

Reproducible Research using



Go here and log in (free):

<https://rstudio.cloud/project/233945>

<https://rstudio.cloud/project/233945>



@StatGarrett

<https://rstudio.cloud/project/233945>

Pop Quiz

What does **IMRAD** refer to? Poll your neighbors.

Standard structure
for a scientific paper.



<https://rstudio.cloud/project/233945>

What does **IMRAD** stand for? Poll your neighbors.

Introduction

What hypothesis was tested and why?

Methods

How was the study done?

Results

What answer was discovered?

And Discussion

What does the answer imply?



<https://rstudio.cloud/project/233945>

Which words do you associate
with **math**?

hypotheses

messy

best guess

discover

axioms

logical

certain

prove

<https://rstudio.cloud/project/233945>

Which words do you associate
with **Science**?

hypotheses

messy

best guess

discover

axioms

logical

certain

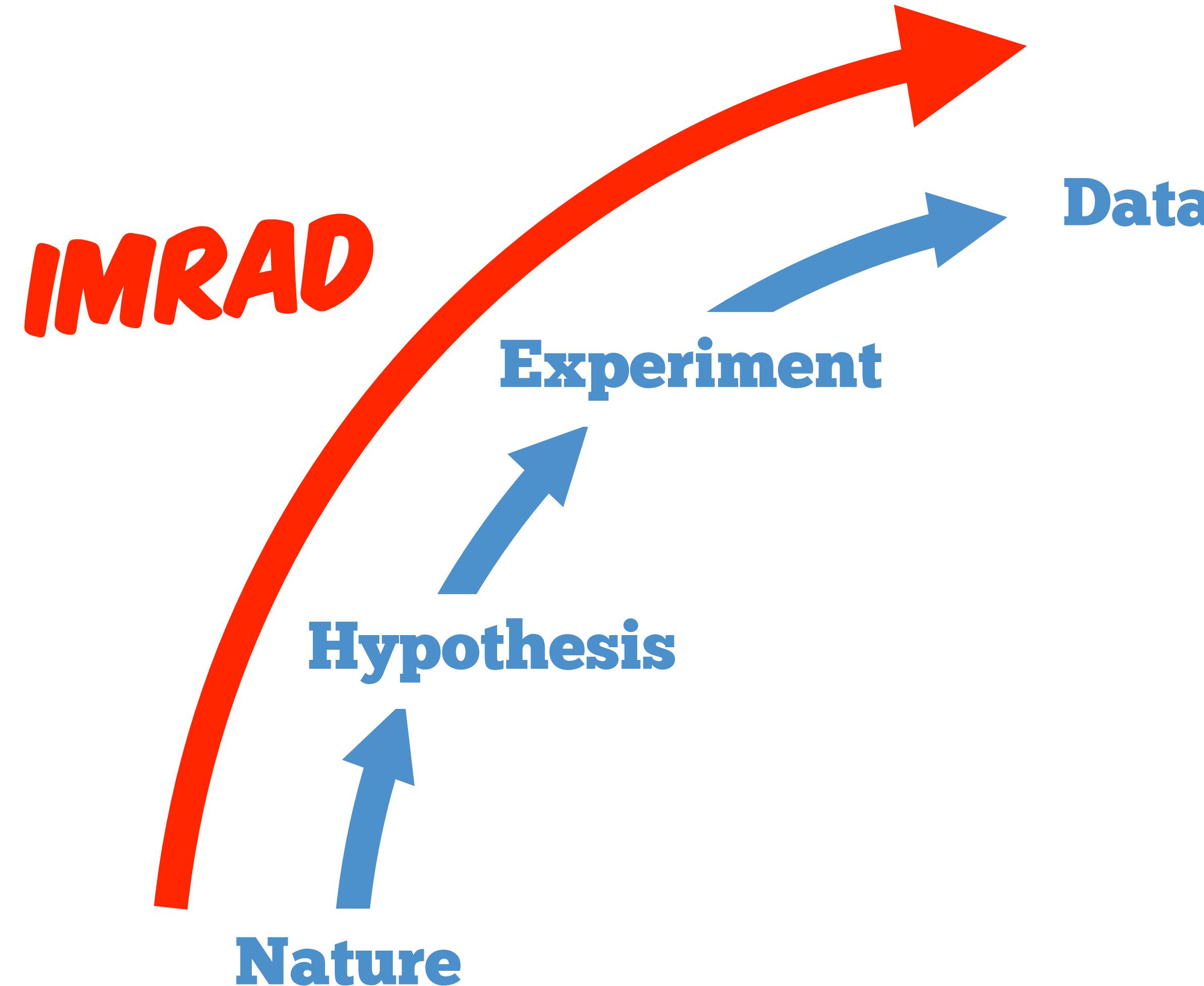
prove



**CREATE MAPS.
NOT PROOFS**

IMRAD

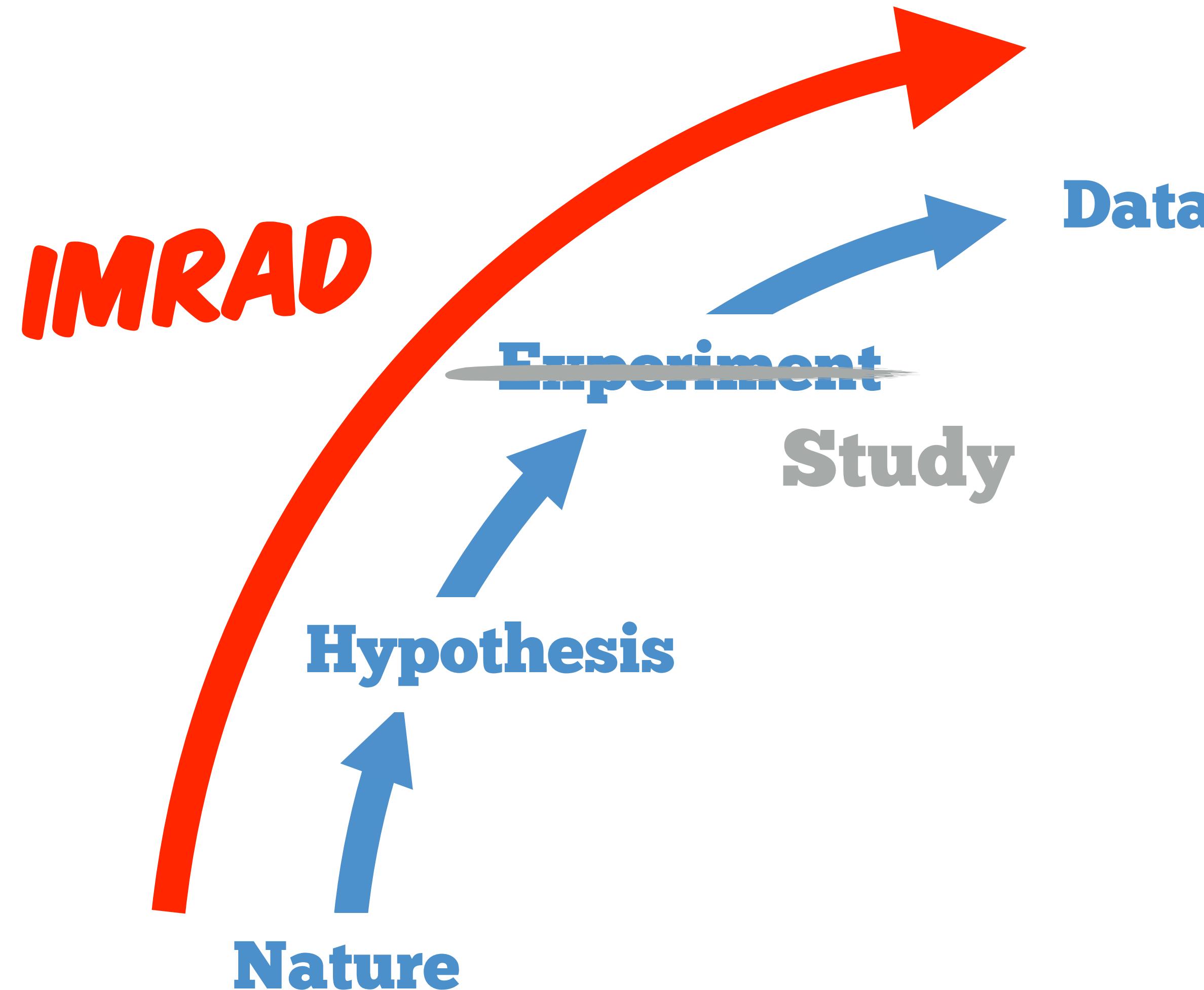
<https://rstudio.cloud/project/233945>



But where's
the math?

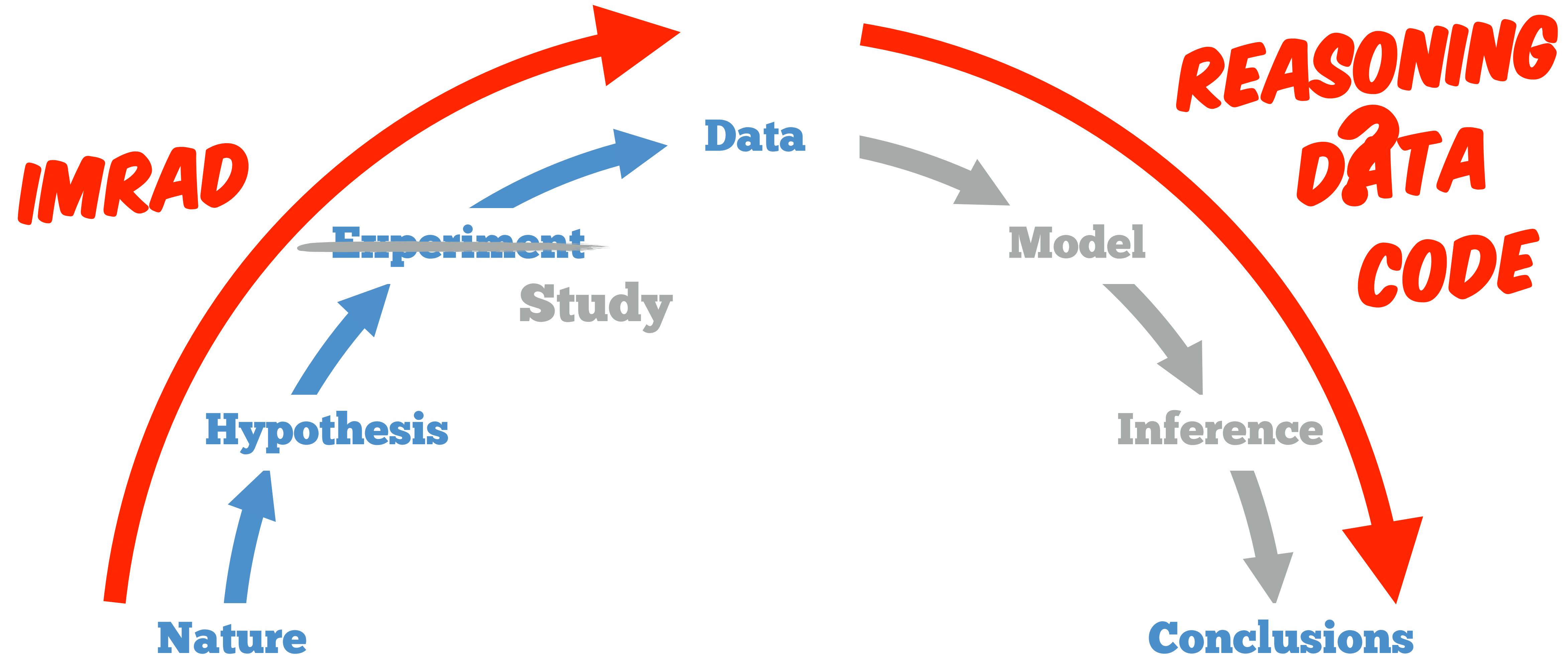
Adapted from Levy DG, Norris D. Methods and Guidelines for Integrity in Multivariate Analysis of Real World (Observational) Data. Unpublished Manuscript.

<https://rstudio.cloud/project/233945>



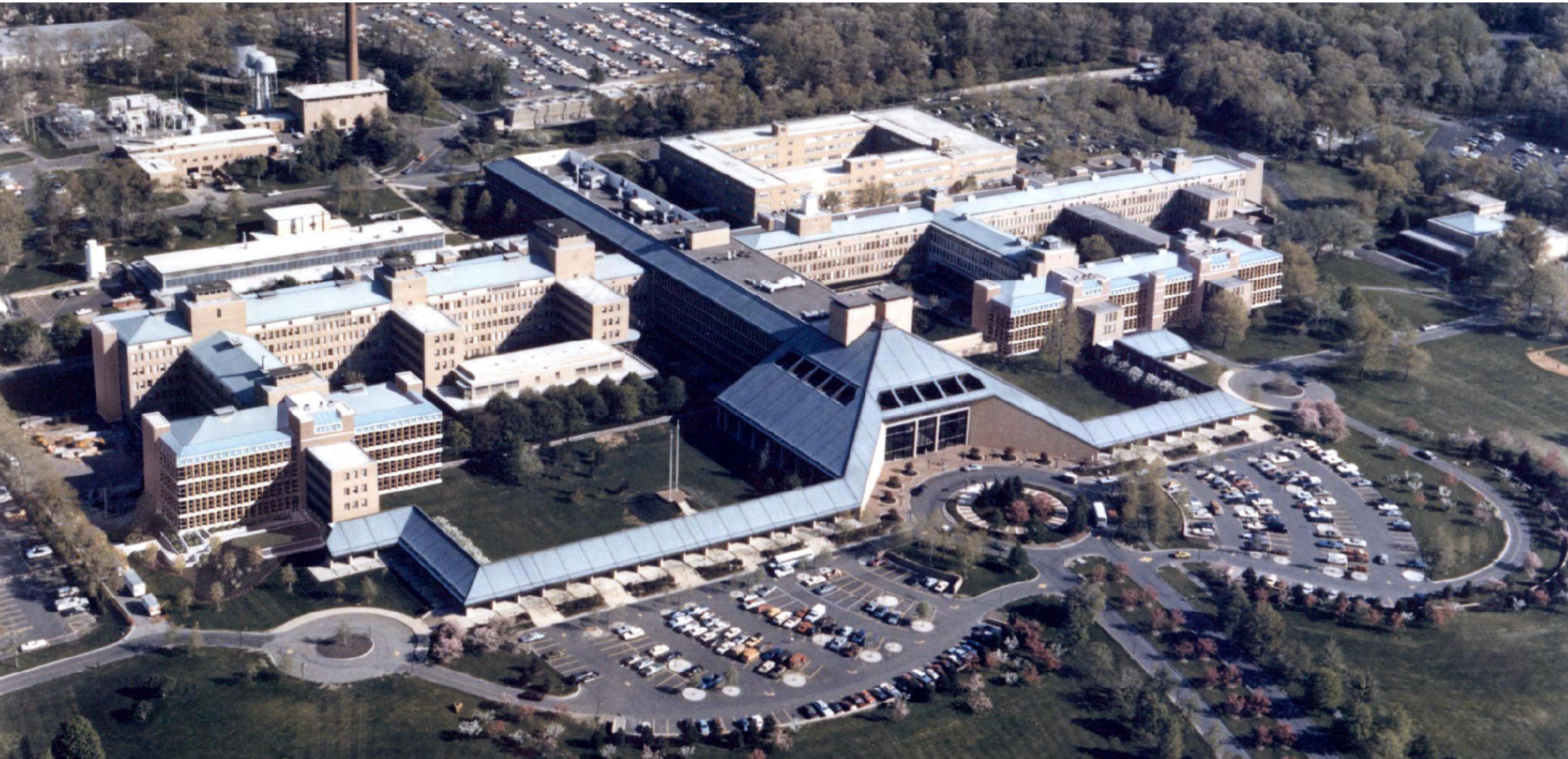
Adapted from Levy DG, Norris D. Methods and Guidelines for Integrity in Multivariate Analysis of Real World (Observational) Data. Unpublished Manuscript.

<https://rstudio.cloud/project/233945>



Adapted from Levy DG, Norris D. Methods and Guidelines for Integrity in Multivariate Analysis of Real World (Observational) Data. Unpublished Manuscript.

But which code language?





<https://rstudio.cloud/project/233945>

1. R and RStudio
2. R Markdown
3. Projects
4. Git and Github

R and RStudio



To run highlighted code:

 + Enter (Mac)

 + Enter (PC)

Your Turn

Open 01-script.R. Read the content and run it.

Determine: what does this code do?

Think

02 : 00

Pair

01 : 00

Your Turn

Open 01-script.R. Read the content and run it.

Determine: what does this code do?

**Share your conclusions with your neighbor.
Do they agree?**

Think

02 : 00

Pair

01 : 00

Your Turn

Open 02-narrative.Rmd, which narrates the analysis in 01-script.R.

Were you right?



R Markdown

R Markdown

Plain text file with 3 types of content:

The screenshot shows the RStudio interface with an R Markdown file open. The file contains the following structure:

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5  
6 Text written in **markdown**  
7  
8 ```{r}  
9 # code written in R  
10 (x <- rnorm(7))  
11 ...  
12  
13 Text written in _markdown_  
14  
15 ```{r}  
16 # code written in R  
17 hist(x)  
18 ...  
19  
20 [1] -1.2 1.0 -0.5 0.9 -0.6 -1.1 -1.5  
21  
22 Text written in _markdown_  
23  
24 ```{r}  
25 # code written in R  
26
```

Annotations explain the three types of content:

- A green callout points to the YAML header (lines 1-4) with the text: "A YAML header surrounded by ---".
- A grey callout points to the text "Text written in **markdown**" (line 6) with the text: "Text in markdown".
- A blue callout points to the code chunk "```{r}" (line 8) with the text: "Code chunks surrounded by ```".

Markdown

A faint watermark of the R logo is visible in the bottom right corner, consisting of two overlapping circles with the letters 'R' inside.

Web sites that use markdown

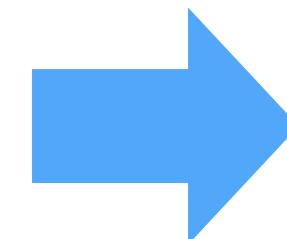
- * **GitHub** www.github.com
- * **StackOverflow** www.stackoverflow.com
- * **Reddit** www.reddit.com
- * many more

Headers

Use # to create headers.

Multiple #'s create lower level headers.

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



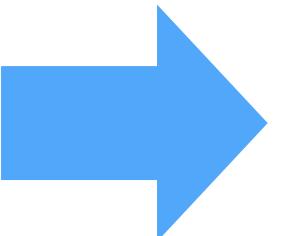
Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

Text

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

Text is rendered as plain text. Surround
text with `_`, `**`, or ``` to format it.

Text
`_italics_`
`**bold**`
``code``



Text
italics
bold
`code`

Lists

Use asterisks to make bullet points.

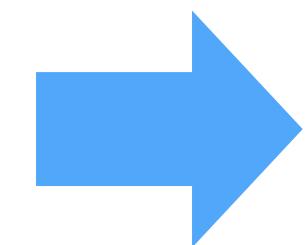
Use numbers to make numbered lists.

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

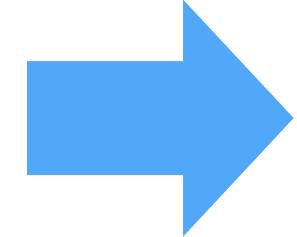
Hyperlinks

Use brackets to denote a link.

Place the URL in parentheses.

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

This is a link.

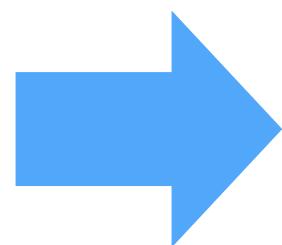


Images

Use a link preceded by an ! to insert an image.

The link text should be a URL (if the image is hosted online), or a file path (if the image is saved as a file)

The RStudio logo.

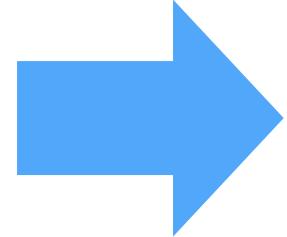


The RStudio logo.

Equations

Write equations with latex math commands and surround them in \$'s.

According to Einstein,
 $E=mc^2$



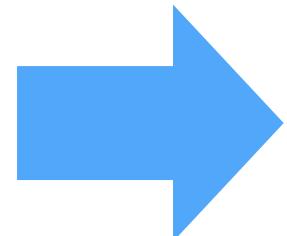
According to Einstein, $E = mc^2$

Equation blocks

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

According to
Einstein,

`$$E=mc^{\{2\}}$$`



According to
Einstein,

$$E = mc^2$$

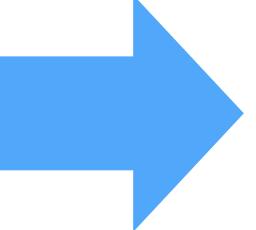
Embed code

A large, semi-transparent watermark of the R logo is positioned in the bottom right corner. The logo consists of a bold, italicized lowercase 'r' enclosed within a circular arrow.

Inline code

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

Today is
`r Sys.Date()`.



Today is 2015-04-16.

Code chunks

Insert a chunk of R code with

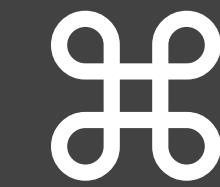
```
```{r}
some code
```
```

When you render the report, R Markdown will run the code and include its results. R Markdown will also remove the ```{r} and ```.

Code chunks

Insert a chunk of R code with

```
```{r}  
some code
```
```



+

Opt

+

A white lowercase letter 'i' enclosed in a rounded rectangle.

(Mac)

Ctrl

+

Alt

+

A white lowercase letter 'i' enclosed in a rounded rectangle.

(PC)

Your Turn

Replace every TODO in **02-narrative.Rmd**, with code from **01-script.R**. Then knit your document.

```
```{r}
some code
```
```



+

Opt

+

i

(Mac)

Ctrl

+

Alt

+

i

(PC)

04 : 00

chunk options

By default, R Markdown includes both the code and its results

Here's some code

```
```{r}
```

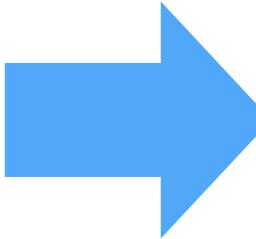
```
dim(iris)
```

```
```
```

Here's some code

```
dim(iris)
```

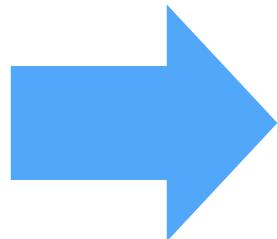
```
## [1] 150 5
```



echo

Add options in the brackets after r.
echo = FALSE hides the code.

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



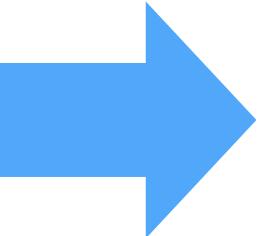
```
Here's some code  
## [1] 150 5
```

Very useful
for plots

eval

eval = FALSE prevents the code from being run. As a result, no results will be displayed with the code.

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



```
Here's some code  
dim(iris)
```

include

include = FALSE runs the code, but prevents both the code and the results from appearing (e.g. to setup).

Here's some code

```
```{r include=FALSE}  
dim(iris)
```
```

Here's some code

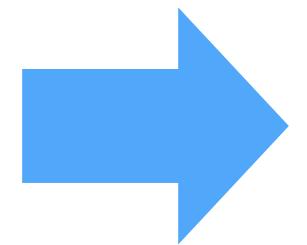
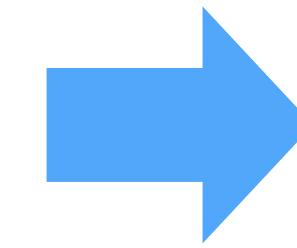


fig.height, fig.width

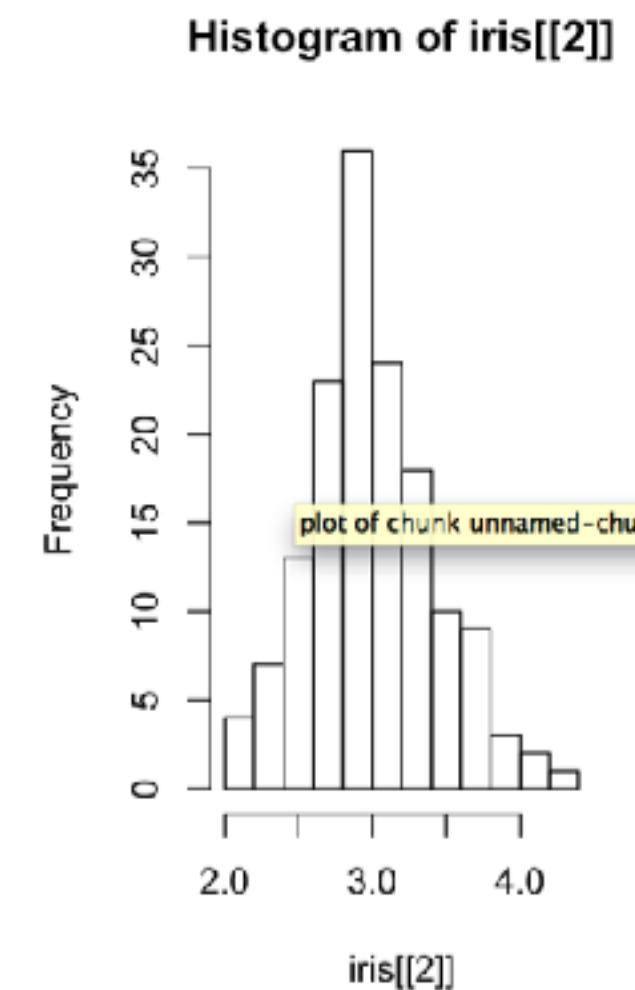
Specify the dimension of plots (in inches) with `fig.width` and `fig.height`. Separate multiple arguments with commas.

Here's a plot

```
```{r echo=FALSE, fig.width=3, fig.height=5}
hist(iris[[2]])
````
```



Here's a plot



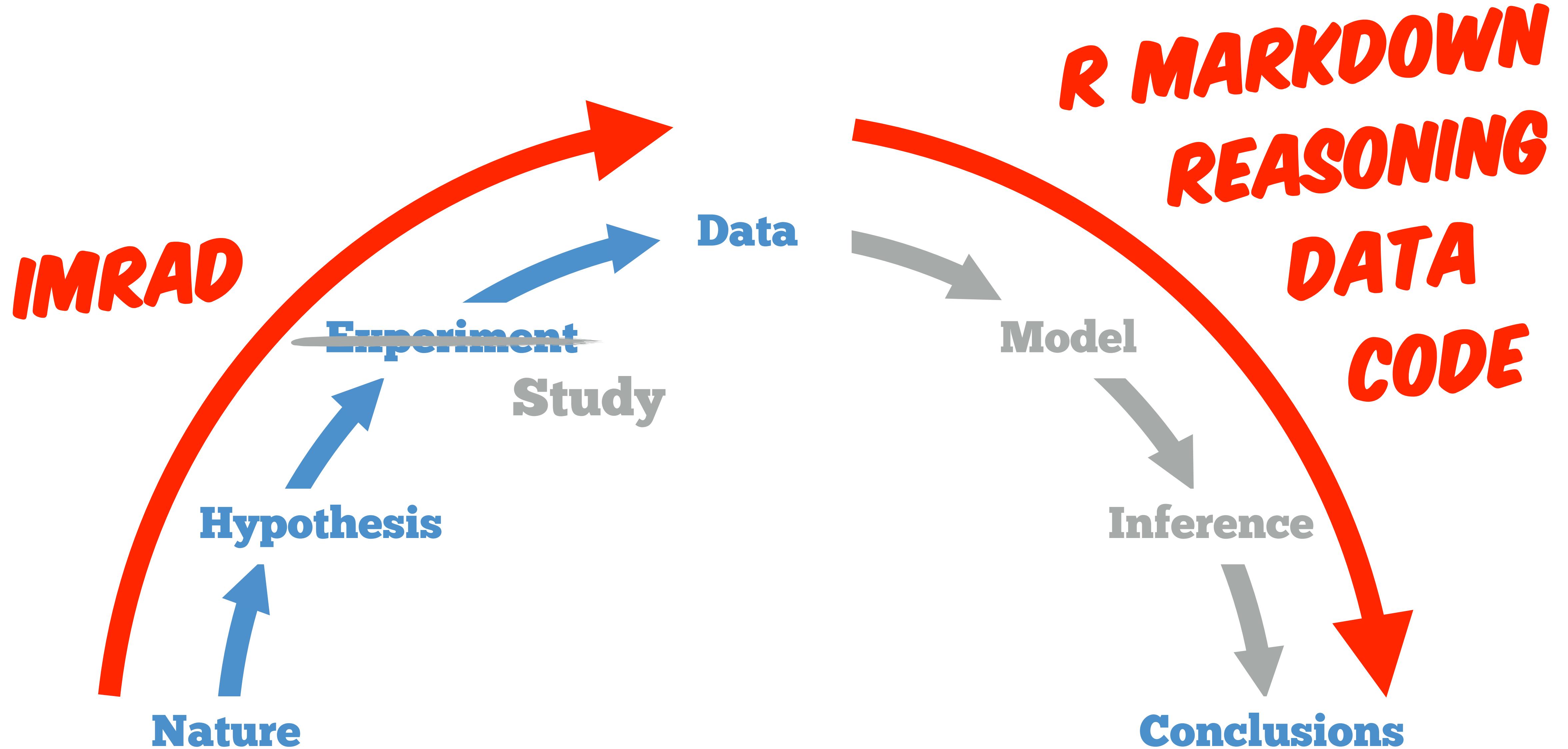
Your Turn

Add `include = FALSE` or `echo = FALSE` to each chunk as appropriate to suppress the code.

Set the height of the figure to 2 inches.

Then re-knit your document.





Adapted from Levy DG, Norris D. Methods and Guidelines for Integrity in Multivariate Analysis of Real World (Observational) Data. Unpublished Manuscript.

Parameters

R

Your Turn

Open **03-report.Rmd**. Then use Knit with Parameters to render the document.

What happens if you use a different parameter?

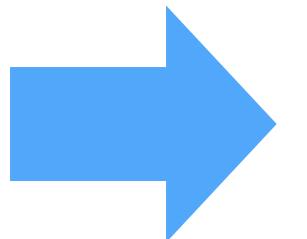


YAML

A section of key:value pairs
separated by dashed lines ----

```
---  
title: "Untitled"  
author: "RStudio"  
date: "February 4, 2015"  
output: html_document  
---
```

Text of document



Untitled

RStudio

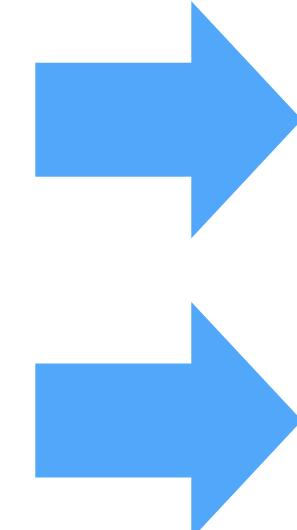
February 4, 2015

Text of document

Parameters

A list of values that you can call in R code chunks

params list
**elements and
values**



```
---
```

```
title: "Untitled"
```

```
output: html_document
```

```
params:
```

```
  filename: "data.csv"
```

```
  symbol: "FB"
```

```
---
```

colon

New line.
Indented two
spaces

Using Parameters

Call parameter values as elements of the params list, **params\$num**

```
---
```

```
params:
```

```
  num: 42
```

```
---
```

The value of the parameter is
`r params\$num`, e.g.

```
```{r}
```

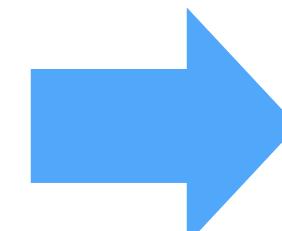
```
params$num
```

```
```
```

The value of the parameter is 42, e.g.

```
params$num
```

```
## [1] 42
```



render()

Render at the command line with default YAML options

```
library(rmarkdown)  
render("03-reports.Rmd")
```

Render at the command line, set parameters.

```
render("03-reports.Rmd",  
       params = list(data = "london.csv"))
```

render() + for

```
datasets <- c("dublin.csv", "london.csv")

for (name in datasets) {
  render("03-reports.Rmd",
        params = list(data = name))
}
```

How it works

R

knitr



pandoc



HTML



ioslides
slidy, Beamer



Powerpoint



Microsoft Word

Logistics

1

Knitr runs the document in a fresh R session, which means you need to load the libraries that the document uses *in the document*

Logistics

1

Knitr runs the document in a fresh R session, which means you need to load the libraries that the document uses *in the document*

2

Objects made in one code chunk will be available to code in later code chunks.

KNITR IS MULTILINGUAL!

 **SAS**

 **PYTHON**

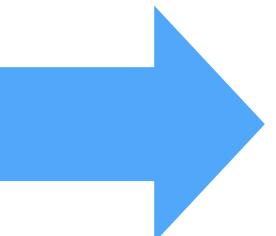
 **MORE**

engine

python

```
Some python code,  
```{python}  
x = 'hello, python
world!'
print(x)
print(x.split(' '))
```
```

To embed non R code, change the chunk label from r to the language to use.



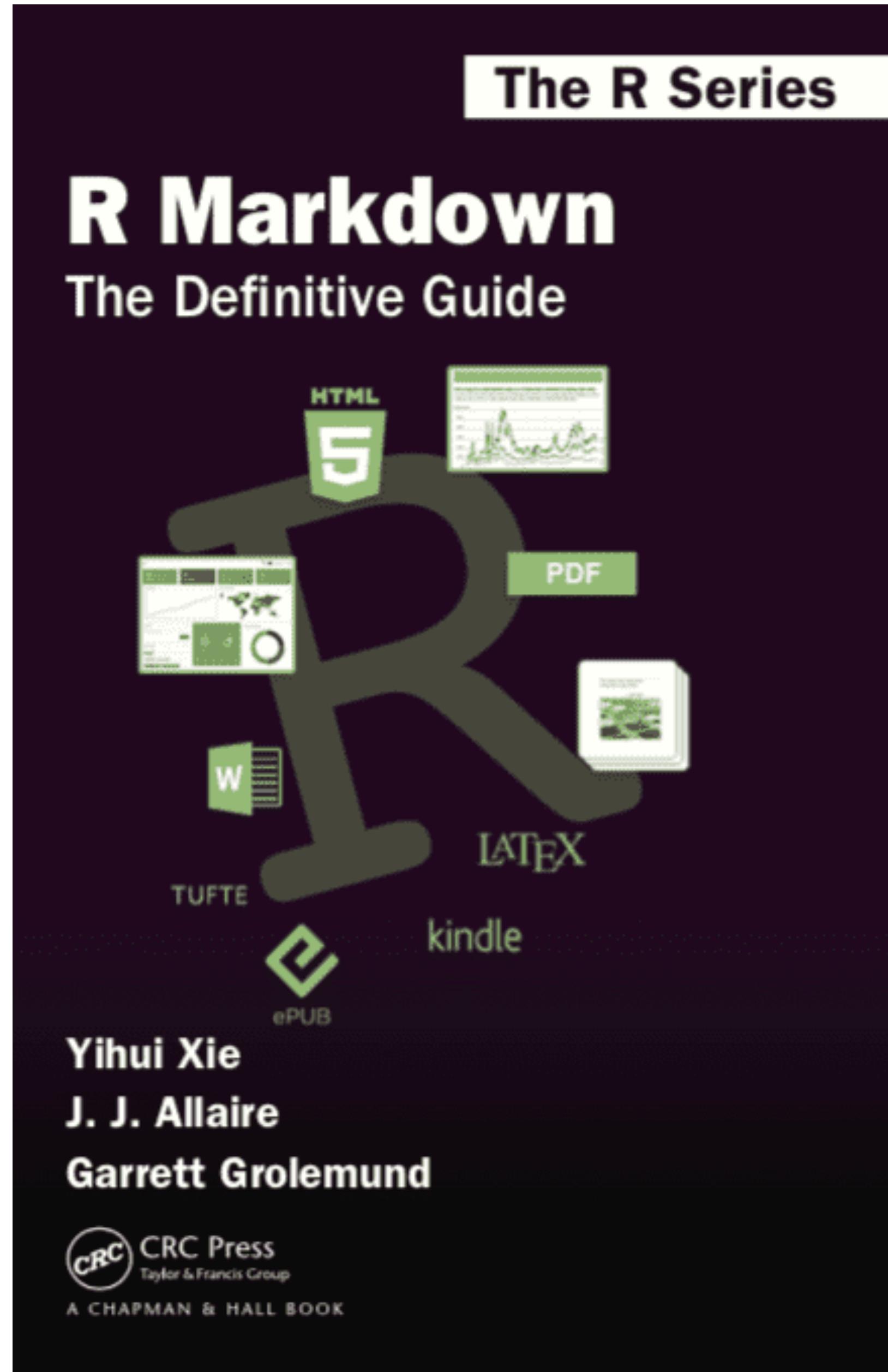
Some python code:

```
x = 'hello, python world!'  
print(x)  
print(x.split(' '))
```

```
## hello, python world!  
## ['hello,', 'python', 'world!']
```

If the language's interpreter is not on your system's PATH variable, use **engine.path** to tell knitr where to find the interpreter

```
```{sas, engine.path = "C:/Program Files/SASHome/9.4/sas.exe"}  
Some SAS code
```
```



bookdown.org/yihui/rmarkdown/

ONLINE, FREE



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 run 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

- Open a new .Rmd file at File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.
- Write document by editing template.
- Knit document to create report; Use knit button or render() to knit.
- Preview output in DE window.
- Publish (optional) to web server.
- Examine build log in R Markdown console.
- Use output file that is saved alongside .Rmd.

render

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

| | | | | | |
|-------------------------------|---|--------------------|---------------------------------------|---|---------------------------------|
| input - file to render | output_options - List of render options, as in YAML. | output_file | 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/getVersion()`.

CODE CHUNKS
One or more lines surrounded with `'''(r)` and `'''`. Place chunk options within curly braces, after `r`. Use `echo` with `getVersion()`.

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) | dependson - chunk dependencies for caching (default = NULL) | fig.align - 'left', 'right', or 'center' (default = 'left') | message - display code messages in document (default = TRUE) | | |
| cache.path - directory to save cached results in (default = "cache/") | echo - Display code in output document: (default = TRUE) | fig.cap - figure caption as character string (default = NULL) | results (default = 'markup') as 's' - passthrough results | | |
| child - file(s) to knit and then include (default = NULL) | engine - code language used in chunk (default = 'R') | fig.height, fig.width - Dimensions of plots in inches | highlight - highlight source code (Default = TRUE) | | |
| collapse - collapse all output into single block (default = FALSE) | error - Display error messages in doc (TRUE) or stop rendering when errors occur (FALSE) (default = FALSE) | hold - put all results below all code | tidy - tidy code for display (default = FALSE) | | |
| comment - prefix for each line of results (default = '#') | eval - run code in chunk (default = TRUE) | print - display code results (default = TRUE) | warning - display code warnings in document (default = TRUE) | | |
| Options not listed above: <code>options</code> , <code>ani.options</code> , <code>animin</code> , <code>background</code> , <code>carpet.comments</code> , <code>carpet.lazy</code> , <code>carpet.reuse</code> , <code>carpet.vars</code> , <code>dev</code> , <code>dev.args</code> , <code>engine.opts</code> , <code>engine.path</code> , <code>fig.asp</code> , <code>fig.env</code> , <code>fig.ext</code> , <code>fig.keep</code> , <code>fig.lp</code> , <code>fig.path</code> , <code>fig.pos</code> , <code>fig.process</code> , <code>fig.tilta</code> , <code>fig.scap</code> , <code>fig.show</code> , <code>fig.showtext</code> , <code>fig.subcap</code> , <code>internal</code> , <code>out.extra</code> , <code>out.height</code> , <code>out.width</code> , <code>prompt</code> , <code>purl</code> , <code>ref.label</code> , <code>render.site</code> , <code>split</code> , <code>tidy.opts</code> | | | | | |

R Markdown :: .rmd Structure

.YAML Header
Optional section of header (e.g., `version`) options written as key:value pairs (YAML).
At start of file
Between lines of

TEXT
Text all formated with markdown, mixed with:

Code Chunks
Chunks of embedded code. Each chunk:
Begin with `'''(r)`
End with `'''`
R Markdown will run the code and append the results to the doc. It will use the location of the `.Rmdfile` as the **working directory**.

Parameters

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

- Add parameters** - Create and set parameters in the header as sub-values of `params`
- Call parameters** - Call parameter values in code as `params$name`
- Set parameters** - Set values with `knit` with parameters or `.Rmd` parameters 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

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

output: `html_document`
`runtimes: shiny`
`---`
`'''(r, echo = FALSE)`
`numericInput("n", "How many cars?", 5)`
`renderTable({`
 `read.csv("cars.csv")`
`})`

Shiny

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 `runApp`.

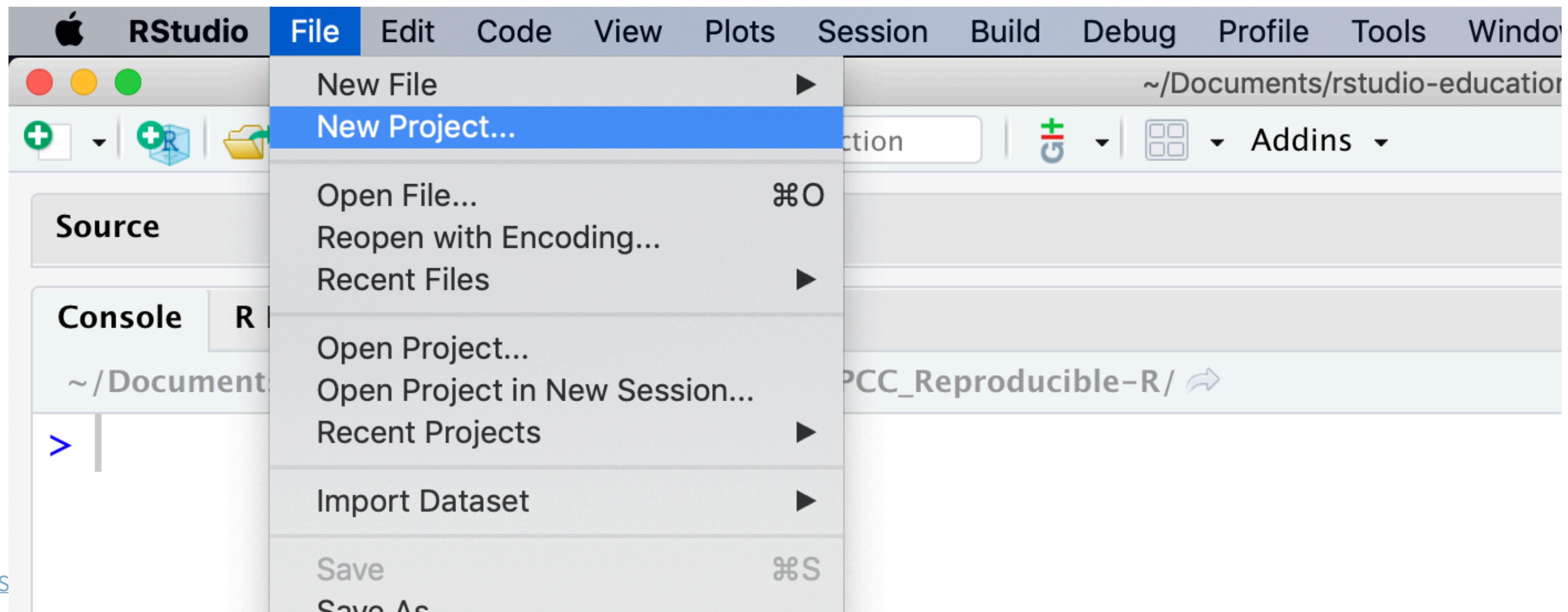
rstudio.com/cheatsheets

Projects



How to keep track of your files?

Put them in a directory and make it a **PROJECT**

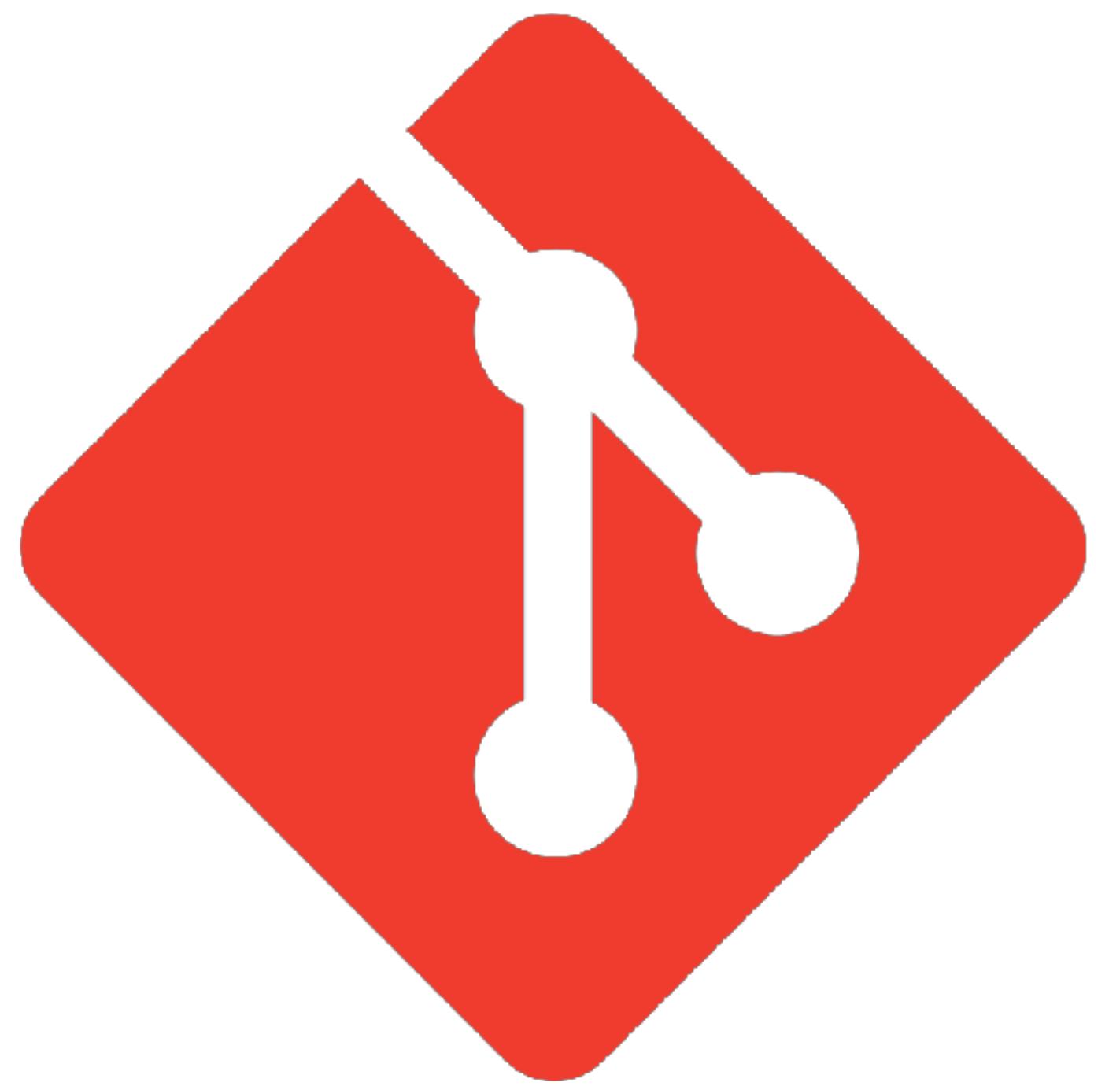


Git and Github

R

What is version control?

Why should you use it?



git

History

1st Commit 2nd Commit 3rd Commit 4th Commit 5th Commit

```
"name","year","time","lat","long"
"Allison",1995,1995-06-03 00:00:00,17.4,-84.3
"Allison",1995,1995-06-03 06:00:00,18.3,-84.9
"Allison",1995,1995-06-03 12:00:00,19.3,-85.7
"Allison",1995,1995-06-03 18:00:00,20.6,-85.8
"Allison",1995,1995-06-04 00:00:00,22,-86
"Allison",1995,1995-06-04 06:00:00,23.3,-86.3
"Allison",1995,1995-06-04 12:00:00,24.7,-86.2
"Allison",1995,1995-06-04 18:00:00,26.2,-86.2
"Allison",1995,1995-06-05 00:00:00,27.6,-86.1
"Allison",1995,1995-06-05 06:00:00,28.5,-85.6
"Allison",1995,1995-06-05 12:00:00,29.6,-84.7
"Allison",1995,1995-06-05 18:00:00,30.7,-83.8
"Allison",1995,1995-06-06 00:00:00,31.8,-82.8
"Allison",1995,1995-06-06 06:00:00,32.7,-81.5
"Allison",1995,1995-06-06 12:00:00,33.6,-80
"Allison",1995,1995-06-06 18:00:00,34.5,-78.1
"Allison",1995,1995-06-07 00:00:00,35.6,-75.9
"Allison",1995,1995-06-07 06:00:00,37.1,-73.6
"Allison",1995,1995-06-07 12:00:00,38.5,-71
"Allison",1995,1995-06-07 18:00:00,39.8,-69.2
"Allison",1995,1995-06-08 00:00:00,41,-67.7
"Allison",1995,1995-06-08 06:00:00,42.4,-66
"Allison",1995,1995-06-08 12:00:00,43.8,-63.7
```

```
# 0-Clean.R

library(dplyr)
library(lubridate)

storms <- read.csv("storms.csv")

storms <- storms %>%
  mutate(time = ymd_h(paste(year, month, day,
    hour))) %>%
  select(name, year, time, lat, long,
  pressure, wind, type)

write.csv(storms, file = "storms.csv",
  row.names = FALSE)
```

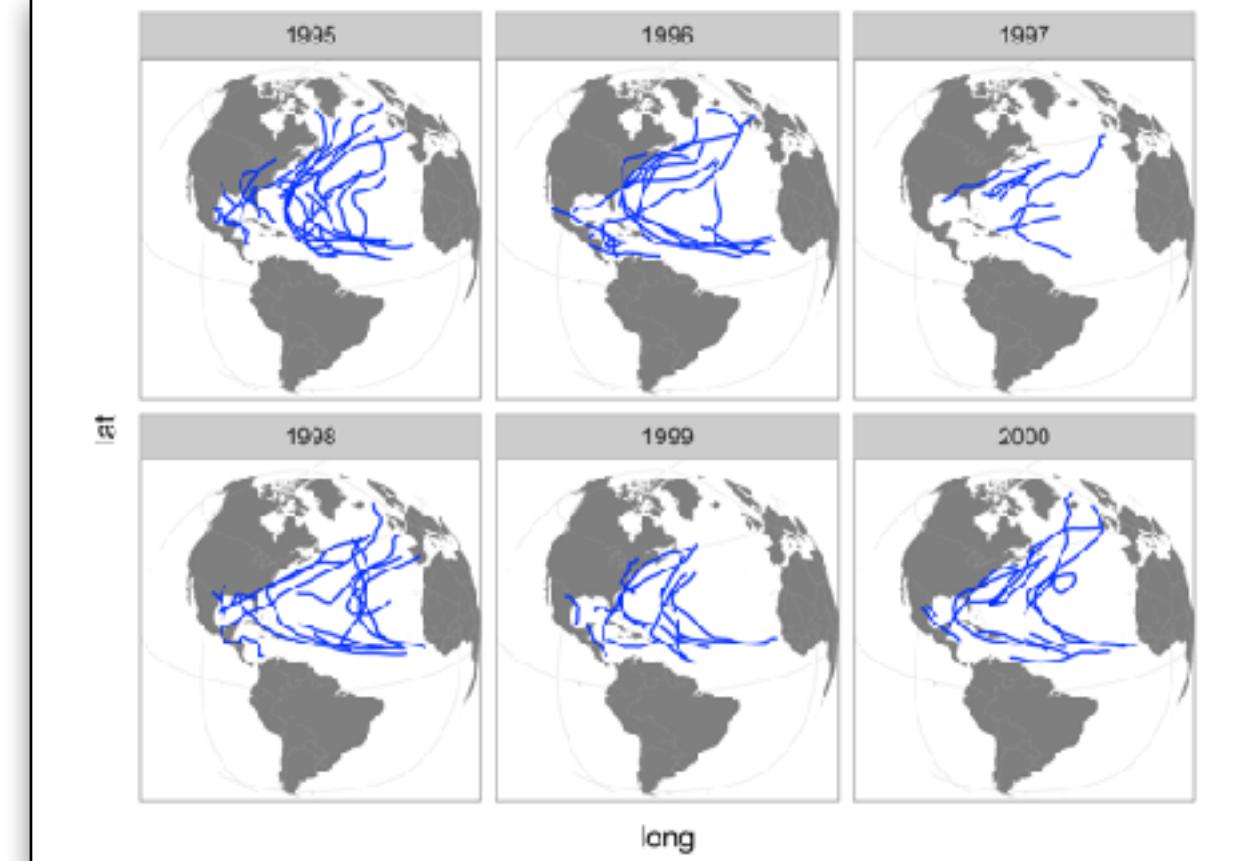
```
# 1-Plot.R

library(ggplot2)
library(dplyr)

map <- map_data("world") %>%
  filter(region != "USSR")

ggplot(storms, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group),
    fill = "grey50", data = map) +
  geom_path(aes(group = name),
    color = "blue") +
  facet_wrap(~ year) +
  theme_bw() +
  coord_map(projection = "ortho",
    orientation = c(21, -60, 0))

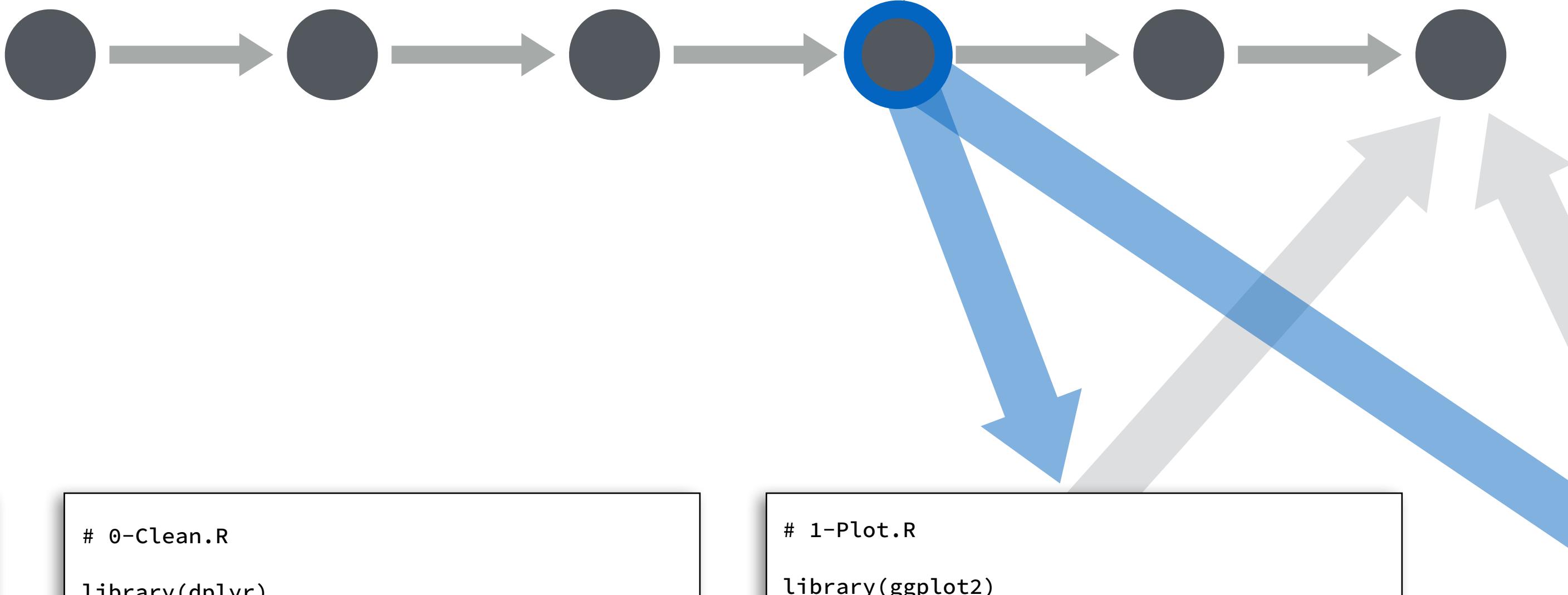
ggsave("storms.png", width = 7, height = 5)
```



Project

History

1st Commit 2nd Commit 3rd Commit 4th Commit 5th Commit 6th Commit



```
"name","year","time","lat","long"
"Allison",1995,1995-06-03 00:00:00,17.4,-84.3
"Allison",1995,1995-06-03 06:00:00,18.3,-84.9
"Allison",1995,1995-06-03 12:00:00,19.3,-85.7
"Allison",1995,1995-06-03 18:00:00,20.6,-85.8
"Allison",1995,1995-06-04 00:00:00,22,-86
"Allison",1995,1995-06-04 06:00:00,23.3,-86.3
"Allison",1995,1995-06-04 12:00:00,24.7,-86.2
"Allison",1995,1995-06-04 18:00:00,26.2,-86.2
"Allison",1995,1995-06-05 00:00:00,27.6,-86.1
"Allison",1995,1995-06-05 06:00:00,28.5,-85.6
"Allison",1995,1995-06-05 12:00:00,29.6,-84.7
"Allison",1995,1995-06-05 18:00:00,30.7,-83.8
"Allison",1995,1995-06-06 00:00:00,31.8,-82.8
"Allison",1995,1995-06-06 06:00:00,32.7,-81.5
```

```
# 0-Clean.R

library(dplyr)
library(lubridate)

storms <- read.csv("storms.csv")

storms <- storms %>%
  mutate(time = ymd_h(paste(year, month, day,
    hour))) %>%
  select(name, year, time, lat, long,
  pressure, wind, type)

write.csv(storms, file = "storms.csv",
  row.names = FALSE)
```

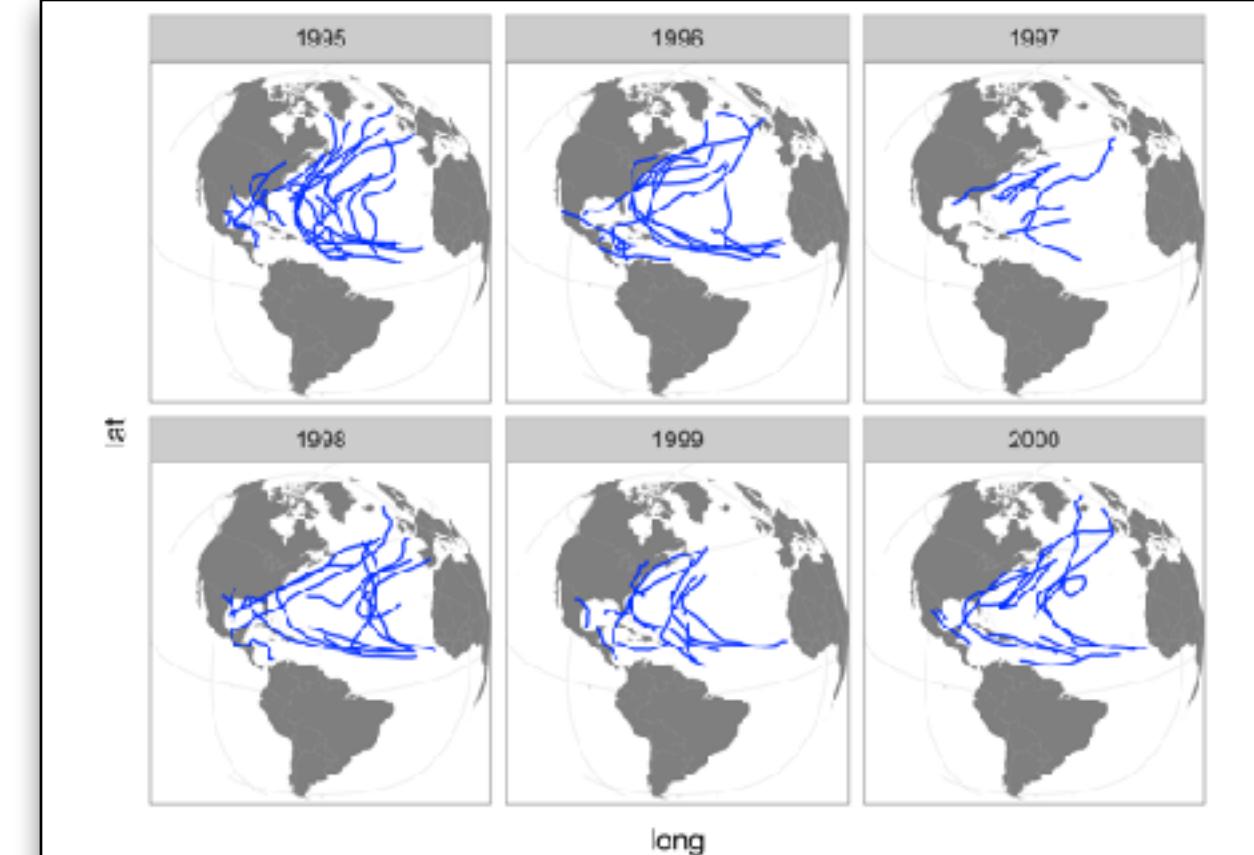
```
# 1-Plot.R

library(ggplot2)
library(dplyr)

map <- map_data("world") %>%
  filter(region != "USSR")

ggplot(storms, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group),
    fill = "grey50", data = map) +
  geom_path(aes(group = name),
    color = "blue") +
  facet_wrap(~ year) +
  theme_bw() +
  coord_map(projection = "ortho",
    orientation = c(21, -60, 0))

ggsave("storms.png", width = 7, height = 5)
```



Project

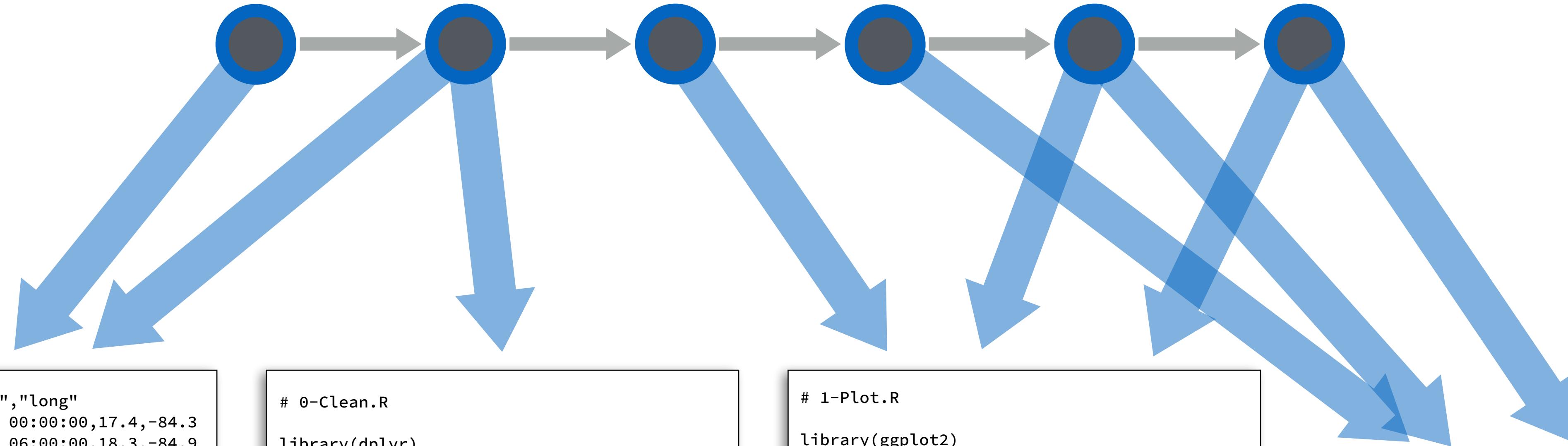
History



.git

History

1st Commit 2nd Commit 3rd Commit 4th Commit 5th Commit 6th Commit



```
"name","year","time","lat","long"
"Allison",1995,1995-06-03 00:00:00,17.4,-84.3
"Allison",1995,1995-06-03 06:00:00,18.3,-84.9
"Allison",1995,1995-06-03 12:00:00,19.3,-85.7
"Allison",1995,1995-06-03 18:00:00,20.6,-85.8
"Allison",1995,1995-06-04 00:00:00,22,-86
"Allison",1995,1995-06-04 06:00:00,23.3,-86.3
"Allison",1995,1995-06-04 12:00:00,24.7,-86.2
"Allison",1995,1995-06-04 18:00:00,26.2,-86.2
"Allison",1995,1995-06-05 00:00:00,27.6,-86.1
"Allison",1995,1995-06-05 06:00:00,28.5,-85.6
"Allison",1995,1995-06-05 12:00:00,29.6,-84.7
"Allison",1995,1995-06-05 18:00:00,30.7,-83.8
"Allison",1995,1995-06-06 00:00:00,31.8,-82.8
"Allison",1995,1995-06-06 06:00:00,32.7,-81.5
"Allison",1995,1995-06-06 12:00:00,33.6,-80
"Allison",1995,1995-06-06 18:00:00,34.5,-78.1
"Allison",1995,1995-06-07 00:00:00,35.6,-75.9
"Allison",1995,1995-06-07 06:00:00,37.1,-73.6
"Allison",1995,1995-06-07 12:00:00,38.5,-71
"Allison",1995,1995-06-07 18:00:00,39.8,-69.2
"Allison",1995,1995-06-08 00:00:00,41,-67.7
"Allison",1995,1995-06-08 06:00:00,42.4,-66
"Allison",1995,1995-06-08 12:00:00,43.8,-63.7
```

```
# 0-Clean.R
library(dplyr)
library(lubridate)

storms <- read.csv("storms.csv")

storms <- storms %>%
  mutate(time = ymd_h(paste(year, month, day,
    hour))) %>%
  select(name, year, time, lat, long,
pressure, wind, type)

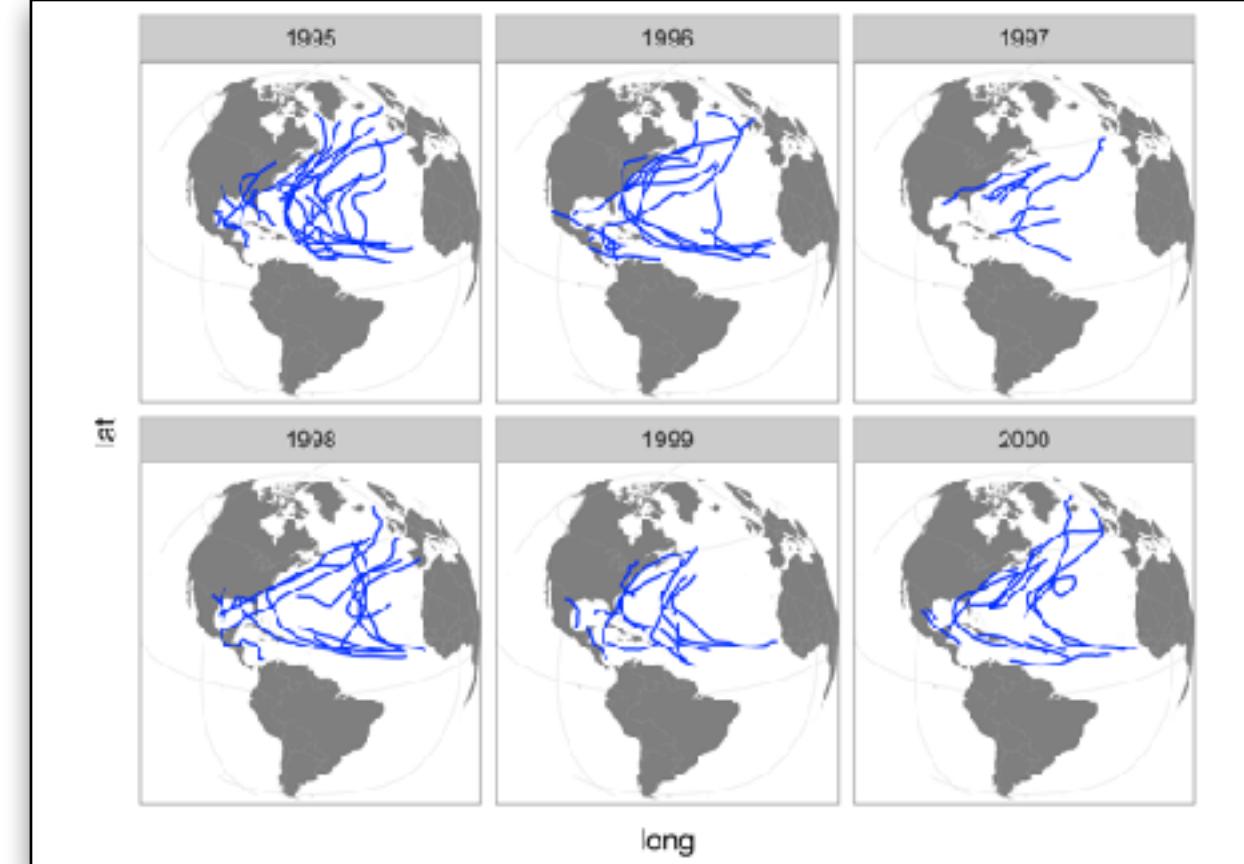
write.csv(storms, file = "storms.csv",
  row.names = FALSE)
```

```
# 1-Plot.R
library(ggplot2)
library(dplyr)

map <- map_data("world") %>%
  filter(region != "USSR")

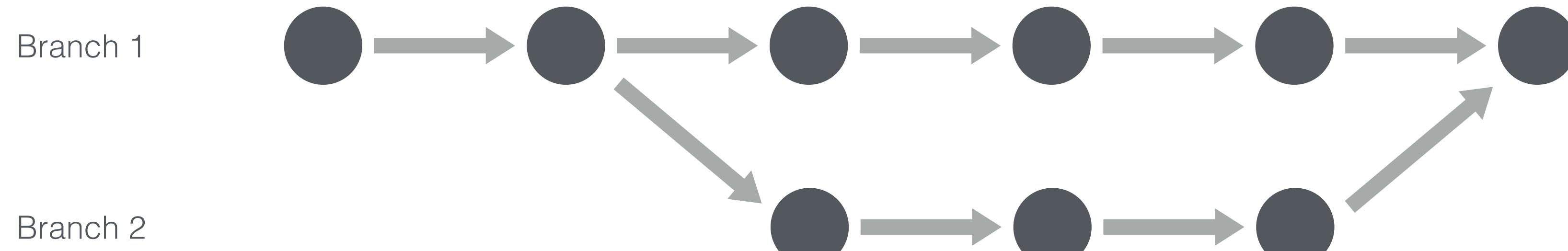
ggplot(storms, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group),
    fill = "grey50", data = map) +
  geom_path(aes(group = name),
    color = "blue") +
  facet_wrap(~ year) +
  theme_bw() +
  coord_map(projection = "ortho",
    orientation = c(21, -60, 0))

ggsave("storms.png", width = 7, height = 5)
```



Project

History



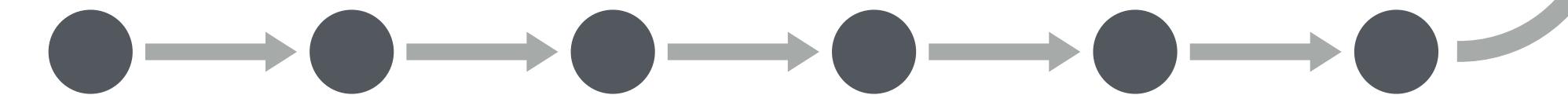
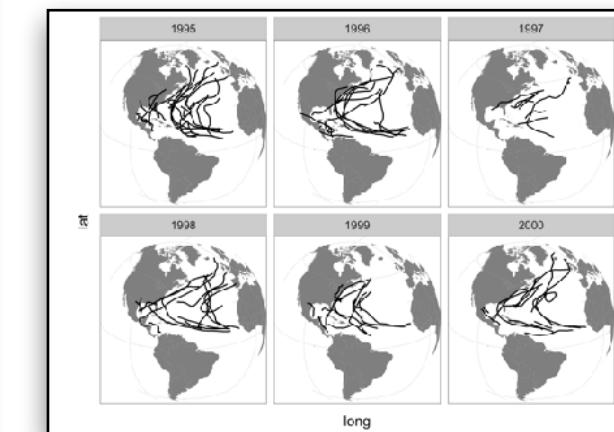
"Official" Version

implied by commit history

```
"name", "year", "time", "lat", "long"  
"Allison", 1995, 1995-06-03 00:00:00, 17.4, -84.3  
"Allison", 1995, 1995-06-03 06:00:00, 18.3, -84.9  
"Allison", 1995, 1995-06-03 12:00:00, 19.3, -85.7  
"Allison", 1995, 1995-06-03 18:00:00, 20.6, -85.8  
"Allison", 1995, 1995-06-04 00:00:00, 22, -86  
"Allison", 1995, 1995-06-04 06:00:00, 23.3, -86.3  
"Allison", 1995, 1995-06-04 12:00:00, 24.7, -86.2  
"Allison", 1995, 1995-06-04 18:00:00, 26.2, -86.2  
"Allison", 1995, 1995-06-05 00:00:00, 27.6, -86.1  
"Allison", 1995, 1995-06-05 06:00:00, 28.5, -85.6  
"Allison", 1995, 1995-06-05 12:00:00, 29.6, -84.7  
"Allison", 1995, 1995-06-05 18:00:00, 30.7, -83.8  
"Allison", 1995, 1995-06-06 00:00:00, 31.8, -82.8  
"Allison", 1995, 1995-06-06 06:00:00, 32.7, -81.5  
"Allison", 1995, 1995-06-06 12:00:00, 33.6, -80  
"Allison", 1995, 1995-06-06 18:00:00, 34.5, -78.1  
"Allison", 1995, 1995-06-07 00:00:00, 35.6, -75.9  
"Allison", 1995, 1995-06-07 06:00:00, 37.1, -73.6  
"Allison", 1995, 1995-06-07 12:00:00, 38.5, -71  
"Allison", 1995, 1995-06-07 18:00:00, 39.8, -69.2  
"Allison", 1995, 1995-06-08 00:00:00, 41, -67.7  
"Allison", 1995, 1995-06-08 06:00:00, 42.4, -66  
"Allison", 1995, 1995-06-08 12:00:00, 43.8, -63.7
```

```
# 0-Clean.R  
  
library(dplyr)  
library(lubridate)  
  
storms <- read.csv("storms.csv")  
  
storms <- storms %>%  
  mutate(time = ymd_h(paste(year,  
    month, day,  
    hour))) %>%  
  select(name, year, time, lat,  
    long, pressure, wind, type)  
  
write.csv(storms, file =  
  "storms.csv",  
  row.names = FALSE)
```

```
# 1-Plot.R  
  
library(ggplot2)  
map <- map_data("world") %>%  
  filter(region != "USSR")  
  
ggplot(storms, aes(x = long, y =  
  lat)) +  
  geom_polygon(aes(group = group),  
    fill = "grey50", data = map) +  
  geom_path(aes(group = name),  
    color = "black") +  
  facet_wrap(~ year) +  
  theme_bw() +  
  coord_map(projection = "ortho",  
    orientation = c(21, -60, 0))  
  
ggsave("storms.png", width = 7,  
  height = 5)
```



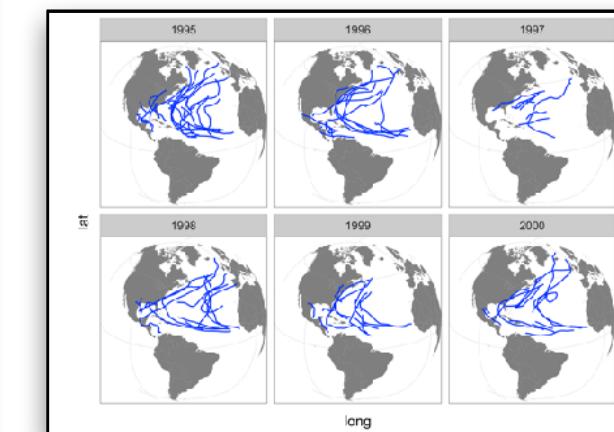
Real Life Version

uncommitted changes
in blue

```
"name", "year", "time", "lat", "long"  
"Allison", 1995, 1995-06-03 00:00:00, 17.4, -84.3  
"Allison", 1995, 1995-06-03 06:00:00, 18.3, -84.9  
"Allison", 1995, 1995-06-03 12:00:00, 19.3, -85.7  
"Allison", 1995, 1995-06-03 18:00:00, 20.6, -85.8  
"Allison", 1995, 1995-06-04 00:00:00, 22, -86  
"Allison", 1995, 1995-06-04 06:00:00, 23.3, -86.3  
"Allison", 1995, 1995-06-04 12:00:00, 24.7, -86.2  
"Allison", 1995, 1995-06-04 18:00:00, 26.2, -86.2  
"Allison", 1995, 1995-06-05 00:00:00, 27.6, -86.1  
"Allison", 1995, 1995-06-05 06:00:00, 28.5, -85.6  
"Allison", 1995, 1995-06-05 12:00:00, 29.6, -84.7  
"Allison", 1995, 1995-06-05 18:00:00, 30.7, -83.8  
"Allison", 1995, 1995-06-06 00:00:00, 31.8, -82.8  
"Allison", 1995, 1995-06-06 06:00:00, 32.7, -81.5  
"Allison", 1995, 1995-06-06 12:00:00, 33.6, -80  
"Allison", 1995, 1995-06-06 18:00:00, 34.5, -78.1  
"Allison", 1995, 1995-06-07 00:00:00, 35.6, -75.9  
"Allison", 1995, 1995-06-07 06:00:00, 37.1, -73.6  
"Allison", 1995, 1995-06-07 12:00:00, 38.5, -71  
"Allison", 1995, 1995-06-07 18:00:00, 39.8, -69.2  
"Allison", 1995, 1995-06-08 00:00:00, 41, -67.7  
"Allison", 1995, 1995-06-08 06:00:00, 42.4, -66  
"Allison", 1995, 1995-06-08 12:00:00, 43.8, -63.7
```

```
# 0-Clean.R  
  
library(dplyr)  
library(lubridate)  
  
storms <- read.csv("storms.csv")  
  
storms <- storms %>%  
  mutate(time = ymd_h(paste(year,  
    month, day, hour))) %>%  
  select(name, year, time, lat,  
    long, pressure, wind, type)  
  
write.csv(storms, file =  
  "storms.csv",  
  row.names = FALSE)
```

```
# 1-Plot.R  
  
library(ggplot2)  
map <- map_data("world") %>%  
  filter(region != "USSR")  
  
ggplot(storms, aes(x = long, y =  
  lat)) +  
  geom_polygon(aes(group = group),  
    fill = "grey50", data = map) +  
  geom_path(aes(group = name),  
    color = "blue") +  
  facet_wrap(~ year) +  
  theme_bw() +  
  coord_map(projection = "ortho",  
    orientation = c(21, -60, 0))  
  
ggsave("storms.png", width = 7,  
  height = 5)
```





GitHub

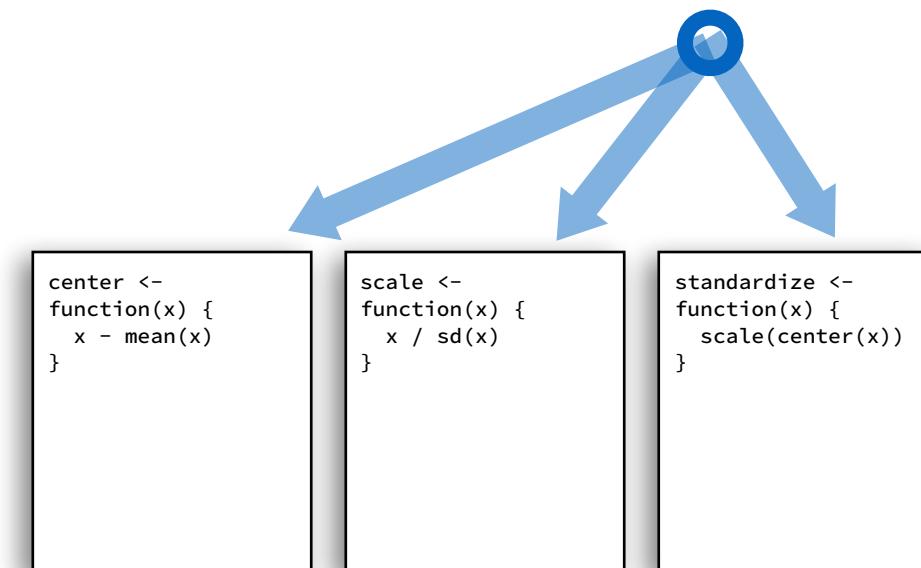
www.github.com



GitHub

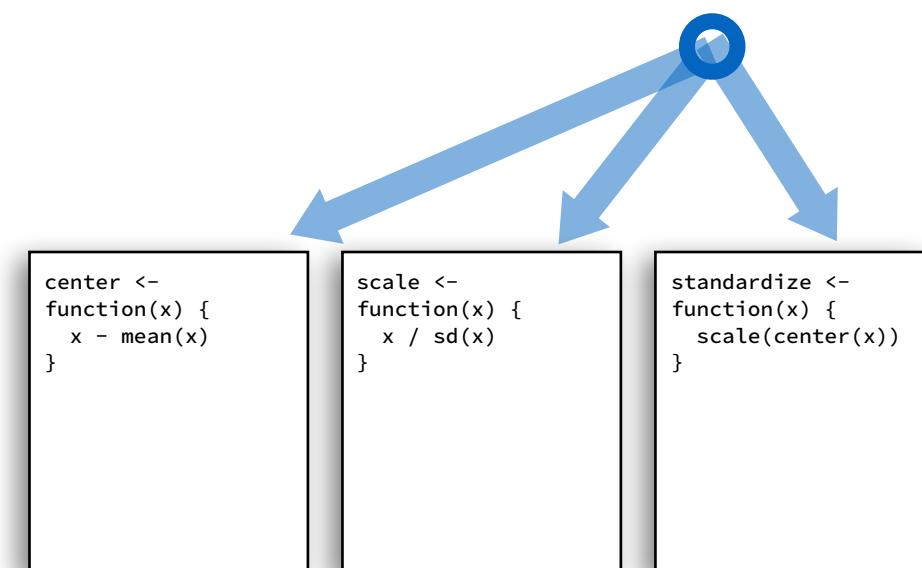


```
center <-  
function(x) {  
  x - mean(x)  
}  
  
scale <-  
function(x) {  
  x / sd(x)  
}  
  
standardize <-  
function(x) {  
  scale(function(x) {  
    x - mean(x)  
})  
}
```



You

Collaborator 1



Collaborator 2



GitHub



```
center <-  
function(x) {  
  x - mean(x)  
}  
  
scale <-  
function(x) {  
  x / sd(x)  
}  
  
standardize <-  
function(x) {  
  scale(center(x))  
}  
  
reverse <-  
function(x) {  
  rev(x)  
}
```



```
center <-  
function(x) {  
  x - mean(x)  
}  
  
scale <-  
function(x) {  
  x / sd(x)  
}  
  
standardize <-  
function(x) {  
  scale(center(x))  
}  
  
reverse <-  
function(x) {  
  rev(x)  
}
```



```
center <-  
function(x) {  
  x - mean(x)  
}  
  
scale <-  
function(x) {  
  x / sd(x)  
}  
  
standardize <-  
function(x) {  
  scale(center(x))  
}  
  
reverse <-  
function(x) {  
  rev(x)  
}
```



You

Collaborator 1

Collaborator 2



GitHub

3. Github Version

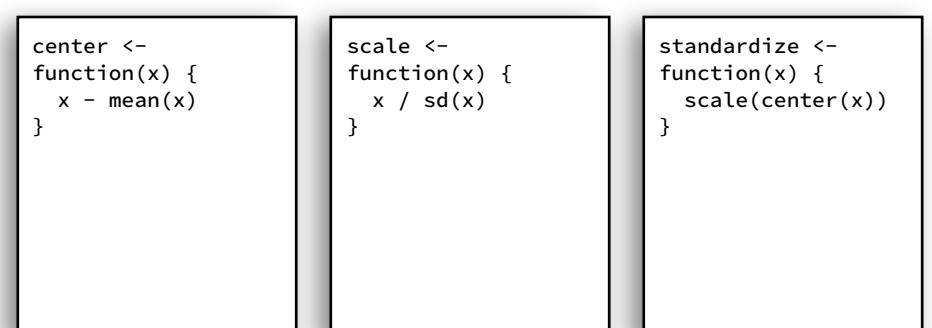
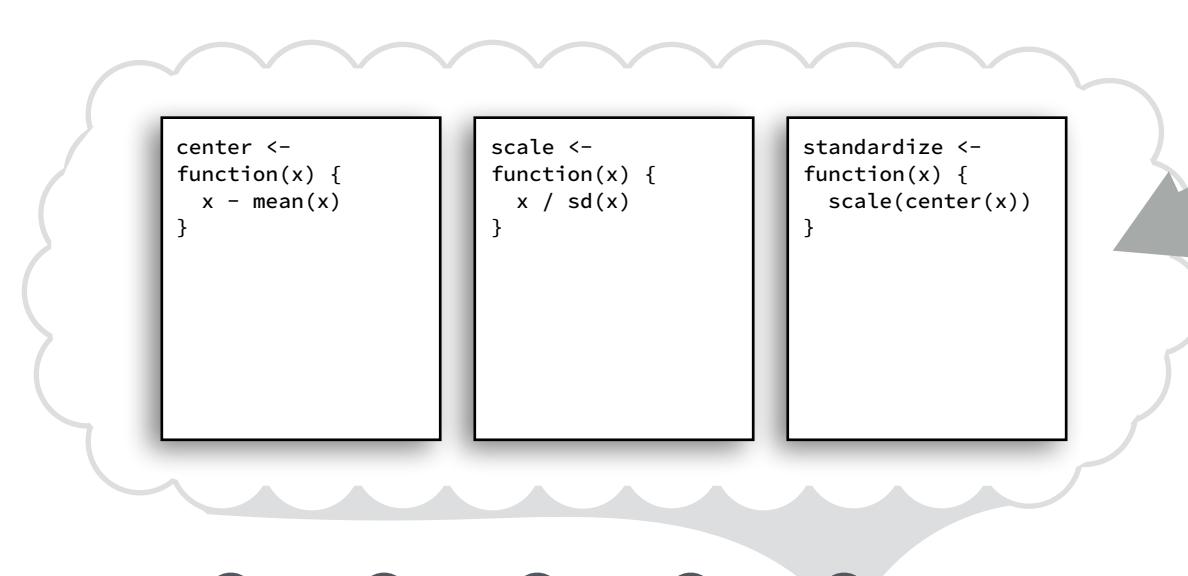
available to public

2."Official" Version

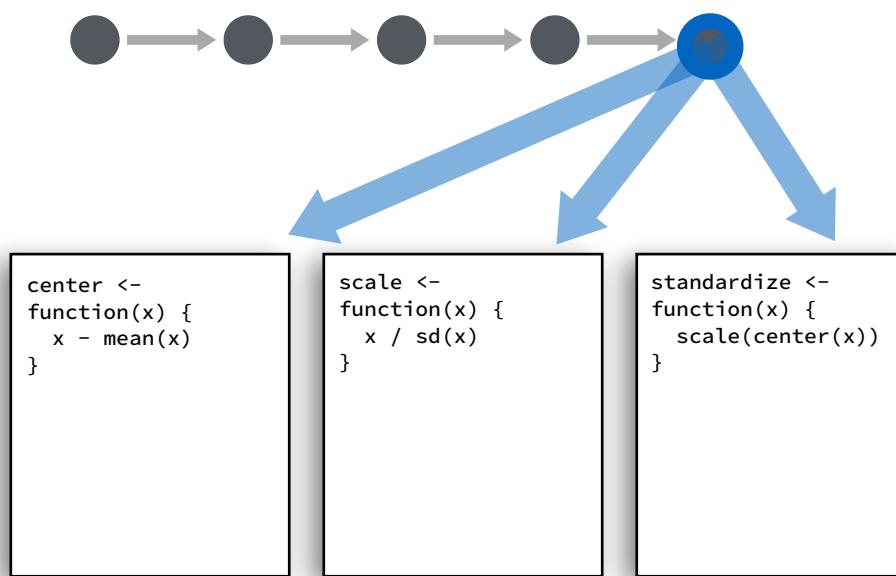
implied by commit history

1. Real Life Version

in your working directory



You





GitHub

3. Github Version

available to public

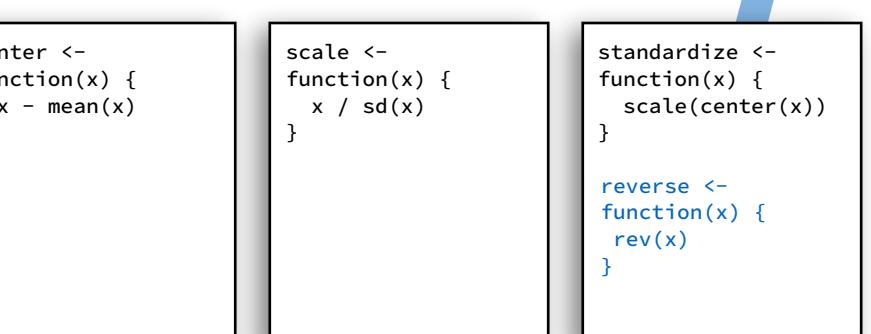
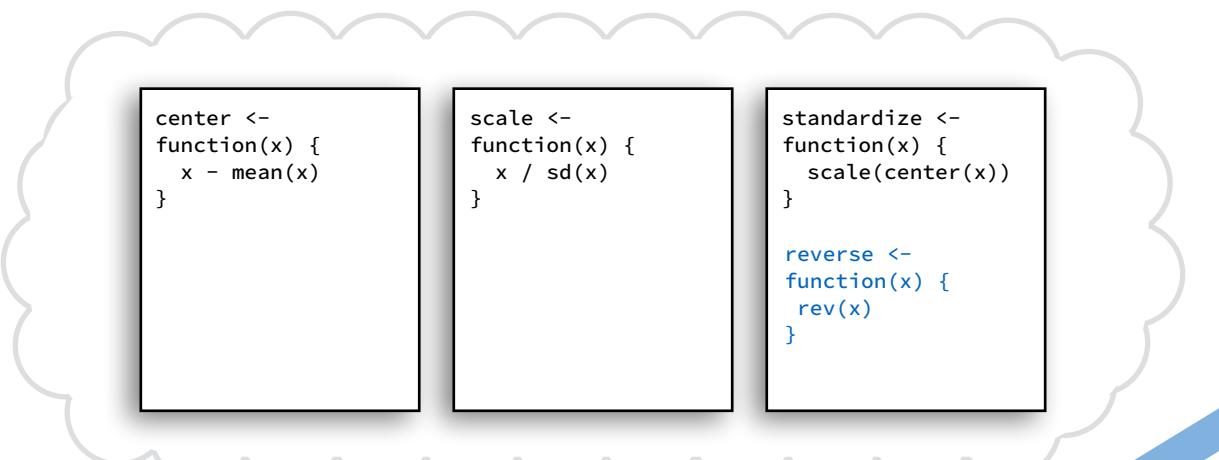
2."Official" Version

implied by commit history

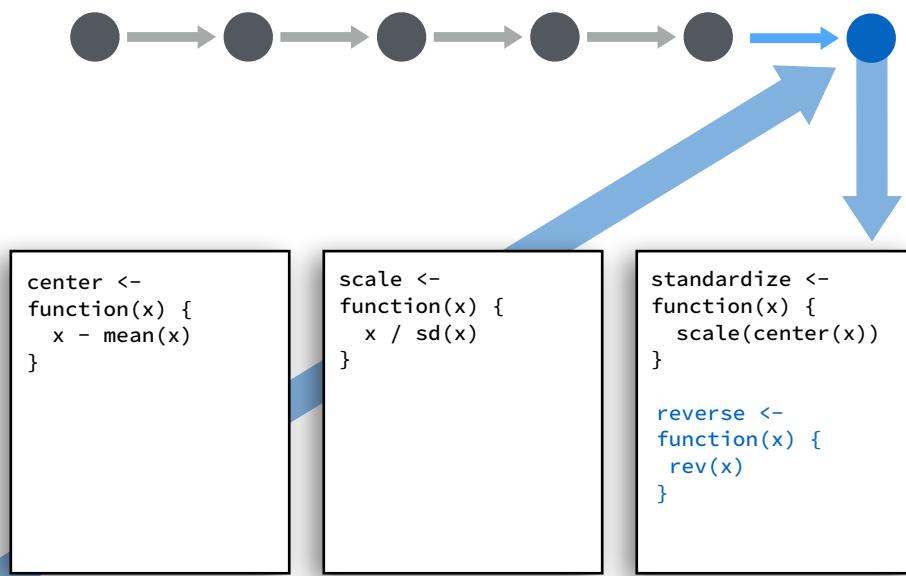
1. Real Life Version

in your working directory

You

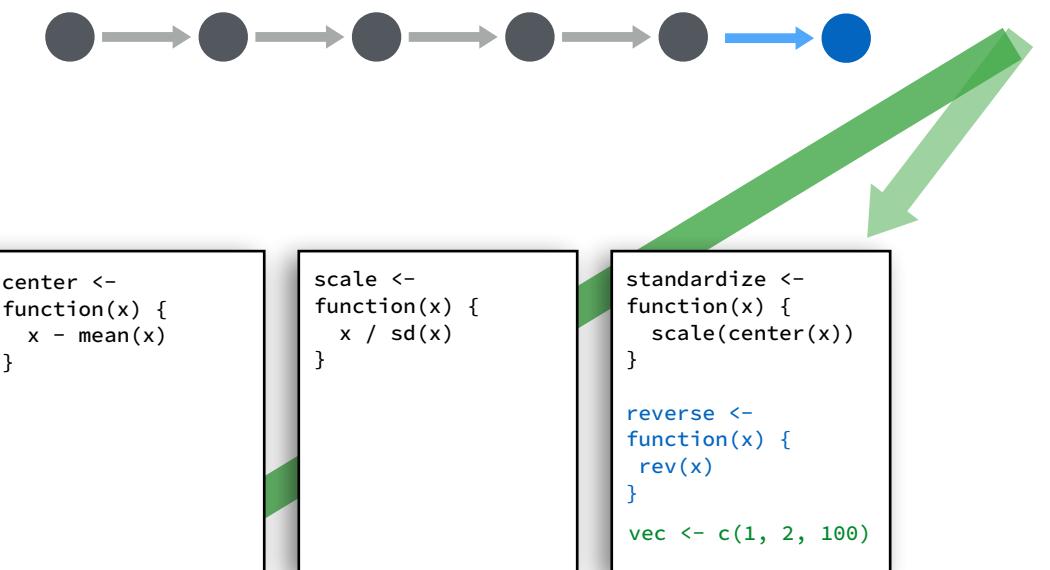


push
add + commit





GitHub



3. Github Version

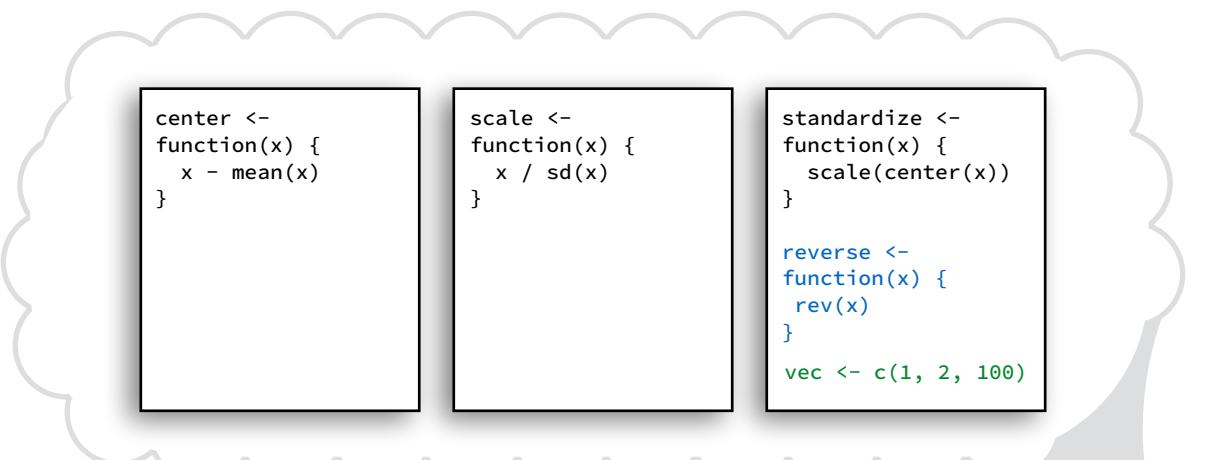
available to public

2."Official" Version

implied by commit history

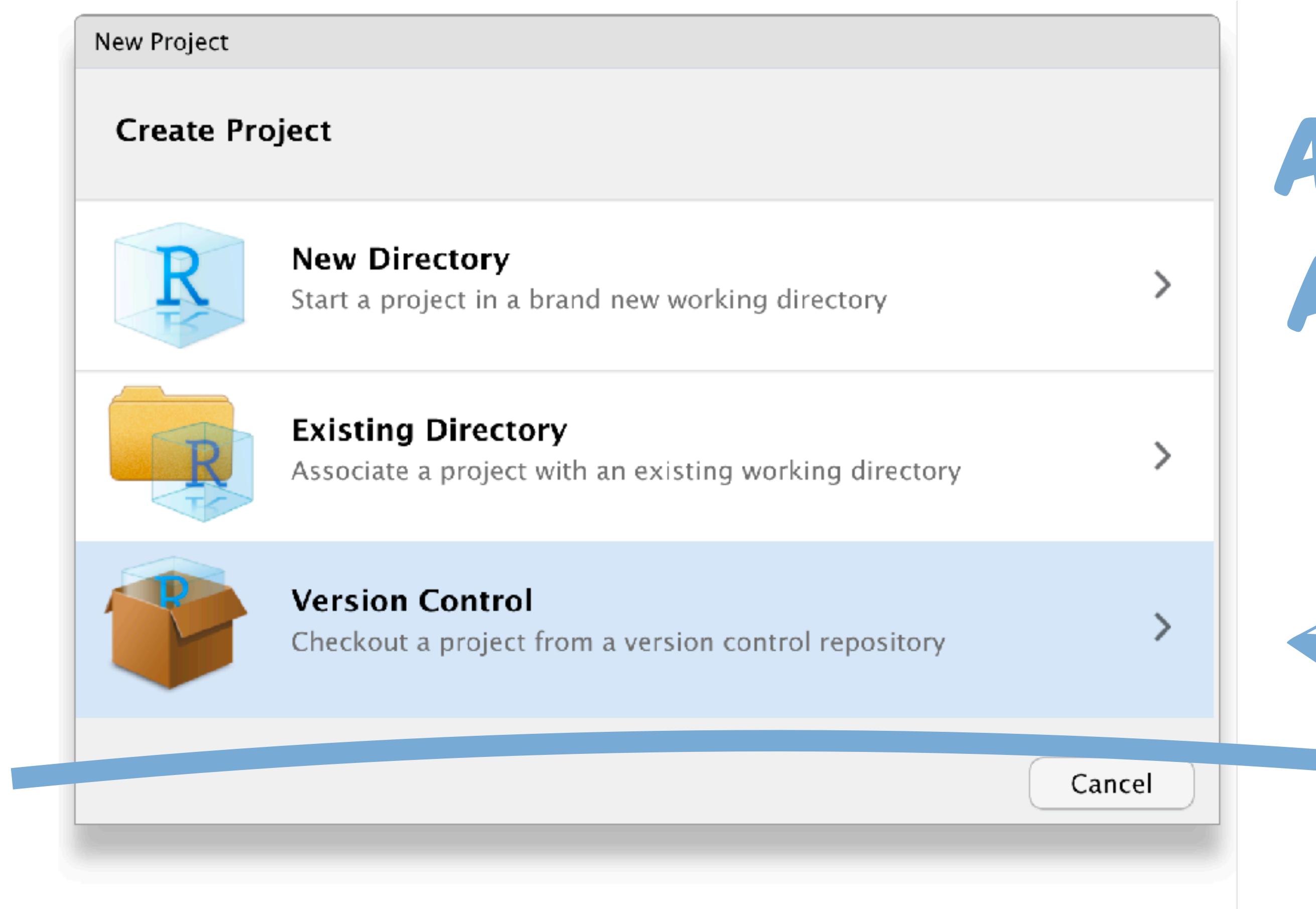
1. Real Life Version

in your working directory

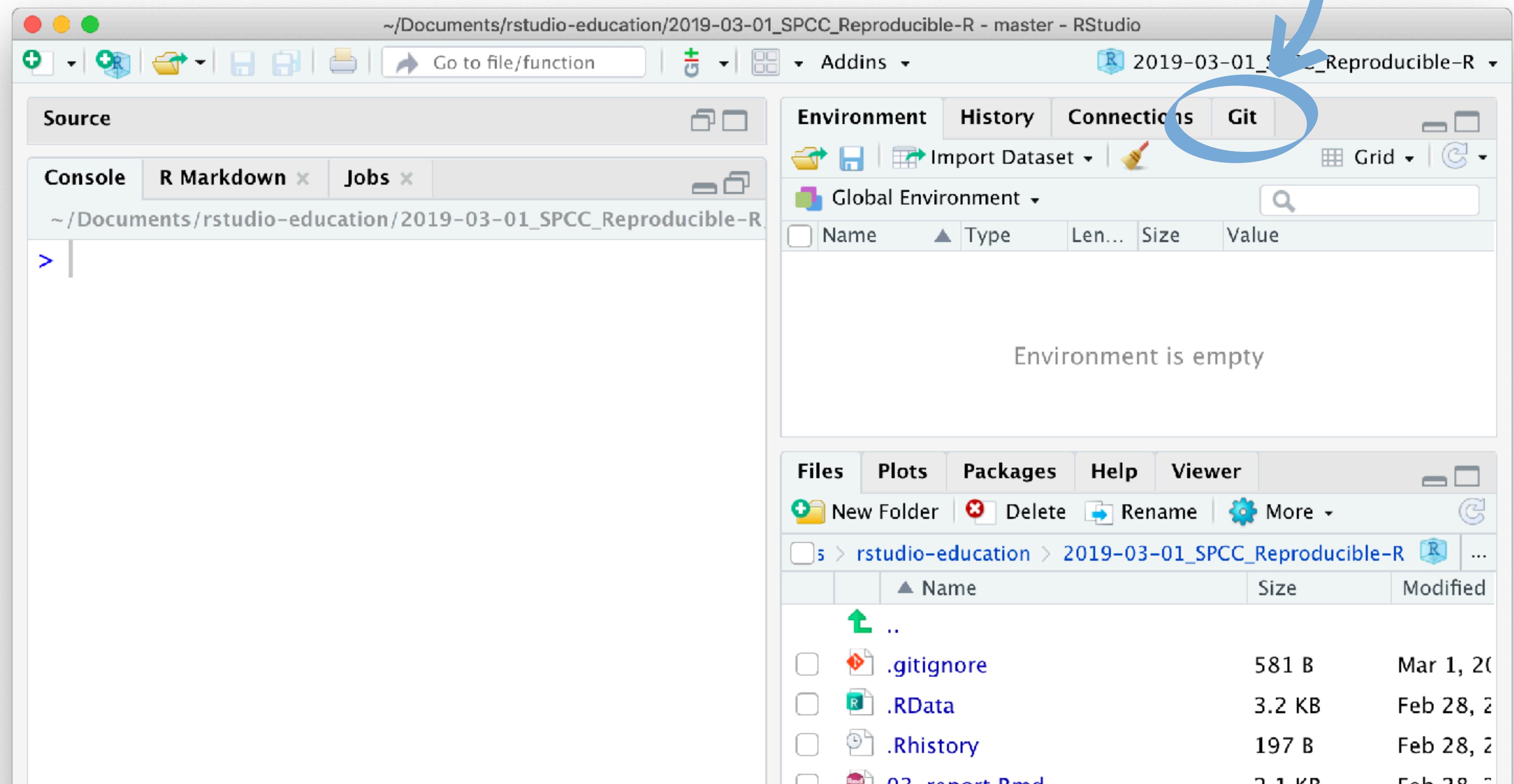


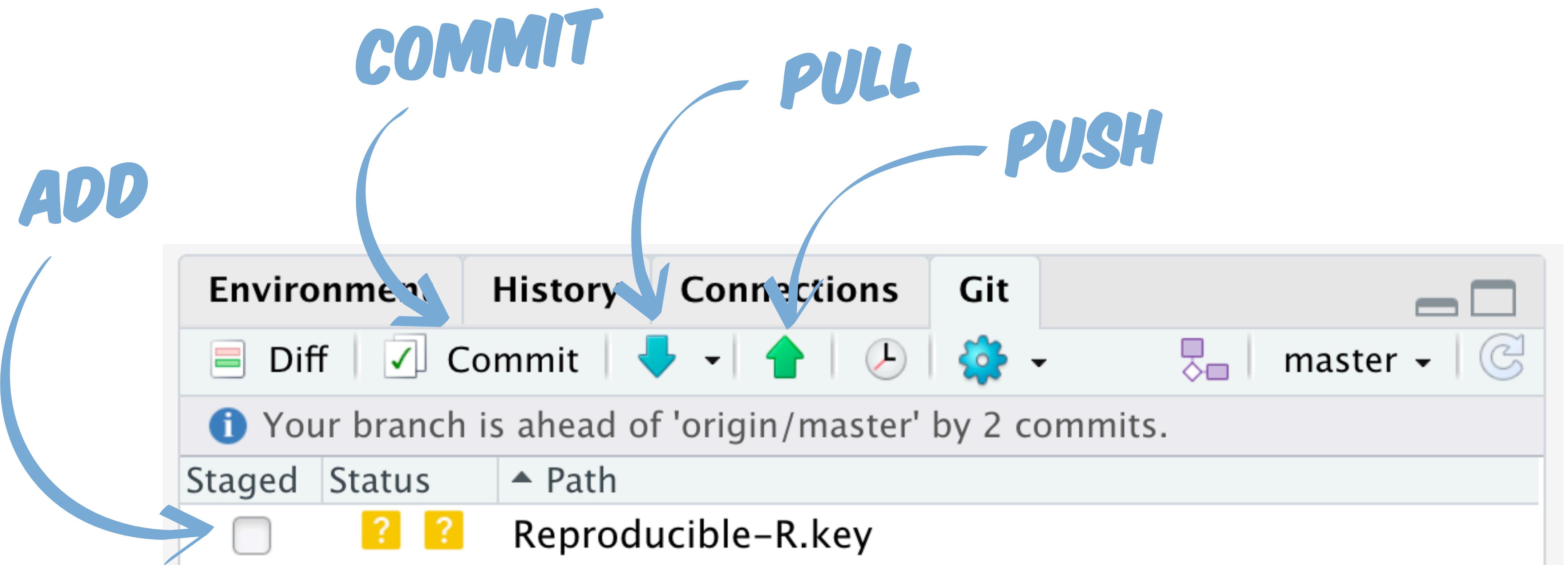
You

DOWNLOAD A REPOSITORY AS A PROJECT



A GIT PANE





R Markdown

Takeaway







— 750mL?





Dublin - Google Sheets

Secure | https://docs.google.com/spreadsheets/d/1OrHXGZJlyE_xyrbYs4YBEIF...

Dublin

File Edit View Insert Format Data

SHARE

bottle

| | A | B | C | D | E |
|----|--------|--------|---|---|---|
| 1 | bottle | amount | | | |
| 2 | 1 | 748.7 | | | |
| 3 | 2 | 746.4 | | | |
| 4 | 3 | 750.1 | | | |
| 5 | 4 | 751.9 | | | |
| 6 | 5 | 747.6 | | | |
| 7 | 6 | 748.8 | | | |
| 8 | 7 | 748.6 | | | |
| 9 | 8 | 752.2 | | | |
| 10 | 9 | 749.9 | | | |
| 11 | 10 | 745.9 | | | |
| 12 | 11 | 747.1 | | | |
| 13 | 12 | 748.3 | | | |
| 14 | 13 | 750.4 | | | |
| 15 | 14 | 749.6 | | | |
| 16 | 15 | 746 | | | |
| 17 | 16 | 750.5 | | | |
| 18 | 17 | 750.5 | | | |
| 19 | 18 | 750.1 | | | |
| 20 | 19 | 743.9 | | | |
| 21 | 20 | 750.6 | | | |



~Dropbox (RStudio)/RStudio/training/Z-RStudio-Instructor-Toolchain/related-talks/EARL-2018-RMarkdown - RStudio

Go to file/function Addins EARL-2018-RMarkdown

rmarkdown-demo.Rmd*

Knit Insert Run

```
1 ---  
2 title: 'Re: Bottle Machine'  
3 output: html_document  
4 ---  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27
```

14:1 (Top Level) R Markdown

Files Plots Packages Help Viewer



~Dropbox (RStudio)/RStudio/training/Z-RStudio-Instructor-Toolchain/related-talks/EARL-2018-RMarkdown - RStudio

rmarkdown-demo.Rmd*

Go to file/function Addins EARL-2018-RMarkdown

Knit Insert Run

```
1 ---  
2 title: 'Re: Bottle Machine'  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE, message=FALSE}  
7 library(googlesheets)  
8 library(tidyverse)  
9  
10 key <- "10rHXGZJIyE_xyrbYs4YBE1FUt_mEYZE7n1rK5HuoR4w"  
11 samples <- gs_read(gs_key(key), lookup = FALSE)  
12 obs <- mean(samples$amount)  
13 ```  
14  
15 ```{r echo = FALSE}  
16 samples$amount  
17 ```  
18  
19 ```{r echo = FALSE}  
20 sim <- map_dbl(1:1000,  
21 ~mean(rnorm(50, mean = 750, sd = 3)))  
22 delta <- abs(750 - obs)  
23 pvalue <- mean(abs(750 - sim) >= delta)  
24 ```  
25  
26 ```{r echo = FALSE, fig.height = 2}  
27 ggplot() +
```

14:1 (Top Level) R Markdown

Files Plots Packages Help Viewer



A screenshot of the RStudio interface showing an R Markdown file named "rmarkdown-demo.Rmd". The code editor displays the following R code:

```
26  ```{r echo = FALSE, fig.height = 2}
27  ggplot() +
28    geom_histogram(aes(x = sim), bins = 30) +
29    geom_rect(aes(xmin = -Inf, xmax = 750 - delta,
30                  ymin = -Inf, ymax = Inf),
31              fill = "#4197D9", alpha = 0.2) +
32    geom_rect(aes(xmin = 750 + delta, xmax = Inf,
33                  ymin = -Inf, ymax = Inf),
34              fill = "#4197D9", alpha = 0.2) +
35    geom_vline(aes(xintercept = obs),
36               color = "#4197D9", size = 3)
37 ```



The code uses ggplot2 to create a histogram with two semi-transparent blue rectangular overlays highlighting specific ranges on the x-axis. It also includes a vertical line at a specific x-intercept value.


```



A screenshot of the RStudio interface showing an R Markdown file named "rmarkdown-demo.Rmd". The code editor displays the following R code:

```
10 key <- "10rHXGZJIyE_xyrbYs4YBE1FUtmESYZE7n1rK5HuoR4w"
11 samples <- gs_read(gs_key(key), lookup = FALSE)
12 obs <- mean(samples$amount)
13 ``
14
15 Does our bottle machine fill each bottle with 750 mL
16 of beer (on average)?
17
18
19
20
21 ````{r echo = FALSE}
```

Does our bottle machine fill each bottle with 750 mL
of beer (on average)?



A screenshot of the RStudio interface showing an R Markdown file named "rmarkdown-demo.Rmd". The code in the editor is as follows:

```
10 key <- "10rHXGZJIyE_xyrbYs4YBE1FUtmESYZE7n1rK5HuoR4w"
11 samples <- gs_read(gs_key(key), lookup = FALSE)
12 obs <- mean(samples$amount)
13 ``
14
15 ## Data
16 # Does the bottle machine fill each bottle with 750 mL
17 # of beer (on average)?
18 I selected 50 bottles at random from the Dublin
19 factory, which contained the following amounts of
20 beer (mL):
21 ````{r echo = FALSE}
```

Data

I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (mL):



The mean amount is `r obs`.

```
13
14
15 Does our bottle machine fill each bottle with 750 mL
16
17 ## Data
18
19 I selected 50 bottles at random from the Dublin
20
21 ````{r echo = FALSE}
22 samples$amount
23 ````
```

The mean amount is `r obs`.



A screenshot of the RStudio interface showing an R Markdown file named "rmarkdown-demo.Rmd". The code editor displays the following content:

```
[34] 748.6 744.7 750.6 748.9 753.2 747.8 746.0  
748.4 754.2 750.6 749.9  
[45] 749.4 754.4 750.7 750.3 745.9 747.1  
  
24  
25 ## Reasoning amount is **`r obs`**.  
26  
27 The amounts of beer in our bottles should be normally  
28 distributed with a mean of 750 ml and a standard  
29 deviation of 3 mL.  
  
30
```

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 ml and a standard deviation of 3 mL.



~Dropbox (RStudio)/RStudio/training/Z-RStudio-Instructor-Toolchain/related-talks/EARL-2018-RMarkdown - RStudio

rmarkdown-demo.Rmd*

[45] 749.4 754.4 750.7 750.3 745.9 747.1

24
25 The mean amount is `r obs`.
26
27 **## Reasoning**
28
29 The amounts of beer in our bottles should be normally distributed with a mean of 750 ml and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.
30
31 ````{r echo = FALSE}`
32 `sim <- map dbl(1:1000.`

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 ml and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



```
~Dropbox (RStudio)/RStudio/training/Z-RStudio-Instructor-Toolchain/related-talks/EARL-2018-RMarkdown - RStudio
rmarkdown-demo.Rmd* Go to file/function Addins EARL-2018-RMarkdown
[45] 749.4 754.4 750.7 750.3 745.9 747.1
24
25 The mean amount is **`r obs`**.
26
27 I've plotted the distribution of the simulated means
28 above. The blue line shows our observed sample mean.
29 Notice that `r paste0(pvalue * 100, "%")` of the
simulations produced a sample mean as extreme as
ours (p-value = `r pvalue`). In other words,
our observations are in accordance with our belief.
30
31 ```{r echo = FALSE}
32 sim <- map dbl(1:1000).
```

I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that `r paste0(pvalue * 100, "%")` of the simulations produced a sample mean as extreme as ours (p-value = `r pvalue`). In other words, our observations are in accordance with our belief.



A screenshot of the RStudio interface showing an R Markdown file named "rmarkdown-demo.Rmd". The code editor displays the following R code:

```
50
51 ## We've plotted the distribution of the simulated means
52 # above. The blue line shows our observed sample mean.
53 # Notice that **r** has a p-value of 100, "%".
54 # This means there is no evidence that the bottle machine
55 # is malfunctioning.
```

The RStudio interface includes a toolbar at the top with various icons for file operations, a "Knit" button, and a "Run" button. Below the toolbar is a plot window showing a histogram of simulated means with a blue vertical line indicating the observed sample mean. The x-axis is labeled "sim" and ranges from 749 to 751.

Conclusion

We do ***not*** have evidence that the bottle machine is malfunctioning.



~Dropbox (RStudio)/RStudio/training/Z-RStudio-Instructor-Toolchain/related-talks/EARL-2018-RMarkdown - RStudio

rmarkdown-demo.Rmd*

Go to file/function Addins EARL-2018-RMarkdown

Knit Insert Run

sim

50
51 I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that `r paste0(pvalue * 100, "%")` of the simulations produced a sample mean as extreme as ours (p-value = `r pvalue`). In other words, our observations are in accordance with our belief.
52
53 **## Conclusion**
54
55 We do ****not**** have evidence that the bottle machine is malfunctioning.
56

14:1 (Top Level) R Markdown

Files Plots Packages Help Viewer

~Dropbox (RStudio)/RStudio/training/Z-RStudio-Instructor-Toolchain/related-talks/EARL-2018-RMarkdown - RStudio

RStudio: View PDF

1 of 1 Automatic Zoom

Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

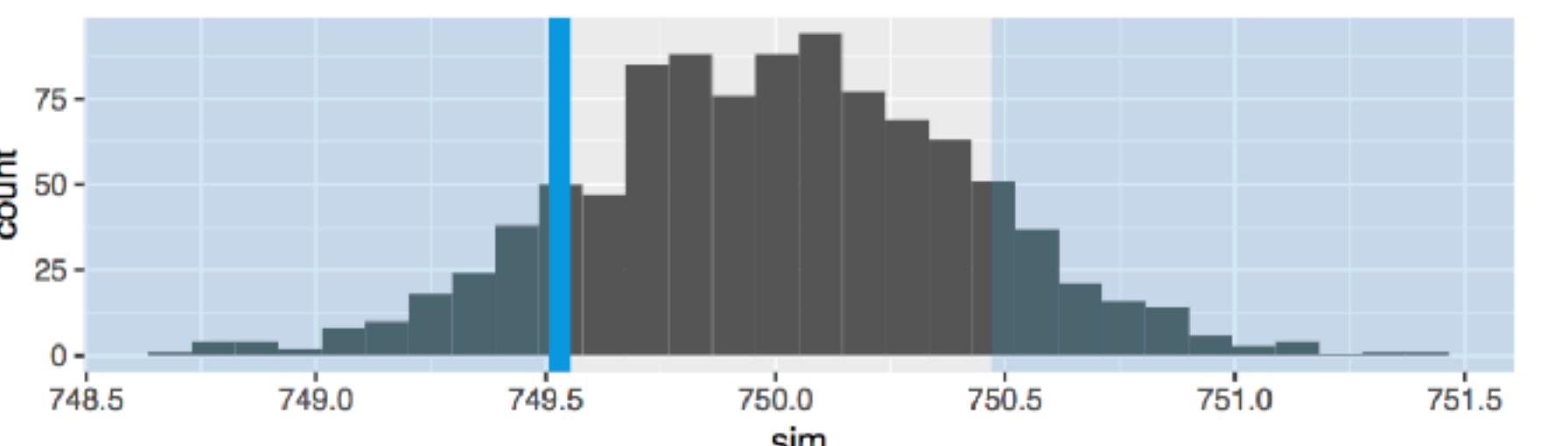
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (mL):

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 748.6 752.2 749.9 745.9 747.1
## [12] 748.3 750.4 749.6 746.0 750.5 750.5 750.1 743.9 750.6 758.0 746.3
## [23] 752.5 751.6 748.1 751.8 744.6 751.0 751.7 753.7 749.4 752.1 747.9
## [34] 748.6 744.7 750.6 748.9 753.2 747.8 746.0 748.4 754.2 750.6 749.9
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is **749.53**.

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 mL and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 26.2% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.262$). In other words, our observations are in accordance with our belief.

Conclusion

We do **not** have evidence that the bottle machine is malfunctioning.

Files Plots Packages Help Viewer

The screenshot shows a dual-pane interface in RStudio. The left pane displays a presentation slide titled "Re: Bottle Machine" with a numbered list from 1 to 7. The right pane shows the R Markdown editor with code and a preview.

Left Pane (Presentation View):

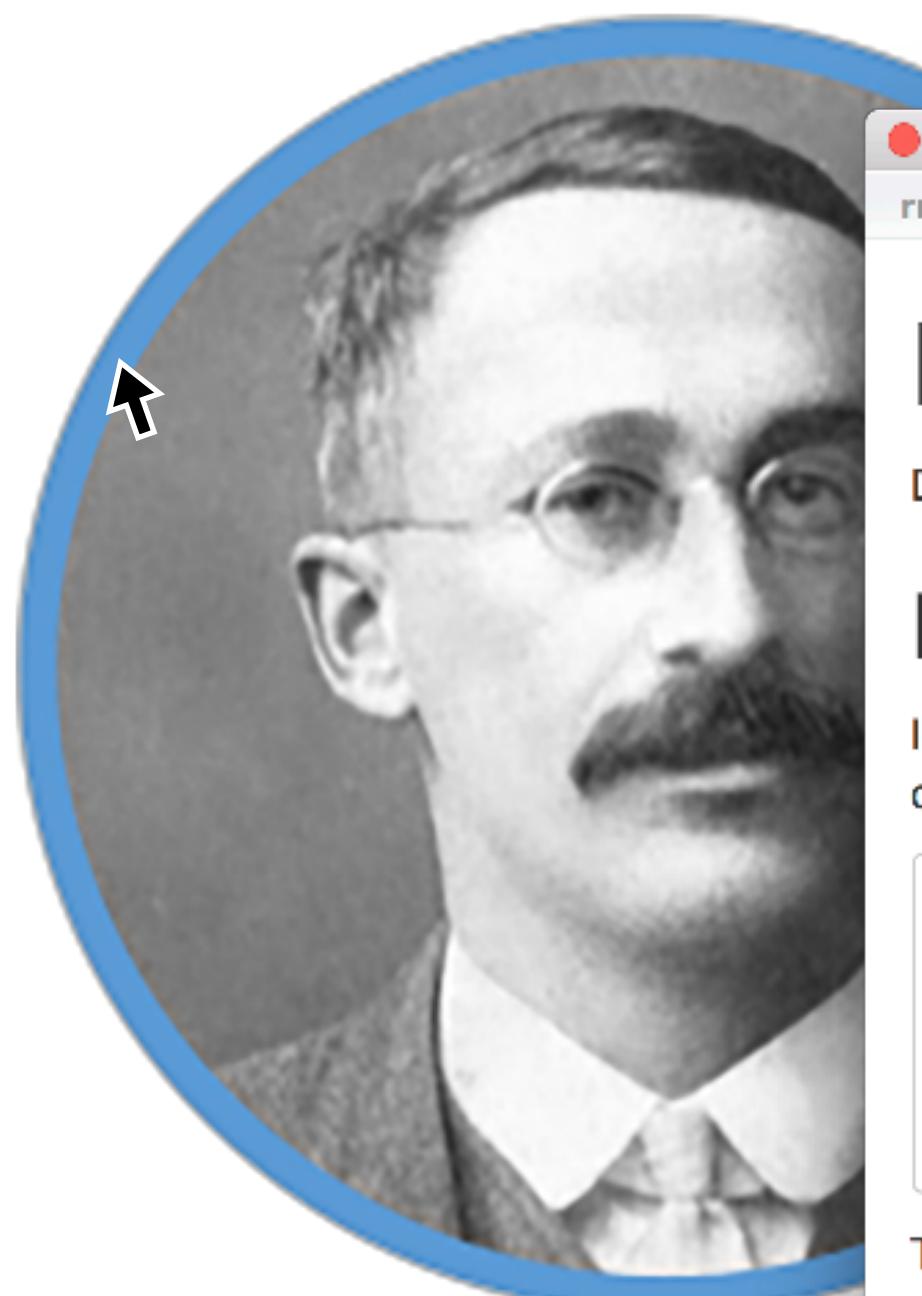
- Slide 1: Re: Bottle Machine
- Slide 2: Data
- Slide 3: Reasoning
- Slide 4: Histogram
- Slide 5: Distribution Plot
- Slide 6: Summary
- Slide 7: Conclusion

Right Pane (R Markdown Editor):

```
21: ````{r echo = FALSE}
22: samples$amount
23: ````
```

14:1 (Top Level) R Markdown

Files Plots Packages Help Viewer



Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

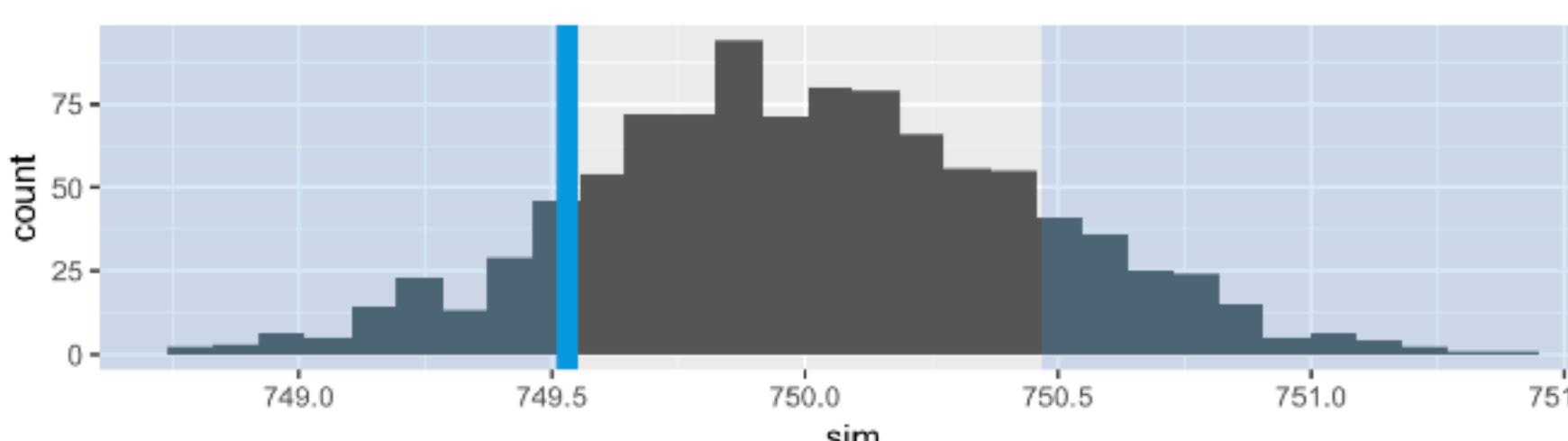
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (mL):

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 748.6 752.2 749.9 745.9 747.1  
## [12] 748.3 750.4 749.6 746.0 750.5 750.5 750.1 743.9 750.6 758.0 746.3  
## [23] 752.5 751.6 748.1 751.8 744.6 751.0 751.7 753.7 749.4 752.1 747.9  
## [34] 748.6 744.7 750.6 748.9 753.2 747.8 746.0 748.4 754.2 750.6 749.9  
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is **749.53**.

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 mL and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 28.3% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.283$). In other words, our observations are in accordance with our belief.

Conclusion

We do **not** have evidence that the bottle machine is malfunctioning.

age=FALSE}

FUtmESYZE7n1rK5HuoR4w"
lookup = FALSE)

ch bottle with 750 mL

from the Dublin
llowing amounts of

R Markdown

RStudio Connect

Secure | https://connect.rstudioservices.com/connect/#/apps/149/access/105

Content / rmarkdown-demo

Garrett

WG bill

Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

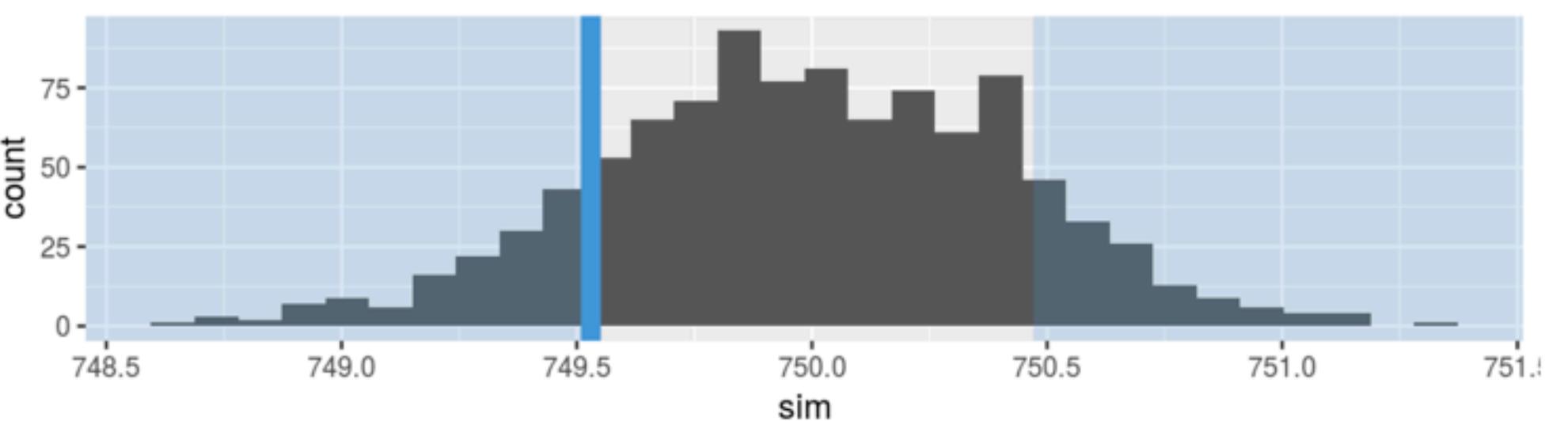
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (mL):

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 748.6 752.2 749.9 745.9 747.1  
## [12] 748.3 750.4 749.6 746.0 750.5 750.5 750.1 743.9 750.6 758.0 746.3  
## [23] 752.5 751.6 748.1 751.8 744.6 751.0 751.7 753.7 749.4 752.1 747.9  
## [34] 748.6 744.7 750.6 748.9 753.2 747.8 746.0 748.4 754.2 750.6 749.9  
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is **749.53**.

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 mL and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 27.4% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.274$). In other words, our observations are in accordance with our belief.

Conclusion

We do **not** have evidence that the bottle machine is malfunctioning.

Info Access Runtime Schedule Tags Vars Logs

Who can view this document

You

Who can change this document

WG William Gosset bill

Add collaborator

Who runs this document on the server

The default user

Administrators can create a custom "vanity" url to access this document. Your custom URL will be appended to your domain to form the complete path to your document.

Your custom url:

https://connect.rstudioservices.com/garrett/rmarkdown-demo

RStudio Connect

Secure | https://connect.rstudioservices.com/connect/#/apps/149/access/105

Content / rmarkdown-demo

Your email request has been submitted.

Email the report to:

- just me
- collaborators
- collaborators & viewers

Send

Does our bottle machine fill each bottle with 750 mL of beer

Data

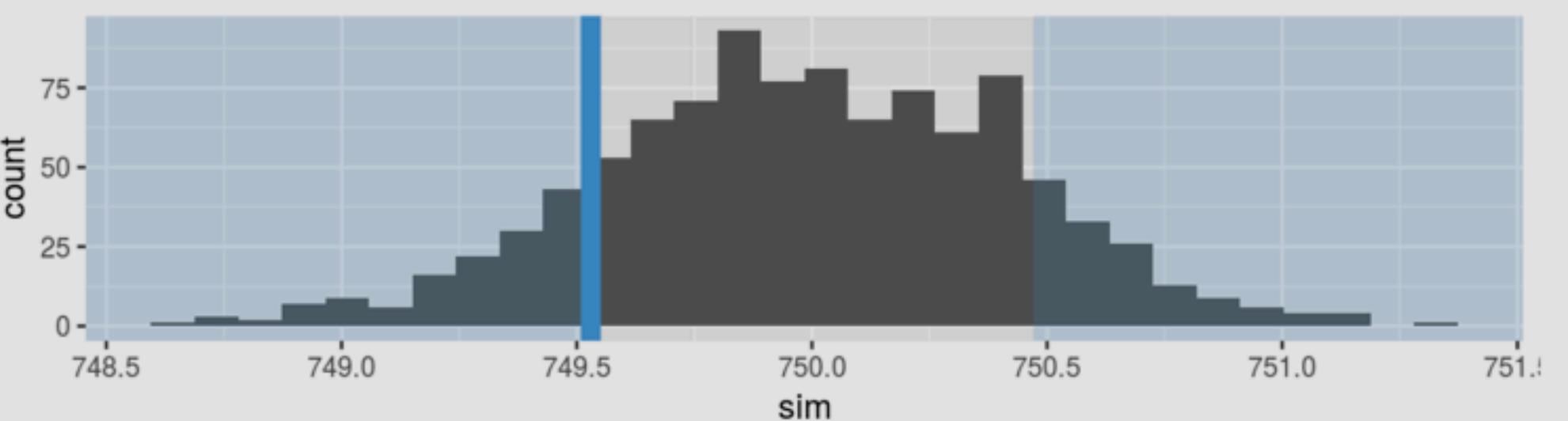
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (in mL):

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 748  
## [12] 748.3 750.4 749.6 746.0 750.5 750.5 750  
## [23] 752.5 751.6 748.1 751.8 744.6 751.0 751  
## [34] 748.6 744.7 750.6 748.9 753.2 747.8 746  
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is **749.53**.

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 mL and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 27.4% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.274$). In other words, our observations are in accordance with our belief.

Conclusion

We do **not** have evidence that the bottle machine is malfunctioning.

Garrett

WG bill

Info Access Runtime Schedule Tags Vars Logs

Who can view this document

Collaborators & you

WG William Gosset bill

VP VP Virginia virginia

Add collaborator

Who runs this document on the server

The default user

Administrators can create a custom "vanity" url to access this document. Your custom URL will be appended to your domain to form the complete path to your document.

Your custom url:

https://connect.rstudioservices.com/garrett/rmarkdown-demo

One more thing

**Reproducibility
is an opportunity**

RStudio Connect X

Secure | https://connect.rstudioservices.com/connect/#/apps/149/access/105

Content / rmarkdown-demo

Garrett

WG bill

Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

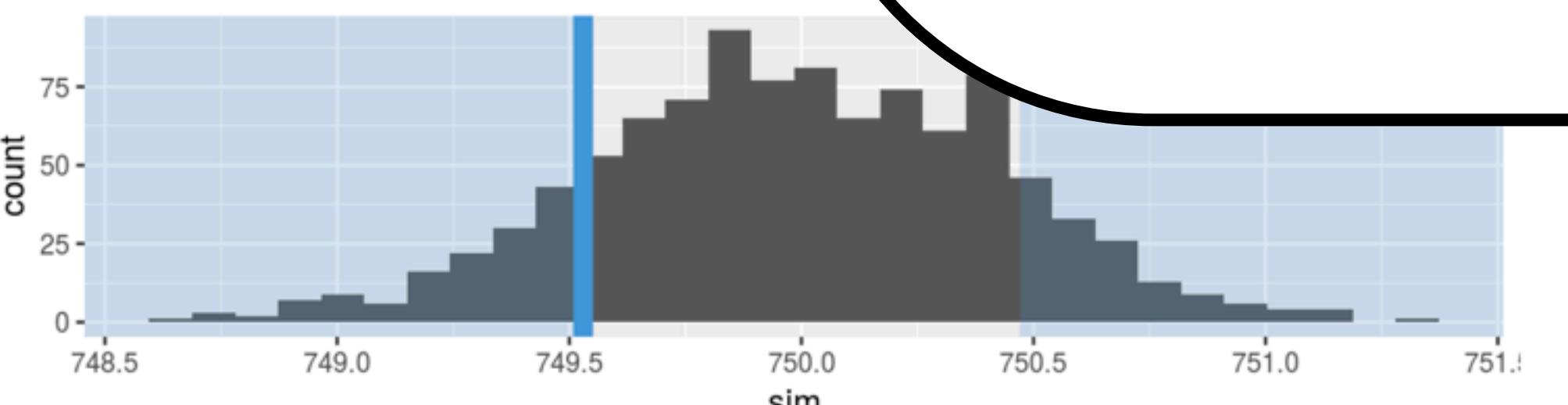
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (mL):

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 748.6 750.1  
## [12] 748.3 750.4 749.6 746.0 750.5 750.5 750.1  
## [23] 752.5 751.6 748.1 751.8 744.6 751.0 751.1  
## [34] 748.6 744.7 750.6 748.9 753.2 747.8 749.1  
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is 749.53.

Reasoning

The amounts of beer in our bottles should be normally distributed. We can use a simulation to calculate the sample means of 1000 samples of size 50.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 27.4% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.274$). In other words, our observations are in accordance with our belief.

Conclusion

We do **not** have evidence that the bottle machine is malfunctioning.

Who can view this document

Collaborators & you

can change this document

William Gosset bill

VP Virginia virginia

Add collaborator

Your custom url:

https://connect.rstudioservices.com/garrett/rmarkdown-demo

A large white speech bubble with a black outline is centered over the histogram. Inside the bubble, the text "Can you do this every week?" is written in a large, bold, black font. A green arrow points from the bottom right of the speech bubble towards a small photo of a woman on the right side of the page. The photo shows a woman with blonde hair, wearing a grey blazer, holding a phone to her ear. A green circle highlights the area around the photo and the bottom right corner of the slide.

RStudio Cloud

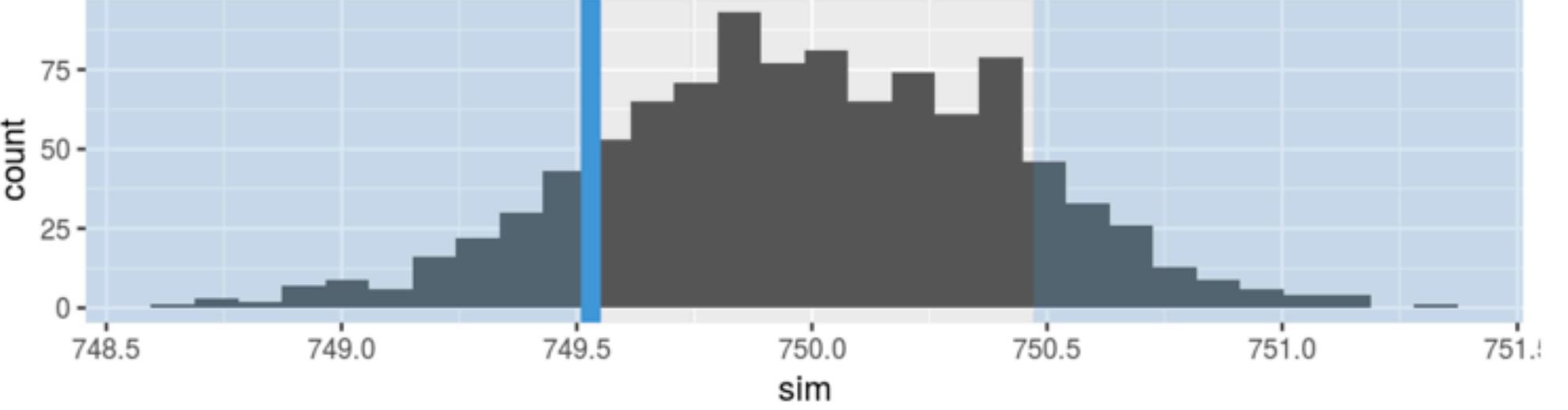
garrett@garrett-macbook-pro:~/Documents/teaching/statistics/bottle-machine

Schedule saved successfully.

The mean amount is 749.53.

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 mL and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 27.4% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.274$). In other words, our observations are in accordance with our belief.

Conclusion

We do **not** have evidence that the bottle machine is malfunctioning.

Schedule saved successfully.

WG bill

Schedule output for default

Fri Sep 07 2018 12:39:20 GMT-0500

Set to Now

Weekly

Run every 1 week.

Monday

Publish output after it is generated

Send email after update

RStudio Connect Garrett

Secure | https://connect.rstudioservices.com/connect/#/apps/149/access/105

Content / rmarkdown-demo

Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

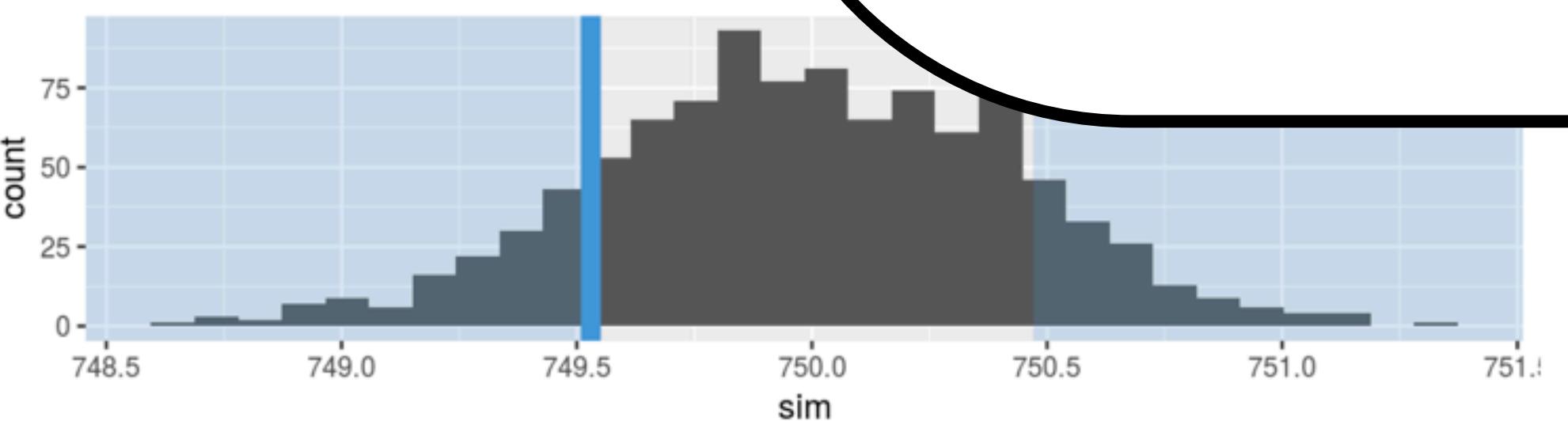
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer:

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 749.1  
## [12] 748.3 750.4 749.6 746.0 750.5 750.5  
## [23] 752.5 751.6 748.1 751.8 744.6 751.1  
## [34] 748.6 744.7 750.6 748.9 753.2 747.1  
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is 749.53.

Reasoning

The amounts of beer in our bottles should be normally distributed. We can use a simulation to calculate the sample means of 1000 samples:



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 27.4% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.274$). In other words, our observations are in accordance with our belief.

Conclusion

We do **not** have evidence that the bottle machine is malfunctioning.

Schedule output for default

Start date & time: Fri Sep 07 2018 12:39:20 GMT-0500

Set to Now

Run every... week

Sunday Monday

Send email after update





```
params:  
  factory:  
    value: Dublin  
    choices:  
      - Dublin  
      - London
```

```
1 ---  
2 title: 'Re: Bottle Machine'  
3 output: html_document  
4 params:  
5   factory:  
6   ```{r setup, include=FALSE, message=FALSE}  
7     library(googlesheets)  
8       value: Dublin  
9       choices:  
10      - Dublin  
11      - London  
12  
13 key <- "10rHXGZJIyE_xyrbYs4YBE1FUtmESYZE7n1rK5Hu0R4w"  
14 samples <- gs_read(gs_key(key), lookup = FALSE)  
15 obs <- mean(samples$amount)  
16  
17 ````
```

14:1 (Top Level) R Markdown

Files Plots Packages Help Viewer

RStudio Connect Garrett

Secure <https://connect.rstudiосervices.com/connect/#/apps/150/access/106>

Content / rmarkdown-demo

Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

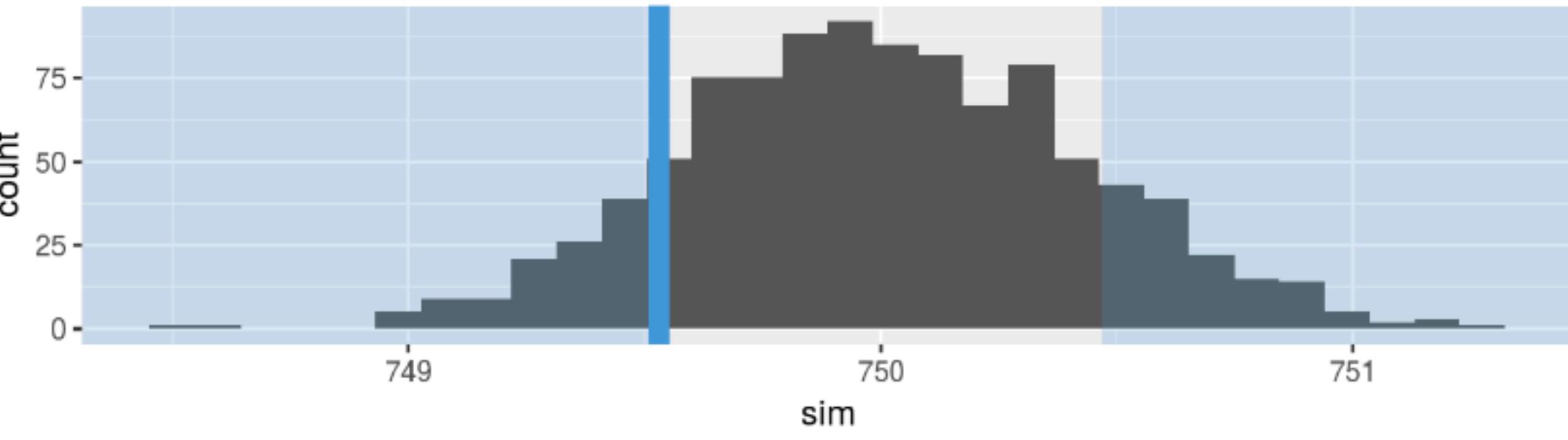
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (mL):

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 748.6 752.2 749.9 745.9 747.1  
## [12] 748.3 750.4 749.6 746.0 750.5 750.5 750.1 743.9 750.6 758.0 746.3  
## [23] 752.5 751.6 748.1 751.8 744.6 751.0 751.7 753.7 749.4 752.1 747.9  
## [34] 748.6 744.7 750.6 748.9 753.2 747.8 746.0 748.4 754.2 750.6 749.9  
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is **749.53**.

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 mL and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 26% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0.026$).

Who can view this document

You

Who can change this document

WG William Gosset bill

Add collaborator

Who runs this document on the server

The default user

Administrators can create a custom "vanity" url to access this document. Your custom URL will be appended to your domain to form the complete path to your document.

Your custom url:

<https://connect.rstudiосervices.com/garrett/rmarkdown-demo>

RStudio Connect Garrett

Secure <https://connect.rstudioservices.com/connect/#/apps/150/access/106>

Content / rmarkdown-demo

Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

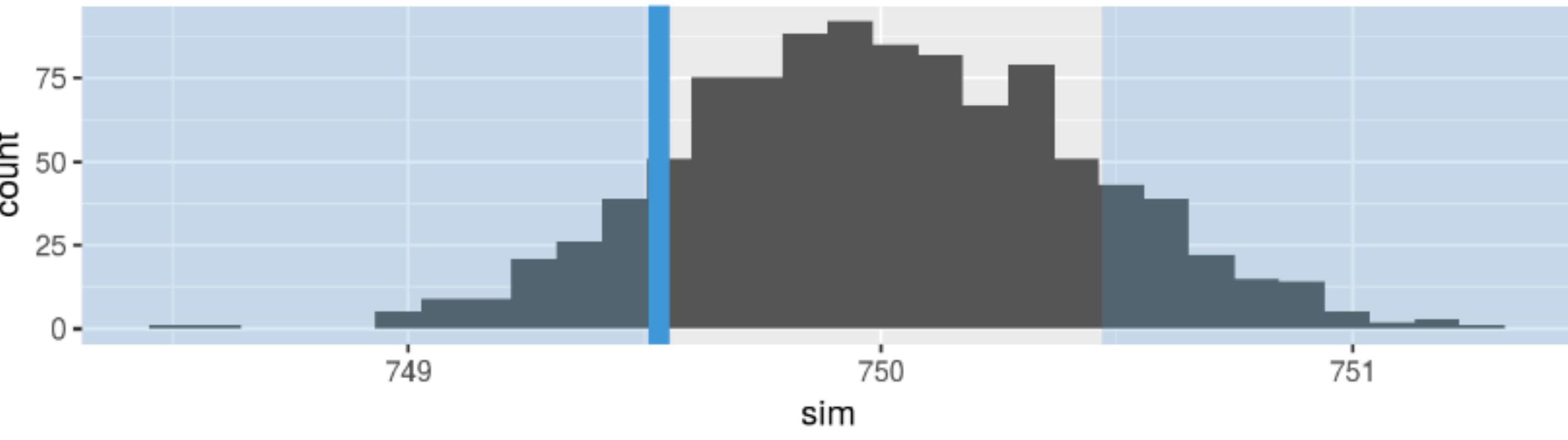
I selected 50 bottles at random from the Dublin factory, which contained the following amounts of beer (mL):

```
## [1] 748.7 746.4 750.1 751.9 747.6 748.8 748.6 752.2 749.9 745.9 747.1  
## [12] 748.3 750.4 749.6 746.0 750.5 750.5 750.1 743.9 750.6 758.0 746.3  
## [23] 752.5 751.6 748.1 751.8 744.6 751.0 751.7 753.7 749.4 752.1 747.9  
## [34] 748.6 744.7 750.6 748.9 753.2 747.8 746.0 748.4 754.2 750.6 749.9  
## [45] 749.4 754.4 750.7 750.3 745.9 747.1
```

The mean amount is 749.53.

Reasoning

The amounts of beer in our bottles should be normally distributed with a mean of 750 mL and a standard deviation of 3 mL. Let's use simulation to calculate the sample means of 1000 samples of size 50 generated from such a distribution.



I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 26% of the simulations produced a sample mean as extreme as ours ($p\text{-value} =$

Who can view this document
You

Who can change this document
WG William Gosset bill

Add collaborator

Who runs this document on the server



RStudio Connect Garrett

Secure <https://connect.rstudioservices.com/connect/#/apps/150/access/106>

Content / rmarkdown-demo

Re: Bottle Mac | Re: Bottle Machine

Does our bottle machine fill each bottle with 750 mL of beer (on average)?

Data

I selected 50 bottles at random from the distribution and measured the following amounts of beer (mL):

```
## [1] 748.7 746.4 750.1 751.9 749.8 750.5 747.7 747.4 751.6 748.1
## [12] 748.3 750.4 749.6 746.8 750.2 748.5 750.8 749.9 750.0 749.5
## [23] 752.5 751.6 748.1 751.3 749.7 750.6 748.9 750.7 749.2 751.4
## [34] 748.6 744.7 750.6 748.3 750.9 749.4 754.4 750.7 750.1 749.8
## [45] 749.4 754.4 750.7 750.5 749.1 750.3 748.8 750.0 749.6 751.2
```

The mean amount is 749.53.

Reasoning

The amounts of beer in our bottles should follow a normal distribution with a standard deviation of 3 mL. Let's use simulation to generate 1000 sample means from such a distribution.

I like it!

The distribution of simulated sample means is centered around 750 mL with a standard deviation of approximately 0.3 mL. The observed sample mean of 749.53 mL is located at the far left of the distribution, corresponding to the blue vertical line.

I've plotted the distribution of the simulated means above. The blue line shows our observed sample mean. Notice that 0% of the simulations produced a sample mean as extreme as ours ($p\text{-value} = 0$). In other words, our sample mean is statistically significant.

count

sim

WG bill

**Reproducibility
is an opportunity**

schedule
parameterize
automate



What should we do
with the leftover beer?



Thank You

