RStudio Toolchain for Instructors

Garrett Grolemund

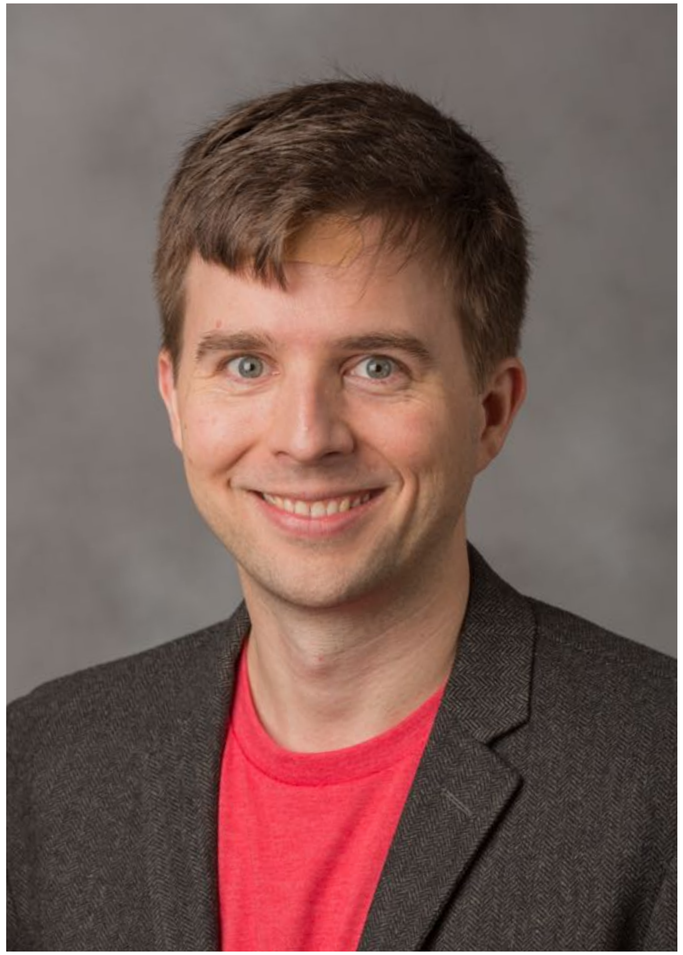
Master Instructor, RStudio May 2018

Photo by Fischer Twins on Unsplash

Warm Up

Introduce yourself to the people next to you. Tell them:

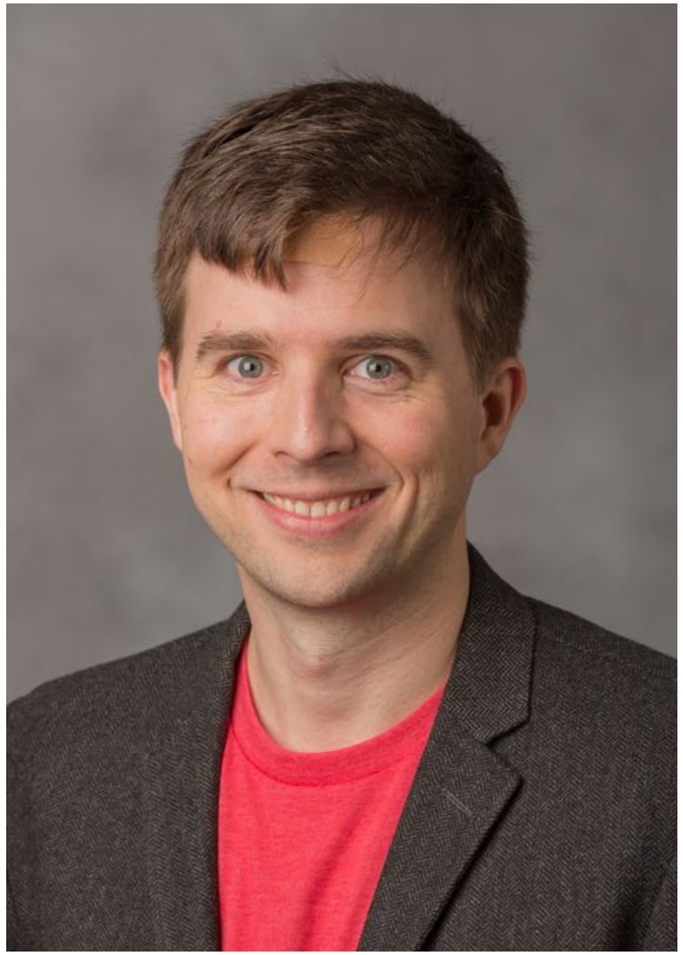
1. Who you are
2. What you teach
3. How long you have been using R

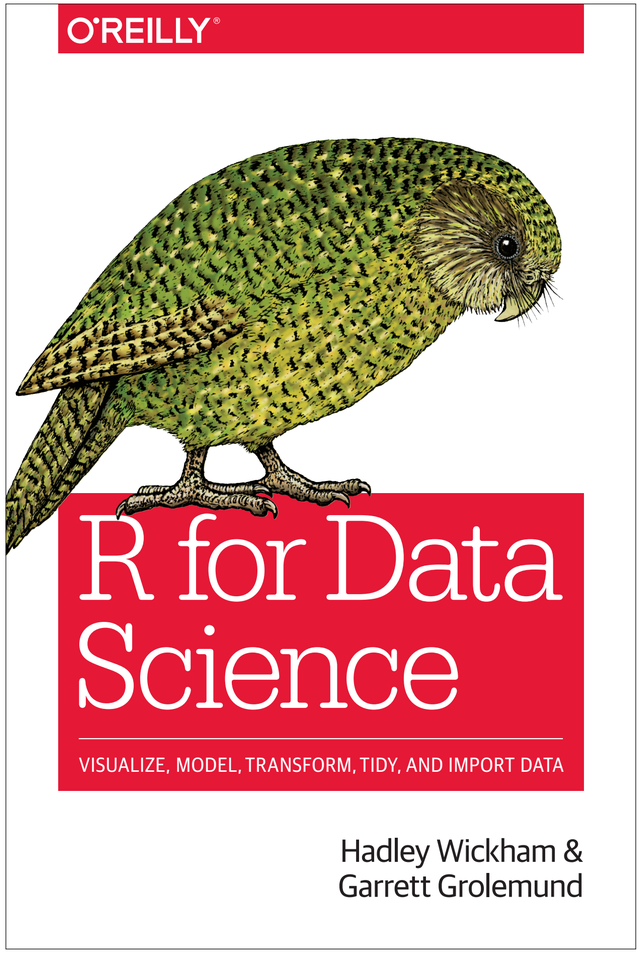


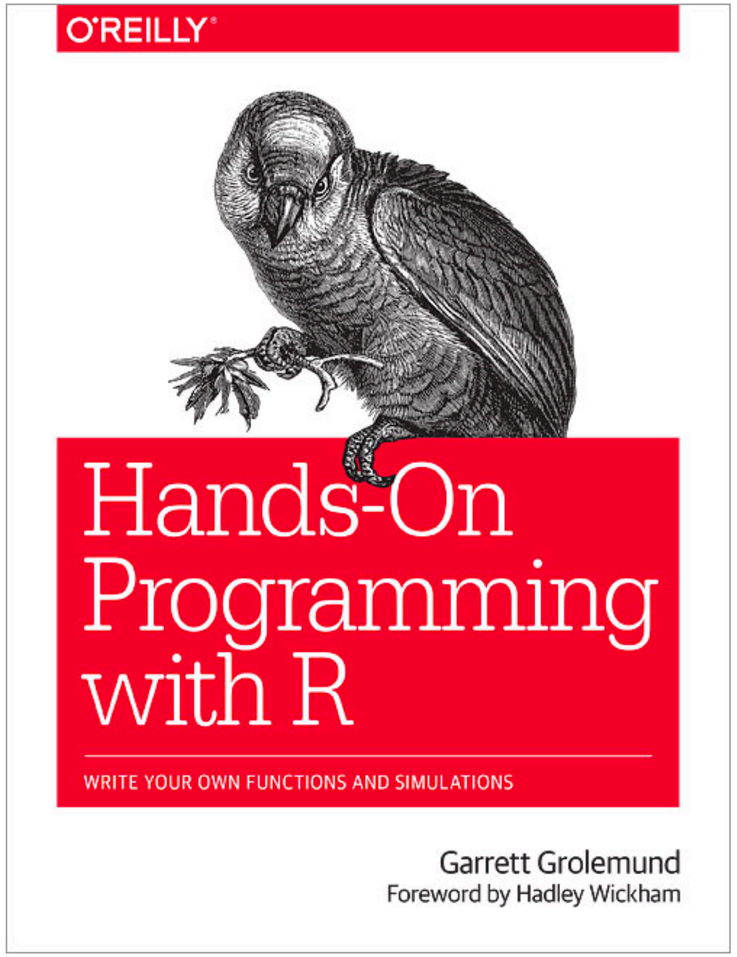
HELLO
my name is

Garrett

@StatGarrett




O'REILLY
R for Data Science
VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA
Hadley Wickham &
Garrett Grolemund


O'REILLY
Hands-On Programming with R
WRITE YOUR OWN FUNCTIONS AND SIMULATIONS
Garrett Grolemund
Foreword by Hadley Wickham

Why R?

Photo by Johnson Wang on Unsplash

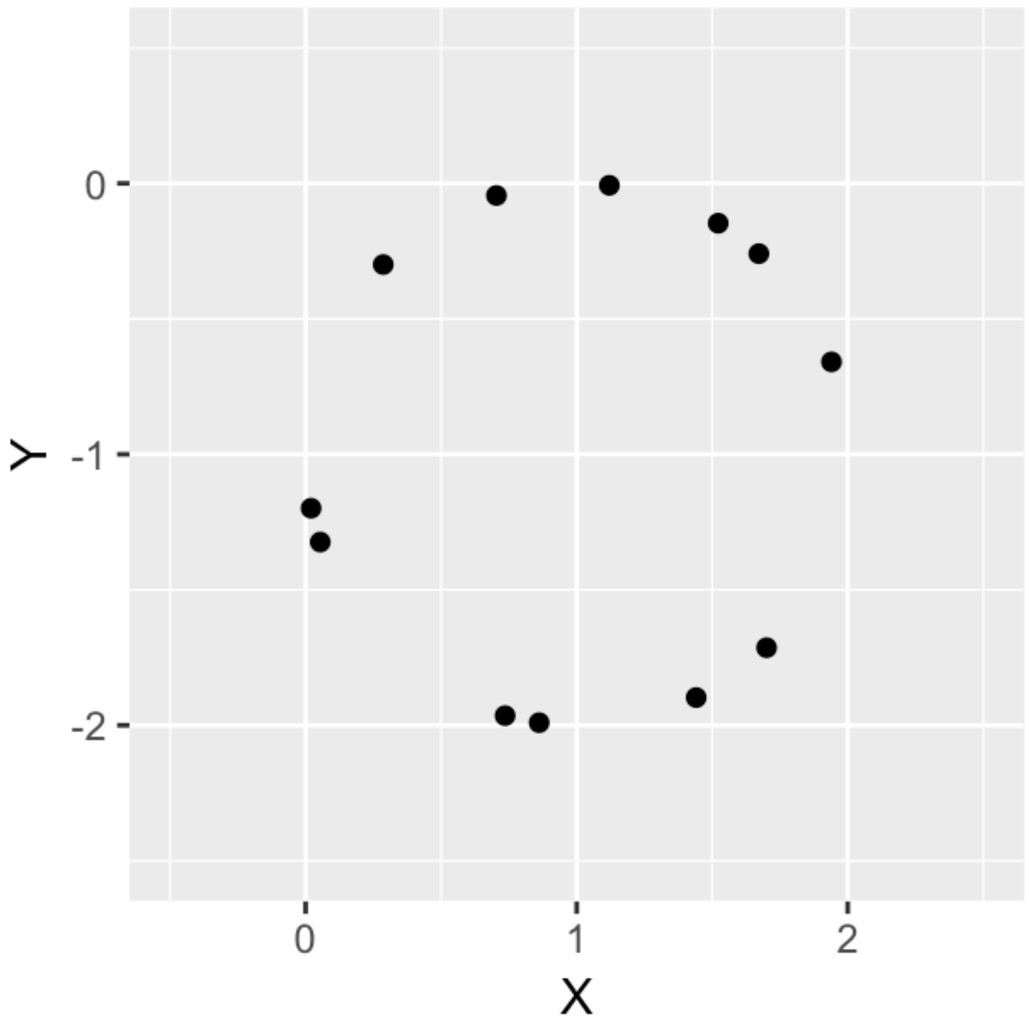
1. Data requires **computation**

X <dbl>	Y <dbl>
0.86	-1.99
0.02	-1.20
0.05	-1.32
1.70	-1.71
1.12	-0.01
1.44	-1.90
1.52	-0.15
0.74	-1.96
0.29	-0.30
0.70	-0.04

Volunteer?

1. Data requires **computation**

X <dbl>	Y <dbl>
0.86	−1.99
0.02	−1.20
0.05	−1.32
1.70	−1.71
1.12	−0.01
1.44	−1.90
1.52	−0.15
0.74	−1.96
0.29	−0.30
0.70	−0.04

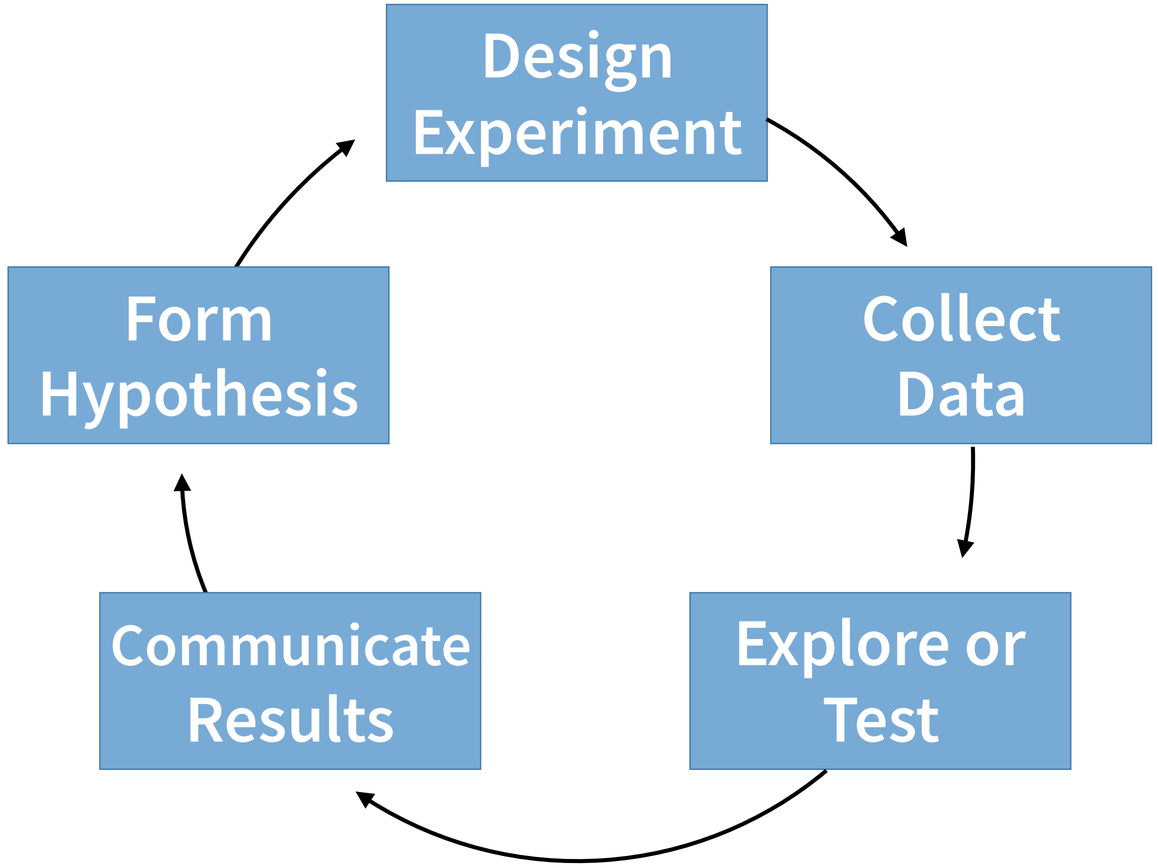


1. Data requires **computation**
2. Science requires **reproducibility**

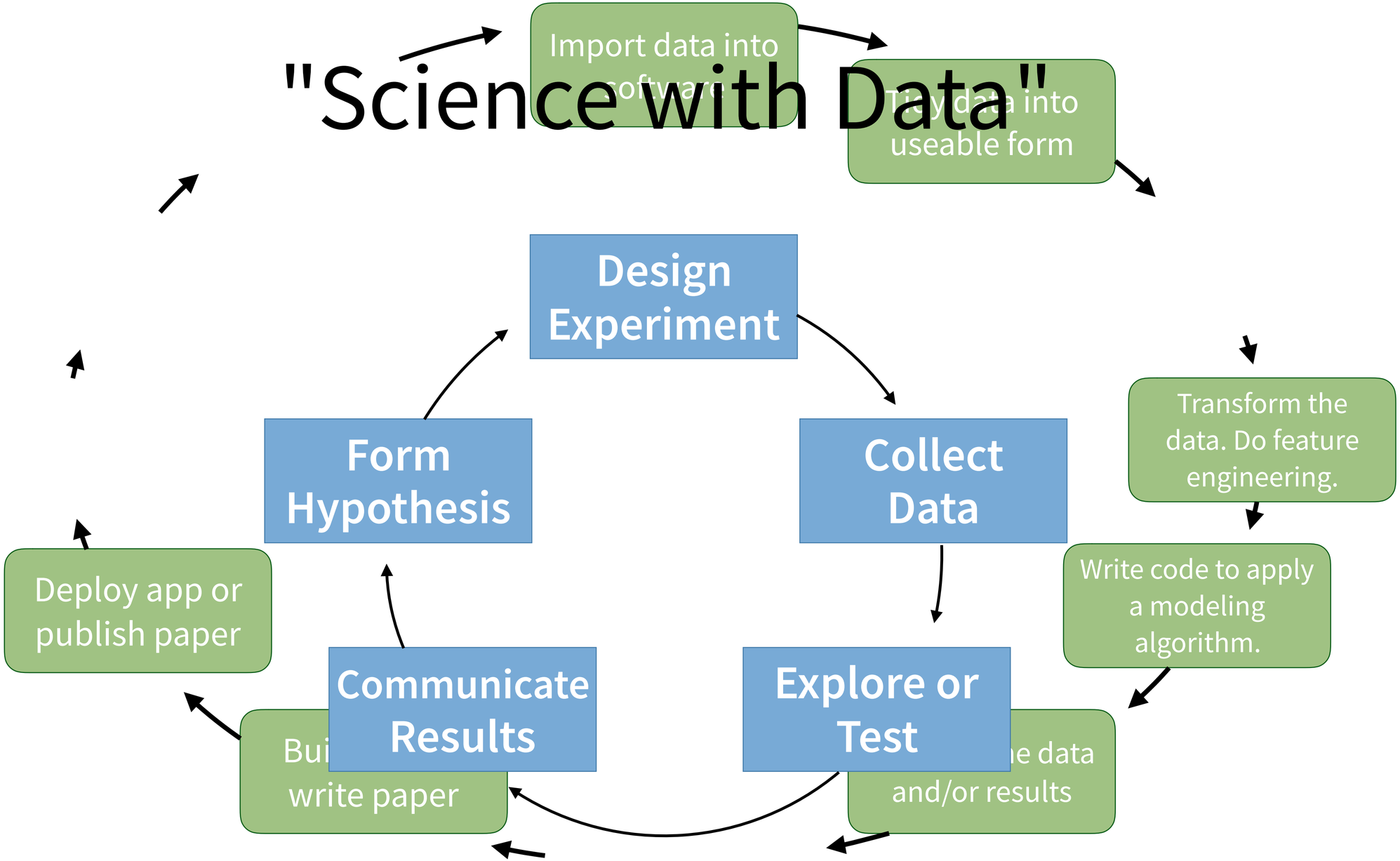
Data Science = Reproducible computation?

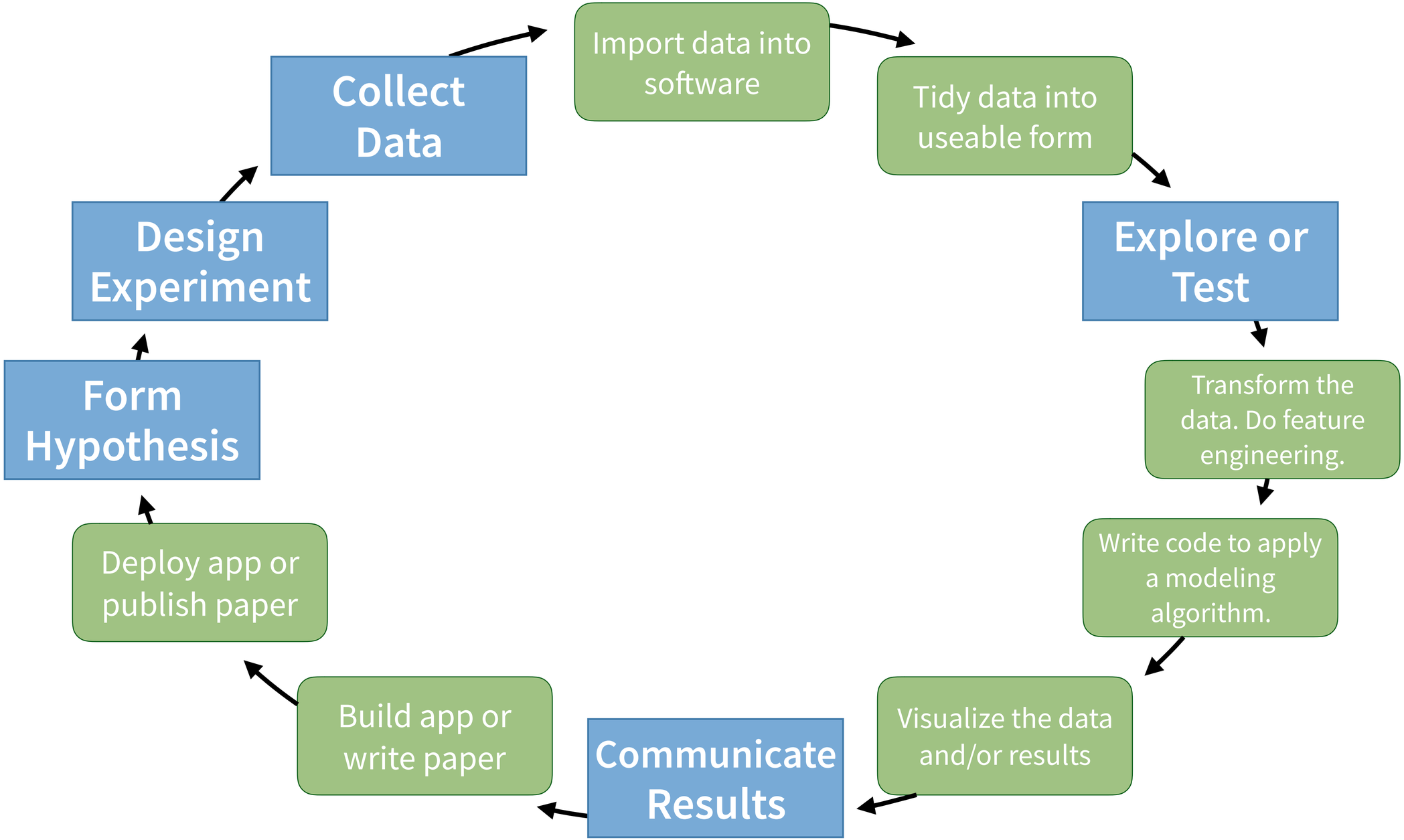
1. Data requires **computation**
2. Science requires **reproducibility**
3. *Code is the most reproducible way to compute*

"Science with Data"

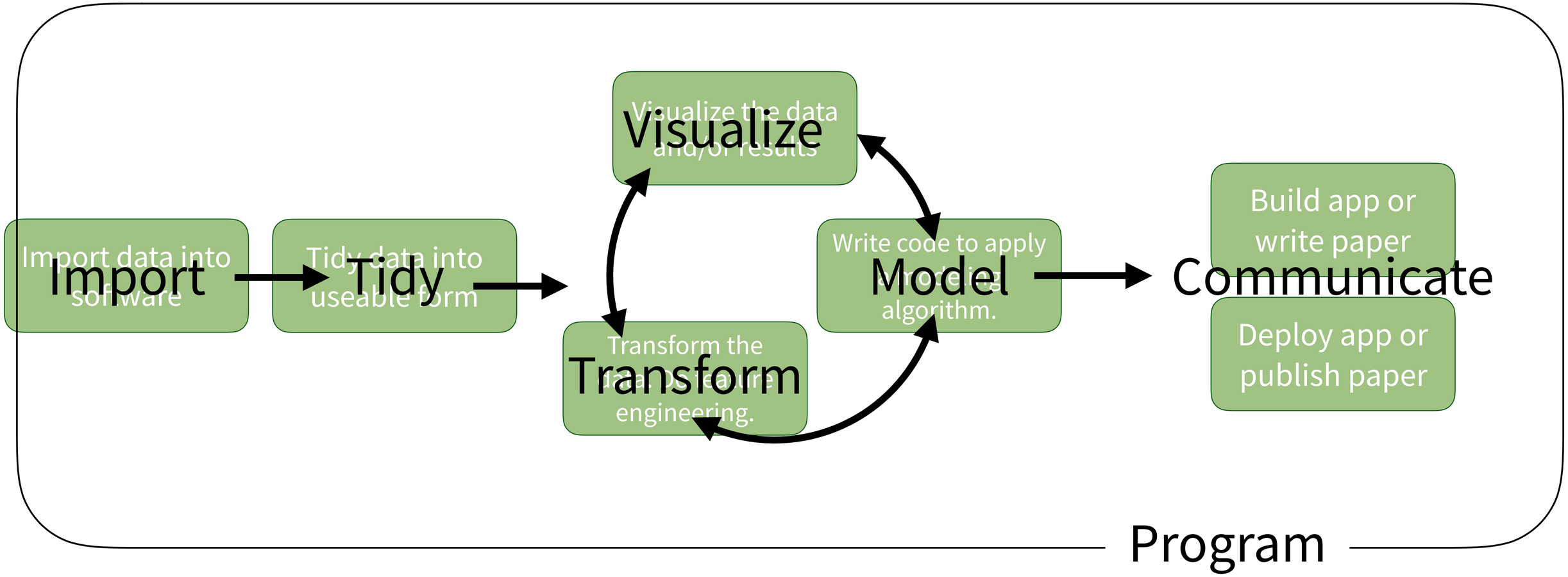


"Science with Data"

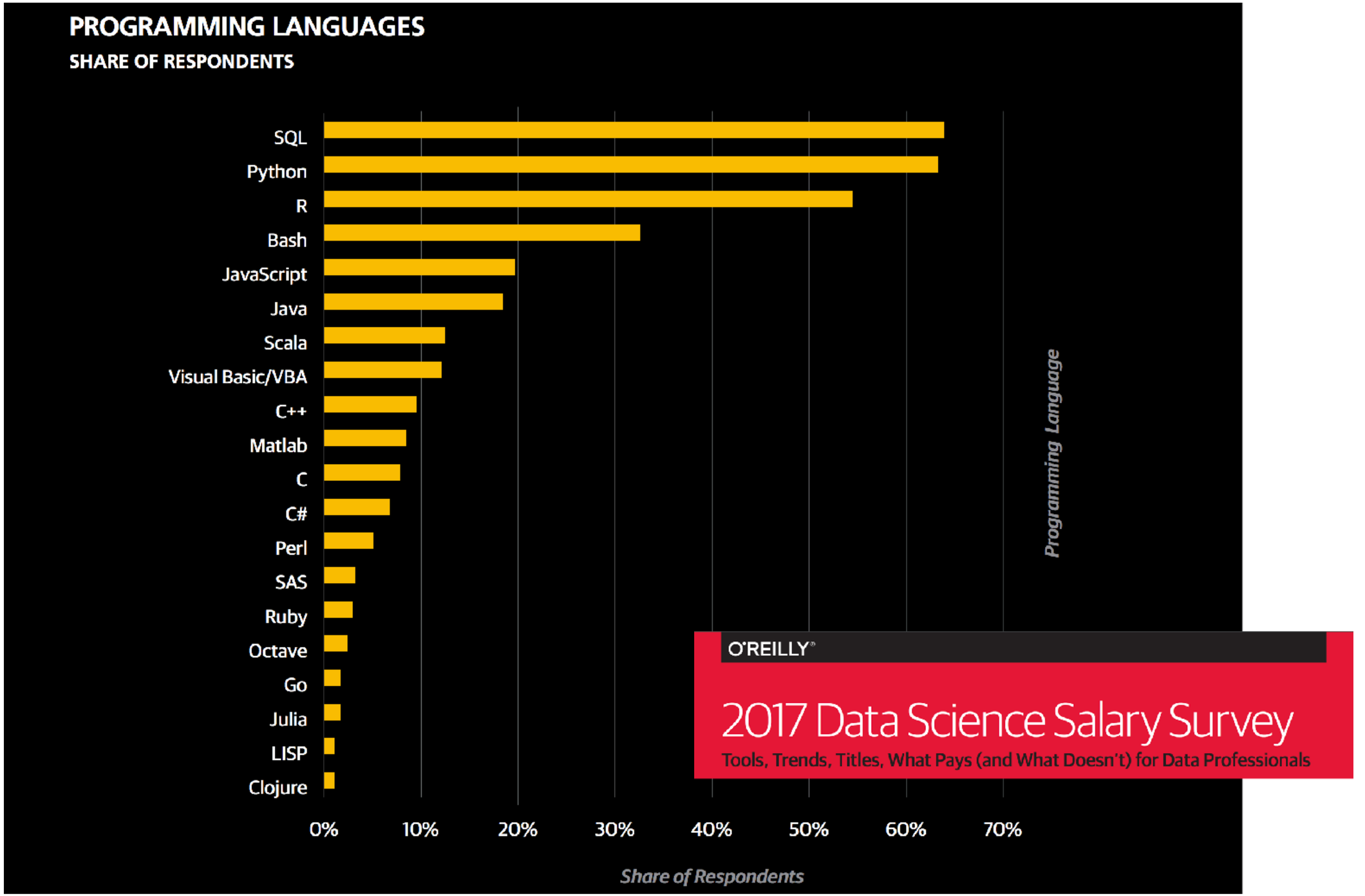



Collect Data
Import data into software
Tidy data into useable form
Explore or Test
Transform the data. Do feature engineering.
Write code to apply a modeling algorithm.
Visualize the data and/or results
Communicate Results
Build app or write paper
Deploy app or publish paper
Form Hypothesis
Design Experiment

(Applied) **Data Science**



1. Data requires **computation**
2. Science requires **reproducibility**
3. *Code is the most reproducible way to compute*
4. The R language is **particularly well suited** to data science


PROGRAMMING LANGUAGES
SHARE OF RESPONDENTS
SQL
Python
R
Bash
JavaScript
Java
Scala
Visual Basic/VBA
C++
Matlab
C
C#
Perl
SAS
Ruby
Octave
Go
Julia
LISP
Clojure
0%
10%
20%
30%
40%
50%
60%
70%
Share of Respondents
Programming Language
O'REILLY
2017 Data Science Salary Survey
Tools, Trends, Titles, What Pays (and What Doesn't) for Data Professionals

Why R?

1. Data requires **computation**
2. Science requires **reproducibility**
3. *Code is the most reproducible way to compute*
4. The R language is **particularly well suited** to data science

Photo by Johnson Wang on Unsplash

Quiz

Where do you get R?

Quiz

Where do you get R?

cran.r-project.org

Quiz

Where do you get RStudio?

Quiz

Where do you get RStudio?

www.rstudio.com/download

Quiz

Where do you get R Packages?

Quiz

Where do you get R Packages?

cran.r-project.org

github.com

(from R)

Quiz

Should you have students install R?

Your Turn

Should you have students install R?

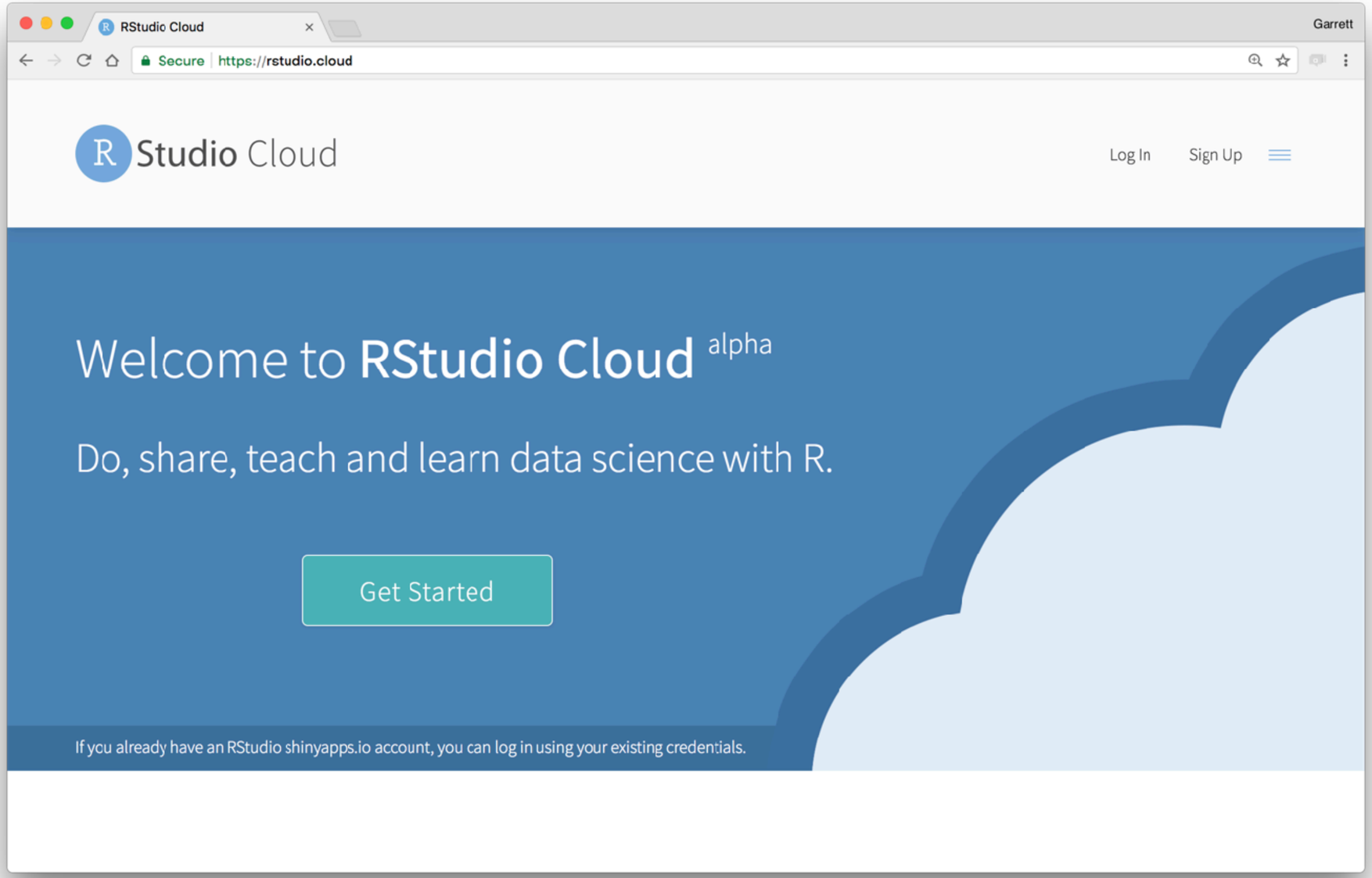
Think of a couple of good reasons to have students install R and a couple of good reasons to **not** have them install R.

After 2 minutes, debate with the people next to you and vote on one or another.

RStudio Cloud

Photo by Jason Blackeye on Unsplash

https://**rstudio.cloud**



Your Turn

Create an account and then Log in to

https://rstudio.cloud

Open a new project.

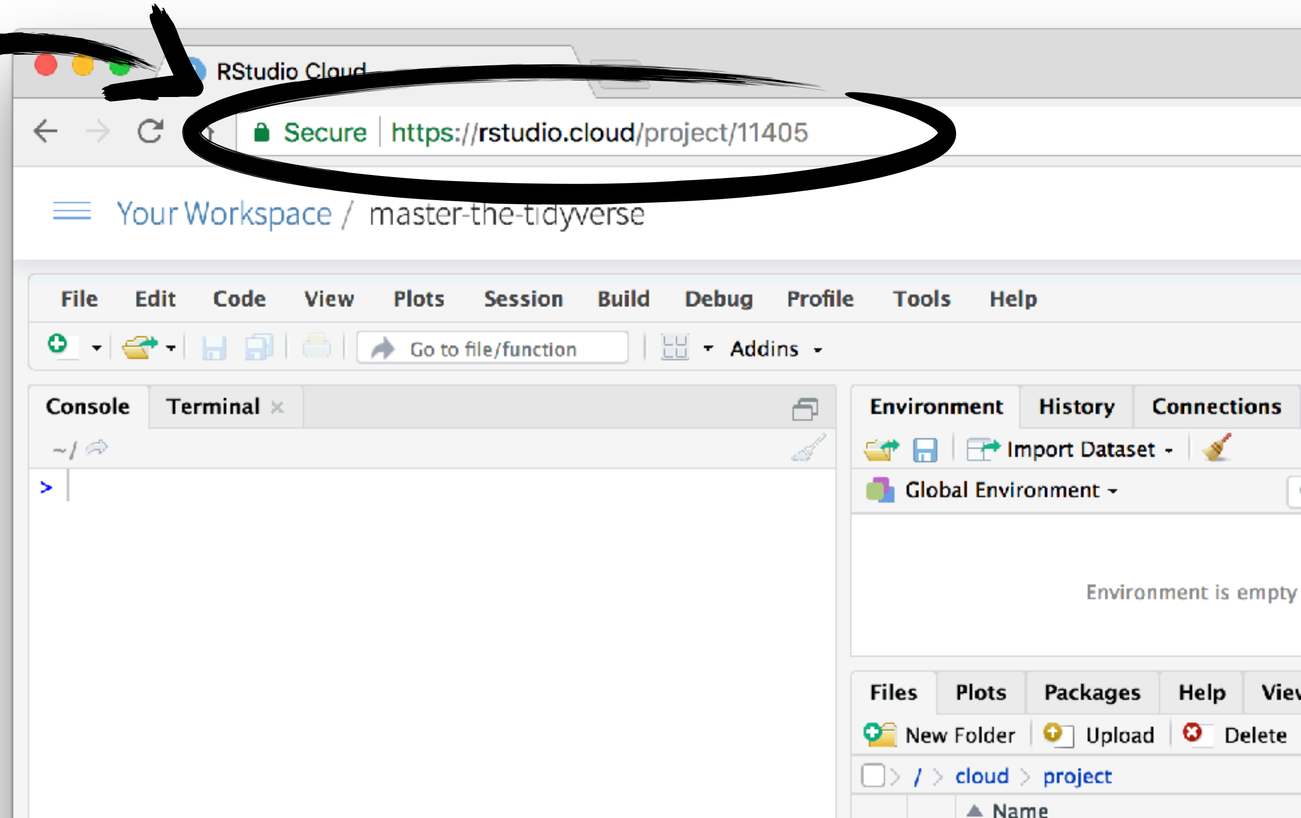
Then open a fresh R Script.

Write and run some code (of your choice).

Sharing projects

Volunteer?

What is the URL of your project?



Your Turn

Check whether your project has the **grader** package loaded, e.g. does this work?

library(grader)

Your Turn

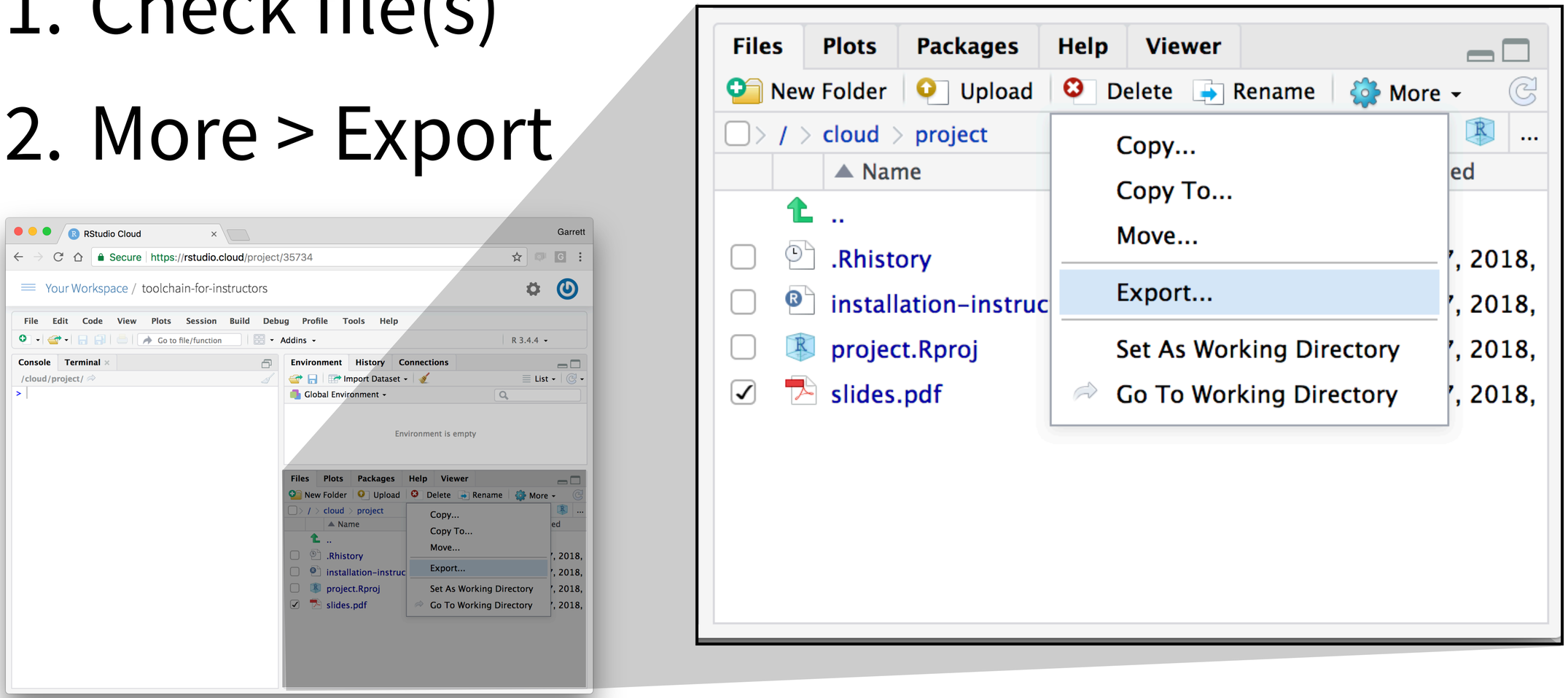
Open a copy of my project. Save it.

Is grader loaded?

rstudio.cloud/project/35734

To Download Files

1. Check file(s)
2. More > Export



Spaces

Space

1. A private repository for projects
2. Invite collaborators
3. Set roles
4. Set a template for new projects in the space

Space

1. A private repository for projects
2. Invite collaborators
3. Set roles
4. Set a template for new projects in the space

One free.
Cannot delete.

Your Turn

Fill in the blanks on your RStudio Cloud handout. Then complete the activity at the end.

RStudio Cloud

1. **No installation**
2. **Sharable** projects
3. **Private** Spaces
4. https://rstudio.cloud

Photo by Jason Blackeye on Unsplash

R Markdown

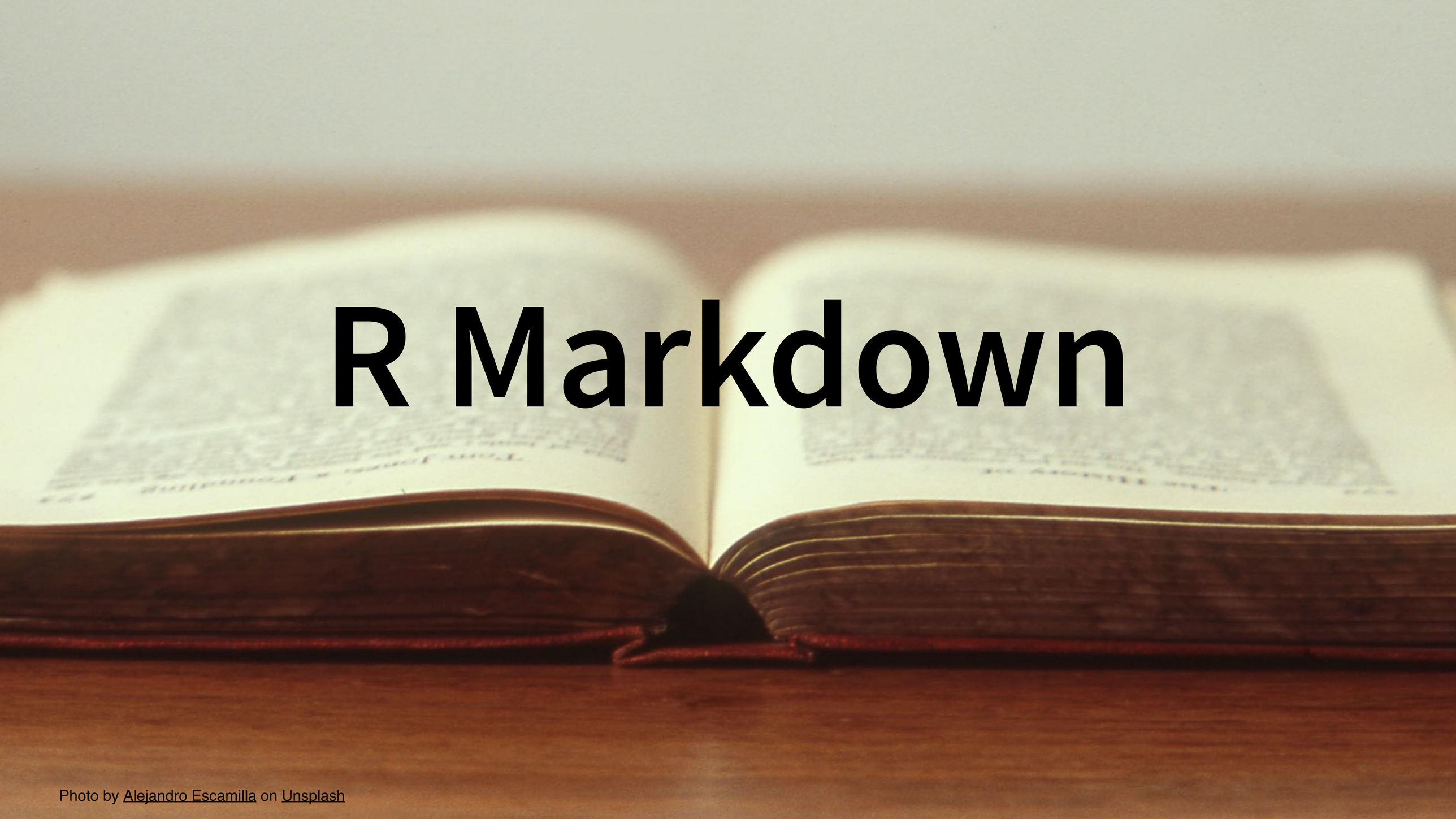
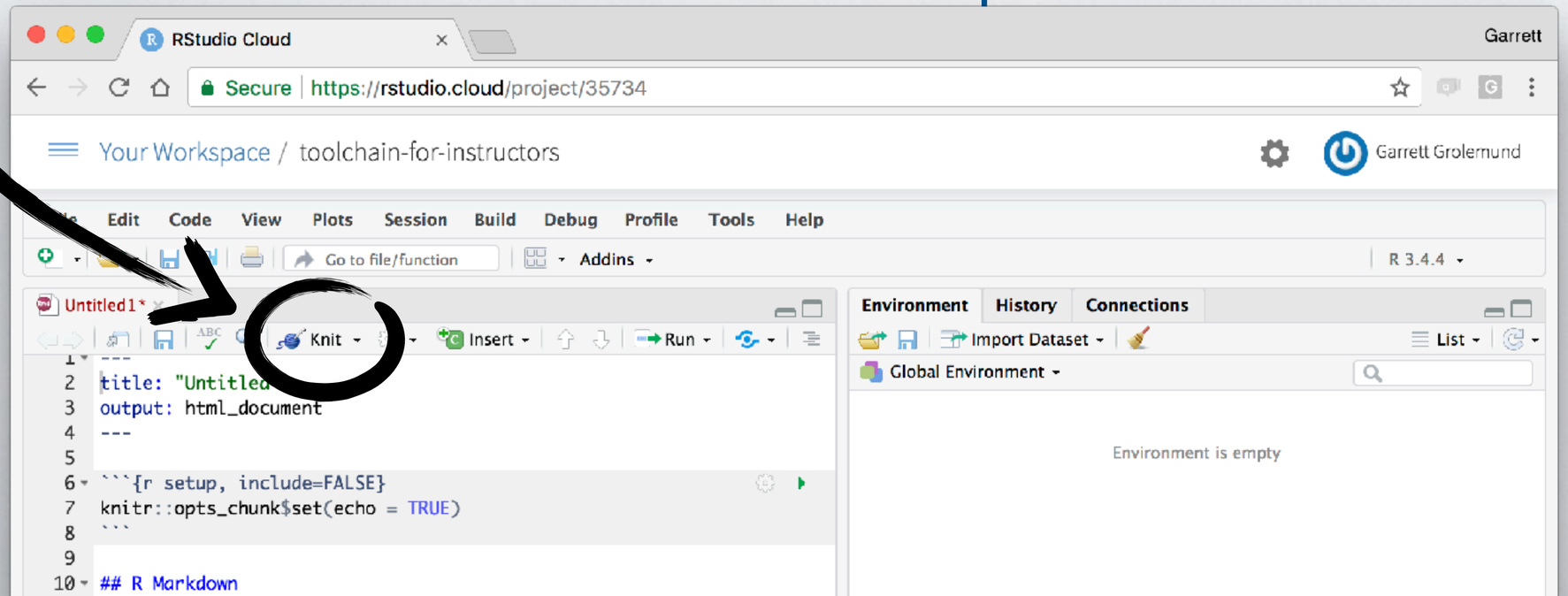


Photo by Alejandro Escamilla on Unsplash

Your Turn

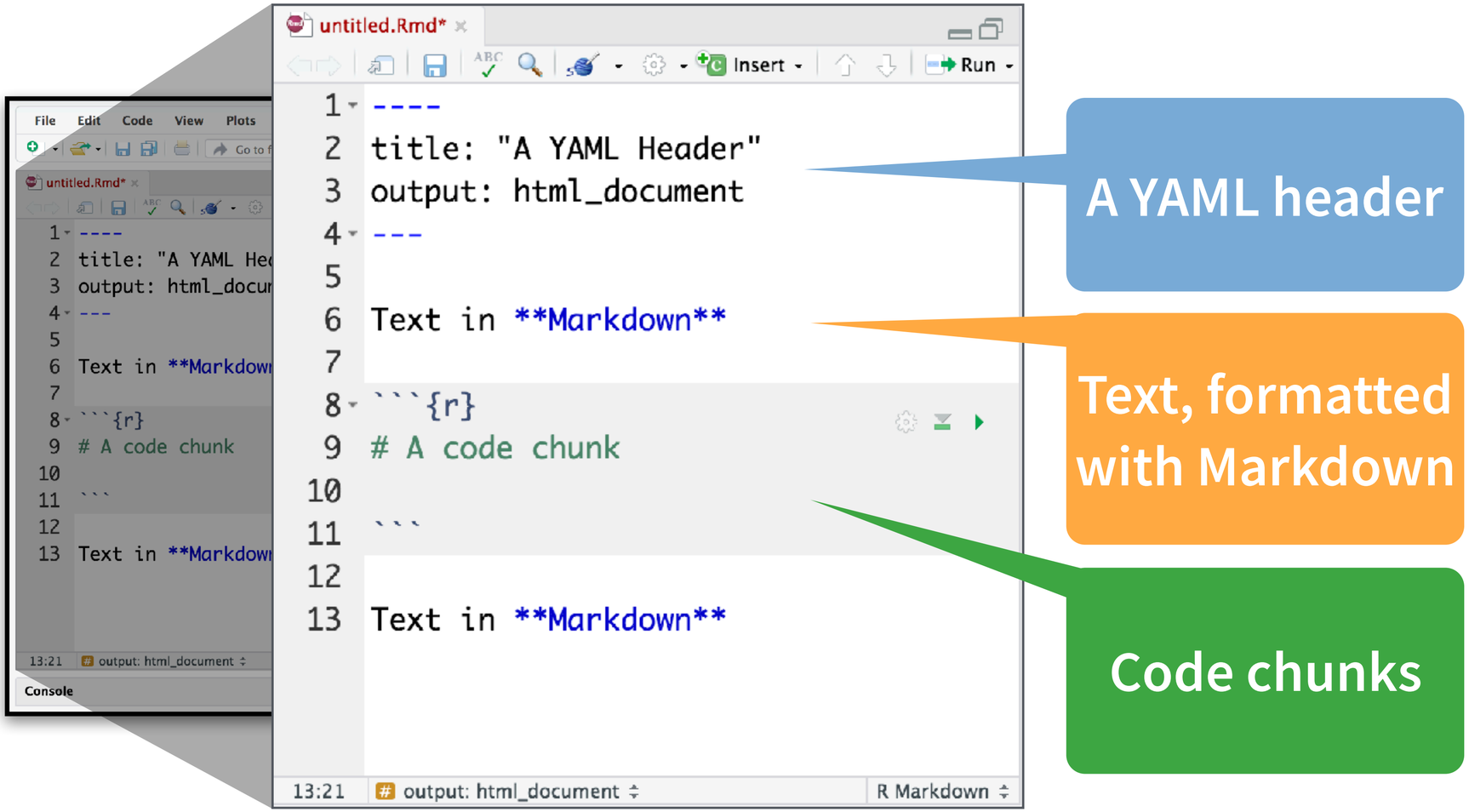
Navigate to **rstudio.cloud/project/35734** and create a new project, open an R Markdown file (go to File > New File > R Markdown).

Knit the file into finished HTML output. Read it.



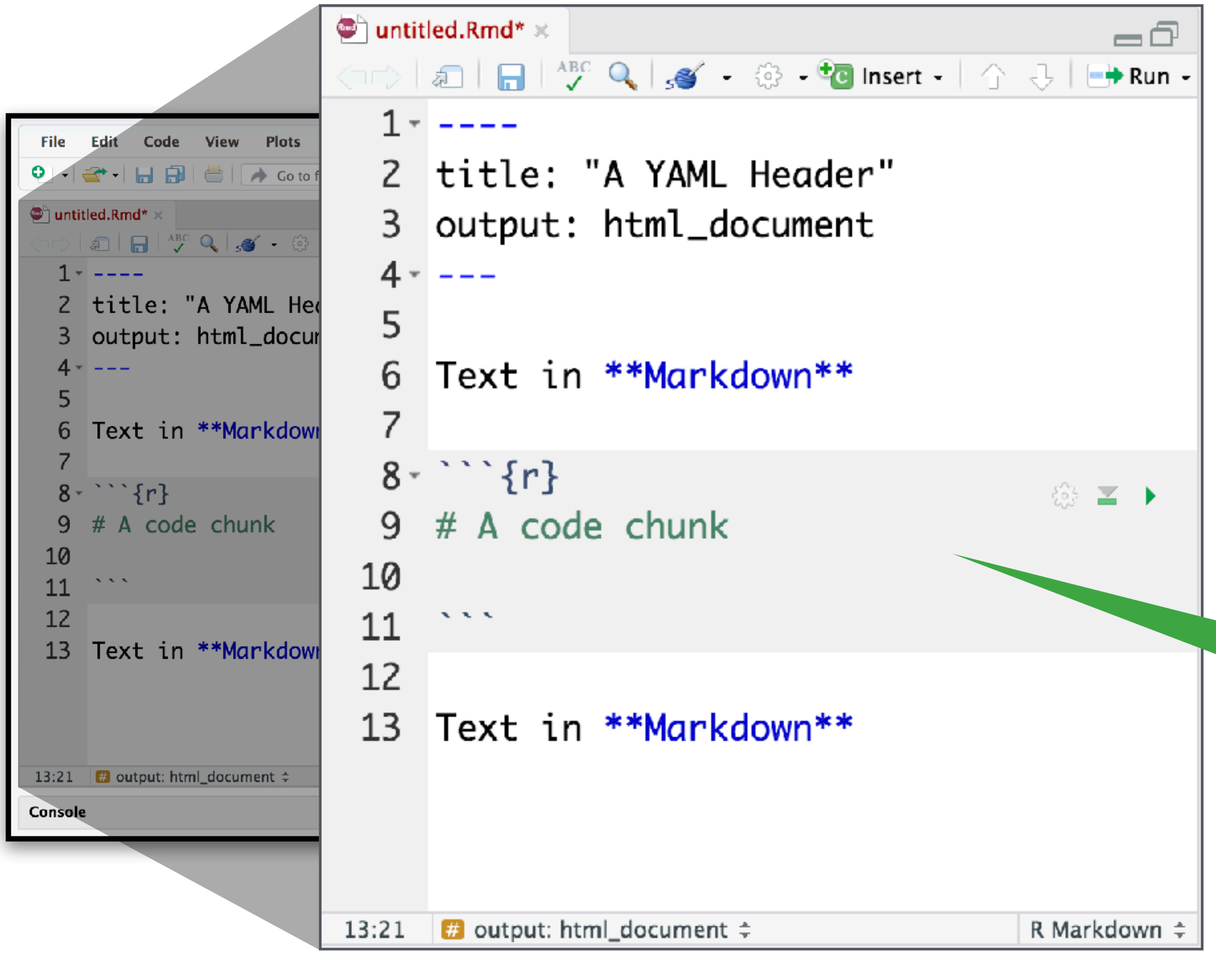
Structure

R Markdown documents contain **three types of content**



Code Chunks

Write and execute code in a **chunk**. Insert with

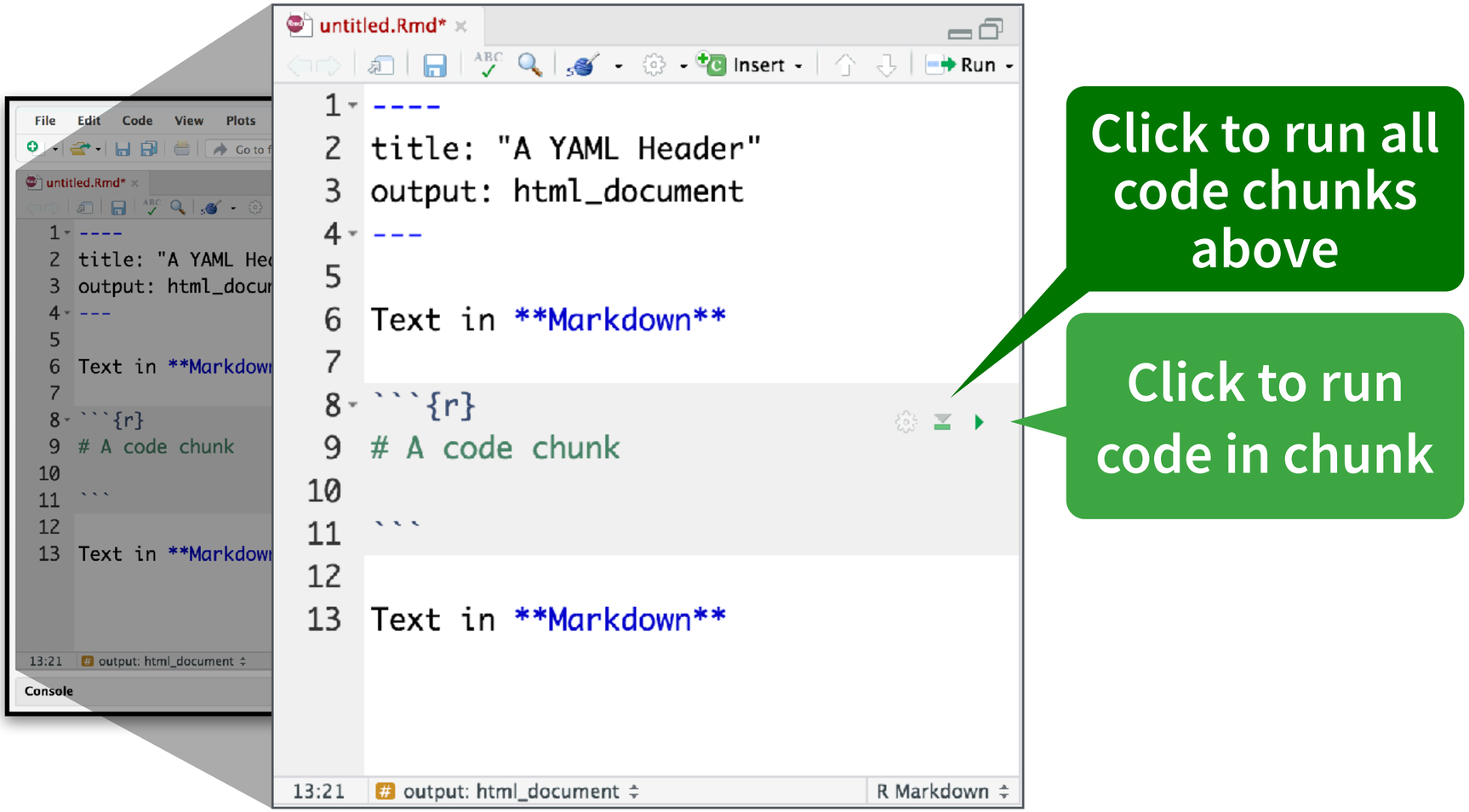


- **Command + Opt + I**
- **Control + Alt + I**
- GUI Insert button
- Typing out the tick marks

Code chunks

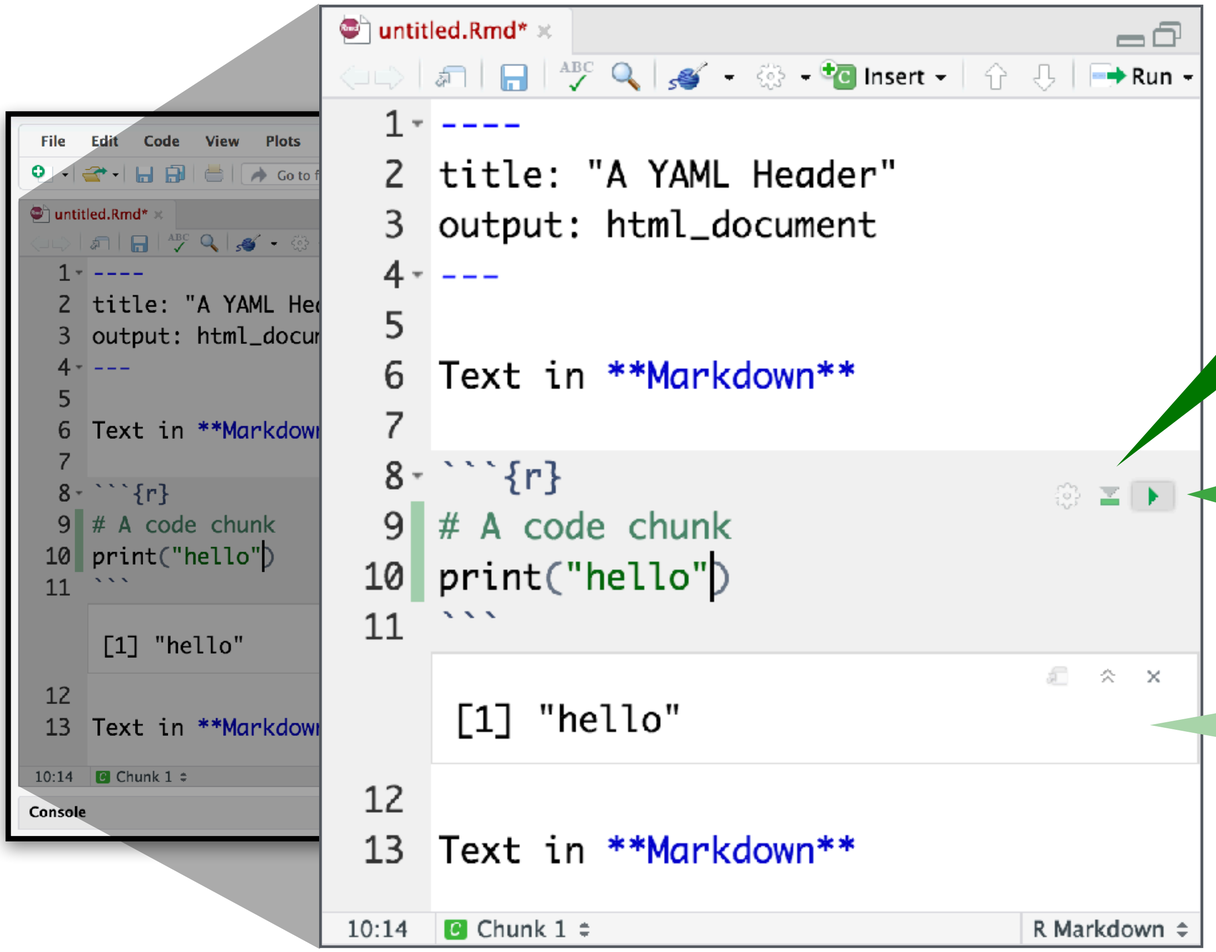
Code Chunks

Write and execute code in a **chunk**.



Code Chunks

Write and execute code in a **chunk**.



Click to run all code chunks above

Click to run code in chunk

Code result

Your Turn

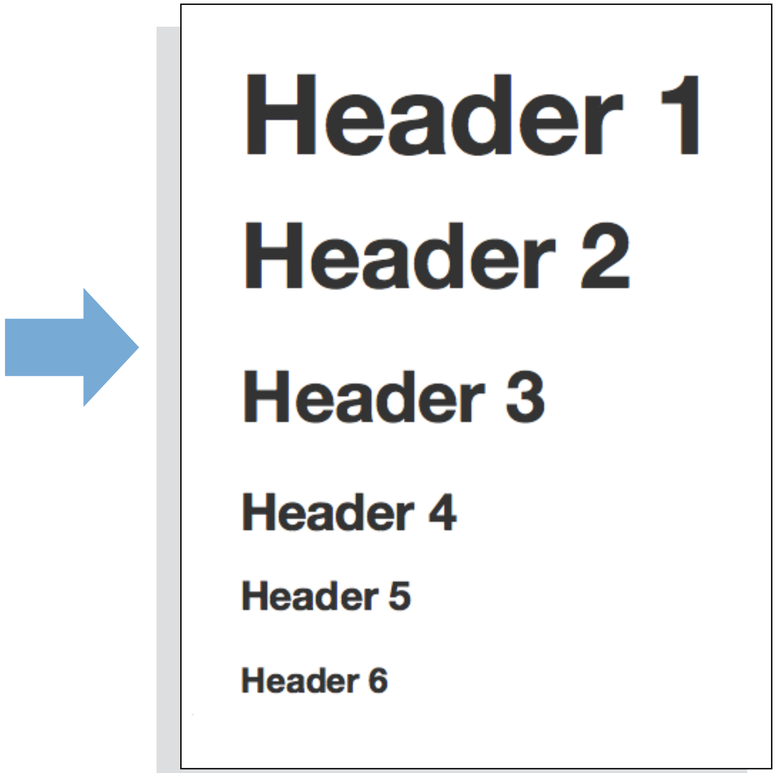
Add a new chunk at the end of your document.
Use it to run:

```
hist(rnorm(100))
```

Reporting

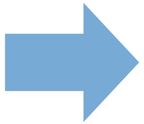
Headers

```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
###### Header 6
```



Text

```
Text
_italics_
__bold__
`code`
```



Text

italics

bold

`code`

Lists

```
Bullets

* bullet 1
* bullet 2

Numbered list

1. item 1
2. item 2
```

Bullets

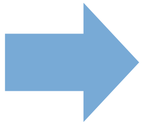
- bullet 1
- bullet 2

Numbered list

1. item 1
2. item 2

Hyperlinks

```
This is a
[link](www.git.com).
```



This is a link.

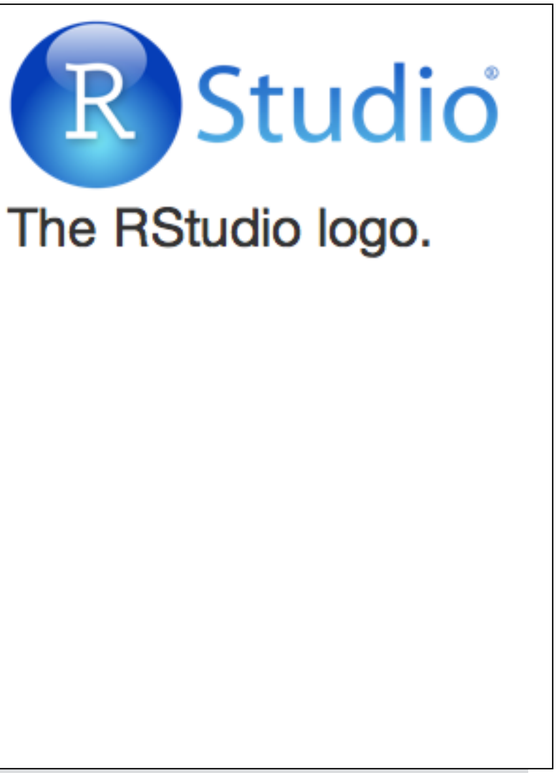
Equations

```
According to
Einstein,
$E=mc^{2}$
```

According to
Einstein, $E = mc^2$

Images

```
![](images/1.png)
The RStudio logo.
```



Code chunks

```
Here's some code
```{r}
dim(iris)
```
```

Here's some code

```
dim(iris)
```

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

Chunk Options

```
Here's some code
```{r echo=FALSE}
dim(iris)
```
```

Here's some code

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

echo = FALSE

```
Here's some code
```{r echo=FALSE}
dim(iris)
```
```

Here's some code

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

Displays code results, but **not code**

eval = FALSE

```
Here's some code
```{r eval=FALSE}
dim(iris)
```
```

Here's some code

```
dim(iris)
```

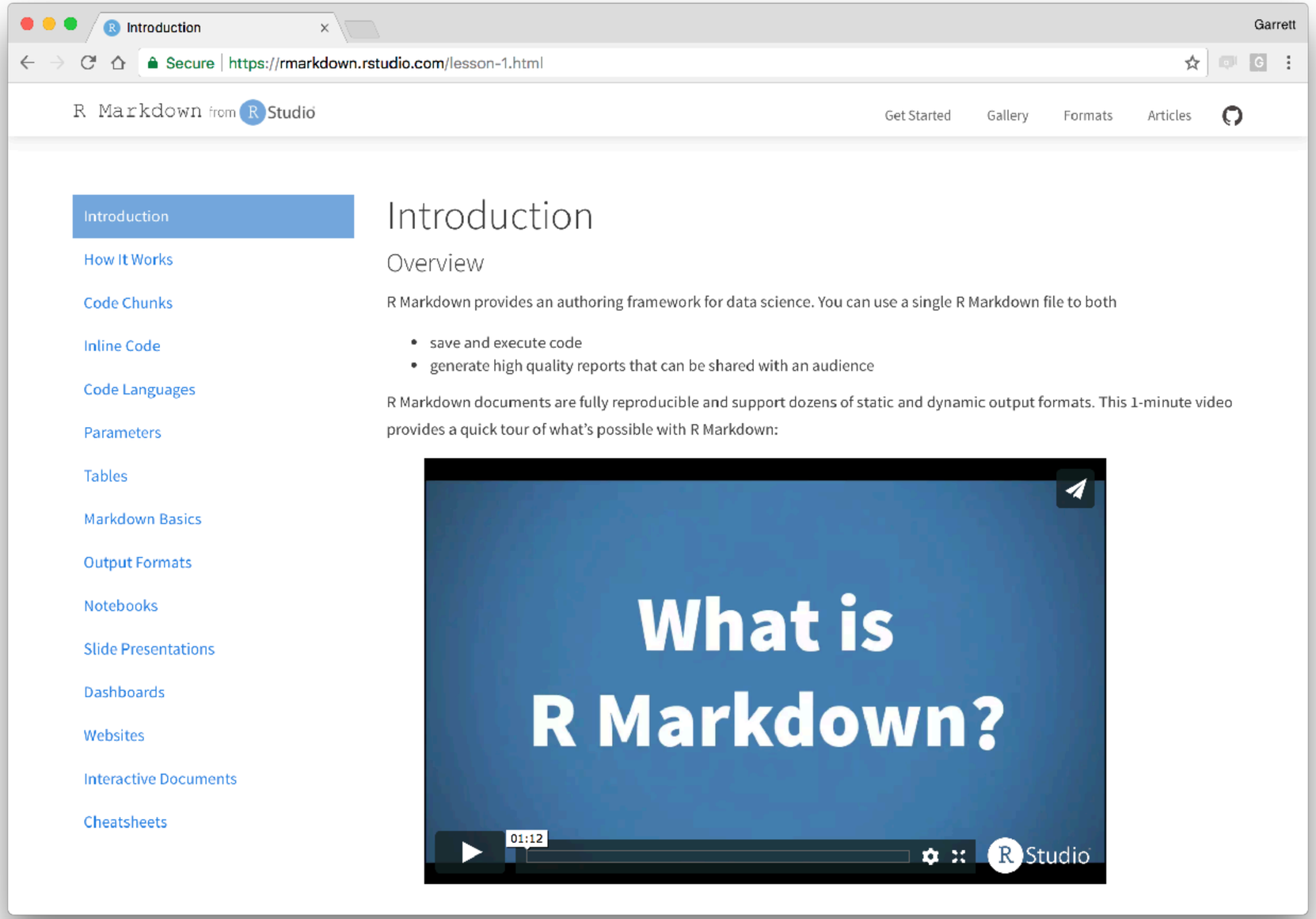
Displays code, but **not results** (code is not run)

include = FALSE

```
Here’s some code
```{r include=FALSE}
dim(iris)
```
```

Displays **neither code not results** (but code is run)

rmarkdown.rstudio.com/lesson-1.html



Quiz

What does an **R Markdown** file reproduce that **code alone** does not?

Quiz

What does an **R Markdown** file reproduce that **code alone** does not?

Your reasoning and conclusions

Quiz

What does an **R Markdown** file reproduce that **a report alone** does not?

Quiz

What does an **R Markdown** file reproduce that **a report alone** does not?

Your code/work

R Markdown

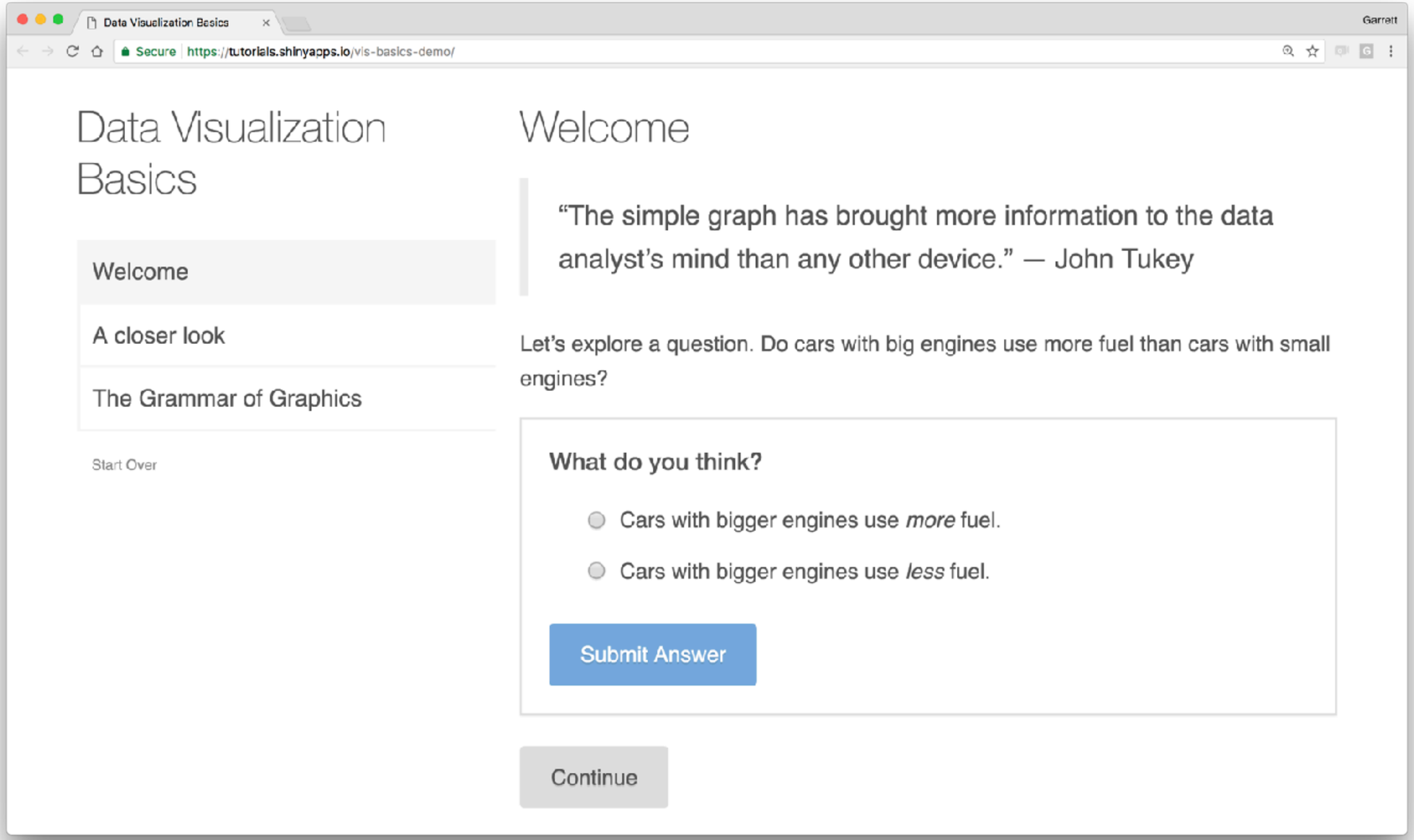
1. Easy **reports**
2. Easy **grading**
3. Always **reproducible**

Photo by Alejandro Escamilla on Unsplash

Learnr

Photo by Annie Spratt on Unsplash

https://**tutorials.shinyapps.io/vis-basics-demo/**



Your Turn

https://**tutorials.shinyapps.io/vis-basics-demo/**

Visit the website above, and work through as much of the tutorial as you can in three minutes.

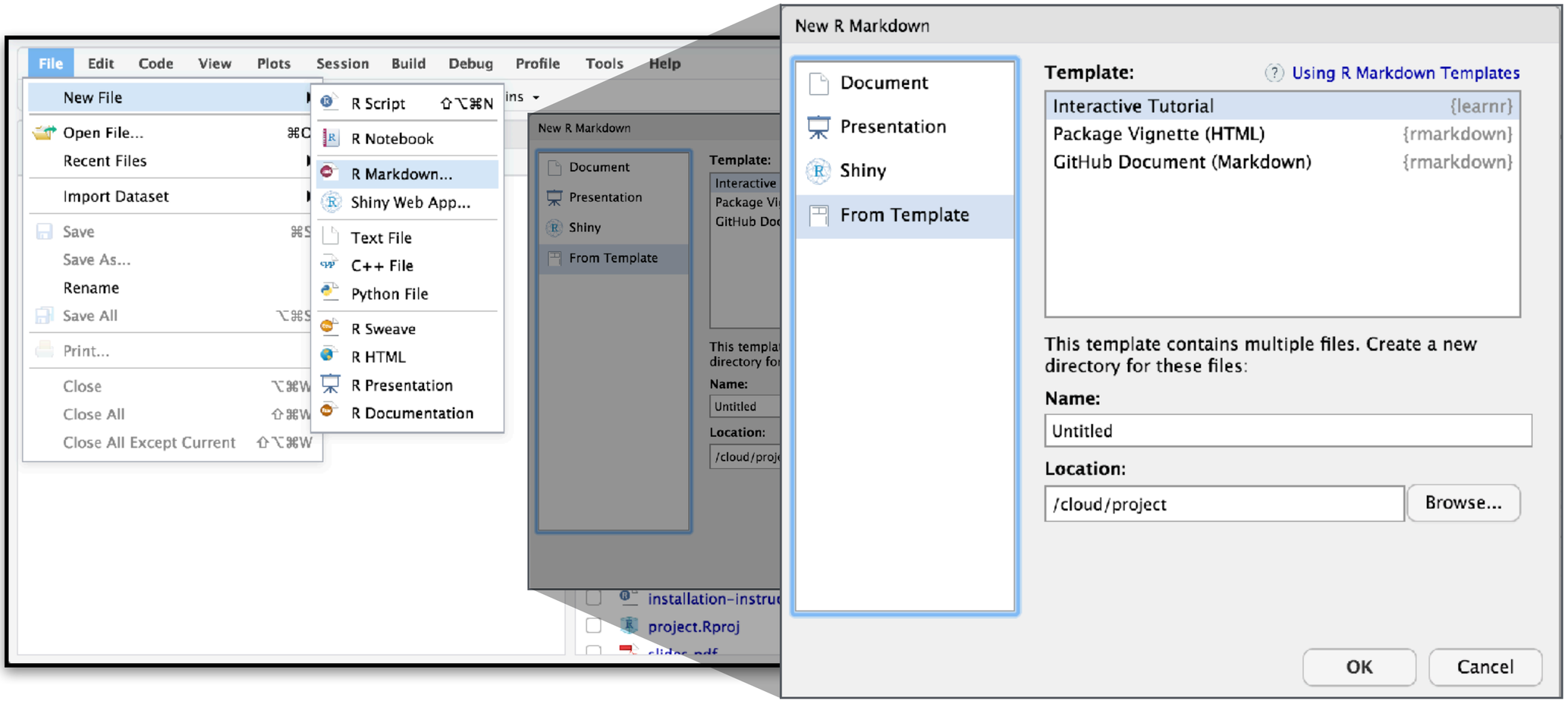
learnr

A package that **extends R Markdown** to make interactive tutorials.

```
install.packages("learnr")
```

learnr

Load a tutorial **template** with the New File > R Markdown wizard



A learnr file is a .Rmd file with a specific **setup**

Untitled.Rmd

```
---
title: "Tutorial"
output: learnr::tutorial
runtime: shiny_prerendered
---

```{r setup, include=FALSE}
library(learnr)
```
```

Set output to learnr:: tutorial

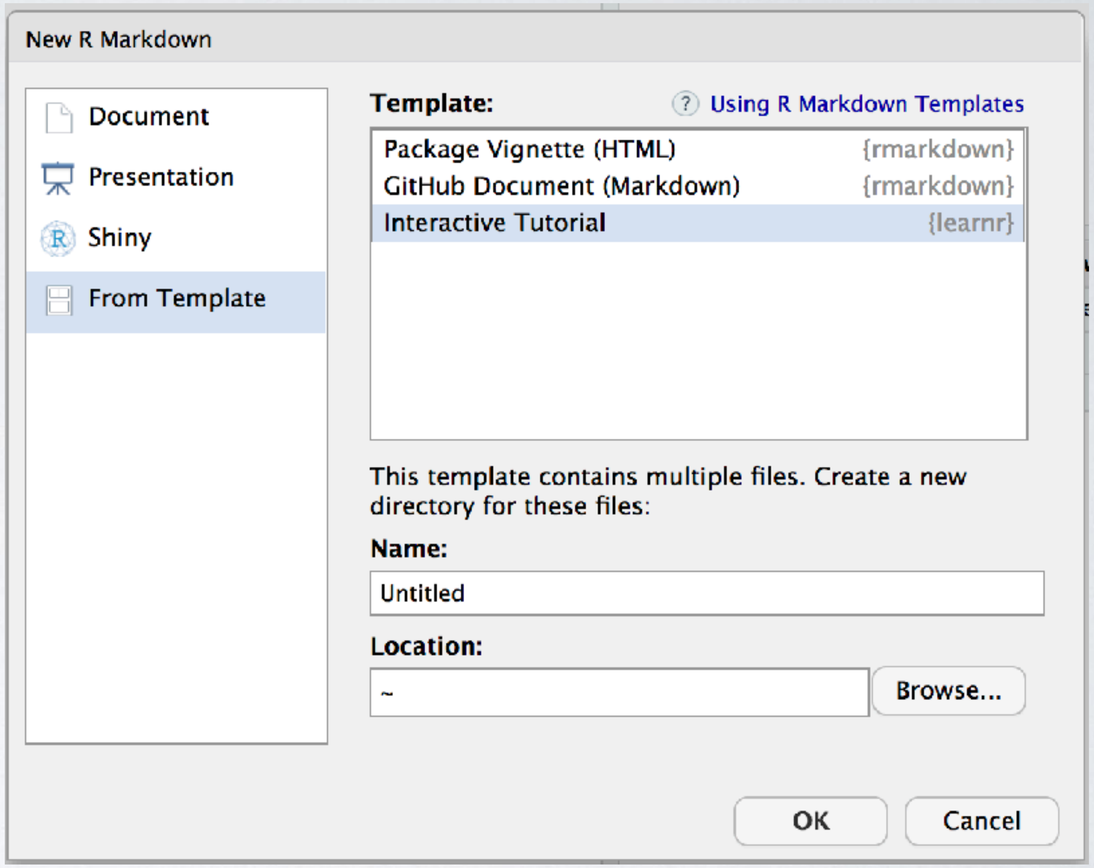
Set runtime to shiny_prerendered

Load the learnr functions

Your Turn

Open a learnr tutorial, erase the body, and then write three paragraphs that summarize:

1. Why R?
2. RStudio Cloud
3. R Markdown



Navigation

Second level headers (##) appear in the **sidebar**

```
## Why R?

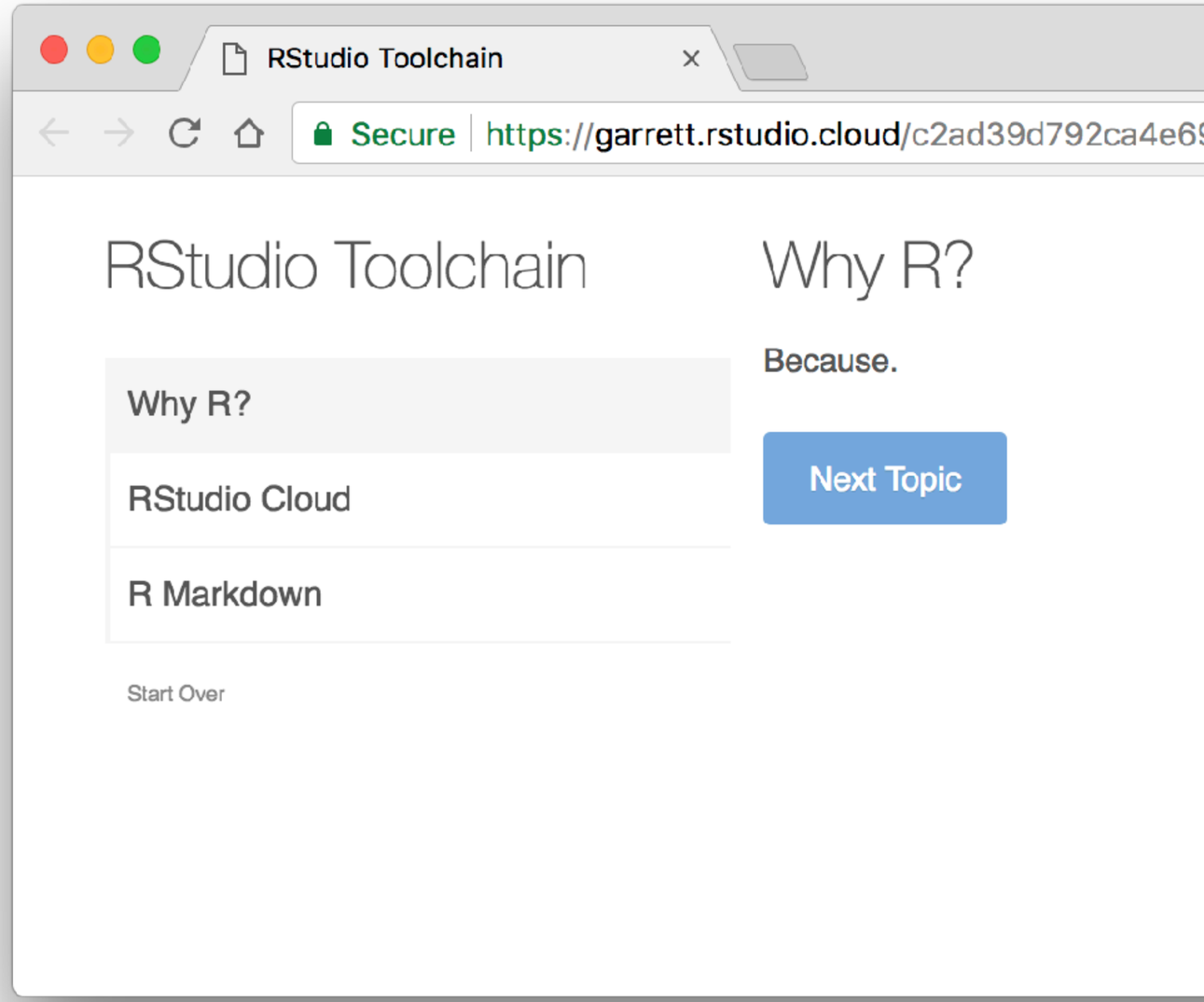
Because.

## RStudio Cloud

Easy.

## R Markdown

Authoring format.
```



Third level headers (###) become **subsections**

```
## Class Notes

### Why R?

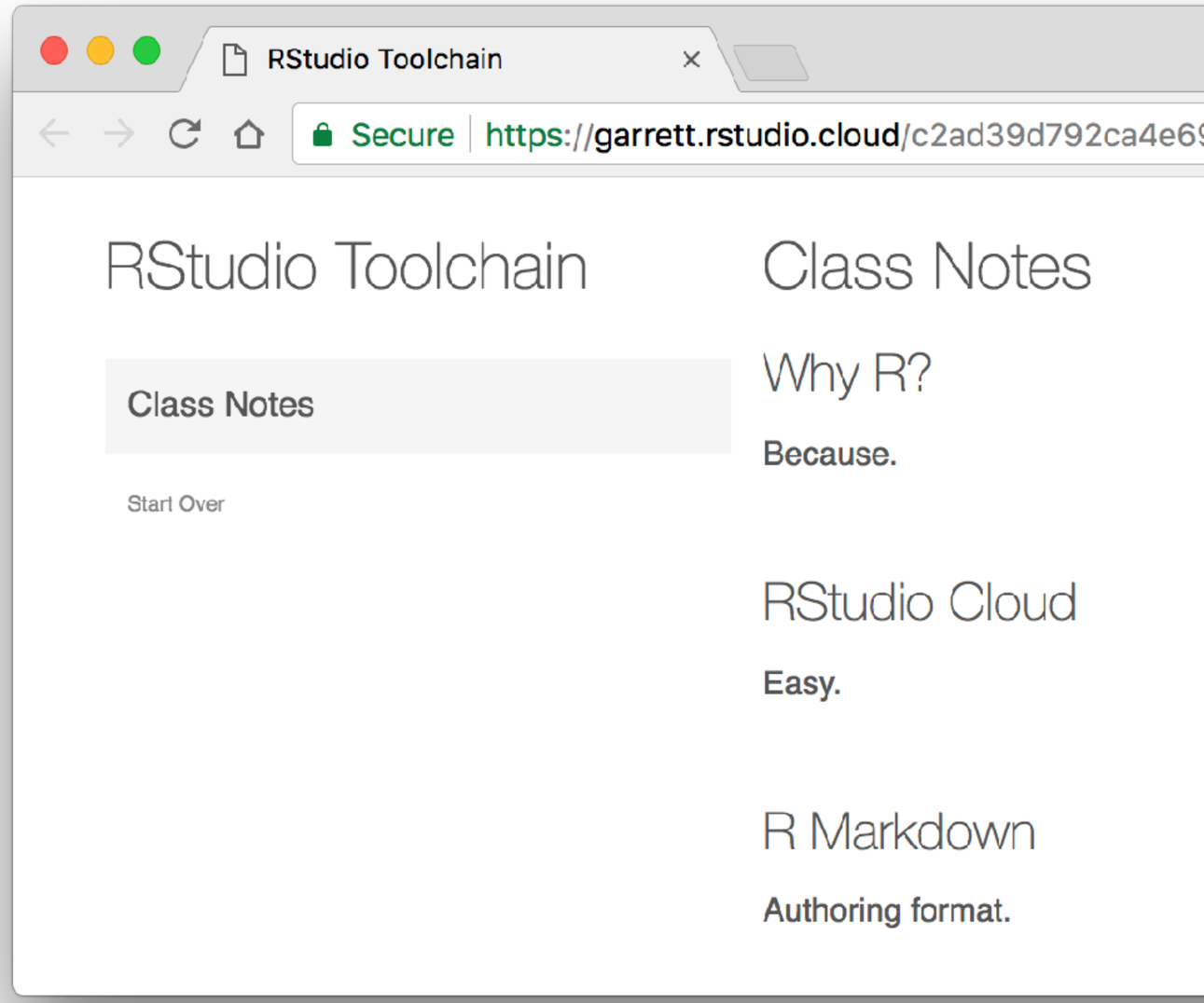
Because.

### RStudio Cloud

Easy.

### R Markdown

Authoring format.
```



Progressive reveal affects third level headers

```
## Class Notes

### Why R?

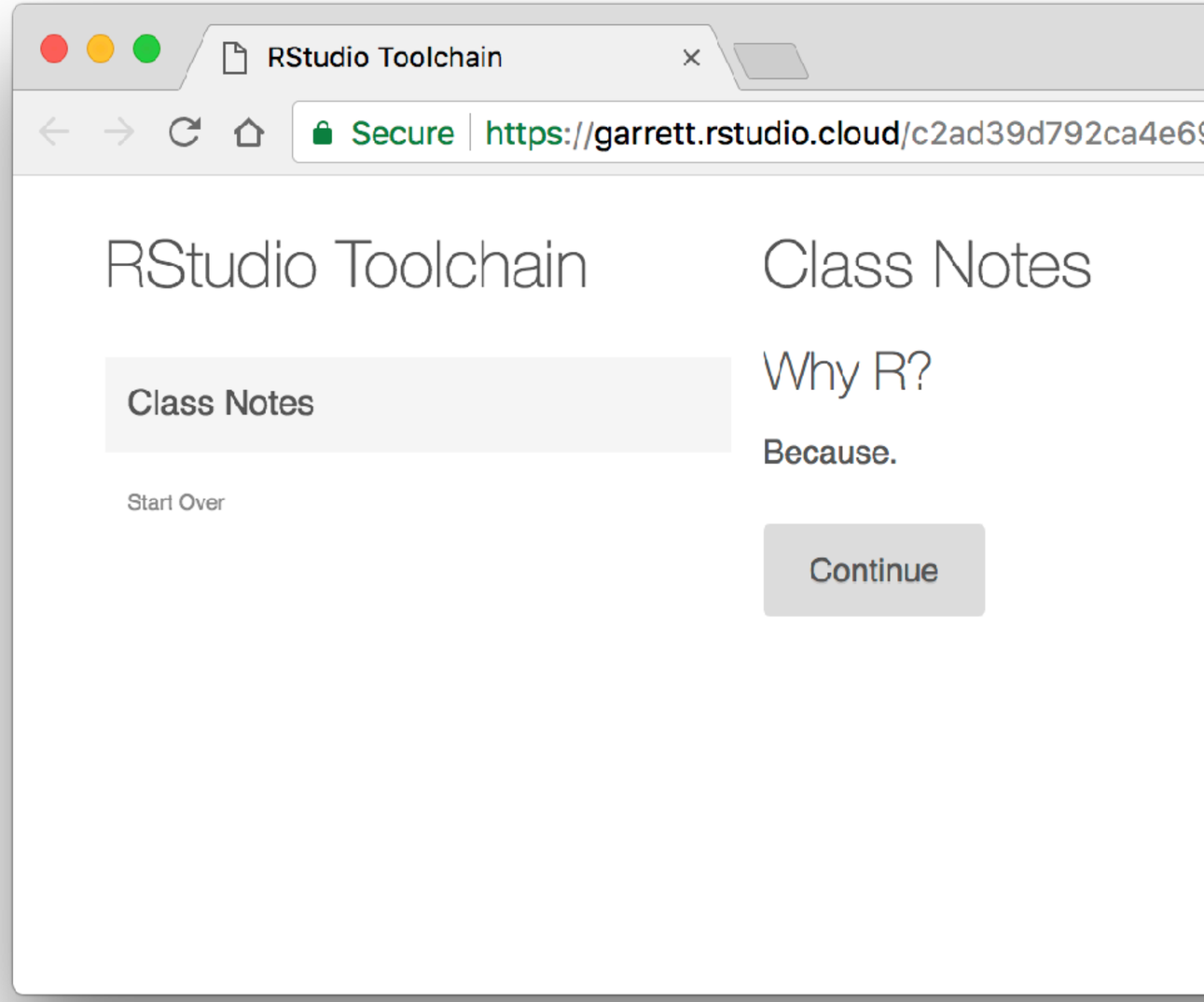
Because.

### RStudio Cloud

Easy.

### R Markdown

Authoring format.
```



progressive reveal

Lets student decide **when to advance** to next subsection

New Line

Indent 2 spaces

Indent 4 spaces

Untitled.Rmd*

```
---
title: "RStudio Toolchain"
output:
  learnr::tutorial:
    progressive: true
    allow_skip: true
runtime: shiny_prerendered
---
```

7:27 RStudio Toolchain R Markdown

Colon at end

New line

Your Turn

Arrange for the paragraphs in your tutorial to appear one at a time with progressive reveal.

Exercises

question() and **answer()**

Construct a **multiple choice question** in a code chunk.

```
```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```
```

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

question()

```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

question()

prompt to display

```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

question()

one answer() per choice

prompt to display

```
```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```
```

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

question()

one answer() per choice

choice to display

prompt to display

```
```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```
```

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

question()

one answer() per choice

choice to display

```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```

prompt to display

correct answer

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

question()

one answer() per choice

choice to display

```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```

prompt to display

correct answer

optional message to display if selected

question() and answer()

Construct a **multiple choice question** in a code chunk.

label

question()

one answer() per choice

choice to display

```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```

prompt to display

correct answer

optional message to display if selected

Your Turn

Add a multiple choice question to your Why R? section.

label

question()

one answer() per choice

choice to display

```{r mc-1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "sorry")
)
```

prompt to display

correct answer

optional message to display if selected

05:00

Code exercises

Turn a chunk into a code exercise with **exercise = TRUE**

```
```{r ex-1, exercise = TRUE}

```
```

Code exercises

Turn a chunk into a code exercise with **exercise = TRUE**

label

```{r ex-1, exercise = TRUE}

```

Code exercises

Turn a chunk into a code exercise with **exercise = TRUE**

label

```
```{r ex-1, exercise = TRUE}

```
```

exercise = TRUE

Code exercises

Turn a chunk into a code exercise with **exercise = TRUE**

label

An exercise with an empty field

```{r ex-1, exercise = TRUE}

```

exercise = TRUE

Code exercises

Turn a chunk into a code exercise with **exercise = TRUE**

label

Code to appear in exercise field

exercise = TRUE

```
```{r ex-1, exercise = TRUE}
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy))
```
```

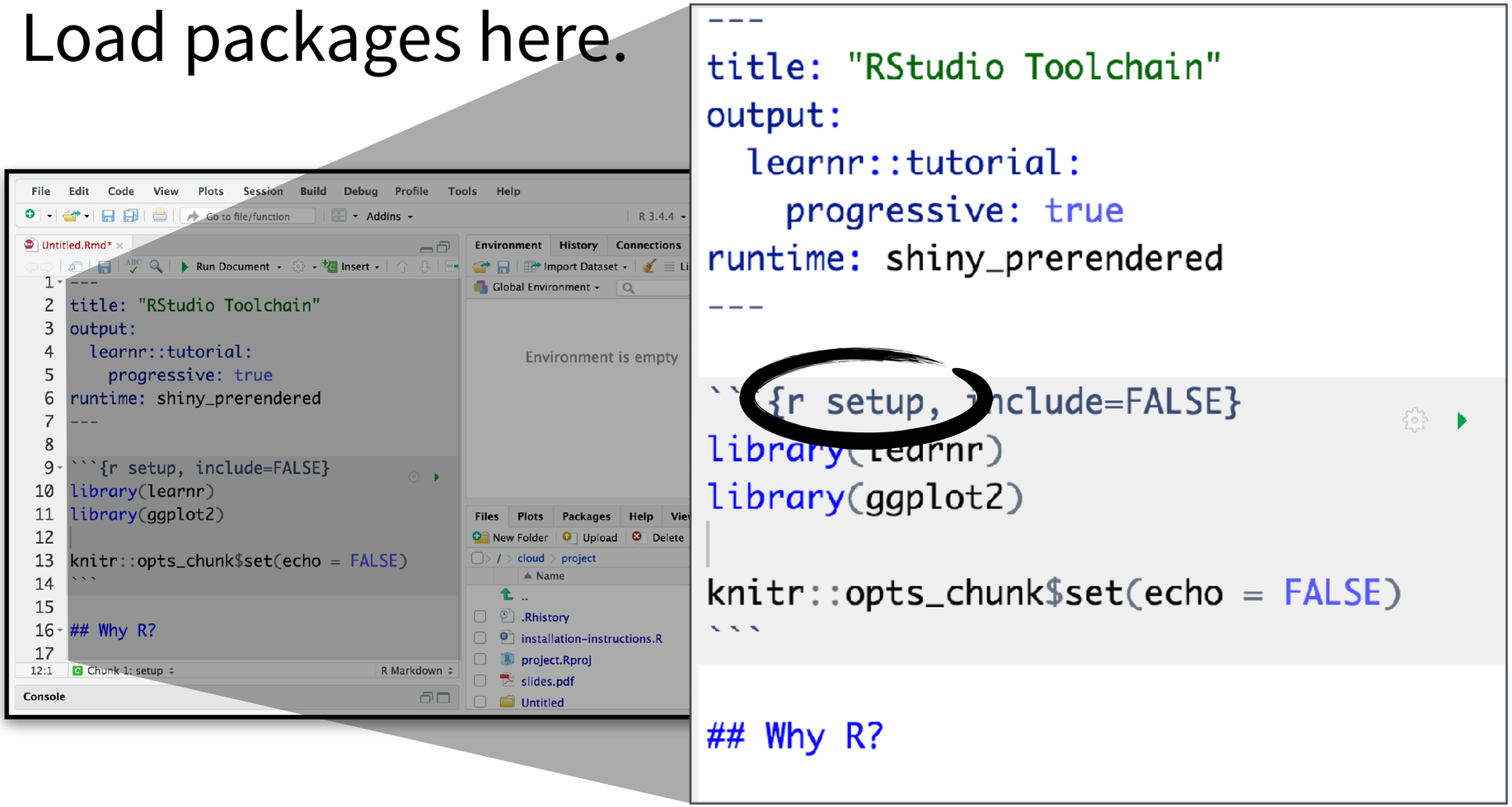
Quiz

What if you want students to be able to use functions from a package in the exercise?

the **global setup** chunk

R Markdown runs the chunk labelled **setup** first.

Load packages here.



Solutions

Supply a solution in a separate code chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

```
```{r ex-1-solution}
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy))
```
```

Solutions

Supply a solution in a separate code chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

solution code

```
```{r ex-1-solution}
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy))
```
```

Solutions

Supply a solution in a separate code chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

solution code

```
```{r ex-1-solution}
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy))
```
```

The label MUST use this syntax

Solutions

Supply a solution in a separate code chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

solution code

-solution

```
```{r ex-1-solution}
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy))
```
```

Hints

Supply hints in chunks suffixed with **-hint** (or -hint-1, -hint-2, …)

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

```
```{r ex-1-hint}
"Use geom_point()"
```
```

Hints

Supply hints in chunks suffixed with **-hint** (or -hint-1, -hint-2, …)

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

```
```{r ex-1-hint}
"Use geom_point()"
```
```

-hint

Your Turn

Add a code exercise with a solution to your Cloud section.

label

exercise = TRUE

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

-solution

solution code

```
```{r ex-1-solution}
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy))
```
```

Security

Every exercise is run in a clean environment

Quiz

Can a student call results from **the setup chunk** in an exercise?

Quiz

Can a student call results from **the setup chunk** in an exercise?

Yes. Setup results are available to every computation in the document.

Quiz

Can a student call results from **any other chunk** in an exercise?

Quiz

Can a student call results from **any other chunk** in an exercise?

No. With one exception: you can make an exercise specific setup chunk.

Exercise Setup

(Re)build objects to use in a **-setup** chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}
mpg2
```
```

```
```{r ex-1-setup}
mpg2 <- mpg %>% filter(cyl == 4)
```
```

Exercise Setup

(Re)build objects to use in a **-setup** chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}
mpg2
```
```

setup code

```
```{r ex-1-setup}
mpg2 <- mpg %>% filter(cyl == 4)
```
```

Exercise Setup

(Re)build objects to use in a **-setup** chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}
mpg2
```
```

name of exercise

setup code

```
```{r ex-1-setup}
mpg2 <- mpg %>% filter(cyl == 4)
```
```

Exercise Setup

(Re)build objects to use in a **-setup** chunk

The exercise chunk

```
```{r ex-1, exercise = TRUE}
mpg2
```
```

name of exercise

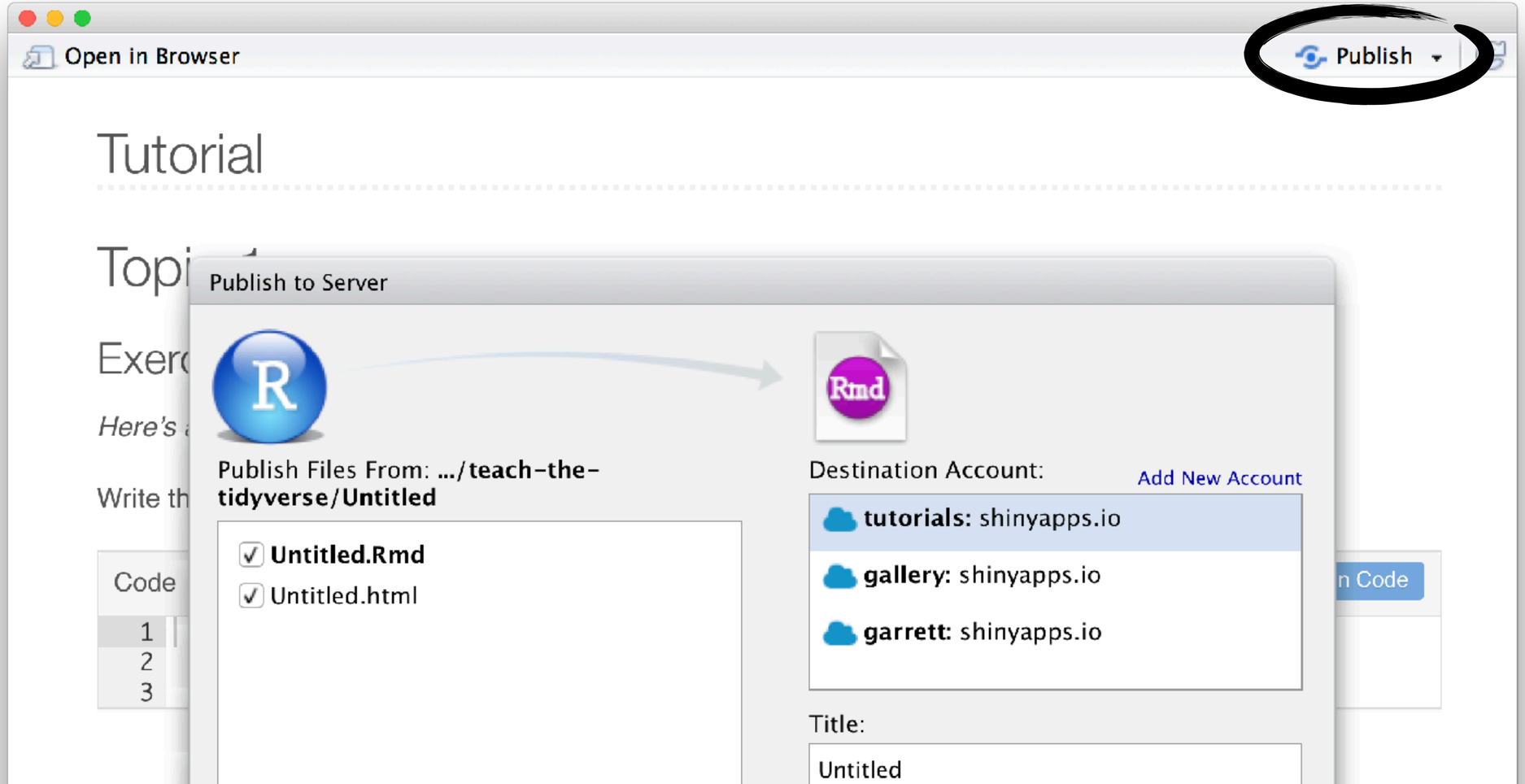
-setup

setup code

```
```{r ex-1-setup}
mpg2 <- mpg %>% filter(cyl == 4)
```
```

Publishing

Tutorials are shiny apps. Host at http://**shinyapps.io**.



rstudio.github.io/learnr/

Interactive Tutorials for R

Garrett

Secure | https://rstudio.github.io/learnr/

learnr | Home | Exercises | Questions | Publishing | Formats | Examples

- Overview
- Getting Started
- Tutorial Types
- Exercises
- Questions
- Videos
- Shiny Components
- External Resources
- Preserving Work
- Publishing

Interactive Tutorials for R

Overview

The **learnr** package makes it easy to turn any R Markdown document into an interactive tutorial. Tutorials consist of content along with interactive components for checking and reinforcing understanding. Tutorials can include any or all of the following:

1. Narrative, figures, illustrations, and equations.
2. Code exercises (R code chunks that users can edit and execute directly).
3. Quiz questions.
4. Videos (supported services include YouTube and Vimeo).
5. Interactive Shiny components.

Tutorials automatically preserve work done within them, so if a user works on a few exercises or questions and returns to the tutorial later they can pick up right where they left off.

Examples

Here are some simple examples of tutorials created with the **learnr** package:

grader

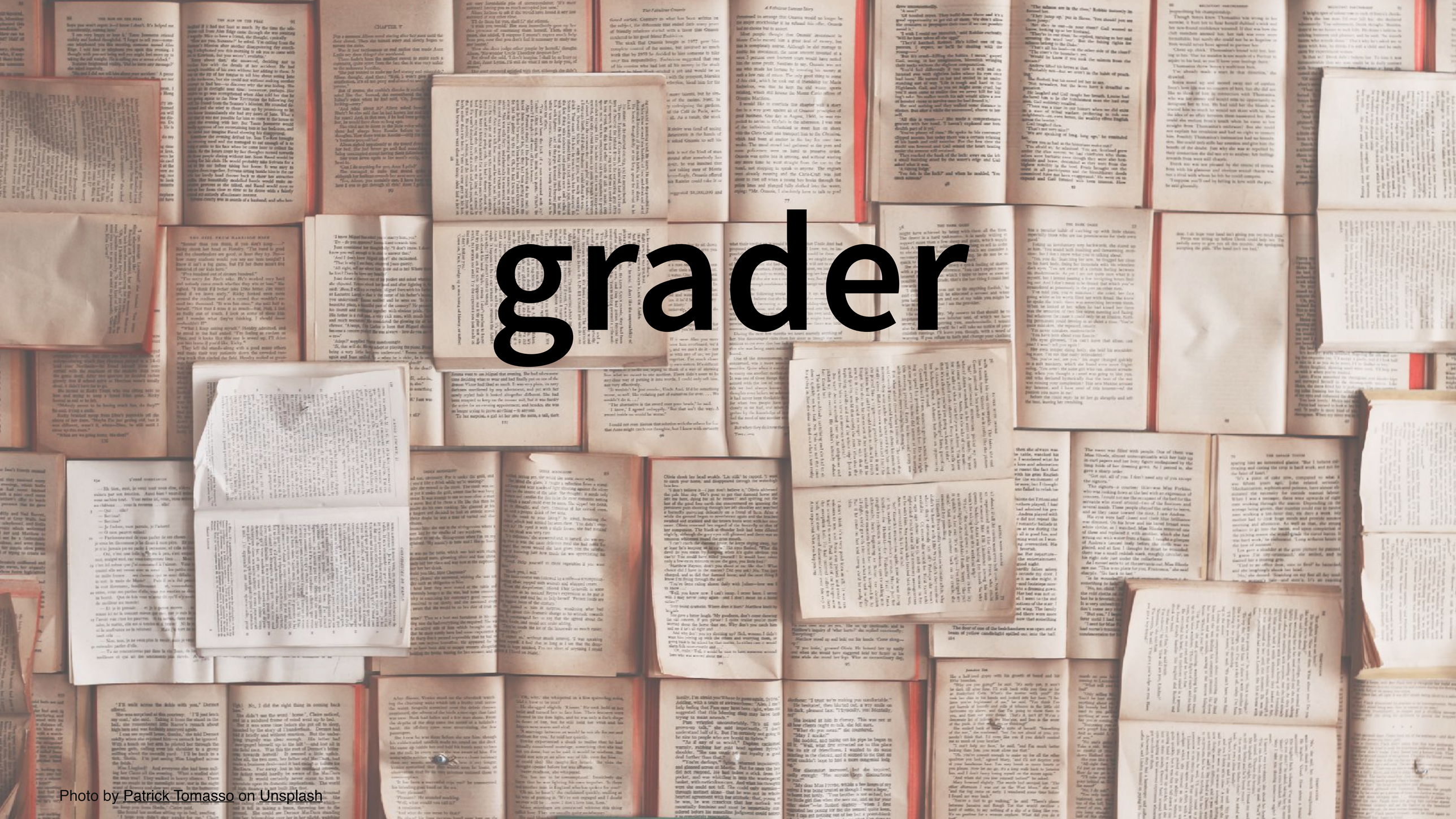


Photo by Patrick Tomasso on Unsplash

grader

Prototype extensions for authoring with learnr
https://**github.com/rstudio-education/grader**

```
install.packages("devtools")
devtools::install_github("rstudio-education/grader")
```

test_solutions()

Tests all solutions and setup chunks to ensure they run

```
test_solutions("untitled/untitled.Rmd")
```

```
setup: ✔
success-1-solution: ✔
warning-2-solution: ✔ NaNs produced
failure-3-solution: ✖ argument "x" is missing
success-4-solution: ✔
```

test_solutions()

Tests all solutions and setup chunks to ensure they run

```
test_solutions("untitled/untitled.Rmd")
```

Filepath to .Rmd

```
setup: ✔
success-1-solution: ✔
warning-2-solution: ✔ NaNs produced
failure-3-solution: ✖ argument "x" is missing
success-4-solution: ✔
```

Your Turn

Run test_solutions **at the command line** to test your solution(s)

```
test_solutions("untitled/untitled.Rmd")
```

Filepath to .Rmd

strict_check()

A workhorse **checking function** for learnr tutorials.

Compares student answer to solution and returns a message wherever it finds a mismatch.

Checking

Checking

1. Use **tutorial_options** to Set the **exercise.checker** tutorial option to **grade_learnr** in the **setup** chunk.

```
```{r setup}
tutorial_options(exercise.checker = grade_learnr)
```
```

Checking

1. Use **tutorial_options** to Set the **exercise.checker** tutorial option to **grade_learnr** in the **setup** chunk.

setup

```
```{r setup}
tutorial_options(exercise.checker = grade_learnr)
```
```

Checking

1. Use **tutorial_options** to Set the **exercise.checker** tutorial option to **grade_learnr** in the **setup** chunk.

setup

```{r setup}
tutorial_options(exercise.checker = grade_learnr)
```

tutorial_options()

Checking

1. Use **tutorial_options** to Set the **exercise.checker** tutorial option to **grade_learnr** in the setup chunk.

setup

```
```{r setup}
tutorial_options(exercise.checker = grade_learnr)
```
```

tutorial_options()

exercise.checker

Checking

1. Use **tutorial_options** to Set the **exercise.checker** tutorial option to **grade_learnr** in the **setup** chunk.

setup

```
```{r setup}
tutorial_options(exercise.checker = grade_learnr)
```
```

tutorial_options()

exercise.checker

grade_learnr

Checking

1. Use **tutorial_options** to Set the **exercise.checker** tutorial option to **grade_learnr** in the **setup** chunk.

setup

```
```{r setup}
tutorial_options(exercise.checker = grade_learnr)
```
```

tutorial_options()

exercise.checker

grade_learnr

Checking

2. Add a **-check** chunk for the exercise that calls **strict_check()**

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

```
```{r ex-1-check}
strict_check(success = "Good Job! Now
  let's graph this data.")
```
```

Checking

2. Add a **-check** chunk for the exercise that calls **strict_check()**

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

```
```{r ex-1-check}
strict_check(success = "Good Job! Now
  let's graph this data.")
```
```

Checking

2. Add a **-check** chunk for the exercise that calls **strict_check()**

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

-check

```
```{r ex-1-check}
strict_check(success = "Good Job! Now
  let's graph this data.")
```
```

Checking

2. Add a **-check** chunk for the exercise that calls **strict_check()**

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

strict_check()

-check

```
```{r ex-1-check}
strict_check(success = "Good Job! Now
  let's graph this data.")
```
```

Checking

2. Add a **-check** chunk for the exercise that calls **strict_check()**

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

strict_check()

-check

optional success message to display

```
```{r ex-1-check}
strict_check(success = "Good Job! Now
  let's graph this data.")
```
```

Checking

The exercise must have a solution chunk

2. Add a **-check** chunk for the exercise that calls **strict_check()**

The exercise chunk

```
```{r ex-1, exercise = TRUE}


```
```

name of exercise

-check

strict_check()

optional success message to display

```
```{r ex-1-check}
strict_check(success = "Good Job! Now
  let's graph this data.")
```
```

Your Turn

Add checking to your code exercise. Then run the tutorial and try typing a wrong answer.

1. Use **tutorial_options** to Set the **exercise.checker** tutorial option to **grade_learnr** in the **setup** chunk.

2. Add a **-check** chunk for the exercise that calls **strict_check()**

grader

1. Test solutions
2. Interactive checking
3. Logging (Coming soon)
4. More exercise types (Coming soon)
5. IDE Integration (Coming soon)

Photo by Patrick Tomasso on Unsplash

The bigger picture

Photo by Edwin Andrade on Unsplash

Quiz

Are your students attempting to become **Data Scientists** or **Computer Scientists**?

Think-Pair-Share

With a partner, generate two lists:

What R topics do **Data Scientists** want to know?

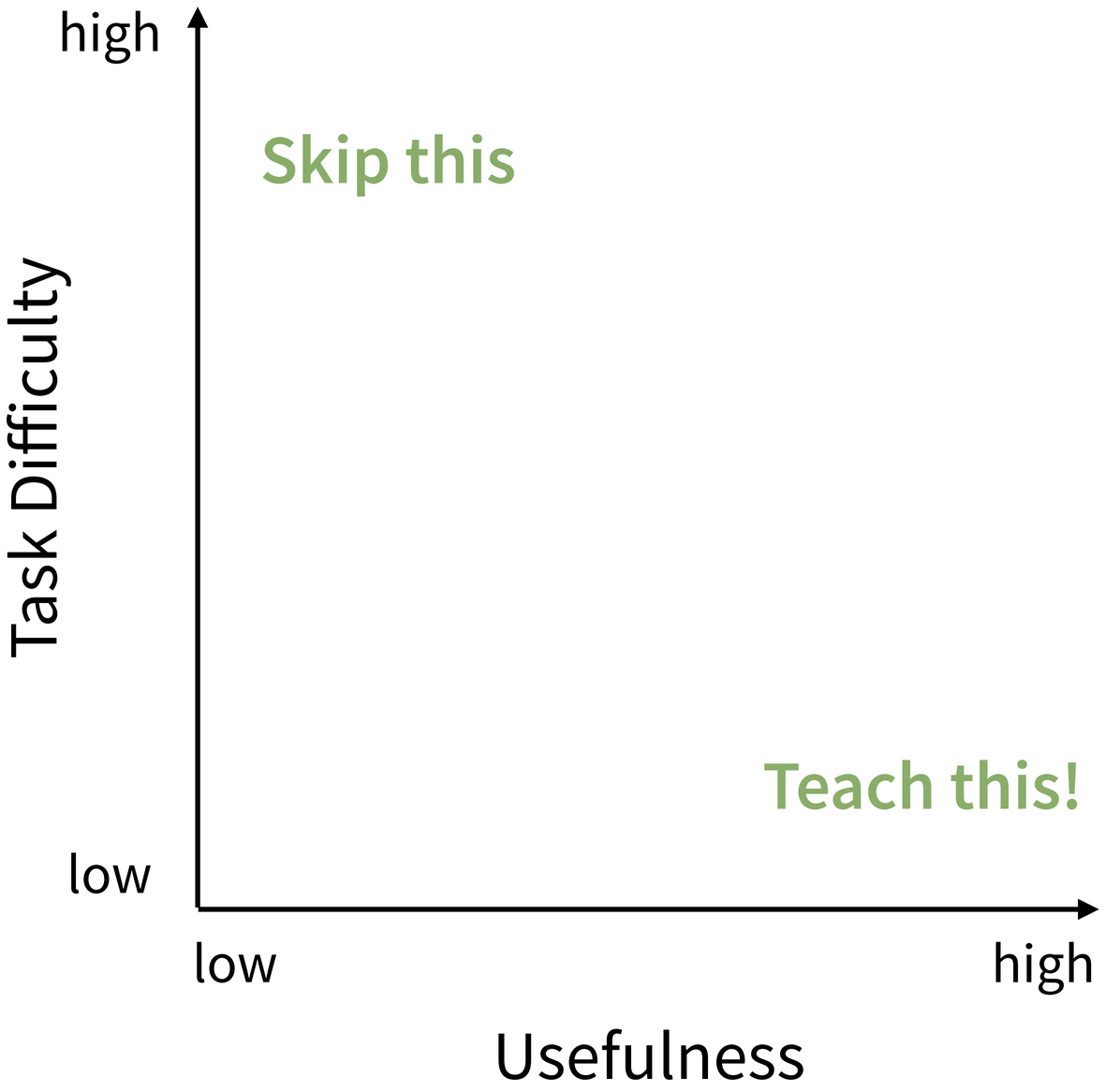
What R topics do **Computer Scientists** want to know?

1. Basics of RStudio
2. Data Structures
3. Subsetting
4. For Loops
5. Writing Functions
6. Writing Data
7. Data Manipulation
8. Writing Reports

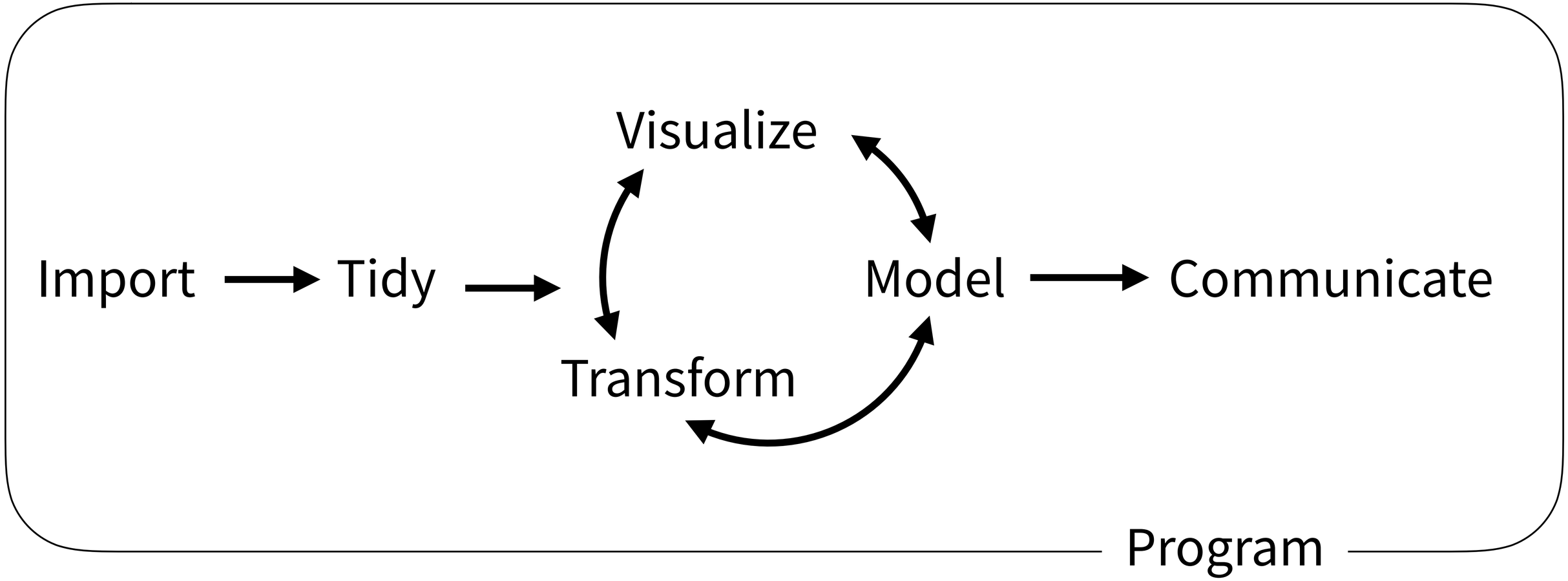
Data Science or Computer Science?

1. Visualize Data
2. Transform Data
3. Tidy Data
4. Import Data
5. Types of Data
6. Iteration
7. Models
8. List Columns

Data Science or Computer Science?


high
Skip this
Task Difficulty
Teach this!
low
low
high
Usefulness

(Applied) **Data Science**



Quiz

Confer with your group.

What relationship do you expect to see between engine size (displ) and mileage (hwy)?

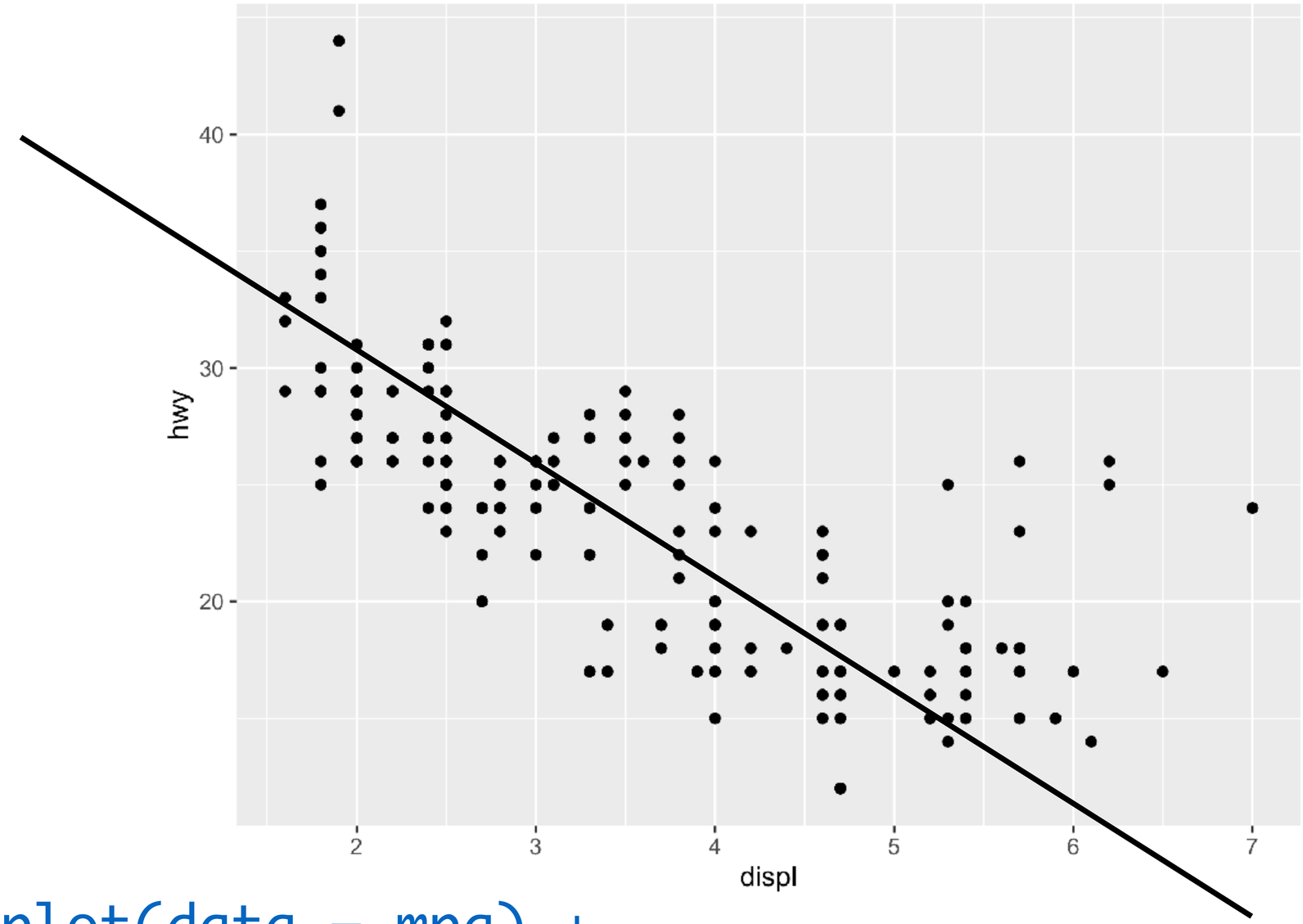
No peeking ahead!

Your Turn 1

Run this code in your notebook to make a graph.

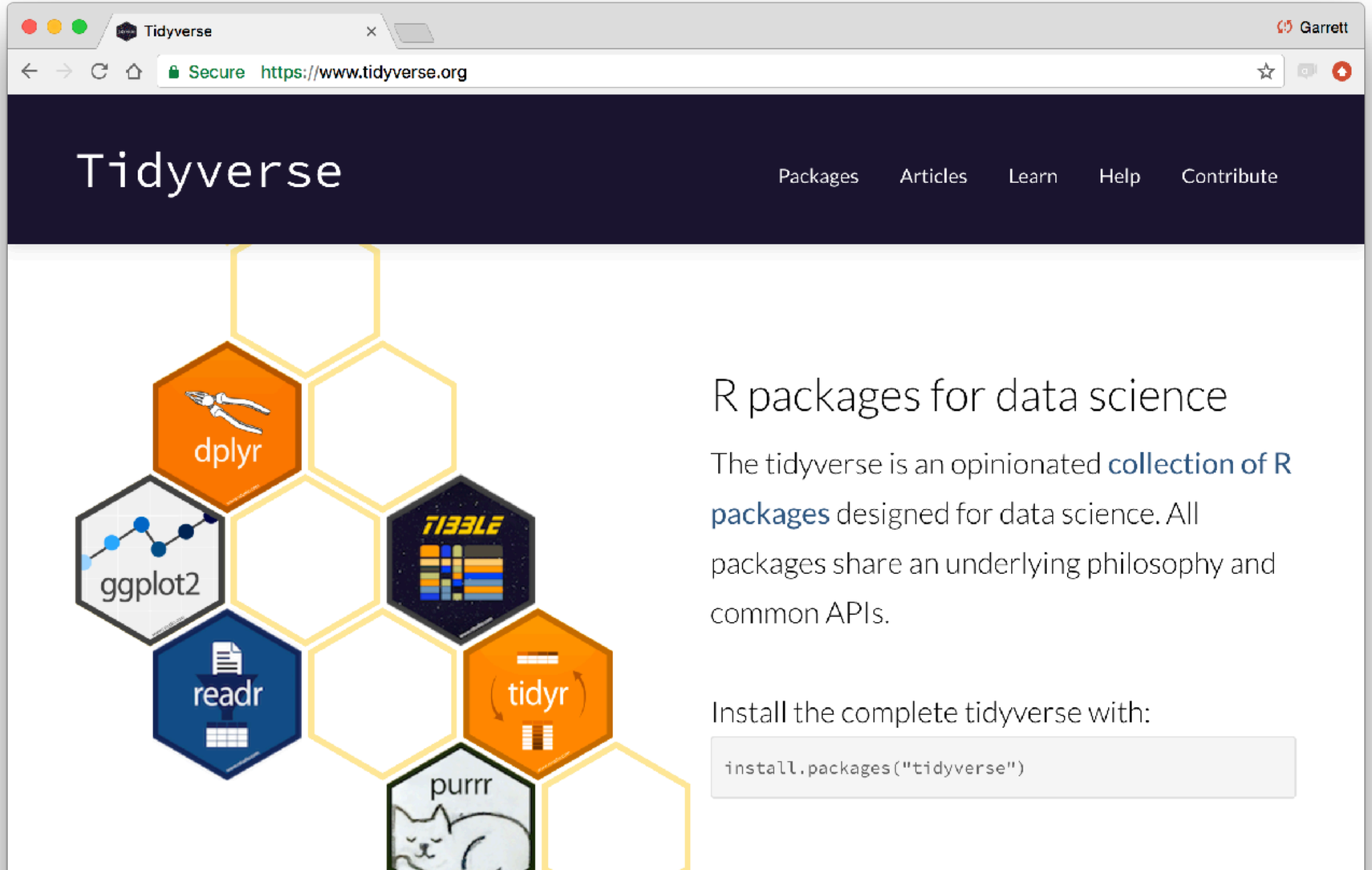
Pay strict attention to spelling, capitalization, and parentheses!

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

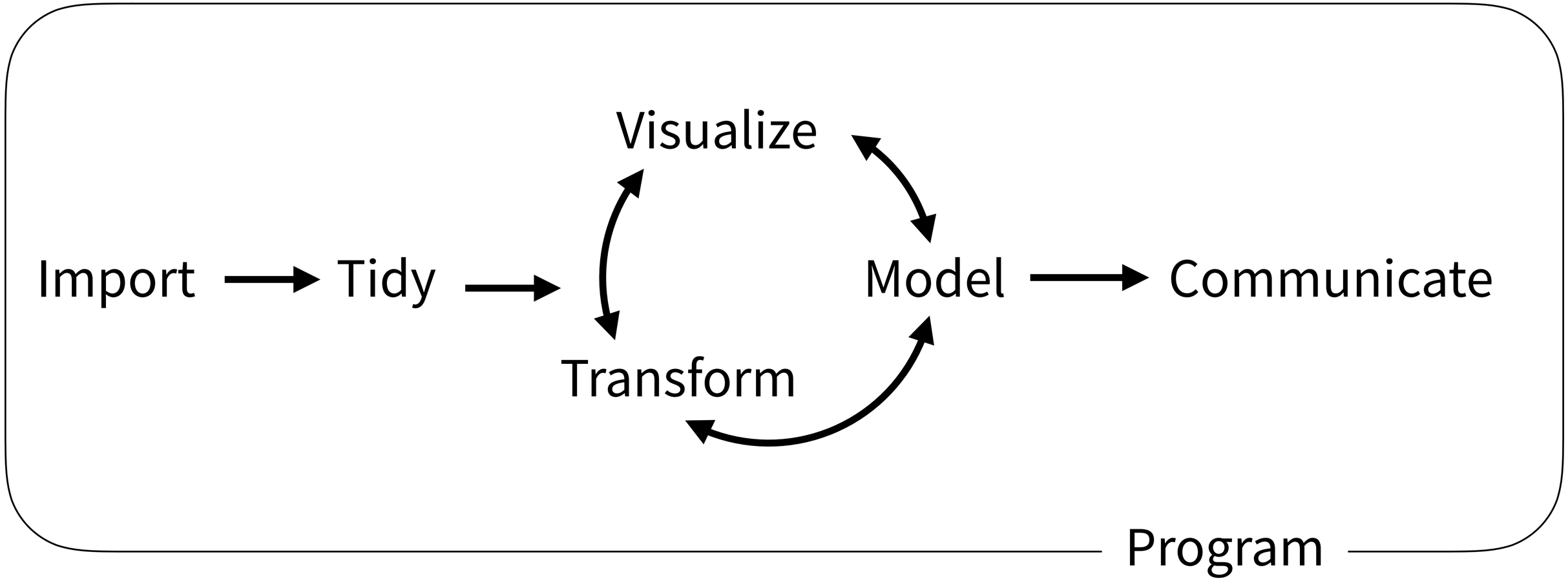



```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

tidyverse.org



(Applied) **Data Science**



www.rstudio.com/resources/cheatsheets

Cheatsheets – RStudio

Garrett

Secure | https://www.rstudio.com/resources/cheatsheets/

RStudio

rstudio::conf Products Resources Pricing About Us Blogs

RStudio Cheat Sheets

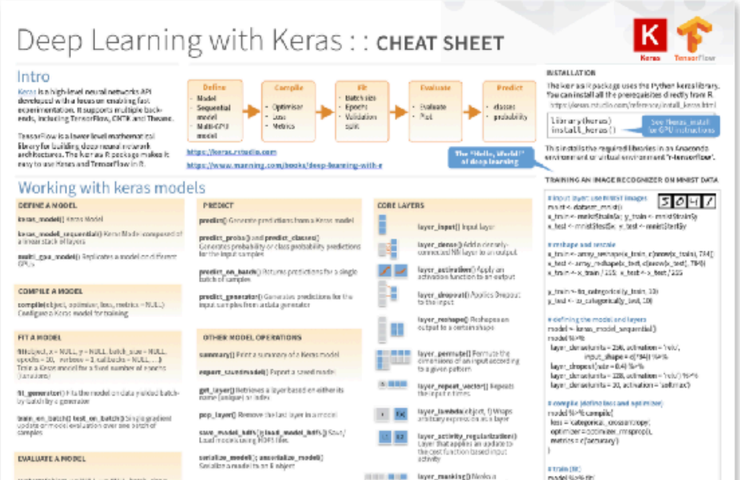
The cheat sheets below make it easy to learn about and use some of our favorite packages. From time to time, we will add new cheat sheets to the gallery. If you'd like us to drop you an email when we do, let us know by clicking the button to the right.

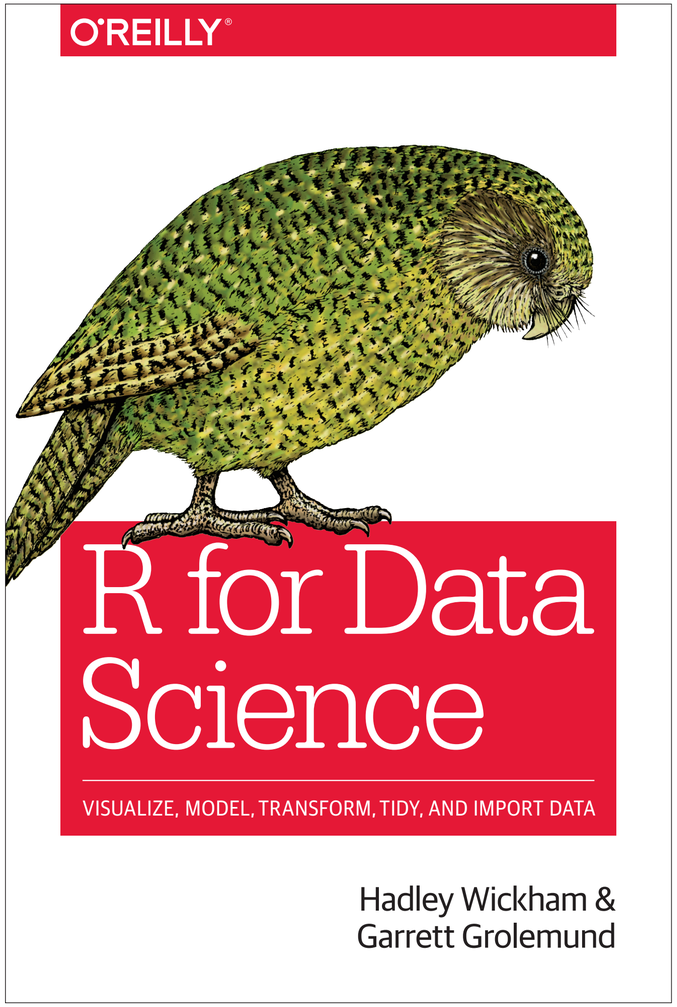
SUBSCRIBE TO CHEAT SHEET UPDATES HERE

Deep Learning with Keras Cheat Sheet

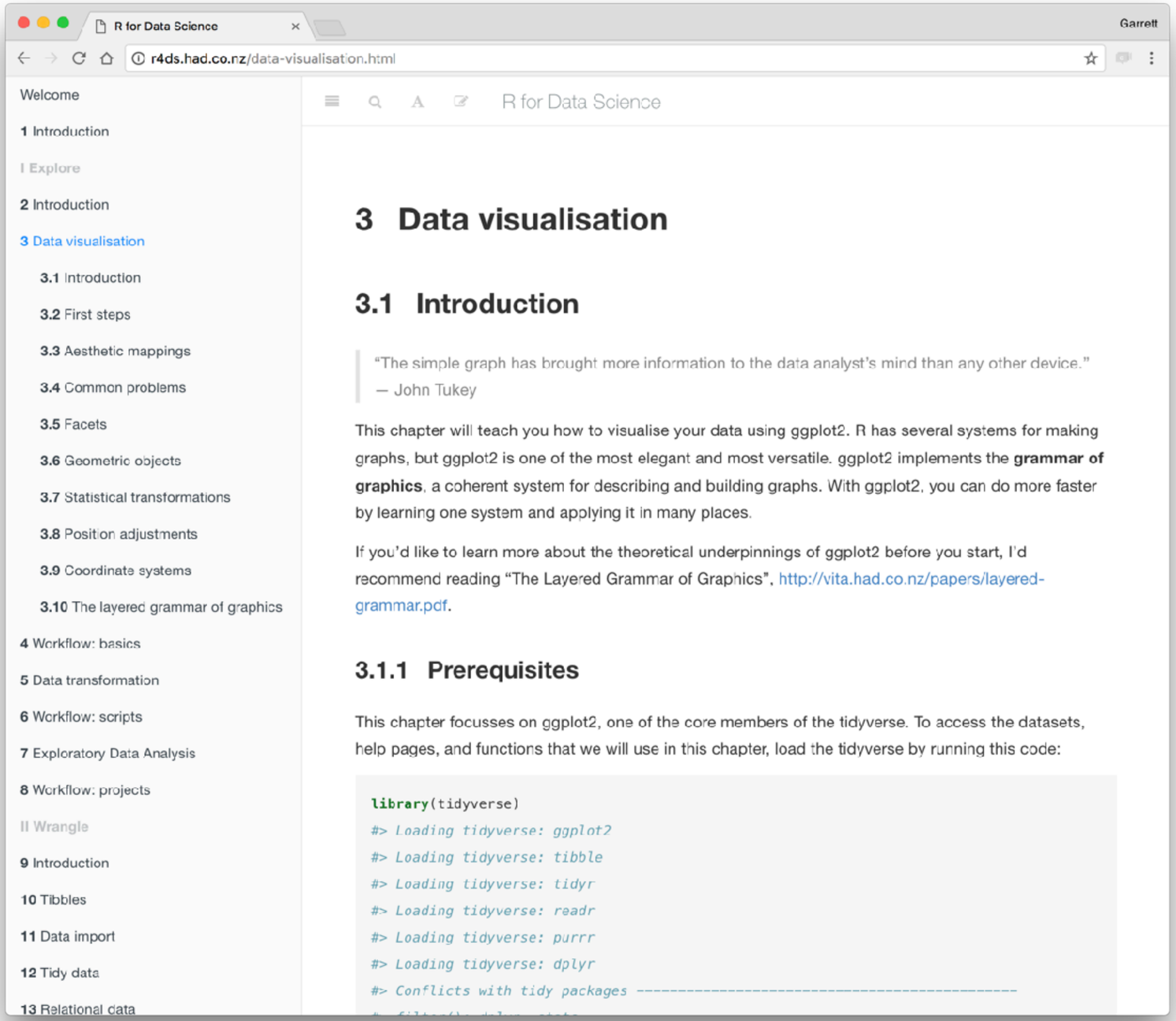
Keras is a high-level neural networks API developed with a focus on enabling fast experimentation. Keras supports both convolution based networks and recurrent networks (as well as combinations of the two), runs seamlessly on both CPU and GPU devices, and is capable of running on top of multiple back-ends including TensorFlow, CNTK, and Theano. Updated 12/17.

DOWNLOAD





http://r4ds.had.co.nz



RStudio Cloud

Garrett

Secure https://rstudio.cloud/learn/primers

Your Workspace | Projects | Learn

Garrett Grolemund

Guide | Primers | DataCamp Courses | Cheat Sheets

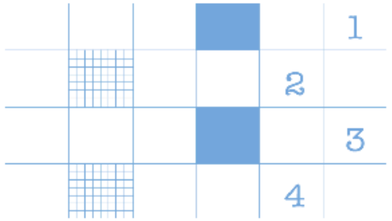
RStudio Primers

Learn data science basics with the interactive tutorials below.

The Basics

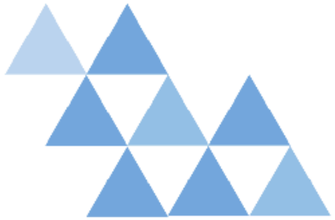
Start here to learn the skills that you will rely on in every analysis (and every primer that follows): how to inspect, visualize, subset, and transform your data, as well as how to run code.

Work with Data



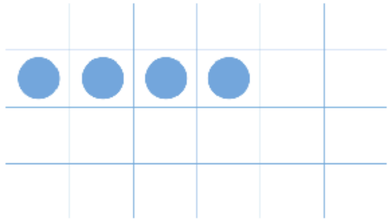
Learn the most important data handling skills in R: how to extract values from a table, subset tables, calculate summary statistics, and derive new variables.

Visualize Data



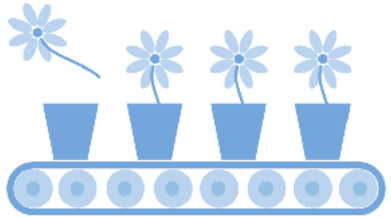
Learn how to use ggplot2 to make any type of plot with your data. Then learn the best ways to visualize patterns within values and relationships between variables.

Tidy Your Data



Unlock the tidyverse by learning how to make and use tidy data, the data format designed for R.

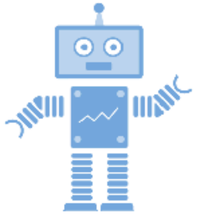
Iterate



Master a core programming paradigm with the purrr package: for each ____ do ____.

Automate Tasks

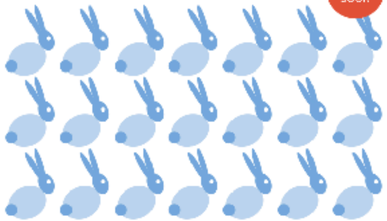
COMING SOON



Functions are the key to programming in R. This primer will teach you how to write and use them.

Report Reproducibly

COMING SOON



Learn to report, reproduce, and parameterize your work with the best authoring format for Data Science: R Markdown.

Build Interactive Web Apps

COMING SOON

Say hello to Shiny, R's package for building interactive web apps. Learn to turn your analyses into elegant tools to share with others

Github

Photo by Owen Massey McKnight on flickr

Github

rundel.github.io/ghclass/articles/ghclass.html

Photo by Owen Massey McKnight on flickr

Photo by SpaceX on Unsplash

Thank You

Please take the class survey

www.surveymonkey.com/r/2MFN9T8