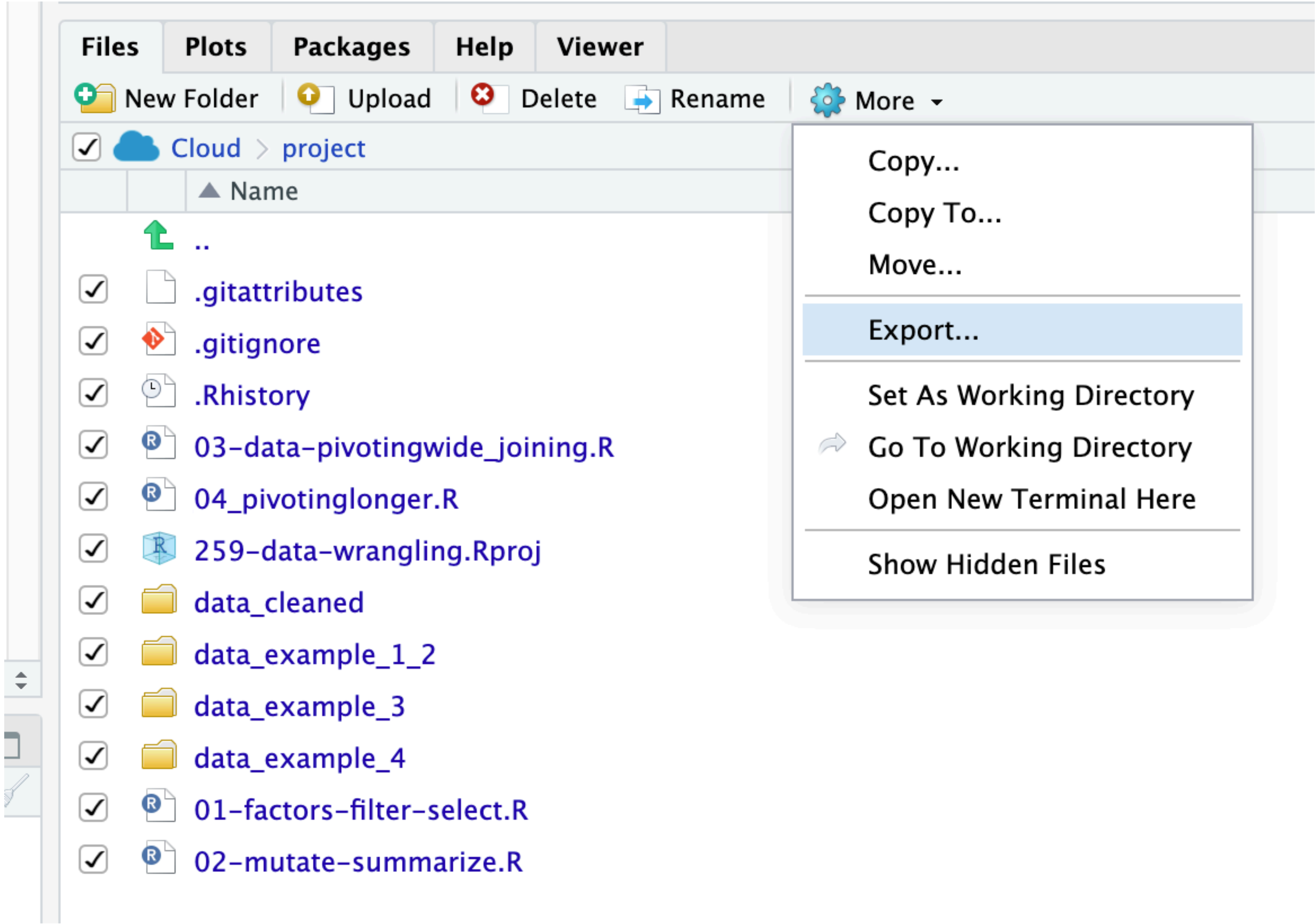
PSYC 259: Principles of Data Science

Week 3: Part 1

Announcements

- Everyone marked homework as “done” on Canvas?
- Starting next week, homework due at 12:30
- We will move projects out of the class workspace each week after grading
- Export files to your own system to back up

Files -> export to save from RStudio Cloud



..
☑ .gitattributes
☑ .gitignore
☑ .Rhistory
☑ 03-data-pivotingwide_joining.R
☑ 04_pivotinglonger.R
☑ 259-data-wrangling.Rproj
☑ data_cleaned
☑ data_example_1_2
☑ data_example_3
☑ data_example_4
☑ 01-factors-filter-select.R
☑ 02-mutate-summarize.R

Outline

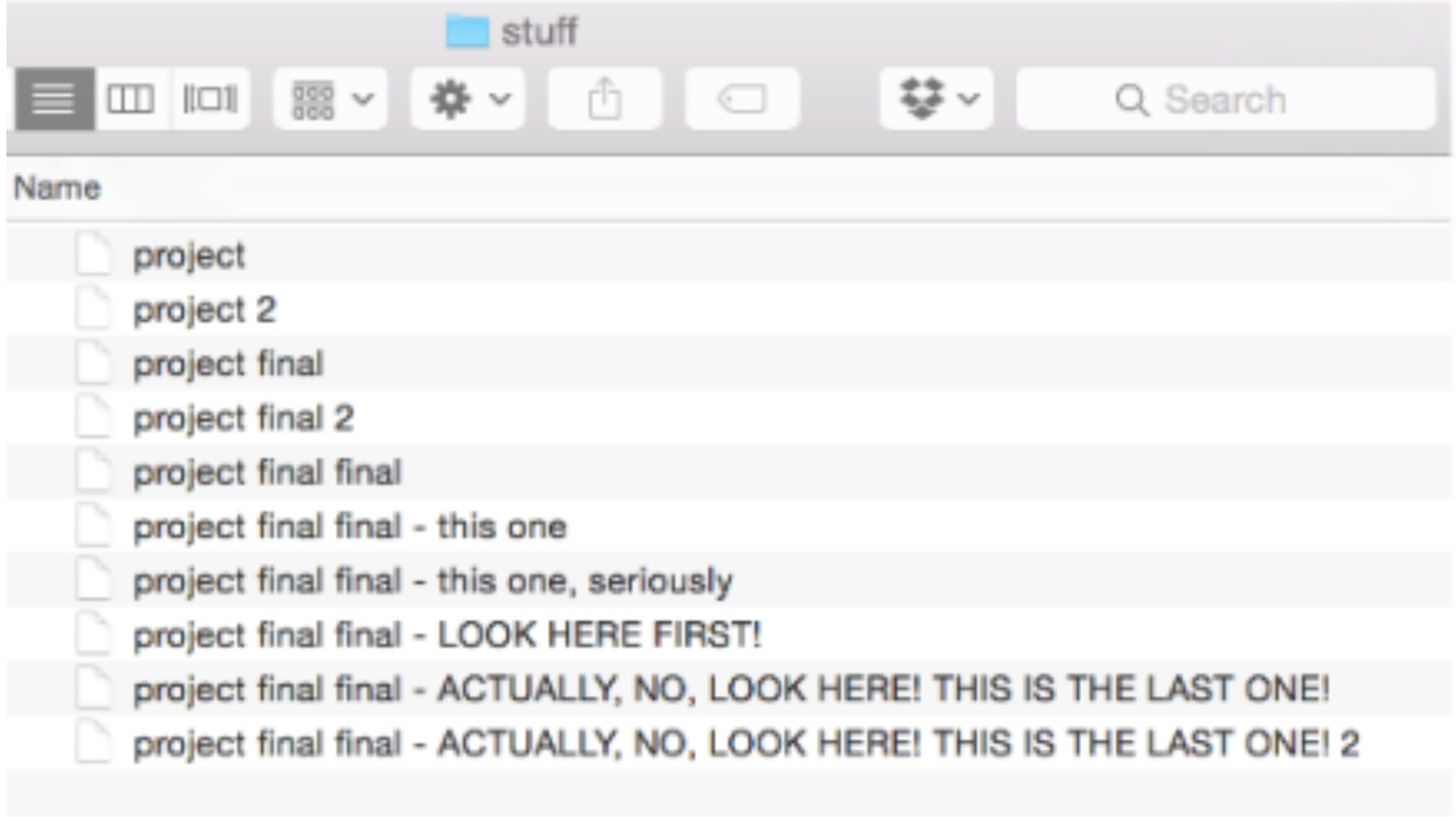
1. Why use version control?
2. git, GitHub tutorial
3. Go over homework 1
4. Go over workflow assignment

What principles should guide project structure?

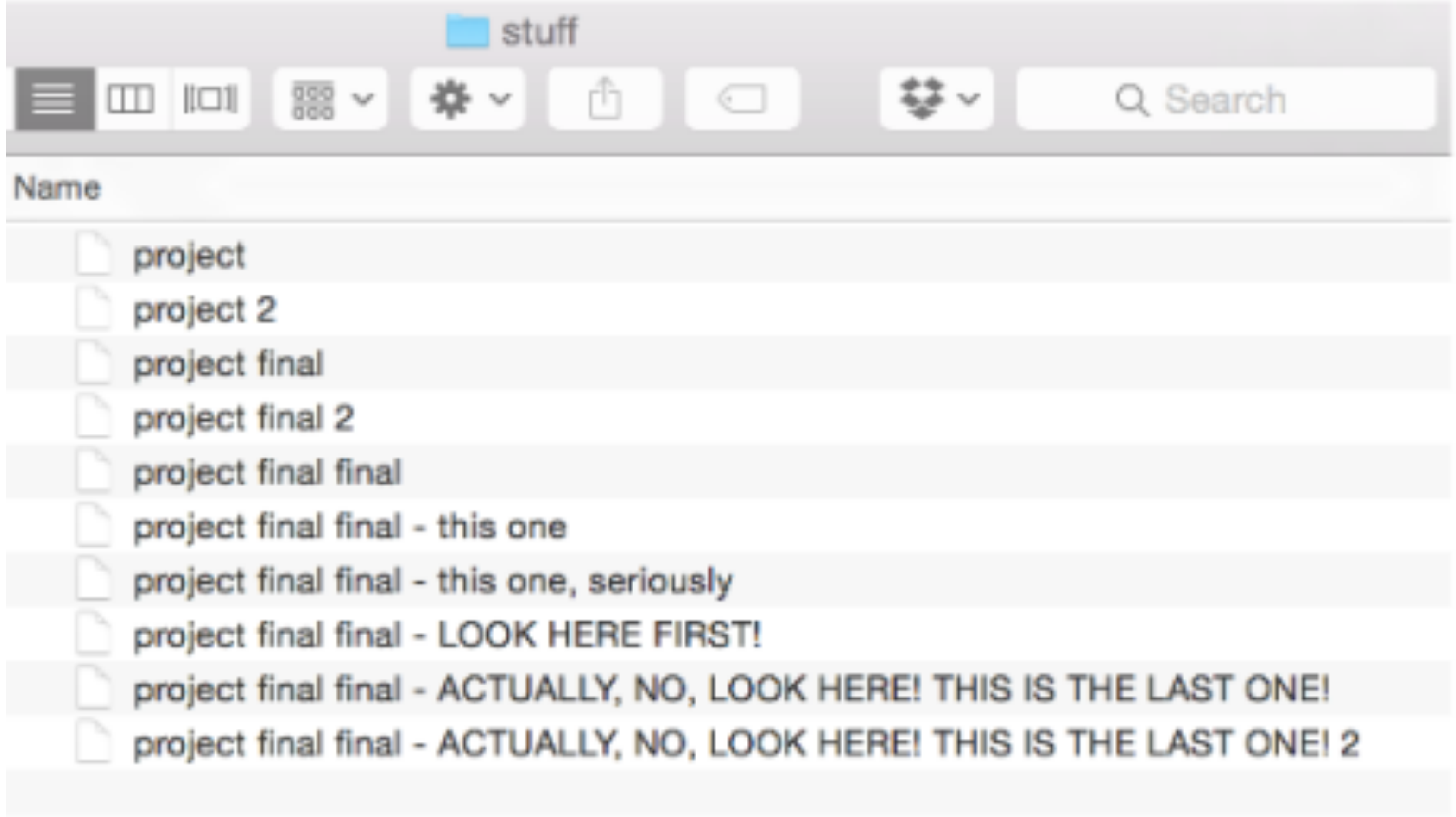
#2 Version Control

Two principles of project workflow

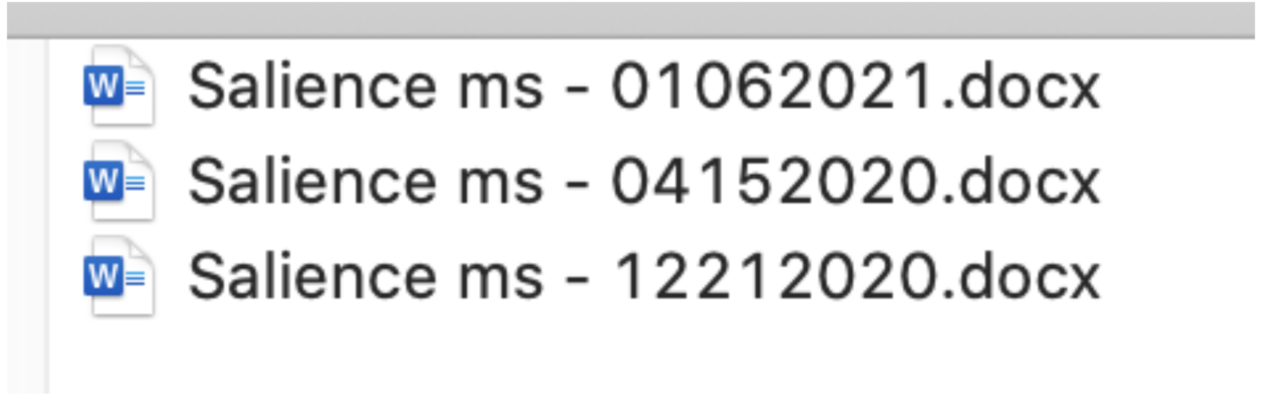
1. Folder organization creates rules and defines a workflow; establishes a location from which to build relative file paths
2. Version control tracks file history without duplication/clutter; allows for collaboration/derivation/experimentation



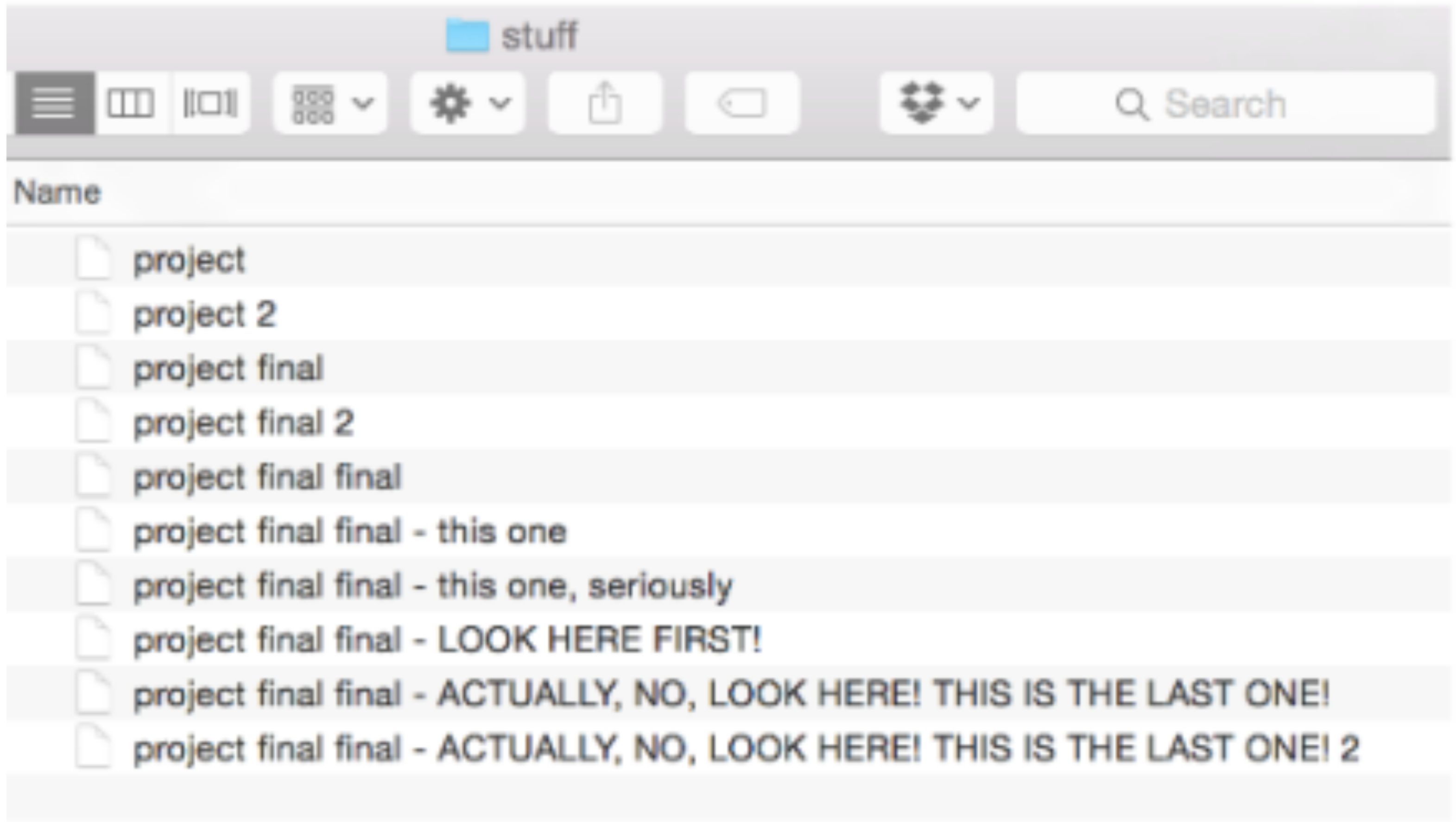
project
project 2
project final
project final 2
project final final
project final final - this one
project final final - this one, seriously
project final final - LOOK HERE FIRST!
project final final - ACTUALLY, NO, LOOK HERE! THIS IS THE LAST ONE!
project final final - ACTUALLY, NO, LOOK HERE! THIS IS THE LAST ONE! 2



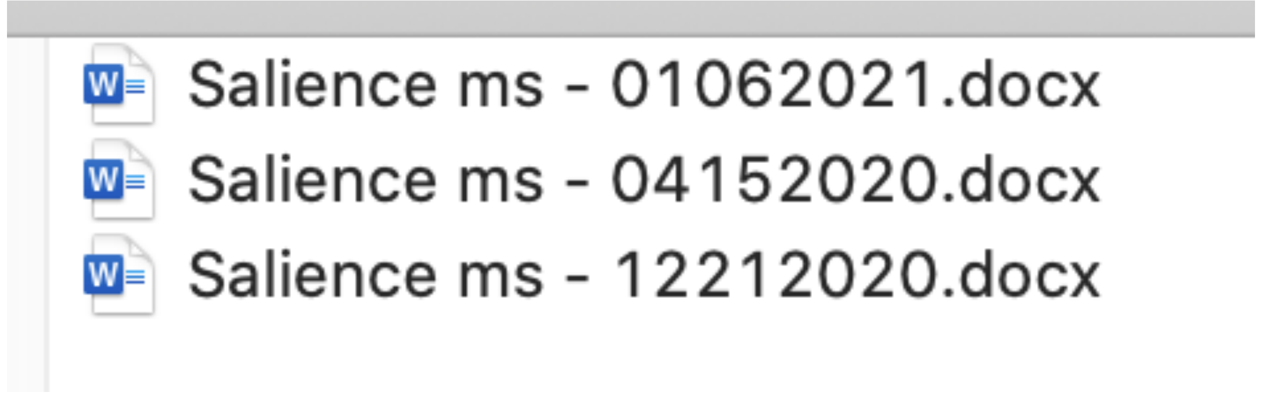
project
project 2
project final
project final 2
project final final
project final final - this one
project final final - this one, seriously
project final final - LOOK HERE FIRST!
project final final - ACTUALLY, NO, LOOK HERE! THIS IS THE LAST ONE!
project final final - ACTUALLY, NO, LOOK HERE! THIS IS THE LAST ONE! 2



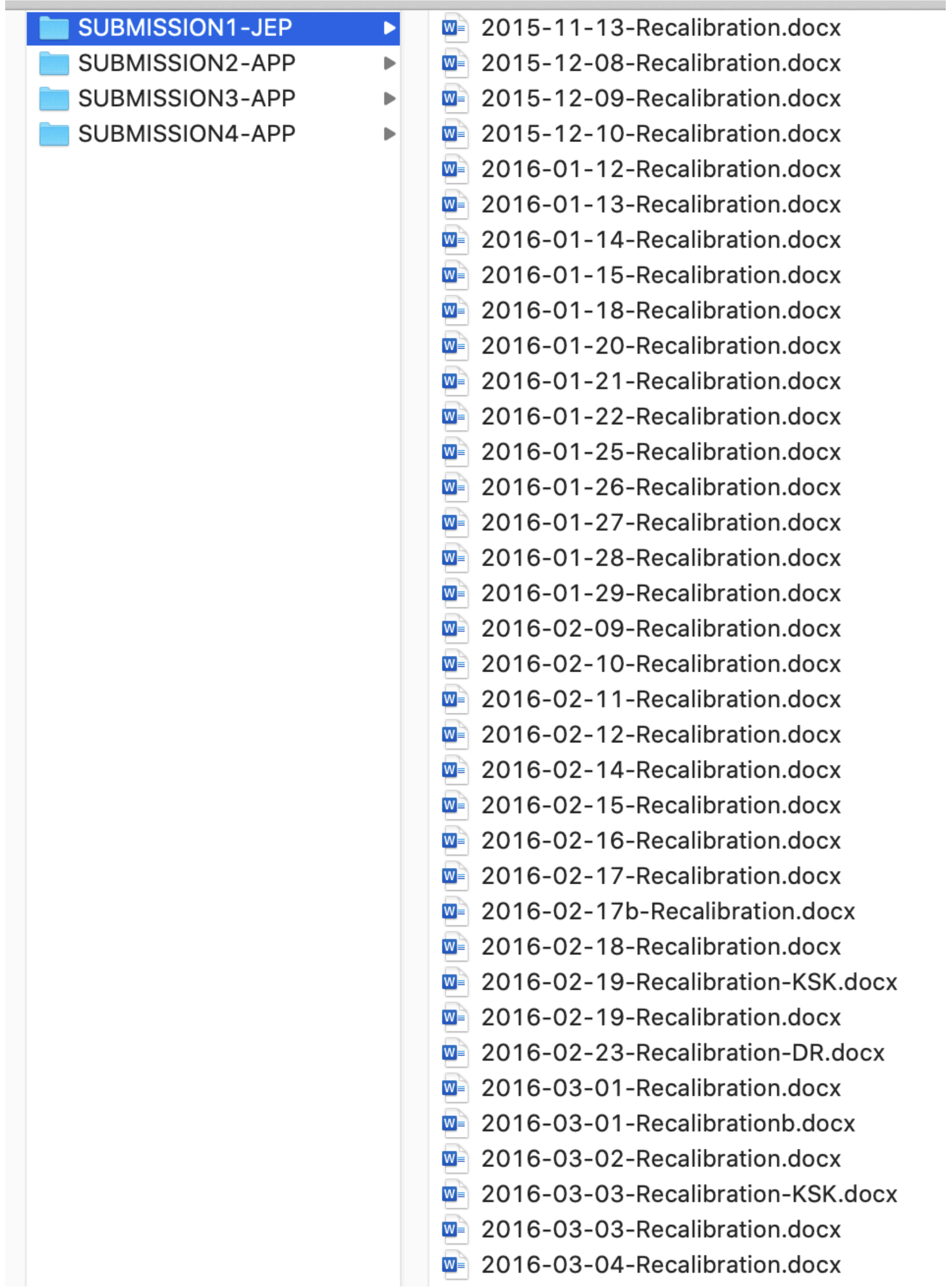
Salience ms - 01062021.docx
Salience ms - 04152020.docx
Salience ms - 12212020.docx



project
project 2
project final
project final 2
project final final
project final final - this one
project final final - this one, seriously
project final final - LOOK HERE FIRST!
project final final - ACTUALLY, NO, LOOK HERE! THIS IS THE LAST ONE!
project final final - ACTUALLY, NO, LOOK HERE! THIS IS THE LAST ONE! 2



Salience ms - 01062021.docx
Salience ms - 04152020.docx
Salience ms - 12212020.docx



SUBMISSION1-JEP
SUBMISSION2-APP
SUBMISSION3-APP
SUBMISSION4-APP

2015-11-13-Recalibration.docx
2015-12-08-Recalibration.docx
2015-12-09-Recalibration.docx
2015-12-10-Recalibration.docx
2016-01-12-Recalibration.docx
2016-01-13-Recalibration.docx
2016-01-14-Recalibration.docx
2016-01-15-Recalibration.docx
2016-01-18-Recalibration.docx
2016-01-20-Recalibration.docx
2016-01-21-Recalibration.docx
2016-01-22-Recalibration.docx
2016-01-25-Recalibration.docx
2016-01-26-Recalibration.docx
2016-01-27-Recalibration.docx
2016-01-28-Recalibration.docx
2016-01-29-Recalibration.docx
2016-02-09-Recalibration.docx
2016-02-10-Recalibration.docx
2016-02-11-Recalibration.docx
2016-02-12-Recalibration.docx
2016-02-14-Recalibration.docx
2016-02-15-Recalibration.docx
2016-02-16-Recalibration.docx
2016-02-17-Recalibration.docx
2016-02-17b-Recalibration.docx
2016-02-18-Recalibration.docx
2016-02-19-Recalibration-KSK.docx
2016-02-19-Recalibration.docx
2016-02-23-Recalibration-DR.docx
2016-03-01-Recalibration.docx
2016-03-01-Recalibrationb.docx
2016-03-02-Recalibration.docx
2016-03-03-Recalibration-KSK.docx
2016-03-03-Recalibration.docx
2016-03-04-Recalibration.docx

Why do we do this to ourselves?

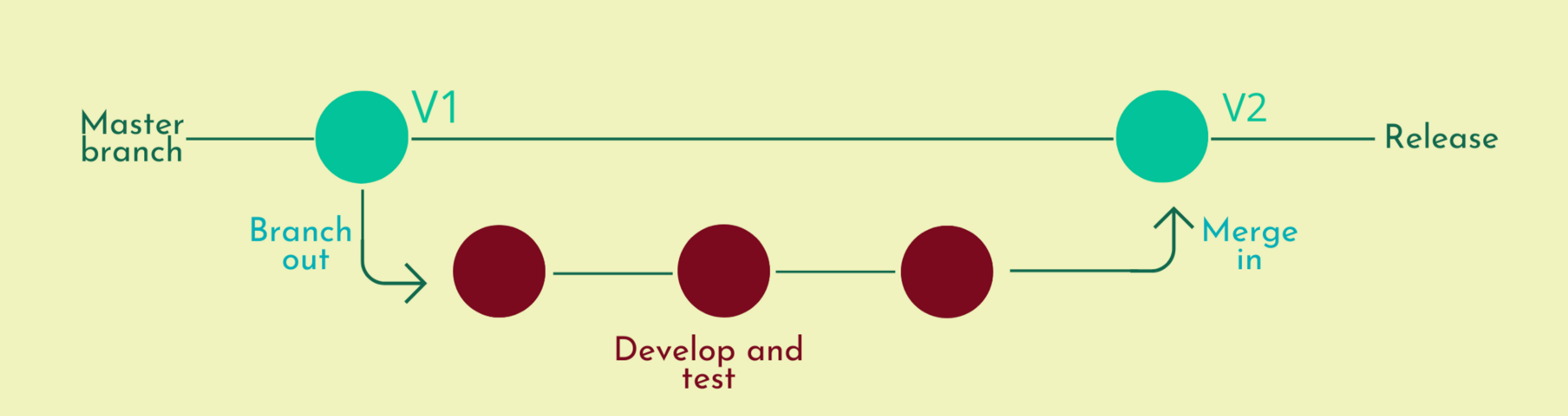
- Want to preserve the document history
  - Easier to take risks, delete things, etc. if you know you can go back (especially for writing code)
  - For collaboration, want to know who made changes
- But it’s messy, inefficient, and easy to break
  - Cluttered file folders (imagine if you wanted to track every change to every type of file)
  - Copy -> paste -> rename is tedious
  - No record of *what* the changes were

The solution: Use version control software (e.g., git, svm)

- Define a **repository** (your project folder) and tell git which files to **track** vs **ignore**
- Place your repository on a central hub (e.g., GitHub)
- Git tracks what/when changes were made, and you tell git whether to **commit** those changes
- You can **push** those changes to the central hub to store them (now they're saved)
- Other users/computers can **pull** those changes from the hub to keep their local copy up-to-date

More advanced git features

- Create branches (alternate timelines) to develop and test new functions, then merge those changes back to the master branch when they're ready



- Fork a project and keep it linked to the original, get updates from the "upstream" project and request to have the original incorporate your changes

Other considerations

- Easiest to get started working on a solo project
- Collaborative version control can get messy if people forget to commit/push their changes or to pull the most recent version from the master repo
- git wants your folders out of dropbox or other cloud synced drives, so you have to commit to using it
- Lots of ways to use git (Github website, Github app, RStudio git window, command line)
- Public vs. private repos
- Flat files vs. binary files

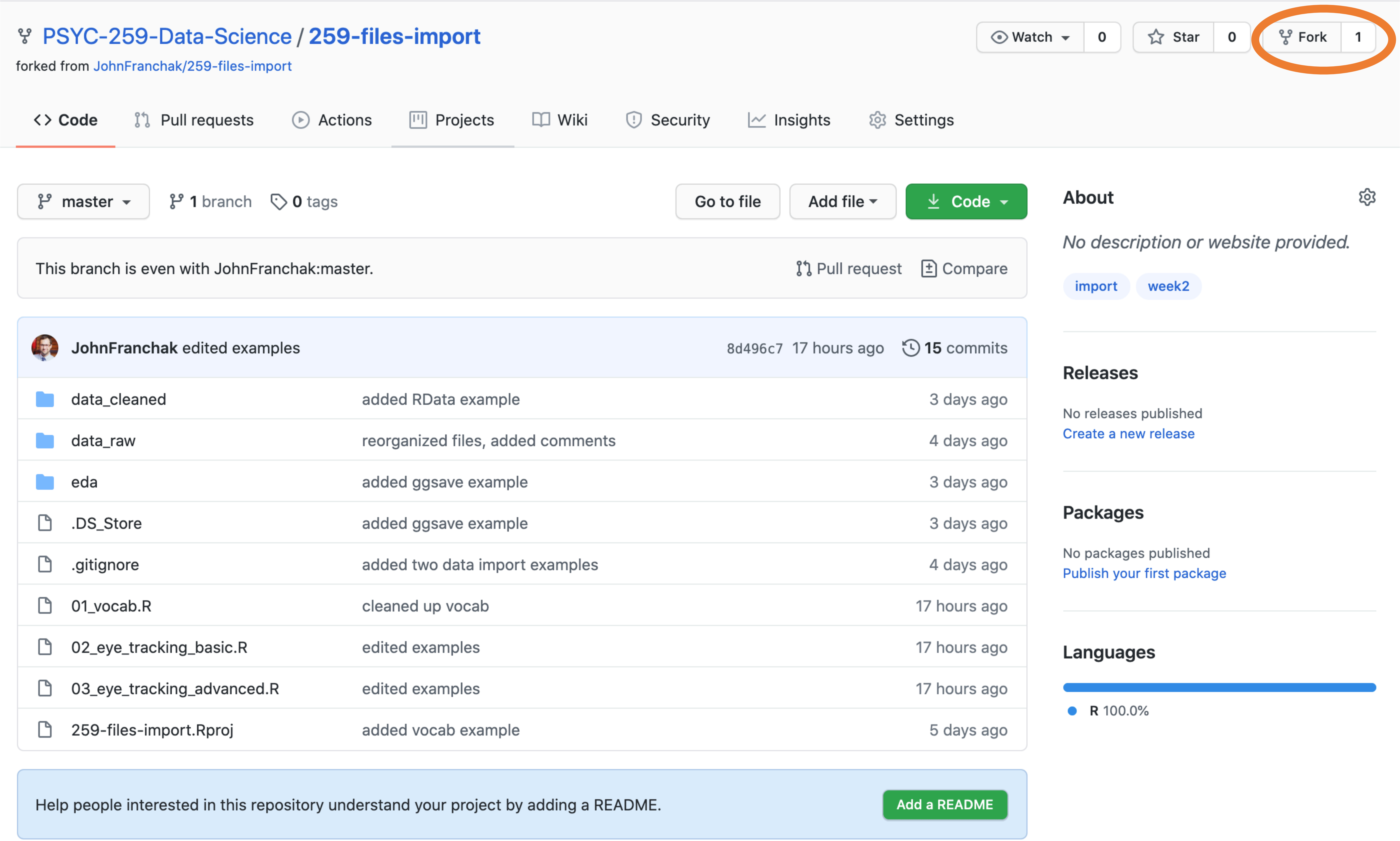
Version control tutorial

Version control tutorial

- Git and GitHub
- Integrating git with R Studio

Fork a repository to make your own copy of it

- Log in to github.com with your username
- Go to https://github.com/PSYC-259-Data-Science/259-files-import



PSYC-259-Data-Science / 259-files-import

forked from JohnFranchak/259-files-import

1 branch 0 tags

This branch is even with JohnFranchak:master.

JohnFranchak edited examples 8d496c7 17 hours ago 15 commits

data_cleaned	added RData example	3 days ago
data_raw	reorganized files, added comments	4 days ago
eda	added ggsave example	3 days ago
.DS_Store	added ggsave example	3 days ago
.gitignore	added two data import examples	4 days ago
01_vocab.R	cleaned up vocab	17 hours ago
02_eye_tracking_basic.R	edited examples	17 hours ago
03_eye_tracking_advanced.R	edited examples	17 hours ago
259-files-import.Rproj	added vocab example	5 days ago

Help people interested in this repository understand your project by adding a README.

About

No description or website provided.

import week2

Releases

No releases published
Create a new release

Packages

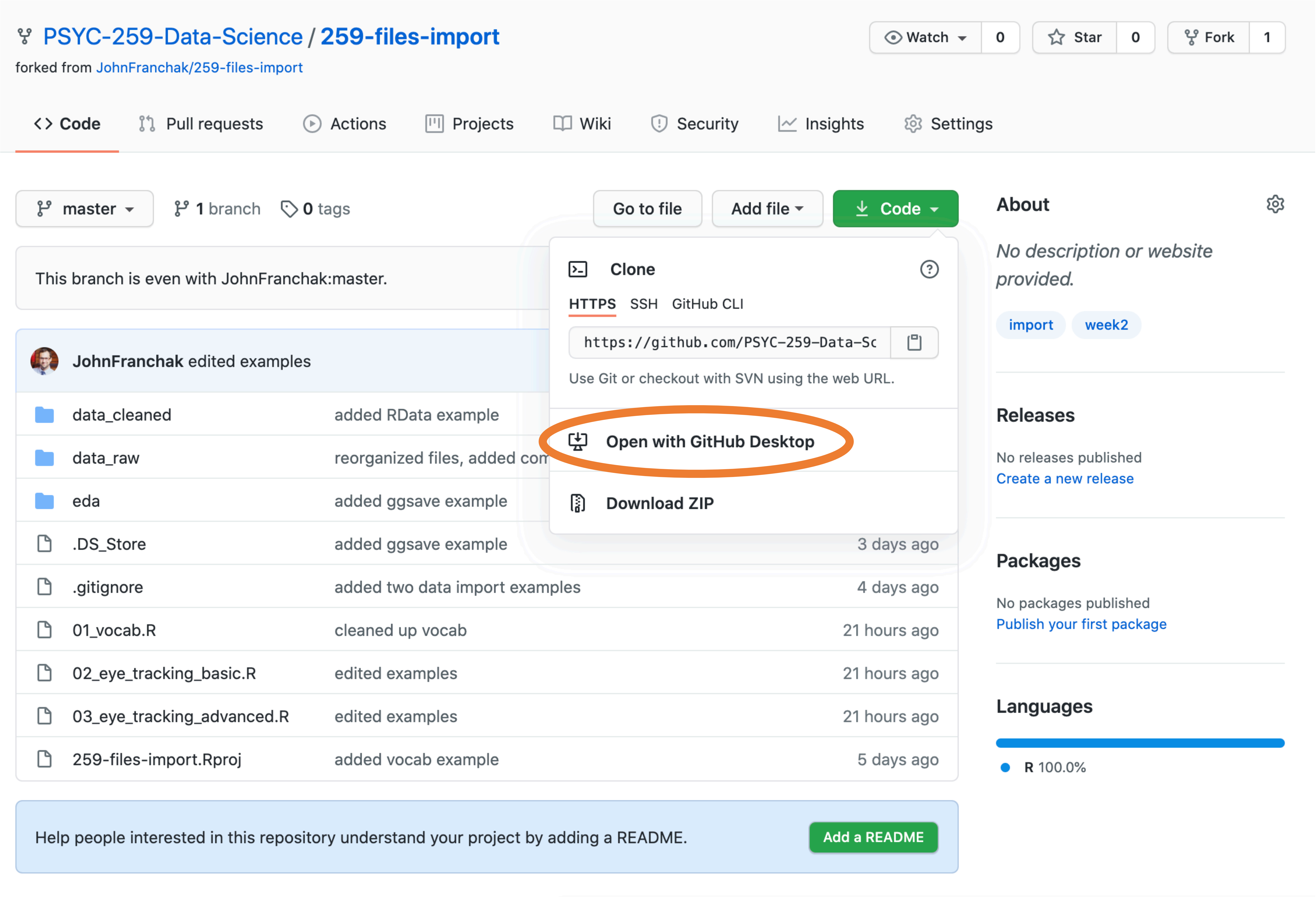
No packages published
Publish your first package

Languages

R 100.0%

Fork a repository to make your own copy of it

- Log in to github.com with your username
- Go to https://github.com/PSYC-259-Data-Science/259-files-import
- Now it won't be PSYC-259-Data-Science's anymore, it will be yours!
- Let's get it off the Github website and onto your local computer by **cloning it**



forked from JohnFranchak/259-files-import

This branch is even with JohnFranchak:master.

JohnFranchak edited examples

data_cleaned	added RData example	
data_raw	reorganized files, added com	
eda	added ggsave example	
.DS_Store	added ggsave example	3 days ago
.gitignore	added two data import examples	4 days ago
01_vocab.R	cleaned up vocab	21 hours ago
02_eye_tracking_basic.R	edited examples	21 hours ago
03_eye_tracking_advanced.R	edited examples	21 hours ago
259-files-import.Rproj	added vocab example	5 days ago

`https://github.com/PSYC-259-Data-Sc`

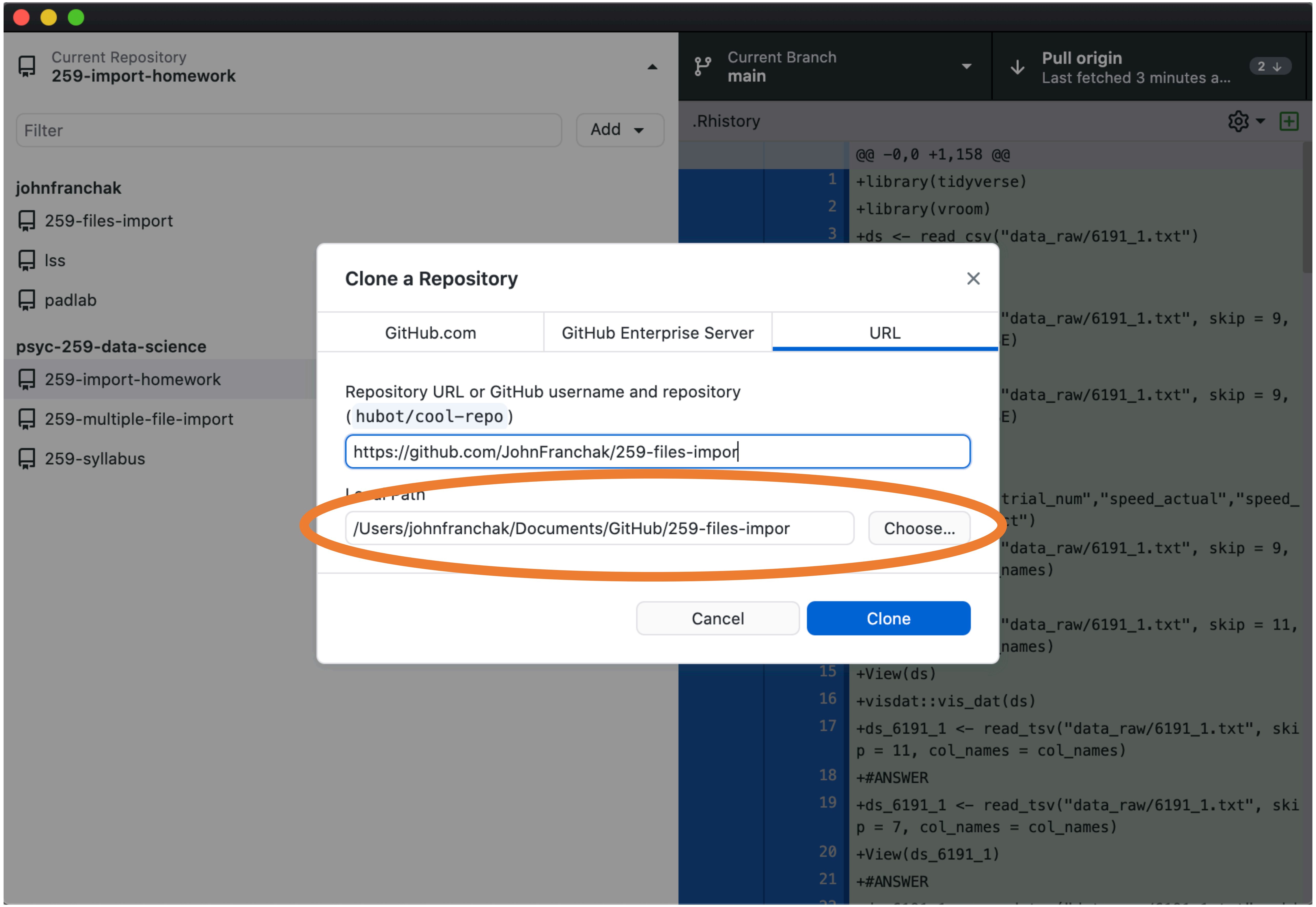
Use Git or checkout with SVN using the web URL.

No description or website provided.

No releases published

No packages published

Help people interested in this repository understand your project by adding a README.



```
@@ -0,0 +1,158 @@
+library(tidyverse)
+library(vroom)
+ds <- read_csv("data_raw/6191_1.txt")
"data_raw/6191_1.txt", skip = 9,
E)
"data_raw/6191_1.txt", skip = 9,
E)
trial_num","speed_actual","speed_
t")
"data_raw/6191_1.txt", skip = 9,
names)
"data_raw/6191_1.txt", skip = 11,
names)
+View(ds)
+visdat::vis_dat(ds)
+ds_6191_1 <- read_tsv("data_raw/6191_1.txt", skip = 11, col_names = col_names)
+#ANSWER
+ds_6191_1 <- read_tsv("data_raw/6191_1.txt", skip = 7, col_names = col_names)
+View(ds_6191_1)
+#ANSWER
```

Clone a Repository

GitHub.com | GitHub Enterprise Server | URL

Repository URL or GitHub username and repository (`hubot/cool-repo`)

https://github.com/JohnFranchak/259-files-impor

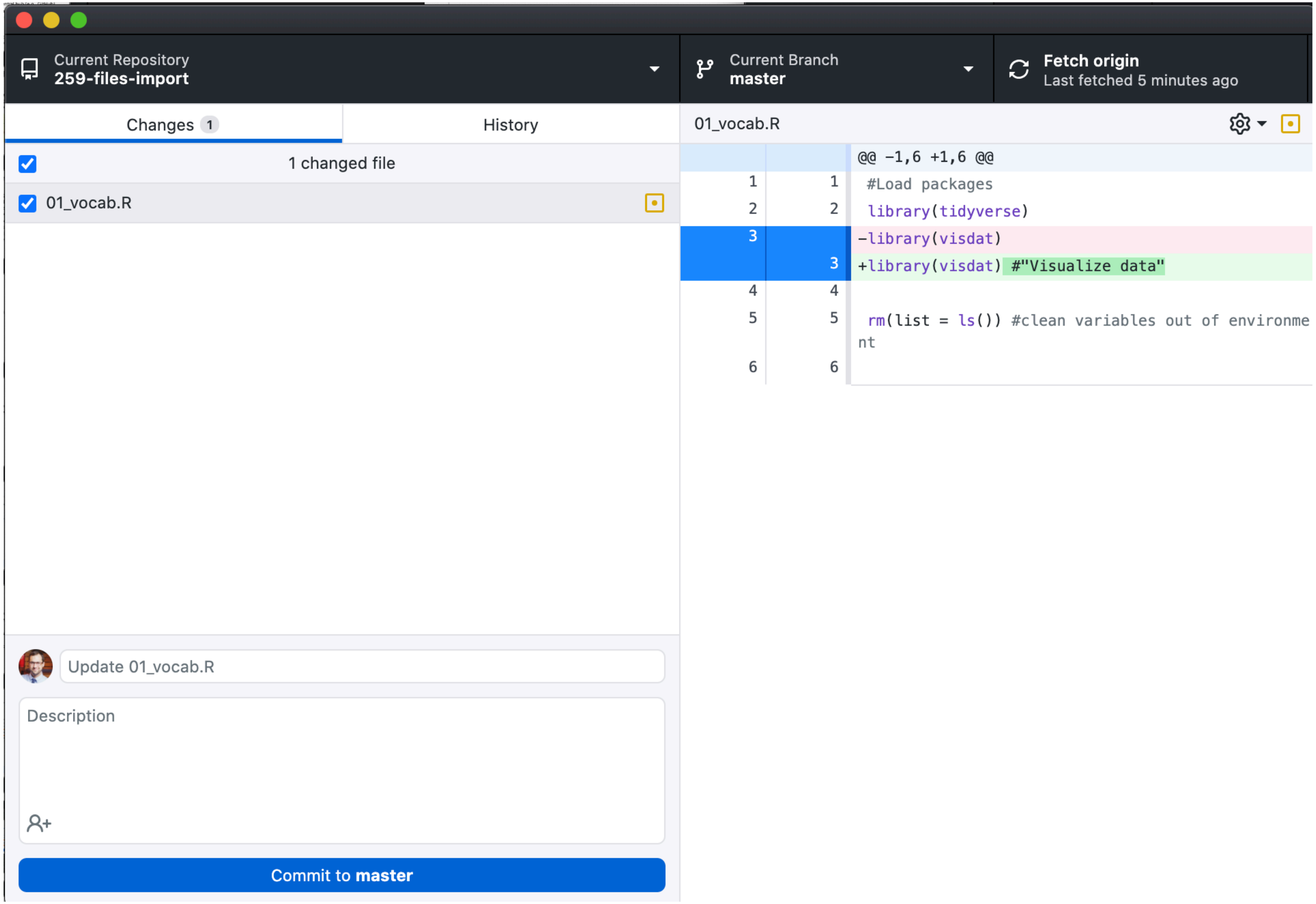
Local Path

/Users/johnfranchak/Documents/GitHub/259-files-impor

Choose a local destination (not on cloud)

Now we’re set up!

- Before starting work, you should always “fetch/pull origin” to get the most recent version from the origin (GitHub)
- Now make a change to a file in the folder (and save the change)
- Go back to the Github app and see what happens

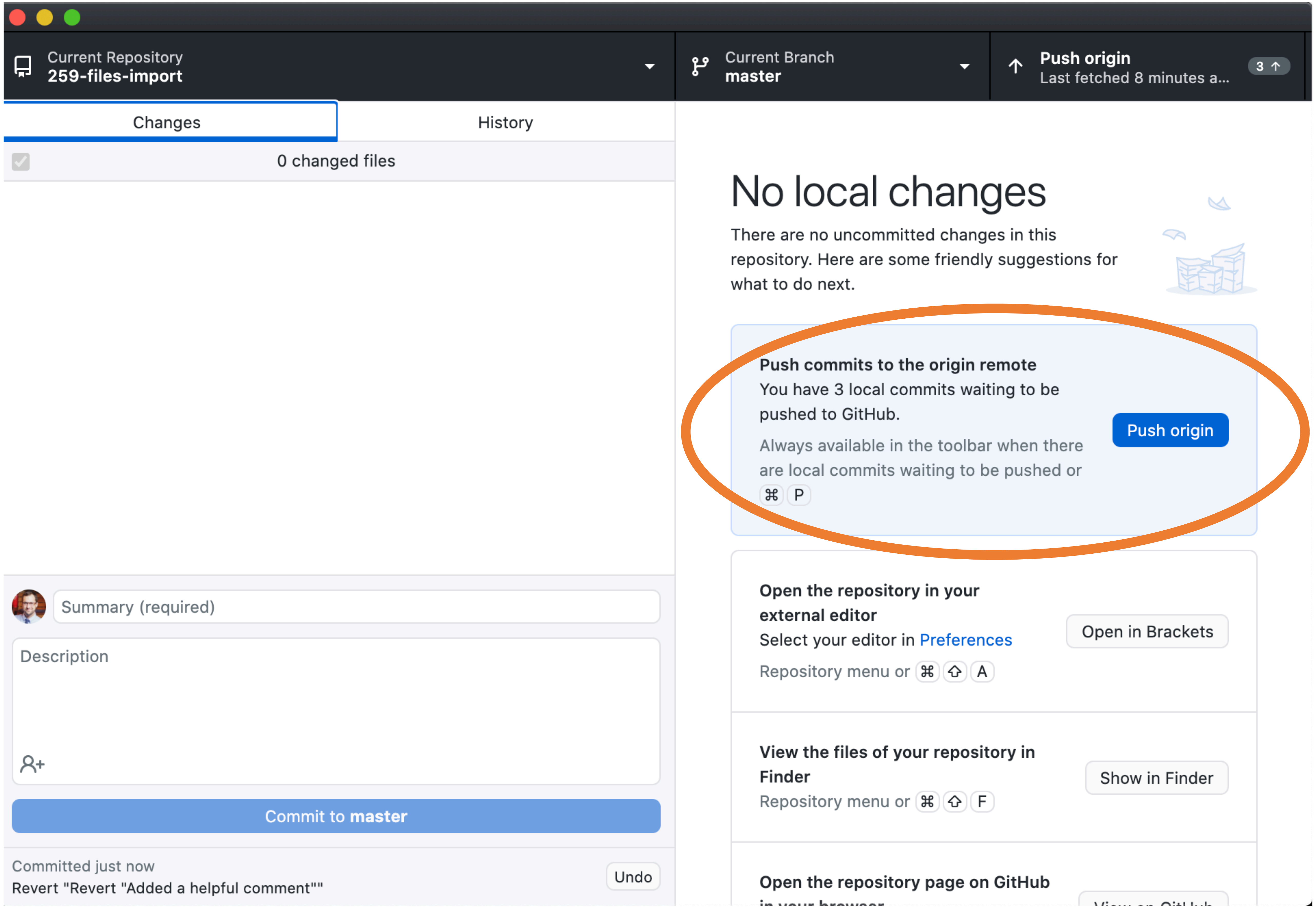


```
@@ -1,6 +1,6 @@
 #Load packages
 library(tidyverse)
-library(visdat)
+library(visdat) #"Visualize data"

 rm(list = ls()) #clean variables out of environme
nt

```

Update 01_vocab.R

Commit those changes

- If you want to keep changes (to one or more files), create a **commit**
  - Check which files to include (**stage**), defaults to all
  - Write a short message and click commit
  - Your changes will disappear ("no local changes") because your local copy is updated
- Next, we **push** those changes to the origin on GitHub



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

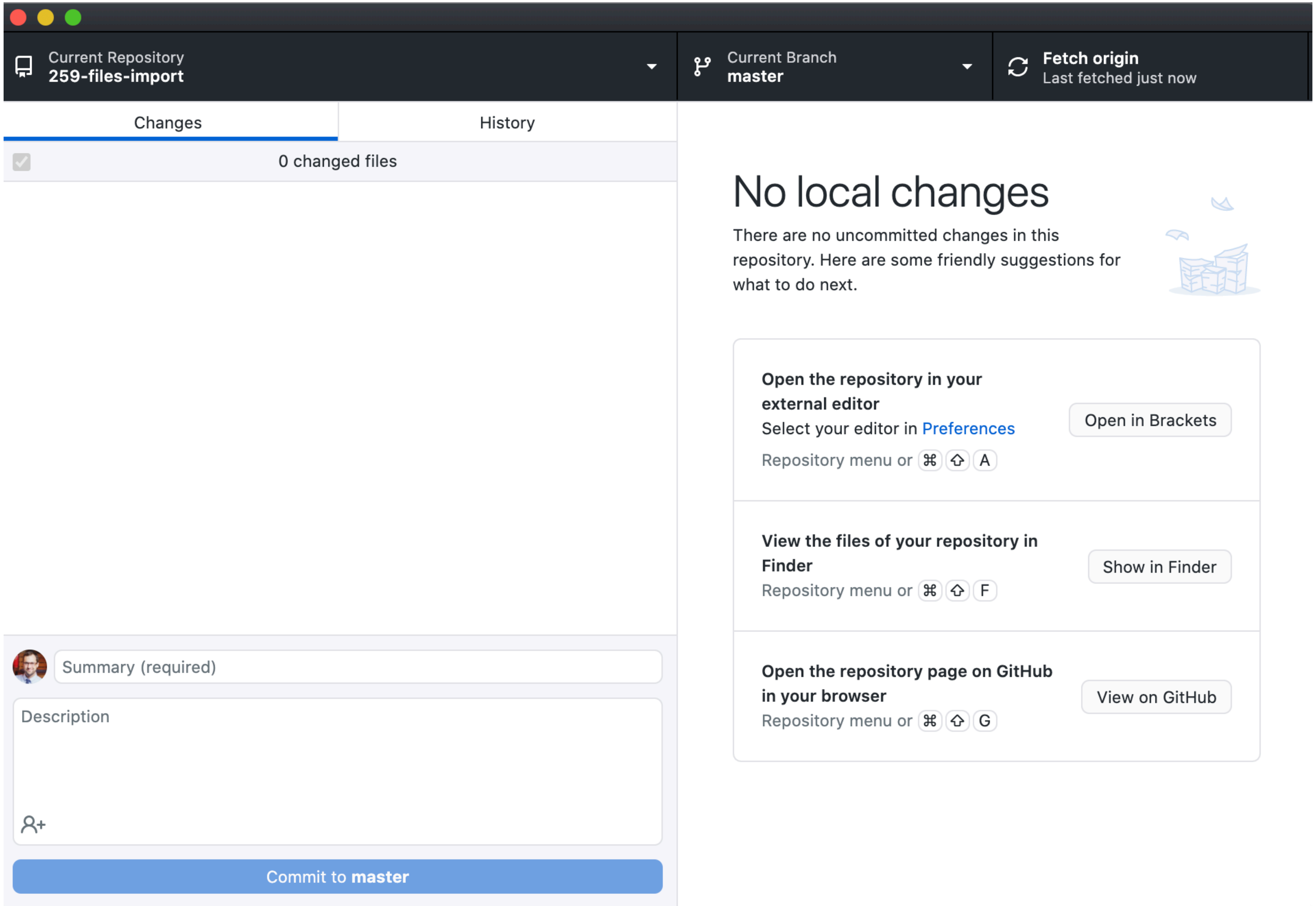
Push commits to the origin remote
You have 3 local commits waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or ⌘ P

Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⇧ A

View the files of your repository in Finder
Repository menu or ⌘ ⇧ F

Open the repository page on GitHub in your browser

Committed just now
Revert "Revert "Added a helpful comment""



Current Repository
259-files-import

Current Branch
master

Fetch origin
Last fetched just now

0 changed files

No local changes

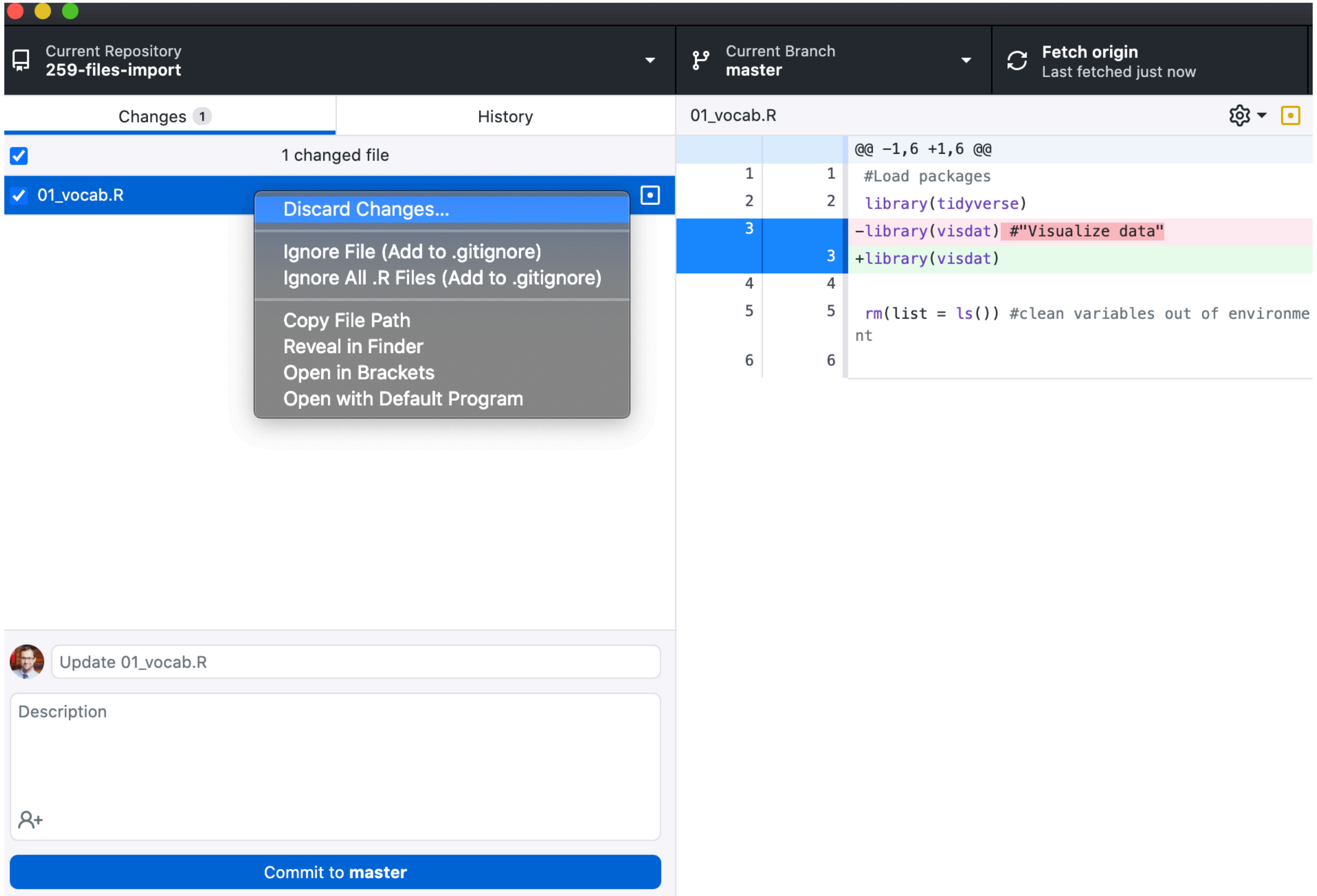
There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⇧ A

View the files of your repository in Finder
Repository menu or ⌘ ⇧ F

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⇧ G

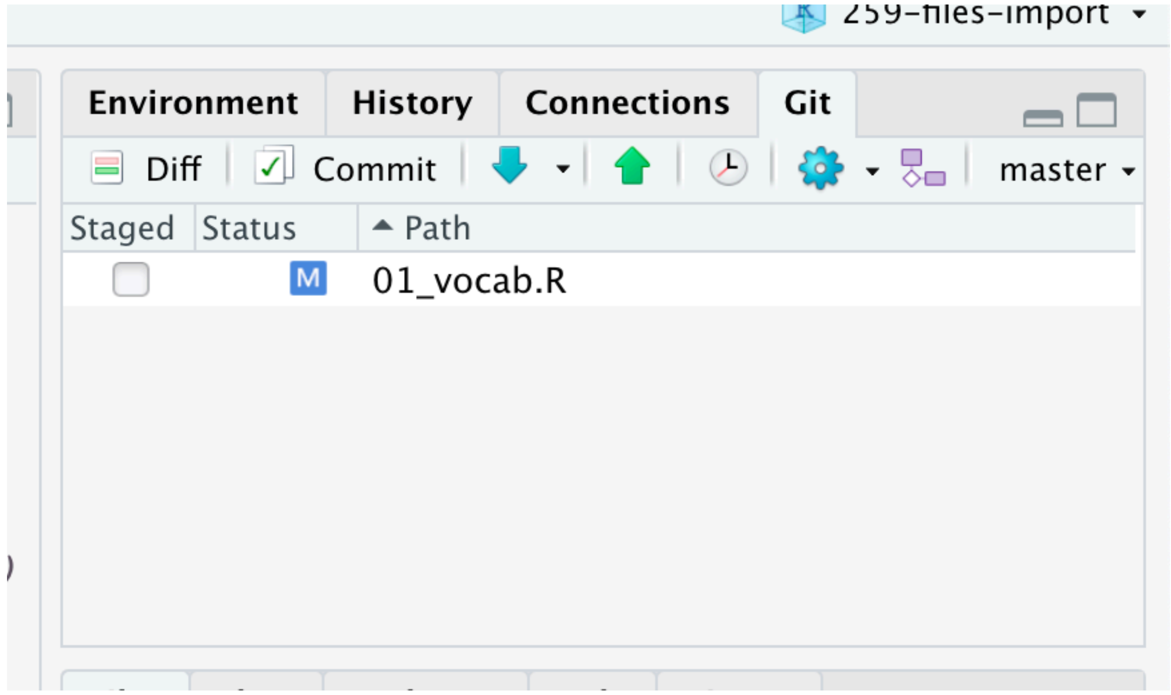
Don’t want those changes?
Discard them!



```
@@ -1,6 +1,6 @@
 #Load packages
 library(tidyverse)
-library(visdat) #"Visualize data"
+library(visdat)

 rm(list = ls()) #clean variables out of environment

```

Make another change in RStudio

- Git menu will appear when working on a project tracked in git
  - Changes will appear
  - You can stage them (checkbox), commit, and then push to origin directly from RStudio

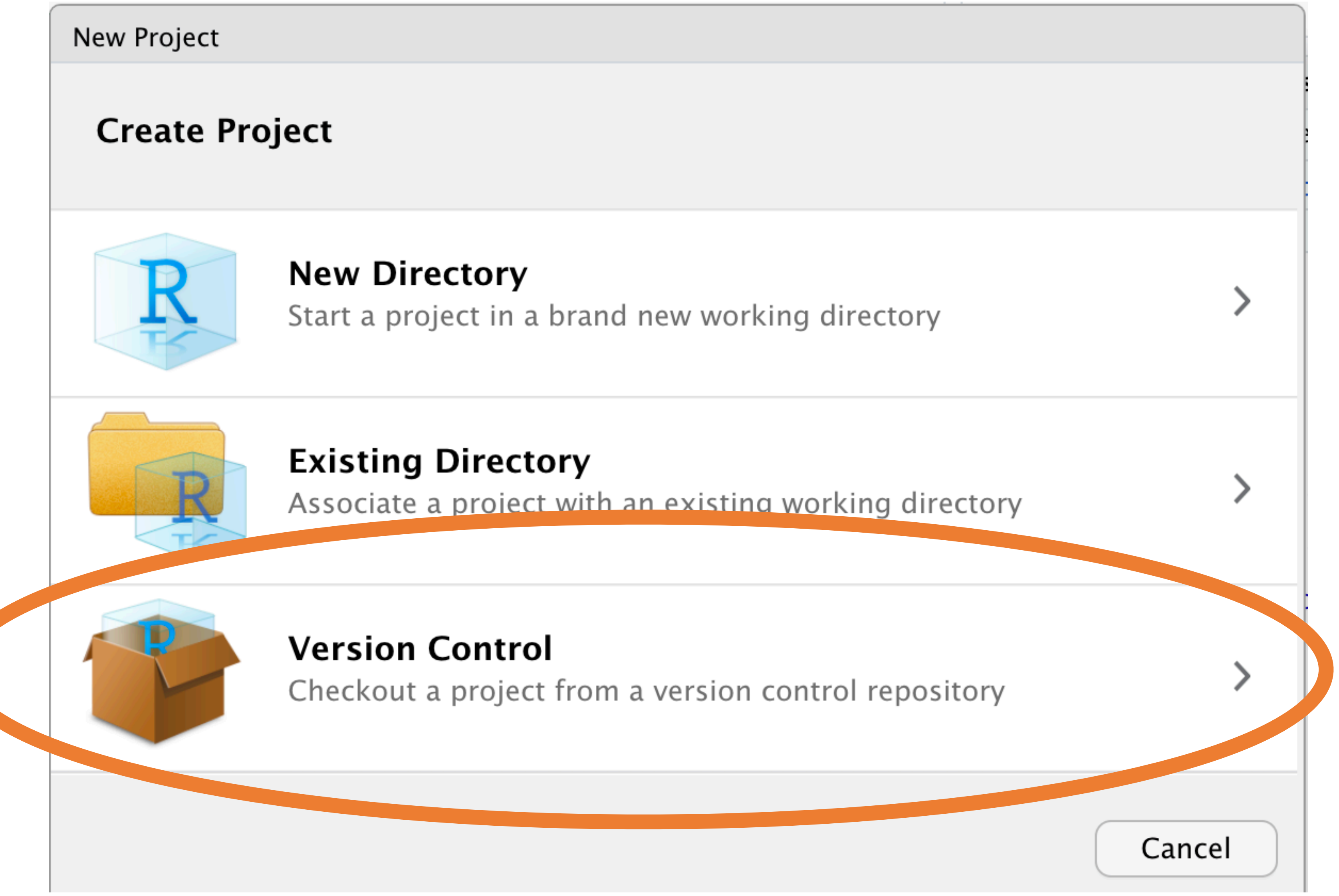


Staged	Status	Path
□	M	01_vocab.R

Other ways to create a new git project

- Clone a project directly into RStudio desktop
- Clone a project directly into RStudio Cloud
- Make any existing folder (such as an RStudio project or other analysis folder) a GitHub repository

Clone a project directly into RStudio from Github



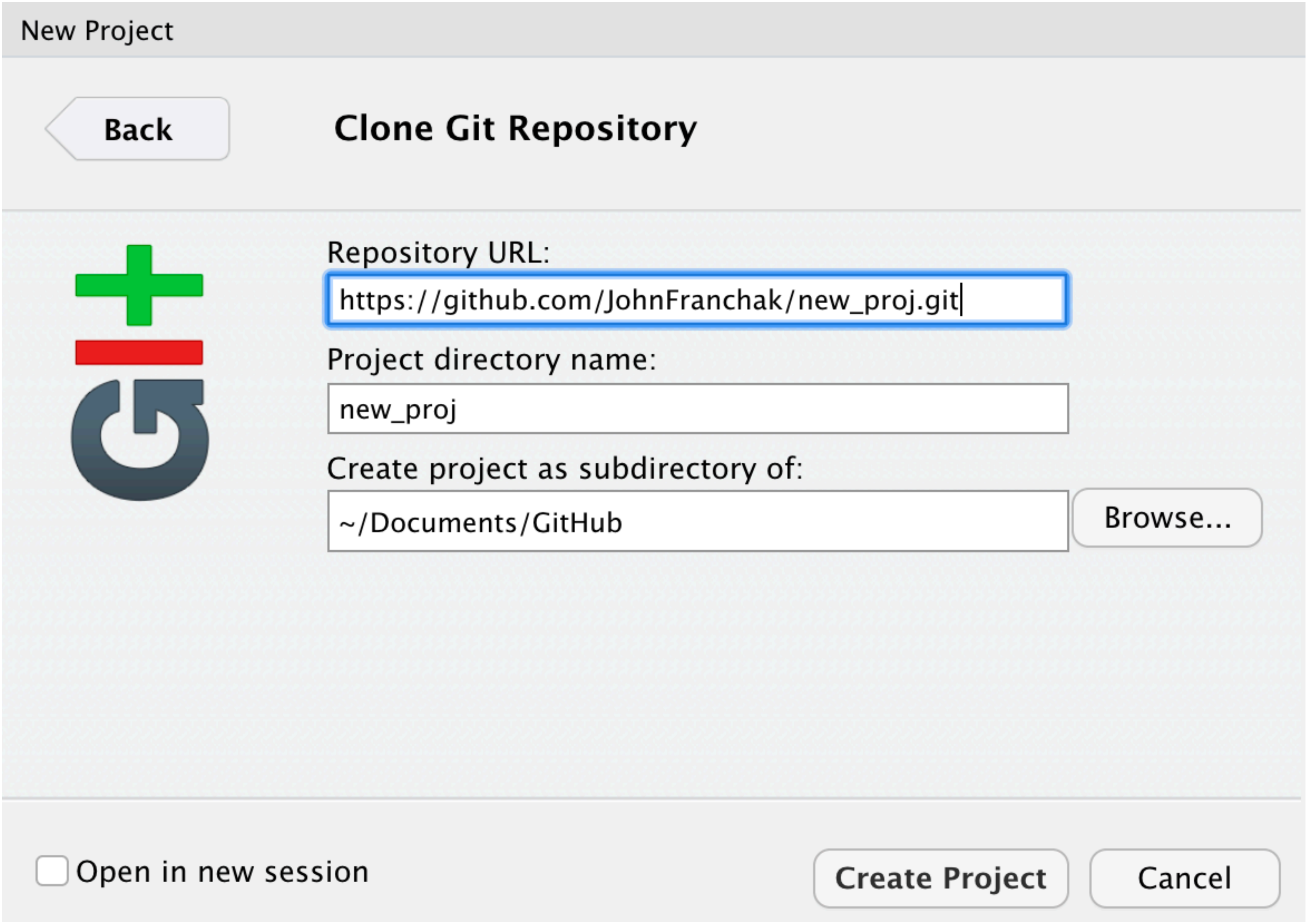
New Project

Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository



New Project

Clone Git Repository

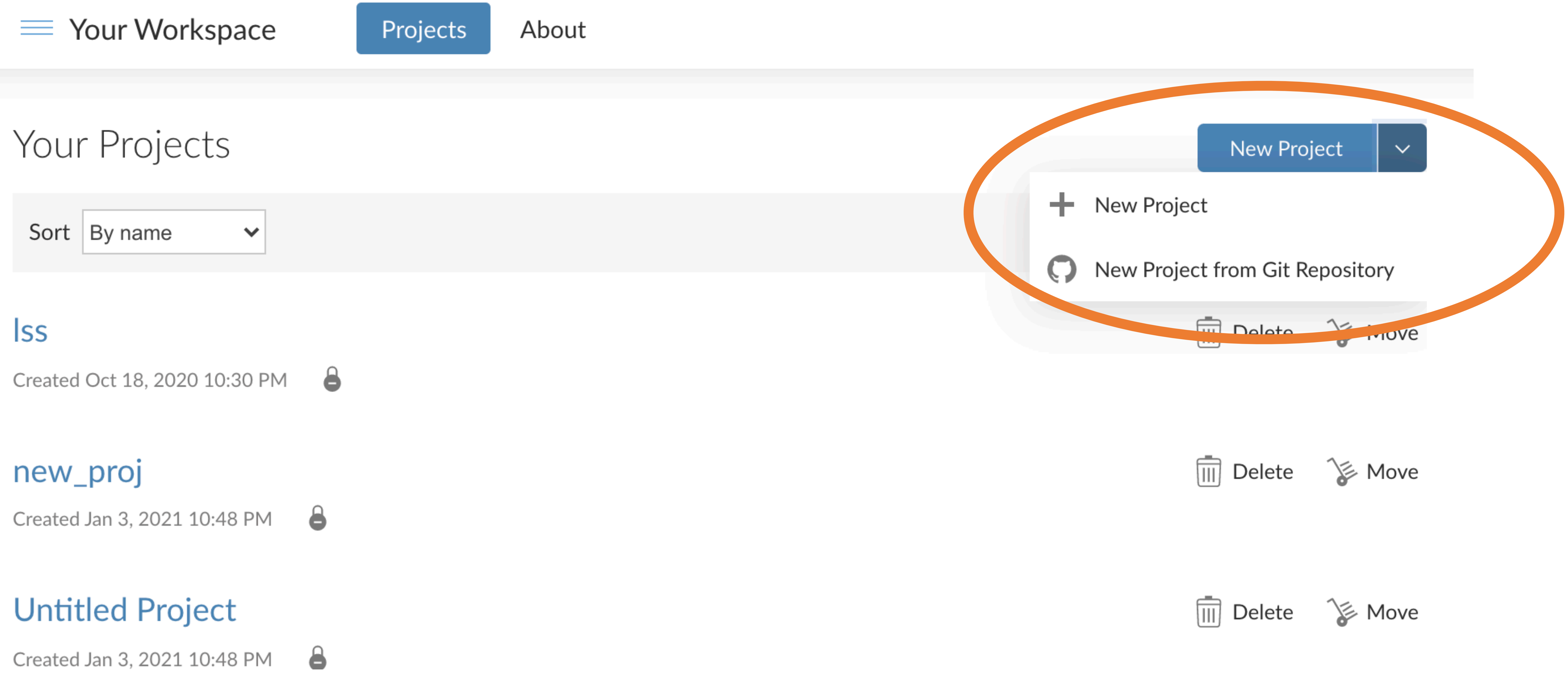
Repository URL: https://github.com/JohnFranchak/new_proj.git

Project directory name: new_proj

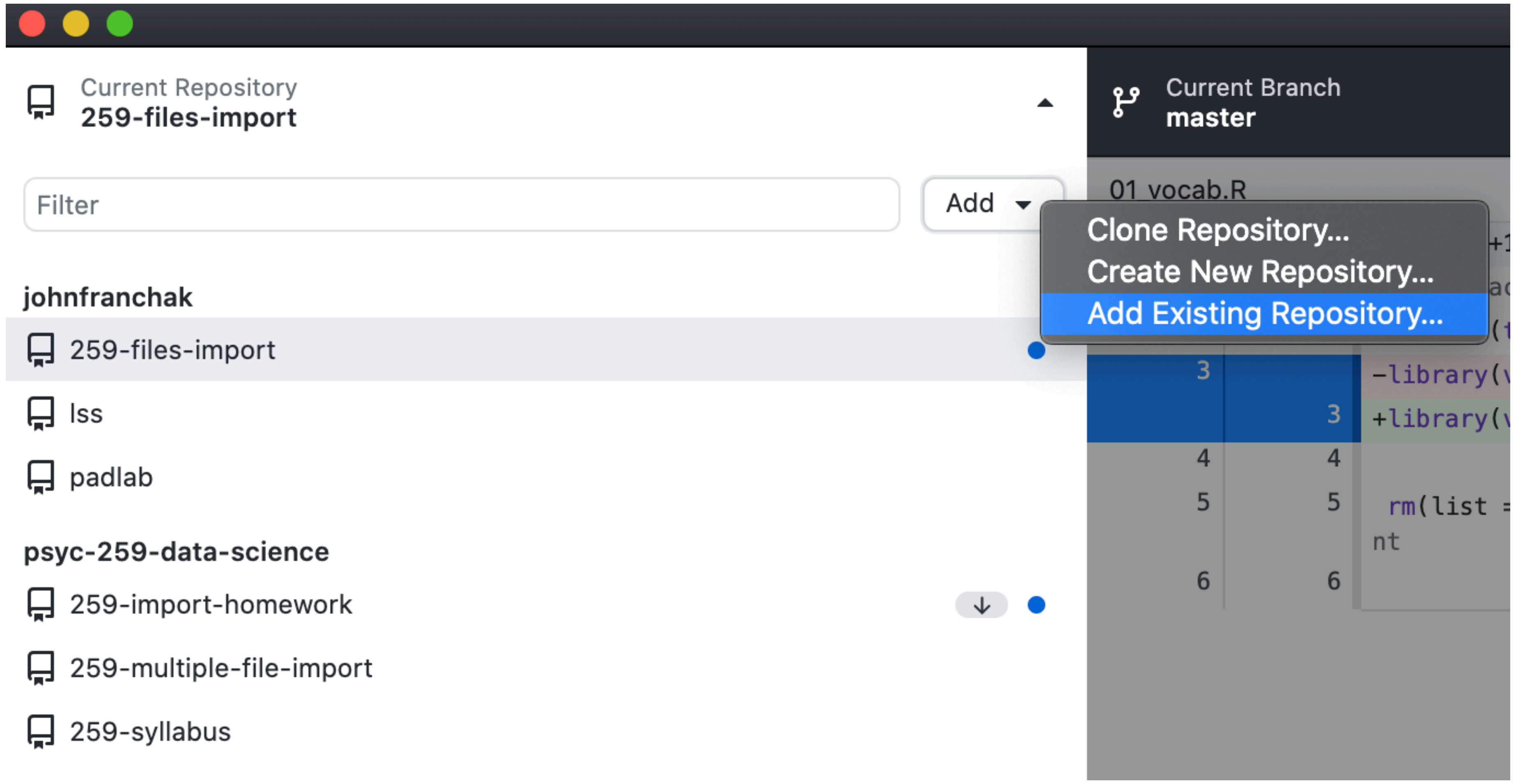
Create project as subdirectory of: ~/Documents/GitHub

☐ Open in new session

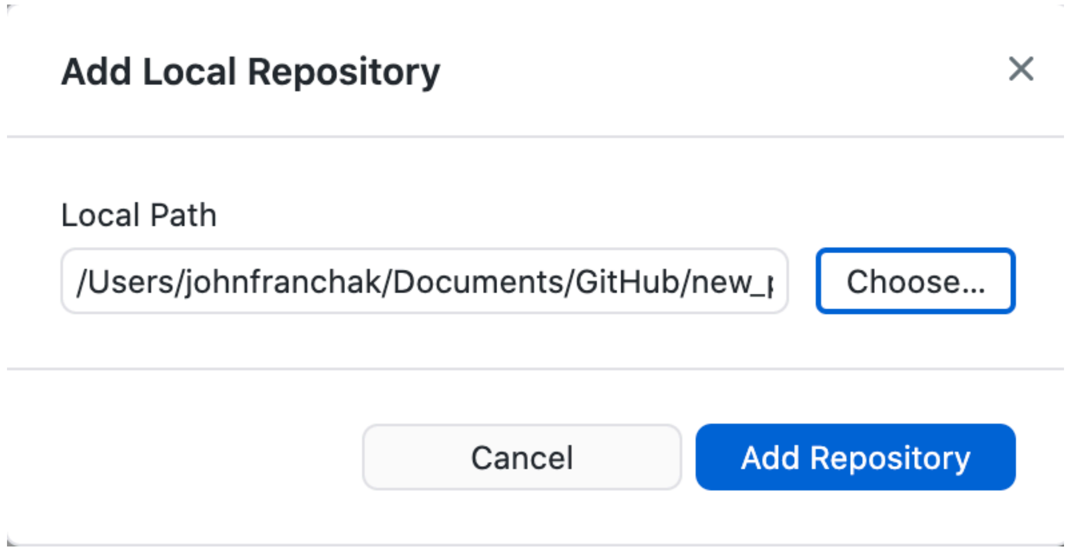
Clone a project directly into RStudio Cloud from Github



Add existing (local) repository to the Github app



```
3        -library(v
      3  +library(v
4     4
5     5   rm(list =
         nt
6     6
```



Add Local Repository

Local Path

/Users/johnfranchak/Documents/GitHub/new_

Homework 1

Answers are on GitHub

- https://github.com/PSYC-259-Data-Science/259-import-homework
- Fork and then clone the Github project to get the answers on your local machine and run them
  - install.packages(“tidyverse”, “vroom”) etc. if you want to run the code and don’t have packages installed