

# 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

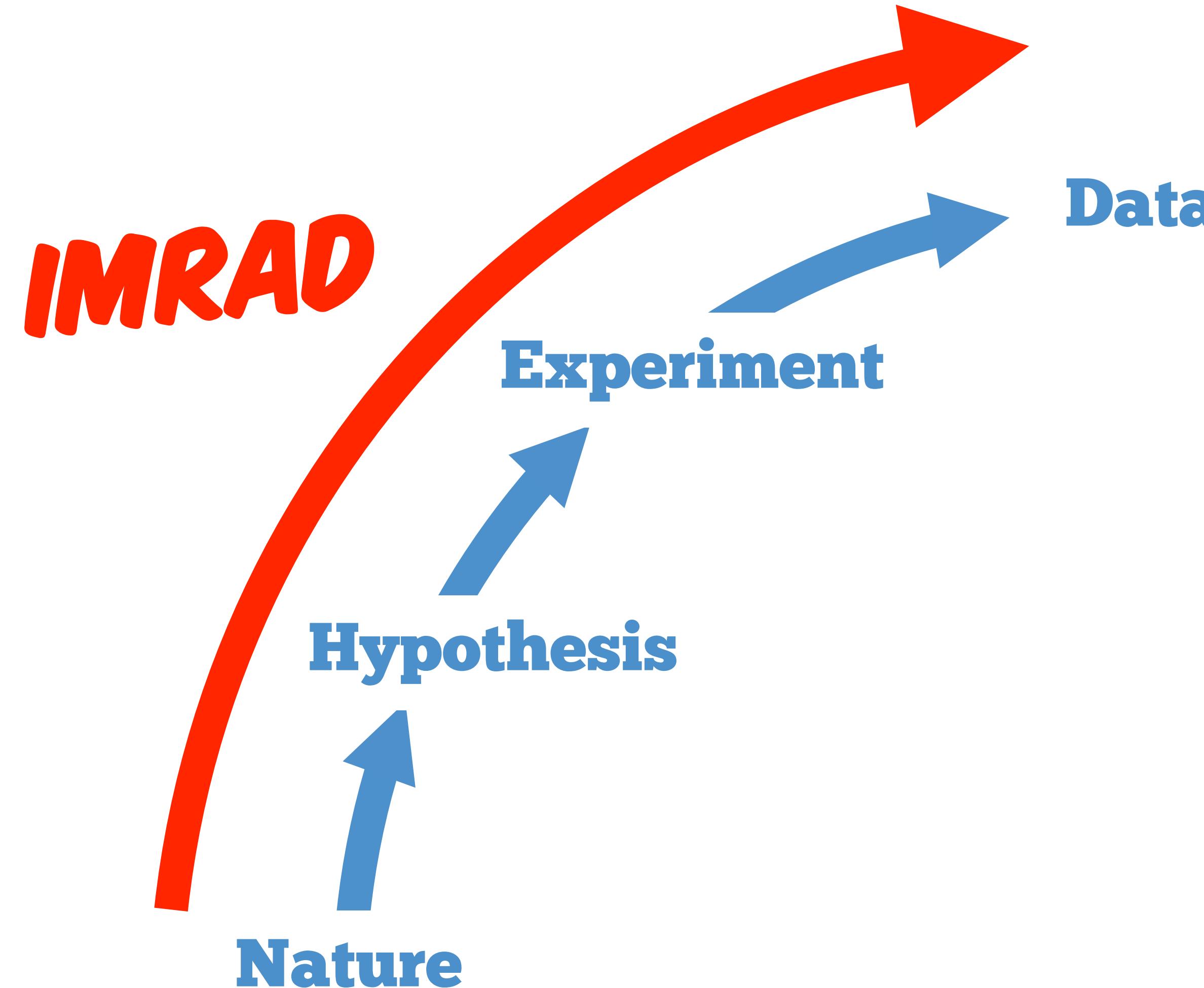
prove



**CREATE MAPS.  
NOT PROOFS**

**IMRAD**

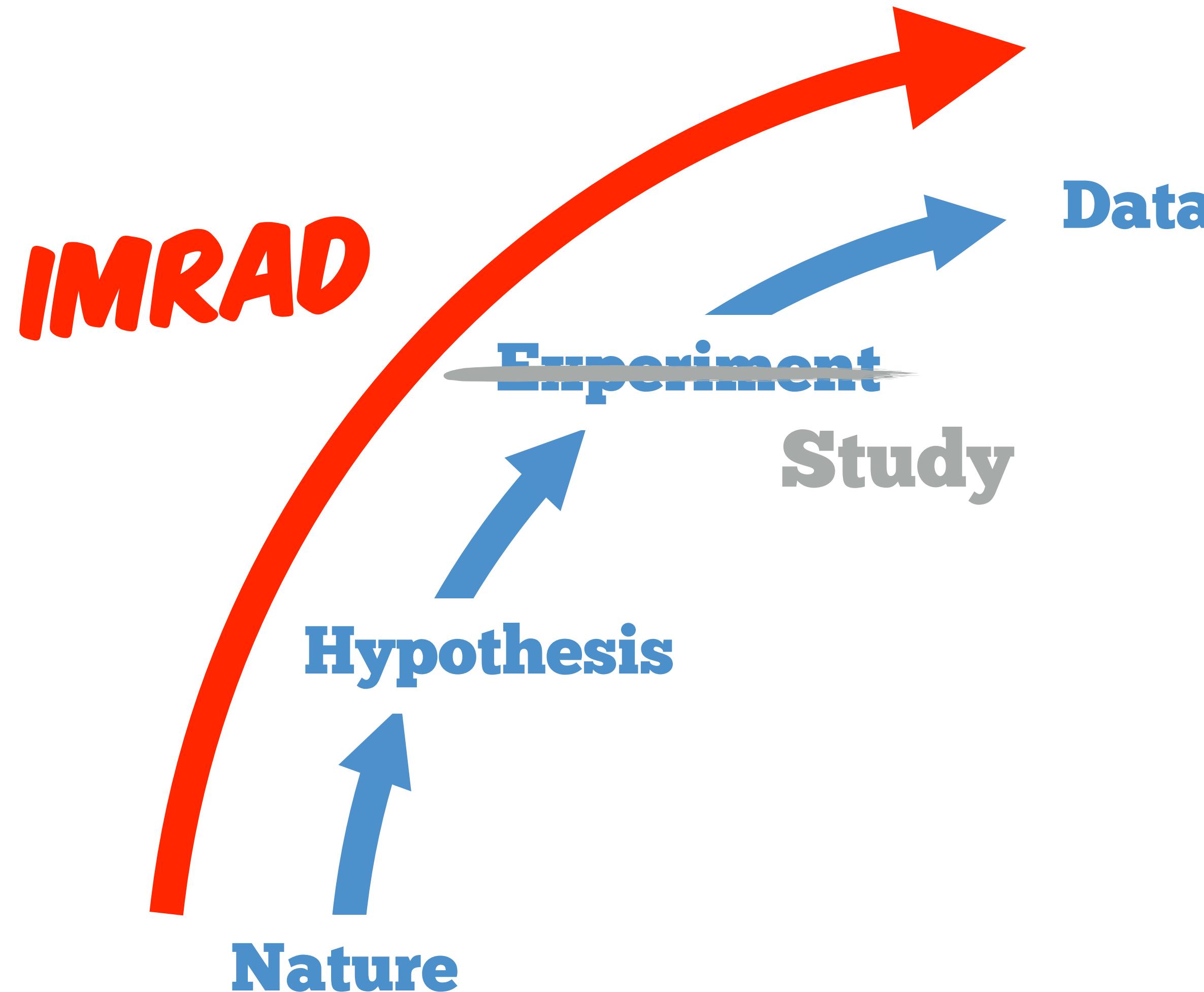
<https://rstudio.cloud/project/233945>



But where's  
the math?

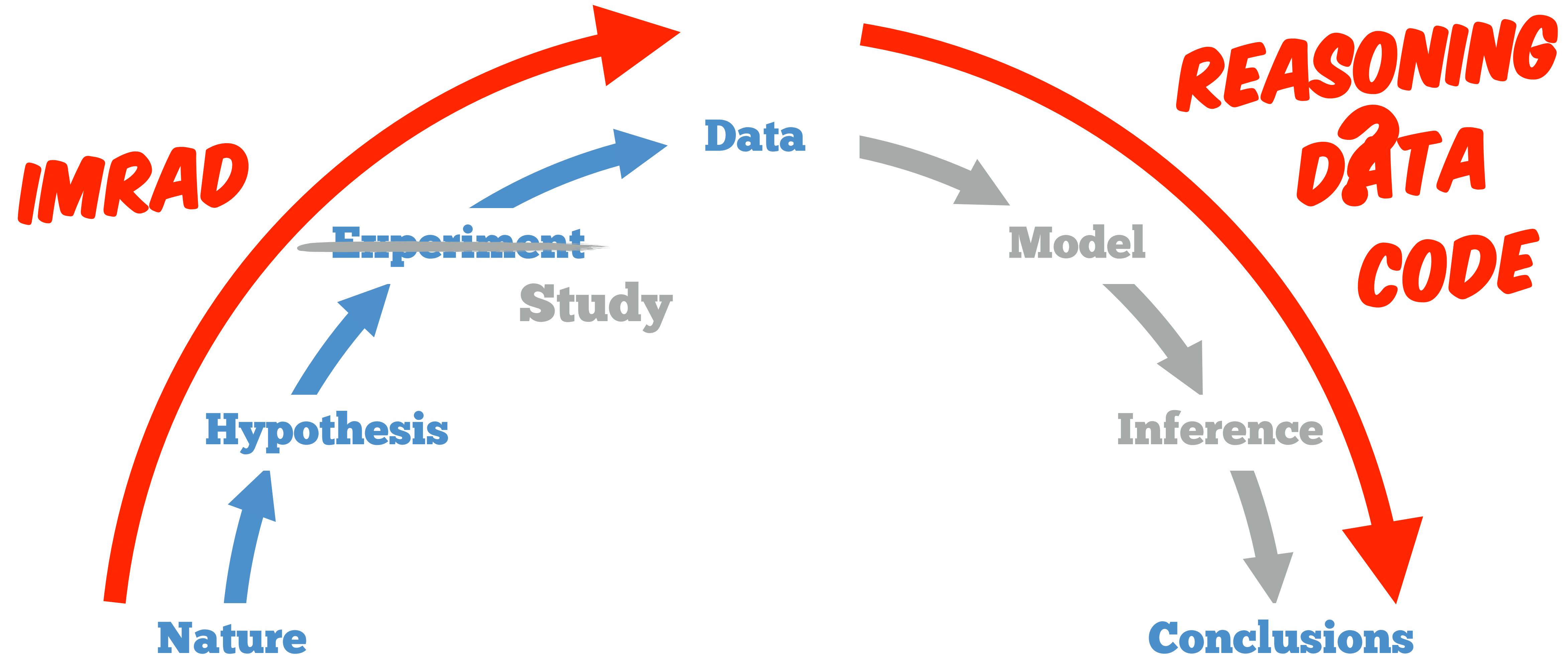
Adapted from Levy DG, Norris D. Methods and Guidelines for Integrity in Multivariate Analysis of Real World (Observational) Data. Unpublished Manuscript.

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



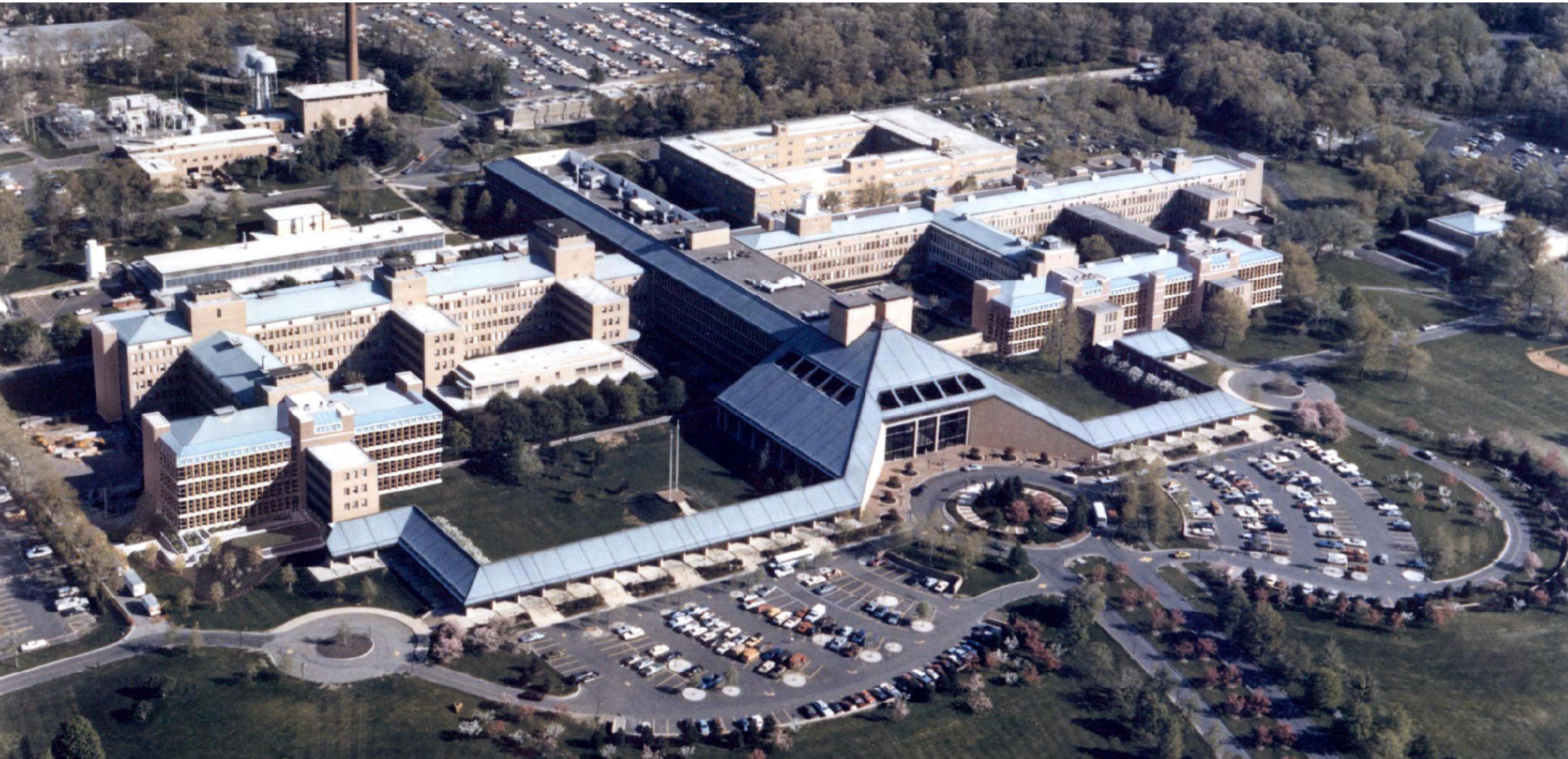
Adapted from Levy DG, Norris D. Methods and Guidelines for Integrity in Multivariate Analysis of Real World (Observational) Data. Unpublished Manuscript.

<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 an 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

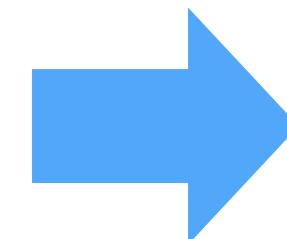
- \* **GitHub** [www.github.com](http://www.github.com)
- \* **StackOverflow** [www.stackoverflow.com](http://www.stackoverflow.com)
- \* **Reddit** [www.reddit.com](http://www.reddit.com)
- \* many more

# Headers

Use # to create headers.

Multiple #'s create lower level headers.

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



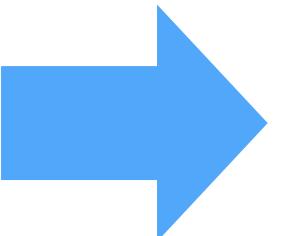
<b>Header 1</b>
<b>Header 2</b>
<b>Header 3</b>
<b>Header 4</b>
<b>Header 5</b>
<b>Header 6</b>

# 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.

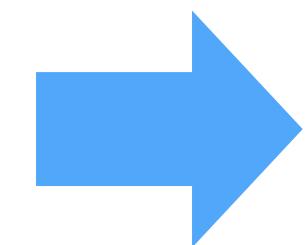
Use numbers to make numbered lists.

## 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

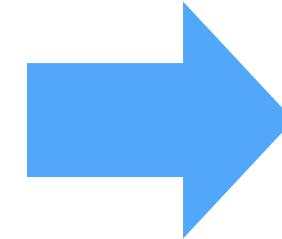
# Hyperlinks

Use brackets to denote a link.

Place the URL in parentheses.

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

**This is a link.**



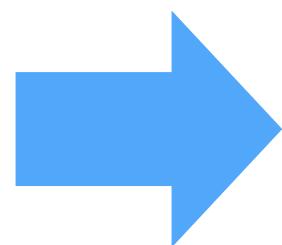
# Images

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.

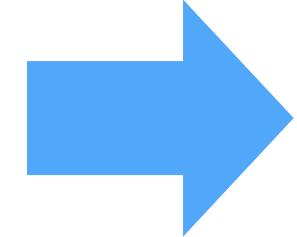


The RStudio logo.

# Equations

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

According to Einstein,  
 $E=mc^2$



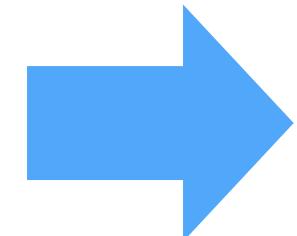
According to Einstein,  $E = mc^2$

# Equation blocks

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

According to  
Einstein,

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



According to  
Einstein,

$$E = mc^2$$

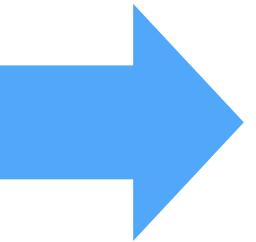
# Embed code

A large, semi-transparent watermark of the R logo is positioned in the bottom right corner. The logo consists of a circular emblem with the letters "R" inside.

# 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

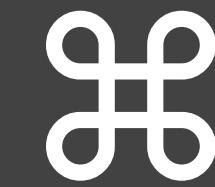
```
```{r}
# some code
```
```

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

+

i

(Mac)

Ctrl

+

Alt

+

i

(PC)

# Your Turn

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

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



+

Opt

+

A small lowercase letter 'i' inside a rounded square.

(Mac)

Ctrl

+

Alt

+

A small lowercase letter 'i' inside a rounded square.

(PC)

04 : 00

# chunk options

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

Here's some code

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

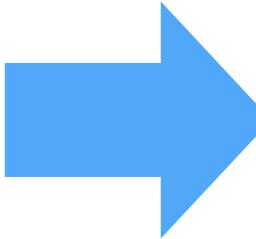
```
dim(iris)
```

```
```
```

Here's some code

```
dim(iris)
```

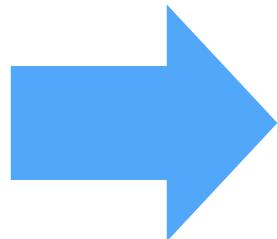
```
## [1] 150 5
```



# echo

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

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



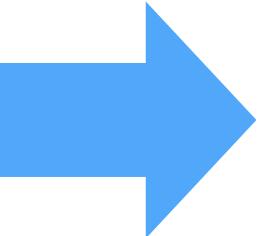
```
Here's some code  
## [1] 150 5
```

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)
```

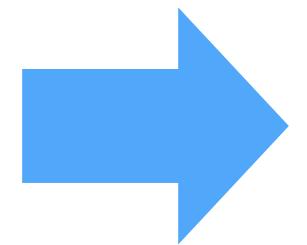
# include

**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

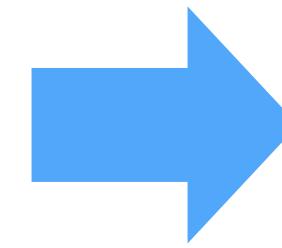


# fig.height, fig.width

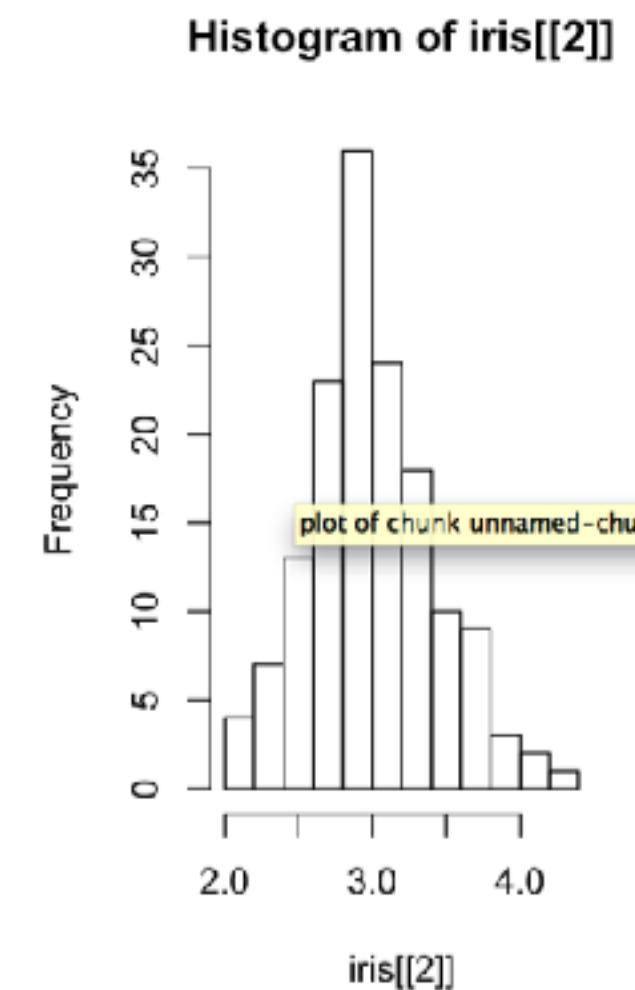
Specify the dimension of plots (in inches) with `fig.width` and `fig.height`. Separate multiple arguments with commas.

Here's a plot

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



Here's a plot



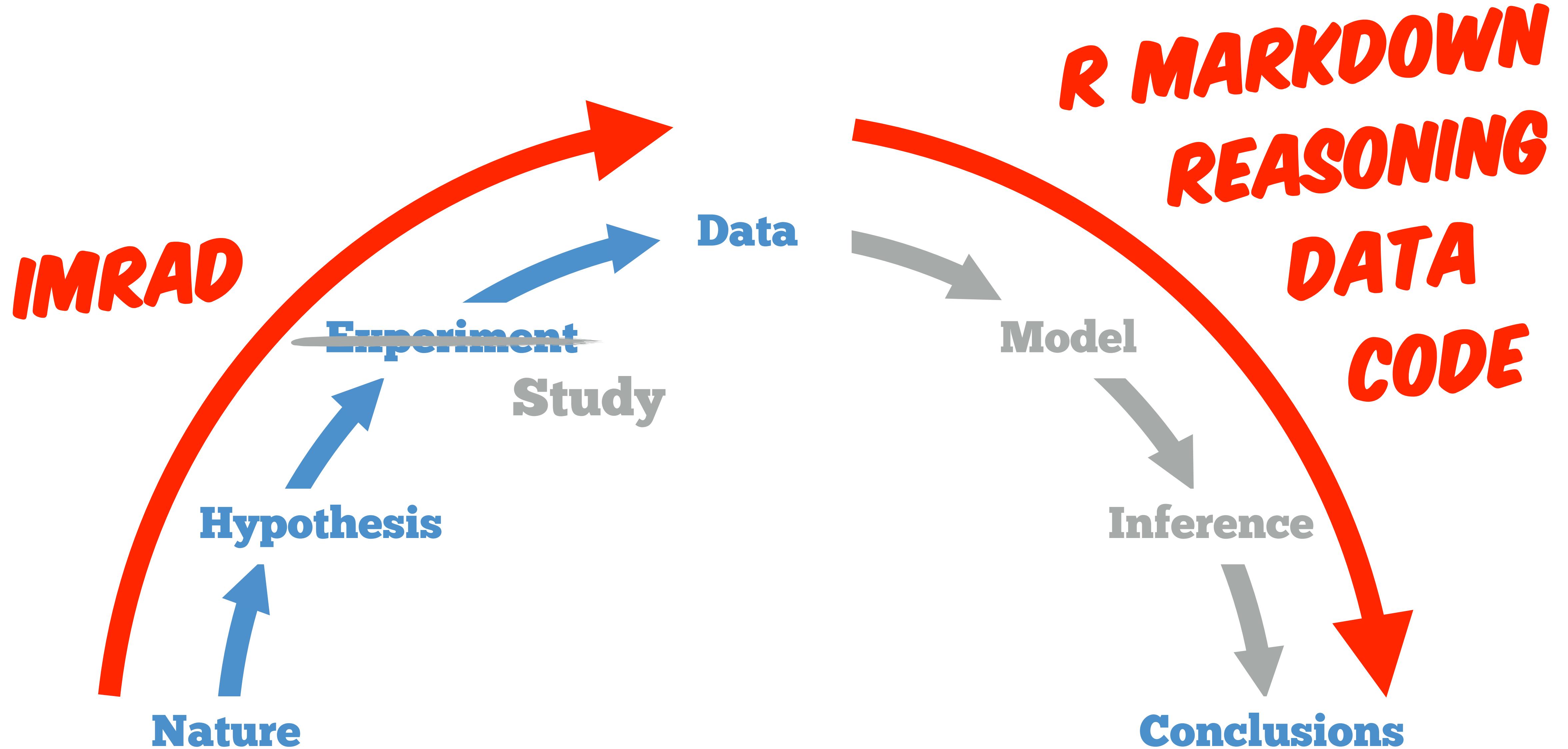
# Your Turn

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?

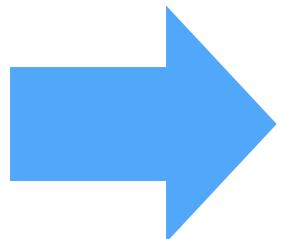


# YAML

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*

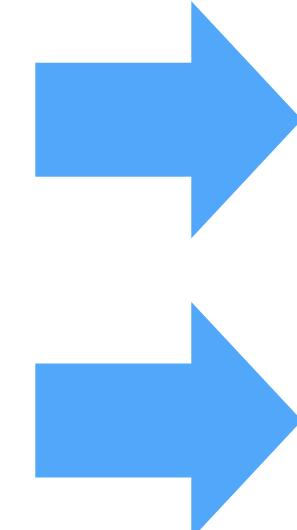
*February 4, 2015*

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}
```

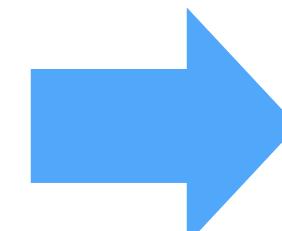
```
params$num
```

```
```
```

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**

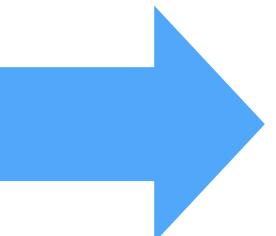
 **MORE**

# 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.



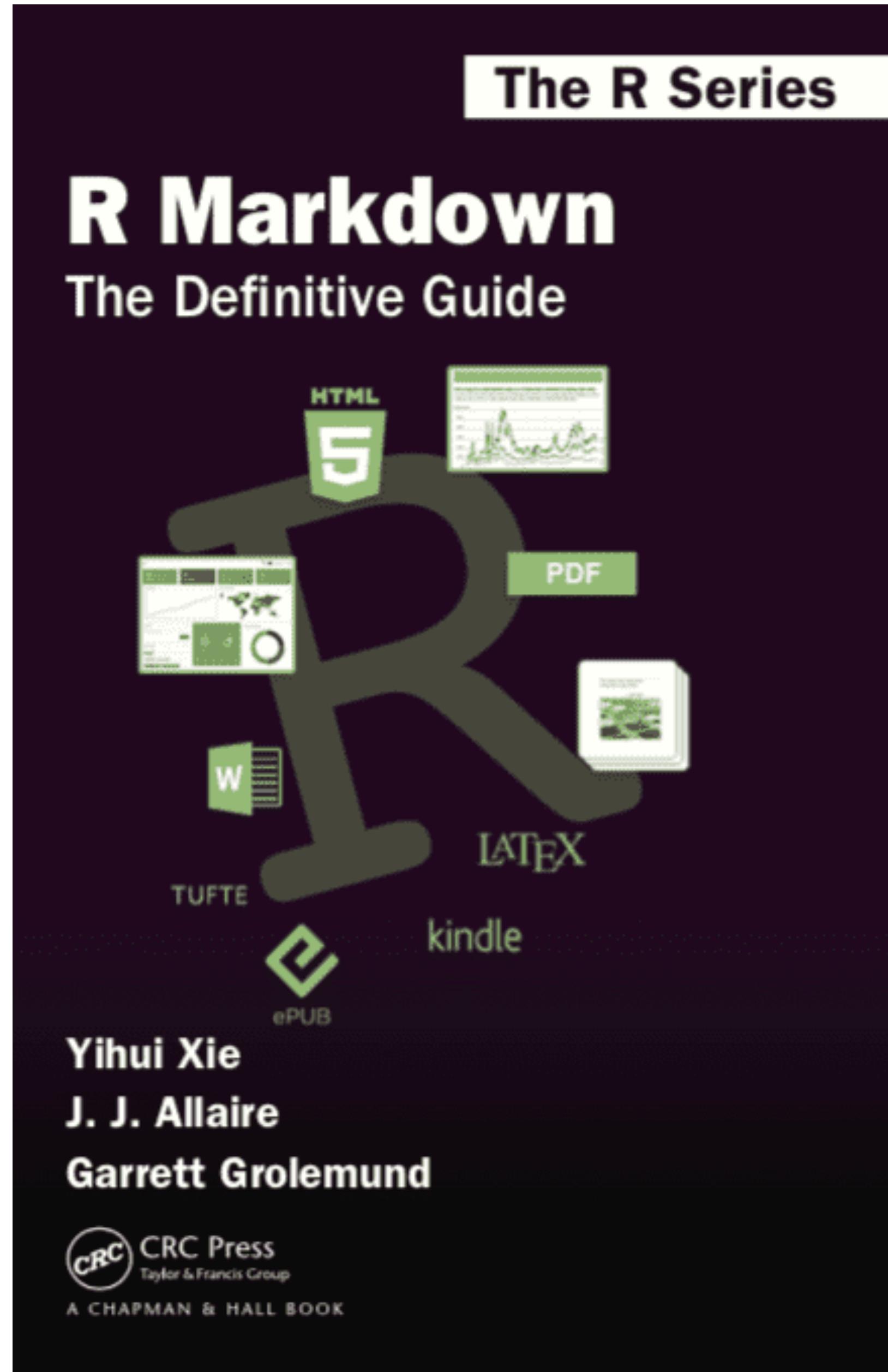
Some python code:

```
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/](http://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:

|                               |   |                    |                                       |   |                                 |
|-------------------------------|---|--------------------|---------------------------------------|---|---------------------------------|
| <b>input</b> - file to render | <b>output_options</b> - List of render options, as in YAML. | <b>output_file</b> | <b>params</b> - list of params to use | <b>envir</b> - environment to evaluate code chunks in | <b>encoding</b> - of input file |
|-------------------------------|---|--------------------|---------------------------------------|---|---------------------------------|

## Embed code with knitr syntax

**INLINE CODE**  
Insert with `!<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 `!>` with `!> {r echo=TRUE}` and `!> getVersion()`.

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

## IMPORTANT CHUNK OPTIONS

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                   |                                                                      |                                                                     |  |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|---------------------------------------------------------------------|--|--|
| <b>cache</b> - cache results for future knits (default = FALSE)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <b>dependson</b> - chunk dependencies for caching (default = NULL)                                                | <b>fig.align</b> - 'left', 'right', or 'center' (default = 'left')   | <b>message</b> - display code messages in document (default = TRUE) |  |  |
| <b>cache.path</b> - directory to save cached results in (default = "cache/")                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <b>echo</b> - Display code in output document: (default = TRUE)                                                   | <b>fig.cap</b> - figure caption as character string (default = NULL) | <b>results</b> (default = 'markup')<br>as 's' - passthrough results |  |  |
| <b>child</b> - file(s) to knit and then include (default = NULL)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <b>engine</b> - code language used in chunk (default = 'R')                                                       | <b>fig.height, fig.width</b> - Dimensions of plots in inches         | <b>highlight</b> - highlight source code (Default = TRUE)           |  |  |
| <b>collapse</b> - collapse all output into single block (default = FALSE)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <b>error</b> - Display error messages in doc (TRUE) or stop rendering when errors occur (FALSE) (default = FALSE) | <b>hold</b> - put all results below all code                         | <b>tidy</b> - tidy code for display (default = FALSE)               |  |  |
| <b>comment</b> - prefix for each line of results (default = '#')                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <b>eval</b> - run code in chunk (default = TRUE)                                                                  | <b>print</b> - display results (default = TRUE)                      | <b>warning</b> - display code warnings in document (default = TRUE) |  |  |
| Options not listed above: <code>options</code> , <code>animate</code> , <code>background</code> , <code>cache.comments</code> , <code>cache.lazy</code> , <code>cache.rebuild</code> , <code>cache.var</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.out.extra</code> , <code>out.height</code> , <code>out.width</code> , <code>prompt</code> , <code>purf</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., `yaml`) 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 `knitWithParams` or `.Rmd` 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(
  readCars(), nrow(n))
```

```

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`.

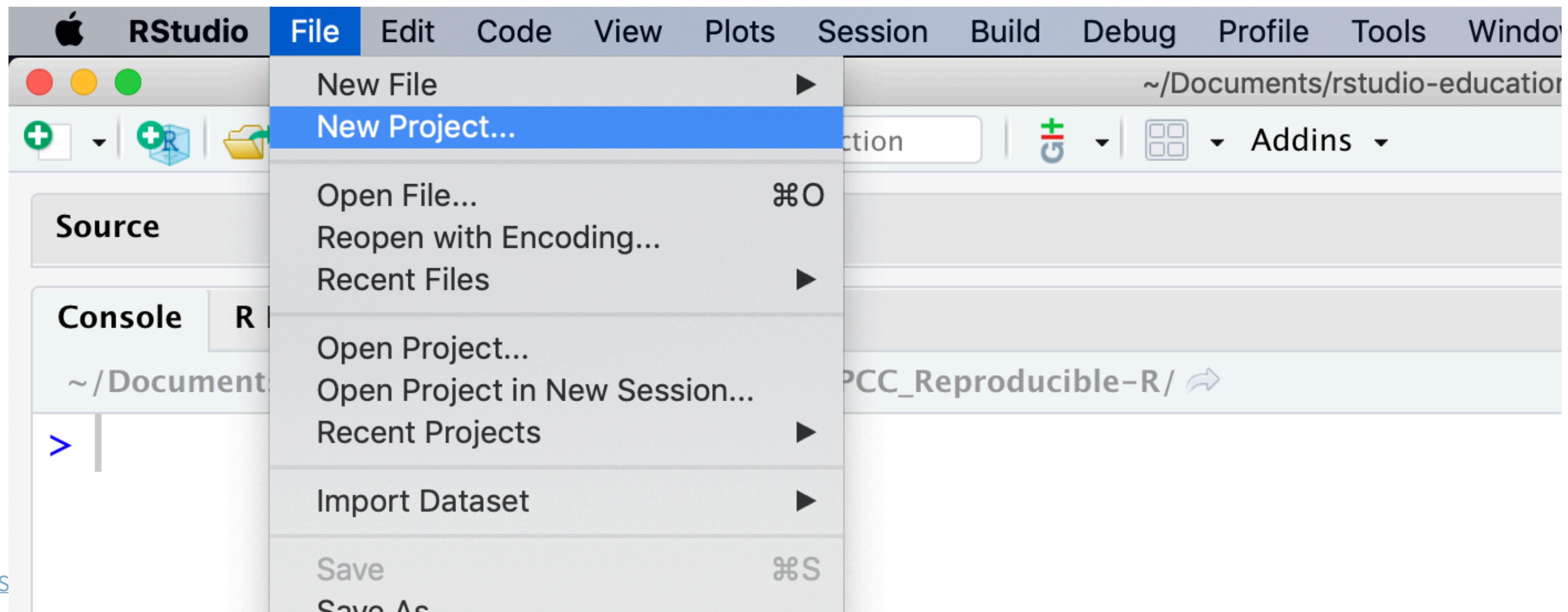
[rstudio.com/cheatsheets](http://rstudio.com/cheatsheets)

# Projects

R

# How to keep track of your files?

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

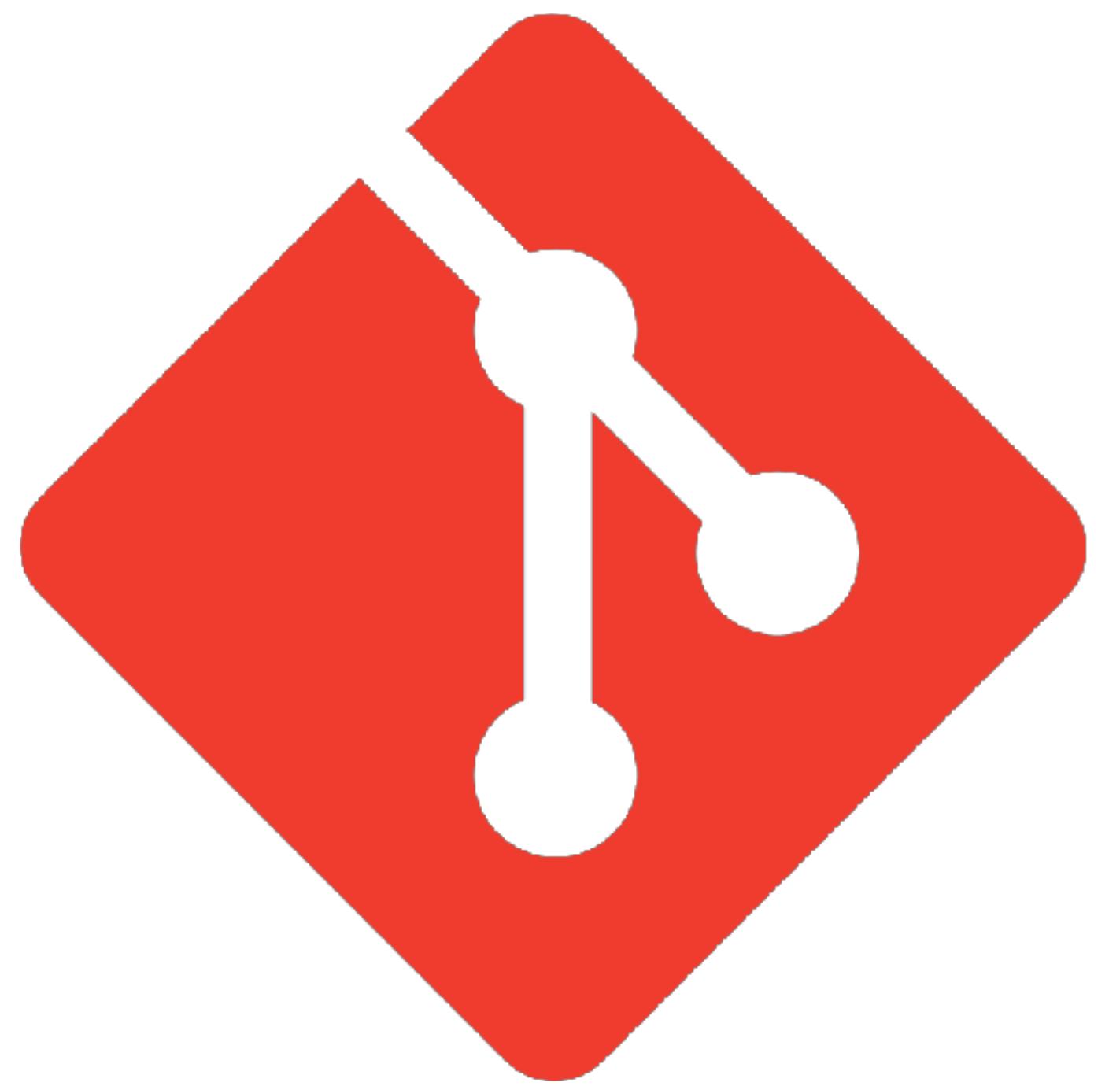


# Git and Github

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)
```

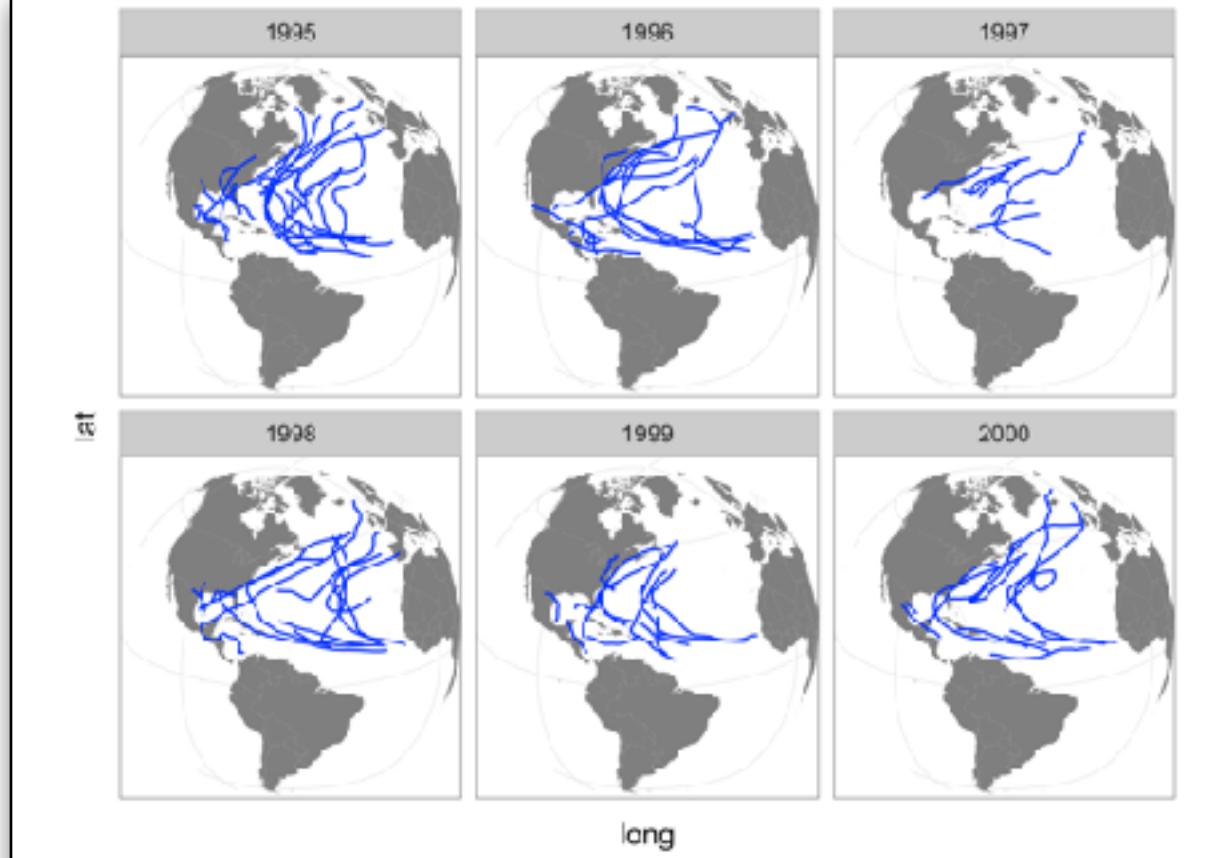
```
# 1-Plot.R

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))

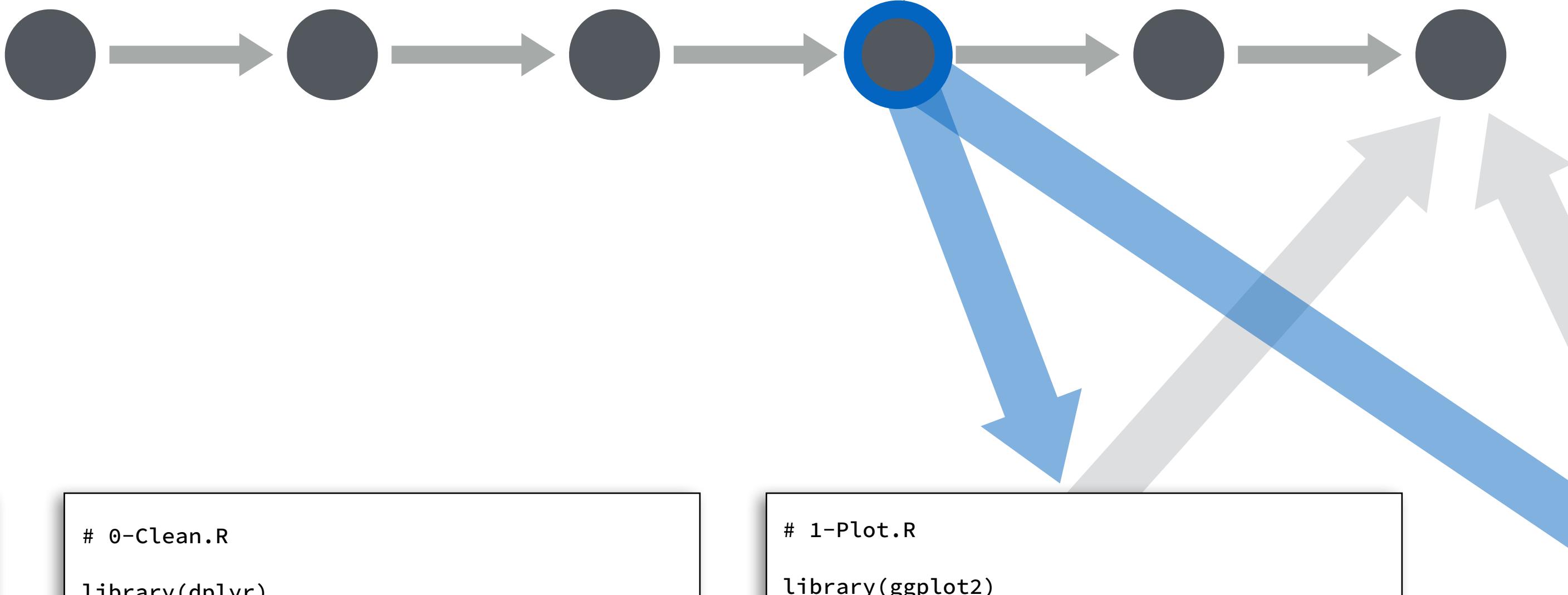
ggsave("storms.png", width = 7, height = 5)
```



# 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)
```

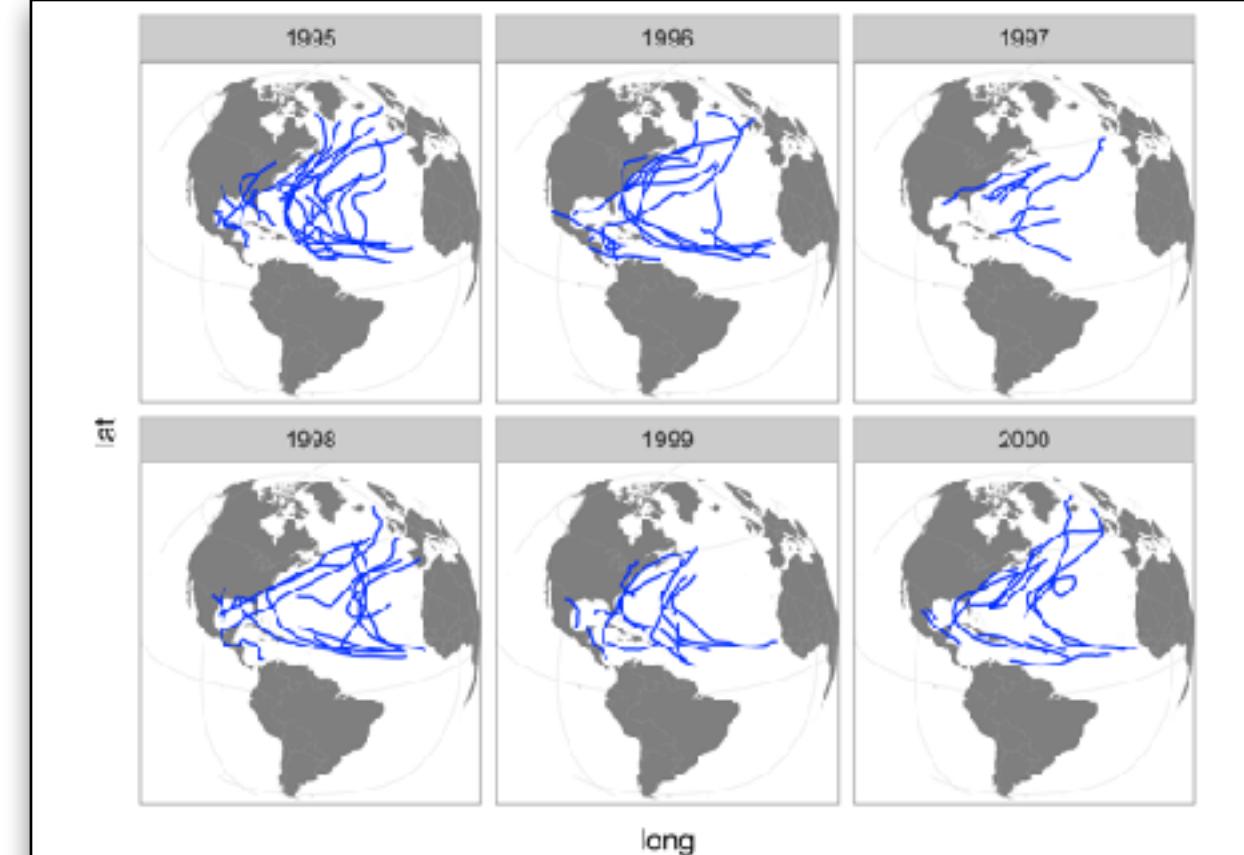
```
# 1-Plot.R

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

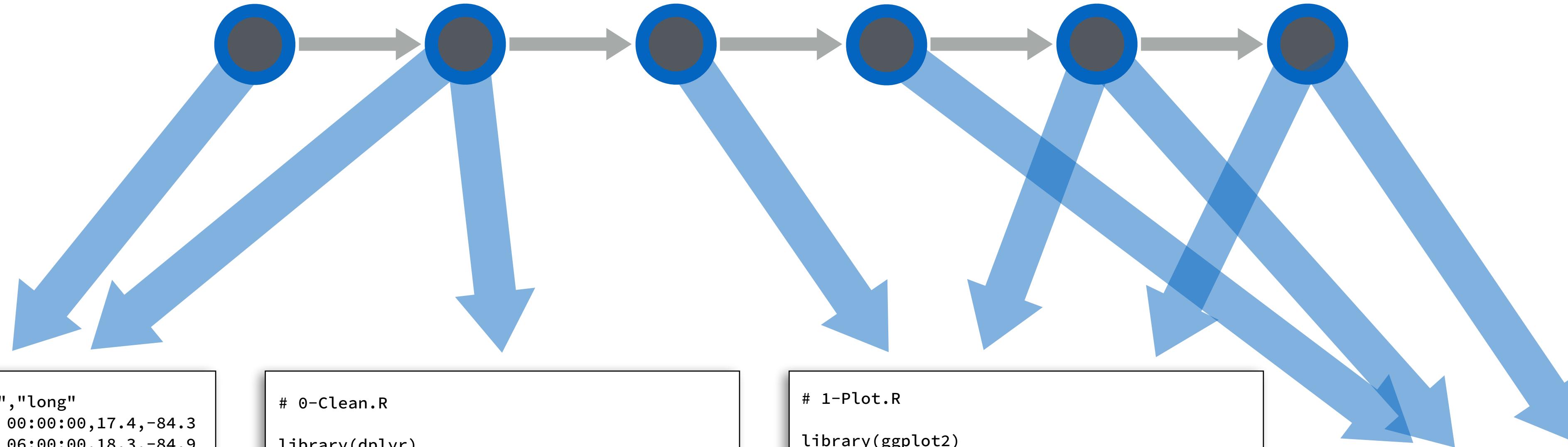
# History



.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)

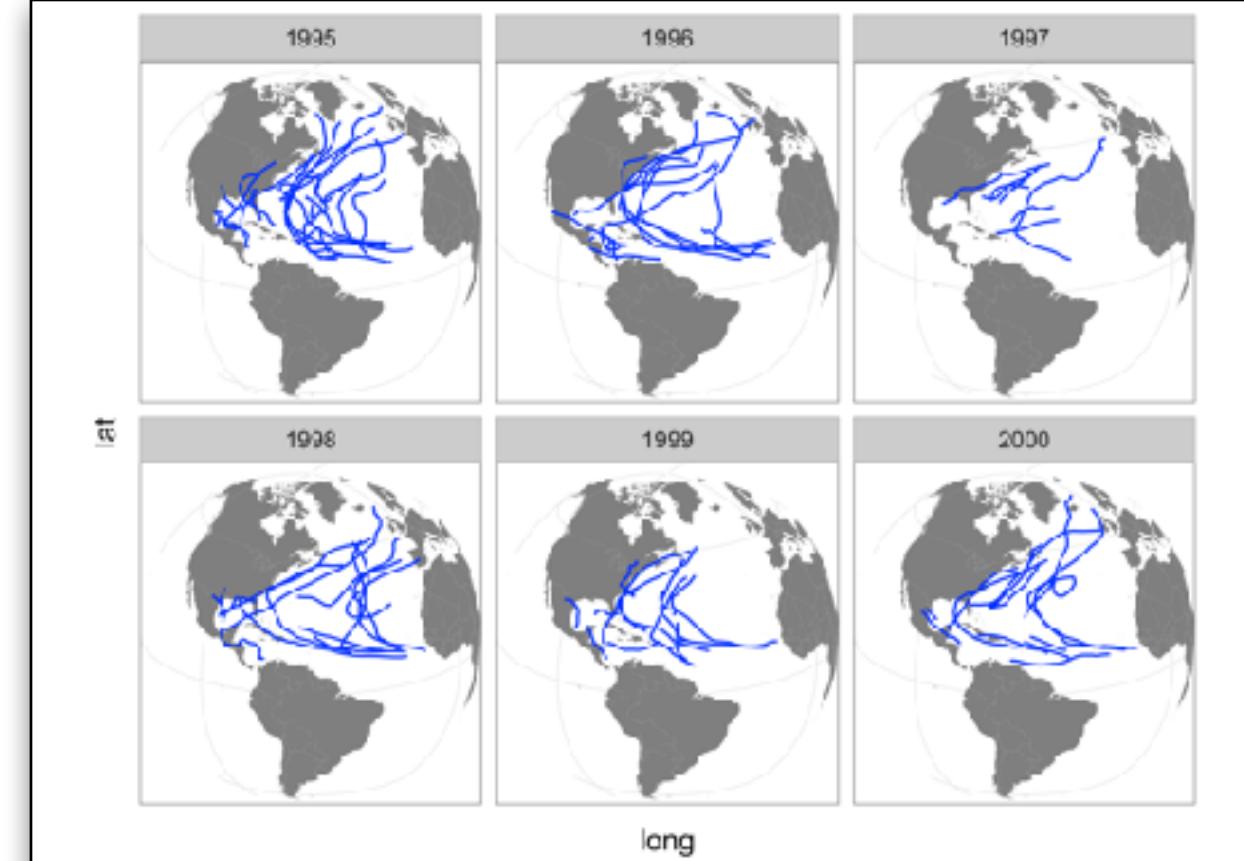
write.csv(storms, file = "storms.csv",
  row.names = FALSE)
```

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

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

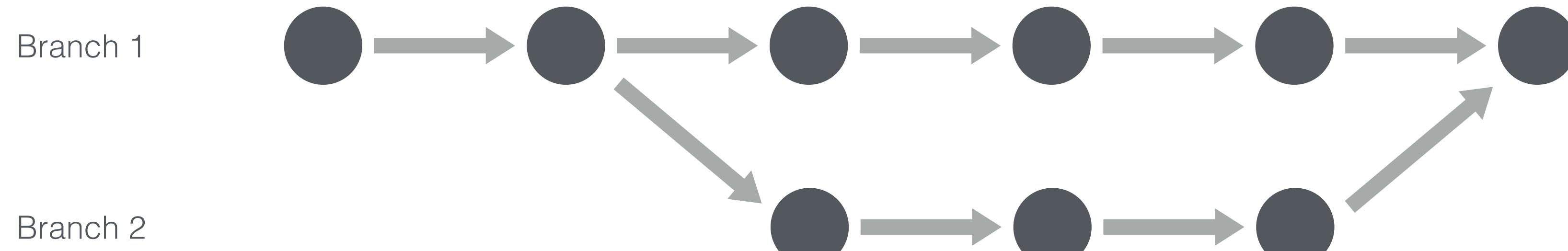
ggplot(storms, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group),
    fill = "grey50", data = map) +
  geom_path(aes(group = name),
    color = "blue") +
  facet_wrap(~ year) +
  theme_bw() +
  coord_map(projection = "ortho",
    orientation = c(21, -60, 0))

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



# Project

# History



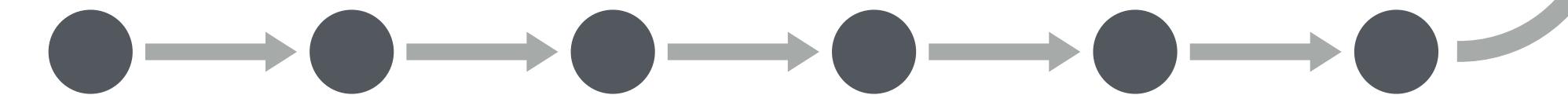
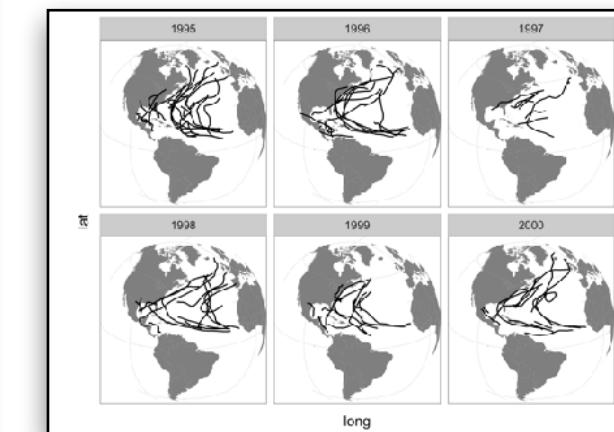
# "Official" Version

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)
```



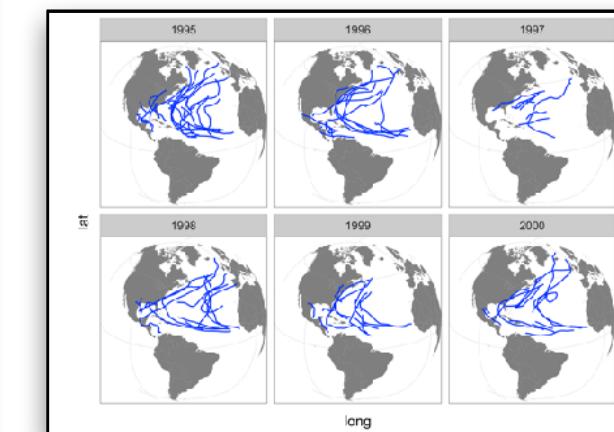
# Real Life Version

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**

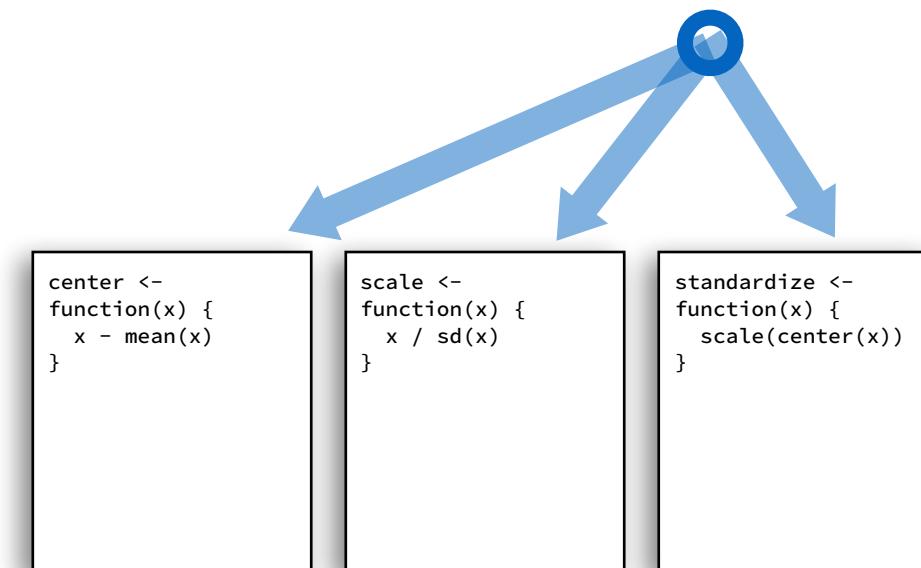
[www.github.com](http://www.github.com)



GitHub

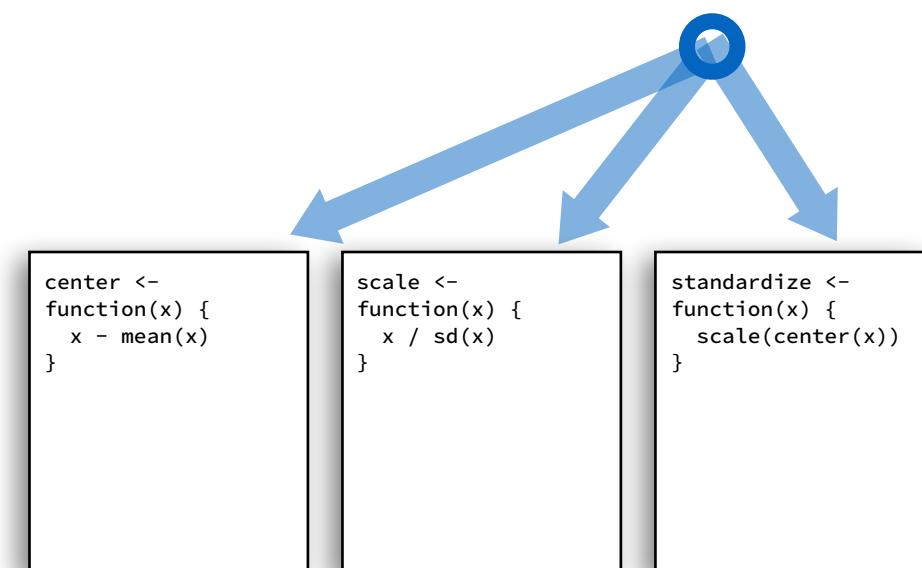


```
center <-  
function(x) {  
  x - mean(x)  
}  
  
scale <-  
function(x) {  
  x / sd(x)  
}  
  
standardize <-  
function(x) {  
  scale(function(x) {  
    x - mean(x)  
})  
}
```



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

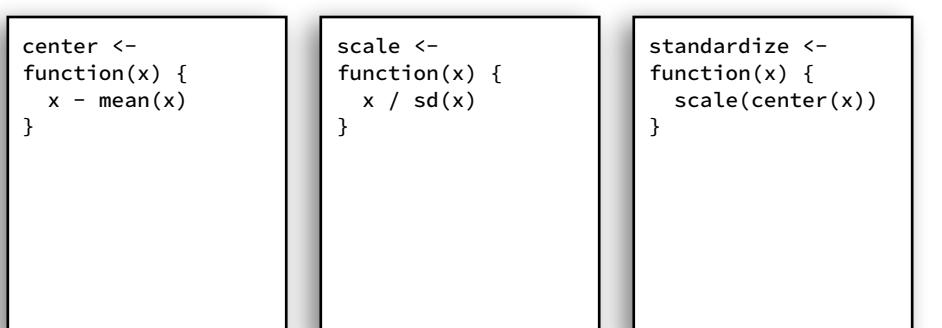
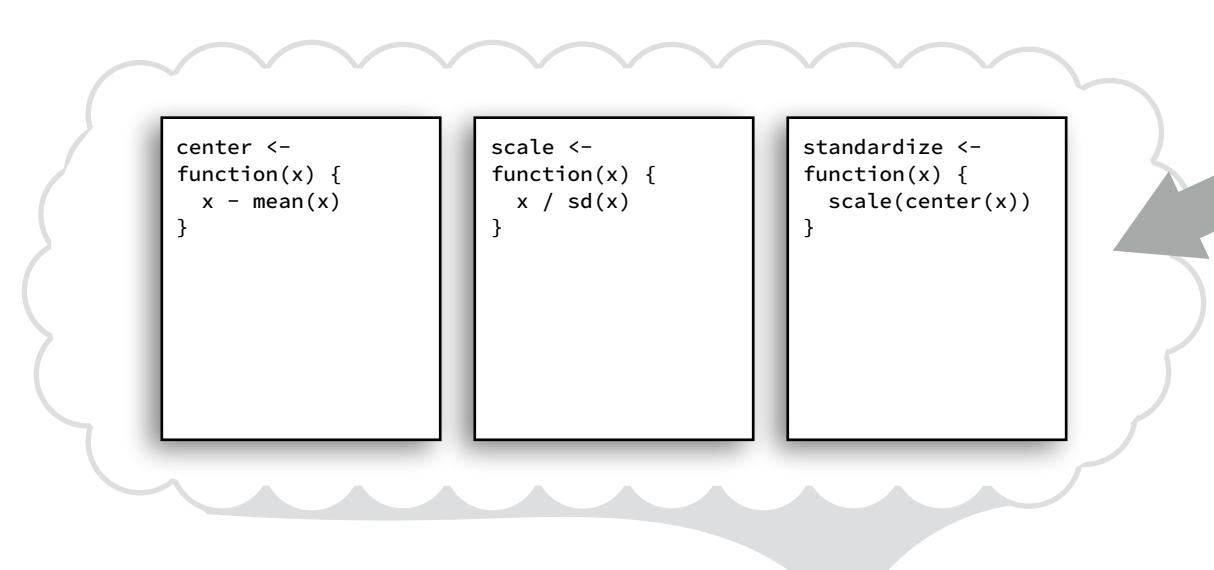
available to public

## 2."Official" Version

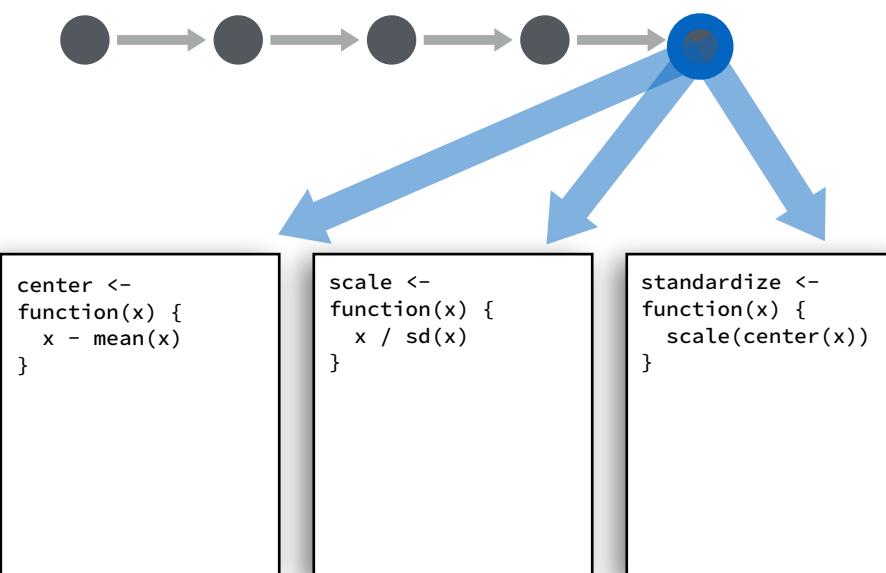
implied by commit history

## 1. Real Life Version

in your working directory



You





# GitHub

## 3. Github Version

available to public

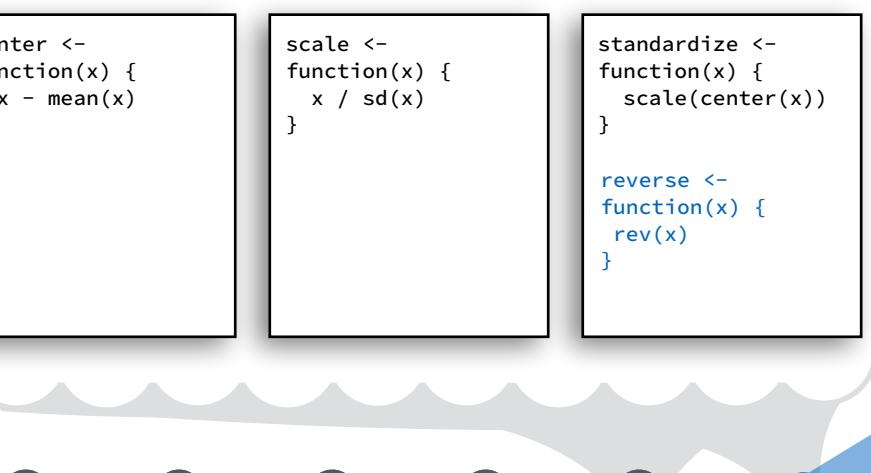
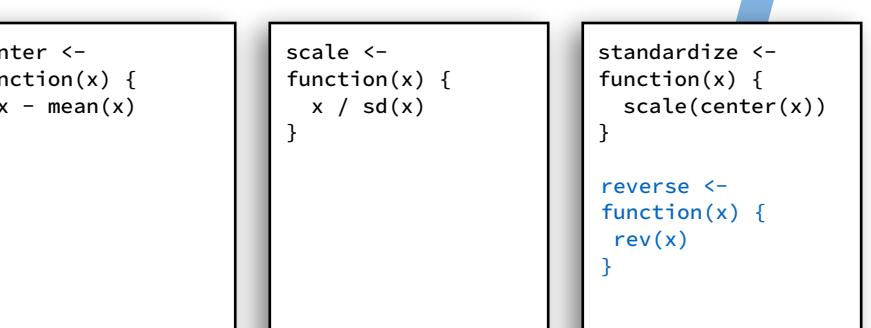
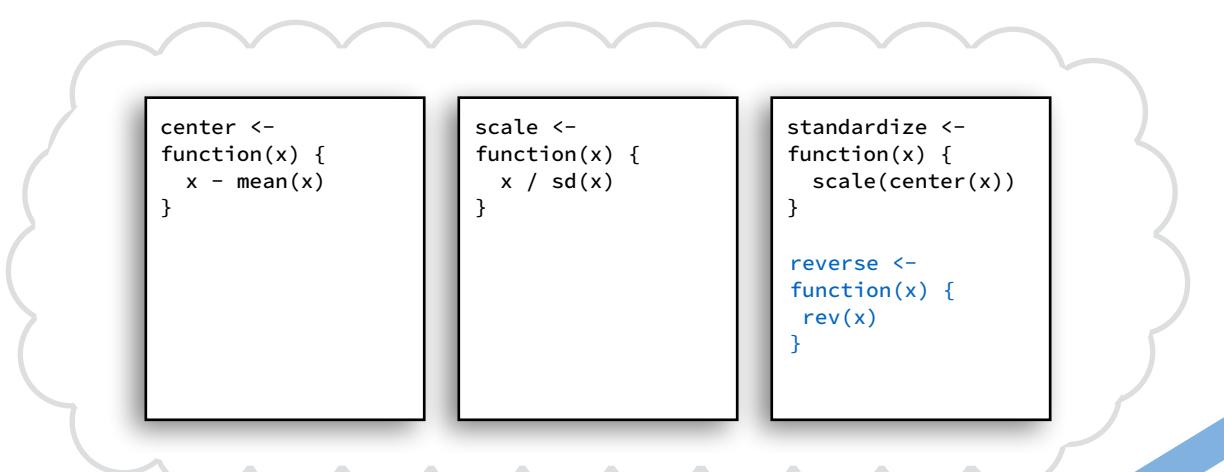
## 2."Official" Version

implied by commit history

## 1. Real Life Version

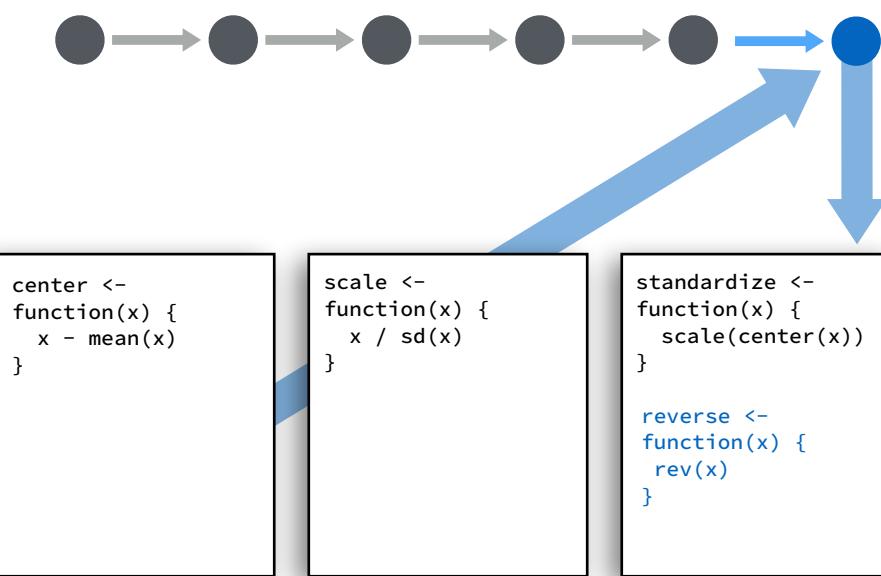
in your working directory

You



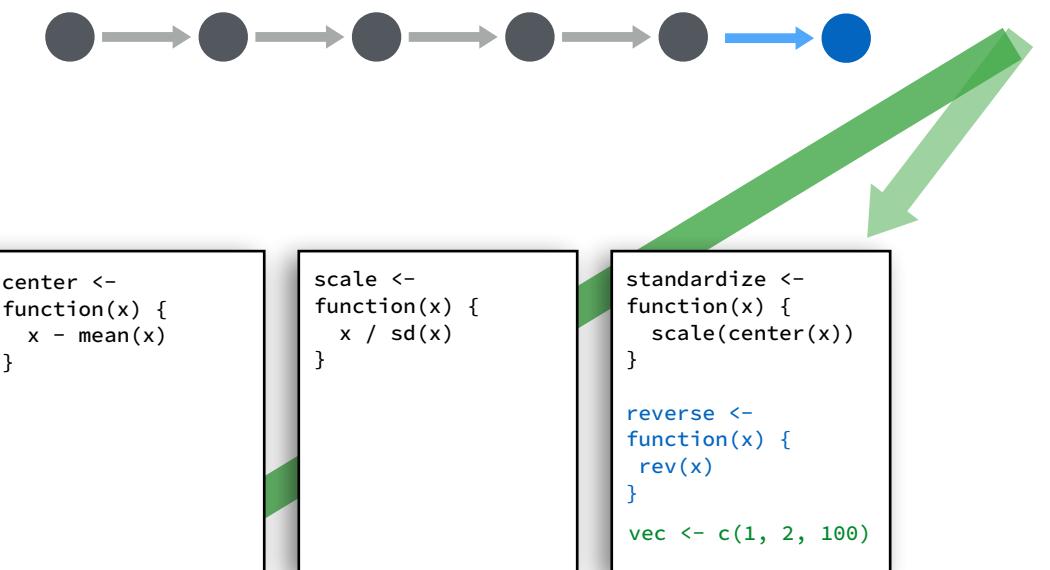
push

add + commit





# GitHub



## 3. Github Version

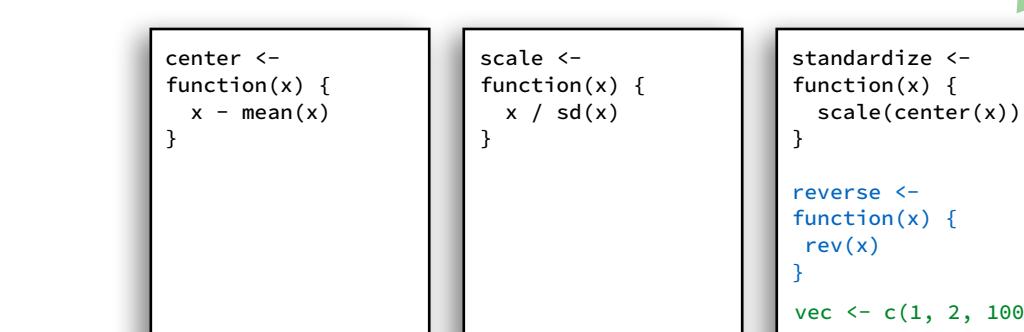
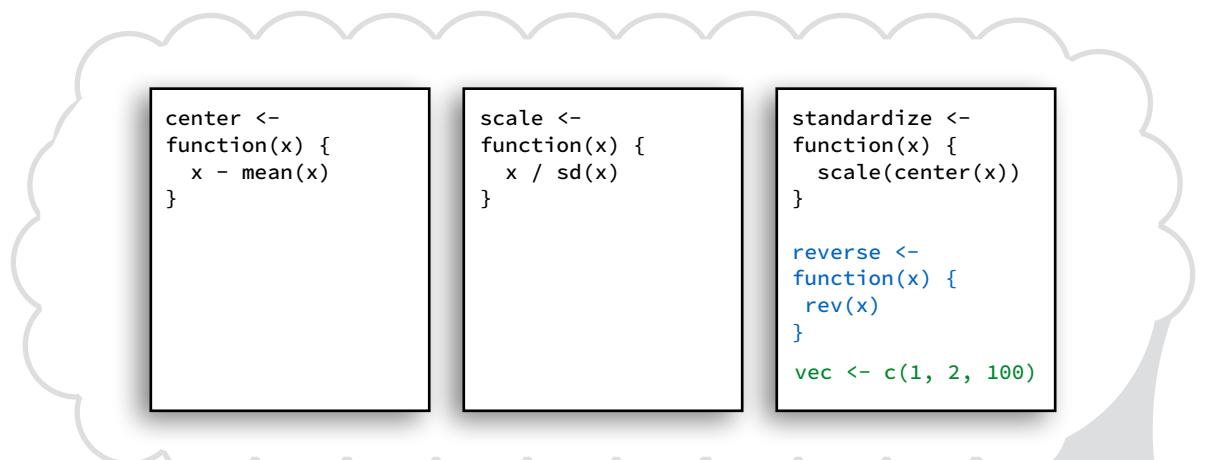
available to public

## 2."Official" Version

implied by commit history

## 1. Real Life Version

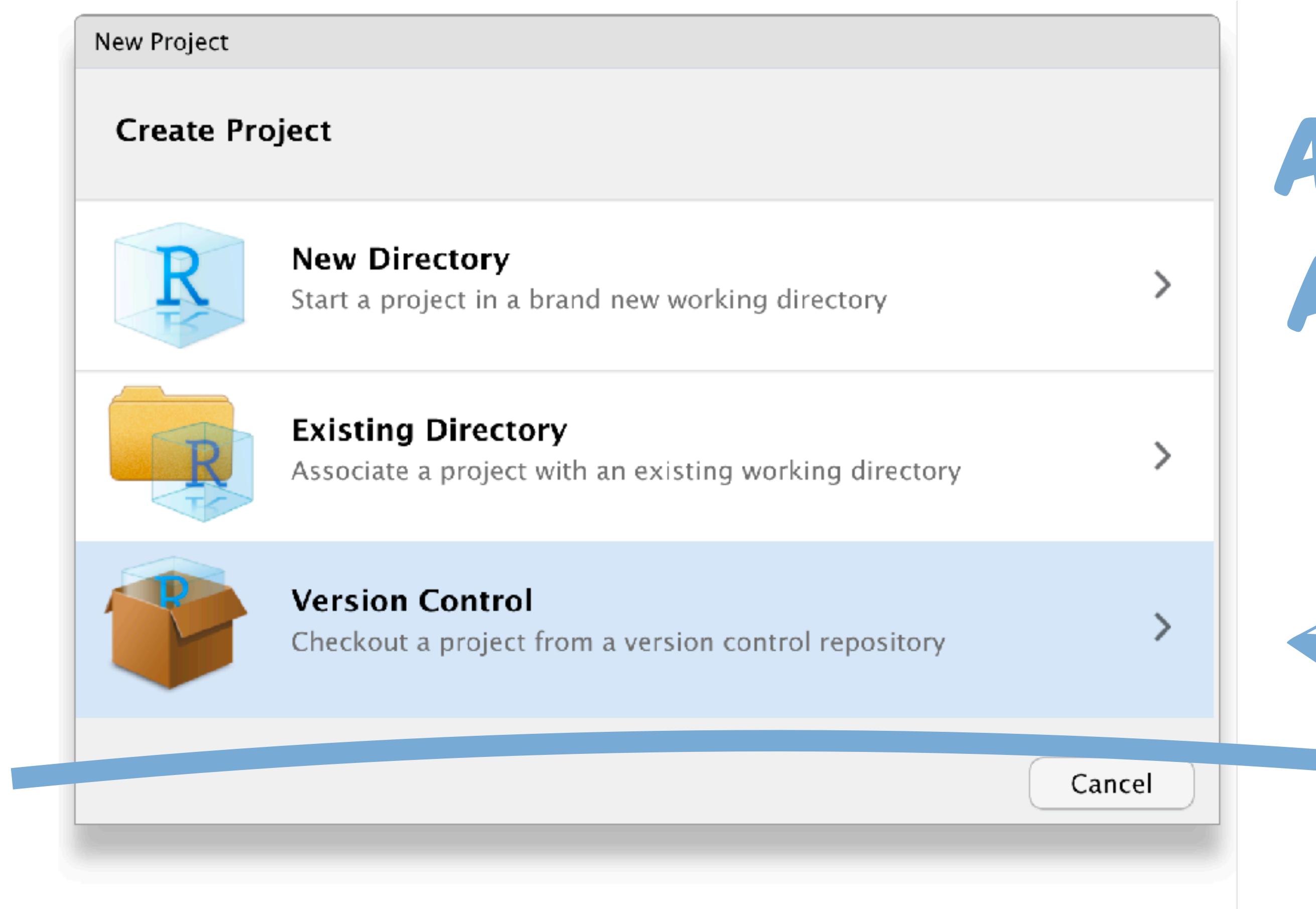
in your working directory



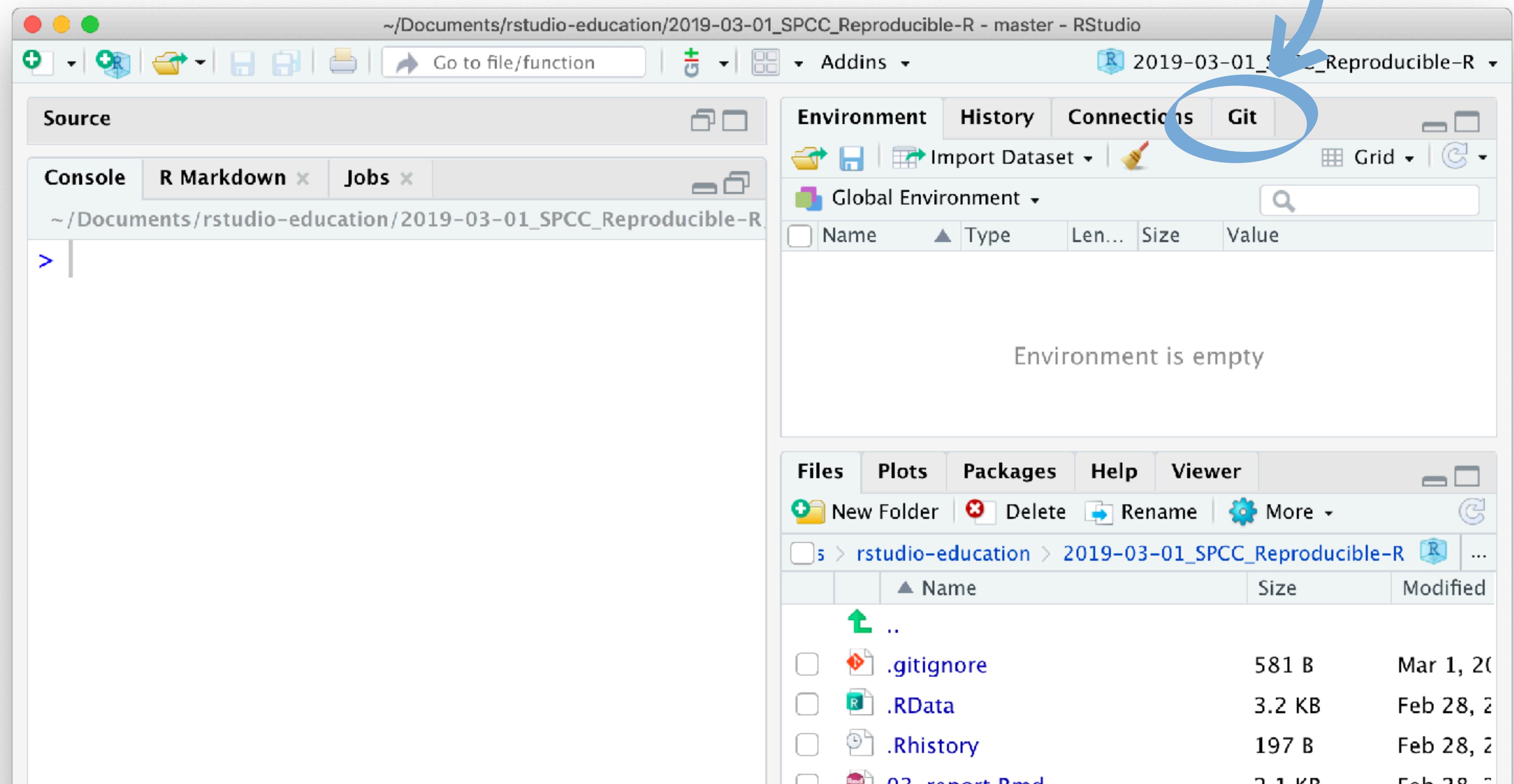
You

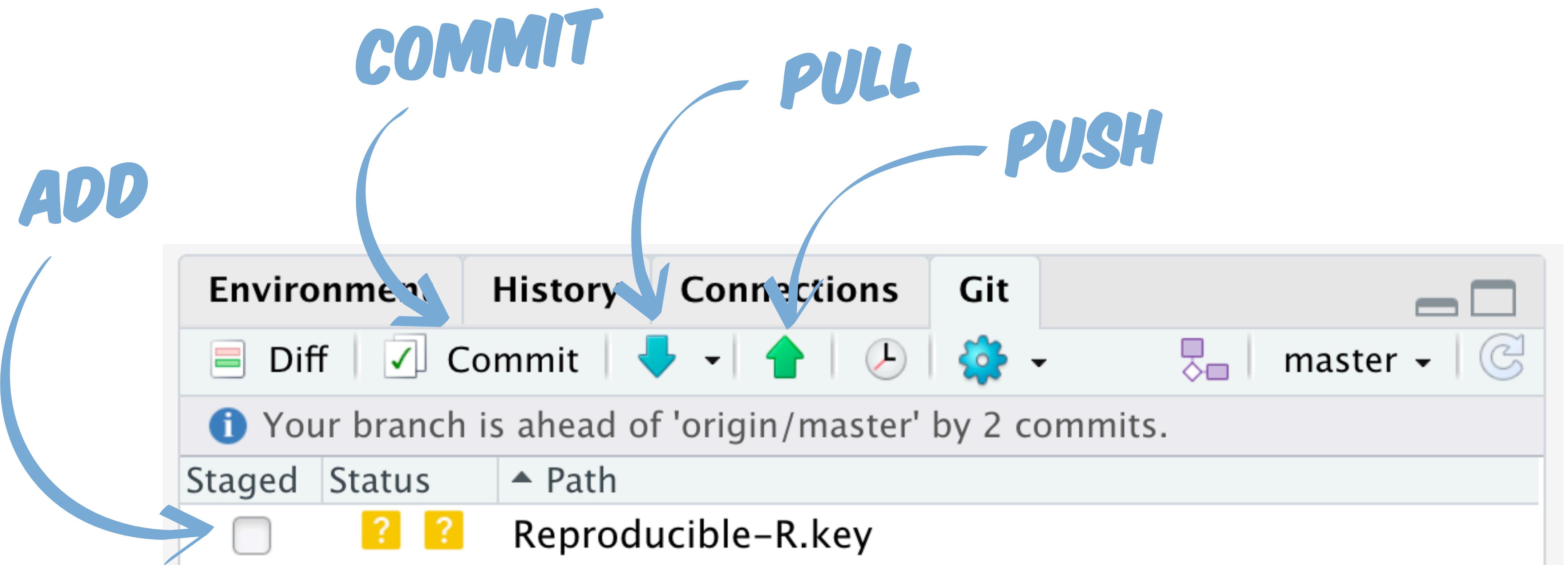
pull

# DOWNLOAD A REPOSITORY AS A PROJECT



# A GIT PANE





# R Markdown

# Takeaway



# One more thing

**Reproducibility  
is an opportunity**

**Reproducibility  
is an opportunity**

schedule  
parameterize  
automate

# Thank you

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

[github.com/rstudio-education/  
2019-03-01\\_SPCC\\_Reproducible-R](https://github.com/rstudio-education/2019-03-01_SPCC_Reproducible-R)