
517 |  
518 |  
519 |  
520 |  
521 |  
522 |  
523 |  
524 |  
525 |  
526 |  
527 |  
528 |  
529 |  
530 |  
531 |  
532 |  
533 |  
534 |  
535 |  
536 |  
537 |  
538 |  
539 |  
540 |  
541 |  
542 |  
543 |  
544 |  
545 |  
546 |  
547 |  
548 |  
549 |  
550 |  
551 |  
552 |  
553 |  
554 |  
555 |  
556 |  
557 |  
558 |  
559 |  
560 |  
561 |  
562 |  
563 |  
564 |  
565 |  
566 |  
567 |  
568 |  
569 |  
570 |  
571 |  
572 |  
573 |  
574 |  
575 |  
576 |  
577 |  
578 |  
579 |  
580 |  
581 |  
582 |  
583 |  
584 |  
585 |  
586 |  
587 |  
588 |  
589 |  
590 |  
591 |  
592 |  
593 |  
594 |  
595 |  
596 |  
597 |  
598 |  
599 |  
600 |  
601 |  
602 |  
603 |  
604 |  
605 |  
606 |  
607 |  
608 |  
609 |  
610 |  
611 |  
612 |  
613 |  
614 |  
615 |  
616 |  
617 |  
618 |  
619 |  
620 |  
621 |  
622 |  
623 |  
624 |  
625 |  
626 |  
627 |  
628 |  
629 |  
630 |  
631 |  
632 |  
633 |  
634 |  
635 |  
636 |  
637 |  
638 |  
639 |  
640 |  
641 |  
642 |  
643 |  
644 |  
645 |  
646 |  
647 |  
648 |  
649 |  
650 |  
651 |  
652 |  
653 |  
654 |  
655 |  
656 |  
657 |  
658 |  
659 |  
660 |  
661 |  
662 |  
663 |  
664 |  
665 |  
666 |  
667 |  
668 |  
669 |  
670 |  
671 |  
672 |  
673 |  
674 |  
675 |  
676 |  
677 |  
678 |  
679 |  
680 |  
681 |  
682 |  
683 |  
684 |  
685 |  
686 |  
687 |  
688 |  
689 |  
690 |  
691 |  
692 |  
693 |  
694 |  
695 |  
696 |  
697 |  
698 |  
699 |  
700 |  
701 |  
702 |  
703 |  
704 |  
705 |  
706 |  
707 |  
708 |  
709 |  
710 |  
711 |  
712 |  
713 |  
714 |  
715 |  
716 |  
717 |  
718 |  
719 |  
720 |  
721 |  
722 |  
723 |  
724 |  
725 |  
726 |  
727 |  
728 |  
729 |  
730 |  
731 |  
732 |  
733 |  
734 |  
735 |  
736 |  
737 |  
738 |  
739 |  
740 |  
741 |  
742 |  
743 |  
744 |  
745 |  
746 |  
747 |  
748 |  
749 |  
750 |  
751 |  
752 |  
753 |  
754 |  
755 |  
756 |  
757 |  
758 |  
759 |  
760 |  
761 |  
762 |  
763 |  
764 |  
765 |  
766 |  
767 |  
768 |  
769 |  
770 |  
771 |  
772 |  
773 |  
774 |  
775 |  
776 |  
777 |  
778 |  
779 |  
780 |  
781 |  
782 |  
783 |  
784 |  
785 |  
786 |  
787 |  
788 |  
789 |  
790 |  
791 |  
792 |  
793 |  
794 |  
795 |  
796 |  
797 |  
798 |  
799 |  
800 |  
801 |  
802 |  
803 |  
804 |  
805 |  
806 |  
807 |  
808 |  
809 |  
810 |  
811 |  
812 |  
813 |  
814 |  
815 |  
816 |  
817 |  
818 |  
819 |  
820 |  
821 |  
822 |  
823 |  
824 |  
825 |  
826 |  
827 |  
828 |  
829 |  
830 |  
831 |  
832 |  
833 |  
834 |  
835 |  
836 |  
837 |  
838 |  
839 |  
840 |  
841 |  
842 |  
843 |  
844 |  
845 |  
846 |  
847 |  
848 |  
849 |  
850 |  
851 |  
852 |  
853 |  
854 |  
855 |  
856 |  
857 |  
858 |  
859 |  
860 |  
861 |  
862 |  
863 |  
864 |  
865 |  
866 |  
867 |  
868 |  
869 |  
870 |  
871 |  
872 |  
873 |  
874 |  
875 |  
876 |  
877 |  
878 |  
879 |  
880 |  
881 |  
882 |  
883 |  
884 |  
885 |  
886 |  
887 |  
888 |  
889 |  
890 |  
891 |  
892 |  
893 |  
894 |  
895 |  
896 |  
897 |  
898 |  
899 |  
900 |  
901 |  
902 |  
903 |  
904 |  
905 |  
906 |  
907 |  
908 |  
909 |  
910 |  
911 |  
912 |  
913 |  
914 |  
915 |  
916 |  
917 |  
918 |  
919 |  
920 |  
921 |  
922 |  
923 |  
924 |  
925 |  
926 |  
927 |  
928 |  
929 |  
930 |  
931 |  
932 |  
933 |  
934 |  
935 |  
936 |  
937 |  
938 |  
939 |  
940 |  
941 |  
942 |  
943 |  
944 |  
945 |  
946 |  
947 |  
948 |  
949 |  
950 |  
951 |  
952 |  
953 |  
954 |  
955 |  
956 |  
957 |  
958 |  
959 |  
960 |  
961 |  
962 |  
963 |  
964 |  
965 |  
966 |  
967 |  
968 |  
969 |  
970 |  
971 |  
972 |  
973 |  
974 |  
975 |  
976 |  
977 |  
978 |  
979 |  
980 |  
981 |  
982 |  
983 |  
984 |  
985 |  
986 |  
987 |  
988 |  
989 |  
990 |  
991 |  
992 |  
993 |  
994 |  
995 |  
996 |  
997 |  
998 |  
999 |  
1000 |  
1001 |  
1002 |  
1003 |  
1004 |  
1005 |  
1006 |  
1007 |  
1008 |  
1009 |  
10010 |  
10011 |  
10012 |  
10013 |  
10014 |  
10015 |  
10016 |  
10017 |  
10018 |  
10019 |  
10020 |  
10021 |  
10022 |  
10023 |  
10024 |  
10025 |  
10026 |  
10027 |  
10028 |  
10029 |  
10030 |  
10031 |  
10032 |  
10033 |  
10034 |  
10035 |  
10036 |  
10037 |  
10038 |  
10039 |  
10040 |  
10041 |  
10042 |  
10043 |  
10044 |  
10045 |  
10046 |  
10047 |  
10048 |  
10049 |  
10050 |  
10051 |  
10052 |  
10053 |  
10054 |  
10055 |  
10056 |  
10057 |  
10058 |  
10059 |  
10060 |  
10061 |  
10062 |  
10063 |  
10064 |  
10065 |  
10066 |  
10067 |  
10068 |  
10069 |  
10070 |  
10071 |  
10072 |  
10073 |  
10074 |  
10075 |  
10076 |  
10077 |  
10078 |  
10079 |  
10080 |  
10081 |  
10082 |  
10083 |  
10084 |  
10085 |  
10086 |  
10087 |  
10088 |  
10089 |  
10090 |  
10091 |  
10092 |  
10093 |  
10094 |  
10095 |  
10096 |  
10097 |  
10098 |  
10099 |  
100100 |  
100101 |  
100102 |  
100103 |  
100104 |  
100105 |  
100106 |  
100107 |  
100108 |  
100109 |  
100110 |  
100111 |  
100112 |  
100113 |  
100114 |  
100115 |  
100116 |  
100117 |  
100118 |  
100119 |  
100120 |  
100121 |  
100122 |  
100123 |  
100124 |  
100125 |  
100126 |  
100127 |  
100128 |  
100129 |  
100130 |  
100131 |  
100132 |  
100133 |  
100134 |  
100135 |  
100136 |  
100137 |  
100138 |  
100139 |  
100140 |  
100141 |  
100142 |  
100143 |  
100144 |  
100145 |  
100146 |  
100147 |  
100148 |  
100149 |  
100150 |  
100151 |  
100152 |  
100153 |  
100154 |  
100155 |  
100156 |  
100157 |  
100158 |  
100159 |  
100160 |  
100161 |  
100162 |  
100163 |  
100164 |  
100165 |  
100166 |  
100167 |  
100168 |  
100169 |  
100170 |  
100171 |  
100172 |  
100173 |  
100174 |  
100175 |  
100176 |  
100177 |  
100178 |  
100179 |  
100180 |  
100181 |  
100182 |  
100183 |  
100184 |  
100185 |  
100186 |  
100187 |  
100188 |  
100189 |  
100190 |  
100191 |  
100192 |  
100193 |  
100194 |  
100195 |  
100196 |  
100197 |  
100198 |  
100199 |  
100200 |  
100201 |  
100202 |  
100203 |  
100204 |  
100205 |  
100206 |  
100207 |  
100208 |  
100209 |  
100210 |  
100211 |  
100212 |  
100213 |  
100214 |  
100215 |  
100216 |  
100217 |  
100218 |  
100219 |  
100220 |  
100221 |  
100222 |  
100223 |  
100224 |  
100225 |  
100226 |  
100227 |  
100228 |  
100229 |  
100230 |  
100231 |  
100232 |  
100233 |  
100234 |  
100235 |  
100236 |  
100237 |  
100238 |  
100239 |  
100240 |  
100241 |  
100242 |  
100243 |  
100244 |  
100245 |  
100246 |  
100247 |  
100248 |  
100249 |  
100250 |  
100251 |  
100252 |  
100253 |  
100254 |  
100255 |  
100256 |  
100257 |  
100258 |  
100259 |  
100260 |  
100261 |  
100262 |  
100263 |  
100264 |  
100265 |  
100266 |  
100267 |  
100268 |  
100269 |  
100270 |  
100271 |  
100272 |  
100273 |  
100274 |  
100275 |  
100276 |  
100277 |  
100278 |  
100279 |  
100280 |  
100281 |  
100282 |  
100283 |  
100284 |  
100285 |  
100286 |  
100287 |  
100288 |  
100289 |  
100290 |  
100291 |  
100292 |  
100293 |  
100294 |  
100295 |  
100296 |  
100297 |  
100298 |  
100299 |  
100300 |  
100301 |  
100302 |  
100303 |  
100304 |  
100305 |  
100306 |  
100307 |  
100308 |  
100309 |  
100310 |  
100311 |  
100312 |  
100313 |  
100314 |  
100315 |  
100316 |  
100317 |  
100318 |  
100319 |  
100320 |  
100321 |  
100322 |  
100323 |  
100324 |  
100325 |  
100326 |  
100327 |  
100328 |  
100329 |  
100330 |  
100331 |  
100332 |  
100333 |  
100334 |  
100335 |  
100336 |  
100337 |  
100338 |  
100339 |  
100340 |  
100341 |  
100342 |  
100343 |  
100344 |  
100345 |  
100346 |  
100347 |  
100348 |  
100349 |  
100350 |  
100351 |  
100352 |  
100353 |  
100354 |  
100355 |  
100356 |  
100357 |  
100358 |  
100359 |  
100360 |  
100361 |  
100362 |  
100363 |  
100364 |  
100365 |  
100366 |  
100367 |  
100368 |  
100369 |  
100370 |  
100371 |  
100372 |  
100373 |  
100374 |  
100375 |  
100376 |  
100377 |  
100378 |  
100379 |  
100380 |  
100381 |  
100382 |  
100383 |  
100384 |  
100385 |  
100386 |  
100387 |  
100388 |  
100389 |  
100390 |  
100391 |  
100392 |  
100393 |  
100394 |  
100395 |  
100396 |  
100397 |  
100398 |  
100399 |  
100400 |  
100401 |  
100402 |  
100403 |  
100404 |  
100405 |  
100406 |  
100407 |  
100408 |  
100409 |  
100410 |  
100411 |  
100412 |  
100413 |  
100414 |  
100415 |  
100416 |  
100417 |  
100418 |  
100419 |  
100420 |  
100421 |  
100422 |  
100423 |  
100424 |  
100425 |  
100426 |  
100427 |  
100428 |  
100429 |  
100430 |  
100431 |  
100432 |  
100433 |  
100434 |  
100435 |  
100436 |  
100437 |  
100438 |  
100439 |  
100440 |  
100441 |  
100442 |  
100443 |  
100444 |  
100445 |  
100446
```

Unnumbered lists

Text in editor:

- * This is an **unnumbered list***
 - + with **sub-items**
 - and **sub-sub-items**,
 - or even deeper.
- * You can use characters *, +, and - to create lists.
 - * The order of the
 - * characters is not important
 - + and characters can be repeated.

What ***is*** important is the ***spacing***!

- + indent each
 - * sub-level with a tab and make sure
 - * there is a space between the character starting the list and the first bit of text,
 - *otherwise the text won't be a new bullet in the list

- This is an **unnumbered list**
 - with *sub-items*
 - and *sub-sub-items*,
 - or even deeper.
- You can use characters *, +, and - to create lists.
 - The order of the
 - characters is not important
 - and characters can be repeated.

What *is* important is the *spacing*!

- indent each
 - sub-level with a tab and make sure
 - there is a space between the character starting the list and the first bit of text, *otherwise the text won't be a new bullet in the list

Numbered lists

Text in editor:

1. This is a ****Numbered list****,
 1. which can have
 - i. sub-items
 - A. and sub-sub-items
 1. Each bullet
 - i. can start with `1.` or `i.`,
 - i. in theory.
- (@) This doesn't always work
- when lists get interrupted though.
- (@) Using `(@)` instead keeps
 - * the numbering continuous.
 - + Note that you can also
- (@) nest unnumbered lists
 - * within numbered lists

Output:

1. This is a **Numbered list**,
 2. which can have
 - i. sub-items
 - A. and sub-sub-items
 1. Each bullet
 - i. can start with 1. or i.,
 - ii. in theory.
1. This doesn't always work
- when lists get interrupted though.
2. Using (@) instead keeps
 - the numbering continuous.
 - Note that you can also
3. nest unnumbered lists
 - within numbered lists

Math, horizontal rule, and hyperlinks

Text in editor:

```
* __Mathematical formulas and symbols__ can be included  
using LaTeX, both as *inline equations* or *formulas*:  
    + Use single `$` for inline equations: $y=\beta_0 +  
      \beta_1x + \varepsilon  
    + Use double `$$` for centered formulas:  
  
$$\hat{y} = \frac{3}{7} + 5 \text{age} + 3^2 \cdot \text{height}$$  
  
* __Horizontal rule__  
  
***  
* **Hyperlinks**  
    + Learn more about LaTeX at this  
[link](http://www.highpoint.edu/physics/files/2014/08/s  
hort-math-guide.pdf).  
]
```

Output:

- **Mathematical formulas and symbols** can be included using LaTeX, both as *inline equations* or *formulas*:
 - Use single \$ for inline equations:
$$y = \beta_0 + \beta_1 x + \varepsilon$$
 - Use double \$\$ for centered formulas:

$$\hat{y} = \frac{3}{7} + 5 \text{age} + 3^2 \cdot \text{height}$$

- **Horizontal rule**
-

- **Hyperlinks**
 - Learn more about LaTeX at this [link](http://www.highpoint.edu/physics/files/2014/08/short-math-guide.pdf).

Insert images

Text in editor:

Gauss and the normal distribution were featured on the 10 Deutsch Mark (DM) bill.
![alternate text: 10 DM bill](DM_10_Gauss.jpeg)



```
<!-- The alternate text only appears if the image fails to load. -->
<!-- By the way, this is how you write comments in markdown!! -->
```

You can also source an image on the internet instead:

![10 DM bill](https://history.info/wp-content/uploads/2015/06/DEU-10m-anv.jpg)

Output:

Gauss and the normal distribution were featured on the 10 Deutsch Mark (DM) bill.



You can also source an image on the internet instead:

Tables created manually

Later we will use R code to create tables from data.

We can create tables using Markdown as well:

Text in editor:

Variable	n	Mean	\pm SE
Age	198	42.3	\pm 3.1 years
Height	194	68.1	\pm 2.6 in

Output:

Variable	n	Mean \pm SE
Age	198	42.3 \pm 3.1 years
Height	194	68.1 \pm 2.6 in

- We **do not recommend** creating tables where the numbers are hard-coded
 - since they are **not reproducible!**

Spell check

Alas, there is no automatic spell check to catch your typos and grammar.

- You can manually do a spell check by clicking on the  icon above the editor window.
- There is no built-in grammar check in RStudio.
 - The [gramr package](#) is an available RStudio Addin.

Practice!

Create an .Rmd file with file name **example1.Rmd** that creates the html output to the right.

- Hint: The first line is not a header.

Example 1

To-do list

Shopping list

Farmers' market

1. Fruit

- raspberries
- marionberries

2. Veggies

- lettuce
- tomatoes

Grocery store

1. milk
2. eggs
3. *baking stuff*
 - flour
 - sugar

Recipe

Mix the following:

1 cup milk
 $\frac{1}{2}$ Tbsp sugar

Add 2 cups berries.
Let sit 30 min and serve.

Enjoy!

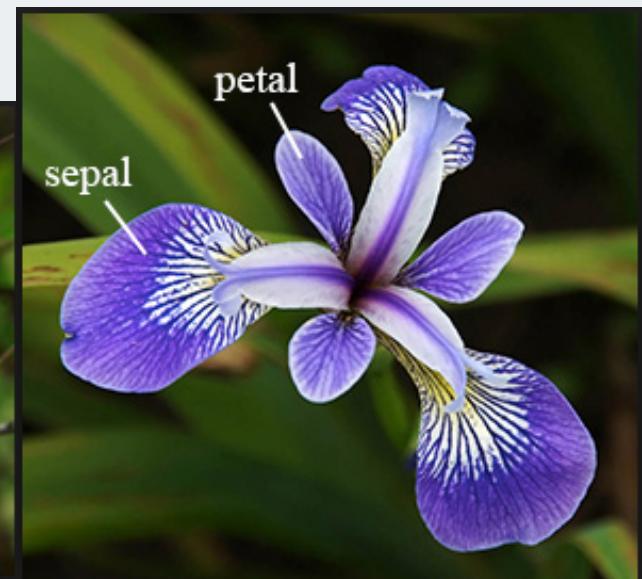
3 types of R Markdown content

1. Text
2. *Code chunks*
3. YAML metadata

Data description: Fisher's (or Anderson's) Iris data set

- $n = 150$
- 3 species of Iris flowers (Setosa, Virginica, and Versicolour)
 - 50 measurements of each type of Iris
- variables:
 - sepal length, sepal width, petal length, petal width, and species

Can the flower species be determined by these variables?



Code chunks

Chunks of R code start with ````{r}` and end with ```` .

```
```{r}  
summary(iris)
```

For example, the chunk `...` produces the output

```
summary(iris)
```

```
Sepal.Length Sepal.Width Petal.Length Petal.Width
Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100
1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300
Median :5.800 Median :3.000 Median :4.350 Median :1.300
Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199
3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800
Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500

Species
setosa :50
versicolor:50
virginica:50
```

# Create a code chunk

Code chunks can be created by either

1. Clicking on  →  at top right of editor window, or

## 2. **Keyboard shortcut**

- Mac: *Command + Option + I*
- PC: *Ctrl + Alt + I*

# Chunk options- most common

Text in editor:

```
No options specified: see both code and output
```{r}
mean(iris$Sepal.Length)
```

```

`__`echo`__` determines whether the R code is **displayed** or not. The default is `TRUE`. When set to `FALSE`, the code is not displayed in the output:

```
```{r echo=FALSE}
mean(iris$Sepal.Length)
```

```

`__`eval`__` determines whether the R code is **run** or not. The default is `TRUE`. When set to `FALSE`, the code is not run but is displayed in the output:

```
```{r eval=FALSE}
mean(iris$Sepal.Length)
```

```

No options specified: see both code and output

```
mean(iris$Sepal.Length)
```

```
[1] 5.843333
```

**echo** determines whether the R code is **displayed** or not. The default is **TRUE**. When set to **FALSE**, the code is not displayed in the output:

```
[1] 5.843333
```

**eval** determines whether the R code is **run** or not. The default is **TRUE**. When set to **FALSE**, the code is not run but is displayed in the output:

```
mean(iris$Sepal.Length)
```

# More chunk options

Text in editor:

```
__`include`__ determines whether to include
the R chunk in the output or not. The default
is `TRUE`. Below the chunk is run, but we do
not see the code or its output:
```{r include=FALSE}  
mean(iris$Sepal.Length)  
```
```

Output:

**include** determines whether to include the R chunk in the output or not. The default is **TRUE**. When set to **FALSE**, the chunk is run but we do not see the code or its output (note that nothing is displayed below):

- Setting **include=FALSE** is useful when you have R code that you want to run, but do not want to display either the code or its output.
- See the [R Markdown cheatsheet](#) for more chunk options.

# Inline code

- You can also report R code output inline with the text
  - *R code is not shown in this case*

Text in editor:

```
```{r include=FALSE}
mean_SepalLength <- mean(iris$Sepal.Length)
````
```

The mean sepal length for all 3 species combined is  
`r round(mean\_SepalLength,1)`  
(SD = `r round(sd(iris\$Sepal.Length),1)` cm.)

Output:

The mean sepal length for all 3 species combined is 5.8 (SD = 0.8) cm.

- The code above is an example of where `include=FALSE` is used a chunk option to evaluate the code but not show the code or its output.
  - It saves the mean as `mean_SepalLength`, which can then be used later on.
- For the standard deviation, the inline code did the calculation.
- Thus it was not necessary to first save the mean as a variable.

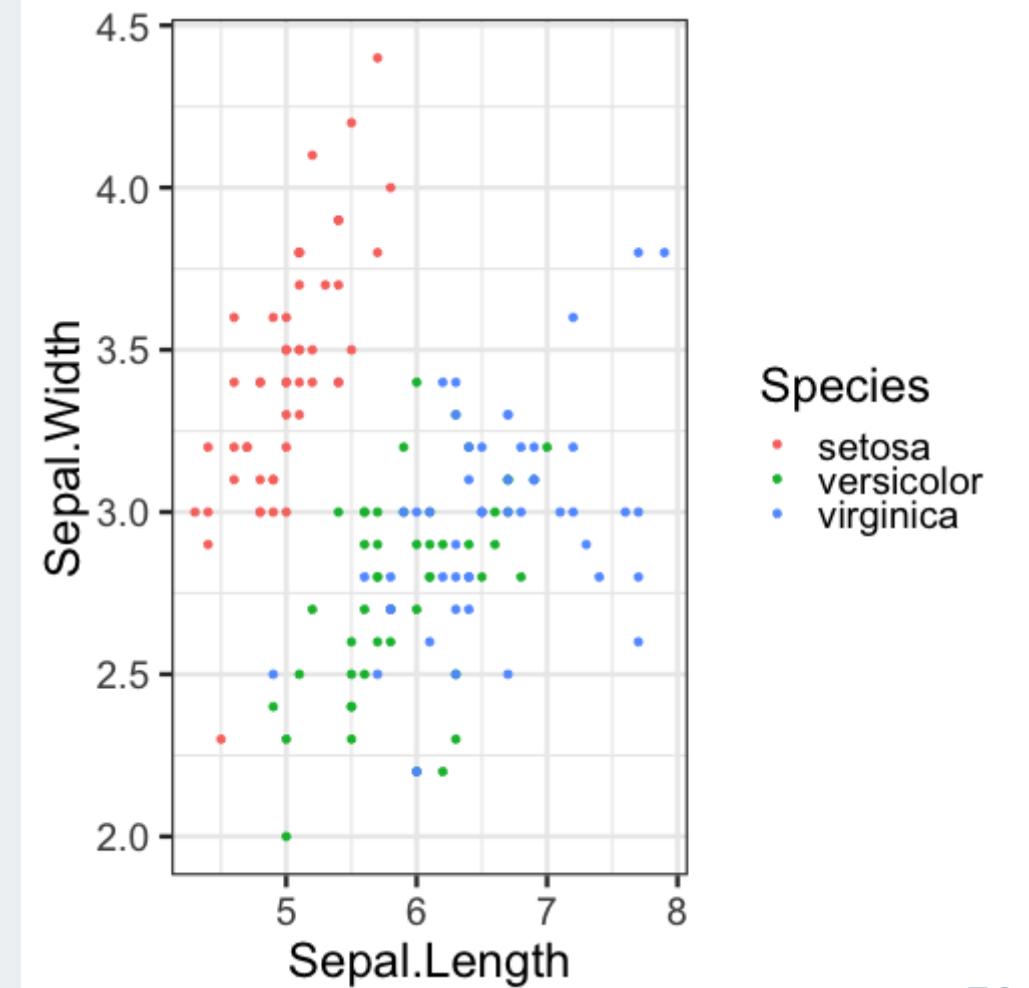
# Figures

Text in editor:

```
```{r Sepal_WidthVsHeight, echo=FALSE, fig.width=7, fig.height=7}
library(ggplot2) # loads ggplot2 package
# Don't need to load ggplot2 if already loaded tidyverse package
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,
                 color = Species)) +
  geom_point()
```
```

- Figure dimensions specified with **fig.width** and **fig.height**
- Figure name specified by the chunk label
  - The figure created by the chunk above is called **Sepal\_WidthVsHeight-1.png**
  - *Chunk names must be unique!*
- **echo=FALSE** was used to hide the code and only display the figure

Output:



# Tables - with no formatting

- Below a summary table with the mean and SD of sepal lengths is created as a tibble
- Then the table is displayed with no special formatting

```
table_sepal_length <- iris %>%
 group_by(Species) %>%
 summarize(mean = mean(Sepal.Length),
 SD = sd(Sepal.Length))

table_sepal_length
```

```
A tibble: 3 x 3
 Species mean SD
 <fct> <dbl> <dbl>
1 setosa 5.01 0.352
2 versicolor 5.94 0.516
3 virginica 6.59 0.636
```

# Tables - with kable

- The **kable** command from the **knitr** package has some basic formatting options
  - **html** tables: harder to read due to squished spacing; can include caption
  - **markdown** tables: nicer formatting; width = page width

Text in editor:

```
```{r echo=FALSE}
library(knitr)
# Only need to load package once in a document.
# Recommend doing this in 1st chunk of the .Rmd

kable(table_sepal_length,
      format = "html", digits = 2,
      caption = "Iris Sepal Lengths")

kable(table_sepal_length,
      format = "markdown", digits = 2,
      caption = "Iris Sepal Lengths")
# Note that the caption isn't shown!!
````
```

Output:

| Iris Sepal Lengths |      |      |
|--------------------|------|------|
| Species            | mean | SD   |
| setosa             | 5.01 | 0.35 |
| versicolor         | 5.94 | 0.52 |
| virginica          | 6.59 | 0.64 |

| Species    | mean | SD   |
|------------|------|------|
| setosa     | 5.01 | 0.35 |
| versicolor | 5.94 | 0.52 |
| virginica  | 6.59 | 0.64 |

# Tables - use kableExtra for more formatting options

Text in editor:

```
```{r echo=FALSE, message=FALSE}
library(kableExtra)

kable(table_sepal_length, digits = 2) %>%
  kable_styling(bootstrap_options = c("striped"),
                full_width = F) %>%
  add_header_above(c(" ", "Sepal Lengtha" = 2)) %>%
  # first column no header, next 2 columns have header
  add_indent(c(1, 2, 3)) %>%
  # specifying rows 1-3 of table; column names aren't a row
  footnote(general = "Fisher's Iris dataset",
            number = c("n = 150", "Data collected by Anderson"),
            alphabet = c("Lengths measured in cm"))
```
```

```

Output:

Species	Sepal Length ¹	
	mean	SD
setosa	5.01	0.35
versicolor	5.94	0.52
virginica	6.59	0.64

Note:

Fisher's Iris dataset

¹ n = 150

² Data collected by Anderson

^a Lengths measured in cm

See [Hao Zhu's webpage](#) for many, many more **kableExtra** options.

Global chunk options

- You can set **global chunk options** that are **applied to all chunks** in the .Rmd file
 - Set global options in a chunk at the beginning of the .Rmd file
 - The template .Rmd file already includes a chunk labeled **setup**
 - Add more options as desired to this chunk
- Options are added within the `knitr:::opts_chunk$set(...)` command
- Any of the many chunk options can be set in the **setup** chunk

```
```{r setup, include=FALSE}
knitr:::opts_chunk$set(
 fig.height=3, fig.width=7, fig.path='figs', fig.align = "center",
 echo = TRUE,
 warning=FALSE, message=FALSE,
 options(knitr.tables.format = `markdown`)
)
...```

```

- `fig.path` sets the folder name where figures generated by the .Rmd file will be saved
- See the [R Markdown cheatsheet](#) for more chunk options.

# Practice! (part 1)

Create a .Rmd file with file name **example2.Rmd** that creates the html output **example2/example2.html** below.

# Data summary

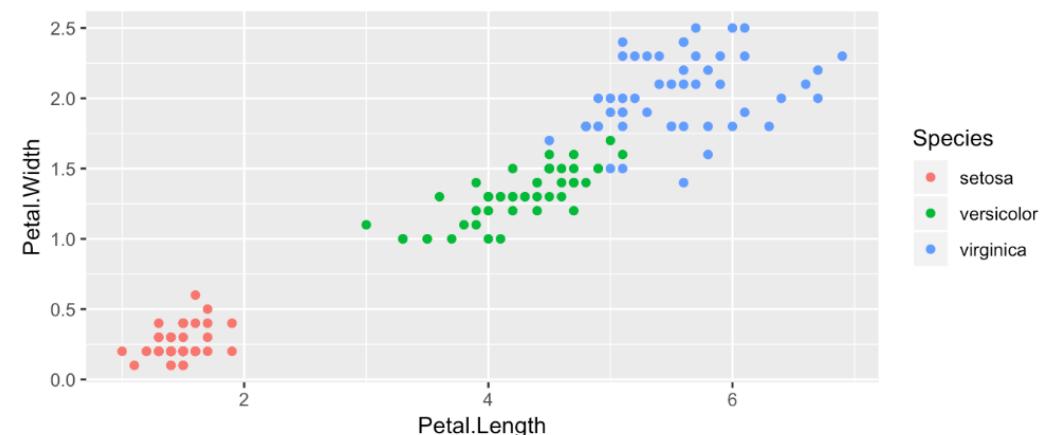
```
library(tidyverse)
library(knitr)
library(kableExtra)
```

**Data:** Fisher's (or Anderson's) Iris data set.

# Petal Widths

## Figure

### Scatterplot of petal widths vs. length by species type



# Practice! (part 2)

Create the table output shown below and at the end of `example2/example2.html` (code [link](#))

If you're already done...

Use code from

[https://haozhu233.github.io/kableExtra/awesome\\_table\\_in\\_html.html](https://haozhu233.github.io/kableExtra/awesome_table_in_html.html)

to figure out how to make the table to the right.

Species	Petal Width <sup>1</sup>		
	mean	SD	median
setosa	0.2	0.1	0.2
versicolor	1.3	0.2	1.3
virginica	2.0	0.3	2.0

<sup>1</sup> n = 50 for each species

# 3 types of R Markdown content

1. Text
2. Code chunks
3. *YAML metadata*

# YAML metadata

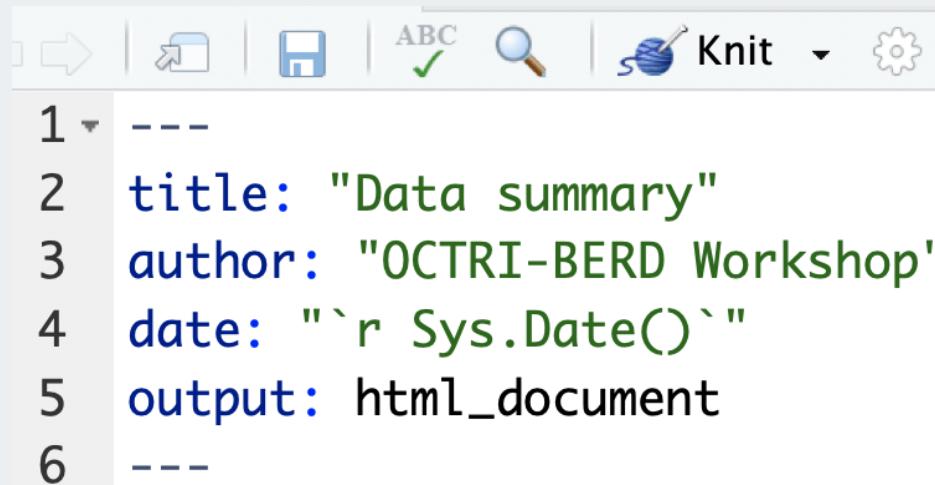
Many output options can be set in the **YAML metadata**, which is the *first set of code in the file starting and ending with ---*.

- YAML is an acronym for *YAML Ain't Markup Language*
- It sets the configuration specifications for the output file
- For more details about YAML in general, see the [YAML Wikipedia](#) page

---

Set the **title**, **author**, and **date** that appear at the top of the output file

Text in editor:

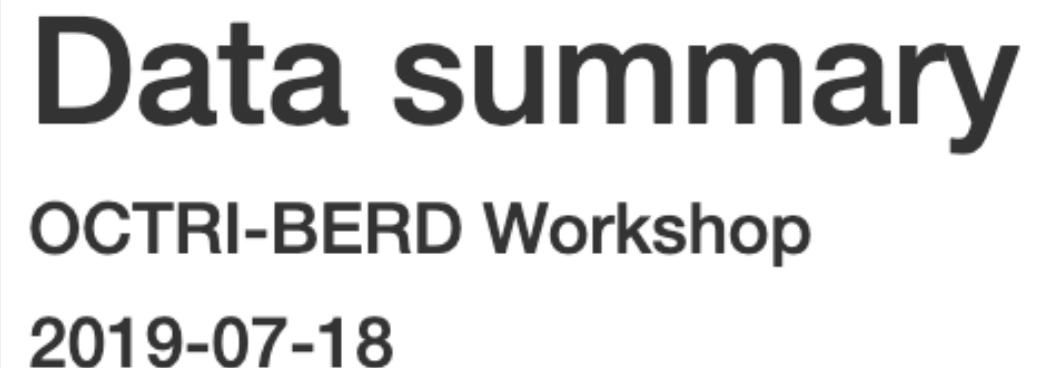


The screenshot shows the RStudio code editor with the following YAML code:

```
1 ---
2 title: "Data summary"
3 author: "OCTRI-BERD Workshop"
4 date: "`r Sys.Date()`"
5 output: html_document
6 ---
```

The editor has a toolbar with icons for back, forward, search, and knit.

Output:



The screenshot shows the generated HTML output with the following content:

# Data summary

OCTRI-BERD Workshop

2019-07-18

# Numbered sections & clickable table of contents

Text in editor:

```
1 ---
2 title: "Data summary"
3 author: "OCTRI-BERD Workshop"
4 date: "`r Sys.Date()`"
5 output:
6 html_document:
7 number_sections: yes
8 toc: yes
9 toc_float:
10 collapsed: no
11 smooth_scroll: yes
12 ---
```

Try out `collapsed: yes` and  
`smooth_scroll: no`

Output:

1 Petal Widths
1.1 Figure
1.2 Summary statistics
1.2.1 Raw table
1.2.2 Simple table
1.2.3 Somewhat better
1.2.4 Even better
1.2.5 If you're already done...

## Data summary

OCTRI-BERD Workshop

2019-07-18

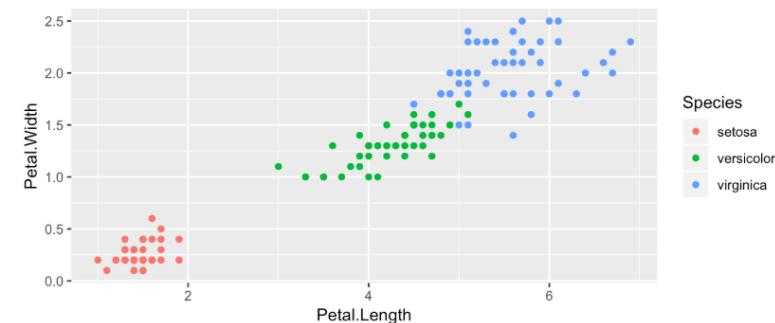
```
library(tidyverse)
library(knitr)
library(kableExtra)
```

Data: Fisher's (or Anderson's) Iris data set.

### 1 Petal Widths

#### 1.1 Figure

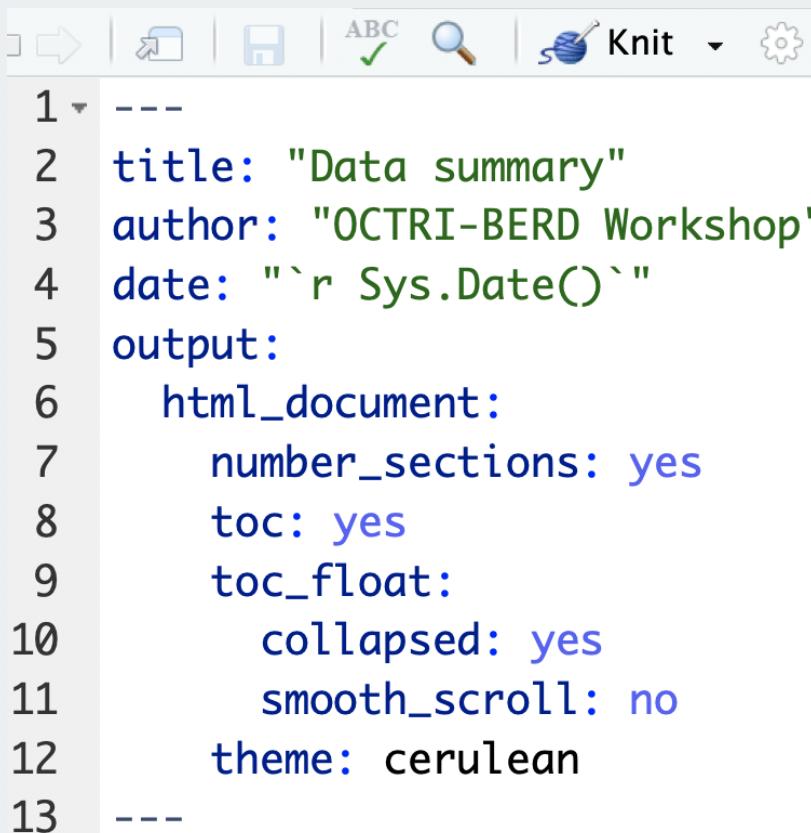
Scatterplot of petal widths vs. length by species type



# Themes

- There are 12 themes to choose from without installing additional packages
- See <http://www.datadreaming.org/post/r-markdown-theme-gallery/> for examples

Text in editor:



The screenshot shows a portion of an RStudio code editor. At the top, there's a toolbar with icons for file operations (New, Open, Save, Find, etc.) and a 'Knit' button. Below the toolbar is a code editor window containing the following YAML configuration:

```
1 ---
2 title: "Data summary"
3 author: "OCTRIBERD Workshop"
4 date: "`r Sys.Date()`"
5 output:
6 html_document:
7 number_sections: yes
8 toc: yes
9 toc_float:
10 collapsed: yes
11 smooth_scroll: no
12 theme: cerulean
13 ---
```

# Code folding

- Code folding creates buttons in the output html file that lets users choose whether they want to see the R code or not
  - This only applies to R code from chunks with `echo = TRUE`
- `code_folding: hide` all R code hidden by default; user must click Code button to see R
- `code_folding: show` all R code shown by default; user must click Code button to hide R
- See <https://bookdown.org/yihui/rmarkdown/html-document.html#code-folding> for more info

The screenshot shows a web page titled "Data summary" for the "OCTRI-BERD Workshop" on "2019-07-18". On the left, there's a sidebar with "1 Petal Widths" and links to "1.1 Figure" and "1.2 Summary statistics". On the right, there's a "Code" dropdown menu with "Show All Code" and "Hide All Code" options, and a "Hide" button below it. A code block at the bottom contains the following R code:

```
library(tidyverse)
library(knitr)
library(kableExtra)
```

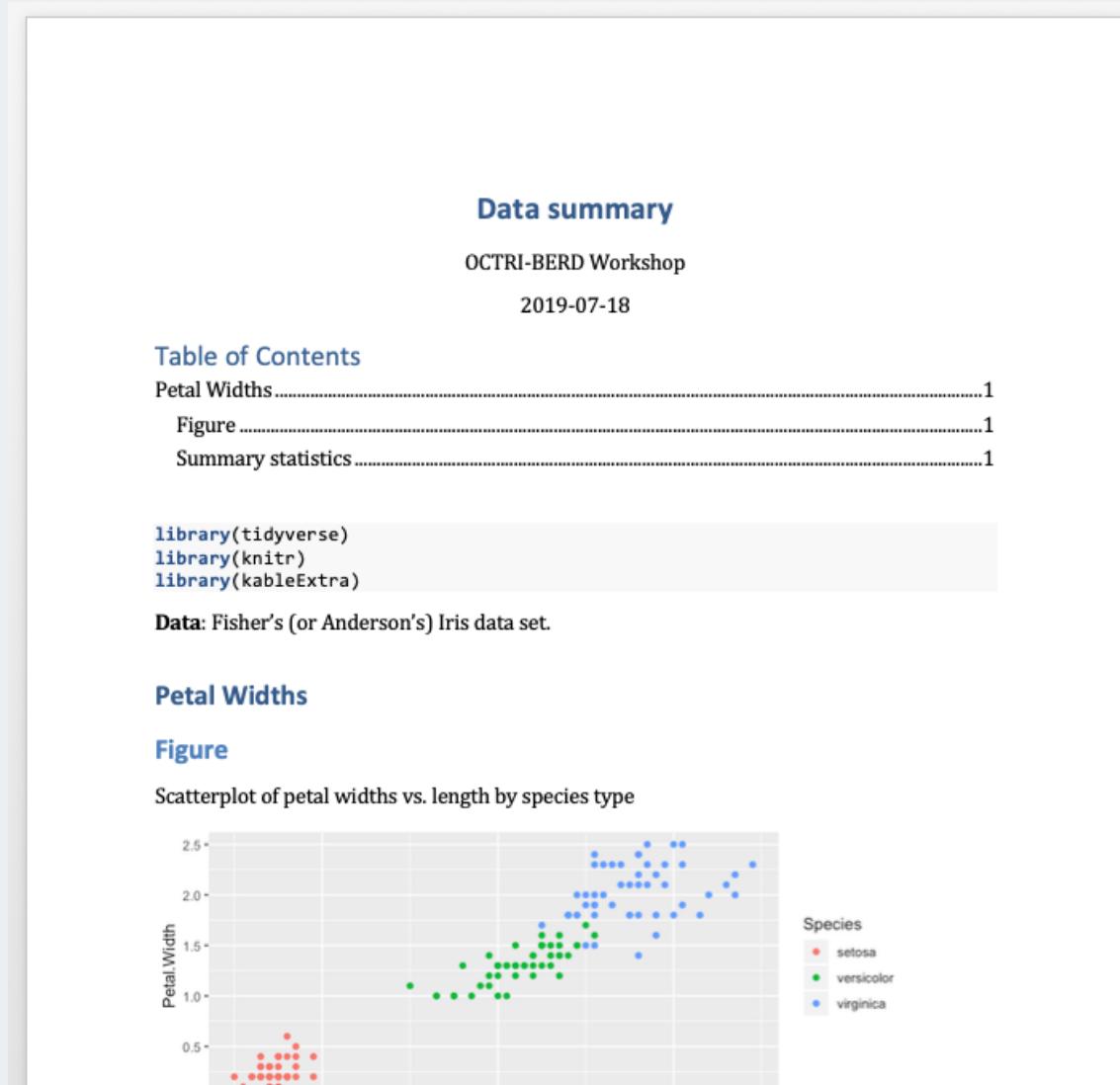
# Word documents

- Not many YAML options
- Cannot include html code or html-specific commands

Text in editor:

```
1 ---
2 title: "Data summary"
3 author: "OCTRIBERD Workshop"
4 date: "`r Sys.Date()`"
5 output:
6 word_document:
7 toc: yes
8 ---
```

Output:



# Word documents - tables options limited

- Cannot use `kableExtra` package options
- `kable` can be used

## Summary statistics

The following table summarizes petal widths by species type:

```
kable(table_petal_width, digits=1,
 format="markdown")
```

Species	mean	SD	median
setosa	0.2	0.1	0.2
versicolor	1.3	0.2	1.3
virginica	2.0	0.3	2.0

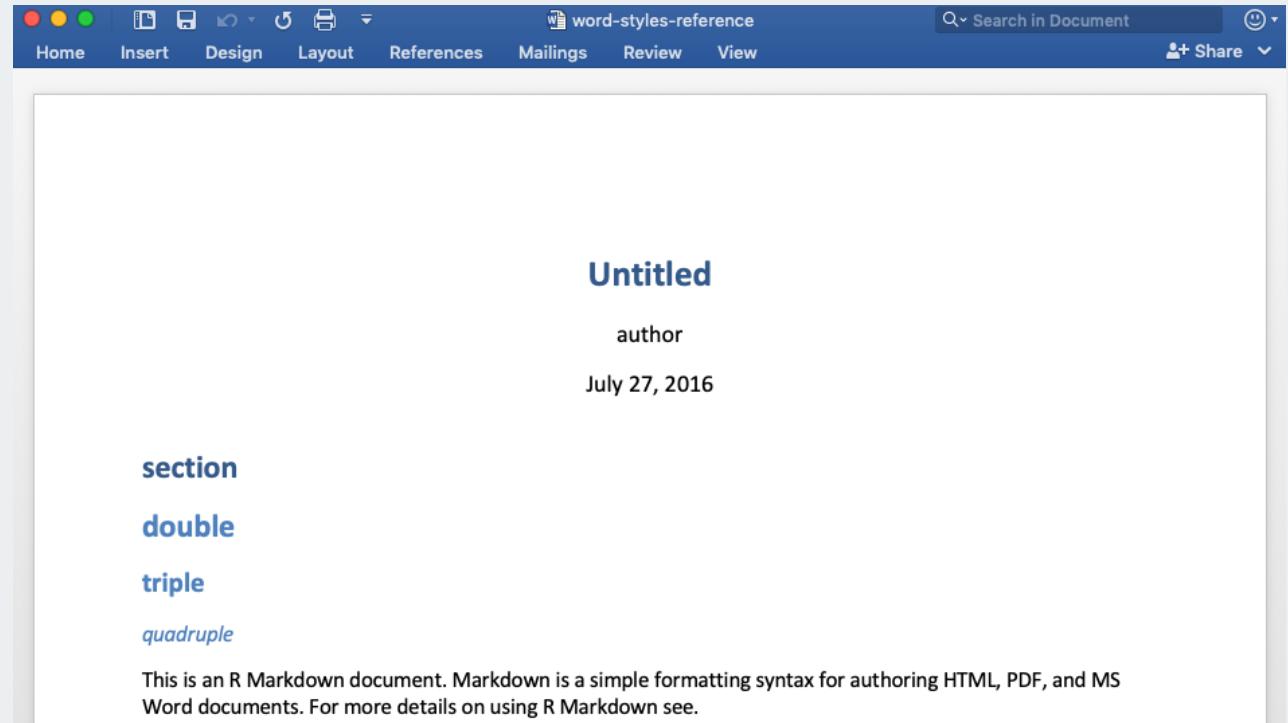
# Word documents - using a style file

- Create a Word doc with preferred formatting
  - font types and sizes, margins, header colors, etc.

YAML with code to include style file:

```
1 ---
2 title: "Data summary"
3 author: "OCTRIBERD Workshop"
4 date: "`r Sys.Date()`"
5 output:
 word_document:
 toc: yes
 reference_docx: word-styles-reference.docx
9 ---
```

Sample style file:



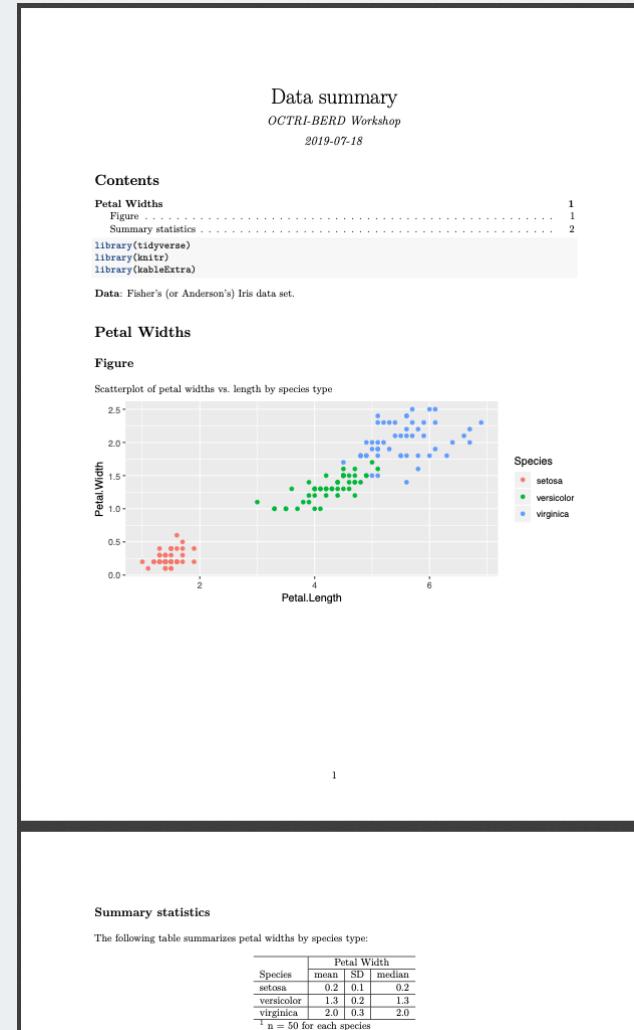
The Word doc created by RStudio will have the same formatting as the specified style file.

# pdf documents

Producing pdf documents requires that [LaTeX](#) be installed on your computer

- Few YAML options
- Lots of table options, including kableExtra
- Can use LaTeX code for formatting

```
1 ---
2 title: "Data summary"
3 author: "OCTRIBERD Workshop"
4 date: "`r Sys.Date()`"
5 output:
6 pdf_document:
7 toc: yes
8 ---
```



# Extensions and Tips

# Real time knitting: `xaringan::inf_mr()`

Instead of clicking "Knit" every time to see your updated document output, try this:

After installing the `xaringan` package:

```
install.packages("xaringan")
```

.Rmd files can be run and rendered "live" as you type when you either run

```
xaringan::inf_mr()
```

in the console when your .Rmd file is open. Or, click on Addins (top of screen), scroll down to "Xaringan" and click on "Infinite Moon Reader"

This is a new feature, so you need the most recent version of `xaringan` and RStudio. It works well for `html_document` output.

# Reproducible Workflow

# Be Organized

Your files must make sense to yourself 6 months from now, and/or other collaborators.



Jenny Bryan's "What They Forgot to Teach you About R" RStudio::conf2018 training

# No! Absolute! File! Paths! (don't setwd())

Absolute paths  $\neq$  reproducible

Relative paths = reproducible (if done correctly)

*If the first line of your R script is*

```
setwd("C:\Users\jenny\path\that\only\I\have")
```

*I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.*

Jenny Bryan's oft quoted opinion; see post on [Project-oriented workflow](#)

# Project directory structure

- The .Rproj file sets your working directory (**USE PROJECTS**)

```
Use a relative path, "relative to" the project folder
read_csv("mydata.csv") # looks in .Rproj folder
```

- When .Rmd files knit, they look for sourced files *in the folder they live in*

```
```{r data, eval=TRUE}  
read_csv("mydata.csv") # looks in .Rmd's folder  
```
```

- It's good practice to organize all your code/data/output into separate folders

These three facts together can cause a headache.

- Enter `here::here()`!

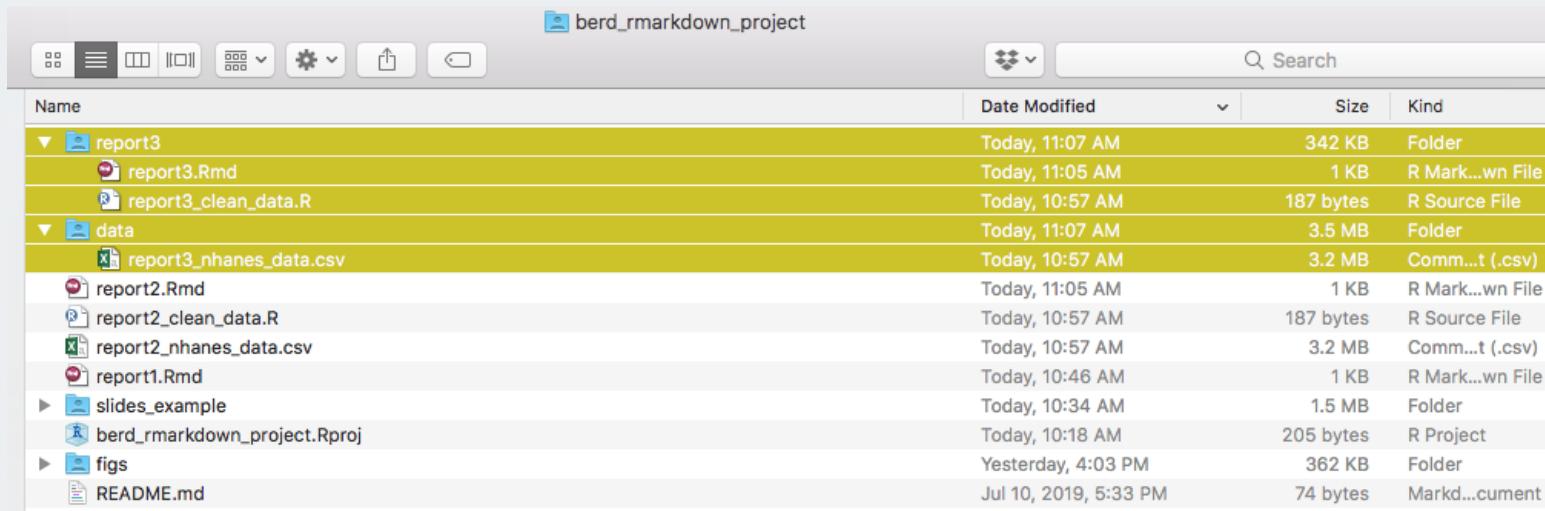
# Everything in one folder

| Name                         | Date Modified         | Size      | Kind                   |
|------------------------------|-----------------------|-----------|------------------------|
| report2.Rmd                  | Today, 11:05 AM       | 1 KB      | R Markdown File        |
| report2_clean_data.R         | Today, 10:57 AM       | 187 bytes | R Source File          |
| report2_nhances_data.csv     | Today, 10:57 AM       | 3.2 MB    | Comma-separated (.csv) |
| report1.Rmd                  | Today, 10:46 AM       | 1 KB      | R Markdown File        |
| slides_example               | Today, 10:34 AM       | 1.5 MB    | Folder                 |
| berd_rmarkdown_project.Rproj | Today, 10:18 AM       | 205 bytes | R Project              |
| figs                         | Yesterday, 4:03 PM    | 362 KB    | Folder                 |
| README.md                    | Jul 10, 2019, 5:33 PM | 74 bytes  | Markdown Document      |

After knitting, this gives you (file 🥑)

| Name                         | Date Modified         | Size      | Kind                   |
|------------------------------|-----------------------|-----------|------------------------|
| report2-figs                 | Today, 11:05 AM       | 402 KB    | Folder                 |
| report2.html                 | Today, 11:05 AM       | 712 KB    | HTML                   |
| report2.Rmd                  | Today, 11:05 AM       | 1 KB      | R Markdown File        |
| report2-output               | Today, 10:58 AM       | 2.7 MB    | Folder                 |
| report2_clean_data.R         | Today, 10:57 AM       | 187 bytes | R Source File          |
| report2_nhances_data.csv     | Today, 10:57 AM       | 3.2 MB    | Comma-separated (.csv) |
| report1.Rmd                  | Today, 10:46 AM       | 1 KB      | R Markdown File        |
| slides_example               | Today, 10:34 AM       | 1.5 MB    | Folder                 |
| berd_rmarkdown_project.Rproj | Today, 10:18 AM       | 205 bytes | R Project              |
| figs                         | Yesterday, 4:03 PM    | 362 KB    | Folder                 |
| README.md                    | Jul 10, 2019, 5:33 PM | 74 bytes  | Markdown Document      |

# Slightly more organized



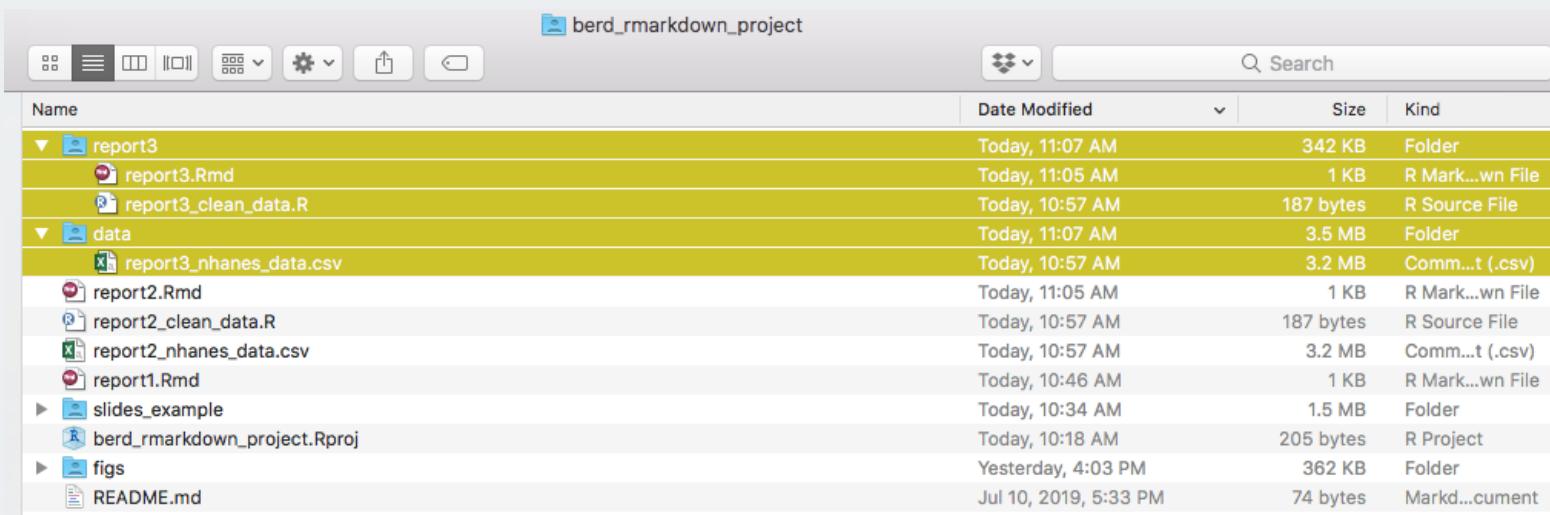
The screenshot shows a file explorer window titled "berd\_rmarkdown\_project". The interface includes a toolbar with icons for file operations like New, Open, Save, and Cut/Paste. A search bar is at the top right. The main area displays a hierarchical list of files and folders:

| Name                         | Date Modified         | Size      | Kind                   |
|------------------------------|-----------------------|-----------|------------------------|
| report3                      | Today, 11:07 AM       | 342 KB    | Folder                 |
| report3.Rmd                  | Today, 11:05 AM       | 1 KB      | R Markdown File        |
| report3_clean_data.R         | Today, 10:57 AM       | 187 bytes | R Source File          |
| data                         | Today, 11:07 AM       | 3.5 MB    | Folder                 |
| report3_nhances_data.csv     | Today, 10:57 AM       | 3.2 MB    | Comma-separated (.csv) |
| report2.Rmd                  | Today, 11:05 AM       | 1 KB      | R Markdown File        |
| report2_clean_data.R         | Today, 10:57 AM       | 187 bytes | R Source File          |
| report2_nhances_data.csv     | Today, 10:57 AM       | 3.2 MB    | Comma-separated (.csv) |
| report1.Rmd                  | Today, 10:46 AM       | 1 KB      | R Markdown File        |
| slides_example               | Today, 10:34 AM       | 1.5 MB    | Folder                 |
| berd_rmarkdown_project.Rproj | Today, 10:18 AM       | 205 bytes | R Project              |
| figs                         | Yesterday, 4:03 PM    | 362 KB    | Folder                 |
| README.md                    | Jul 10, 2019, 5:33 PM | 74 bytes  | Markdown Document      |

# Dot dot: A tip about "moving up" a directory/folder

- In unix, to point to the folder one level up (it contains the folder you're in), use .. or ../
  - As in `cd ..` moves up one directory,
  - or `cp ../myfile.txt newfile.txt` copies a file one level up into the current folder (working directory)
- In .Rmd when you want to source the data in the `data/` folder, you could use .. to move up a folder into the main directory, and then back down into the `data/` folder:

```
From the .Rmd folder, move up one folder then down to the data folder
mydata <- read_csv("../data/report3_nhances_data.csv")
```



The screenshot shows a Mac OS X Finder window titled "berd\_rmarkdown\_project". The sidebar on the left lists the project structure: "report3" (containing "report3.Rmd", "report3\_clean\_data.R"), "data" (containing "report3\_nhances\_data.csv"), "report2.Rmd", "report2\_clean\_data.R", "report2\_nhances\_data.csv", "report1.Rmd", "slides\_example", "berd\_rmarkdown\_project.Rproj", "figs", and "README.md". The main pane displays a file list with columns for Name, Date Modified, Size, and Kind. The "data" folder and its contents are highlighted in yellow.

| Name                         | Date Modified         | Size      | Kind                |
|------------------------------|-----------------------|-----------|---------------------|
| report3                      | Today, 11:07 AM       | 342 KB    | Folder              |
| report3.Rmd                  | Today, 11:05 AM       | 1 KB      | R Markdown File     |
| report3_clean_data.R         | Today, 10:57 AM       | 187 bytes | R Source File       |
| data                         | Today, 11:07 AM       | 3.5 MB    | Folder              |
| report3_nhances_data.csv     | Today, 10:57 AM       | 3.2 MB    | CSV Document (.csv) |
| report2.Rmd                  | Today, 11:05 AM       | 1 KB      | R Markdown File     |
| report2_clean_data.R         | Today, 10:57 AM       | 187 bytes | R Source File       |
| report2_nhances_data.csv     | Today, 10:57 AM       | 3.2 MB    | CSV Document (.csv) |
| report1.Rmd                  | Today, 10:46 AM       | 1 KB      | R Markdown File     |
| slides_example               | Today, 10:34 AM       | 1.5 MB    | Folder              |
| berd_rmarkdown_project.Rproj | Today, 10:18 AM       | 205 bytes | R Project           |
| figs                         | Yesterday, 4:03 PM    | 362 KB    | Folder              |
| README.md                    | Jul 10, 2019, 5:33 PM | 74 bytes  | Markdown Document   |

# Exercises:

Within your project folder, open these files and follow the instructions:

- report2.Rmd
- report3/report3.Rmd

# Find the .. confusing? Use here::here()!



Allison Horst

# here::here() → relative paths to the project directory

- The `here` package's `here()` function solves this issue of inconsistent working directories.
- The point of RStudio project workflow is to always have the same "home" working directory = where the `.Rproj` file is.
- `here::here()` returns the project directory as a string
- Fully reproducible if the whole folder is moved or shared or posted to github
- Portable to ALL systems (Mac, PC, unix), don't worry about / (Mac) or \ (PC) or spaces etc

```
here::here()
```

```
[1] "/Users/minnier/Google Drive/BERD R Classes/berd_r_courses_github"
```

# here::here() with folders and filenames

- `here::here("folder", "filename")` returns the entire file path as a string
- These file paths work when running a `.Rmd` file interactively like a notebook, when knitting it, when copying it to the console, wherever, whenever!!

```
here::here("data", "mydatafile.csv")
```

```
[1] "/Users/minnier/Google Drive/BERD R Classes/berd_r_courses_github/data/mydatafile.csv"
```

```
here::here("data", "raw-data", "mydatafile.csv")
```

```
[1] "/Users/minnier/Google Drive/BERD R Classes/berd_r_courses_github/data/raw-data/mydatafi
```

We will explore how and when to use this in the exercises.

# Exercises:

Within your project folder, open this file and follow the instructions:

- report3-here/report3\_here.Rmd

# Even more organized: child documents

If you want to have separate .Rmd files that are sourced in one large document, you can have "child document chunks":

A file called `report_prelim.Rmd` in the `analysis/` folder

(No YAML):

```
Details about experiment

Here are some details.
I can make a plot, too.

```{r plotstuff}
plot(x,y)
```
```

In the main doc `main_doc.Rmd`

```

title: "Main Report"
output: html_document

Preliminary Analysis
```{r child = here("analysis","report_prelim.Rmd")}

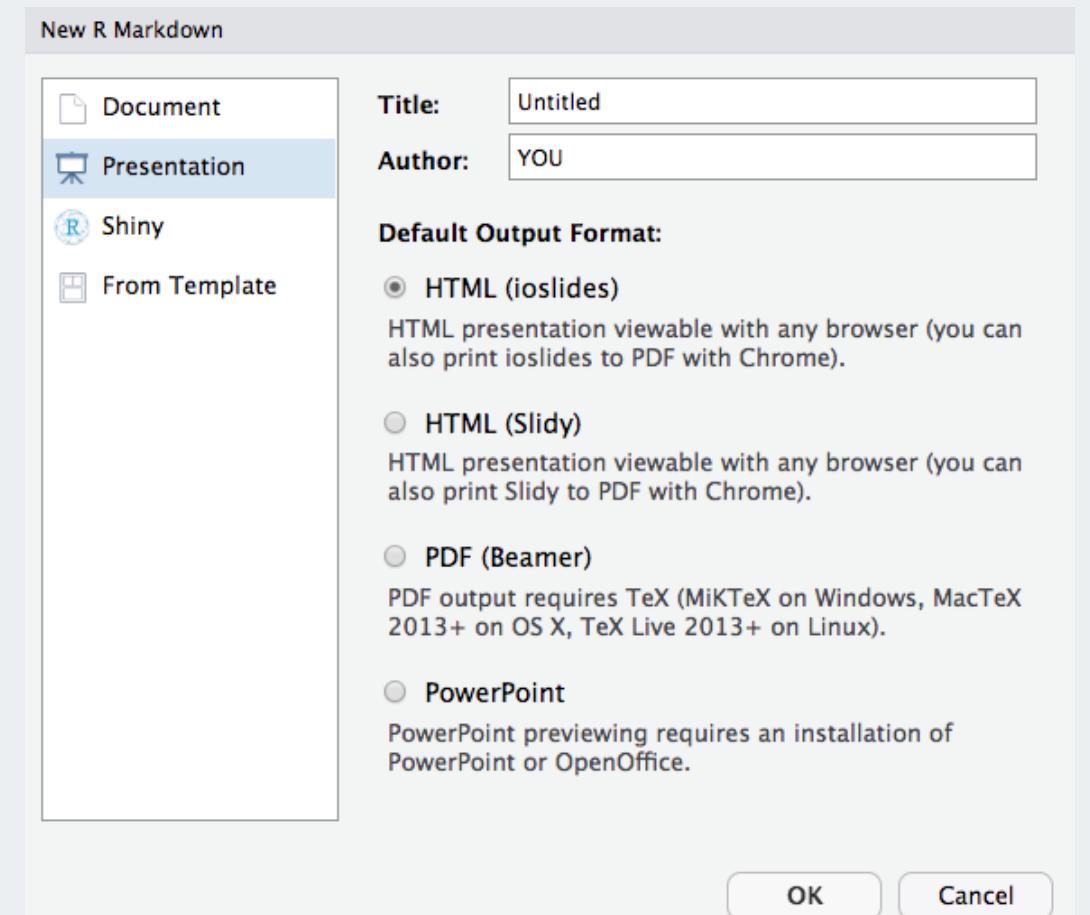
```

Conclusion
```{r}
kable(summarytable)
```
```

# More Extensions and Tips

# Make presentation slides

- These slides were made using a `.Rmd` file with the `xaringan` package!
- Simple templates can be found in `File -> new File -> R Markdown -> Presentation`
- Each type of presentation uses different syntax to start a new slide, such as
  - `# Slide Header`, or
  - `---`
- `ioslides` and `Slidy` are html slides; simple options
- `Beamer` is from LaTeX
- `Xaringan` is html based on java script `remark.js`; has the most flexibility for customizing slides
- `PowerPoint` is in the newest RStudio release; can use custom templates



# Presentations Exercise

Open `report2_pres.Rmd` and follow instructions.

Bonus: Try using `xaringan::inf_mr()` to update the output in real time.

# Tabs

A nice feature for showing multiple images or sections is with [tabbed sections](#):

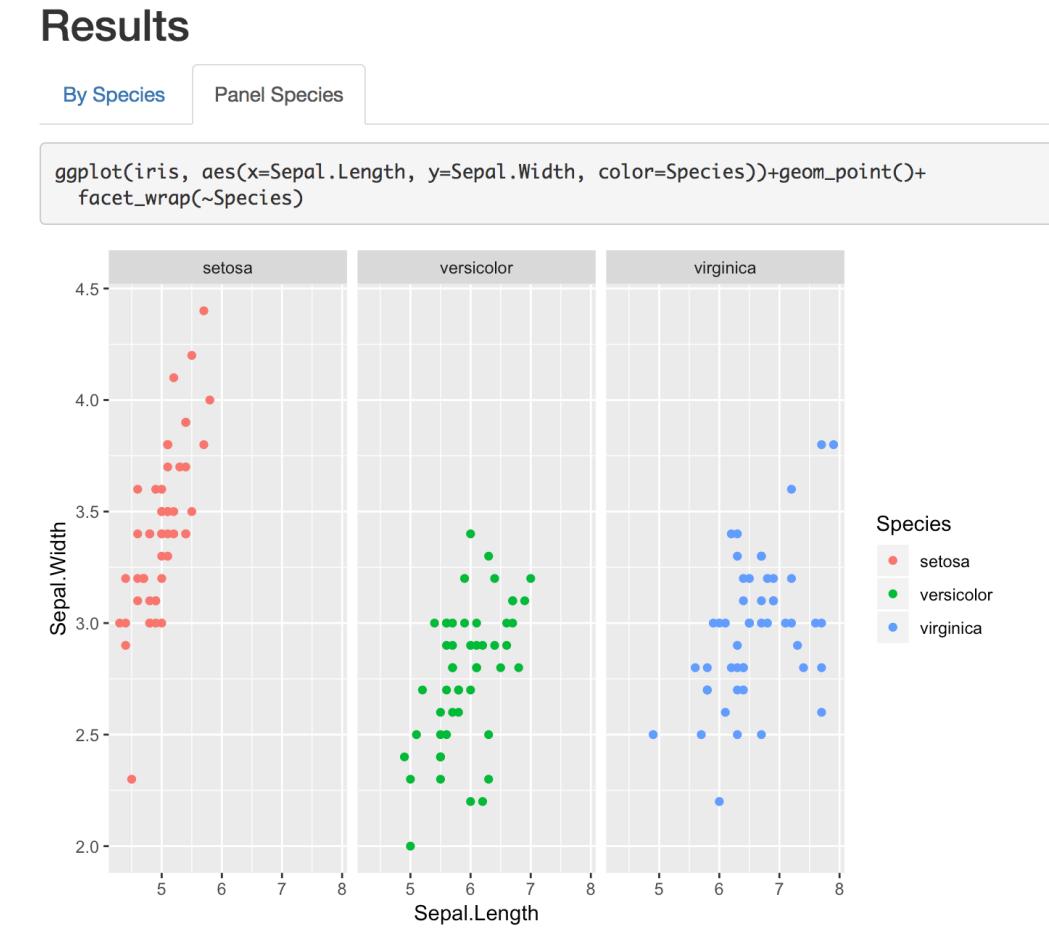
```
Results {.tabset}

By Species

```{r}  
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width,  
geom_point())  
```  

Panel Species

```{r}  
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width,  
geom_point())+  
  facet_wrap(~Species)  
```
```



# Using other programming languages

- RStudio can run multiple programming languages in the same .Rmd (if they are installed on the computer), including SAS, STATA, and python.
- For more on how to use STATA and SAS, for example, see the documentation for these packages:
  - [StataMarkdown](#)
  - [SASMarkdown](#)

```
names(knitr::knit_engines$get())
```

```
[1] "awk" "bash" "coffee" "gawk" "groovy"
[6] "haskell" "lein" "mysql" "node" "octave"
[11] "perl" "psql" "Rscript" "ruby" "sas"
[16] "scala" "sed" "sh" "stata" "zsh"
[21] "highlight" "Rcpp" "tikz" "dot" "c"
[26] "fortran" "fortran95" "asy" "cat" "asis"
[31] "stan" "block" "block2" "js" "css"
[36] "sql" "go" "python" "julia" "sass"
[41] "scss"
```

# Other languages: Limitations

- Each code chunk is run separately as a batch job when using other languages, so it's tricky to pass on objects/data to later code chunks.
- Easy way:
  - Use one language to clean data & save the cleaned data as a file
  - source the file and continue in another language.
- Other packages can be loaded that help to link objects from various languages, i.e.
  - [reticulate](#) can store objects created by python code for use in R
  - StataMarkdown and SASMarkdown use chunk option `collectcode=TRUE` to save code output.

```
```{r setup}
library(SASmarkdown)
```

```{sas clean_data, collectcode=TRUE}
/* clean data with SAS code */
/* export to file */
```

```{sas analyze_data}
/* analyze data from above code */
```

```{r analyze_data}
# source clean data file and run code
```

```

# Knit other types of output

- Journal articles, custom [templates](#)
  - File → New File → R Markdown → From template
- Dashboards: [flexdashboard](#) report output
- Interactive reports with [shiny](#)
- Interactive tutorials with [learnr](#)
- Websites: [blogdown](#)
- Books: [bookdown](#)
- Posters: [posterdown](#)
- Grad school theses: [thesisdown](#)
- It's really endless....

# rmarkdown::render()

It can sometimes be easier to set options and change output files/locations when using the `render()` function in the `rmarkdown` package. This is also useful for rendering multiple documents in a batch, or using [parameterized reports](#).

In a `.R` file, or in the console, run commands to knit the documents:

```
library(rmarkdown)
render("report1.Rmd")

Render in a directory
render(here::here("report3","report3.Rmd"))

Render a single format
render("report1.Rmd", output_format = "html_document")

Render multiple formats
render("report1.Rmd", output_format = c("html_document", "pdf_document"))

Render to a different file name or folder
render("report1.Rmd",
 output_format = "html_document",
 output_file = "output/report1_2019_07_18.html")
```

# knitr::purl() → .R file

Run in the console or keep in a separate R file to extract all the R code into a **.R** file.

```
makes an R file report1.R in same director
knitr::purl("report1.Rmd")

Can be more specific with output
knitr::purl(here::here("report3","report3.Rmd"), # Rmd location
 out = here::here("report3","report3_code_only.R")) # R output location
```

# knitr::knit\_exit(): End document early

- Exit the document early.
- Place this in your `.Rmd` to end document there and ignore the rest.
- Run parts of the document at a time

```
```{r}
knitr::knit_exit()
```
```

# Parameterized Reports

```

```

```
title: My Report
output: html_document
params:
 data: file.csv
 printcode: TRUE
 year: 2018
```

```

```

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

```{r}
mydata <- read_csv(params$data)
mydata <- mydata %>%
  filter(year==params$year)
```

```

- Use the Knit button and you will be prompted for values
- Use `rmarkdown::render` (default values are set in YAML)
- See [chapter in R Markdown book](#) for details

```
rmarkdown::render(
 "myreport.Rmd",
 params =
 list(data = "newfile.csv",
 year = "2019",
 printcode = FALSE),
 output_file = "report2019_newfile.html"
)
```

# Many more bonus tips

- Use [git](#) and [github](#) for version control, and use output format [github\\_document](#) - see an [example](#)
- Quickly convert `.R` files to `.html` with the [notebook/compile button](#) or `knitr::spin()`
- Include [HTML headers](#) or Latex [preambles](#) and files for definitions in YAML
- Add references and a [bibliography](#) with BibTex `.bib` files
- Similar to `.Rmd` are RStudio "notebooks" -- like an `.Rmd` but all the output is saved as it is run in the notebook.
- Publish rendered html on [Rpubs](#) with Publish button, or through [github + netlify](#).
- [Look at these slides by Alison Hill](#) and [these by Yihui Xie](#) for many, many more tips and examples

# References

- RStudio's R Markdown lessons
- Xie Y. et al R Markdown: The Definitive Guide book online
- Explanation of difference between knitr/Rmd/pandoc
- Teach data science: Getting started with R Markdown
- Alison Hill & Yihui Xie's Advanced R Markdown Workshop Materials
- UCLA's Intro to R Markdown slides
- Software Carpentry Learning R Markdown Materials

## Cheatsheets:

- R Markdown cheatsheet
- R Markdown reference guide

# Possible Future Workshop Topics?

- tables
- ggplot2 visualization
- advanced tidyverse: functions, purrr (apply/map)
- statistical modeling in R

## Contact info:

- Jessica Minnier: *minnier@ohsu.edu*
- Meike Niederhausen: *niederha@ohsu.edu*

## This workshop info:

- Code for these slides are on github, with links to other course materials:  
[jminnier/berd\\_r\\_courses](https://jminnier/berd_r_courses)
- The `.Rmd` file that generated the slides is on [github](#) and can be downloaded [here](#), though you need to download the whole [R project](#) to knit the file.
- The project folder of examples can be downloaded at  
[github.com/jminnier/berd\\_rmarkdown\\_project](https://github.com/jminnier/berd_rmarkdown_project) & the solutions are in the `sols`/ folder.