R Markdown

Improving reproducibility using literate programming

Russell Goebel
2022-01-20

Outline

- Motivation: Why use R Markdown?
- History of R Markdown
- Introduction to R Markdown
- Using R Markdown
    - Bibliographies/Citations
    - Cross-referencing within documents
    - The **rticles** package and journal templates
    - R package vignettes
    - Presentation Slides (Like these!)

Why use R Markdown?

nature ecology & evolution

PERSPECTIVE

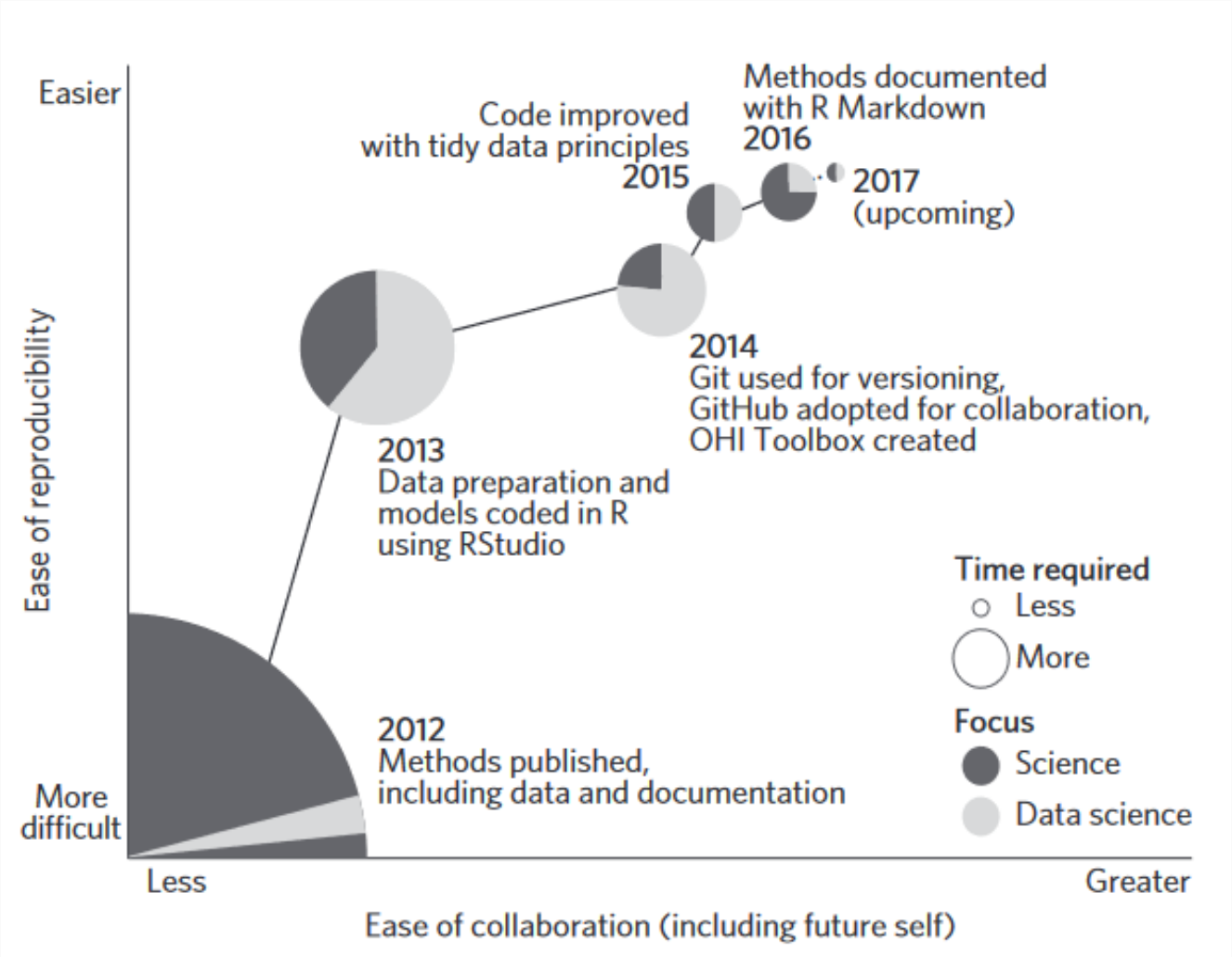
PUBLISHED: 23 MAY 2017 | VOLUME: 1 | ARTICLE NUMBER: 0160

Our path to better science in less time using open data science tools

Julia S. Stewart Lowndes[1*], Benjamin D. Best[2], Courtney Scarborough[1], Jamie C. Afflerbach[1], Melanie R. Frazier[1], Casey C. O'Hara[1], Ning Jiang[1] and Benjamin S. Halpern[1,3,4]

Reproducibility has long been a tenet of science but has been challenging to achieve—we learned this the hard way when our old approaches proved inadequate to efficiently reproduce our own work. Here we describe how several free software tools have fundamentally upgraded our approach to collaborative research, making our entire workflow more transparent and streamlined. By describing specific tools and how we incrementally began using them for the Ocean Health Index project, we hope to encourage others in the scientific community to do the same—so we can all produce better science in less time.

Why use R Markdown?



Why use R Markdown?

Sharing methods and instruction. We use R Markdown not only for data preparation but also for broader communication. R Markdown files can be generated into a wide variety of formatted outputs, including PDFs, slides, Microsoft Word documents, HTML files, books or full websites[61,62]. These can all be published online for free through GitHub using the same RStudio–GitHub workflow that we use for our analyses, which has made communication an ongoing part of our work, instead of a final step in completed analyses.

We built a website using GitHub and RStudio publishing tools: http://ohi-science.org. Team members can update content directly, and using the same workflow makes it easier for us to keep it current. Our website is intended for scientists interested in our meth-

A brief history of R Markdown


Donald
Knuth '60
50th Reunion

Literate Programming

"Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do."

-Donald Ervin Knuth, Literate Programming

The Software Comprehension Problem

- **Maintenance:** How can we reduce time spent understanding code?
- **Motivation:** How can we motivate programmers to document well?

Literate programs are "psychologically **arranged for comprehension** by humans".

They "provide significant incentives for programmers to **document** their understanding."

...For our purposes, literate programming encourages **reproducible research!**

Quotes are from **Towards Modern Literate Programming** by Smith and Churcher.

Early example: Donald Knuth's WEB

March 22, 2002 at 22:33

1. Introduction. The ur-game for computers — Adventure — was originally written by Will Crowther in 1975 and greatly extended by Don Woods in 1976. I have taken Woods's original FORTRAN program for Adventure Version 1.0 and recast it in the CWEB idiom.

I remember being fascinated by this game when John McCarthy showed it to me in 1977. I started with no clues about the purpose of the game or what I should do; just the computer's comment that I was at the end of a forest road facing a small brick building. Little by little, the game revealed its secrets, just as its designers had cleverly plotted. What a thrill it was when I first got past the green snake! Clearly the game was potentially addictive, so I forced myself to stop playing — reasoning that it was great fun, sure, but traditional computer science research is great fun too, possibly even more so.

Now here I am, 21 years later, returning to the great Adventure after having indeed had many exciting adventures in Computer Science. I believe people who have played this game will be able to extend their fun by reading its once-secret program. Of course I urge everybody to *play the game first, at least ten times*, before reading on. But you cannot fully appreciate the astonishing brilliance of its design until you have seen all of the surprises that have been built in.

I believe this program is entirely faithful to the behavior of Adventure Version 1.0, except that I have slightly edited the computer messages (mostly so that they use both lowercase and uppercase letters). I have also omitted Woods's elaborate machinery for closing the cave during the hours of prime-time computing; I believe John McCarthy insisted on this, when he saw the productivity of his AI Lab falling off dramatically (although it is rumored that he had a special version of the program that allowed him to play whenever he wanted). And I have not adopted the encryption scheme by which Woods made it difficult for users to find any important clues in the binary program file or core image; such modifications would best be done by making a special version of CTANGLE. All of the spelunking constraints and interactive behavior have been retained, although the structure of this CWEB program is naturally quite different from the FORTRAN version I began with.

Many of the phrases in the following documentation have been lifted directly from comments in the

Early example: Donald Knuth's WEB

123. This'll teach you a lesson.

```
⟨ Throw the axe at the bear 123 ⟩ ≡
  {
    drop(AXE, loc);
    prop[AXE] = 1; base[AXE] = AXE;      /* it becomes immovable */
    if (place[BEAR] ≡ loc) move(BEAR, loc);      /* put bear first in its list */
    report("The axe misses and lands near the bear where you can't get at it");
  }
```

This code is used in section 122.

124. If you toss the vase, the skillful troll will catch it before it breaks.

```
⟨ Snarf a treasure for the troll 124 ⟩ ≡
  {
    drop(obj, limbo);
    destroy(TROLL); destroy(TROLL_);
    drop(TROLL2, swside); drop(TROLL2_, neside);
    move(BRIDGE, swside); move(BRIDGE_, neside);
    report("The troll catches your treasure and scurries away out of sight");
  }
```

This code is used in section 122.

125. When you try to attack, the action becomes violent.

```
⟨ Handle cases of transitive verbs and continue 97 ⟩ +≡
case KILL: if (obj ≡ NOTHING) ⟨ See if there's a unique object to attack 126 ⟩;
  switch (obj) {
  case 0: report("There is nothing here to attack.");
```

Early example: Donald Knuth's WEB

123. This'll teach you a lesson.

```
⟨ Throw the axe at the bear 123 ⟩ ≡
  {
    drop(AXE, loc);
    prop[AXE] = 1; base[AXE] = AXE;      /* it becomes immovable */
    if (place[BEAR] ≡ loc) move(BEAR, loc);      /* put bear first in its list */
    report("The axe misses and lands near the bear where you can't get at it");
  }
```

This code is used in section 122.

124. If you toss the vase, the skillful troll will catch it before it breaks.

```
⟨ Snarf a treasure for the troll 124 ⟩ ≡
  {
    drop(obj, limbo);
    destroy(TROLL); destroy(TROLL_);
    drop(TROLL2, swside); drop(TROLL2_, neside);
    move(BRIDGE, swside); move(BRIDGE_, neside);
    report("The troll catches your treasure and scurries away out of sight");
  }
```

This code is used in section 122.

125. When you try to attack, the action becomes violent.

```
⟨ Handle cases of transitive verbs and continue 97 ⟩ +≡
case KILL: if (obj ≡ NOTHING) ⟨ See if there's a unique object to attack 126 ⟩;
  switch (obj) {
  case 0: report("There is nothing here to attack.");
```

Early example: Donald Knuth's WEB

201. Index. A large cloud of green smoke appears in front of you. It clears away to reveal a tall wizard, clothed in grey. He fixes you with a steely glare and declares, "This adventure has lasted too long." With that he makes a single pass over you with his hands, and everything around you fades away into a grey nothingness.

ADVENTURE

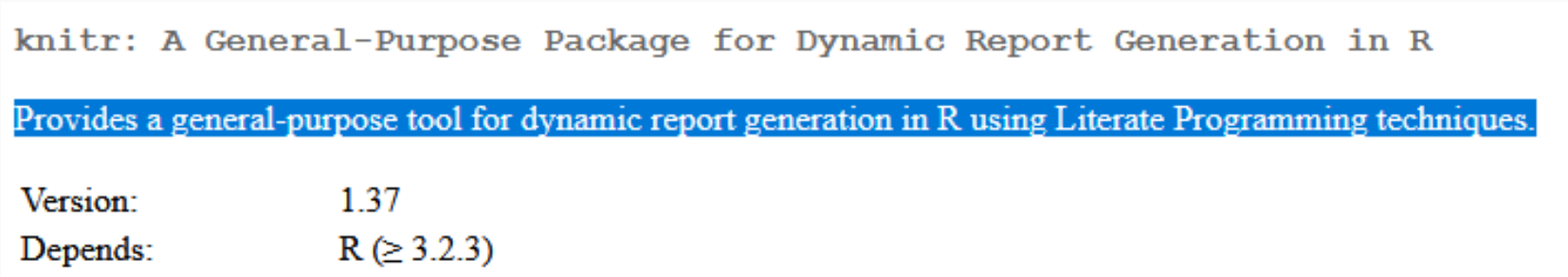
What is R Markdown?

Geneology of R Markdown

- **knitr**
- **Markdown**
- **Pandoc**

knitr

```
knitr::include_graphics("./images/knitr_package.PNG")
```



knitr: A General-Purpose Package for Dynamic Report Generation in R

Provides a general-purpose tool for dynamic report generation in R using Literate Programming techniques.

Version:	1.37
Depends:	R (≥ 3.2.3)

(from https://cran.r-project.org/web/packages/knitr/index.html)

knitr

```
knitr::include_graphics("./images/knitr_overview.PNG")
```

knitr

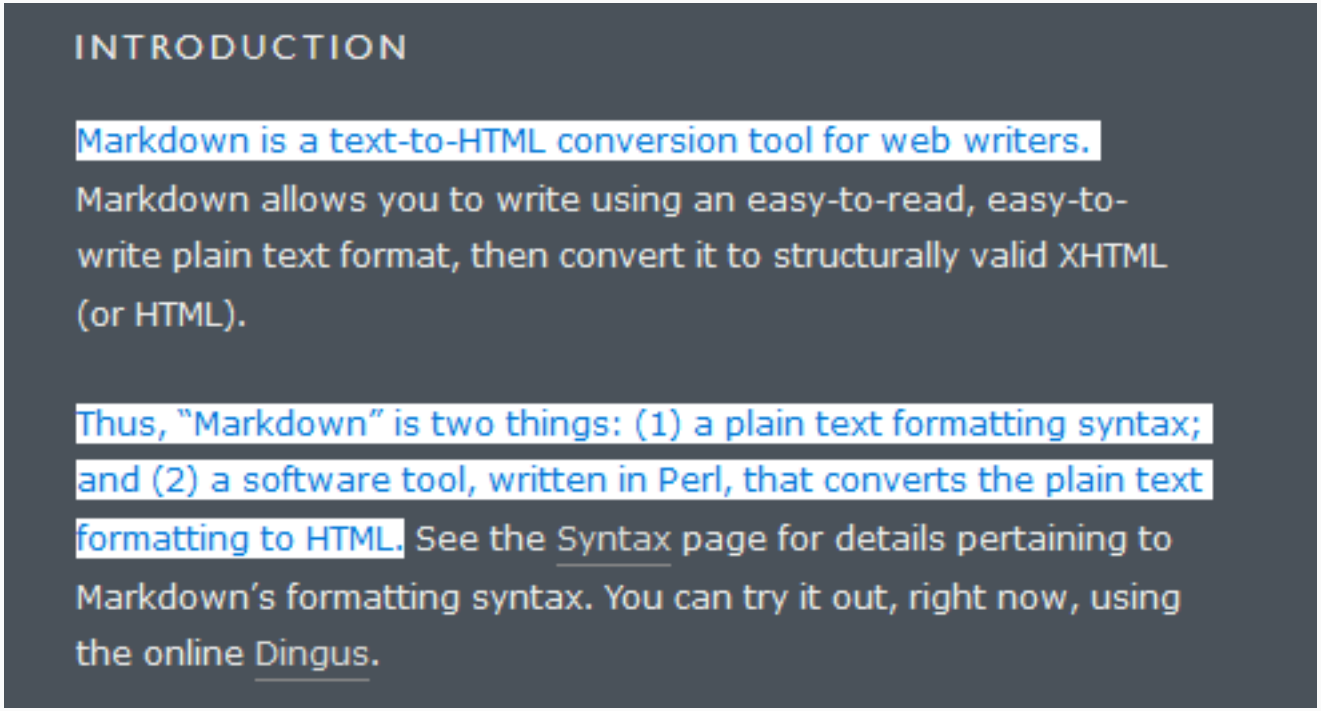
Elegant, flexible, and fast dynamic report generation with R

Overview

Inspired by Sweave, the **knitr** package was designed to be a transparent engine for dynamic report generation with R, and combine features in other add-on packages into one package (**knitr** ≈ Sweave + cacheSweave + pgfSweave + weaver + `animation::saveLatex()` + `R2HTML::RweaveHTML()` + `highlight::HighlightWeaveLatex()` + 0.2 * brew + 0.1 * SweaveListingUtils + more).

(from https://yihui.org/knitr/)

Markdown



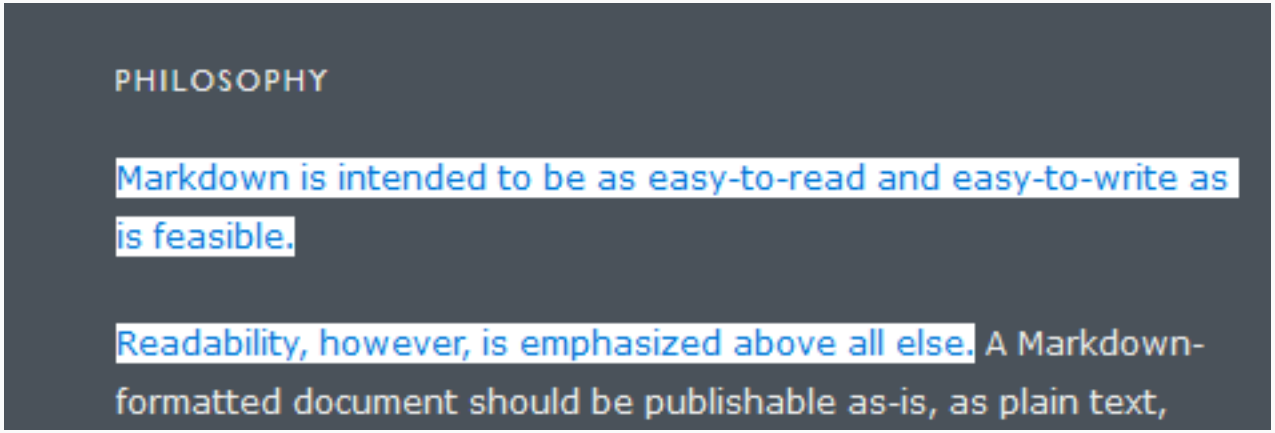
INTRODUCTION

Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

Thus, "Markdown" is two things: (1) a plain text formatting syntax; and (2) a software tool, written in Perl, that converts the plain text formatting to HTML. See the Syntax page for details pertaining to Markdown's formatting syntax. You can try it out, right now, using the online Dingus.

(from https://daringfireball.net/projects/markdown/)

Markdown



PHILOSOPHY

Markdown is intended to be as easy-to-read and easy-to-write as is feasible.

Readability, however, is emphasized above all else. A Markdown-formatted document should be publishable as-is, as plain text,

(from https://daringfireball.net/projects/markdown/syntax)

Pandoc

(from https://pandoc.org/index.html)

Summary so far

R Markdown...

- ... aids in **reproducibility** and **collaboration**
- ... is an implementation of **literate programming**
- ... originated in the **knitr** package
  - embeds code chunks in in **Markdown documents**
- ... uses **Pandoc** to convert Markdown documents to many output formats

> In a nutshell, R Markdown stands on the shoulders of **knitr** and Pandoc. The former executes the computer code embedded in Markdown, and converts R Markdown to Markdown. The latter renders Markdown to the output format you want (such as PDF, HTML, Word, and so on).

(from https://bookdown.org/yihui/rmarkdown/)

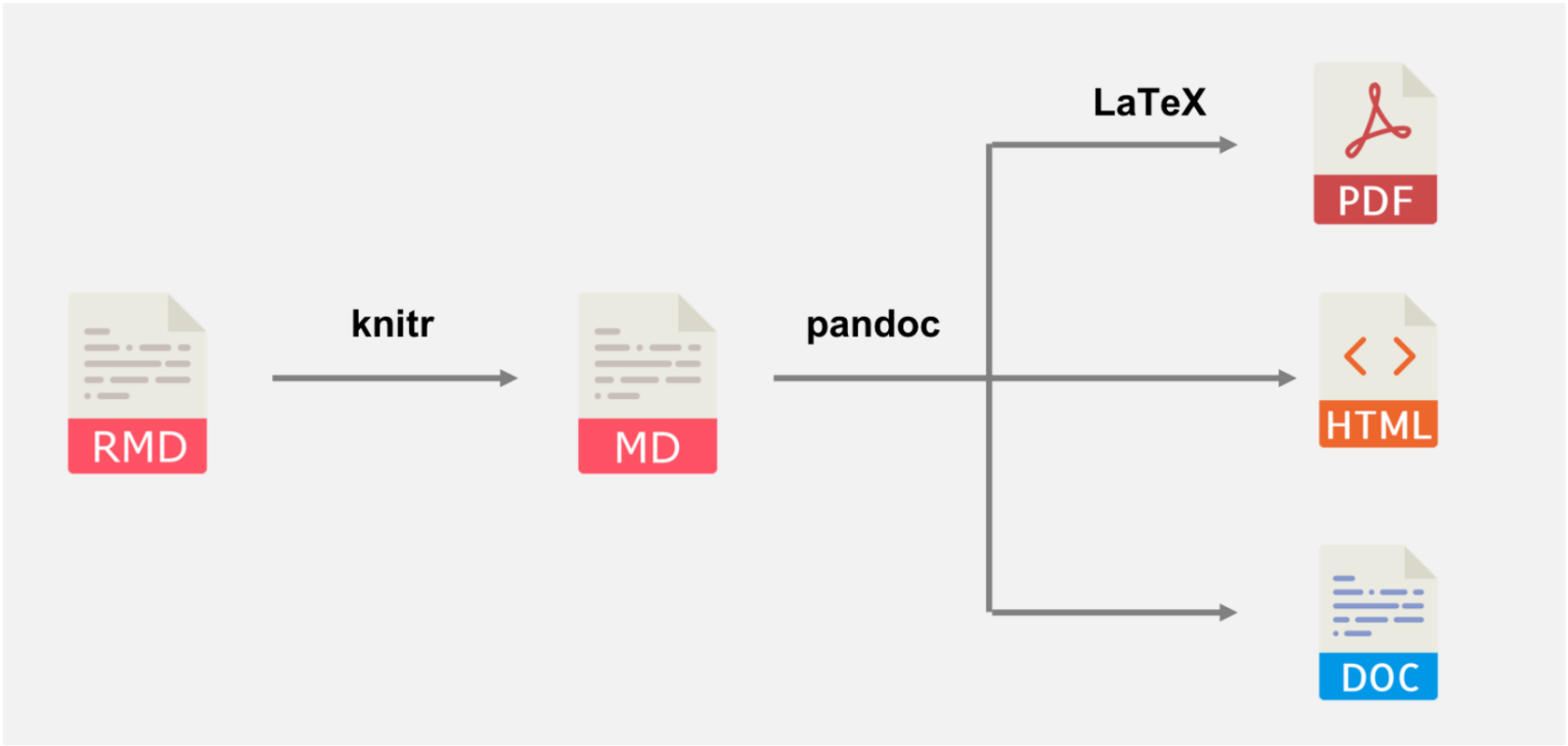
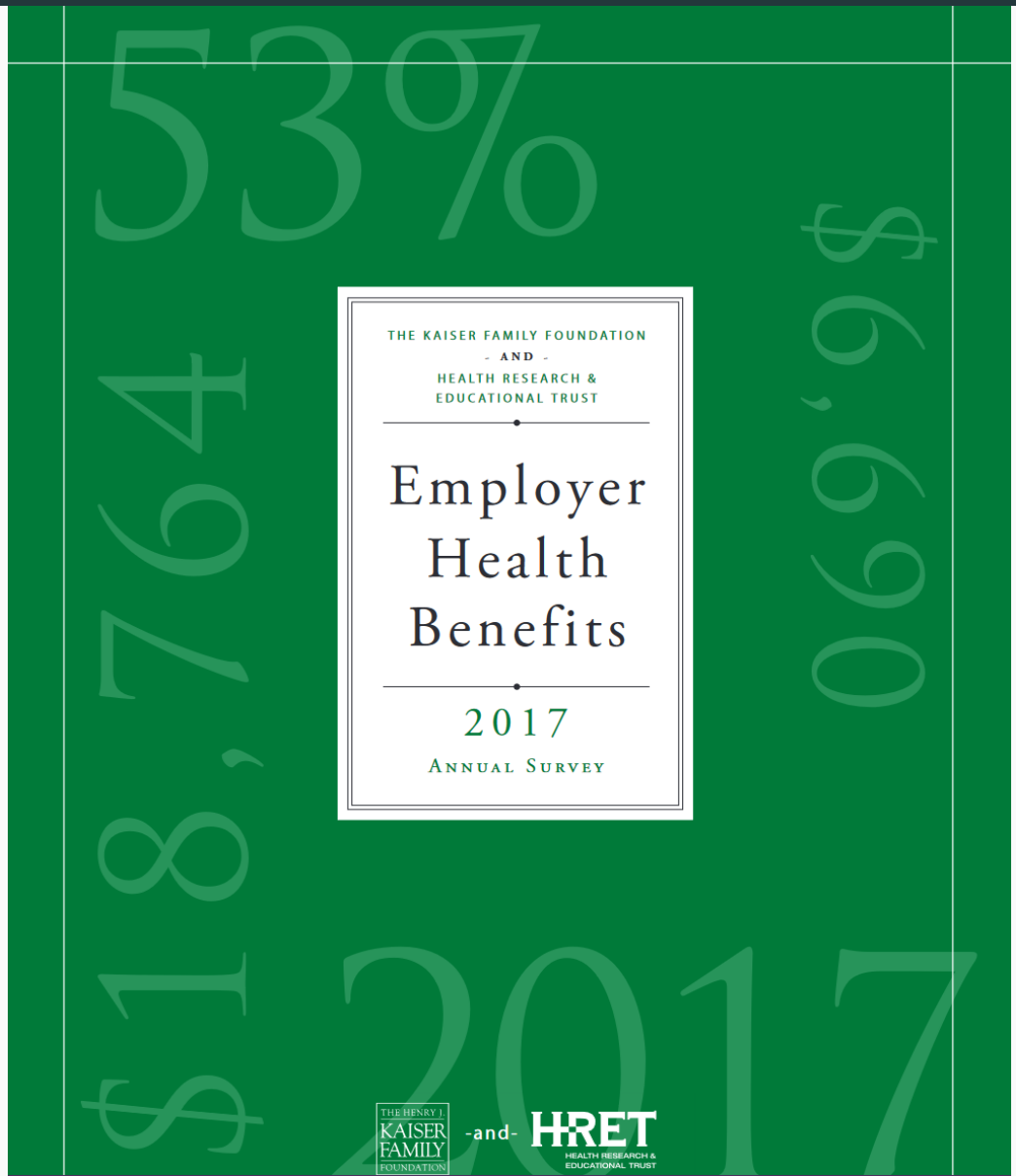


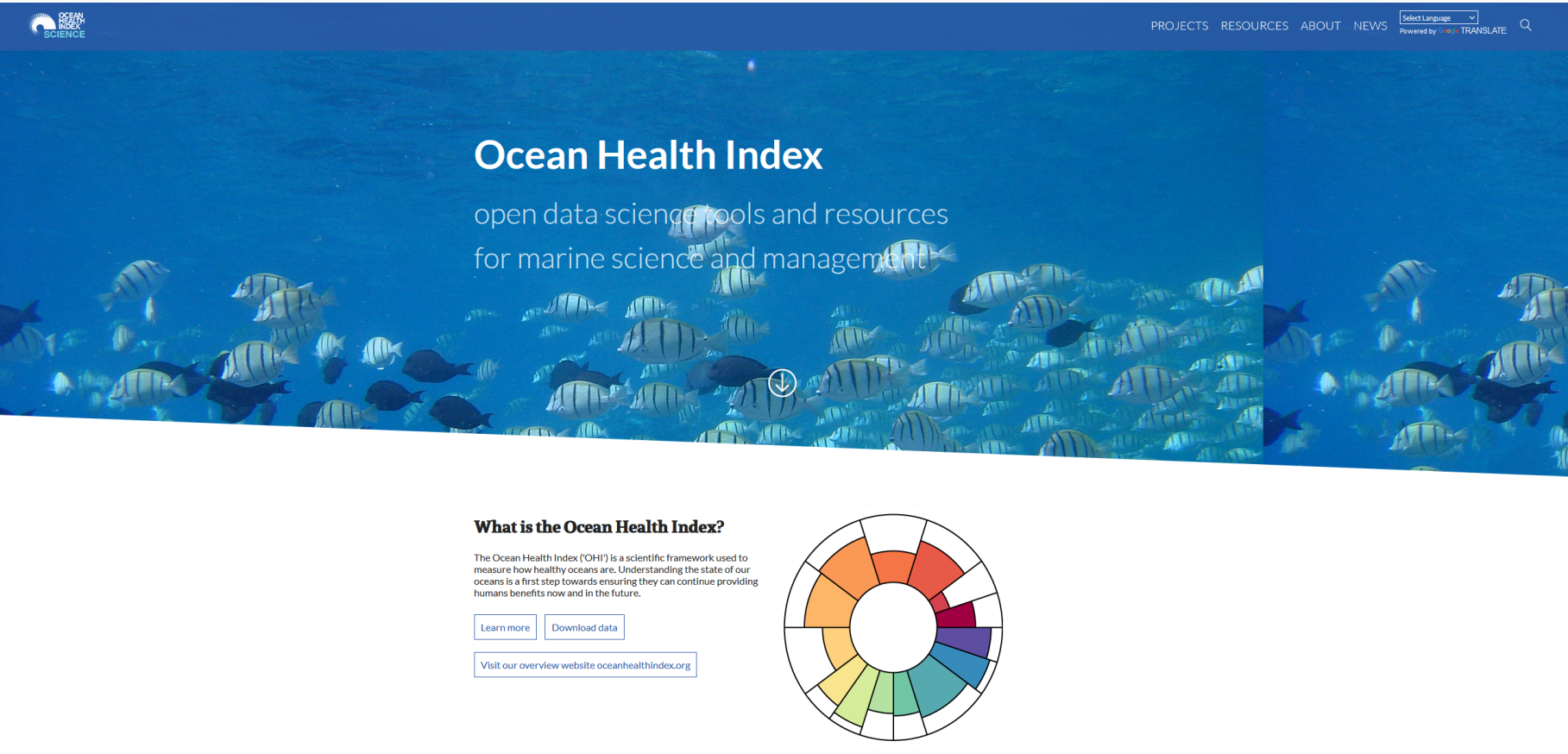
FIGURE 2.1: A diagram illustrating how an R Markdown document is converted to the final output document.

Examples

Books


$18,764
53%
$6,696
2017
THE KAISER FAMILY FOUNDATION
- AND -
HEALTH RESEARCH &
EDUCATIONAL TRUST
Employer
Health
Benefits
2017
Annual Survey
THE HENRY J.
KAISER
FAMILY
FOUNDATION
-and-
HRET
HEALTH RESEARCH &
EDUCATIONAL TRUST

Websites


OCEAN HEALTH INDEX SCIENCE
PROJECTS
RESOURCES
ABOUT
NEWS
Select Language
Powered by Google TRANSLATE
Ocean Health Index
open data science tools and resources
for marine science and management
What is the Ocean Health Index?
The Ocean Health Index ('OHI') is a scientific framework used to measure how healthy oceans are. Understanding the state of our oceans is a first step towards ensuring they can continue providing humans benefits now and in the future.
Learn more
Download data
Visit our overview website oceanhealthindex.org

Homework

In my opinion, this decision boundary has reasonably good aggreement with the data compared to the linear decision rule. This is reflected in considerably better misclassification rates, which improve from about 0.485 to 0.335. That is, the linear rule is barely better at classifying the training data than guessing, but this decision boundary is almost twice as likely to correctly classify trainining data than not.

(d)

In general, the strategy here is as follows: first, we must come up with an expression for

$$f_{Y|X}(y|x) = \frac{f_{X|Y}(x|y)f_Y(y)}{\sum_{y=0}^{1}\int_{\mathbb{R}} f_{X|Y}(x|y)f_Y(y)dx}.$$

For fixed x, we can identify which class y results in the maximum $f_{Y|X}(y)$. After identifying regions of such x for each class, we can integrate the joint density over each region to obtain the accuracy of Bayes classifier, which we can easily relate to the Bayes rate.

As a note, due to the symmetry of the normal distribution, we can guess that the decision boundary will be the axes. It is an interesting exercise to show this.

I think we need to make an additional assumption: that a point is equally likely to be generated from each center. To keep track of this, instead of working with Y it may be more convenient to work with a random variable C whose realizations are which center a point was generated from:

$$P(C = (x_1, x_2)) = 1/4, \qquad x_1, x_2 \in \{-1, 1\}.$$

ORIGINAL RESEARCH REPORT

Too Good to be False: Nonsignificant Results Revisited

C. H. J. Hartgerink, J. M. Wicherts and M. A. L. M. van Assen

Due to its probabilistic nature, Null Hypothesis Significance Testing (NHST) is subject to decision errors. The concern for false positives has overshadowed the concern for false negatives in the recent debates in psychology. This might be unwarranted, since reported statistically nonsignificant findings may just be 'too good to be false'. We examined evidence for false negatives in nonsignificant results in three different ways. We adapted the Fisher test to detect the presence of at least one false negative in a set of statistically nonsignificant results. Simulations show that the adapted Fisher method generally is a powerful method to detect false negatives. We examined evidence for false negatives in the psychology literature in three applications of the adapted Fisher method. These applications indicate that (i) the observed effect size distribution of nonsignificant effects exceeds the expected distribution assuming a null-effect, and approximately two out of three (66.7%) psychology articles reporting nonsignificant results contain evidence for at least one false negative, (ii) nonsignificant results on gender effects contain evidence of true nonzero effects, and (iii) the statistically nonsignificant replications from the Reproducibility Project Psychology (RPP) do not warrant strong conclusions about the absence or presence of true zero effects underlying these nonsignificant results. We conclude that false negatives deserve more attention in the current debate on statistical practices in psychology. Potentially neglecting effects due to a lack of statistical power can lead to a waste of research resources and stifle the scientific discovery process.

Keywords: NHST; reproducibility project; nonsignificant; power; underpowered; effect size; Fisher test; gender

Popper's (Popper, 1959) falsifiability serves as one of the main demarcating criteria in the social sciences, which stipulates that a hypothesis is required to have the possibility of being proven false to be considered scientific. Within the theoretical framework of scientific hypothesis testing, accepting or rejecting a hypothesis is unequivocal, because the hypothesis is either true or false. Statistical hypothesis testing, on the other hand, is a probabilistic operationalization of scientific hypothesis testing (Meehl, 1978) and, in lieu of its probabilistic nature, is subject to decision errors. Such decision errors are the topic of this paper.

Null Hypothesis Significance Testing (NHST) is the most prevalent paradigm for statistical hypothesis testing in the social sciences (American Psychological Association, 2010). In NHST the hypothesis H_0 is tested, where H_0 most often regards the absence of an effect. If deemed false, an alternative, mutually exclusive hypothesis H_1 is accepted. These decisions are based on the p-value; the probability of the sample data, or more extreme data, given H_0 is true. If the p-value is smaller than the decision criterion (i.e., α; typically .05; [Nuijten, Hartgerink, van Assen, Epskamp, & Wicherts, 2015]), H_0 is rejected and H_1 is accepted.

Table 1 summarizes the four possible situations that can occur in NHST. The columns indicate which hypothesis is true in the population and the rows indicate what is decided based on the sample data. When there is discordance between the true- and decided hypothesis, a decision error is made. More specifically, when H_0 is true in the population, but H_1 is accepted ('H_1'), a Type I error is made (α); a false positive (lower left cell). When H_1 is true in the population and H_0 is accepted ('H_0'), a Type II error is made (β); a false negative (upper right cell). However, when the null hypothesis is true in the population and H_0 is accepted ('H_0'), this is a true negative (upper left cell; $1 - \alpha$). The true negative rate is also called specificity of the test. Conversely, when the alternative hypothesis is true in the population and H_1 is accepted ('H_1'), this is a true positive (lower right cell). The probability of finding a statistically significant result if H_1 is true is the power ($1 - \beta$), which is also called the sensitivity of the test. Power is a positive function of the (true) population effect size, the sample size, and the alpha of the study, such that higher power can always be achieved by altering either the sample size or the alpha level (Aberson, 2010).

Unfortunately, NHST has led to many misconceptions and misinterpretations (e.g., Goodman, 2008; Bakan, 1966). The most serious mistake relevant to our paper

Department of Methodology and Statistics, Tilburg University, NL
Corresponding author: C. H. J. Hartgerink
(c.h.j.hartgerink@tilburguniversity.edu)

Introduction to R Markdown

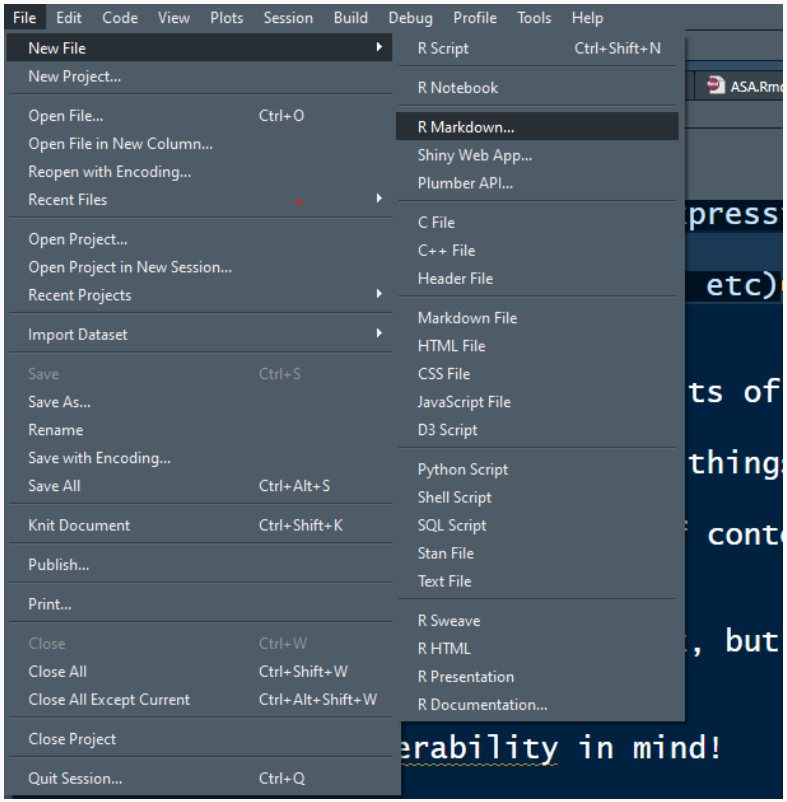
You will want R and R Studio.

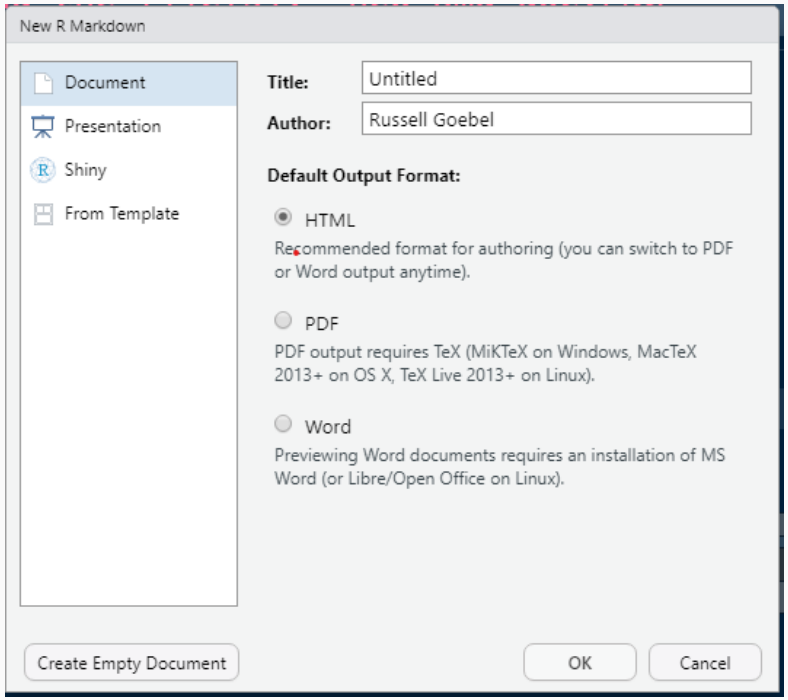
Installation

You will want to install the **rmarkdown** package in R using `install.packages('rmarkdown')` in R.

If you want to generate PDF output, you will need to install LaTeX. One recommendation is to install **TinyTeX** using `install.packages('tinytex')`.

Creating an R Markdown Document





An R Markdown Document

````
---
title: "Untitled"
author: "Russell Goebel"
date: "1/18/2022"
output: html_document
---

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

## R Markdown

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

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

```{r cars}
summary(cars)
```

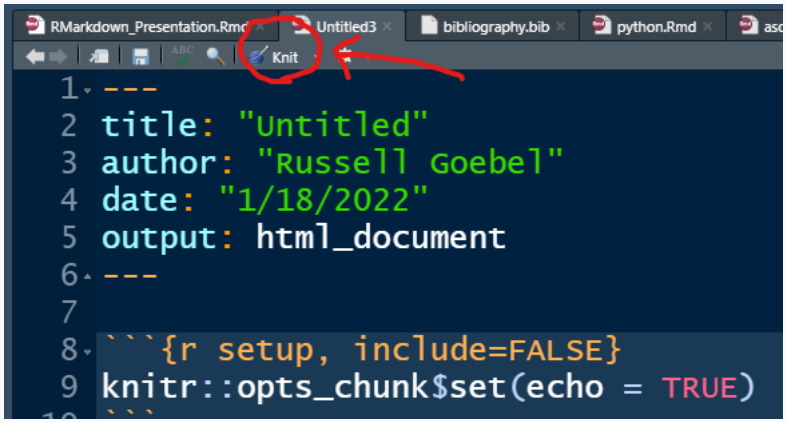
## Including Plots

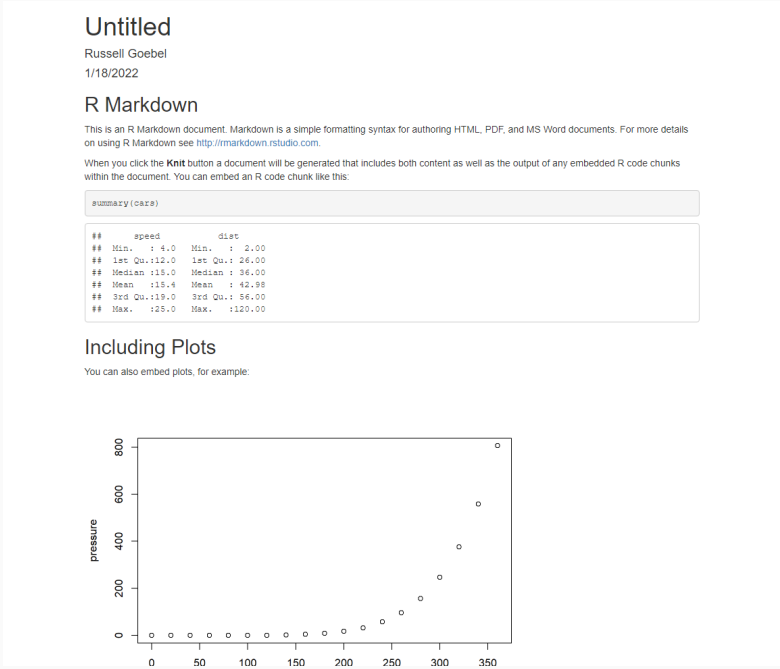
You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
````

Knit


RMarkdown_Presentation.Rmd
Untitled3
bibliography.bib
python.Rmd
Knit
1 ---
2 title: "Untitled"
3 author: "Russell Goebel"
4 date: "1/18/2022"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)


Untitled
Russell Goebel
1/18/2022
R Markdown
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.
When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
summary(cars)
speed dist
Min. : 4.0 Min. : 2.00
1st Qu.:12.0 1st Qu.: 26.00
Median :15.0 Median : 36.00
Mean :15.4 Mean : 42.98
3rd Qu.:19.0 3rd Qu.: 56.00
Max. :25.0 Max. :120.00
Including Plots
You can also embed plots, for example:
pressure
0
200
400
600
800
0
50
100
150
200
250
300
350

Important tip: use cheatsheets!

Cheatsheet

rmarkdown : : CHEAT SHEET

What is rmarkdown?

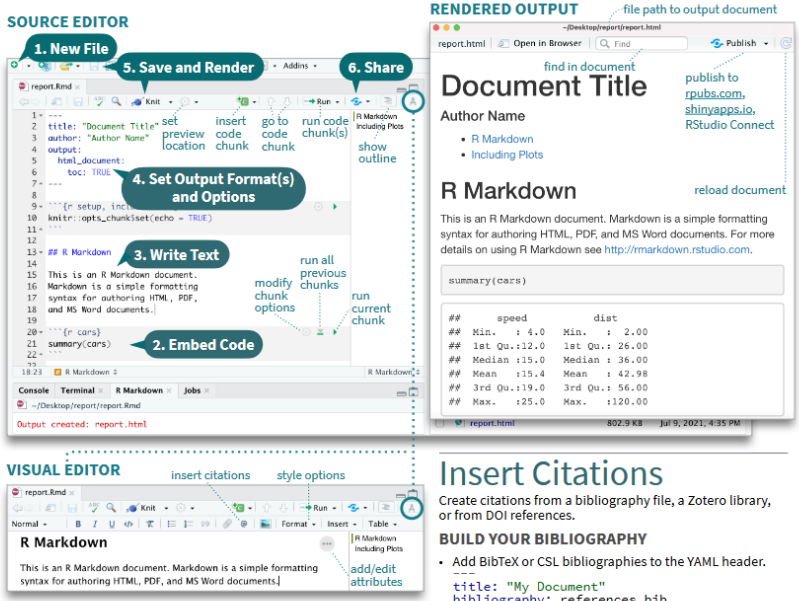
.Rmd files • Develop your code and ideas side-by-side in a single document. Run code as individual chunks or as an entire document.

Dynamic Documents • Knit together plots, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS Powerpoint.

Reproducible Research • Upload, link to, or attach your report to share. Anyone can read or run your code to reproduce your work.

Workflow

1. Open a **new .Rmd file** in the RStudio IDE by going to *File > New File > R Markdown*.
2. **Embed code** in chunks. Run code by line, by chunk, or all at once.
3. **Write text** and add tables, figures, images, and citations. Format with Markdown syntax or the RStudio Visual Markdown Editor.
4. **Set output format(s) and options** in the YAML header. Customize themes or add parameters to execute or add interactivity with Shiny.
5. **Save and render** the whole document. Knit periodically to preview your work as you write.
6. **Share your work!**



Embed Code with knitr

CODE CHUNKS

Surround code chunks with ```` ```{r} ```` and ```` ``` ```` or use the Insert Code Chunk button. Add a chunk label and/or chunk options inside the curly braces after **r**.

```
```{r chunk-label, include=FALSE}
summary(mtcars)
```
```

SET GLOBAL OPTIONS

Set options for the entire document in the first chunk.

```
```{r include=FALSE}
knitr::opts_chunk$set(message = FALSE)
```
```

INLINE CODE

Insert **`r <code>`** into text sections. Code is evaluated at render and results appear as text.

"Built with `r getRversion()`" --> "Built with 4.1.0"

OPTION	DEFAULT	EFFECTS
echo	TRUE	display code in output document
error	FALSE	TRUE (display error messages in doc) FALSE (stop render when error occurs)
eval	TRUE	run code in chunk
include	TRUE	include chunk in doc after running
message	TRUE	display code messages in document
warning	TRUE	display code warnings in document
results	"markup"	"asis" (passthrough results) "hide" (don't display results) "hold" (put all results below all code)
fig.align	"default"	"left", "right", or "center"
fig.alt	NULL	alt text for a figure
fig.cap	NULL	figure caption as a character string
fig.path	"figure/"	prefix for generating figure file paths
fig.width & fig.height	7	plot dimensions in inches
out.width		rescales output width, e.g. "75%", "300px"
collapse	FALSE	collapse all sources & output into a single block
comment	"##"	prefix for each line of results
child	NULL	files(s) to knit and then include
purl	TRUE	include or exclude a code chunk when extracting source code with knitr::purl()

See more options and defaults by running **str(knitr::opts_chunk$get())**

Insert Citations

Create citations from a bibliography file, a Zotero library, or from DOI references.

BUILD YOUR BIBLIOGRAPHY

- Add BibTeX or CSL bibliographies to the YAML header.

```
---
title: "My Document"
bibliography: references.bib
link-citations: TRUE
---
```

- If Zotero is installed locally, your main library will automatically be available.
- Add citations by DOI by searching "from DOI" in the **Insert Citation** dialog.

INSERT CITATIONS

- Access the **Insert Citations** dialog in the Visual Editor by clicking the **@** symbol in the toolbar or by clicking **Insert > Citation**.
- Add citations with markdown syntax by typing **[@cite]** or **@cite**.

Insert Tables

Output data frames as tables using **kable**(data, caption).

```
```{r}
data <- faithful[1:4, ]
knitr::kable(data,
             caption = "Table with kable")
```
```

Other table packages include **flextable**, **gt**, and **kableExtra**.

Write with Markdown

The syntax on the left renders as the output on the right.

Plain text. End a line with two spaces to start a new paragraph. Also end with a backslash\ to make a new line.	Plain text. End a line with two spaces to start a new paragraph. Also end with a backslash to make a new line.
italics and **bold** superscript^2^/subscript~2~ ~~strikethrough~~ escaped: * _ \\ endash: --, emdash: ---	*italics* and **bold** superscript^2/subscript$_2$ ~~strikethrough~~ escaped: * _ \ endash: –, emdash: —
# Header 1 ## Header 2 ... ###### Header 6	**Header 1** **Header 2** ... **Header 6**
- unordered list - item 2 - item 2a (indent 1 tab) - item 2b	• unordered list • item 2 • item 2a (indent 1 tab) • item 2b
1. ordered list 2. item 2 - item 2a (indent 1 tab) - item 2b	1. ordered list 2. item 2 • item 2a (indent 1 tab) • item 2b
<link url> [This is a link.](link url) [This is another link][id]. At the end of the document: [id]: link url	http://www.rstudio.com/ This is a link. This is another link.
![Caption](image.png) or ![Caption][id2] At the end of the document: [id2]: image.png	 Caption.
`verbatim code` ``` multiple lines of verbatim code ```	`verbatim code` multiple lines of verbatim code
> block quotes	block quotes
equation: $e^{i\pi} + 1 = 0$ equation block: $$E = mc^{2}$$	equation: $e^{i\pi} + 1 = 0$ equation block: $$E = mc^2$$
horizontal rule: ---	horizontal rule:

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

HTML Tabsets

```
# Results {.tabset}
## Plots text
text

## Tables
more text
```

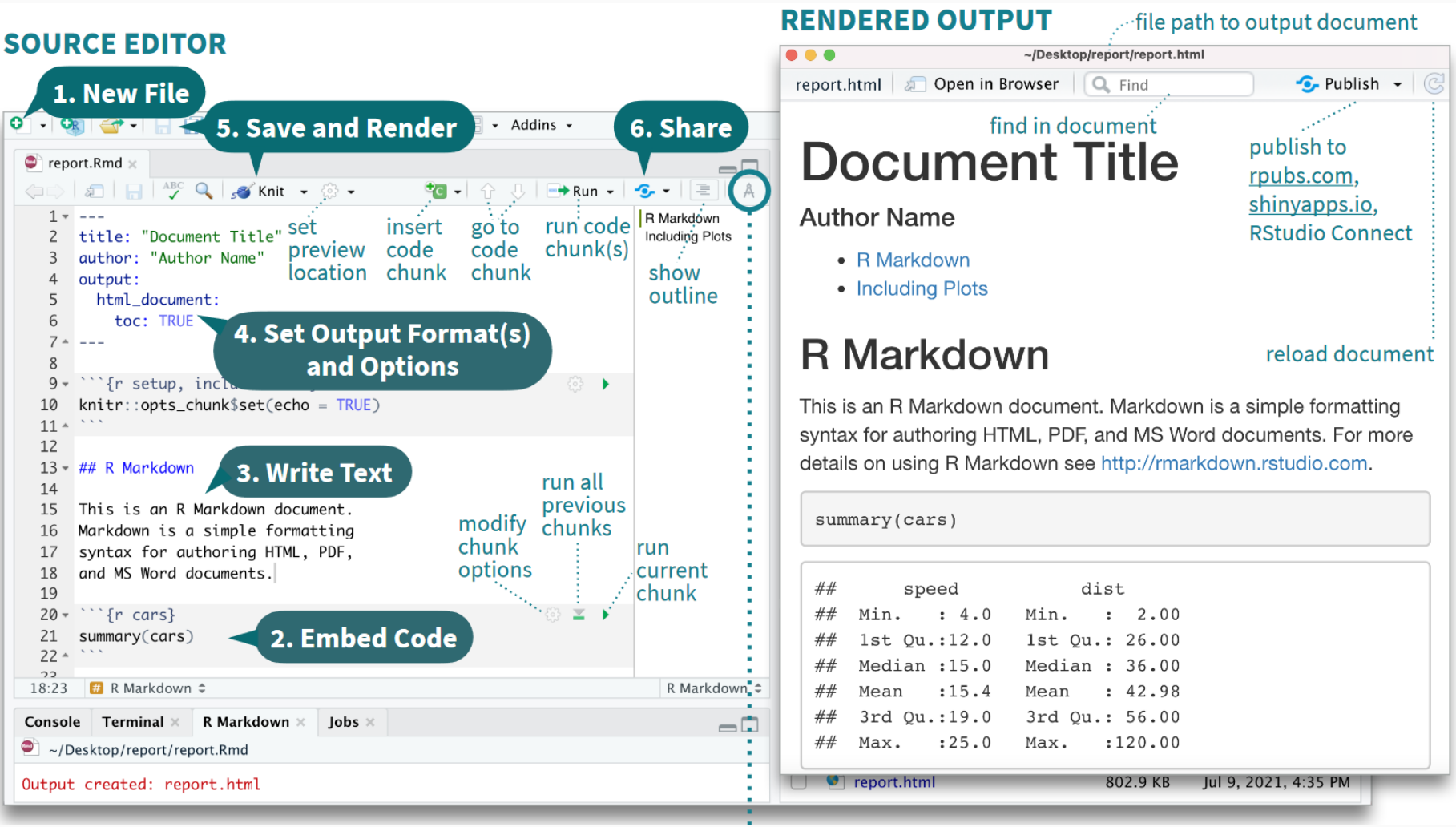
Results

Plots | Tables

text

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **rmarkdown.rstudio.com** • rmarkdown 2.9.4 • Updated: 2021-08

Cheatsheet


SOURCE EDITOR
1. New File
5. Save and Render
6. Share
Addins
report.Rmd
Knit
Run
R Markdown
Including Plots
set preview location
insert code chunk
go to code chunk
run code chunk(s)
show outline
4. Set Output Format(s) and Options
3. Write Text
run all previous chunks
modify chunk options
run current chunk
2. Embed Code
18:23
R Markdown
Console
Terminal
R Markdown
Jobs
~/Desktop/report/report.Rmd
Output created: report.html
RENDERED OUTPUT
file path to output document
~/Desktop/report/report.html
report.html
Open in Browser
Find
Publish
find in document
publish to rpubs.com, shinyapps.io, RStudio Connect
Document Title
Author Name
R Markdown
Including Plots
R Markdown
reload document
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.
summary(cars)
speed dist
Min. : 4.0 Min. : 2.00
1st Qu.:12.0 1st Qu.: 26.00
Median :15.0 Median : 36.00
Mean :15.4 Mean : 42.98
3rd Qu.:19.0 3rd Qu.: 56.00
Max. :25.0 Max. :120.00
report.html
802.9 KB
Jul 9, 2021, 4:35 PM

Components

There are **three** main components of an R Markdown document:

- **Metadata**
- **Text**
- **Code**

An R Markdown Document

```
---
title: "R Markdown"
subtitle: "Improving reproducibility using literate programming"
author: "Russell Goebel"
date: "`r Sys.Date()`"
output:
  xaringan::moon_reader:
    lib_dir: libs
    nature:
      highlightStyle: github
      countIncrementalSlides: false
    css: [default, metropolis, metropolis-fonts,style.css]
---
# Why use R Markdown?

```{r, echo = FALSE}
knitr::include_graphics("./images/nature_ecology_paper.PNG")
```

Why use R Markdown?


```{r, echo = FALSE, out.width = "80%"}
knitr::include_graphics("./images/nature_ecology_graphic.PNG")
```

Why use R Markdown?

```{r, echo = FALSE}
knitr::include_graphics("./images/nature_ecology_Rmarkdown.PNG")
```
```

Metadata

Code

Text

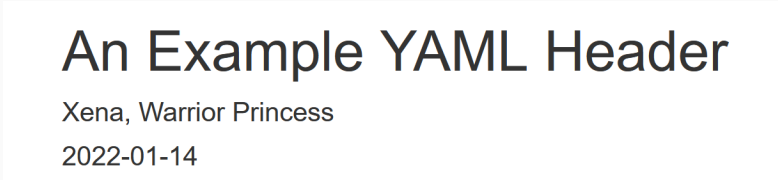
Metadata: YAML

YAML (YAML Ain't Markup Language)

Code:

```yaml
---
title: "An Example YAML Header"
author: "Xena, Warrior Princess"
date: "`r Sys.Date()`"
output: html_document
---
```

Output:



- Indentation matters
- You can run code here! (e.g, `Sys.Date()`)

Output formats

Documentation for many output formats can be found at https://bookdown.org/yihui/rmarkdown/documents.html.

- `html_document`
- `html_notebook`
- `pdf_document`
- `word_document`
- `html_vignette` (R package vignette)
- `odt_document` (OpenDocument Text document)
- `rtf_document` (Rich Text Format document)
- `md_document` (Markdown document)

Metadata: Example

Often, a **table of contents** goes a long way in organizing a document.

Some formats allow **themes** that can help your documents look nice.

Code:

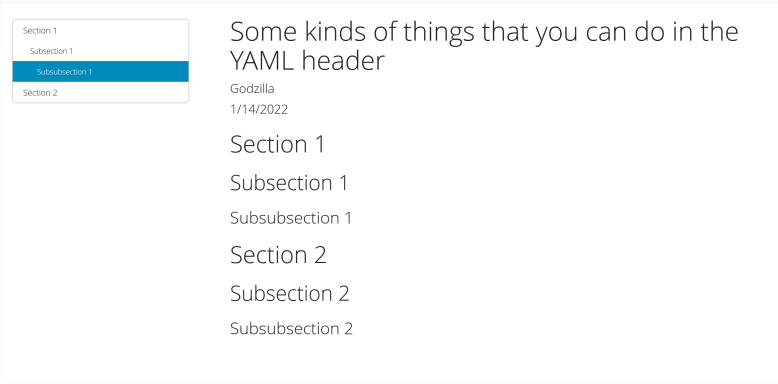
```yaml
---
title: "Some kinds of things that you can do in the YAML header"
author: "Godzilla"
date: "1/14/2022"
output:
  html_document:
    toc: true
    toc_float: true
    theme: yeti
---

# Section 1
## Subsection 1
### Subsubsection 1

# Section 2
## Subsection 2
### Subsubsection 2


```

Output:



Cheatsheet: YAML

Set Output Formats and their Options in YAML

Use the document's YAML header to set an **output format** and customize it with **output options**.

```
---
title: "My Document"
author: "Author Name"
output:
  html_document:
    toc: TRUE
---
```

Indent format 2 characters, indent options 4 characters

OUTPUT FORMAT	CREATES
html_document	.html
pdf_document*	.pdf
word_document	Microsoft Word (.docx)
powerpoint_presentation	Microsoft Powerpoint (.pptx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	Markdown for Github
ioslides_presentation	ioslides HTML slides
slidy_presentation	Slidy HTML slides
beamer_presentation*	Beamer slides

* Requires LaTeX, use **tinytex::install_tinytex()**

Also see **flexdashboard**, **bookdown**, **distill**, and **blogdown**.

IMPORTANT OPTIONS	DESCRIPTION	HTML	PDF	MS Word	MS PPT
anchor_sections	Show section anchors on mouse hover (TRUE or FALSE)	X			
citation_package	The LaTeX package to process citations ("default", "natbib", "biblatex")		X		
code_download	Give readers an option to download the .Rmd source code (TRUE or FALSE)	X			
code_folding	Let readers to toggle the display of R code ("none", "hide", or "show")	X			
css	CSS or SCSS file to use to style document (e.g. "style.css")	X			
dev	Graphics device to use for figure output (e.g. "png", "pdf")	X	X		
df_print	Method for printing data frames ("default", "kable", "tibble", "paged")	X	X	X	X
fig_caption	Should figures be rendered with captions (TRUE or FALSE)	X	X	X	X
highlight	Syntax highlighting ("tango", "pygments", "kate", "zenburn", "textmate")	X	X	X	
includes	File of content to place in doc ("in_header", "before_body", "after_body")	X	X		
keep_md	Keep the Markdown .md file generated by knitting (TRUE or FALSE)	X	X	X	X
keep_tex	Keep the intermediate TEX file used to convert to PDF (TRUE or FALSE)		X		
latex_engine	LaTeX engine for producing PDF output ("pdflatex", "xelatex", or "lualatex")		X		
reference_docx/_doc	docx/pptx file containing styles to copy in the output (e.g. "file.docx", "file.pptx")			X	X
theme	Theme options (see Bootswatch and Custom Themes below)	X			
toc	Add a table of contents at start of document (TRUE or FALSE)	X	X	X	X
toc_depth	The lowest level of headings to add to table of contents (e.g. 2, 3)	X	X	X	X
toc_float	Float the table of contents to the left of the main document content (TRUE or FALSE)	X			

Use **?<output format>** to see all of a format's options, e.g. **?html_document**

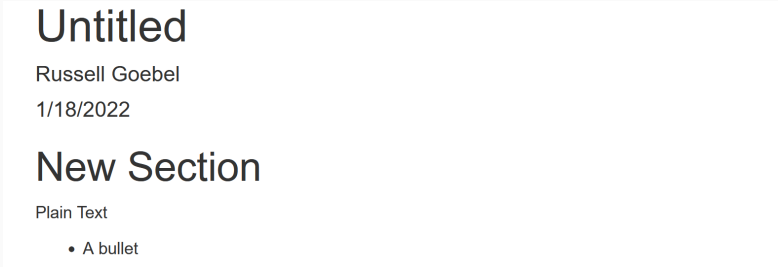
Text

```
---
title: "Untitled"
author: "Russell Goebel"
date: "1/18/2022"
output: html_document
---

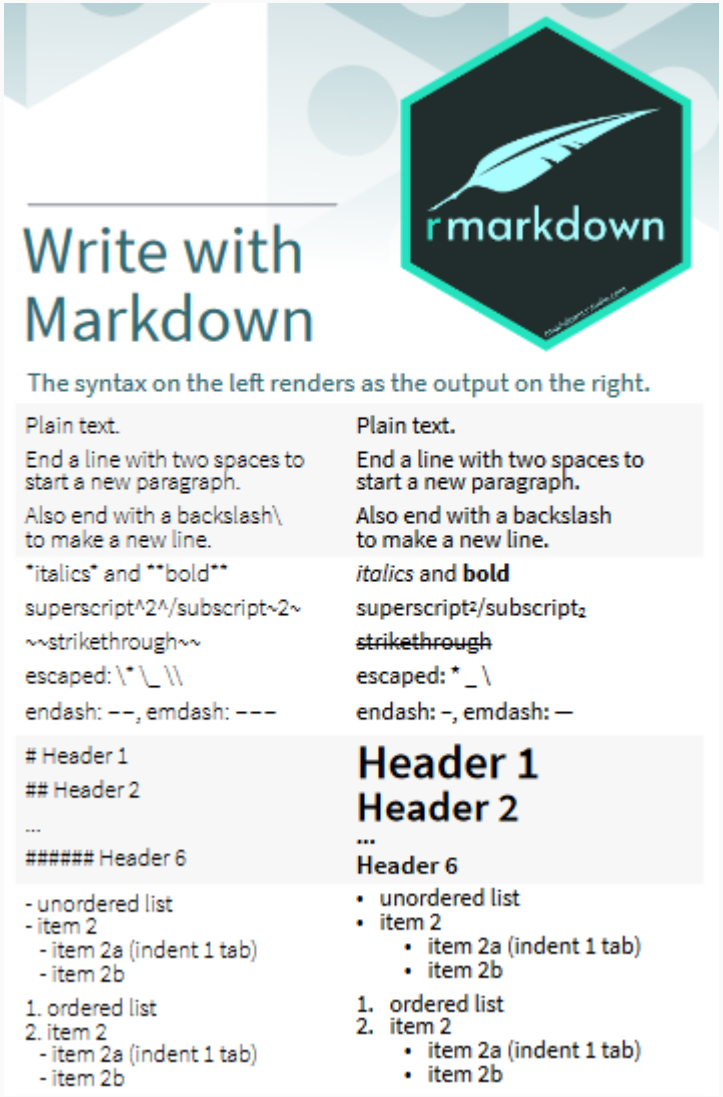
# New Section

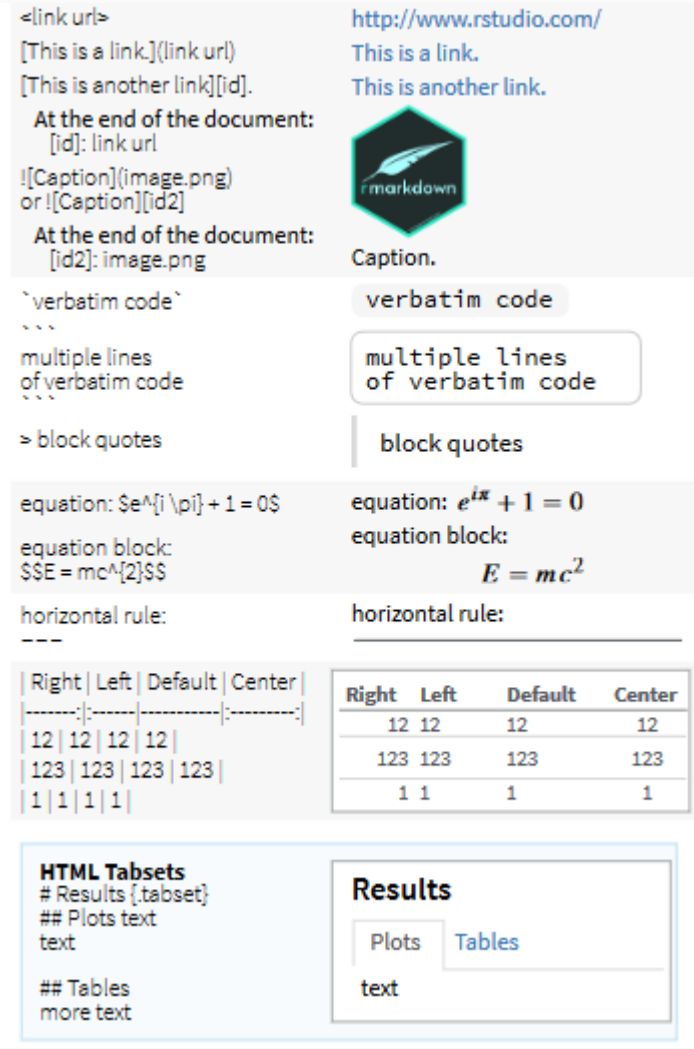
Plain Text

* A bullet
```



Text: Markdown





html tabsets

```
# Section Title { .tabset}

## A tab

## Another tab
```

```
---
title: "Untitled"
author: "Russell Goebel"
date: "1/18/2022"
output: html_document
---

# Tabsets {.tabset}

## Summary

```{r cars}
summary(cars)
```

Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
```

html tabsets

Untitled

Russell Goebel

1/18/2022

Tabsets

Summary | Including Plots

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

html tabsets

Untitled

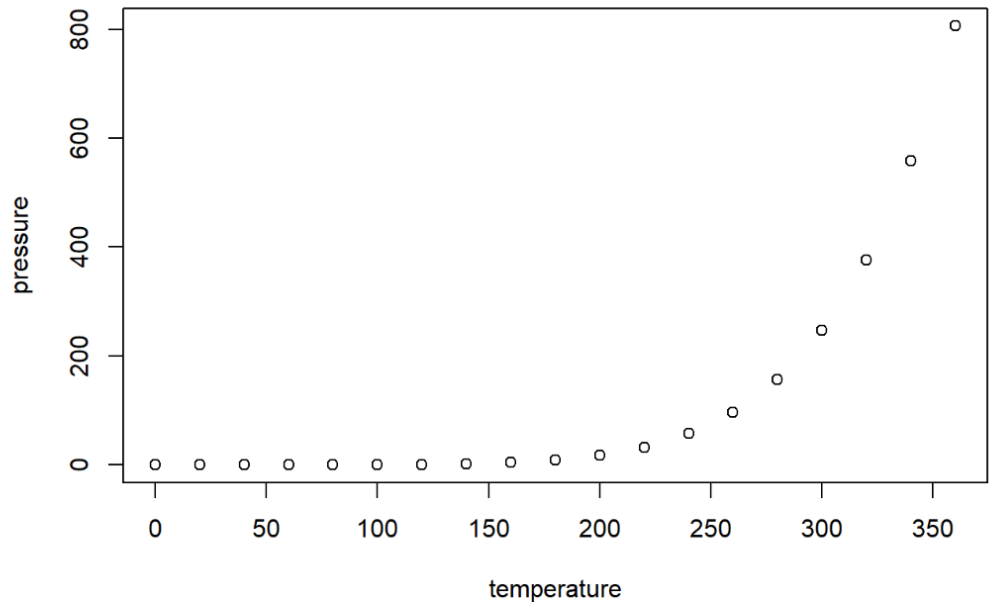
Russell Goebel

1/18/2022

Tabsets

Summary | Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Text in html or LaTeX formats

For LaTeX output formats:

- LaTeX code can be used to customise appearance
- LaTeX **tables** can be used:

```
\begin{table}[]
\begin{tabular}{|l|l|l|l|l|}
\hline
 &  &  &  &  \\ \hline
 &  &  &  &  \\ \hline
 &  &  &  &  \\ \hline
 &  &  &  &  \\ \hline
\end{tabular}
\end{table}
```

better: `knitr::kable(data,format = "latex")`

For html output formats:

- **CSS** can be used to customise appearance

```
.small { font-size: 70% }
.large { font-size: 130% }
.huge  { font-size: 150%}
```

- HTML **tabsets** can be used
- Enables some interaction (such as **plotly**)

To check you can use `knitr::is_html_output()` or `knitr::is_latex_output()`.

Warning!

Always consider **transferability**.

A single source document can create documents with multiple formats.

Tailoring to a single output format (like html or pdf) may be at the expense of this transferability.

(https://bookdown.org/yihui/rmarkdown-cookbook/latex-output.html)

Code: chunks

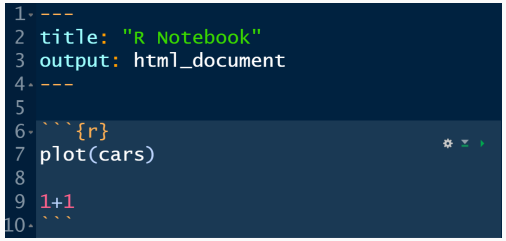
CODE CHUNKS

Surround code chunks with ```` ```{r} ```` and ```` ``` ```` or use the Insert Code Chunk button.  Add a chunk label and/or chunk options inside the curly braces after **r**.

````
```{r chunk-label, include=FALSE}
summary(mtcars)
```
````

Shortcut: Ctrl + Alt + I (OS X: Cmd + Option + I)

Chunks



```
---
title: "R Notebook"
output: html_document
---

```{r}
plot(cars)

1+1
```
```

Chunk options

OPTION	DEFAULT	EFFECTS
echo	TRUE	display code in output document
error	FALSE	TRUE (display error messages in doc) FALSE (stop render when error occurs)
eval	TRUE	run code in chunk
include	TRUE	include chunk in doc after running
message	TRUE	display code messages in document
warning	TRUE	display code warnings in document
results	"markup"	"asis" (passthrough results) "hide" (don't display results) "hold" (put all results below all code)
fig.align	"default"	"left", "right", or "center"
fig.alt	NULL	alt text for a figure
fig.cap	NULL	figure caption as a character string
fig.path	"figure/"	prefix for generating figure file paths
fig.width & fig.height	7	plot dimensions in inches
out.width		rescales output width, e.g. "75%", "300px"
collapse	FALSE	collapse all sources & output into a single block
comment	"##"	prefix for each line of results
child	NULL	files(s) to knit and then include
purl	TRUE	include or exclude a code chunk when extracting source code with knitr::purl()

See more options and defaults by running **str(knitr::opts_chunk$get())**

Setting Global Options

It can be helpful to set global options:

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

Chunk options RStudio shortcut

Option: child

Organize lengthy documents using `child`:

```
29 ```{r, child = c("one.Rmd","two.Rmd")}
30
31
32 ```
```

More information here: https://bookdown.org/yihui/rmarkdown-cookbook/child-document.html

Options can contain expressions

````
```{r}
include_chunks = TRUE
```

```{r, include = include_chunks}
1+1
```
````

knitr::kable()

Place this function in an **R chunk** to render a table programatically:

```
my_data_frame = data.frame(A = 1:5, B = 6:10)
knitr::kable(my_data_frame) # makes a table
```

A	B
1	6
2	7
3	8
4	9
5	10

knitr::kable tip

If you are generating a table using data in R, `knitr::kable` can be used in the console to print LaTeX code for that table!

```
> my_data <- data.frame(A = 1:3, B = 4:6)
> knitr::kable(my_data,format = "latex")

\begin{tabular}{r|r}
\hline
A & B\\
\hline
1 & 4\\
\hline
2 & 5\\
\hline
3 & 6\\
\hline
\end{tabular}
> 
```

Inline Code

INLINE CODE

Insert **`r <code>`** into text sections. Code is evaluated at render and results appear as text.

"Built with `r getRversion()`" --> "Built with 4.1.0"

Other Languages (Python, C++, etc)

Other programming languages than R can be run:

````
```{python, message = FALSE}
def hello():
  print("Hello!")

hello()
```
````

```
Hello!
```

Hide

```
def hello():
  print("Hello!")

hello()
```

```
Hello!
```

A list of supported languages can be obtained using `knitr::knit_engines()`:

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

```
##  [1] "awk"         "bash"        "coffee"
##  [4] "gawk"        "groovy"      "haskell"
##  [7] "lein"        "mysql"       "node"
## [10] "octave"      "perl"        "psql"
## [13] "Rscript"     "ruby"        "sas"
## [16] "scala"       "sed"         "sh"
## [19] "stata"       "zsh"         "highlight"
## [22] "Rcpp"        "tikz"        "dot"
## [25] "c"           "cc"          "fortran"
## [28] "fortran95"   "asy"         "cat"
## [31] "asis"        "stan"        "block"
## [34] "block2"      "js"          "css"
## [37] "sql"         "go"          "python"
## [40] "julia"       "sass"        "scss"
## [43] "R"           "bslib"       "theorem"
## [46] "lemma"       "corollary"   "proposition"
## [49] "conjecture"  "definition"  "example"
## [52] "exercise"    "hypothesis"  "proof"
## [55] "remark"      "solution"
```

Summary so far:

- An R Markdown document consists of **metadata**, **text** and **code**.
- The **YAML** header controls things like title/author, output format, or output options
  - include a floating table of contents in an html document with `toc: true` and `toc_float: true`, or change the theme
- Text uses **markdown** syntax, but depending on the output format, you might be able to use LaTeX, CSS or html as well.
  - Keep transferability in mind!
- **Chunks** can be included to show and run code
  - Use chunk options to customize how code interacts with your document
  - Chunk options can include R expressions
  - The code can be from one of many languages (such as Python)

Using R Markdown

Bibliographies and Citations

Cross-referencing within a document

The **rticles** package

R Package Vignettes

Presentations

Bibliographies and Citations

Bibliographies and Citations

- We will need to create a new **.bib** file
- We will need to adjust the **YAML** header
- We can then reference using syntax similar to in LaTeX.

Bibliographies and Citations

YAML Header:

```
---
output: html_document
bibliography: references.bib
---
```

...or, to use **natbib/biblatex:**

```
output:
  pdf_document:
    citation_package: natbib
  bookdown::pdf_book:
    citation_package: biblatex
```

.bib entries:

```
@Book{Guide,
    title = {R Markdown: The Definitive Guide
    author = {Yihui Xie and J.J. Allaire and
    publisher = {Chapman and Hall/CRC},
    address = {Boca Raton, Florida},
    year = {2018},
    note = {ISBN 9781138359338},
    url = {https://bookdown.org/yihui/rmarkdo
  }
```

See: https://bookdown.org/yihui/rmarkdown-cookbook/bibliography.html

Bibliographies and Citations

To cite an entry, use `@key` or `[@key]`.

- `@key` renders without parentheses, e.g. `@Guide` renders as Xie, Allaire, and Grolemund (2018).
- `[@key]` renders with parentheses, e.g. `@Guide` renders as (Xie, Allaire, and Grolemund 2018).

Bibliographies and Citations

To change **style**, an additional line should be included:

```
---
output: html_document
bibliography: references.bib
csl: biomed-central.csl
---
```

Here `csl` stands for "Citation Style Language".

Cross-referencing

Cross-referencing within a document

We can reference **chunks** (tables and figures), **sections**, and **equations** in `bookdown` output formats

- **bookdown** extends Pandoc
- Examples of bookdown formats are `bookdown::pdf_document2` or `bookdown::html_document2`.

For Bookdown Formats

Examples of bookdown formats are `bookdown::pdf_document2` or `bookdown::html_document2`.

You can most easily reference **chunks** and **sections**.

- Recall that we can create **figures** or **tables** in chunks!

Syntax:

- Figure `\@ref(fig:chunk-name)`
- Table `\@ref(tab:chunk-name)`
- Section `\@ref(slug)`

````
```{ r chunk-name}
knitr::kable(data.frame(a = 1:3, b = 4:6))
```
````

```
 8 # Section {#my-section}
 9
10 Section \@ref(my-section)
```

Equations in Bookdown formats

Assign labels using `(\#eq:label)` and reference using `\@ref(label)`. For instance,

```
\begin{equation}
  f\left(k\right) = \binom{n}{k} p^k\left(1-p\right)^{n-k}
  (\#eq:binom)
\end{equation}
```

$$f\left(k\right) = \binom{n}{k} p^k (1-p)^{n-k}$$

can be referenced using `\@ref(eq:binom)`.

This **only works in bookdown formats** to the best of my knowledge.

Not to be confused with using LaTeX syntax in LaTeX output formats.

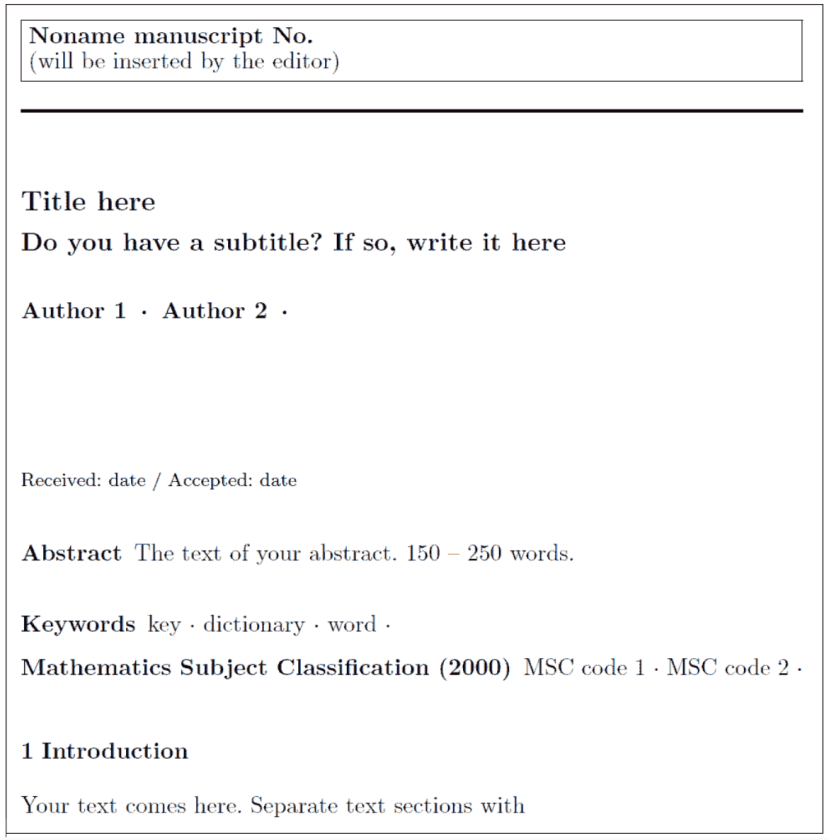
(See: https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html)

The **rticles** package for academic journals

Academic Journals

The **rticles** package is designed to simplify the creation of documents that conform to submission standards.

This package provides **templates** like this one:



Noname manuscript No.
(will be inserted by the editor)

Title here

Do you have a subtitle? If so, write it here

Author 1 · Author 2 ·

Received: date / Accepted: date

Abstract The text of your abstract. 150 – 250 words.

Keywords key · dictionary · word ·

Mathematics Subject Classification (2000) MSC code 1 · MSC code 2 ·

1 Introduction

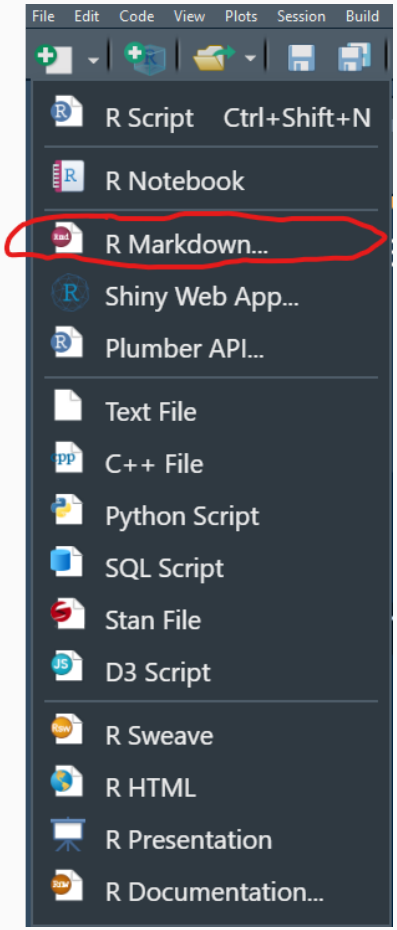
Your text comes here. Separate text sections with

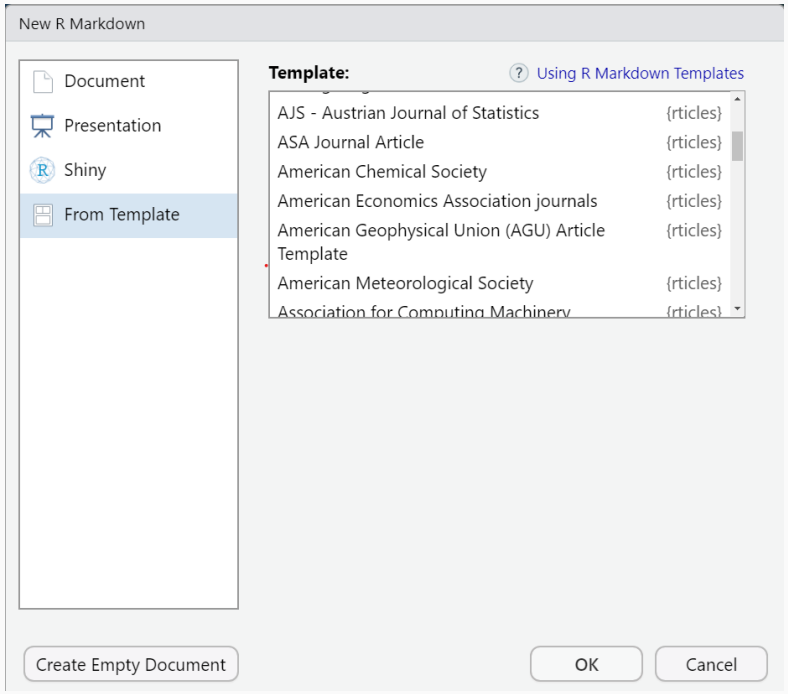
List of Supported Journals

```
rticles::journals()
```

```
##  [1] "acm"           "acs"           "aea"           "agu"           "ajs"
## [18] "ims"           "jasa"          "jedm"          "joss"          "jss"
## [35] "sage"          "sim"           "springer"      "tf"            "trb"
```

Using Templates





Example: JSS

```yaml
---
documentclass: jss
author:
  - name: FirstName LastName
    affiliation: University/Company
    # use this syntax to add text on several lines
    address: |
      | First line
      | Second line
    email: \email{name@company.com}
    url: http://rstudio.com
  - name: Second Author
    affiliation: 'Affiliation \AND'
    # To add another line, use \AND at the end of the previous one as above
  - name: Third Author
    address: |
      | Department of Statistics and Mathematics,
      | Faculty of Biosciences,
      | Universitat Autònoma de Barcelona
    affiliation: |
      | Universitat Autònoma
      | de Barcelona
    # use a different affiliation in adress field (differently formated here)
    affiliation2: Universitat Autònoma de Barcelona
title:
  formatted: "A Capitalized Title: Something about a Package \\pkg{foo}"
  # If you use tex in the formatted title, also supply version without
  plain:     "A Capitalized Title: Something about a Package foo"
  # For running headers, if needed
  short:     "\\pkg{foo}: A Capitalized Title"
abstract: >
  The abstract of the article.
keywords:
  # at least one keyword must be supplied
  formatted: [keywords, not capitalized, "\\proglang{Java}"]
  plain:     [keywords, not capitalized, Java]
preamble: >
  \usepackage{amsmath}
output: rticles::jss_article
---
```

R Package Vignettes

(very briefly)

R Package Vignettes

There is some synergy between R Markdown and R packages!

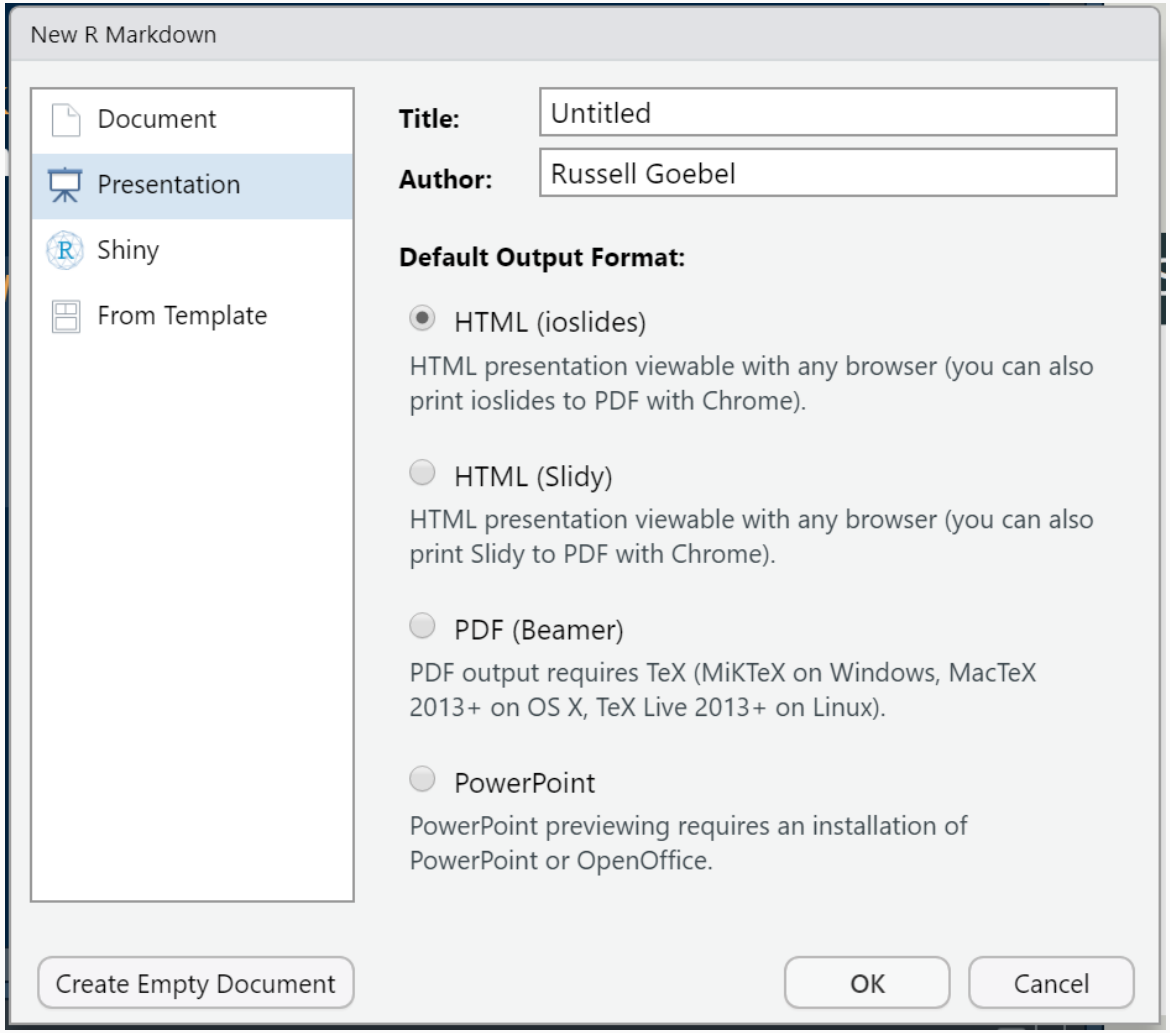
- For info about R Packages, see: https://r-pkgs.org/index.html
- In https://r-pkgs.org/vignettes.html there is a **simple workflow** to generate html vignettes for R packages. We won't go into it here.

Example Vignette

Use `browseVignettes(package name)` to see vignettes for a package. Here is an example from dplyr:

Presentations

Presentations



What's the difference?

- **Slidy:**
  - Many features for fancy slides and great online resources
  - A bit complicated / time consuming
- **Ioslides**
  - Easy to understand
  - For more features CSS code is needed
- **Beamer:**
  - Beamer
  - Beamer
- **Powerpoint**
  - Creates Powerpoint slides when you knit!

Slidy

```
---
title: "Slidy Presentation"
author: "Russell Goebel"
date: "1/17/2022"
output: slidy_presentation
---

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

R Markdown

Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

Slide with R Output

```{r cars, echo = TRUE}
summary(cars)
```

Slide with Plot

```{r pressure}
plot(pressure)
```
```

Slidy Presentation

Russell Goebel

1/17/2022

R Markdown

Slide with Bullets

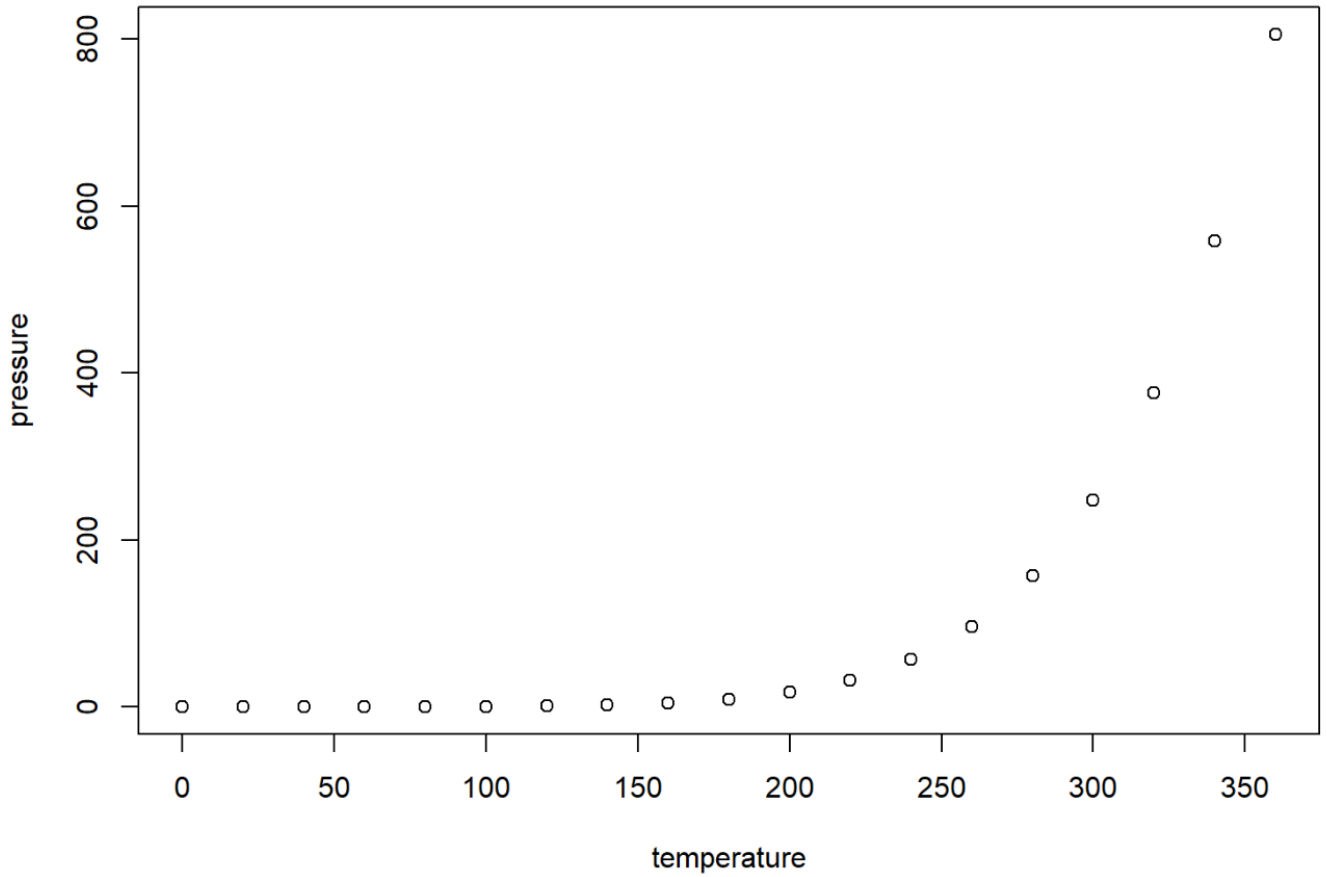
- Bullet 1
- Bullet 2
- Bullet 3

Slide with R Output

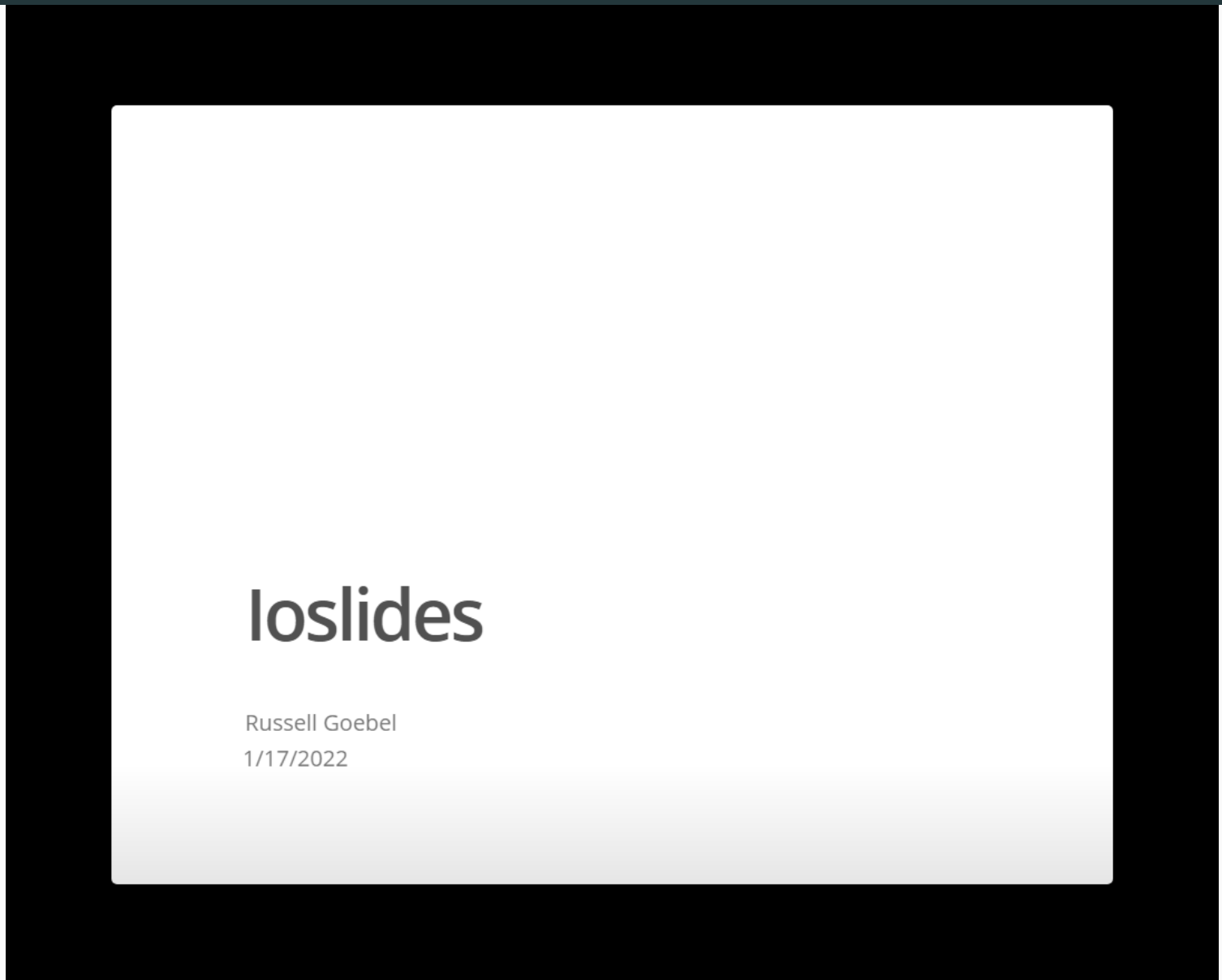
```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

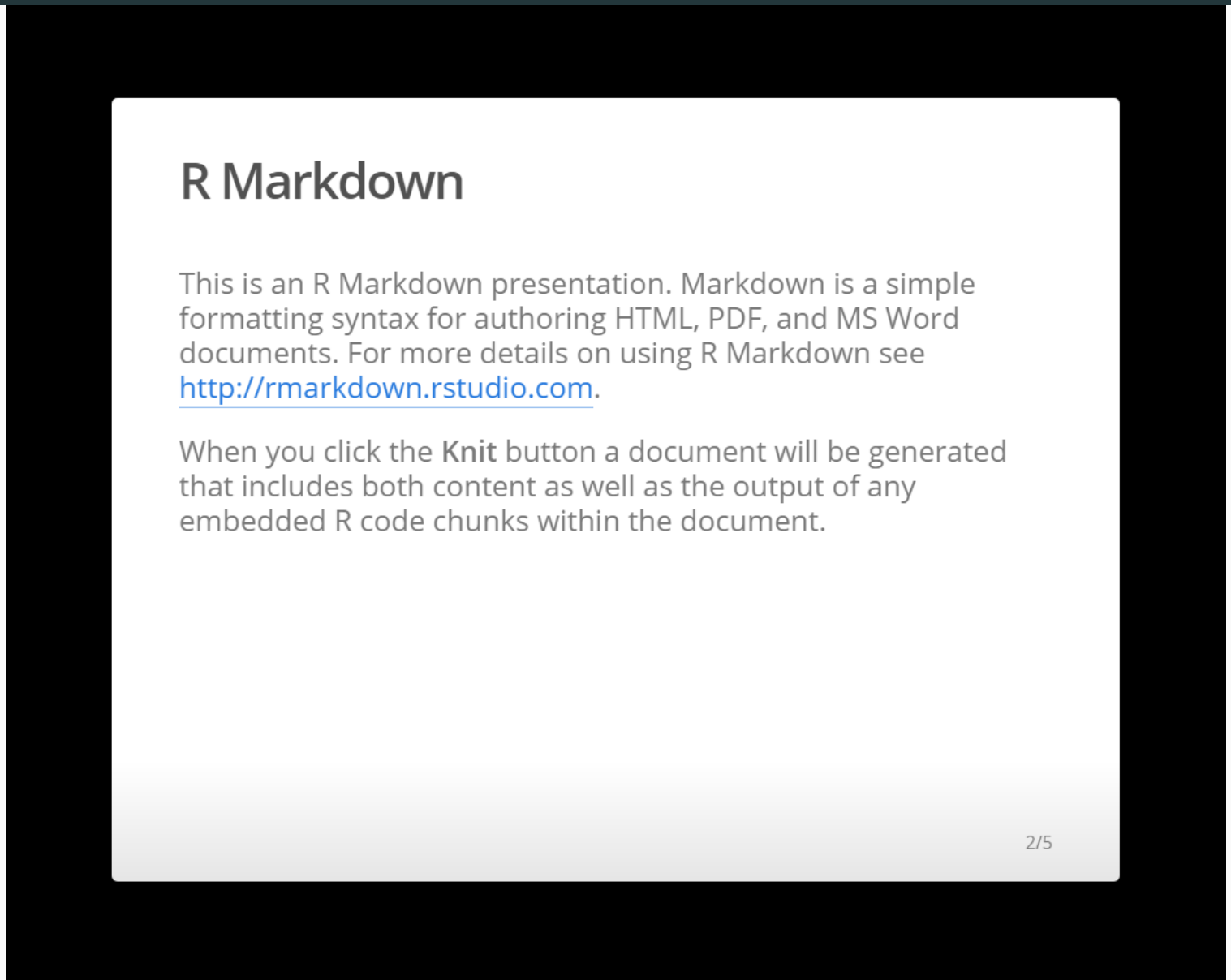
Slide with Plot

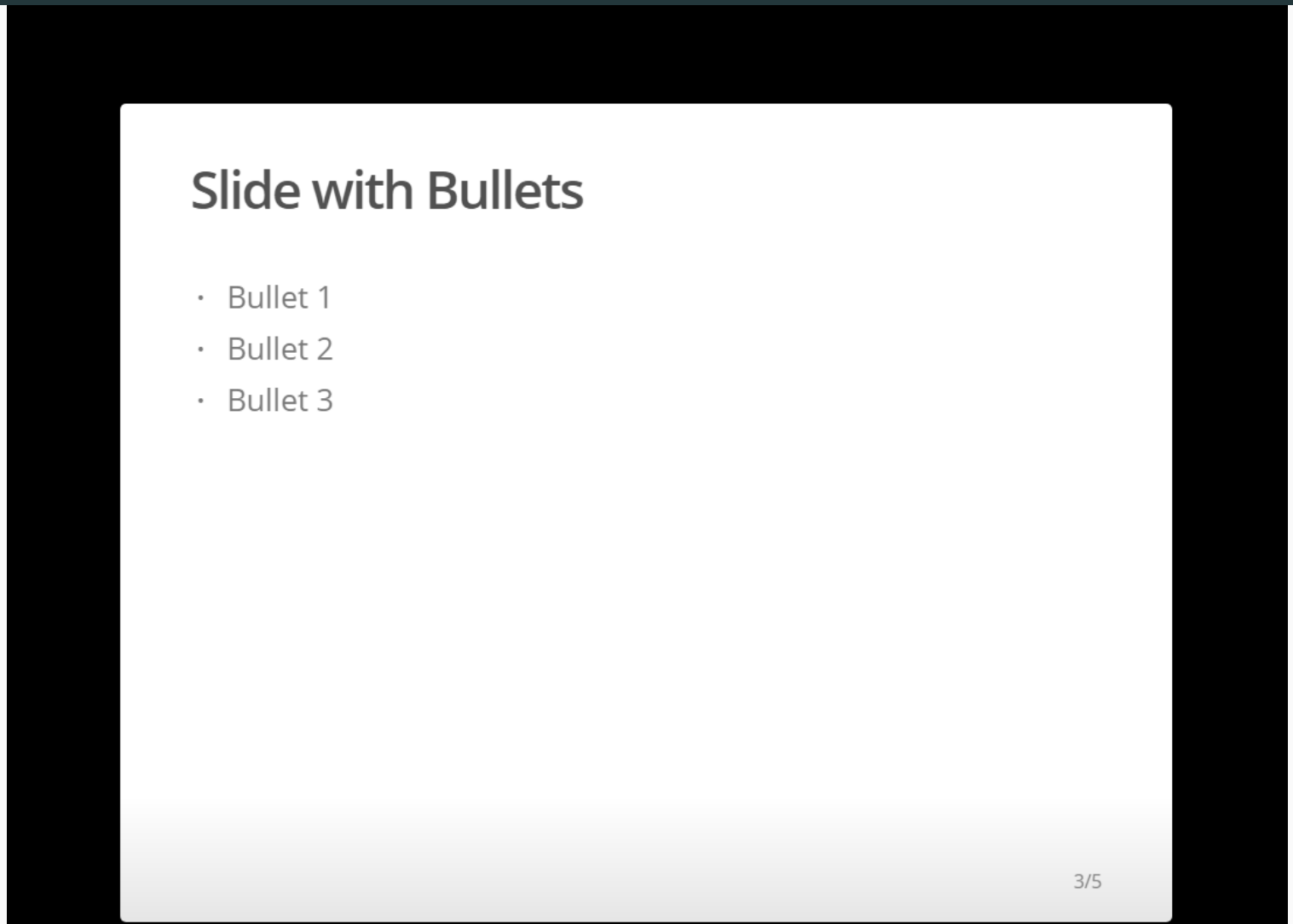


Ioslides



Ioslides


R Markdown
This is an R Markdown presentation. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.
When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.
2/5

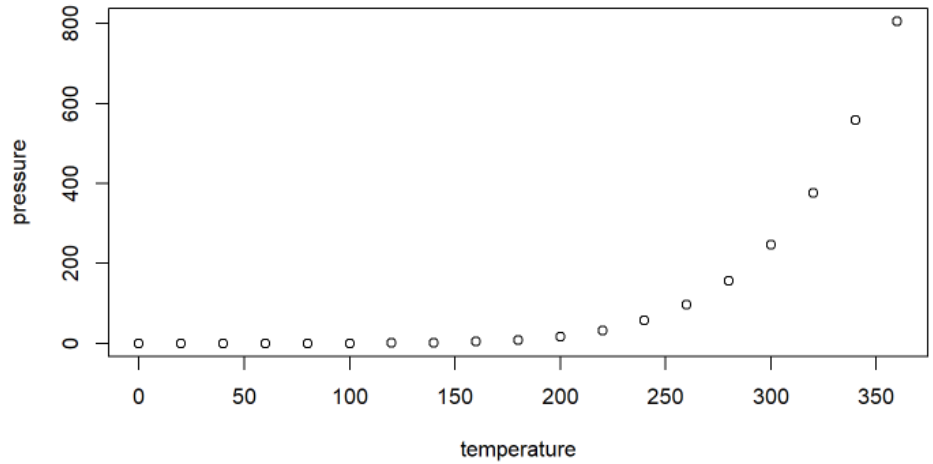

Slide with Bullets
Bullet 1
Bullet 2
Bullet 3
3/5

Slide with R Output

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Slide with Plot



5/5

Other presentation formats

Other formats include **reveal.js** and **xaringan**

This presentation is in Xaringan.

Resources

R Markdown Definitive Guide: https://bookdown.org/yihui/rmarkdown/

R Markdown Cookbook: https://bookdown.org/yihui/rmarkdown-cookbook/

knitr: https://yihui.org/knitr/

Markdown: https://daringfireball.net/projects/markdown/

Pandoc: https://pandoc.org/

R Packages: https://r-pkgs.org/

Donald Knuth's website: https://www-cs-faculty.stanford.edu/~knuth/

Our Path to Better Science Paper: https://www.nature.com/articles/s41559-017-0160

Ocean Health Index Website: https://ohi-science.org/

Questions?

Thank you!