Week 11 - Binder for fully reproducible research in R (data, code, and computational environment)

Andrew Stewart

Andrew.Stewart@manchester.ac.uk

@ajstewart_lang

https://github.com/ajstewartlang

Week	Topic
1	Introduction, Open Science, and Power
2	Introduction to R
3	Data Wrangling and Visualisation
4	General Linear Model - Regression
5	General Linear Model - Regression
6	No Timetabled Lecture - Reading Week
7	Consolidation Lab
8	General Linear Model - ANOVA
9	General Linear Model - ANOVA
10	Tidy Thursday Data Wrangling & Visualisation Challenge
11	Reproducing your Computational Environment using Binder
12	Dynamic, Reproducible Presentations Using xaringan

<u>Semester 1 Assignments</u>

Data wrangling and visualisation – Due December 5th

ANOVA/ANCOVA – Due January 17th

Open and Reproducible Research

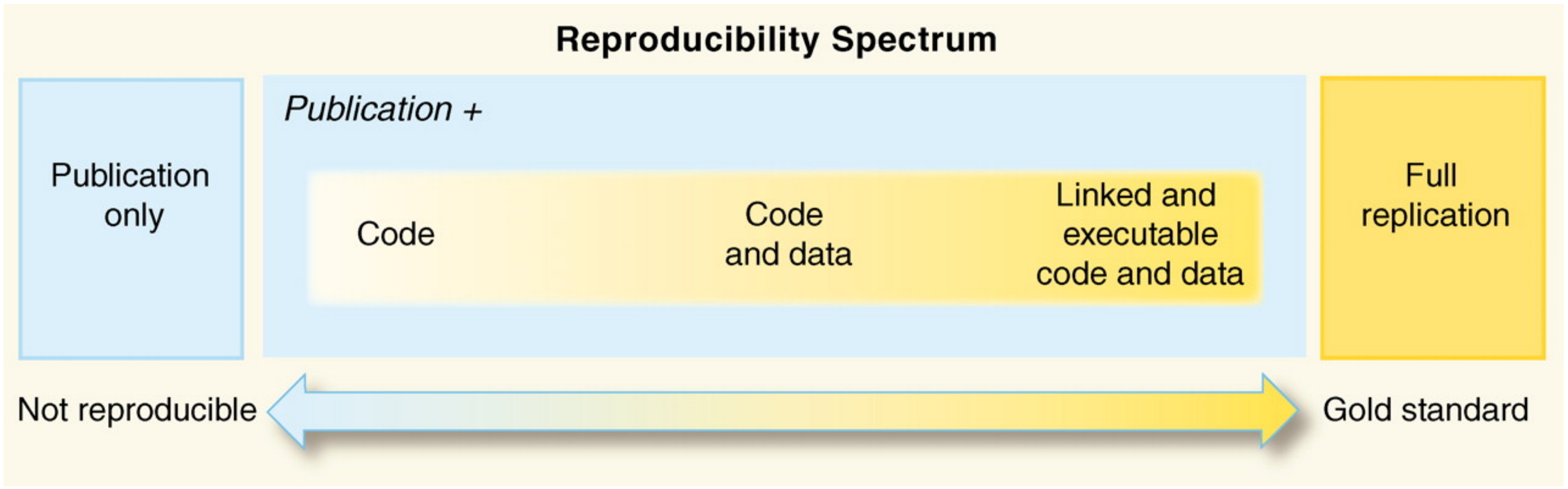
- Shared Data - we already know this is important for reproducibility.
- Shared Code - we already know this is important for reproducibility.
- Shared Computational environment - why is this important and how do we do it?

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

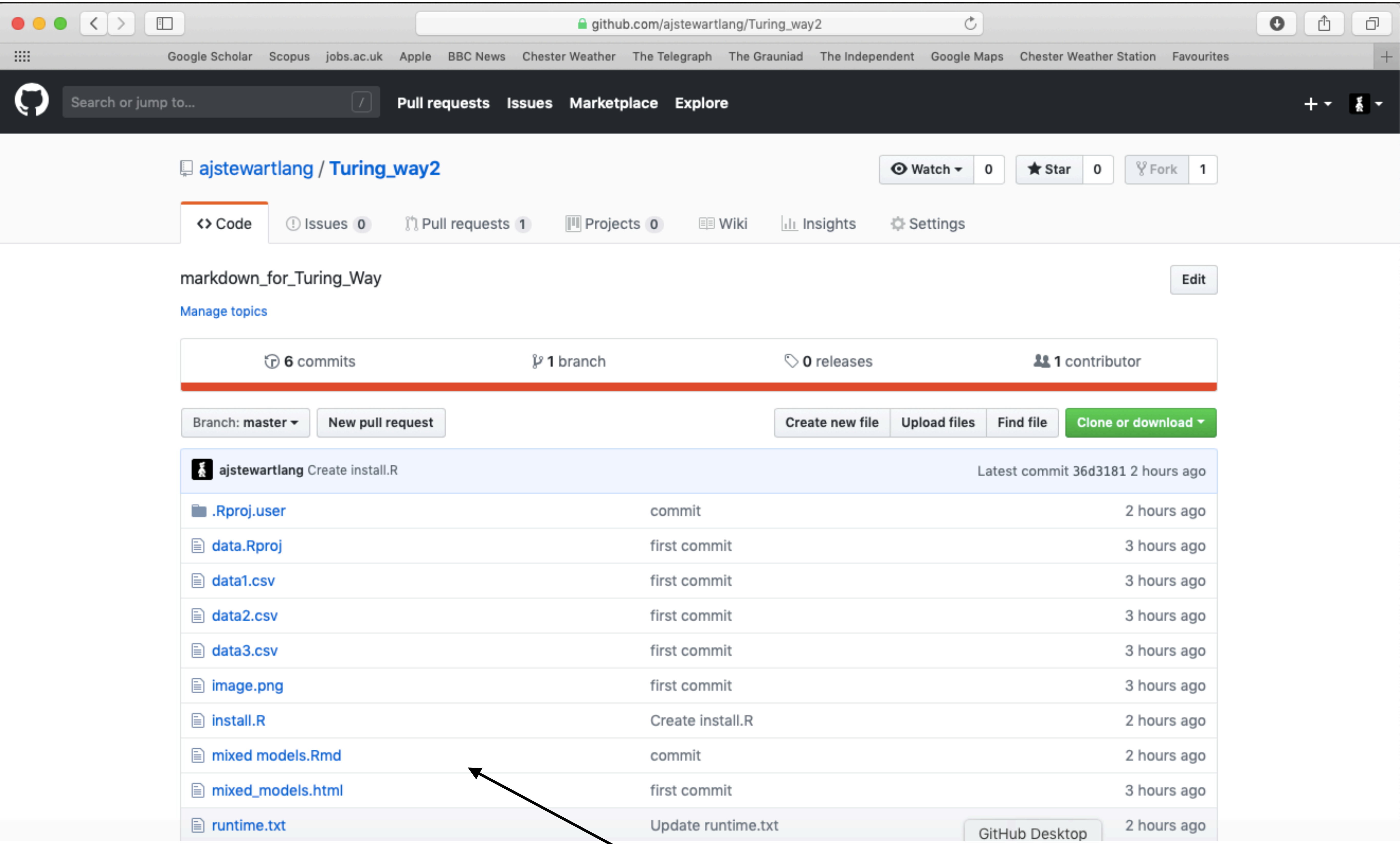
Docker packages your data, code and all its dependencies in the form called a docker container to ensure that your application works seamlessly in any environment.

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

Google Scholar Scopus jobs.ac.uk Apple BBC News Chester Weather The Telegraph The Grauniad The Independent Google Maps Chester Weather Station Favourites

Search or jump to... Pull requests Issues Marketplace Explore

ajstewartlang / Turing_way2

Watch 0 Star 0 Fork 1

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

markdown_for_Turing_Way

Manage topics

6 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

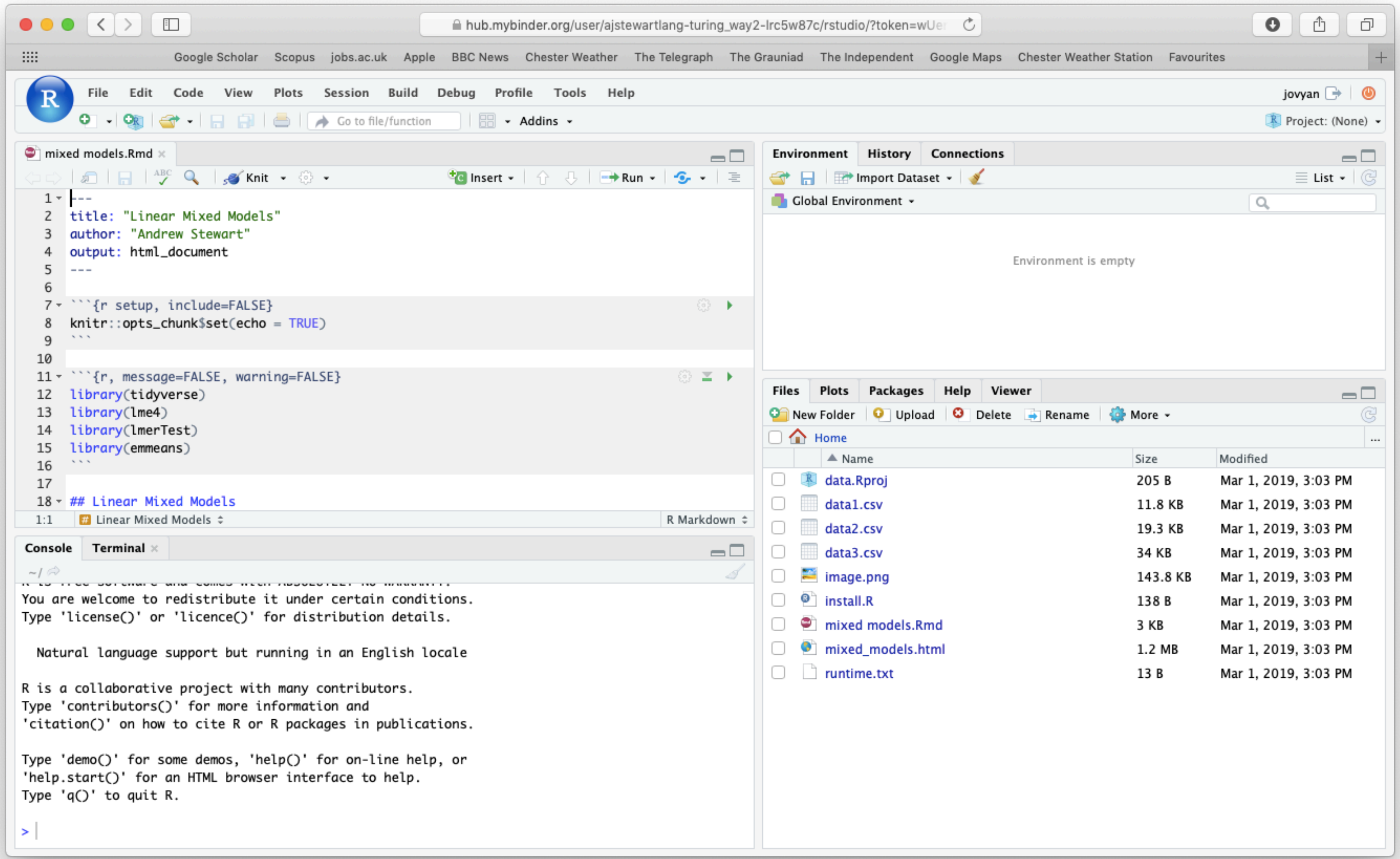
ajstewartlang Create install.R Latest commit 36d3181 2 hours ago

.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

GitHub Desktop

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)
```

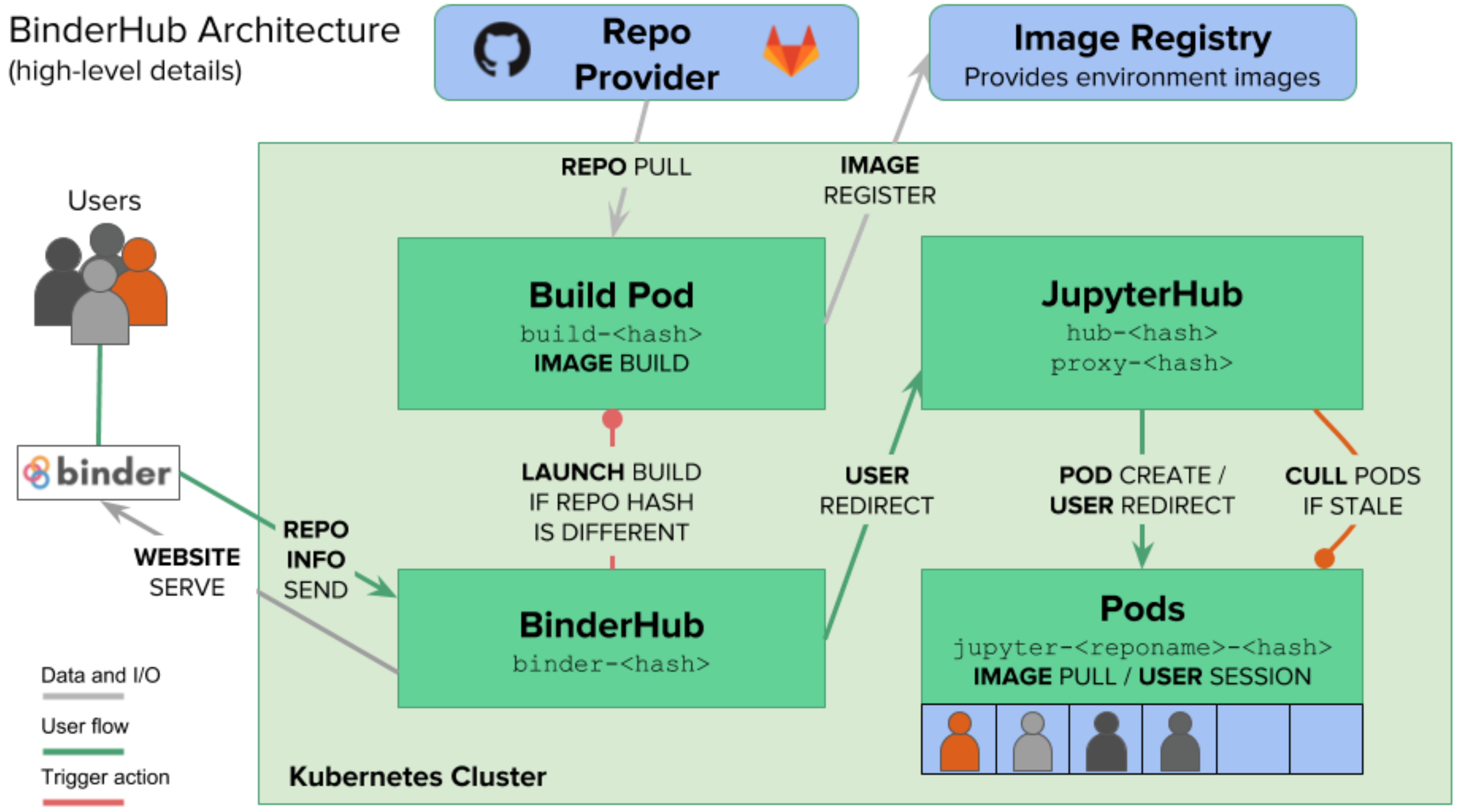
Linear Mixed Models
```

```
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.

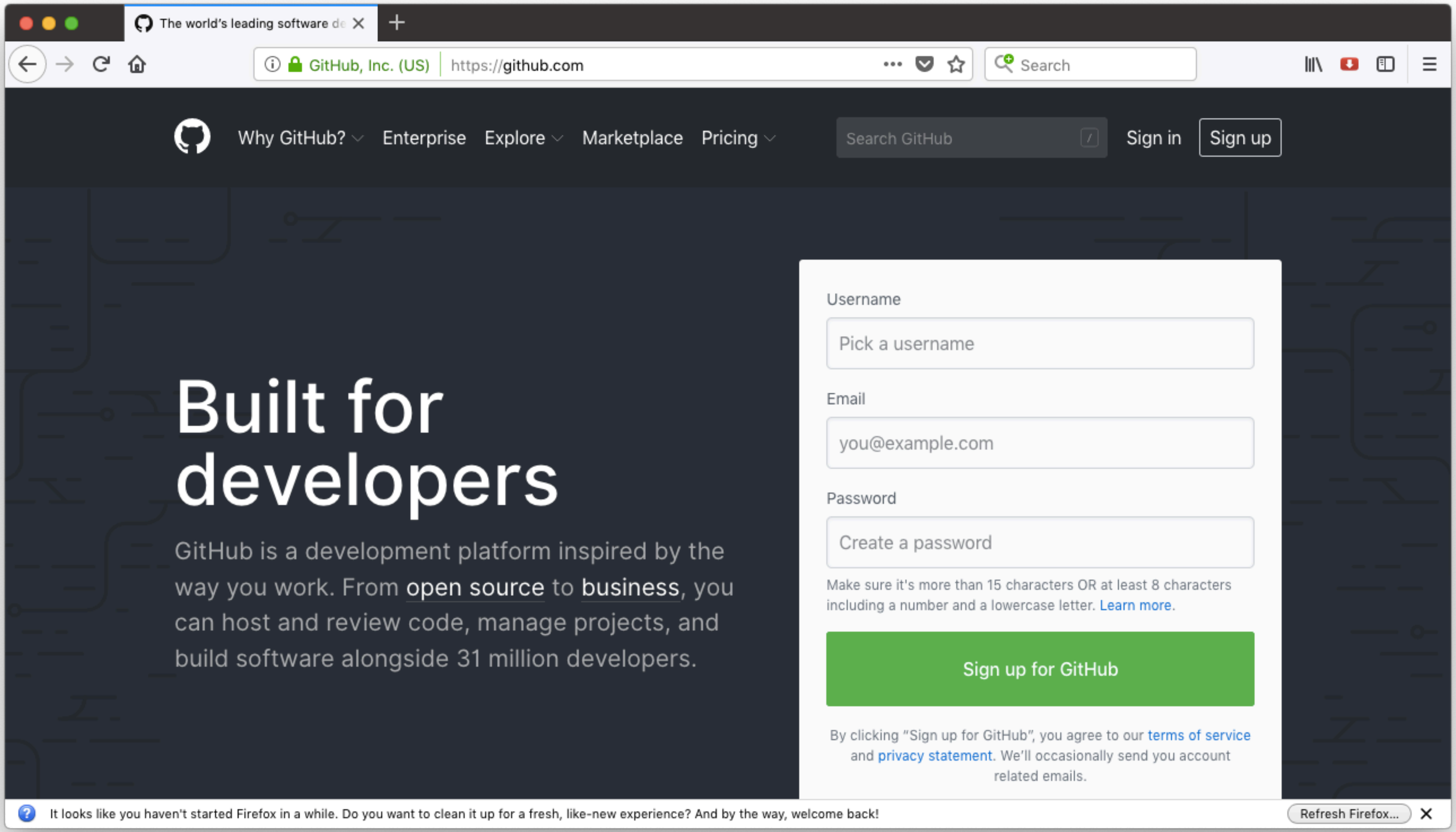
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

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



The world's leading software de

GitHub, Inc. (US) https://github.com

Why GitHub? Enterprise Explore Marketplace Pricing Search GitHub Sign in Sign up

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.

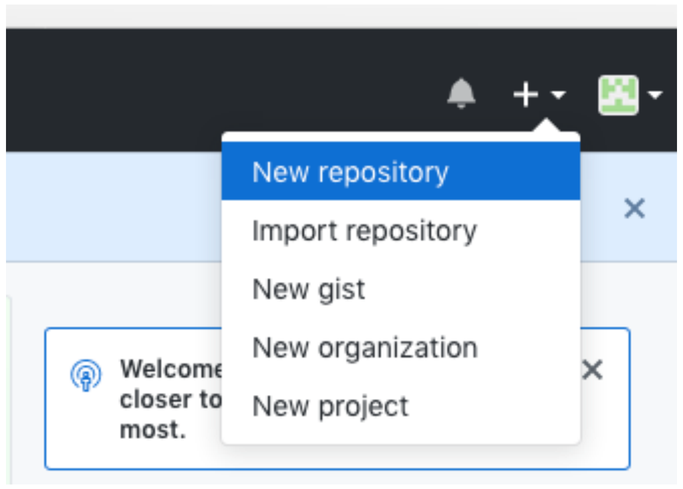
Sign up for GitHub

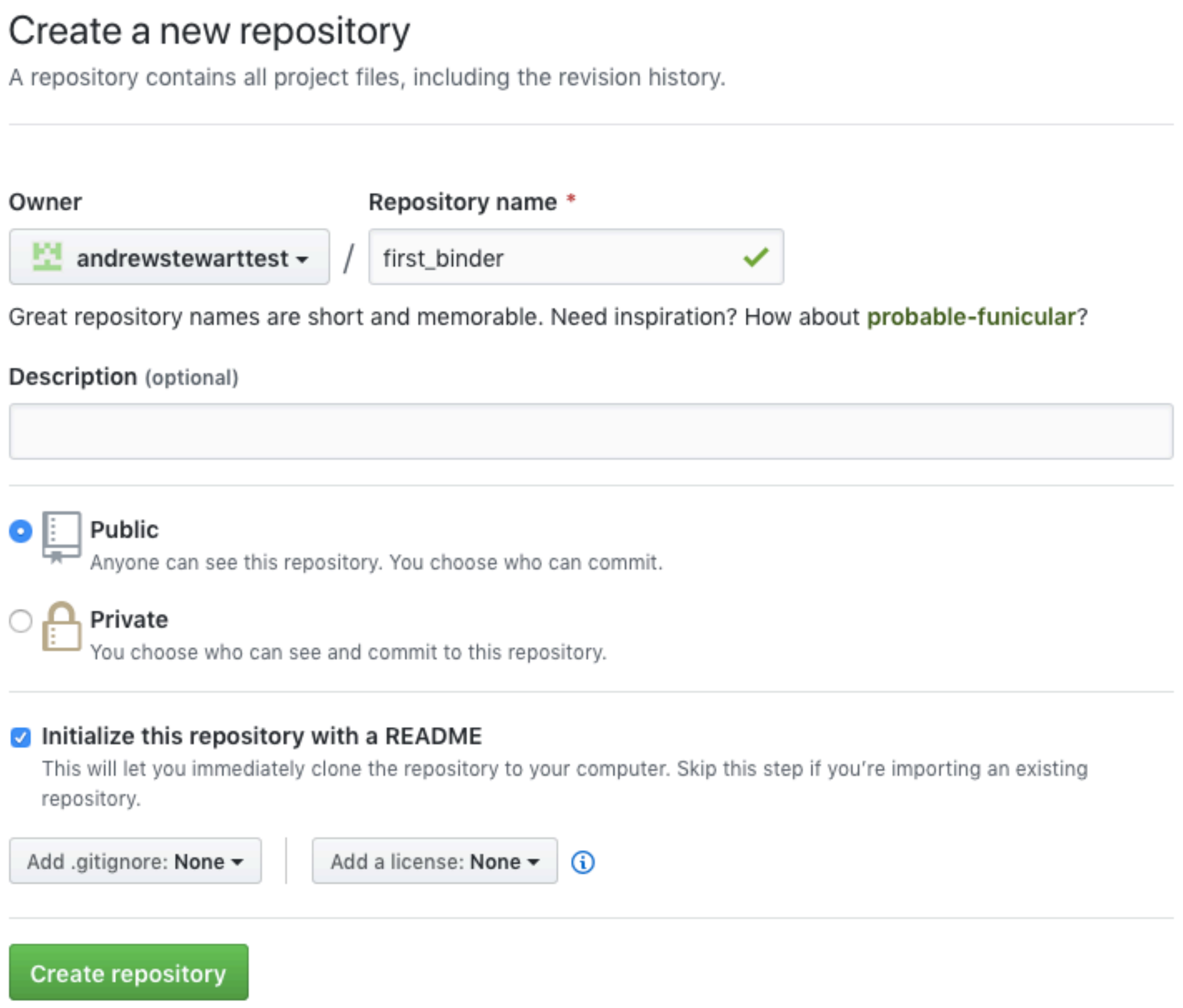
By clicking "Sign up for GitHub", you agree to our terms of service and privacy statement. We'll occasionally send you account related emails.

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!

Refresh Firefox...

# 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.

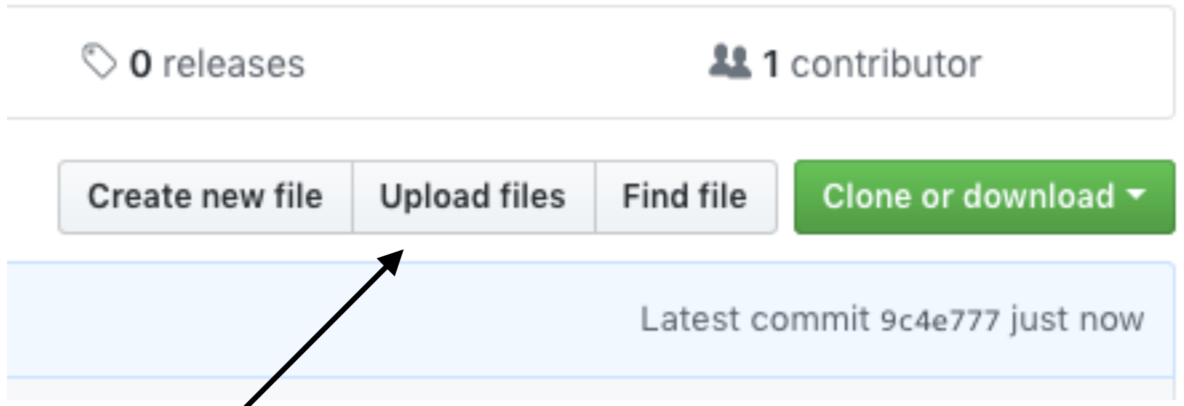
[✓] Initialize this repository with a README

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

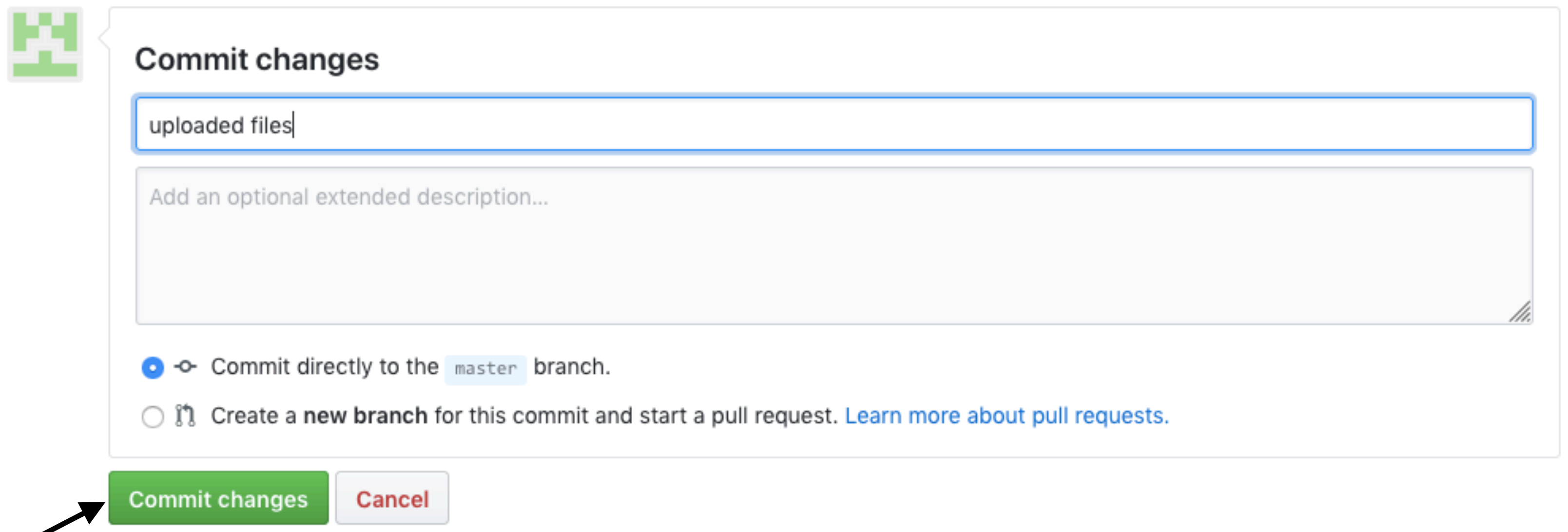
Add .gitignore: None

Add a license: None

# 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...

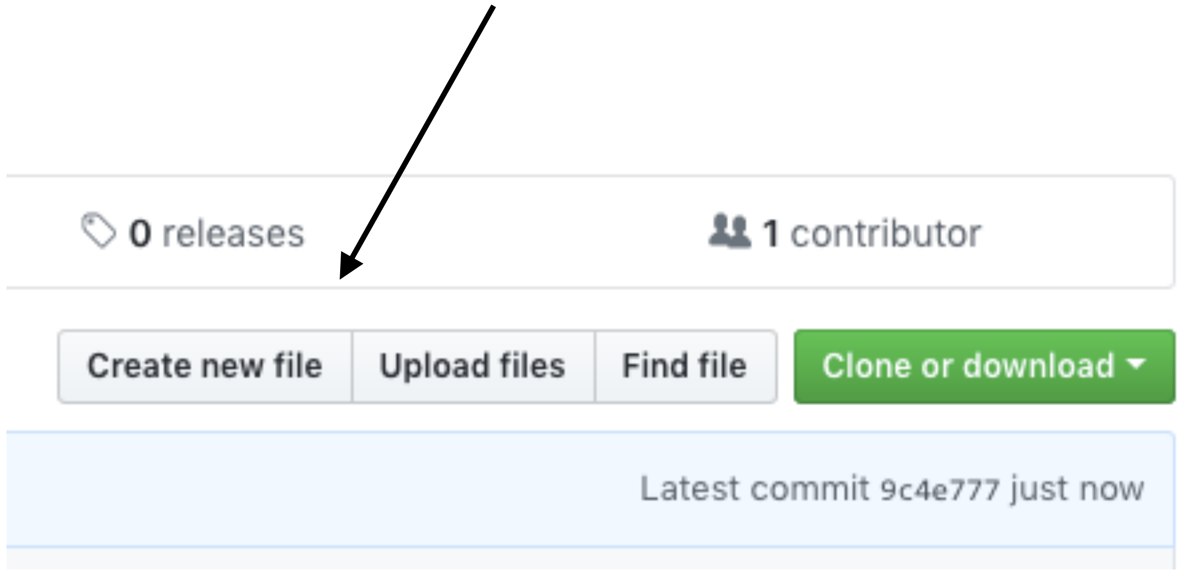
[✓ Commit directly to the `master` branch.]

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

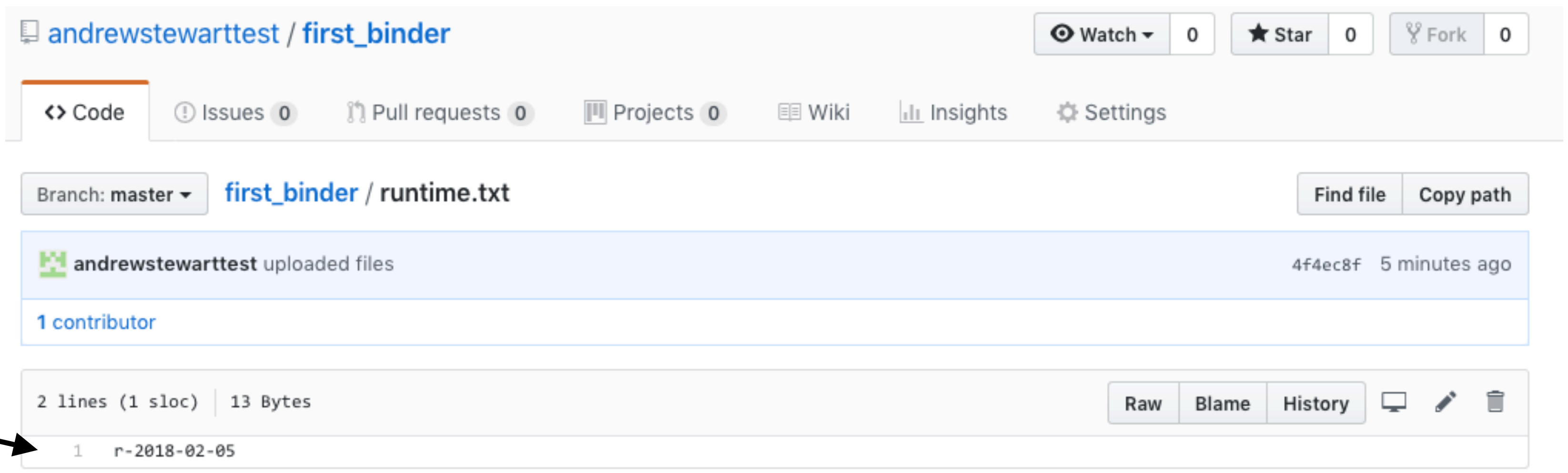
Click here to Commit

## 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”



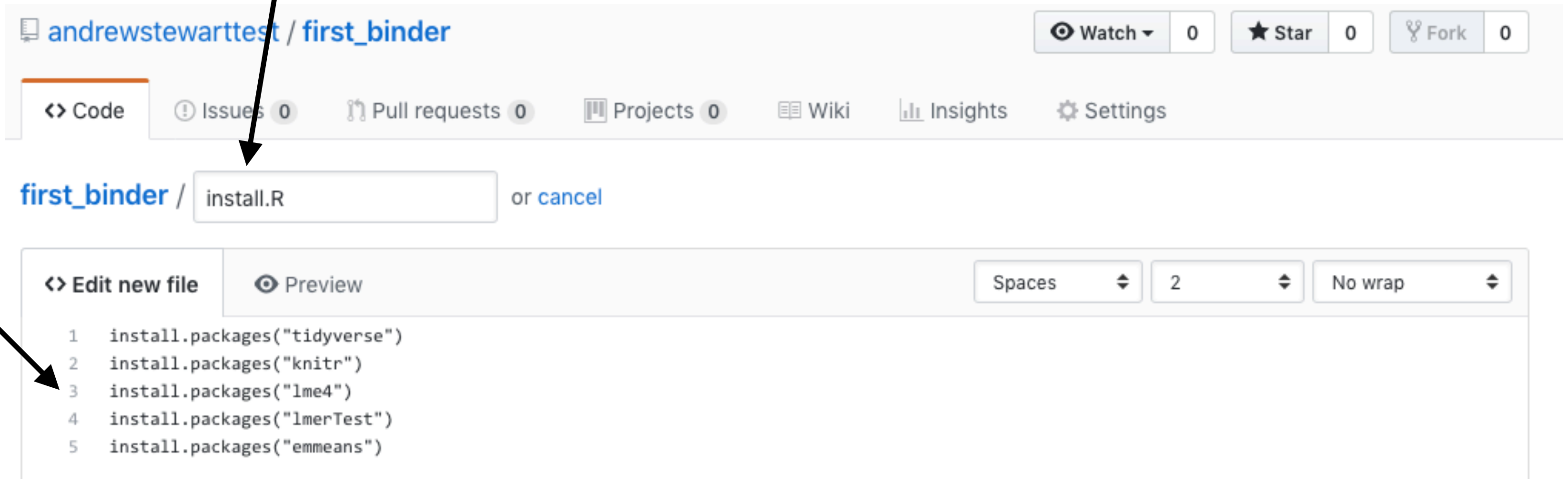
In the runtime.txt file type the date you want in the format r-YYYY-MM-DD



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

Name your file

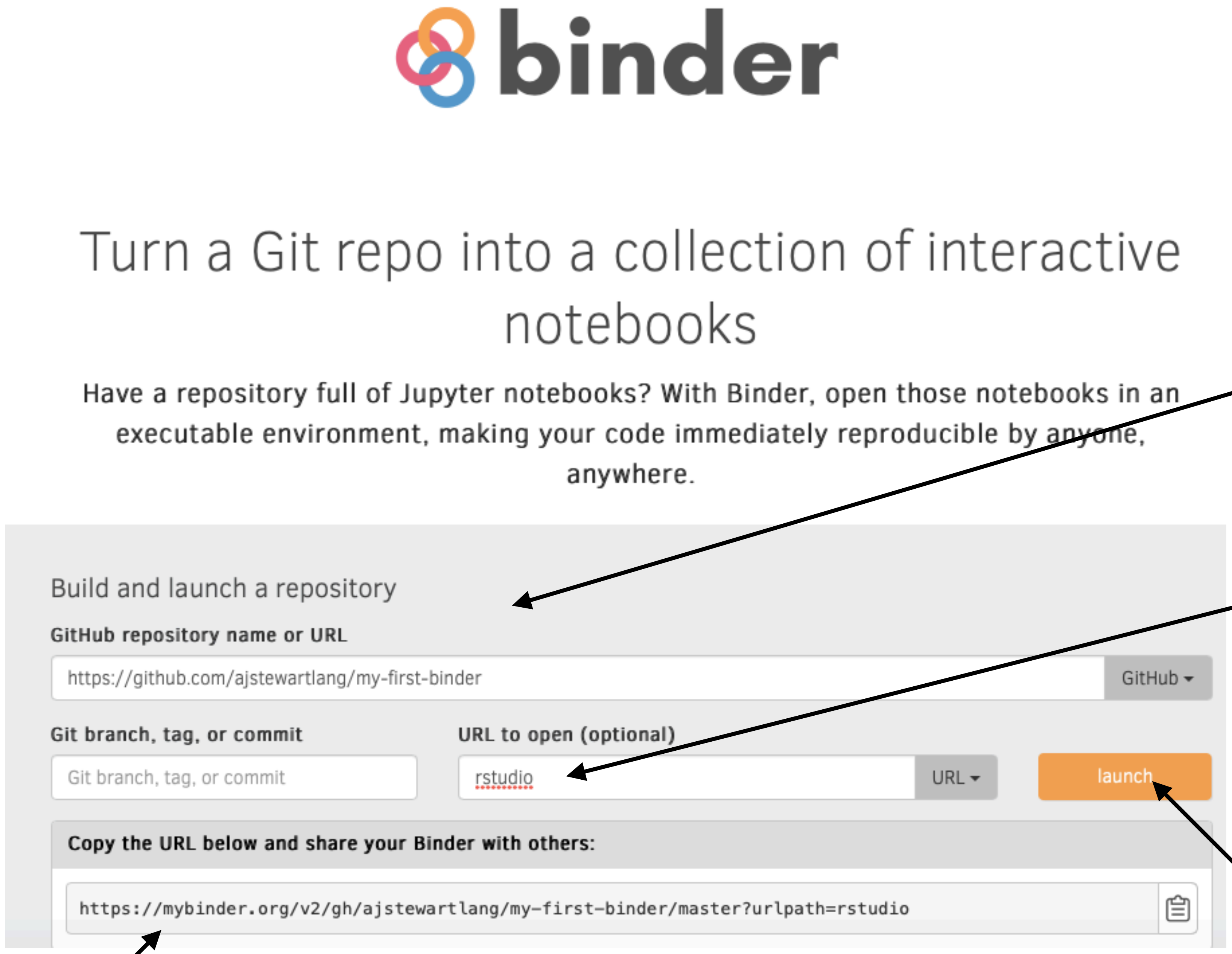
List your packages like this in the install.R file



```
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)



binder

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: Git branch, tag, or commit

URL to open (optional): rstudio

Copy the URL below and share your Binder with others:

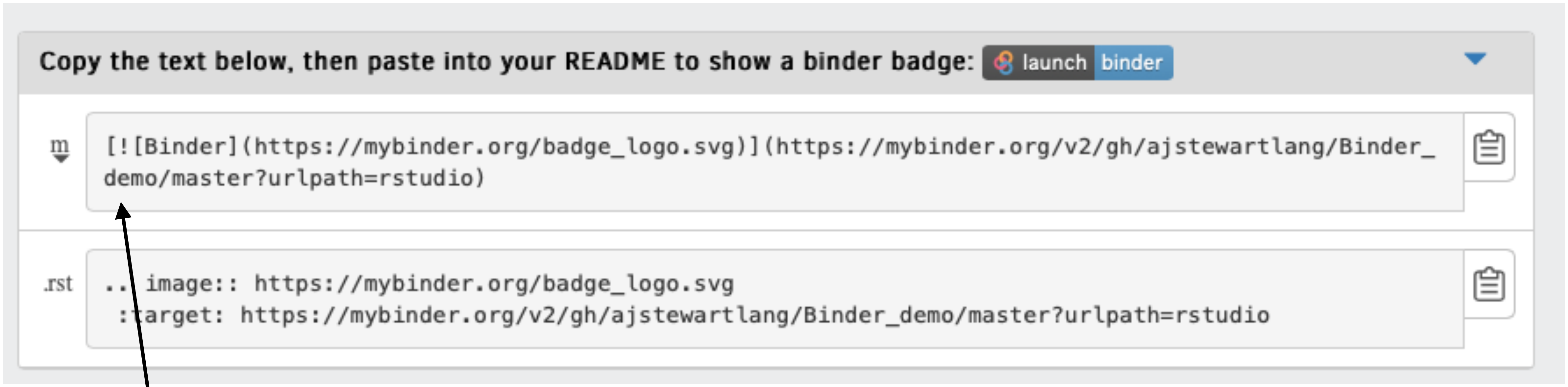
```
https://mybinder.org/v2/gh/ajstewartlang/my-first-binder/master?urlpath=rstudio
```

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.



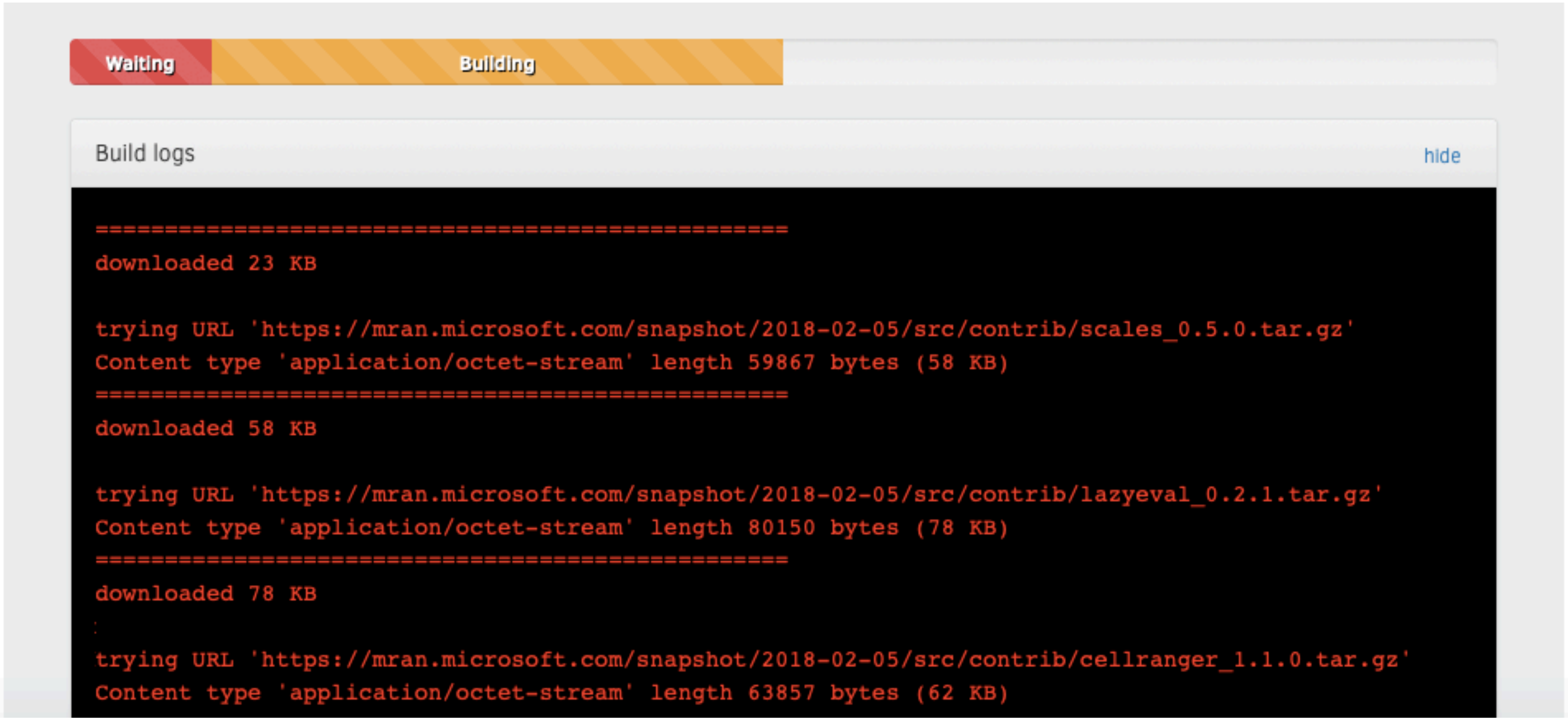
Copy the text below, then paste into your README to show a binder badge:

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

```
.. 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’…

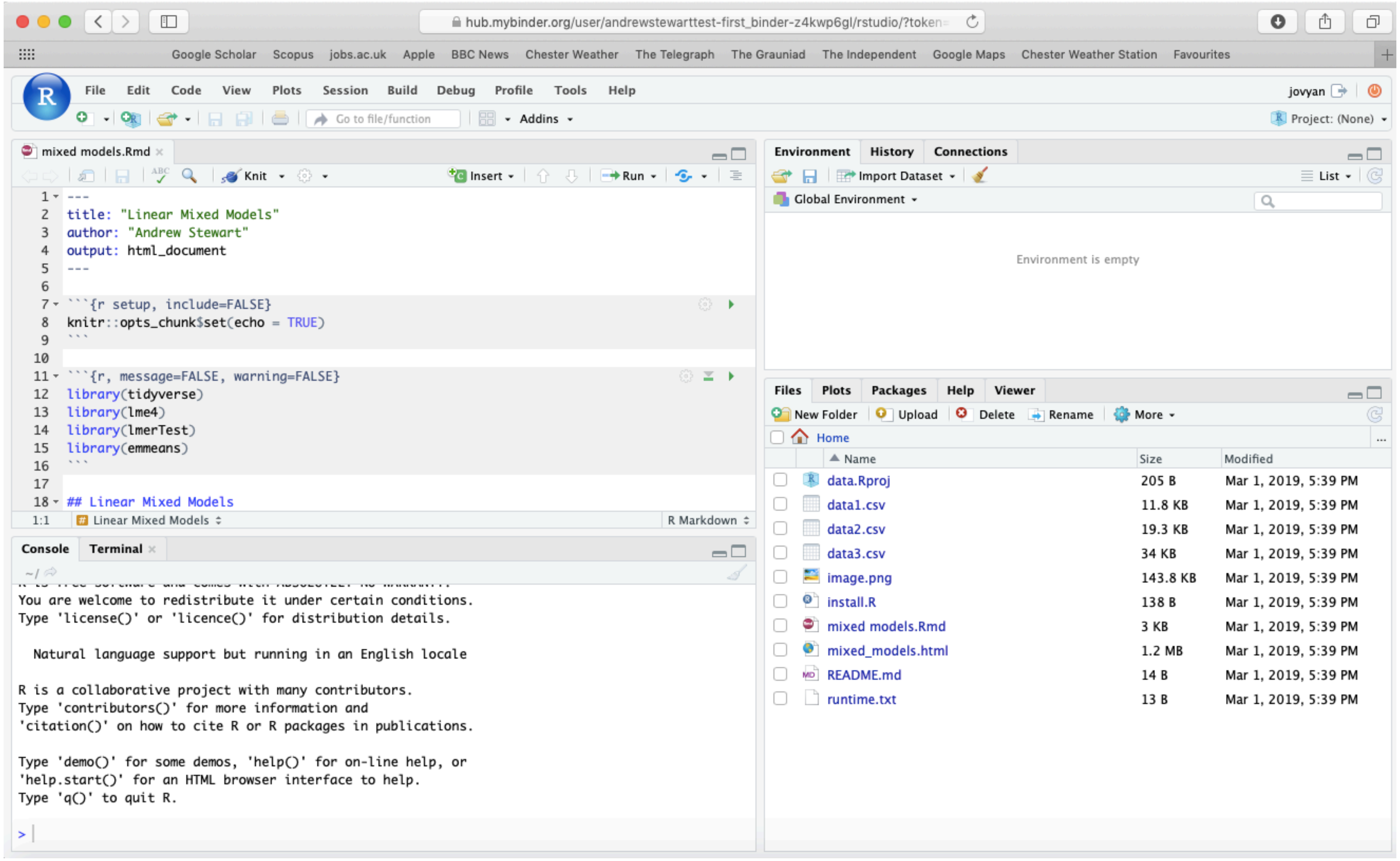


```
==================================================
downloaded 23 KB

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
```

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…

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.

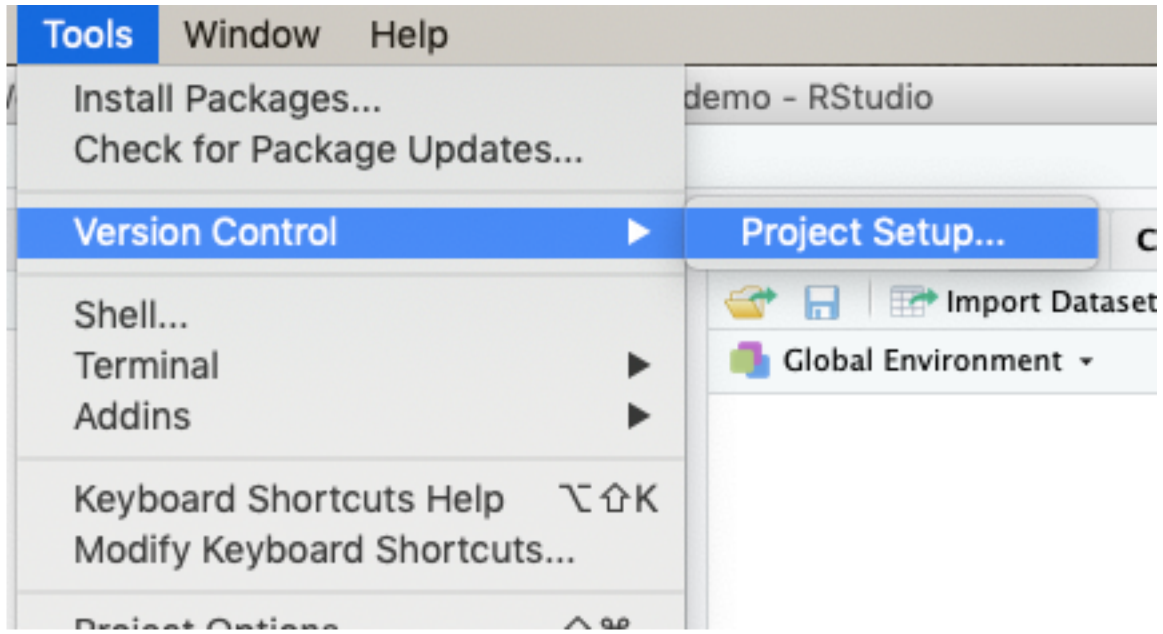
- You need to initialise the local R project folder with Git version control (or clone a repo from GitHub).

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

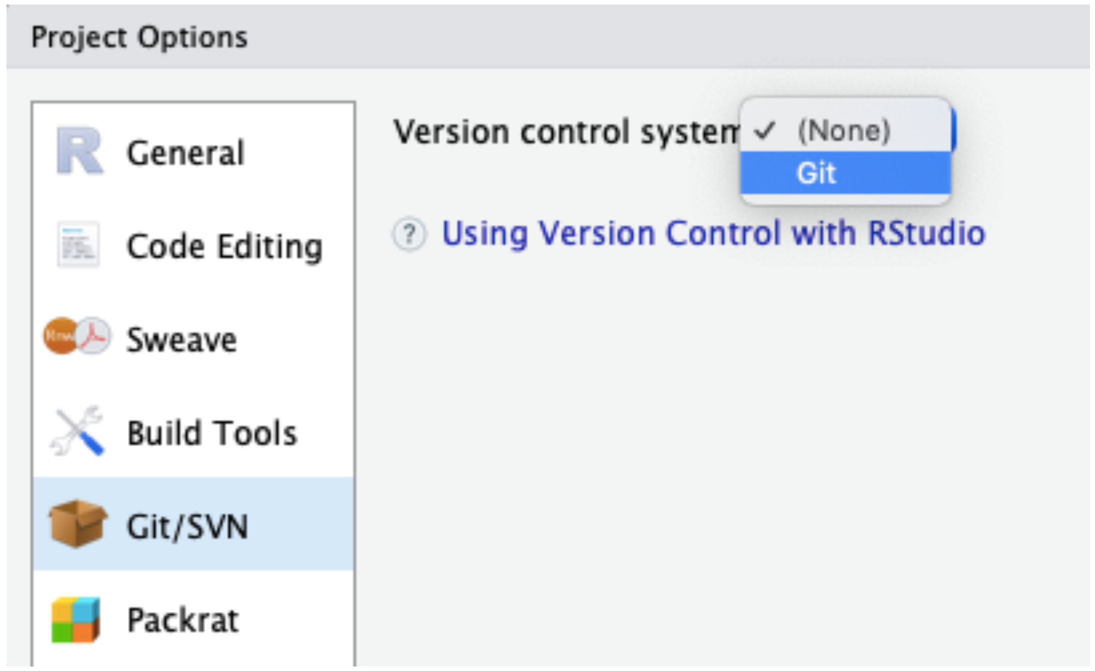
```
> remotes::install_github("karthik/holepunch")

> 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...

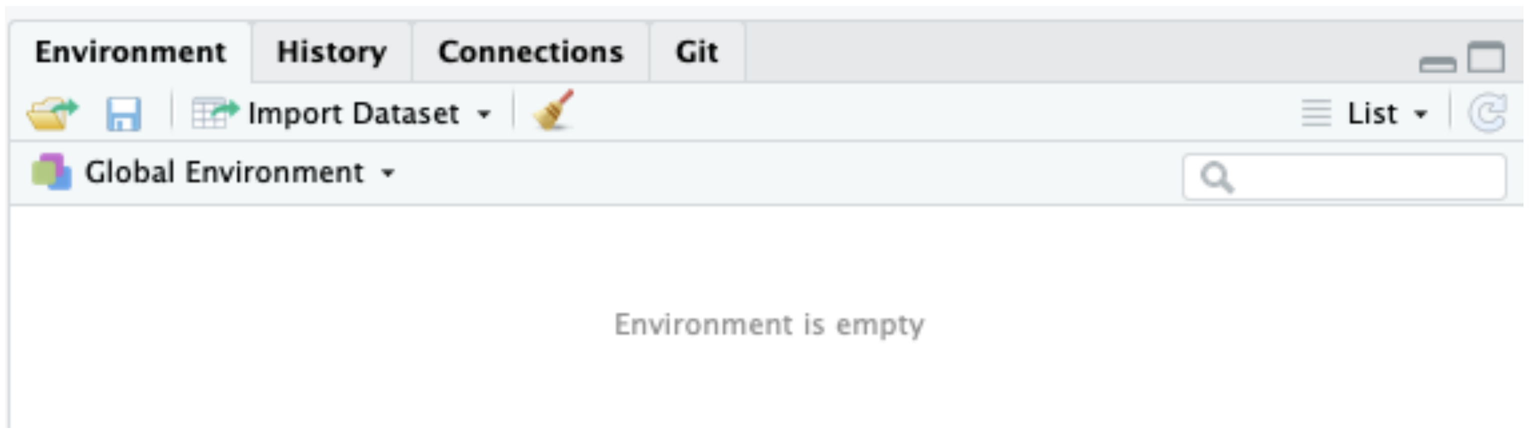
First we need to ensure our folder associated with a project is a repository with git version control.



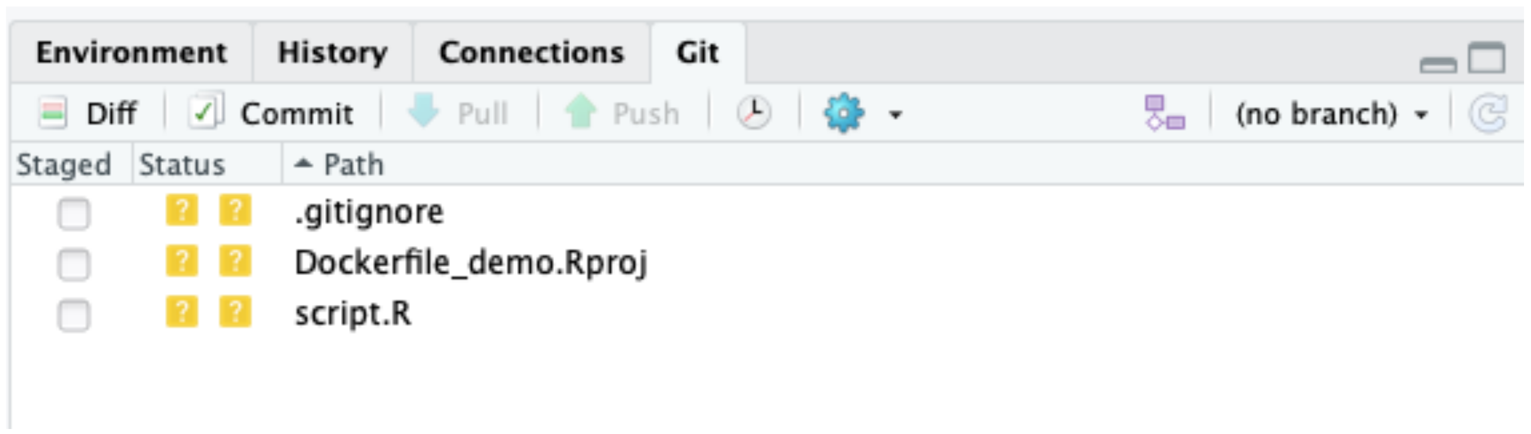
Select Git - you will need to restart your R session at this point.



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

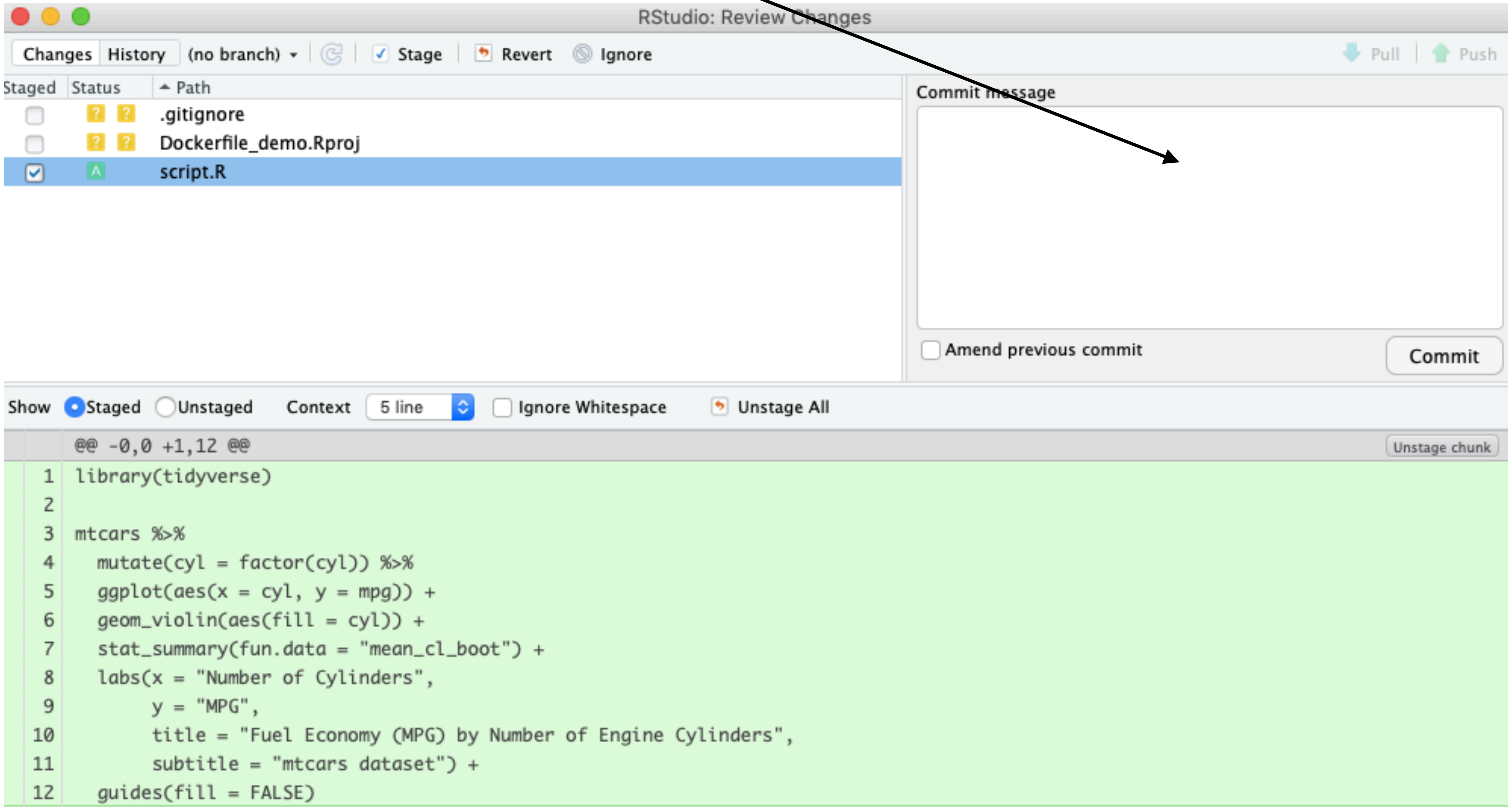


If you click on the Tab you'll see the contents of your folder.



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.



```
@@ -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