Workshop 6 - Reproducible Computational Environments and Presentations

Andrew Stewart

Andrew.Stewart@manchester.ac.uk

@ajstewart_lang

https://github.com/ajstewartlang

Workshop	Topic
1	Reproducibility and R
2	General Linear Model (Regression)
3	General Linear Model (ANOVA)
4	Mixed Models
5	Data Simulation and Advanced Data Visualisation
6	Reproducible Computational Environments and Presentations

Assignment

Assignment to be completed by Semester 1 exam period.

Open and Reproducible Research

- Shared Data - we already know this is important for reproducibility.
- Shared Code - we already know this is important for reproducibility.
- Shared Computational environment - why is this important and how do we do it?

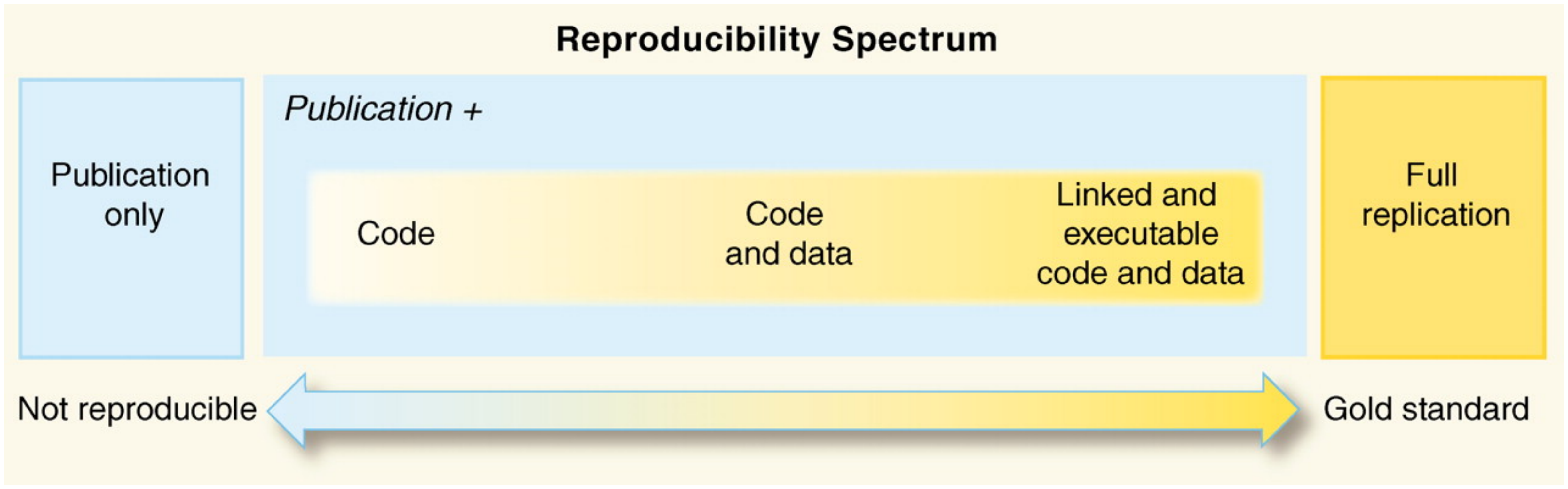
PERSPECTIVE

Reproducible Research in Computational Science

Roger D. Peng

+ See all authors and affiliations

Science 02 Dec 2011:
Vol. 334, Issue 6060, pp. 1226-1227
DOI: 10.1126/science.1213847



Why do we need to reproduce the computational environment?

- Quite often analysis code ‘breaks’ - often in one of two ways:
- Code that worked previously now doesn’t - maybe a function in an R package was updated (e.g., `lsmeans` became `emmeans` so old code using `lsmeans` wouldn’t now run).
- Code that worked previously still works - but produces a slightly different result or now throws a warning where it didn’t previousy (e.g., convergence/singular fit warnings in `lme4` version 1.1-19 vs. version 1.1-20).

Capturing your local computational environment

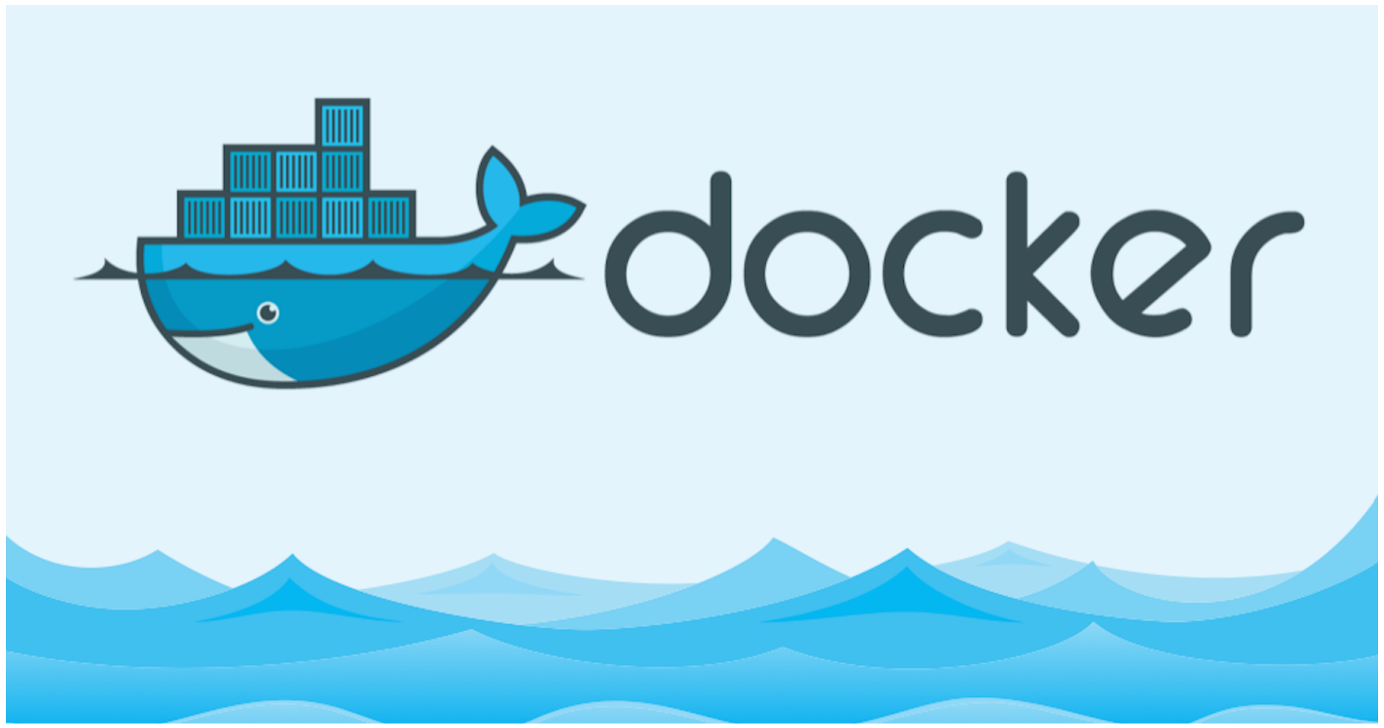
- You need to capture the versions of the different R packages (plus their dependencies).

- May sound trivial but trying running some old R code and be amazed at how many things now don’t work as they once did!

Docker for beginners

Docker packages your data, code and all its dependencies in the form called a docker container to ensure that your application works seamlessly in any environment.

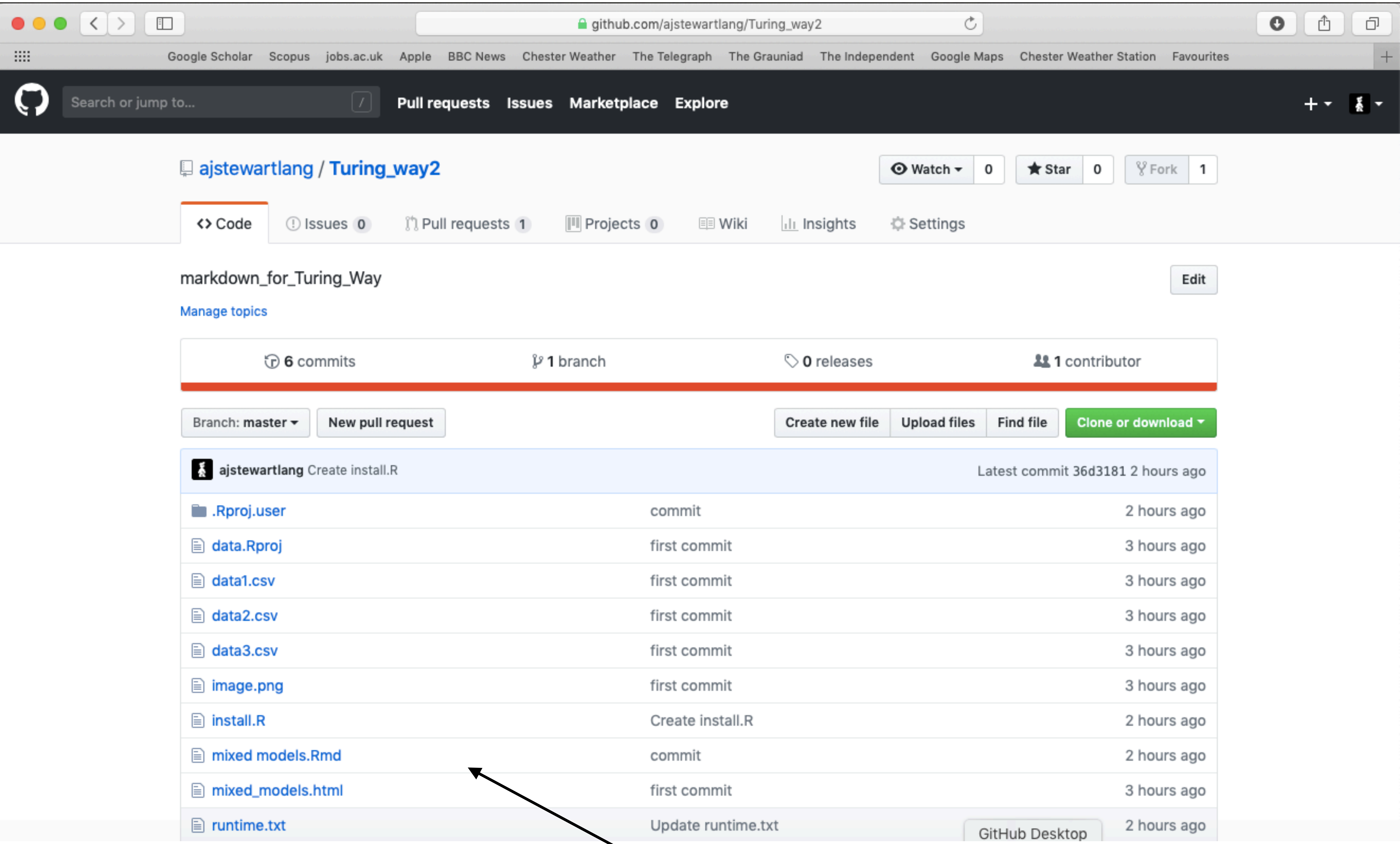
When you run a docker container it's like running your analysis on a virtual computer that has the same configuration as our own one at the point in time when you ran the analysis.



`https://medium.com/the-andela-way/docker-for-beginners-61e8e0ce6a19`

So what’s Binder?

- Binder is powered by BinderHub, which is an open-source tool that deploys the Binder service in the cloud.
- Binder works by pulling a repository that you set up on GitHub into a Docker container - `repo2docker`.
- Think of a repository as a folder containing your R code, your data, and a few other small bits and pieces - but it sits in the cloud rather than on your computer.


github.com/ajstewartlang/Turing_way2

ajstewartlang / Turing_way2

markdown_for_Turing_Way

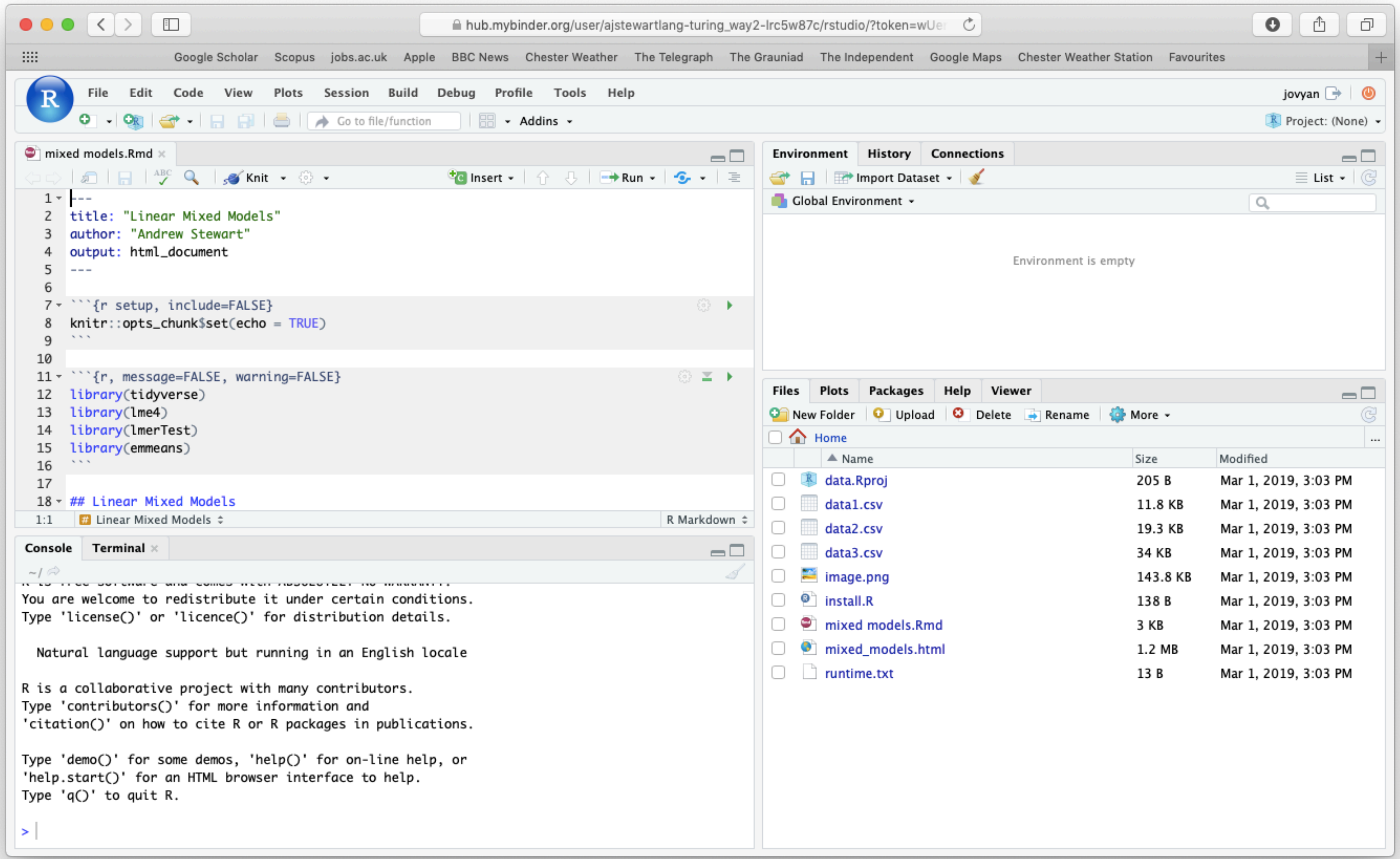
6 commits | 1 branch | 0 releases | 1 contributor

ajstewartlang Create install.R — Latest commit 36d3181 2 hours ago

.Rproj.user	commit	2 hours ago
data.Rproj	first commit	3 hours ago
data1.csv	first commit	3 hours ago
data2.csv	first commit	3 hours ago
data3.csv	first commit	3 hours ago
image.png	first commit	3 hours ago
install.R	Create install.R	2 hours ago
mixed models.Rmd	commit	2 hours ago
mixed_models.html	first commit	3 hours ago
runtime.txt	Update runtime.txt	2 hours ago

My R code and data files.

- When I link my GitHub repository to Binder and launch it I then get the following in my web browser.
- This is RStudio running the cloud using my code, my data and the appropriate versions of the packages that I was using when I did the analysis originally!



~~~
---
title: "Linear Mixed Models"
author: "Andrew Stewart"
output: html_document
---

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

```{r, message=FALSE, warning=FALSE}
library(tidyverse)
library(lme4)
library(lmerTest)
library(emmeans)
```

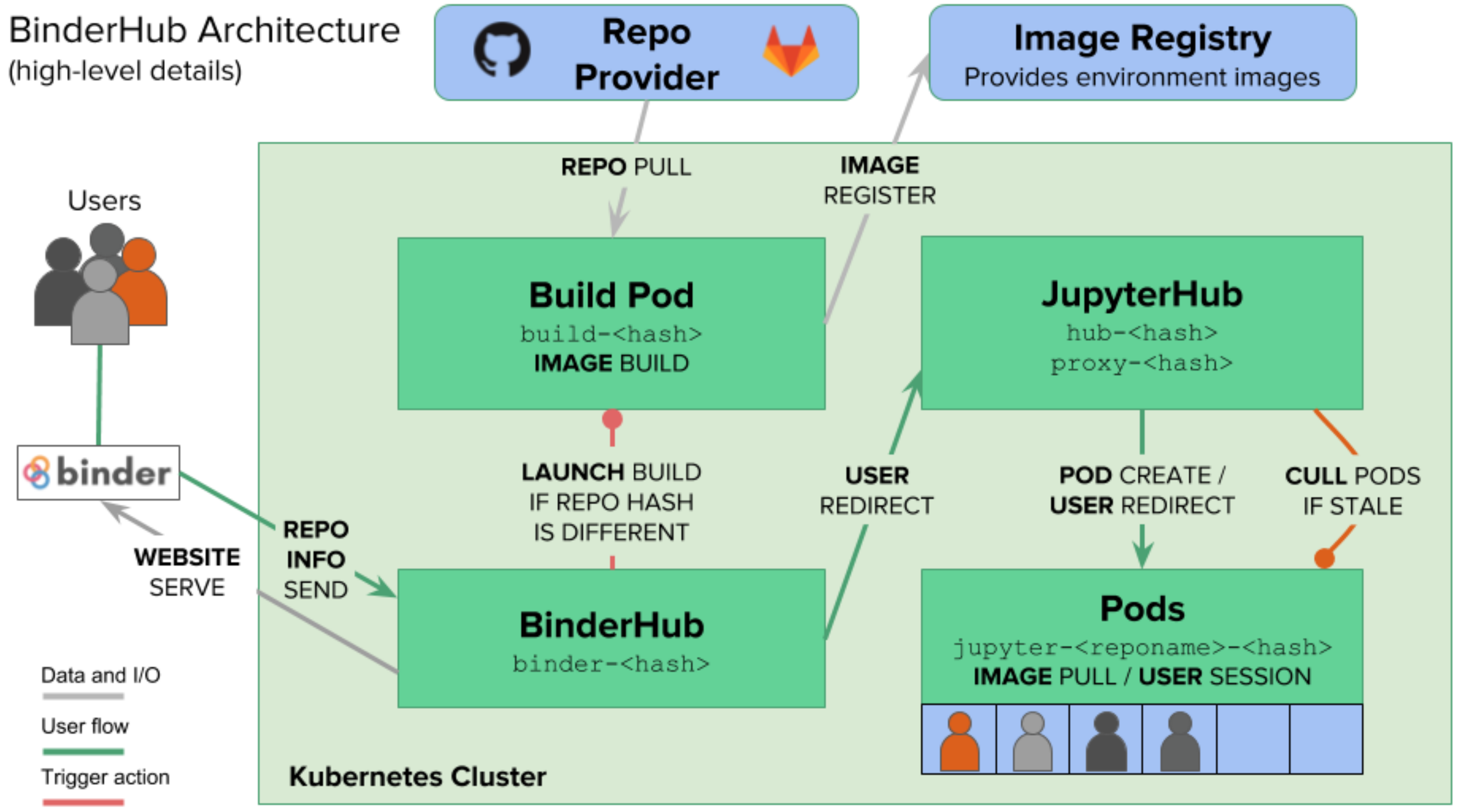
Linear Mixed Models
```

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

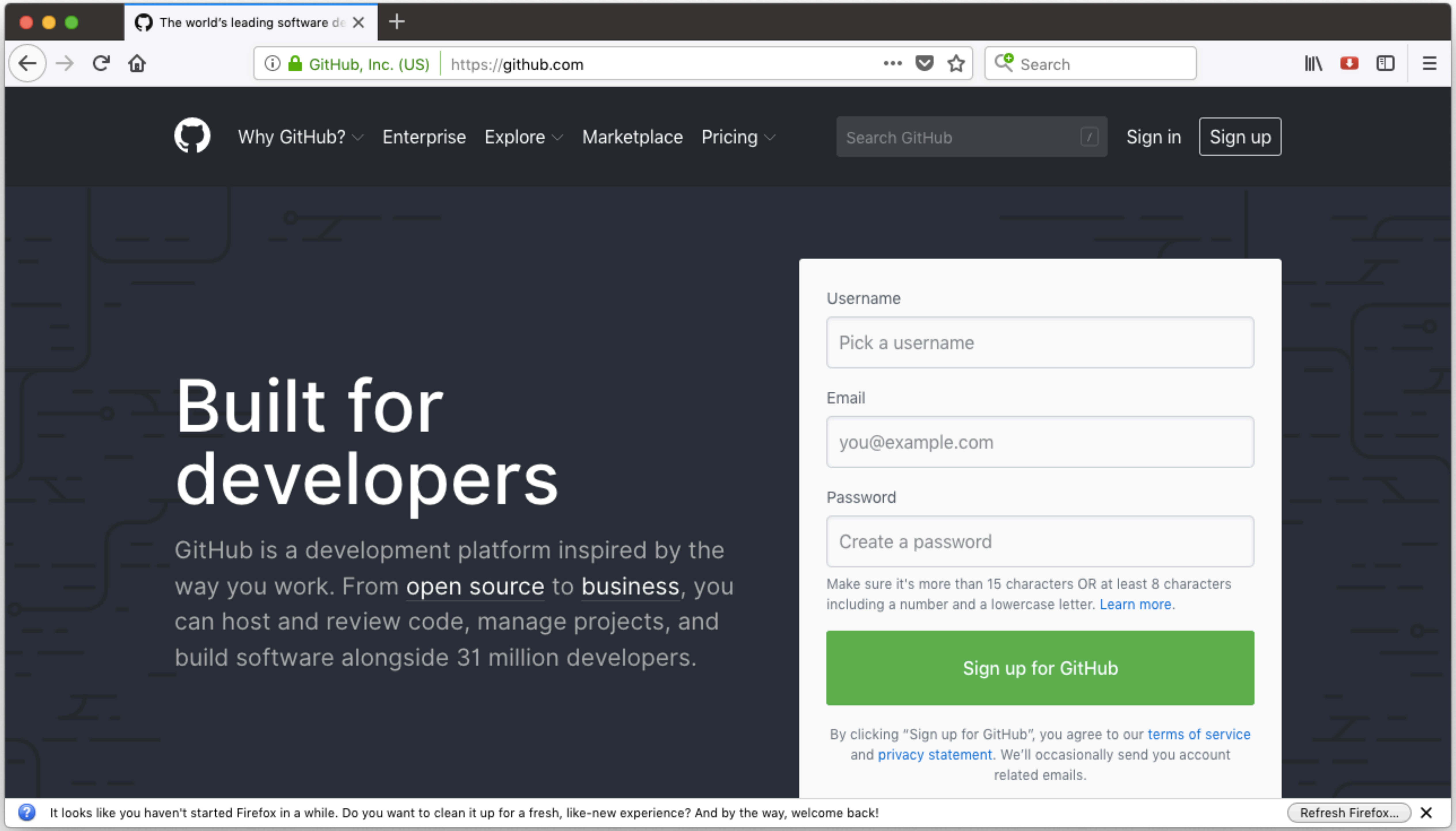
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Name	Size	Modified
data.Rproj	205 B	Mar 1, 2019, 3:03 PM
data1.csv	11.8 KB	Mar 1, 2019, 3:03 PM
data2.csv	19.3 KB	Mar 1, 2019, 3:03 PM
data3.csv	34 KB	Mar 1, 2019, 3:03 PM
image.png	143.8 KB	Mar 1, 2019, 3:03 PM
install.R	138 B	Mar 1, 2019, 3:03 PM
mixed models.Rmd	3 KB	Mar 1, 2019, 3:03 PM
mixed_models.html	1.2 MB	Mar 1, 2019, 3:03 PM
runtime.txt	13 B	Mar 1, 2019, 3:03 PM

https://mybinder.org/v2/gh/ajstewartlang/Turing_way2/master?urlpath=rstudio
```



https://binderhub.readthedocs.io/en/latest/index.html

## Step 1 - Set up a GitHub account



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 31 million developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

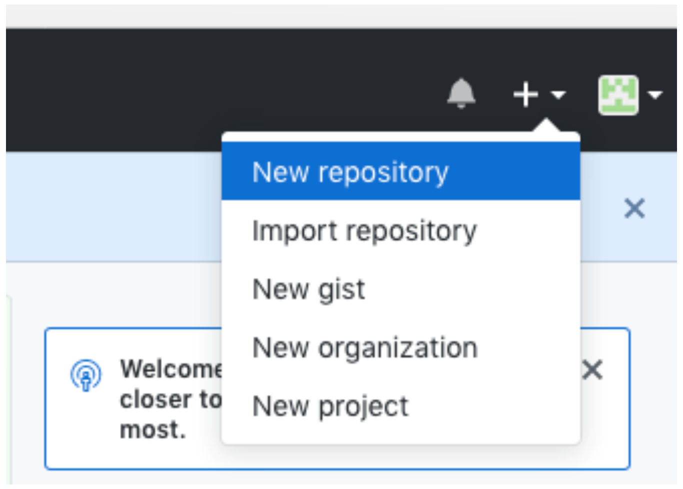
Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

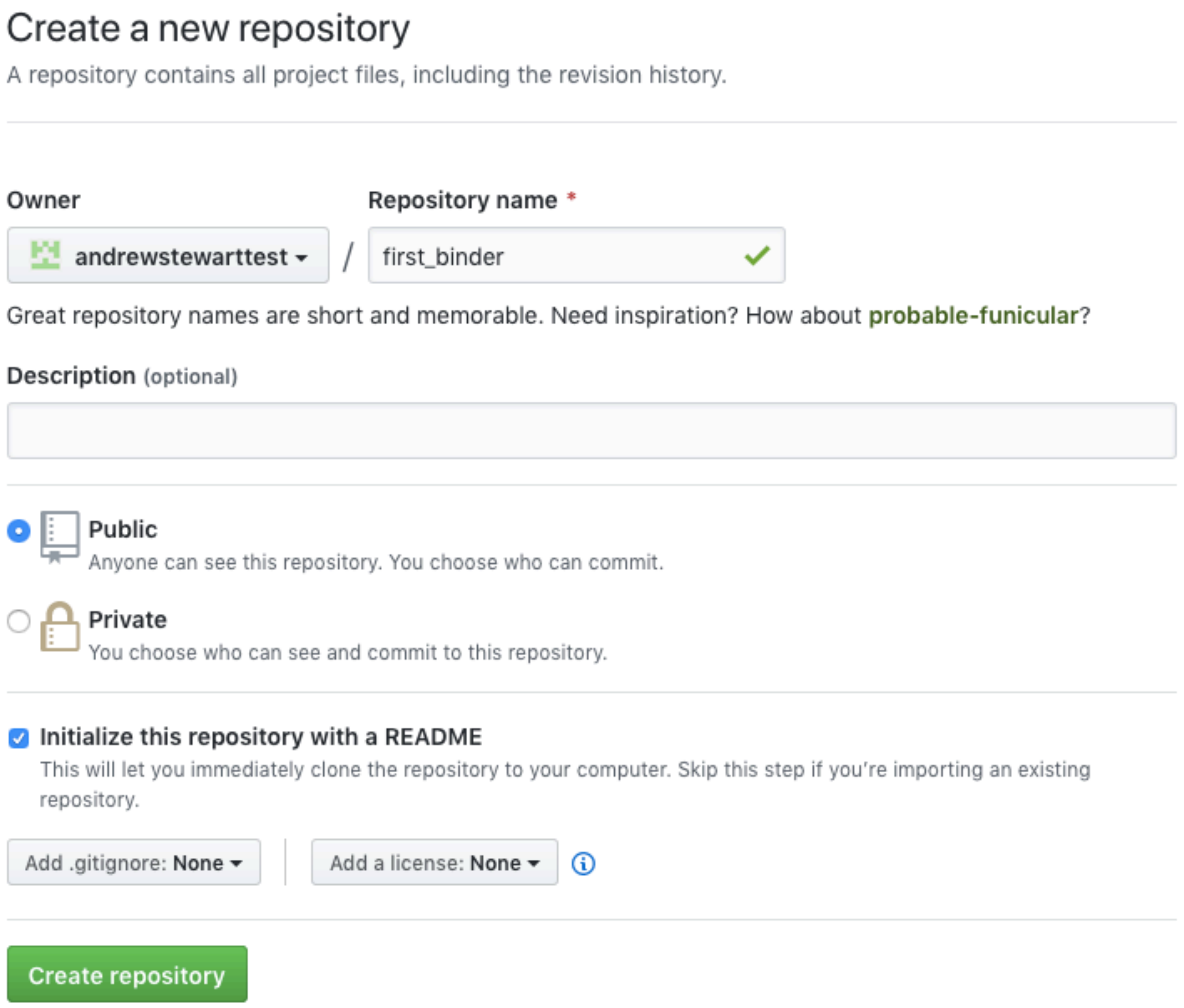
Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our terms of service and privacy statement. We'll occasionally send you account related emails.

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

# Step 2 - Create a new repository





Create a new repository

A repository contains all project files, including the revision history.

Owner: andrewstewarttest / Repository name *: first_binder ✓

Great repository names are short and memorable. Need inspiration? How about probable-funicular?

Description (optional)

[✓ Public] Anyone can see this repository. You choose who can commit.

Private — You choose who can see and commit to this repository.

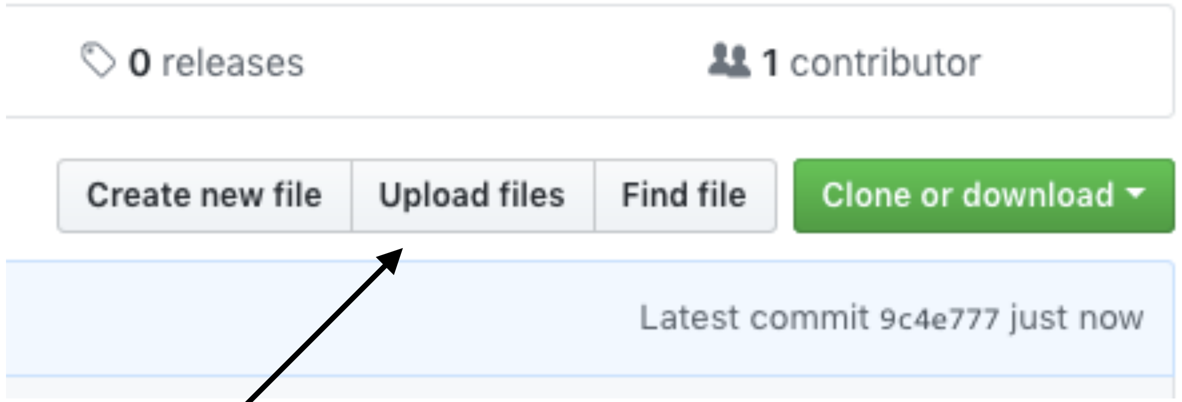
☑ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

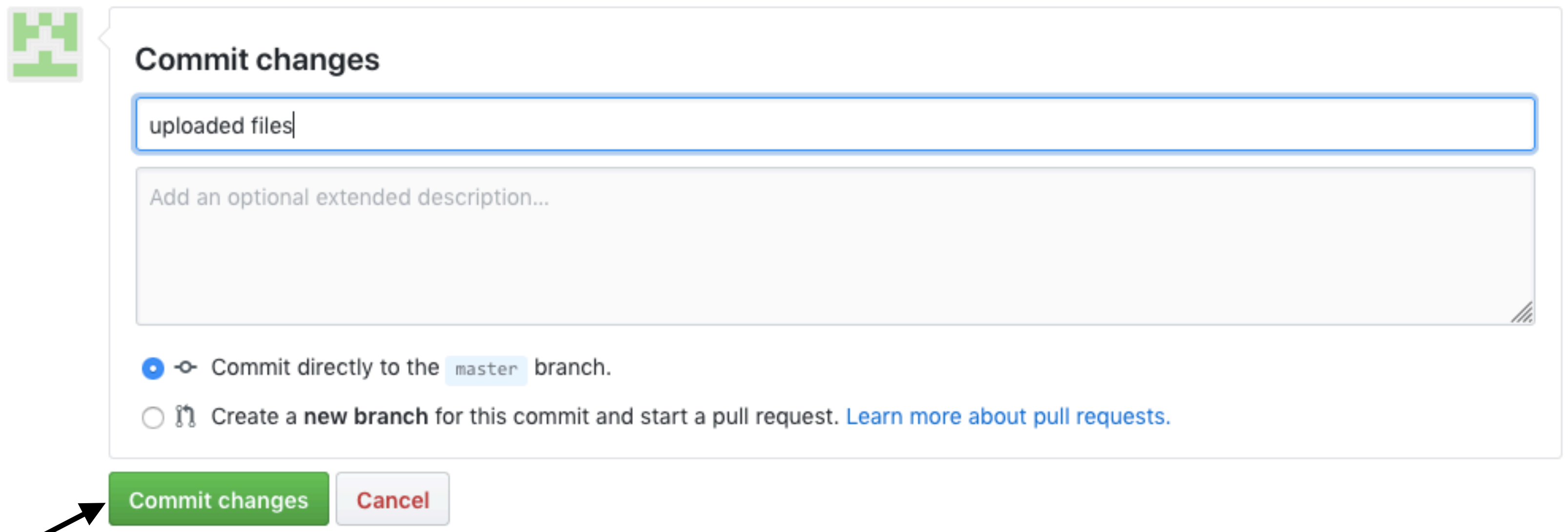
Add a license: None

Create repository

# Step 3 - Upload your R script and data and make your first "Commit"



Click here to upload


Commit changes

uploaded files

Add an optional extended description...

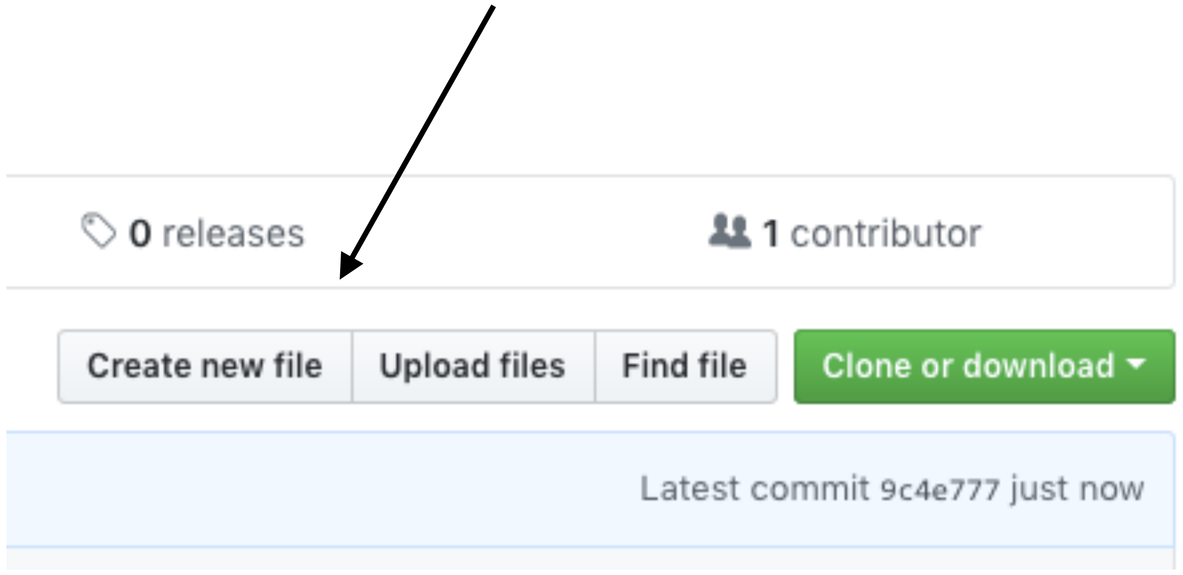
● Commit directly to the `master` branch.

○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

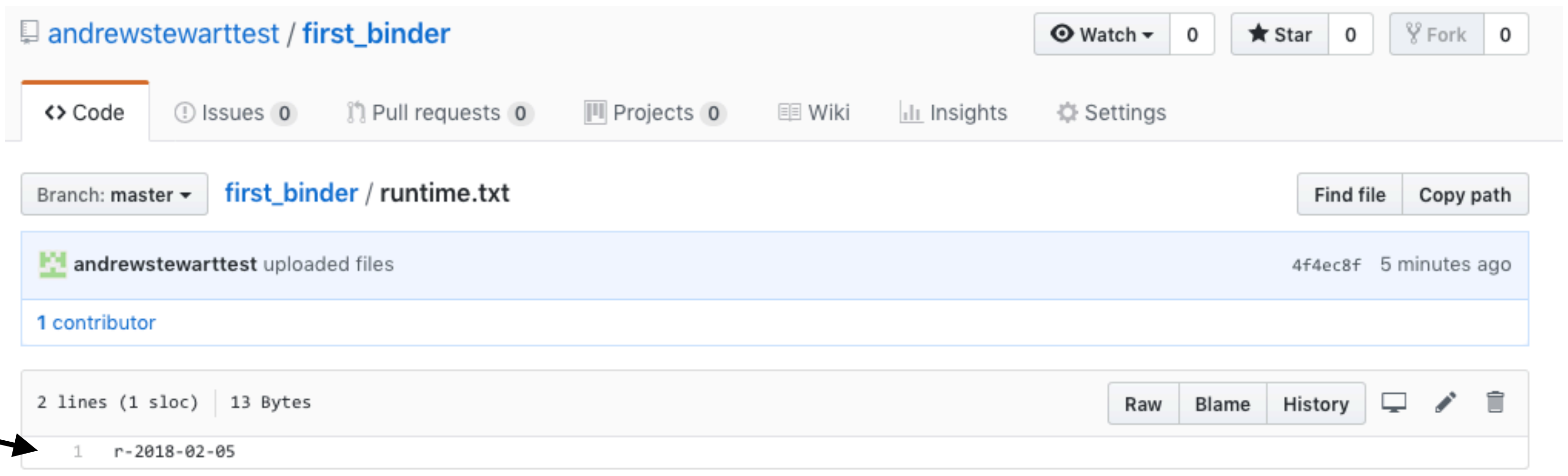
Click here to Commit

## Step 3 - Upload your R script and data and make your first “Commit”

- We need two other files at this point - one is called “runtime.txt” and contains the date of R and its associated packages that you want to simulate.
- The other is called “install.R” and contains the list of R packages that need to be installed in order for your script to run.
- To create a new file select “Create new file”



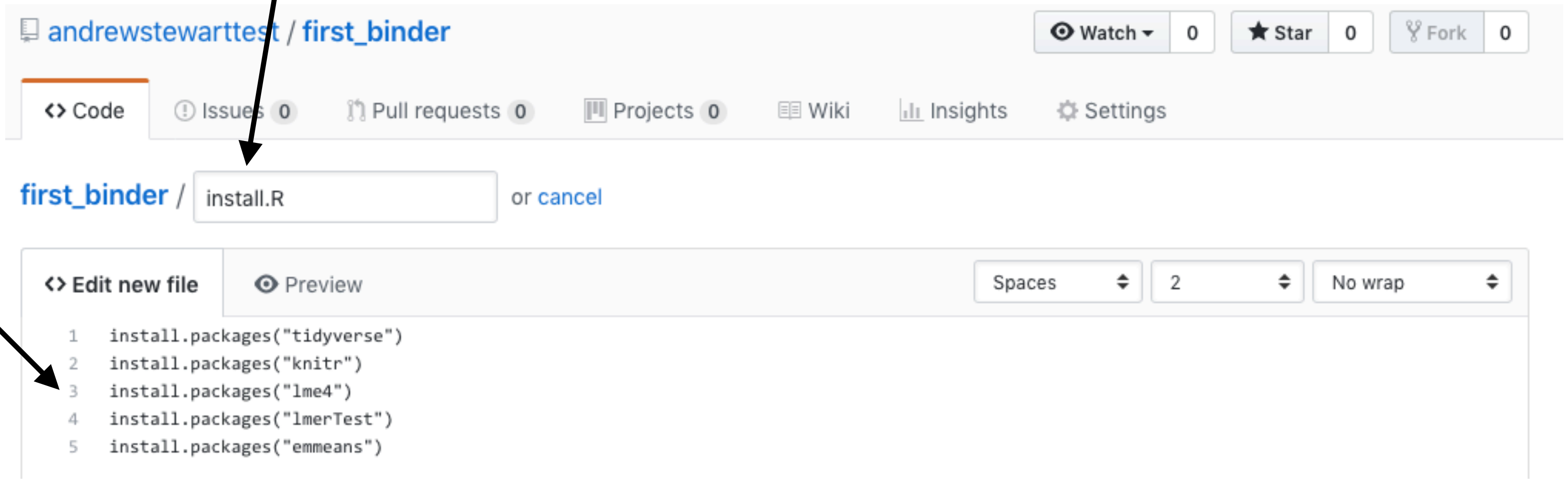
In the runtime.txt file type the date you want in the format r-YYYY-MM-DD



```
r-2018-02-05
```

Name your file

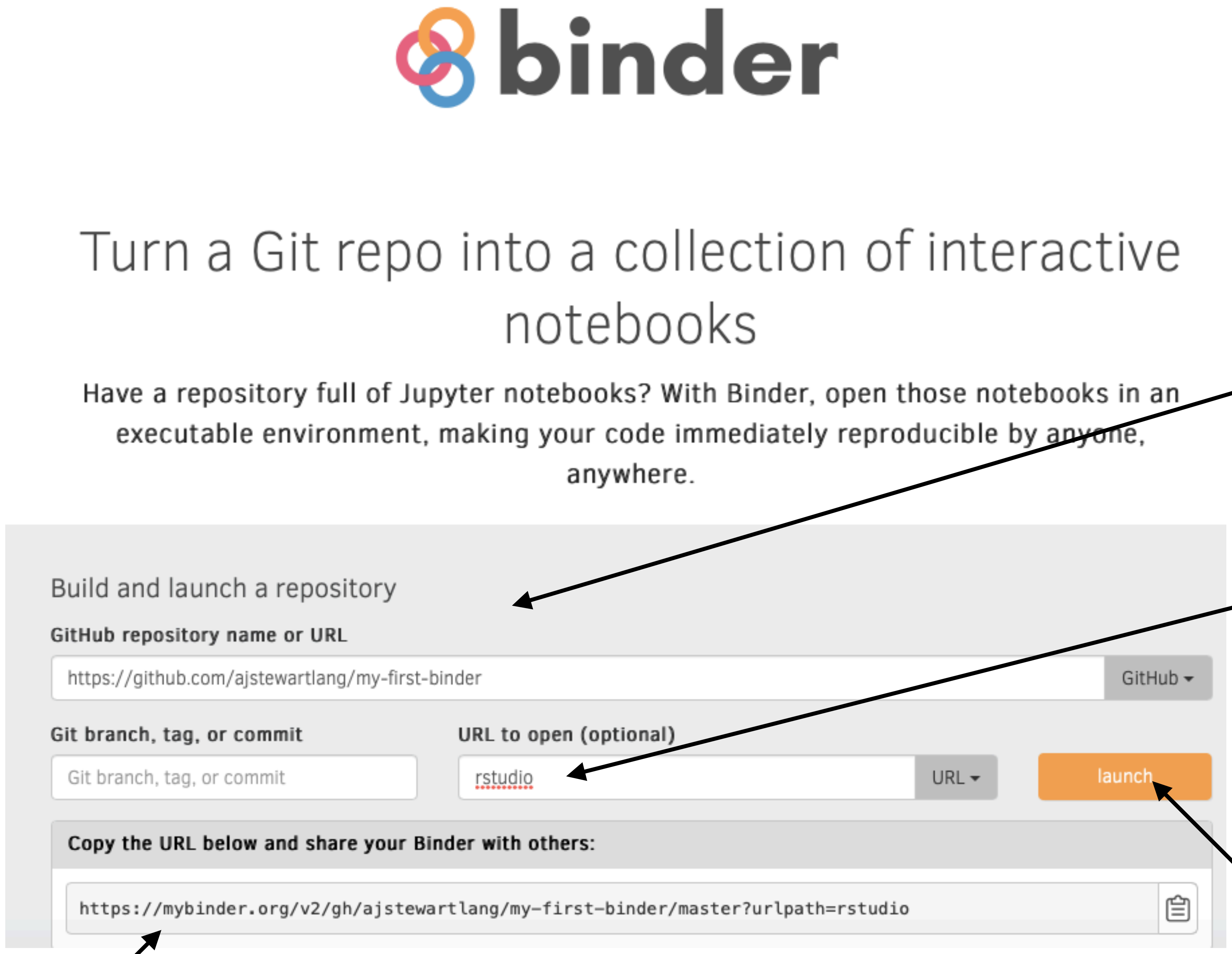
List your packages like this in the install.R file



```
install.packages("tidyverse")
install.packages("knitr")
install.packages("lme4")
install.packages("lmerTest")
install.packages("emmeans")
```

Don’t forget to click “Commit” after you’ve created each file!

Step 5 - Now we need to link our repo to Binder (mybinder.org)



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

Build and launch a repository

GitHub repository name or URL: https://github.com/ajstewartlang/my-first-binder

Git branch, tag, or commit

URL to open (optional): rstudio

Copy the URL below and share your Binder with others:

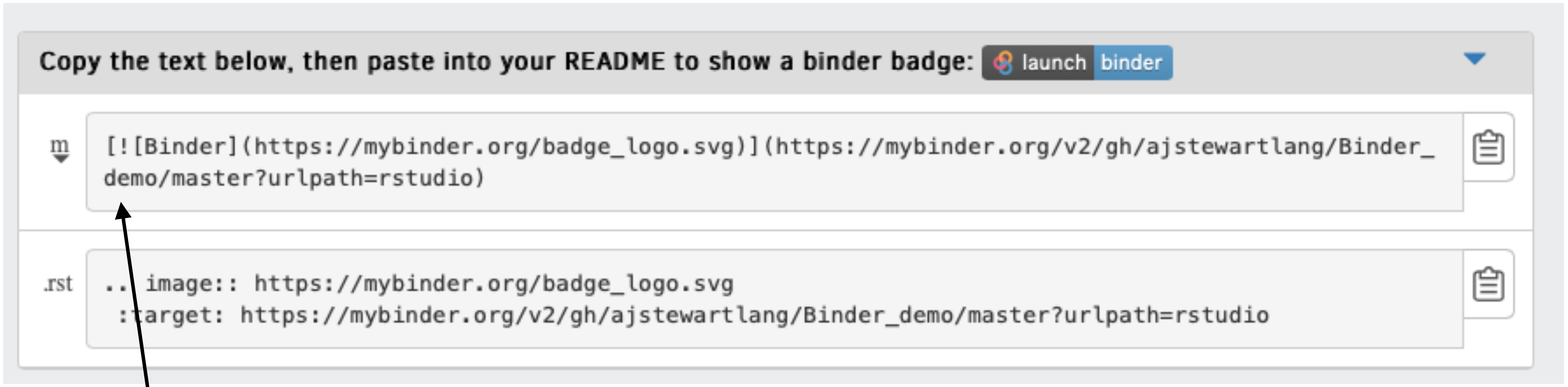
```
https://mybinder.org/v2/gh/ajstewartlang/my-first-binder/master?urlpath=rstudio
```

1. Paste the link to your repo here.

2. Type rstudio here and select "URL"

3. Then click on "launch"

4. This is the URL to share with others.



Copy the text below, then paste into your README to show a binder badge: launch binder

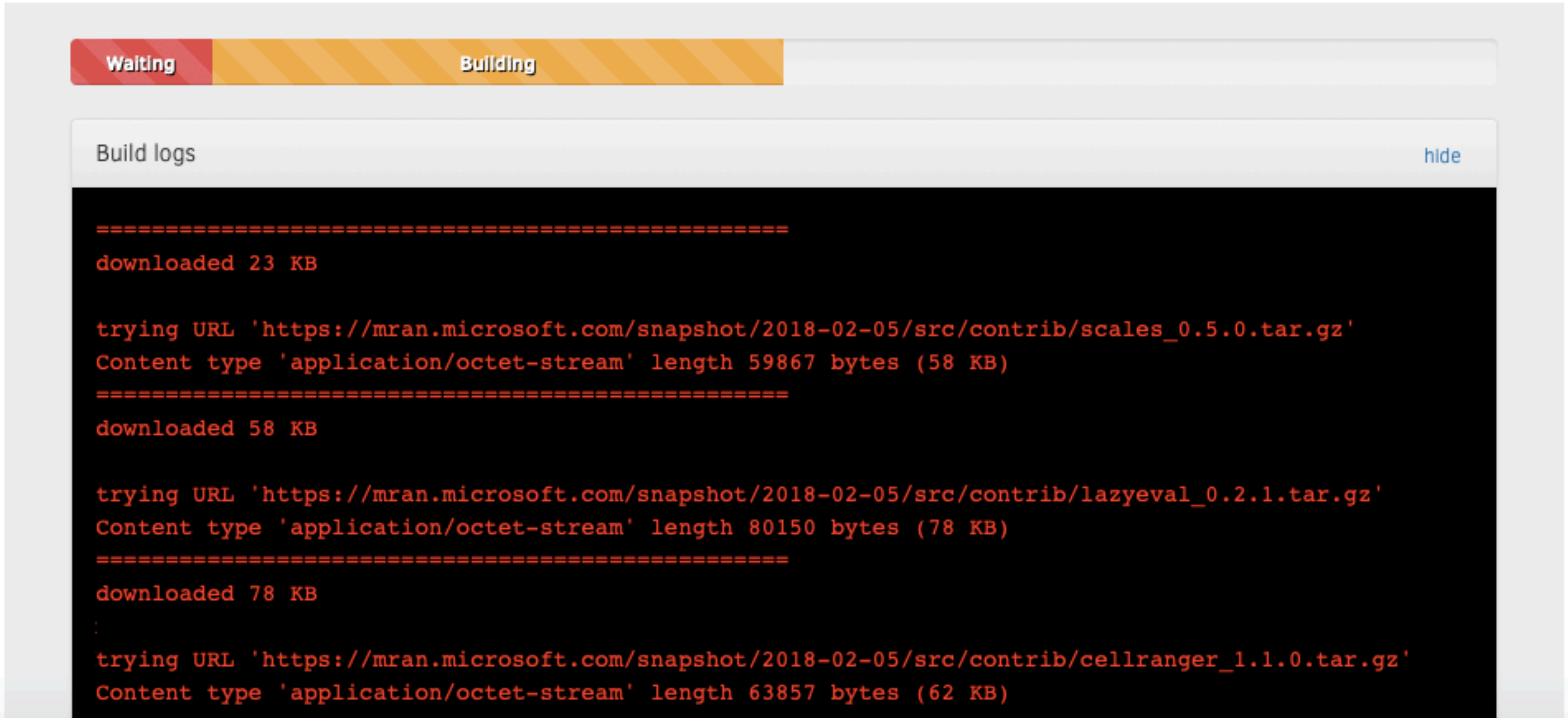
```
[![Binder](https://mybinder.org/badge_logo.svg)](https://mybinder.org/v2/gh/ajstewartlang/Binder_demo/master?urlpath=rstudio)
```

.rst

```
.. image:: https://mybinder.org/badge_logo.svg
 :target: https://mybinder.org/v2/gh/ajstewartlang/Binder_demo/master?urlpath=rstudio
```

- Paste this code into your GitHub repo README.md - you'll then be able to click on the 'launch binder' button in your repository to launch the actual binder once it has been built - makes it easy for others to go from you GitHub repo to your code running in Binder.

Once you click ‘Launch’…

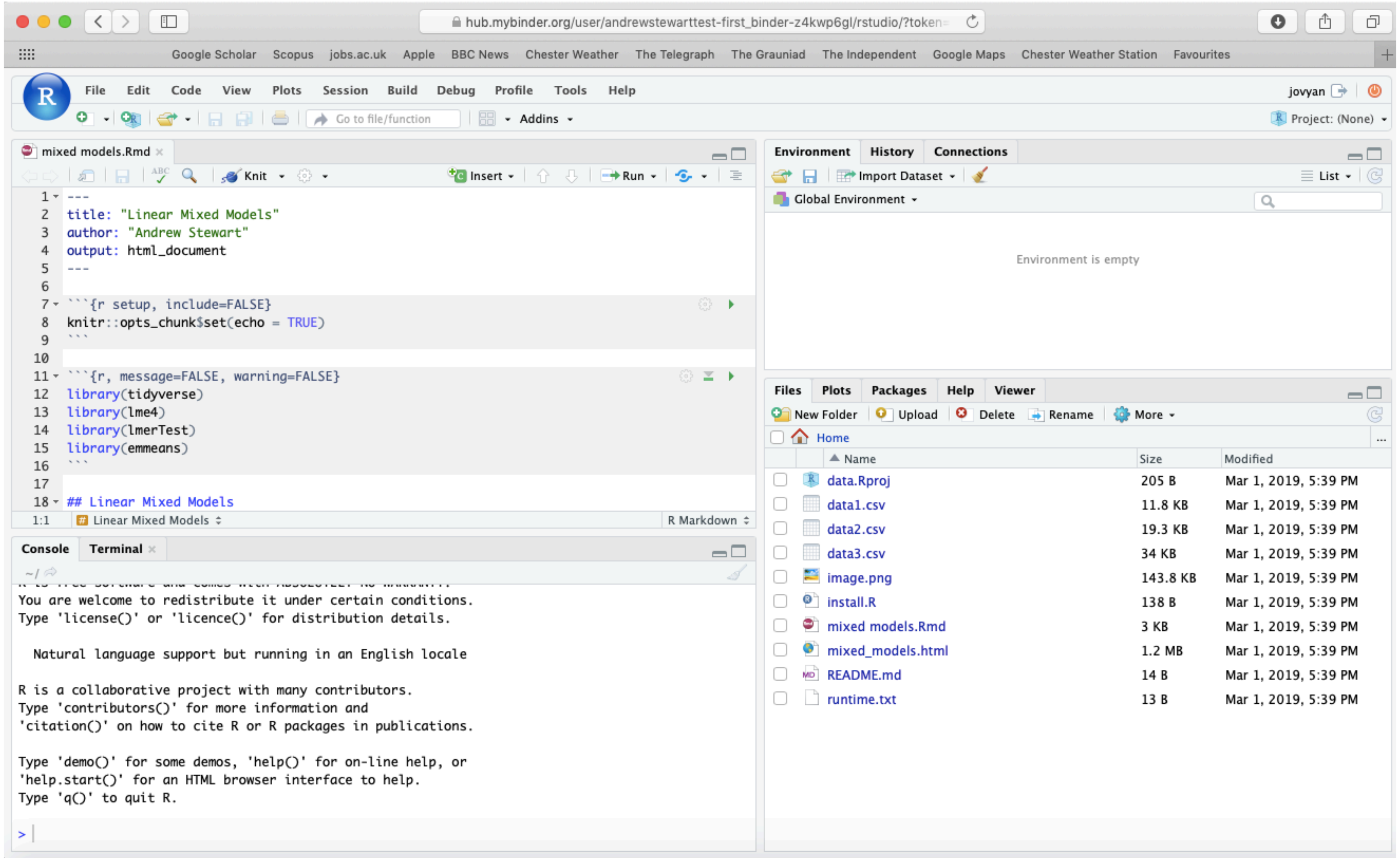


```
==================================================
downloaded 23 KB

trying URL 'https://mran.microsoft.com/snapshot/2018-02-05/src/contrib/scales_0.5.0.tar.gz'
Content type 'application/octet-stream' length 59867 bytes (58 KB)
==================================================
downloaded 58 KB

trying URL 'https://mran.microsoft.com/snapshot/2018-02-05/src/contrib/lazyeval_0.2.1.tar.gz'
Content type 'application/octet-stream' length 80150 bytes (78 KB)
==================================================
downloaded 78 KB

trying URL 'https://mran.microsoft.com/snapshot/2018-02-05/src/contrib/cellranger_1.1.0.tar.gz'
Content type 'application/octet-stream' length 63857 bytes (62 KB)
```

You can check the progress of the build by clicking on the “Build logs” bar.

- If Binder can find an image that you’ve built previously, it will simply launch that.
- If you’ve made changes to your GitHub repo, it will rebuild the Docker image and create a new Binder.
- Either way, once Binder launches you get the following in your browser (even on mobile devices so you can even R away on your phone)…

And then…


hub.mybinder.org/user/andrewstewarttest-first_binder-z4kwp6gl/rstudio/?token=
Google Scholar Scopus jobs.ac.uk Apple BBC News Chester Weather The Telegraph The Grauniad The Independent Google Maps Chester Weather Station Favourites
File Edit Code View Plots Session Build Debug Profile Tools Help
jovyan
Go to file/function
Addins
Project: (None)
mixed models.Rmd
Knit
Insert
Run
Environment History Connections
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Upload Delete Rename More
Home
1:1
Linear Mixed Models
R Markdown
Console Terminal
~/

~~~
---
title: "Linear Mixed Models"
author: "Andrew Stewart"
output: html_document
---

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

```{r, message=FALSE, warning=FALSE}
library(tidyverse)
library(lme4)
library(lmerTest)
library(emmeans)
```

Linear Mixed Models
```

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Name	Size	Modified
data.Rproj	205 B	Mar 1, 2019, 5:39 PM
data1.csv	11.8 KB	Mar 1, 2019, 5:39 PM
data2.csv	19.3 KB	Mar 1, 2019, 5:39 PM
data3.csv	34 KB	Mar 1, 2019, 5:39 PM
image.png	143.8 KB	Mar 1, 2019, 5:39 PM
install.R	138 B	Mar 1, 2019, 5:39 PM
mixed models.Rmd	3 KB	Mar 1, 2019, 5:39 PM
mixed_models.html	1.2 MB	Mar 1, 2019, 5:39 PM
README.md	14 B	Mar 1, 2019, 5:39 PM
runtime.txt	13 B	Mar 1, 2019, 5:39 PM
```

https://mybinder.org/v2/gh/ajstewartlang/SIPS_visualisation_6/master?urlpath=rstudio

A few other things…

- Installing the entire Tidyverse in a Binder can take a long time - better to install only the packages you use (e.g., ggplot2, dplyr, readr etc.) - this will also ensure the packages are consistent with the date in your runtime.txt file.
- Even with just a couple of packages it can take ~15 minutes or so for your Binder to be built.
- To change the version of R that Binder builds (to 3.6 say) change the runtime.txt file to "`r-3.6-YYYY-MM-DD`"

A few other things…

- Some R packages need system-level packages to also be installed - you can do that via an additional apt.txt file which lists those packages - this is used by apt-install to install those packages from the Ubuntu apt repository.
- You can close your laptop if Binder is taking too long - the image and your Binder will continue to be built in the Cloud. And it’s always a good excuse for another coffee…

For Ultimate Reproducibility

- Make sure you have updated all your packages before you run your script.
- Build your Binder and specify the day your ran your analysis in the runtime.txt file - and add a version of R if you don't want it to default to 3.5
- Patience while your Binder builds…

Your turn to build a Binder!
Take a script you've already
written, and start at slide 12...

Advanced...

- If you use Binder via the repo2docker route, you will notice that some Binders take quite a long time to build initially - oftentimes this happens when you're wanting to install the entire tidyverse or lots of packages with dependencies..

- By writing a Dockerfile, you're able to pull a pre-built Docker (Rocker) image into Binderhub so it will launch a lot more quickly. Typically this image will include the Tidyverse packages (and others) so things don't need to be built on-the-fly.

- More about Rocker here:

  `https://www.rocker-project.org`

How?

- The `holepunch::` package by Karthik Ram allows you to write a Dockerfile, and build your GitHub repo from within RStudio.

`https://github.com/karthik/holepunch`

- The Dockerfile will capture the date of the last time you updated any file of your project and then pull a pre-built Rocker image associated with that date into Binderhub when you launch Binder.

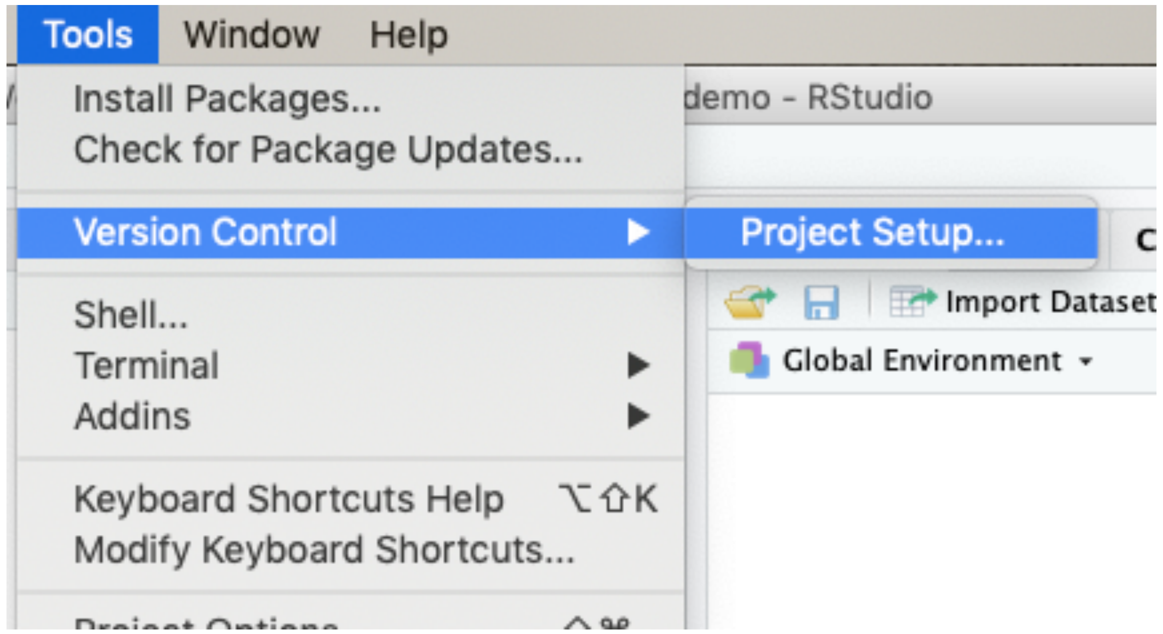
- You need to initialise the local R project folder with Git version control (or clone a repo from GitHub).

- First install the latest version of "holepunch" from GitHub - you may be prompted to update some other packages - please do so.

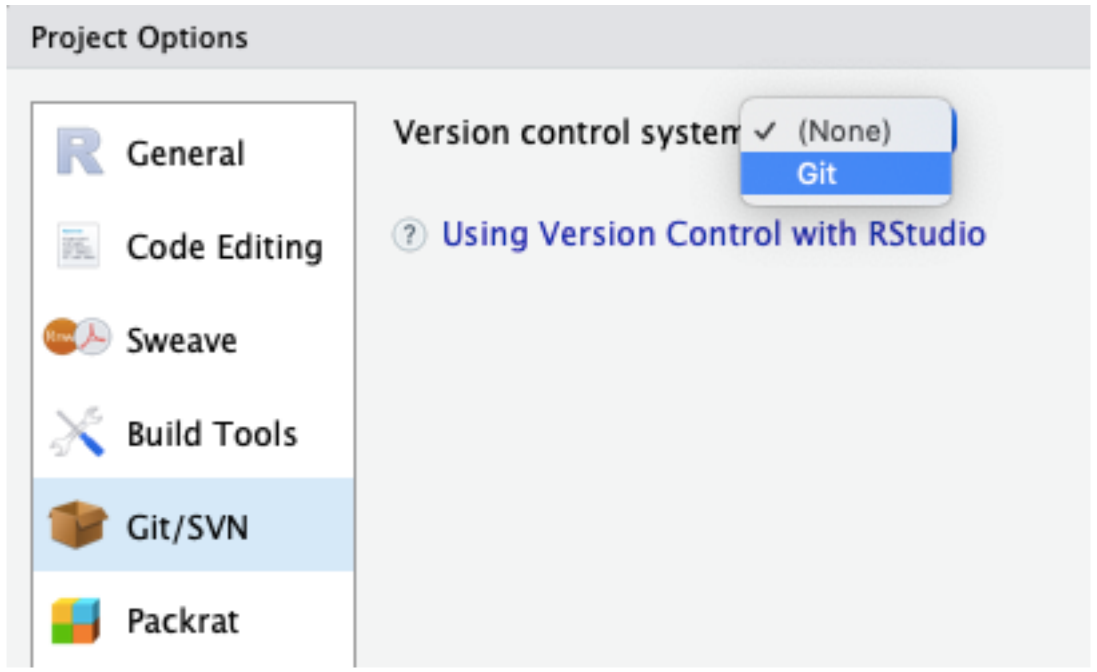
  ```
  > remotes::install_github("karthik/holepunch")

  > library(holepunch)
  ```

- You can either clone a pre-existing repo from GitHub, or create a new R Project and turn that folder into a git version controlled repo - in which case...

First we need to ensure our folder associated with a project is a repository with git version control.

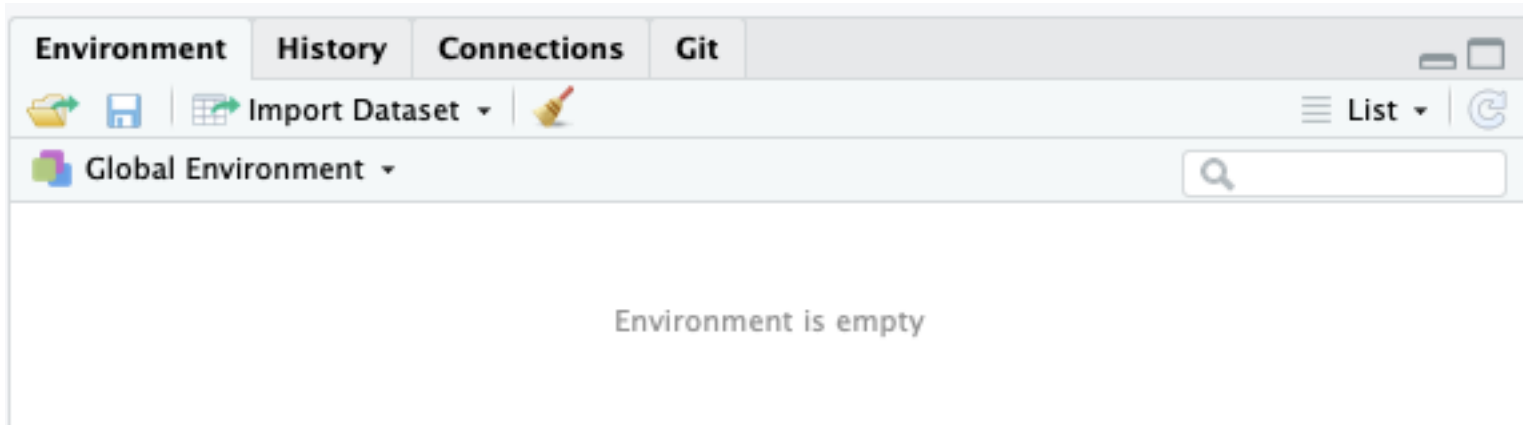


Select Git - you will need to restart your R session at this point.



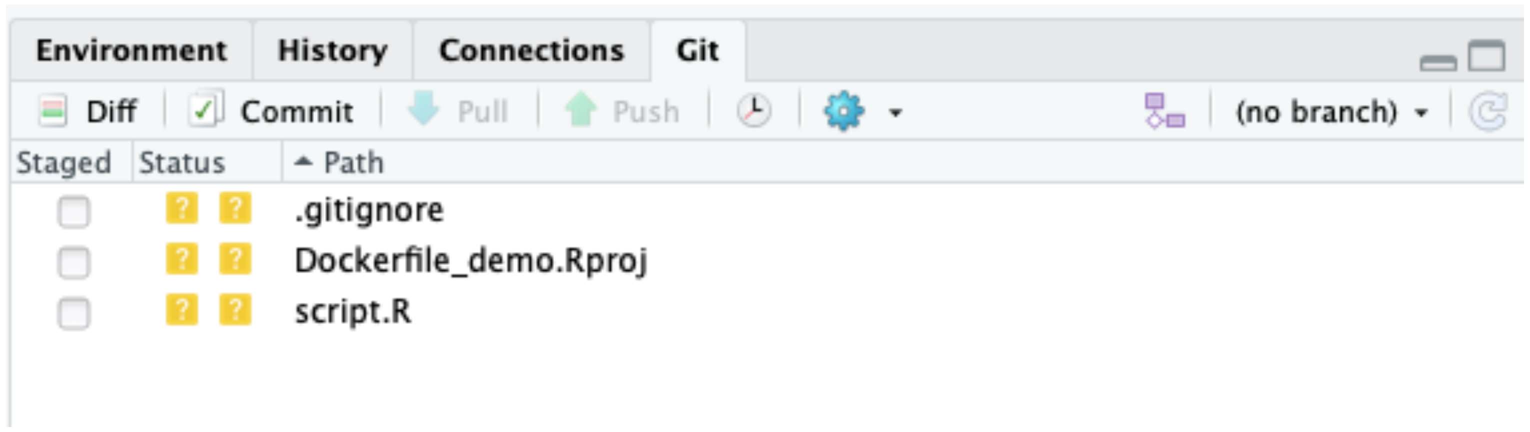
Version control system: [✓ Git]

When you restart, you'll see you now have a new Git tab in your Environment window.



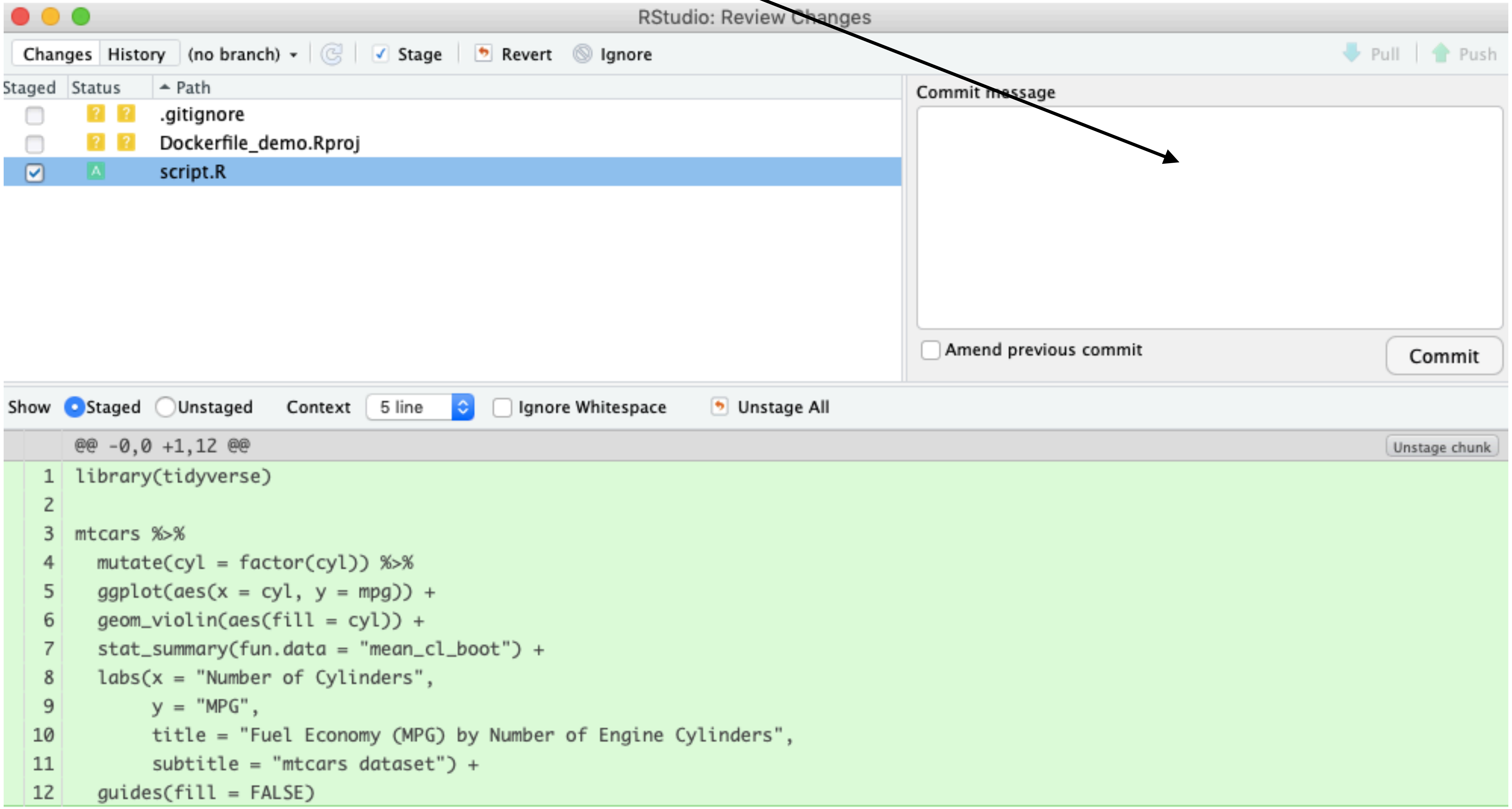
Environment is empty

If you click on the Tab you'll see the contents of your folder.



Staged	Status	Path
☐	? ?	.gitignore
☐	? ?	Dockerfile_demo.Rproj
☐	? ?	script.R

In this example, I've selected the script.R file, saved it (you can't Commit without saving first), and then clicked on Commit - the following window now appears. Write a meaningful Commit message in here and the click on the 'Commit' button underneath.



Staged	Status	Path
☐	? ?	.gitignore
☐	? ?	Dockerfile_demo.Rproj
✓	A	script.R

```
@@ -0,0 +1,12 @@
library(tidyverse)

mtcars %>%
  mutate(cyl = factor(cyl)) %>%
  ggplot(aes(x = cyl, y = mpg)) +
  geom_violin(aes(fill = cyl)) +
  stat_summary(fun.data = "mean_cl_boot") +
  labs(x = "Number of Cylinders",
       y = "MPG",
       title = "Fuel Economy (MPG) by Number of Engine Cylinders",
       subtitle = "mtcars dataset") +
  guides(fill = FALSE)
```

- Once you have a git repository set up locally, push it to GitHub either within RStudio or via GitHub desktop (or the command line).

- Alternatively, you can set up a repo on GitHub and then clone it locally - but you'll still need to push changes to GitHub.

- You can then write your Dockerile via the console in RStudio.

Step 1 - write a Dockerfile

```
> write_dockerfile(maintainer = "your_name",
r_date = "2019-06-27")
```

- If you leave out the date, `holepunch` will create a Dockerfile associated with the date you last changed your repo. It uses the version of R and packages on MRAN associated with the date you specify (or the last change date if you don't specify an actual date).

Step 2 - generate a binder badge

```
> generate_badge()
```

- this will generate the code you need to paste in your repo README that will launch Binder upon clicking.

Step 3 - build your binder

```
> build_binder()
```

- will start building your Binder in the background - this will still be much quicker that building from scratch as the Dockerfile will pull a Rocker image and associated R packages for the date you specified during `write_dockerfile()`

Any caveats?

- `holepunch::` is very much still in development but Karthik responds super quickly to issues, enhancement suggestions, and bug reports - and it will be on CRAN (and therefore more stable) sooner rather than later...

- Great `rstudio::conf 2019` video of Karthik talking about reproducibility in general and `holepunch`::

https://resources.rstudio.com/rstudio-conf-2019/a-guide-to-modern-reproducible-data-science-with-r

Xaringan

- The xaringan package for R allows you to write presentations in Markdown which can then be rendered as .html files.
- Allows you to include R analysis (data, code, and output) in a presentation without any cutting and pasting.
- Also allows you to rebuild a presentation at the press of a button if your analysis changes, you add more data etc.
- Allows for fully reproducible and open presentations!

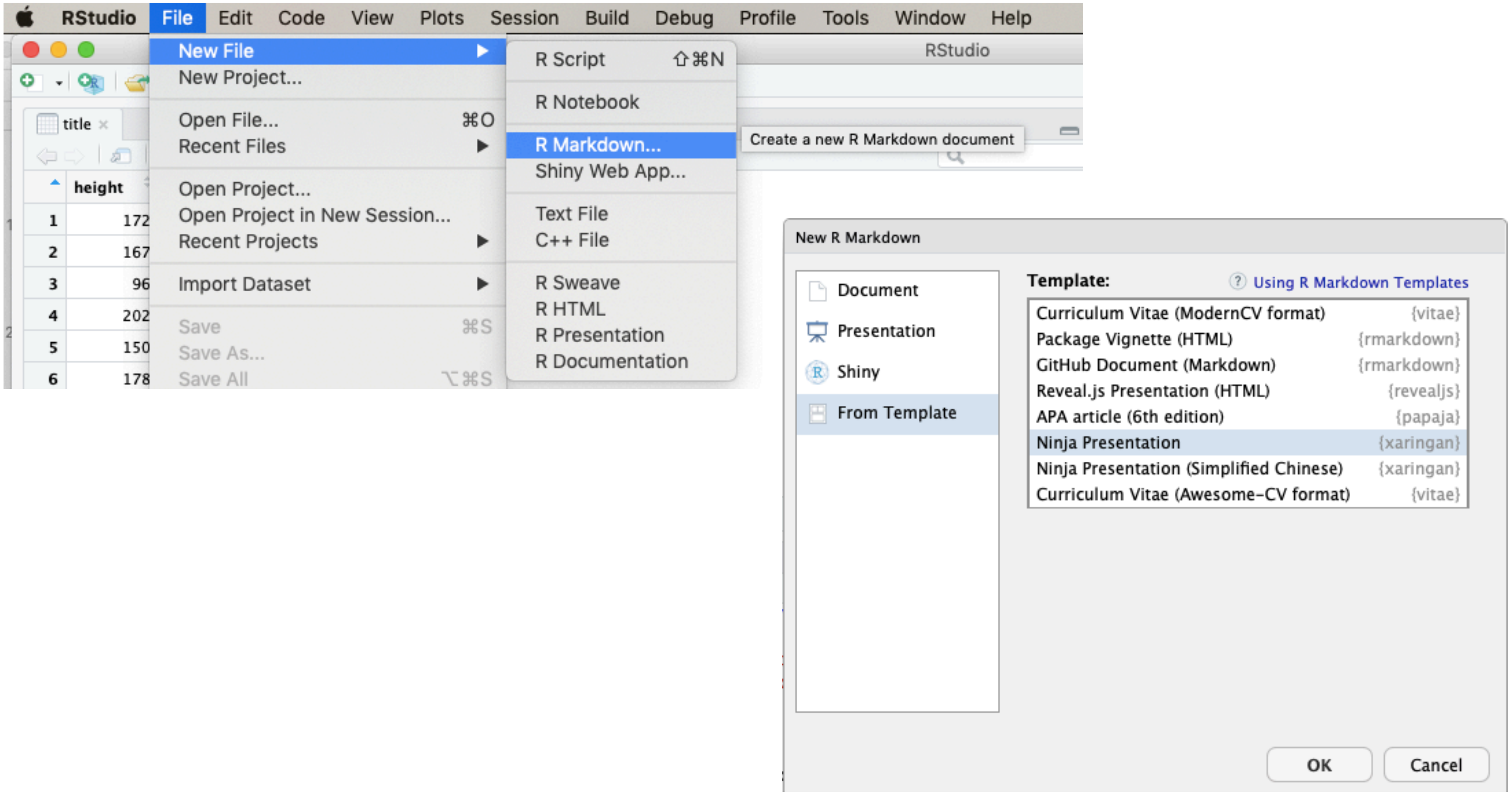
https://bookdown.org/yihui/rmarkdown/xaringan.html

An example...

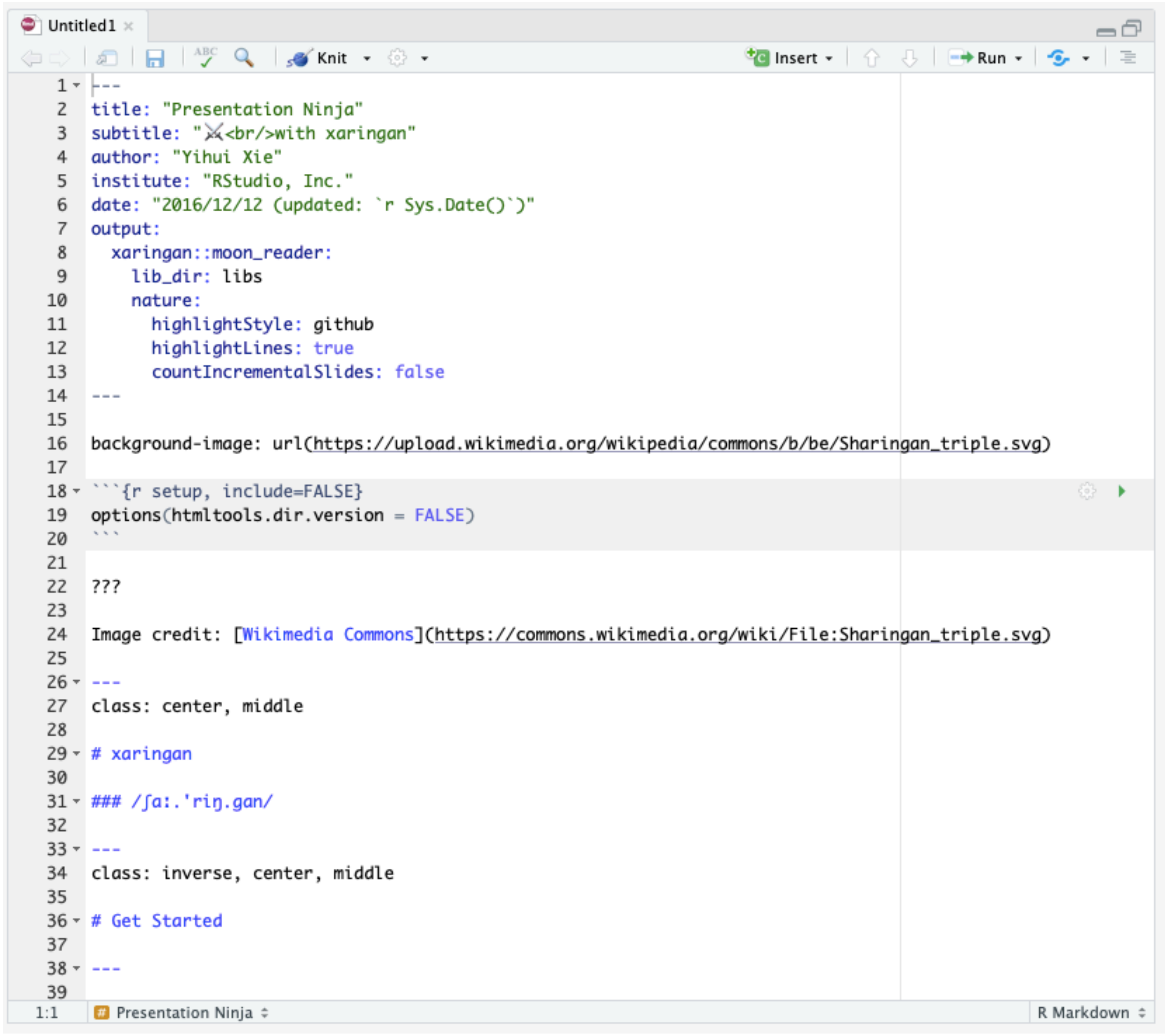
https://ajstewartlang.github.io/SIPS_2019/SIPS_presentation.html#51

- First, install the package xaringan.

- Then...



Curriculum Vitae (ModernCV format)	{vitae}
Package Vignette (HTML)	{rmarkdown}
GitHub Document (Markdown)	{rmarkdown}
Reveal.js Presentation (HTML)	{revealjs}
APA article (6th edition)	{papaja}
Ninja Presentation	{xaringan}
Ninja Presentation (Simplified Chinese)	{xaringan}
Curriculum Vitae (Awesome-CV format)	{vitae}



~~~
---
title: "Presentation Ninja"
subtitle: "⚔<br/>with xaringan"
author: "Yihui Xie"
institute: "RStudio, Inc."
date: "2016/12/12 (updated: `r Sys.Date()`)"
output:
  xaringan::moon_reader:
    lib_dir: libs
    nature:
      highlightStyle: github
      highlightLines: true
      countIncrementalSlides: false
---

background-image: url(https://upload.wikimedia.org/wikipedia/commons/b/be/Sharingan_triple.svg)

```{r setup, include=FALSE}
options(htmltools.dir.version = FALSE)
```

???

Image credit: [Wikimedia Commons](https://commons.wikimedia.org/wiki/File:Sharingan_triple.svg)

---
class: center, middle

# xaringan

### /ʃaː.'riŋ.gan/

---
class: inverse, center, middle

# Get Started

---

~~~

- Try 'knitting' the template to see what happens...

- You can look at the code at the same time that you cycle through the slides - xaringan is Markdown but with a few extra things that allow you to change the format of your slides...

- Your RMD script can include CSS (Cascading Style Sheets) code which describes how HTML elements are to be displayed.

- Delete everything after from line 15 and paste this at line 15:

```
---

class: center, inverse

# A new slide

Content.
```

- In the header (around line 9) add `seal:false` after the `xaringan::moon_reader:` line - this will allow you to write your own title slide:

```
output:
  xaringan::moon_reader:
    seal: false
```

- You can write text as you do in R Markdown - to make text italics use _ either side like this _*italics*_ and to make text bold use two underscores __**bold**__

- You can use .pull-left[] to enclose the text you want to be presented on the left hand side of a slide and .pull-right[] to enclose text to be presented on the right...

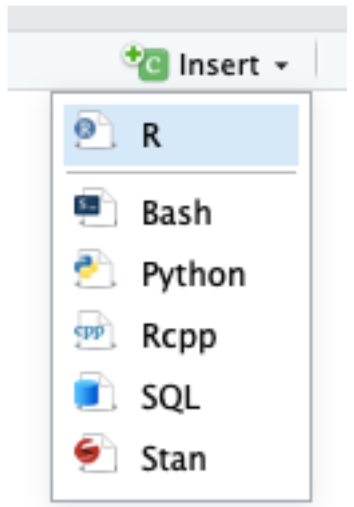
```
---
.pull-left[
here is some text in _italics_
]

.pull-right[
and here is some in  __bold__
]
```

- You can add chunks of R code, and suppress displaying the code (use echo=FALSE to not display the code, message=FALSE to not display messages, warning=FALSE to not display warnings, and eval=FALSE to not run the code).

- See the R Markdown cheatsheet for other options:

```
https://github.com/ajstewartlang/
MRes_Advanced_Data_Skills/blob/master/
R_cheatsheets/R_Markdown%20cheatsheet.pdf
```

- You can insert a chunk of R code by clicking on Insert R.
- Or by clicking CMD-Alt-I (on a Mac) and Ctrl-Alt-I (on a PC).

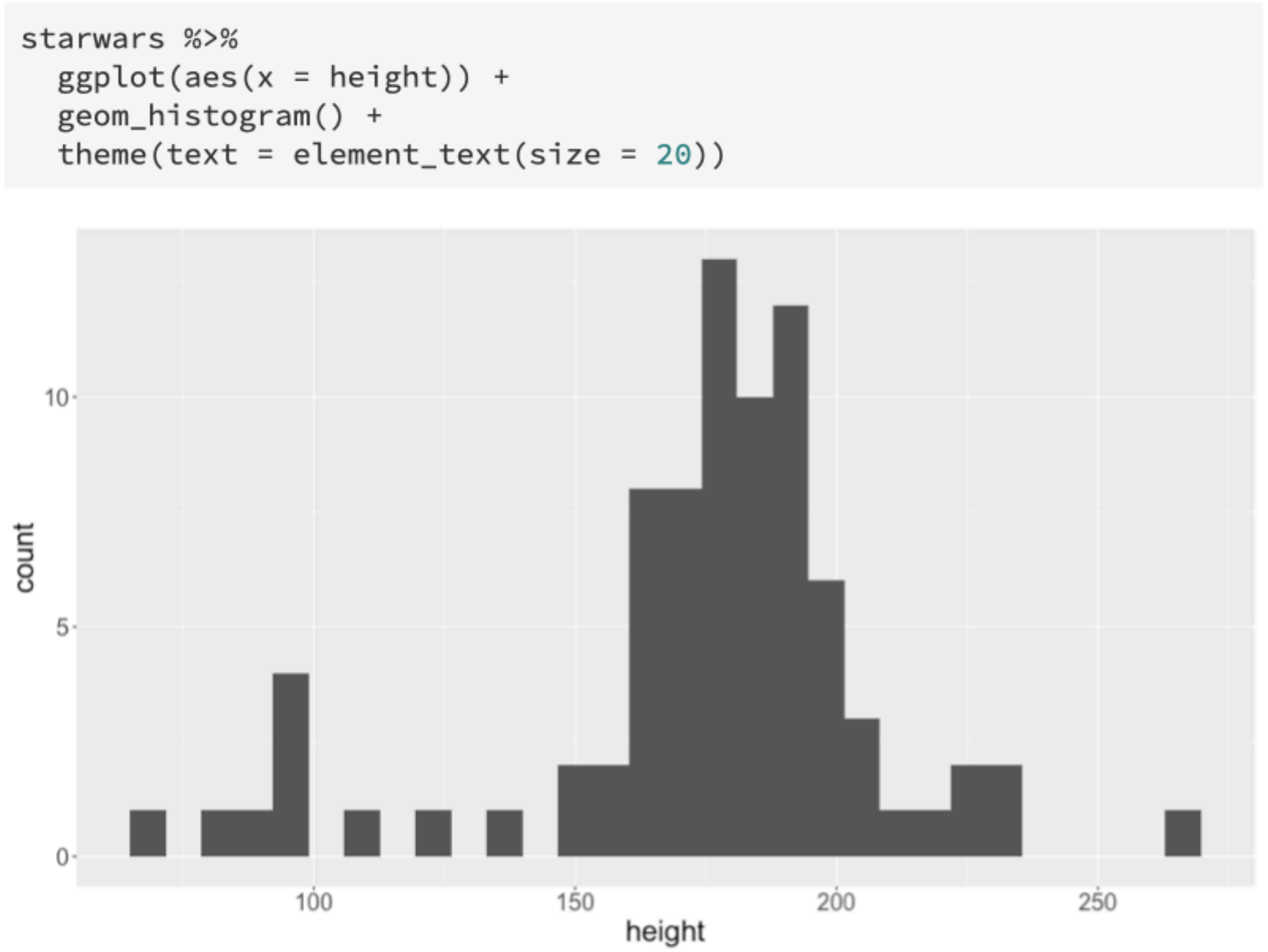


~~~
---
```{r, echo=FALSE, message=FALSE}
library(tidyverse)
```

```{r}
head(starwars)
```
~~~

```r
head(starwars)
```

```
## # A tibble: 6 x 13
##   name  height  mass hair_color skin_color eye_color birth_year gender
##   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr>
## 1 Luke…    172    77 blond      fair       blue            19   male
## 2 C-3PO    167    75 <NA>       gold       yellow         112   <NA>
## 3 R2-D2     96    32 <NA>       white, bl… red             33   <NA>
## 4 Dart…    202   136 none       white      yellow          41.9 male
## 5 Leia…    150    49 brown      light      brown           19   female
## 6 Owen…    178   120 brown, gr… light      blue            52   male
## # … with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

~~~
```{r, warning=FALSE, message = FALSE, fig.width=12}
starwars %>%
  ggplot(aes(x = height)) +
  geom_histogram() +
  theme(text = element_text(size = 20))
```
~~~



```
starwars %>%
  ggplot(aes(x = height)) +
  geom_histogram() +
  theme(text = element_text(size = 20))
```

- You can also add images in formats including .jpg and .png - just make sure you keep the images at the same level as your RMD script, or specify the path needed to find them.

~~~
# Opeth 🤘
```{r, echo=FALSE, out.width="100%"}
knitr::include_graphics("opeth.jpg")
```
~~~

Opeth 🤘

And that's largely it!

A few caveats...

- The first time you use xaringan you will find it slow - it also takes a while to stop thinking in Powerpoint or Keynote terms and start thinking in R and Markdown terms.

- Xaringan presentations are probably most useful when you want to re-run your code without changing the rest of the presentation - you just need to re-knit you Markdown script.

You have a go at writing a brief presentation using xaringan which includes some R code and output...