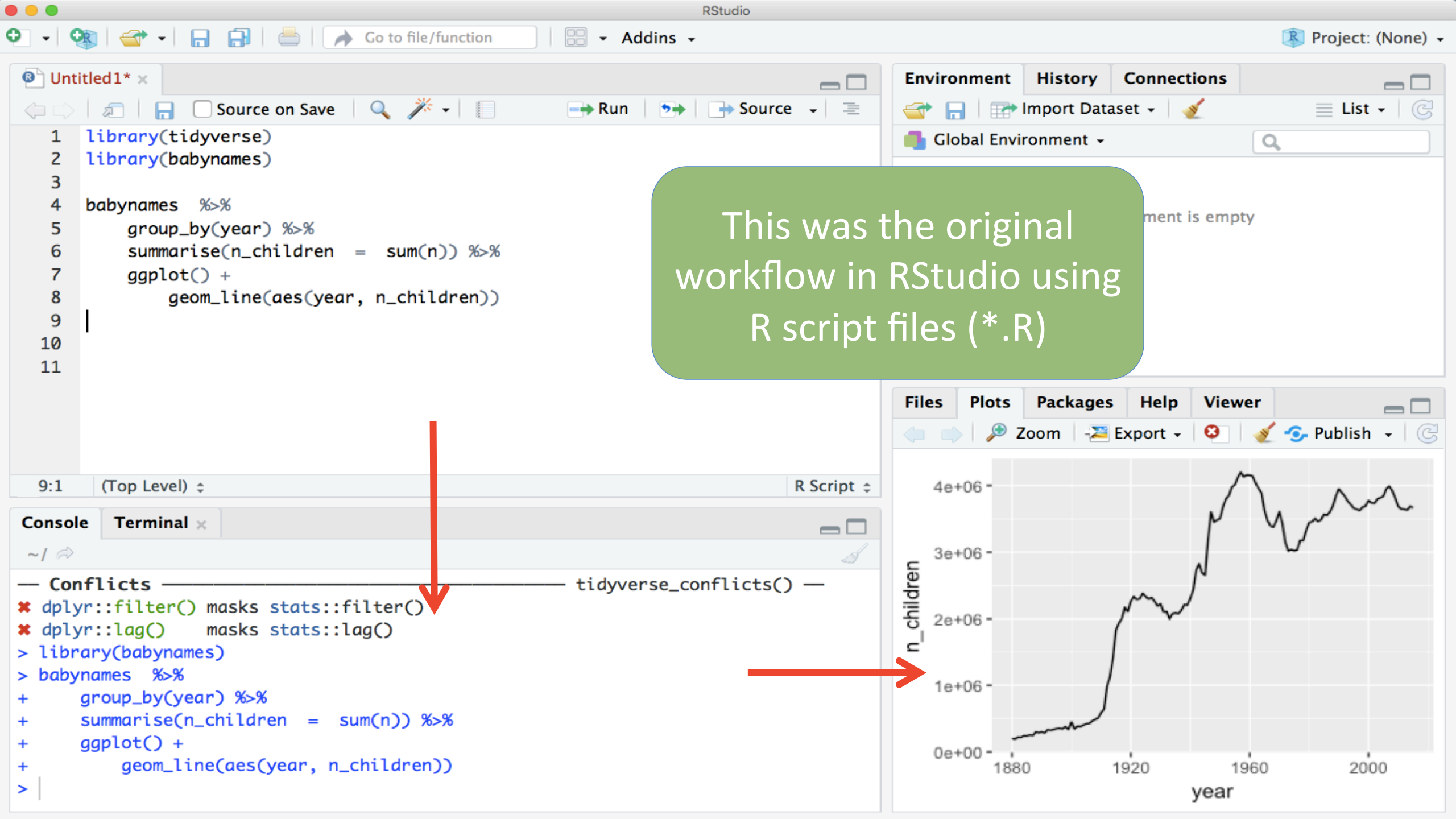
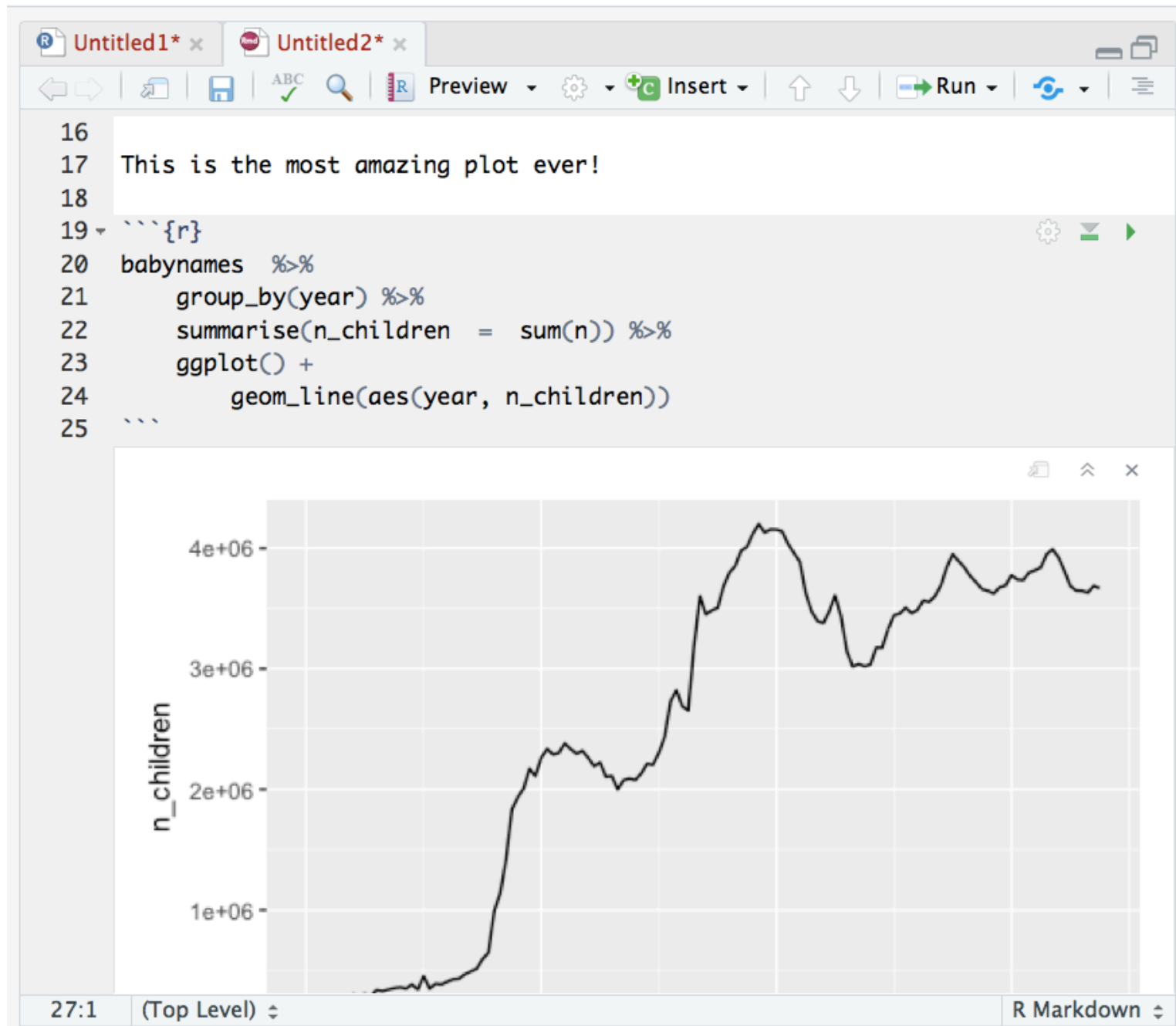


# R Markdown

The Basics

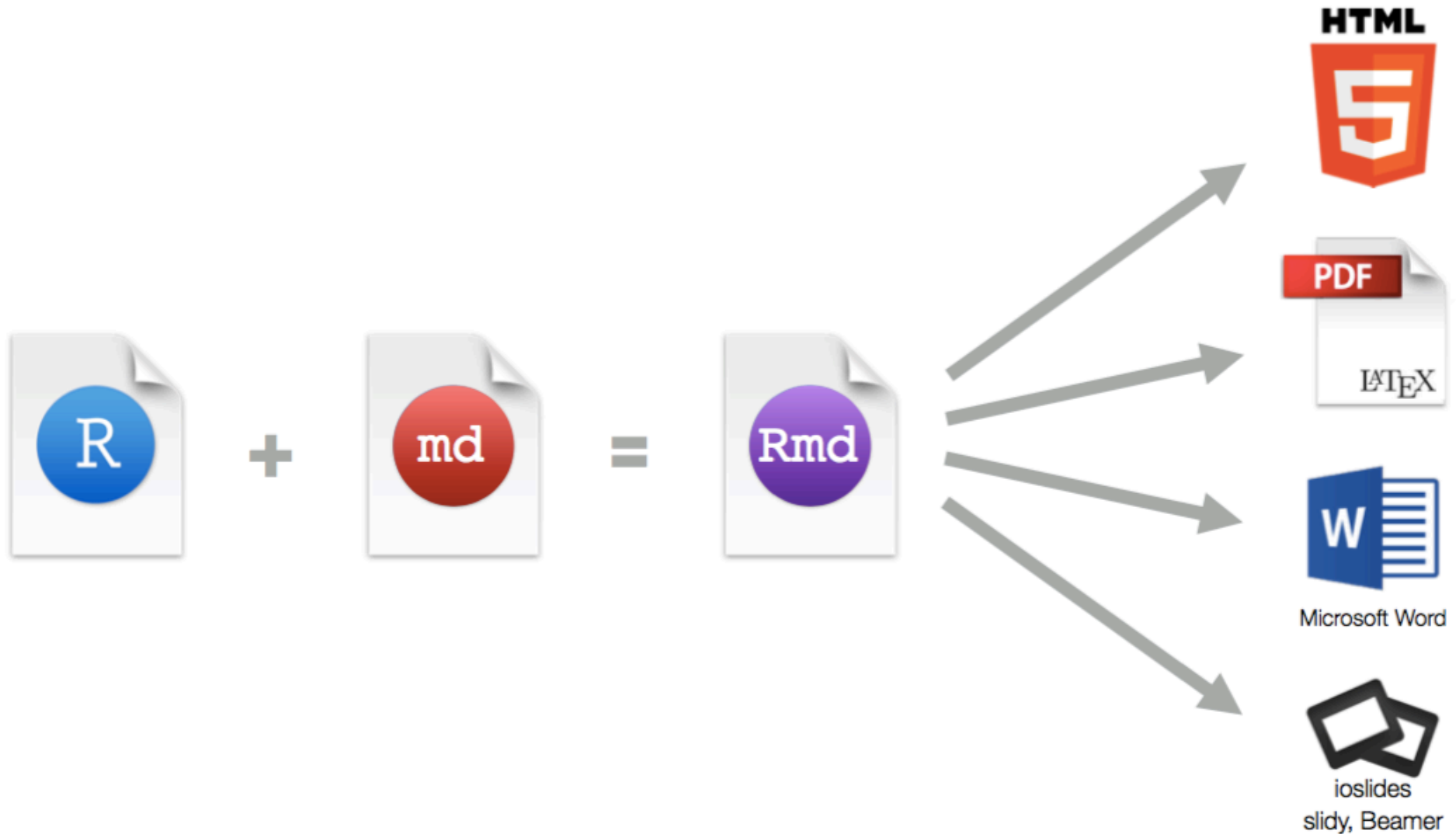




R Notebooks bring together prose, code, and results. They are great for

- Communicating results to those who do not need to see the code
- Collaborating with other data scientists interested in the results and the code
- Actually doing data science

# R Markdown



# Components of an R Notebook

```
00-Introduction.Rmd x
← → | ↗ | 📁 | ABC | 🔍 | 🌐 Knit | ⚙️ | +c Insert | ↑ ↓ | ➡ Run | 🔁 | ☰
1 |---
2 title: "R Notebook"
3 output: html_notebook
4 ---
5
6 This is an R Notebook. You can use it to take notes and run code. For
7 example, you can write your name on the line below. Try it:
8
9
10 ```{r}
11 # You can write code in chunks that look like this.
12 # This chunk uses some code from base R to plot a data set.
13 # To run the code click the Green play button to the right. Try it!
14 plot(cars)
15 ```
16
17 Good job! The results of a code chunk will appear beneath the chunk. You can
18 click the x above the results to make them go away, but let's not do that.
19 You can open a new R Notebook by going to File > New File > R Notebook
20 (or click on the sheet-with-green-plus-sign icon in the upper right corner).
21 # Adding chunks
22
23 To add a new chunk, press *Cmd+Option+I* (*Ctrl+Alt+I* on Windows), or click
the *Insert* button at the top of this document, then select *R*.
1:1 | # R Notebook | R Markdown
```

YAML header

Text using Markdown  
(and LaTeX)

Code using R (or other  
languages)

# Text Using Markdown

# Headers

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

##### Header 6



Header 1

Header 1

Header 1

Header 1

Header 1

Header 1

Use #'s to create headers for chapters, sections, subsections, etc.



Add two spaces at the end of a line to start a new line

# Text

Text○

italics

**bold**

`code`



Text

*italics*

**bold**

`code`

Text is rendered as plain text. Use underscores (   ) to make italics, two underscores (  ) to make bold, back ticks (  ) to make code.

# Lists

## Bullet points

- \* item 1
- \* item 2

## Numbered lists

1. item 1
2. item 2



## Bullet points

- item 1
- item 2

## Numbered lists

1. item 1
2. item 2

Use asterisks to make bullet points.  
Use numbers to make numbered lists.

# Hyperlinks

This is a  
[link](www.ship.edu).



This is a [link](http://www.ship.edu).

Use brackets to denote a link. Place the URL in parentheses.

# Equations

According to  
Einstein,  
`$E=mc^2$`.



According to  
Einstein,  $E = mc^2$ .

Write equations with  
LaTeX commands and  
surround them in `$`'s.

# Equation Blocks

According to  
Einstein,  
 $E=mc^2$ .



According to  
Einstein,  
$$E = mc^2.$$

Use two \$'s to make  
centered equation  
blocks.

# Images

`` The RStudio logo.



The RStudio logo.

Use a link preceded by an ! to insert an image. Link should be a URL or file path.

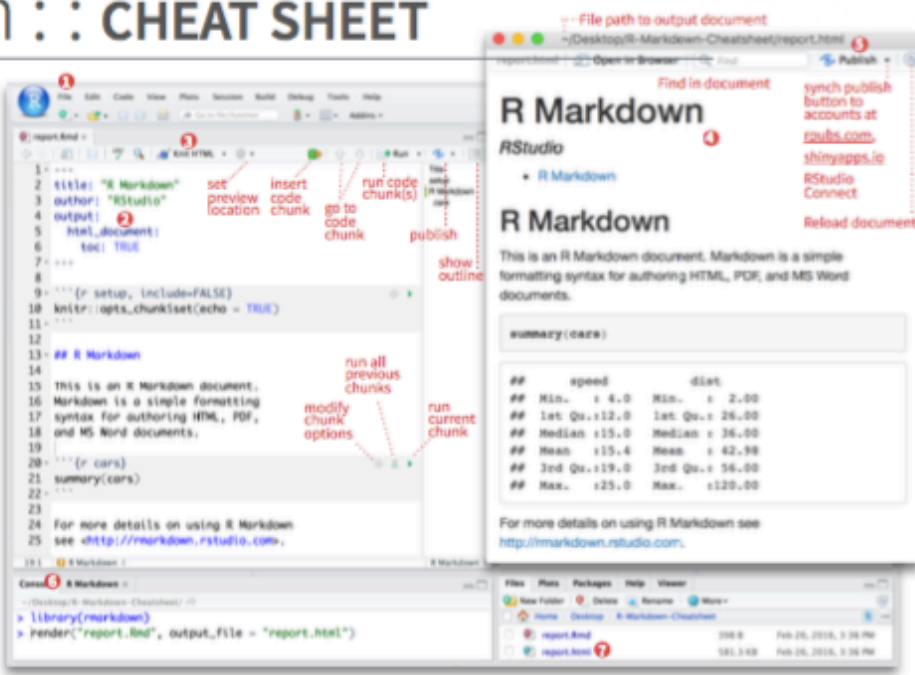
# R Markdown: : CHEAT SHEET

## What is R Markdown?

- .Rmd files** - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.
- Reproducible Research** - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.
- Dynamic Documents** - You can choose to export the finished report in a variety of formats, including HTML, PDF, MS Word, or RTF documents; HTML or PDF based slides, Notebooks, and more.

## Workflow

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



## render

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

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

## Embed code with knitr syntax

**INLINE CODE**  
Insert with `'r<code>'`. Results appear as text without code.  
Built with `'r getRversion()'` → Built with 3.2.3

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

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

### IMPORTANT CHUNK OPTIONS

- cache** - cache results for future knits (default = FALSE)
- cache.path** - directory to save cached results in (default = "cache/")
- child** - file(s) to knit and then include (default = NULL)
- collapse** - collapse all output into single block (default = FALSE)
- comment** - prefix for each line of results (default = "#")

- dependson** - chunk dependencies for caching (default = NULL)
- echo** - Display code in output document (default = TRUE)
- engine** - code language used in chunk (default = "R")
- error** - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)
- eval** - Run code in chunk (default = TRUE)

- fig.align** - 'left', 'right', or 'center' (default = 'center')
- fig.cap** - figure caption as character string (default = NULL)
- fig.height, fig.width** - Dimensions of plots in inches
- highlight** - highlight source code (default = TRUE)
- include** - include chunk in doc after running (default = TRUE)

- message** - display code messages in document (default = TRUE)
- results** - (default = "markup") 'asis' - passthrough results 'hide' - do not display results 'hold' - put all results below all code
- tidy** - tidy code for display (default = FALSE)
- warning** - display code warnings in document (default = TRUE)

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

## .rmd Structure

**YAML Header**  
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).

At start of file

Between lines of ---

**Text**  
Narration formatted with markdown, mixed with:

**Code Chunks**  
Chunks of embedded code. Each chunk:

Begins with ````{r}`

ends with `````

R Markdown will run the code and append the results to the doc.

It will use the location of the .Rmd file as the **working directory**

## Parameters

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

1. **Add parameters** - Create and set parameters in the header as sub-values of params

```
params:
  n: 100
  d: Sys.Date()
```

2. **Call parameters** - Call parameter values in code as `params$d`

```
Today's date is
r params$d
```

3. **Set parameters** - Set values with Knit with parameters or the params argument of render()

render("doc.Rmd", params = list(n = 1, d = as.Date("2015-01-01")))

## Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

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



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

NOTE: Your report will be rendered as a Shiny app, which means you must choose an HTML output format, like `html_document`, and serve it with an active R Session.

## Pandoc's Markdown

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

**Plain text**  
End a line with two spaces to start a new paragraph.  
**Italics** and **bold**  
`verbatim code`  
`sub/superscript^2^2-2-`  
`---strike through---`  
`escaped: \* \, \`  
`endash: --- emdash: ---`  
`equation: SA = pi*r^2`  
`equation block:`

`SSE = mc^2/55`

`> block quote`

`# Header1 (Ranchor)`

`## Header 2 (Rcss_id)`

`### Header 3 (Rcss_class)`

`#### Header 4`

`##### Header 5`

`##### Header 6`

`<!--Text comment-->`

`<|tbl|Text ignored in HTML|`

`<em>HTML ignored in pdfs</em>`

`<http://www.rstudio.com>`

`[link](http://www.rstudio.com)`

`Jump to [Header 1](Ranchor)`

`image:`

`[[Caption]](smallorb.png)`

`* unordered list`

`+ sub-item 1`

`+ sub-item 2`

`- sub-sub-item 1`

`- sub-sub-item 2`

`Continued (indent 4 spaces)`

`1. ordered list`

`2. item 2`

`i) sub-item 1`

`A. sub-sub-item 1`

`A. sub-sub-item 1`

`§§) A list whose numbering`

`continues after`

`§§) an interruption`

`Term 1`

`Definition 1`

`- slide bullet 1`

`- slide bullet 2`

`(>- to have bullets appear on click)`

`horizontal rule/slide break`

`***`

`A footnote [^1]`

`[^1]: Here is the footnote.`

**Plain text**  
End a line with two spaces to start a new paragraph.  
**Italics** and **bold**  
`verbatim code`  
`sub/superscript^2^2-2-`  
`---strike through---`  
`escaped: \* \, \`  
`endash: --- emdash: ---`  
`equation: SA = pi*r^2`  
`equation block:`

`E = mc^2`

`> block quote`

`# Header1`

`## Header 2`

`### Header 3`

`Header 4`

`Header 5`

`Header 6`

`HTML ignored in pdfs`

`http://www.rstudio.com`

`link:`

`Jump to Header 1`

`image:`

`[[Caption]](smallorb.png)`

`* unordered list`

`+ sub-item 1`

`+ sub-item 2`

`- sub-sub-item 1`

`- sub-sub-item 2`

`Continued (indent 4 spaces)`

`1. ordered list`

`2. item 2`

`i) sub-item 1`

`A. sub-sub-item 1`

`A. sub-sub-item 1`

`§§) A list whose numbering`

`continues after`

`§§) an interruption`

`Term 1`

`Definition 1`

`- slide bullet 1`

`- slide bullet 2`

`(>- to have bullets appear on click)`

`horizontal rule/slide break`

`***`

`A footnote [^1]`

`[^1]: Here is the footnote.`



# Code Chunks



# Code Chunks

```
8  ```{r}
9  # Some code
10 dim(iris)
11  ```
```

[1] 150 5

In R notebook

In final knitted document (html, pdf, etc.)

```
# Some code
dim(iris)
```

```
## [1] 150 5
```

# Chunk Options

For code: **eval**  
determines if the  
code is evaluated

```
10 {r eval=TRUE}  
11 # Some code  
12 dim(iris)  
13 ````
```



# Chunk Options

For results: **echo** determines if the output is shown, **message** and **warning** determine if these are shown.

```
10 `{{r echo=TRUE, message=TRUE, warning=TRUE}}
11 # Some code
12 dim(iris)
13 `{{`
```



# Chunk Options

For plots: **fig.height** and **fig.width** control the size of the plot, and **fig.align** controls its position

```
10 ```{r fig.height=5, fig.width=3, fig.align='center'}  
11 # Some code  
12 plot(cars)  
13 ```
```





# Inline Code

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



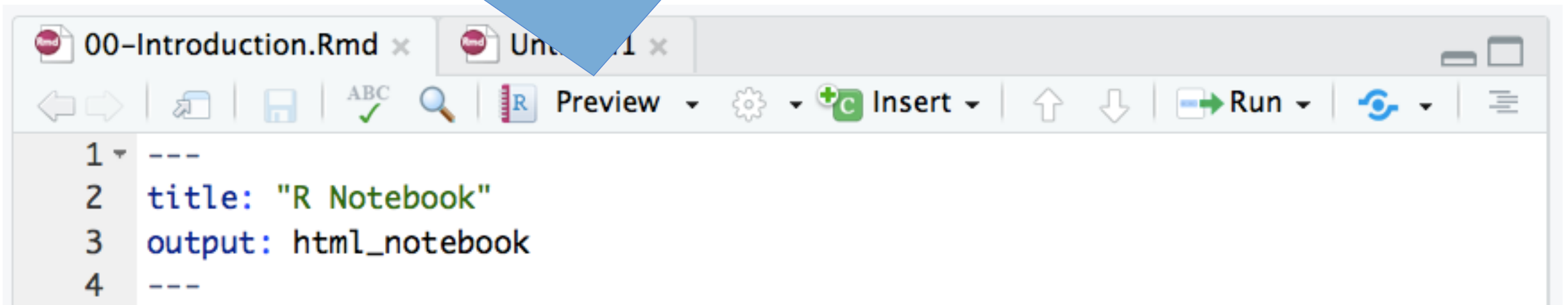
Today is  
2018-02-06.

Place code in a sentence with ``r #code``. R Markdown will replace the code with its results.

# Output Format

# Preview

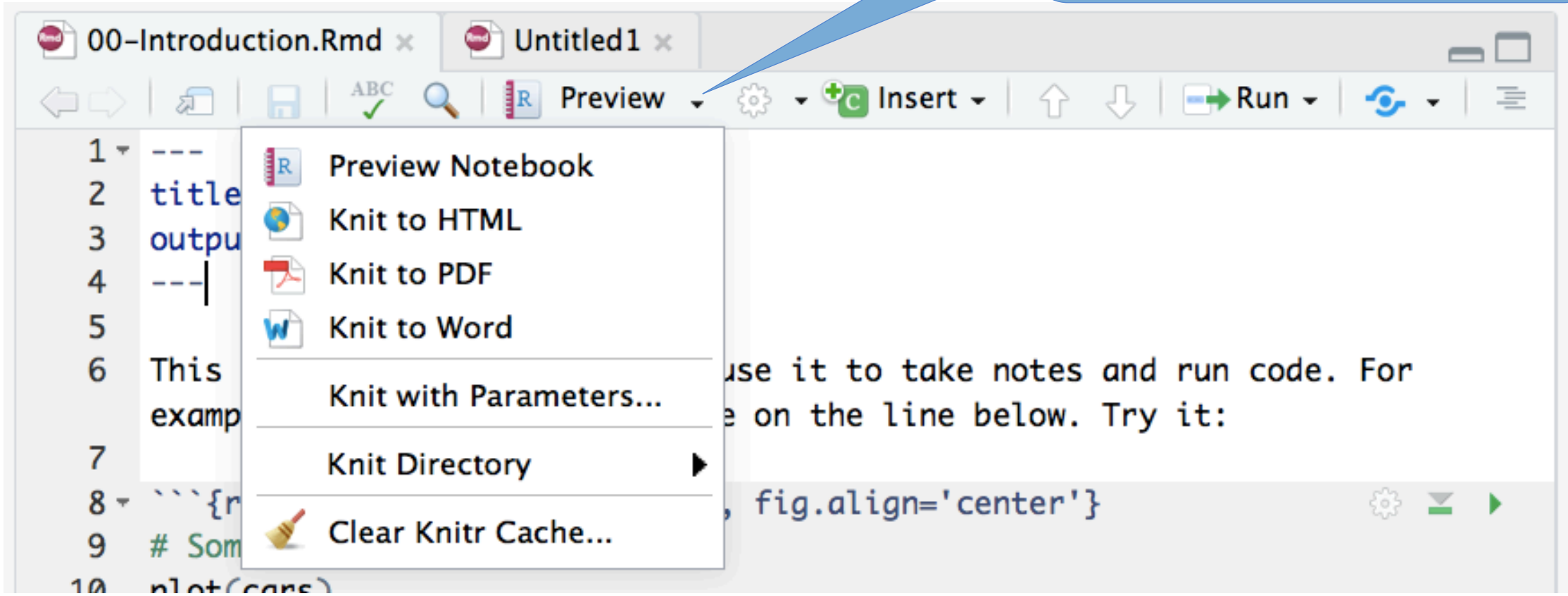
While writing your document, you can periodically **Preview** it.





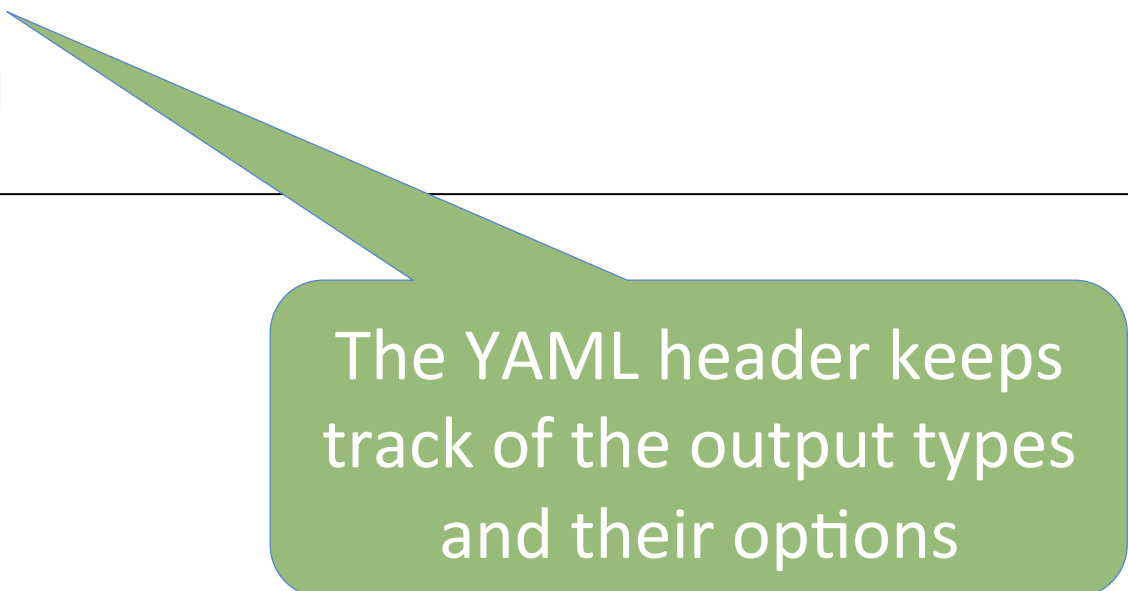
# Knit

You can knit to html, pdf, or word by clicking on the triangle



# YAML Header

```
1 ---  
2 title: "R Notebook"  
3 output:  
4   pdf_document: default  
5   html_document:  
6     df_print: paged  
7 ---
```



The YAML header keeps track of the output types and their options

# YAML Header

```
1 ---  
2 title: "R Notebook"  
3 output:  
4   pdf_document: default  
5   html_document:  
6     df_print: paged  
7   html_notebook: default  
8 ---
```

If the ability to Preview the notebook disappears, add this line to the YAML

The *setup* code chunk

# Chunk Options

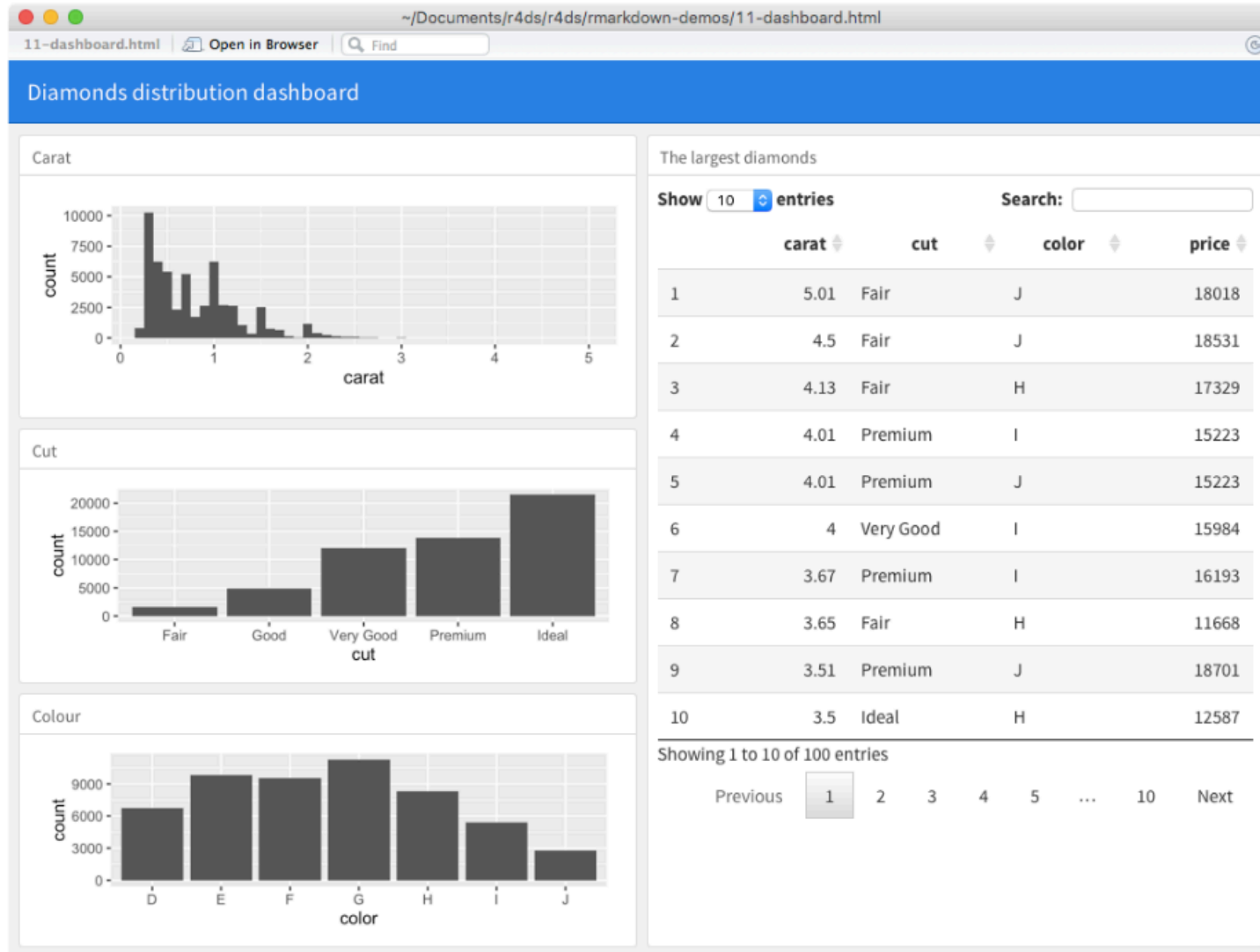
The setup code chunk is always run once before any other chunks. In this course, this will be the first code chunk that will contain the packages you are using. You can hide the warnings and messages for loading packages

```
```${r} setup, warning = FALSE, message = FALSE}  
knitr::opts_chunk$set(error = TRUE)  
  
library(tidyverse)  
```
```



Eventually...

# Interactive Webpages



# Websites

MAT 219 Data Science I Spring 2019

Syllabus

Setting Up RStudio

DataCamp

## Daily Schedule

---

*Before Jan. 22, 2019*

### 00: Getting started

#### In class

- *None*

#### DataCamp

- Use the link sent to you in an email to get access to DataCamp and make sure you are in the group **MAT 219 Data Science I Spring 2019**.

#### Reading

- [Set up RStudio](#) (Follow the directions to get access to RStudio and install the tidyverse package)

*Jan. 22, 2019*

### 01: Welcome to data science and the course

#### In class

- (Slides) [Introduction to data science](#)
- [The AWESOME process of data science: The walking dead edition](#)
- Data basics (a first look at notebooks and datasets)
- [Packages to install](#)

#### DataCamp

- (Chapter) Introduction to R: Intro to basics
- (Chapter) Working with RStudio IDE (Part 1): Orientation

#### Reading

- [Code Style](#) (Just skim through chapter 2 to get an idea of good coding practices and then refer to it as needed)
- [Documentation](#)
- [RStudio IDE Cheat sheet](#) (Just know



# Books

## R for Data Science

### Welcome

#### 1 Introduction

#### I Explore

#### 2 Introduction

#### 3 Data visualisation

#### 4 Workflow: basics

#### 5 Data transformation

#### 6 Workflow: scripts

#### 7 Exploratory Data Analysis

#### 8 Workflow: projects

#### II Wrangle

#### 9 Introduction

#### 10 Tibbles

#### 11 Data import

#### 12 Tidy data

#### 13 Relational data

#### 14 Strings

#### 15 Factors



R for Data Science

## R for Data Science

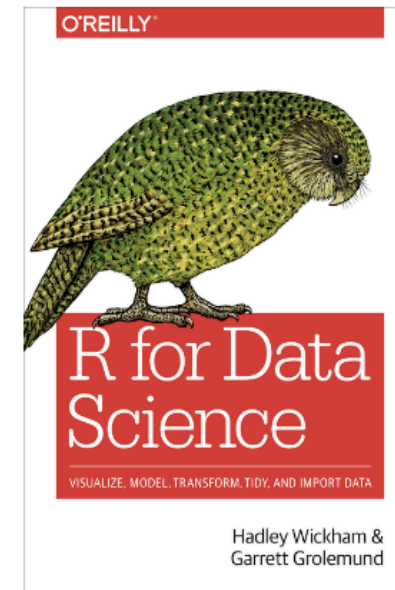
*Garrett Grolemund*

*Hadley Wickham*

## Welcome

This is the website for “**R for Data Science**”. This book will teach you how to do data science with R: You’ll learn how to get your data into R, get it into the most useful structure, transform it, visualise it and model it. In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you’ll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You’ll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You’ll also learn how to manage cognitive resources to facilitate discoveries when wrangling, visualising, and exploring data.

This website is (and will always be) **free to use**, and is licensed



# Journal Articles

## Overview

---

The `rticles` package provides a suite of custom [R Markdown](#) LaTeX formats and templates for various formats, including:

- [ACM](#) articles
- [ACS](#) articles
- [AEA](#) journal submissions
- [AMS](#) articles
- [Biometrics](#) articles
- [Bulletin de l'AMQ](#) journal submissions
- [CTeX](#) documents
- [Elsevier](#) journal submissions
- [IEEE Transaction](#) journal submissions
- [JSS](#) articles
- [MDPI](#) journal submissions
- [Monthly Notices of the Royal Astronomical Society](#) articles
- [NNRAS](#) journal submissions
- [PeerJ](#) articles
- [Royal Society Open Science](#) journal submissions
- [Sage](#) journal submissions