Lecture 3.2 – Version Control

Learning Objectives:

2.3.1 – Learn basic skills in version control using git (cloning, adding, removing, committing, pushing, and pulling).

2.3.3 – Successfully add content by committing and pushing to the repository on Github using RStudio's git interface.

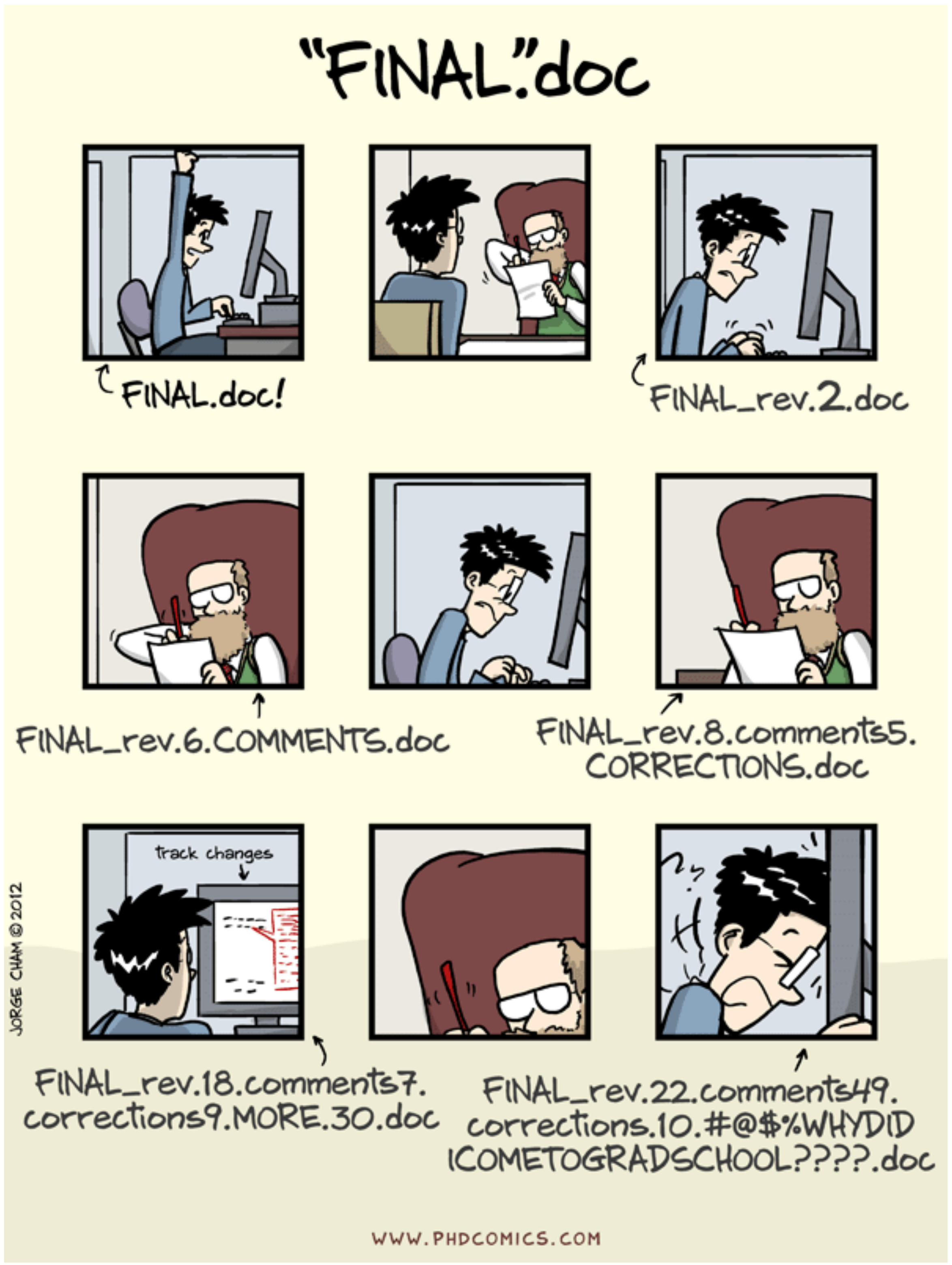
2.3.4 – Maintain a R Project in RStudio using git and Github.

- **Download and Install git:**

git download:
https://git-scm.com/downloads

Put it in default directory

Why Bother with Version Control?



If you "break" your code, how do you get it back working?

If your computer crashes, how do you get back your code?

What if you are on a group project and people need to work on the same code at once?

https://swcarpentry.github.io/git-novice/01-basics/index.html

What is Automated Version Control?

- **Automated version control (VC)** is the process by which someone uses specialized software to keep track of changes made to a digital document.

  - Several different types of VC software, with a long history. (Classic example is track changes!)

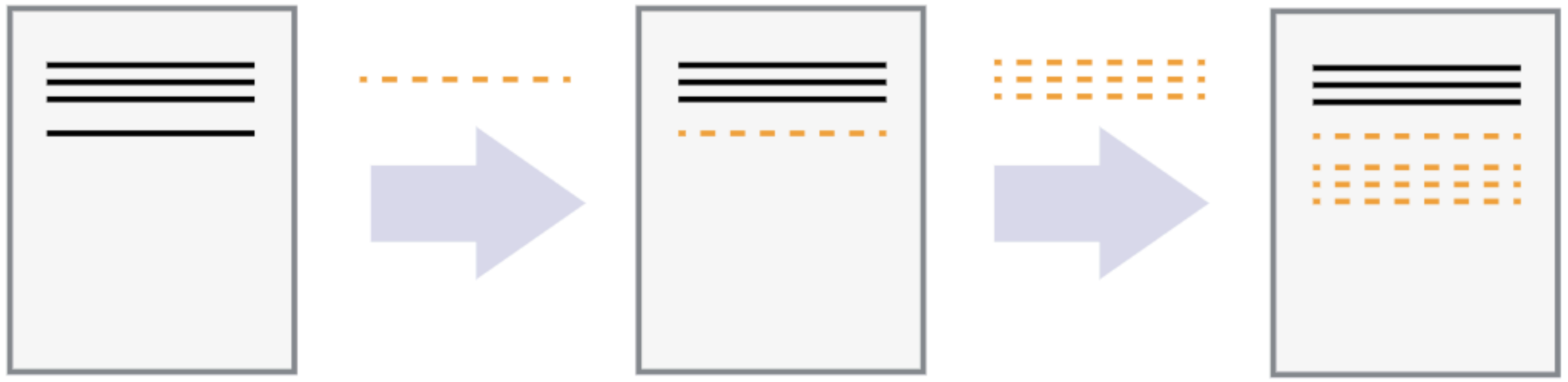
  - For code, software like SVN and git are often used to track line-by-line changes.

  - VC is like the ultimate undo button: you can undo or revert any change to a document that you've tracked.

  - VC also allows multiple users to work on documents simultaneously, keeping only the best edits in the main document!

How Does Automated Version Control Work?

- VC works by keeping a record of changes to individual lines of a document.
    - Changes are made by one user to an original document, and *how the document has changed* is recorded.

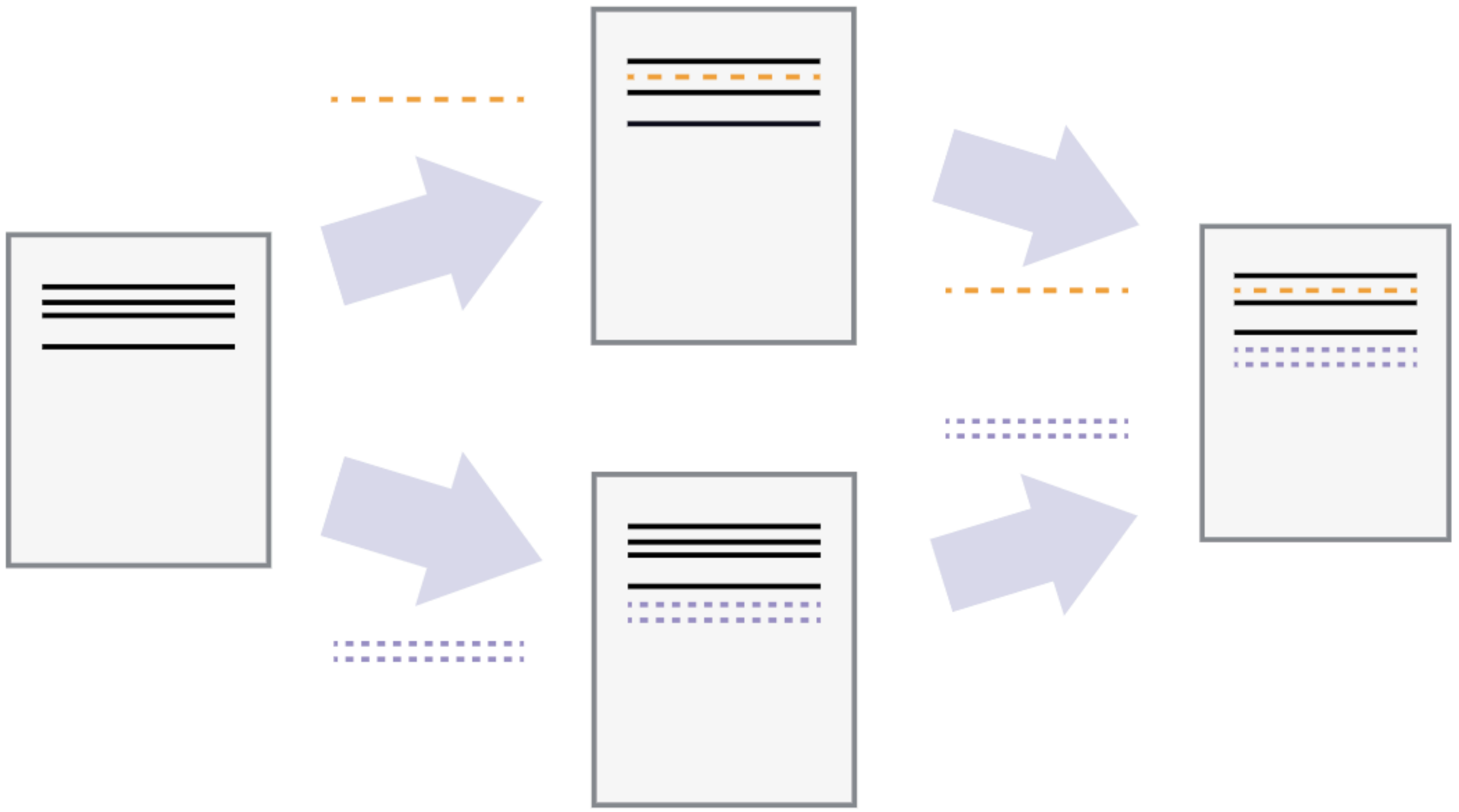


    - Note: this does not mean it makes a full copy of the original (like you might if you had two separate versions saved on your computer). **It only tracks what has changed!**

https://swcarpentry.github.io/git-novice/01-basics/index.html

How Does Automated Version Control Work?

- If two people are working on the same document, VC will keep track of these changes separately.



- Later, VC allows the user to combine the changes back into the original document.

https://swcarpentry.github.io/git-novice/01-basics/index.html

How Does Automated Version Control Work?

- Why not use Track Changes or Google Docs for doing code?

    - MS Word (and other word-processing programs) are complex editors that introduce many “hidden” characters into files, as well as formatting/typesetting code (again, hidden).

    - This extra, hidden code often messes up code meant to run in a different language. Scripts need to edited in a script editor (to prevent these sorts of issues).

I heavily advise against trying to edit/copy code from any word processing documents!
USE R STUDIO!!!

About git

- git is a popular VC software for coding that supports nearly all document types.

    - It has a simple interface through RStudio we can use to track our code.

    - git easily integrates with Github, a website for collective management of software projects.

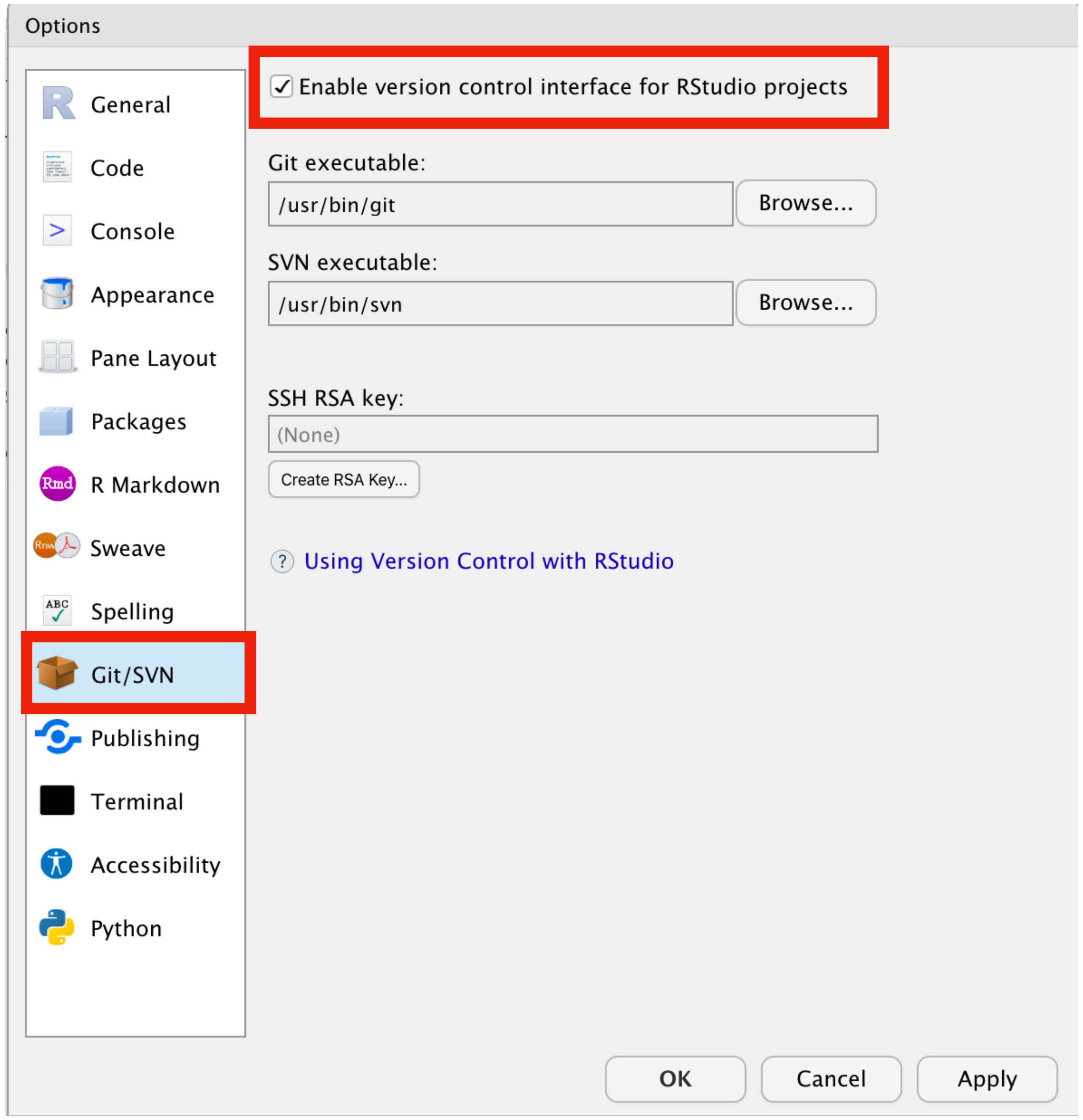
- **Download and Install git:**

git download:
https://git-scm.com/downloads

Put it in default directory

Version Control in RStudio

- RStudio has built-in support for git when using RProjects!
  - Activate VC by going to "Preferences…"
  - Select "Git/SVN"
  - Check box that says "Enable version control interface"



✓ Enable version control interface for RStudio projects

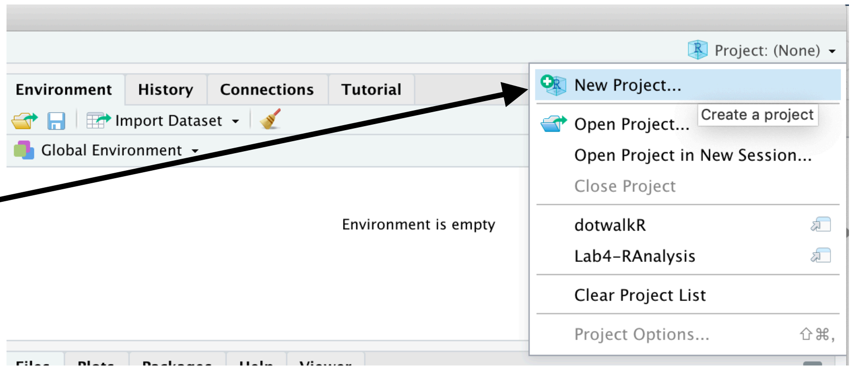
Git executable: /usr/bin/git

SVN executable: /usr/bin/svn

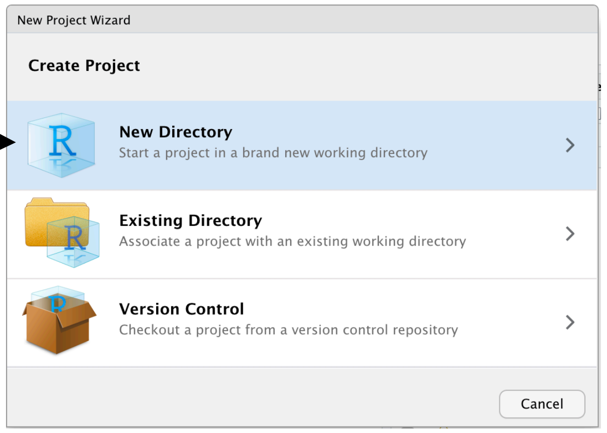
SSH RSA key: (None)

Creating an R Project

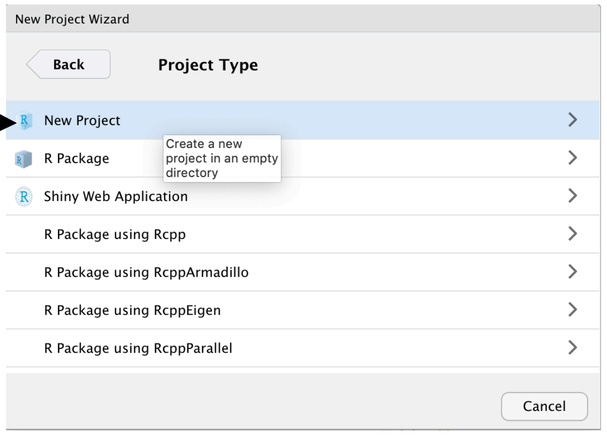
1. Either under "File…" or at the top right corner, select "New Project…"



2. Select "New Directory" in the Wizard.

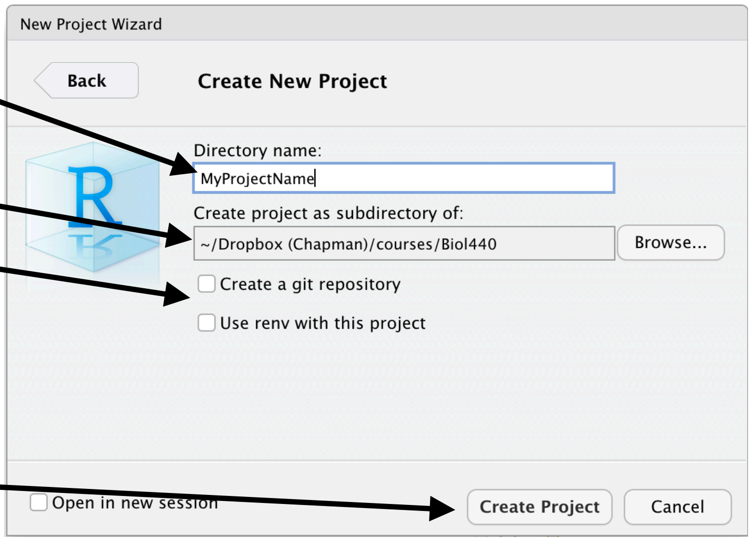

New Project Wizard
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel

3. Select "New Project" again.


New Project Wizard
Back
Project Type
New Project
Create a new project in an empty directory
R Package
Shiny Web Application
R Package using Rcpp
R Package using RcppArmadillo
R Package using RcppEigen
R Package using RcppParallel
Cancel

4. Create a directory name for your project. Choose where you want the project directory to be (using Browse…). Check "create a git repository".

5. Click "Create Project"!



Directory name: MyProjectName

Create project as subdirectory of: ~/Dropbox (Chapman)/courses/Biol440

☐ Create a git repository

☐ Use renv with this project

☐ Open in new session

Basics of Operating git

Your computer

Saved!

```
Your code
New code
New code
```

You add some new code.

→ add filename →

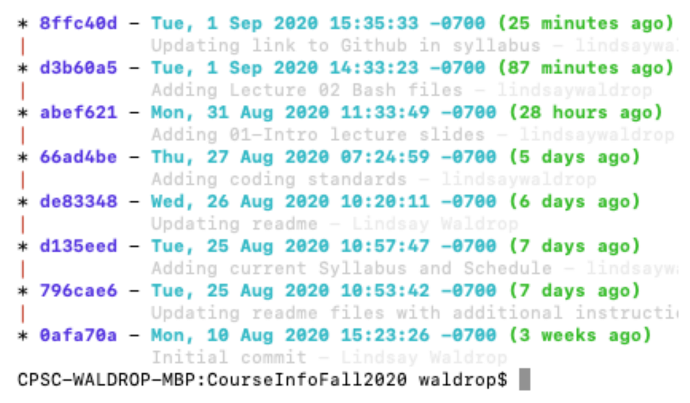
Staged!

```
Your code
New code
New code
```

Changes to scripts recorded and catalogued.

→ commit →

Your git repository

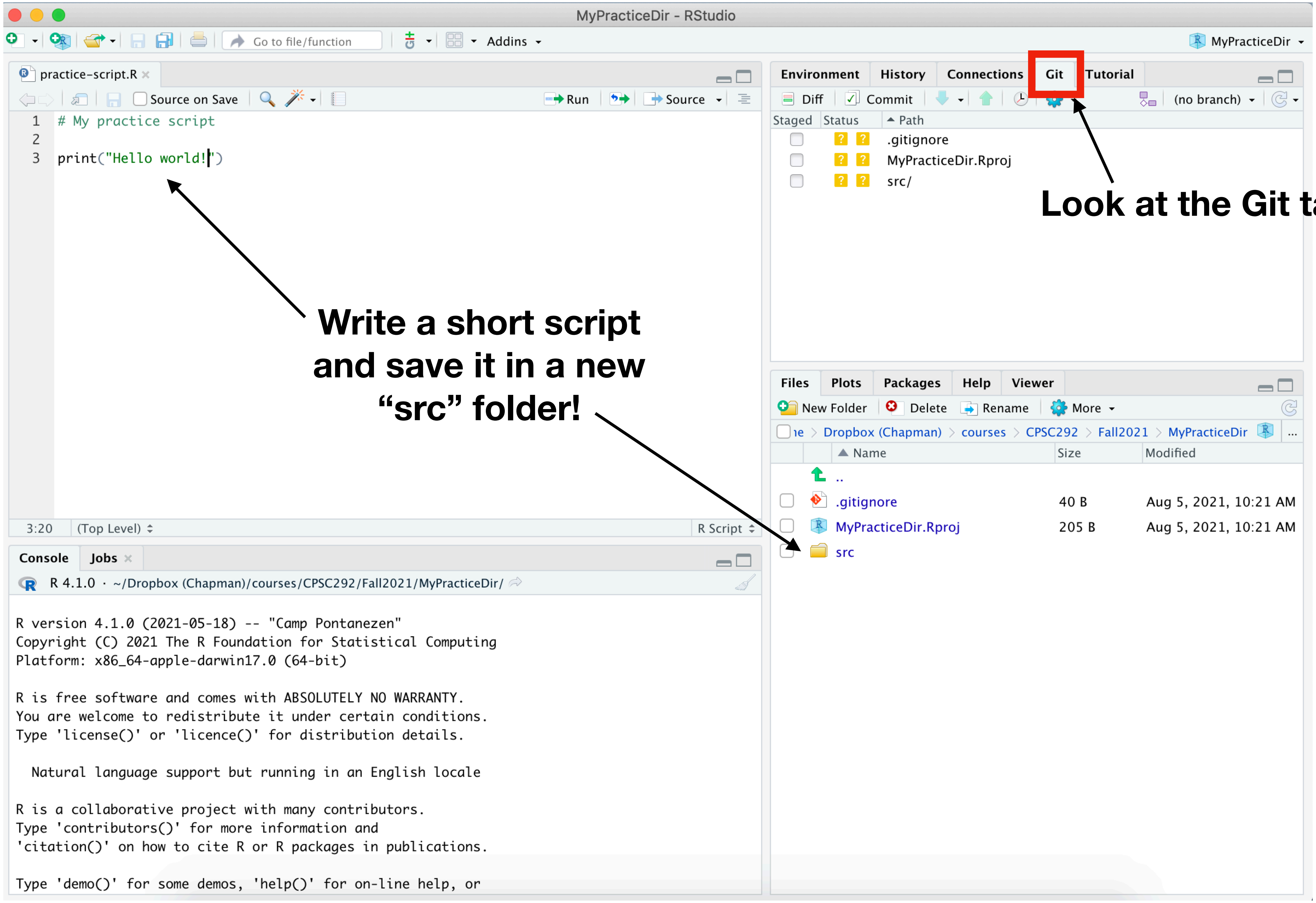


```
* 8ffc40d - Tue, 1 Sep 2020 15:35:33 -0700 (25 minutes ago)
|           Updating link to Github in syllabus - lindsaywa
* d3b60a5 - Tue, 1 Sep 2020 14:33:23 -0700 (87 minutes ago)
|           Adding Lecture 02 Bash files - lindsaywaldrop
* abef621 - Mon, 31 Aug 2020 11:33:49 -0700 (28 hours ago)
|           Adding 01-Intro lecture slides - lindsaywaldrop
* 66ad4be - Thu, 27 Aug 2020 07:24:59 -0700 (5 days ago)
|           Adding coding standards - lindsaywaldrop
* de83348 - Wed, 26 Aug 2020 10:20:11 -0700 (6 days ago)
|           Updating readme - Lindsay Waldrop
* d135eed - Tue, 25 Aug 2020 10:57:47 -0700 (7 days ago)
|           Adding current Syllabus and Schedule - lindsayw
* 796cae6 - Tue, 25 Aug 2020 10:53:42 -0700 (7 days ago)
|           Updating readme files with additional instructi
* 0afa70a - Mon, 10 Aug 2020 15:23:26 -0700 (3 weeks ago)
            Initial commit - Lindsay Waldrop
CPSC-WALDROP-MBP:CourseInfoFall2020 waldrop$
```

Staged changes added to history

Basics of Operating git in RStudio

1. Save files to your project.



```
# My practice script

print("Hello world!")
```

Staged	Status	Path
☐	? ?	.gitignore
☐	? ?	MyPracticeDir.Rproj
☐	? ?	src/

ıe > Dropbox (Chapman) > courses > CPSC292 > Fall2021 > MyPracticeDir

Name	Size	Modified
..		
.gitignore	40 B	Aug 5, 2021, 10:21 AM
MyPracticeDir.Rproj	205 B	Aug 5, 2021, 10:21 AM
src		

R 4.1.0 · ~/Dropbox (Chapman)/courses/CPSC292/Fall2021/MyPracticeDir/

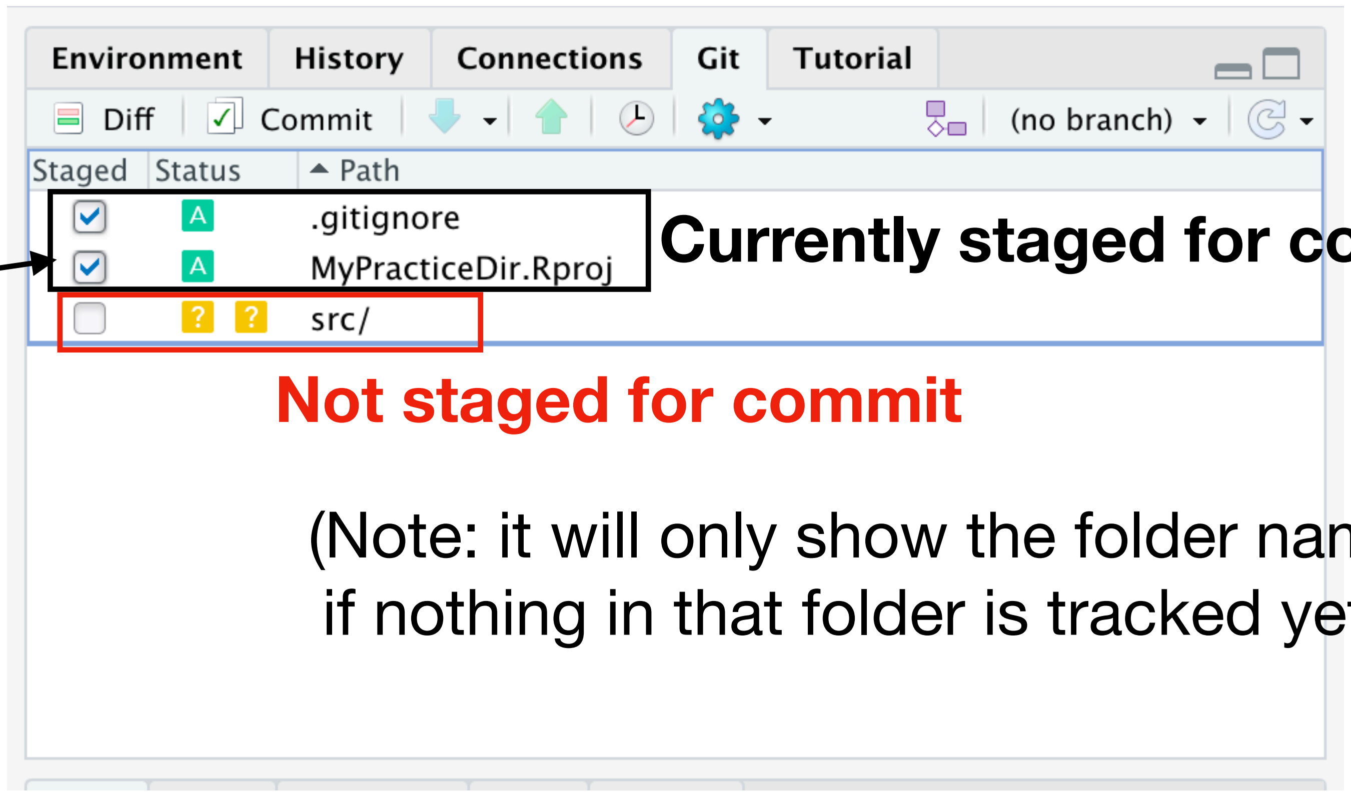
```
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
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
```

Basics of Operating git in RStudio

2. Stage files for commit by adding.

All new files will have the ?? icons.

Add files by clicking the “Staged” checkbox!



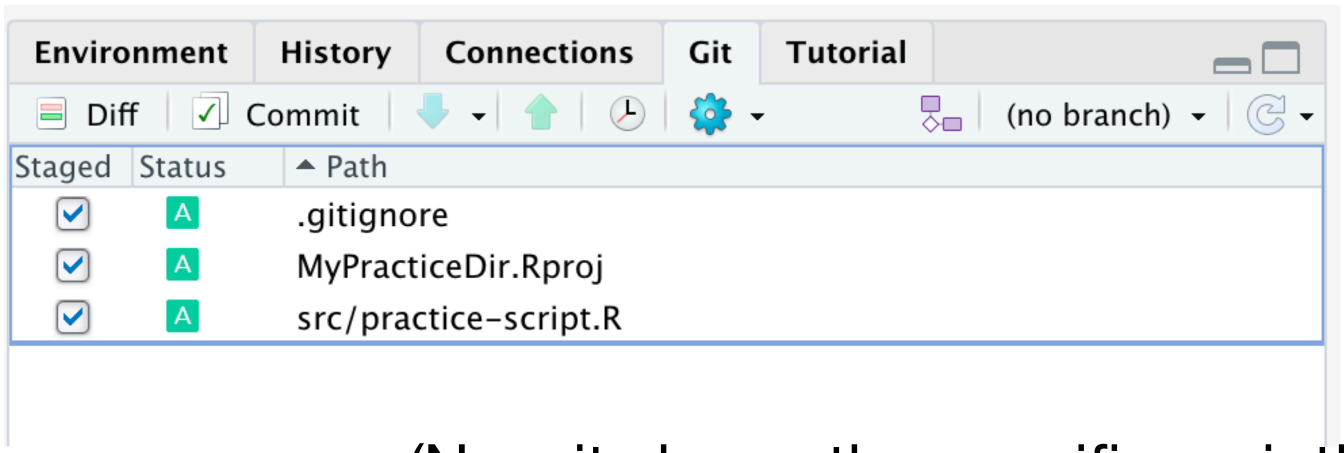
Staged	Status	Path
✓	A	.gitignore
✓	A	MyPracticeDir.Rproj
	? ?	src/

Currently staged for commit

Not staged for commit

(Note: it will only show the folder name if nothing in that folder is tracked yet!)

Now everything is staged!



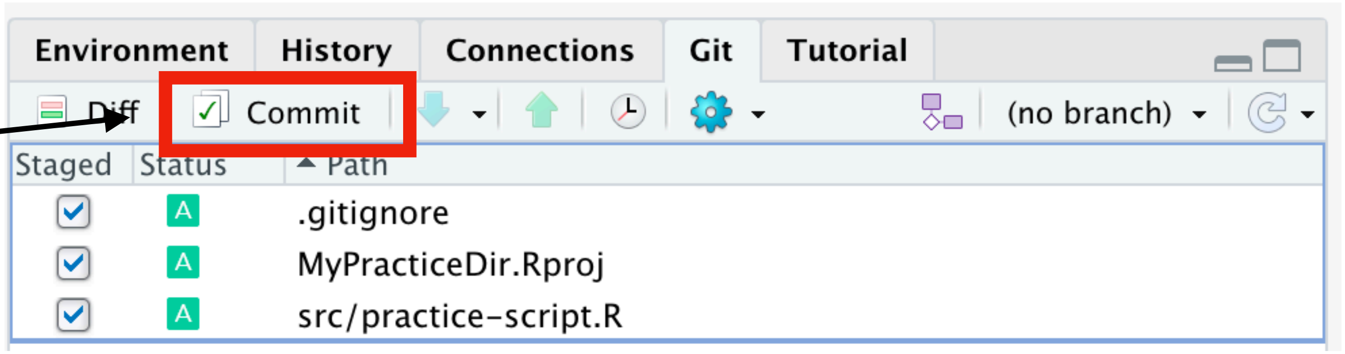
Staged	Status	Path
✓	A	.gitignore
✓	A	MyPracticeDir.Rproj
✓	A	src/practice-script.R

(Now it shows the specific script!)

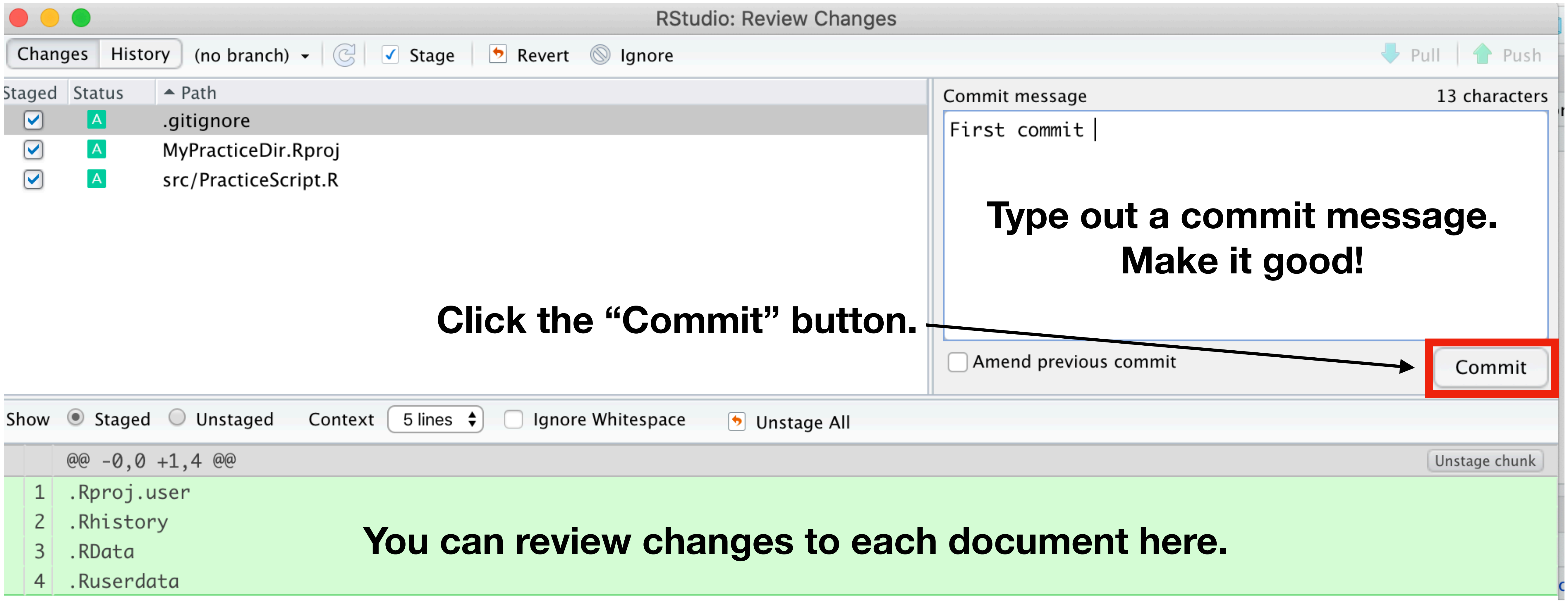
Basics of Operating git in RStudio

3. Commit your changes.

Click the “Commit” button.



Staged	Status	Path
✓	A	.gitignore
✓	A	MyPracticeDir.Rproj
✓	A	src/practice-script.R



Staged	Status	Path
✓	A	.gitignore
✓	A	MyPracticeDir.Rproj
✓	A	src/PracticeScript.R

```
First commit
```

Type out a commit message. Make it good!

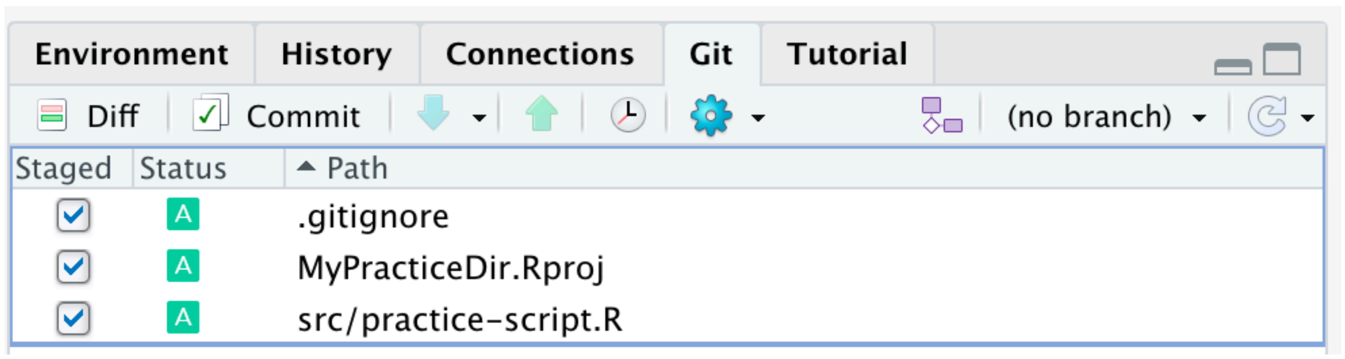
Click the “Commit” button.

```
@@ -0,0 +1,4 @@
.Rproj.user
.Rhistory
.RData
.Ruserdata
```

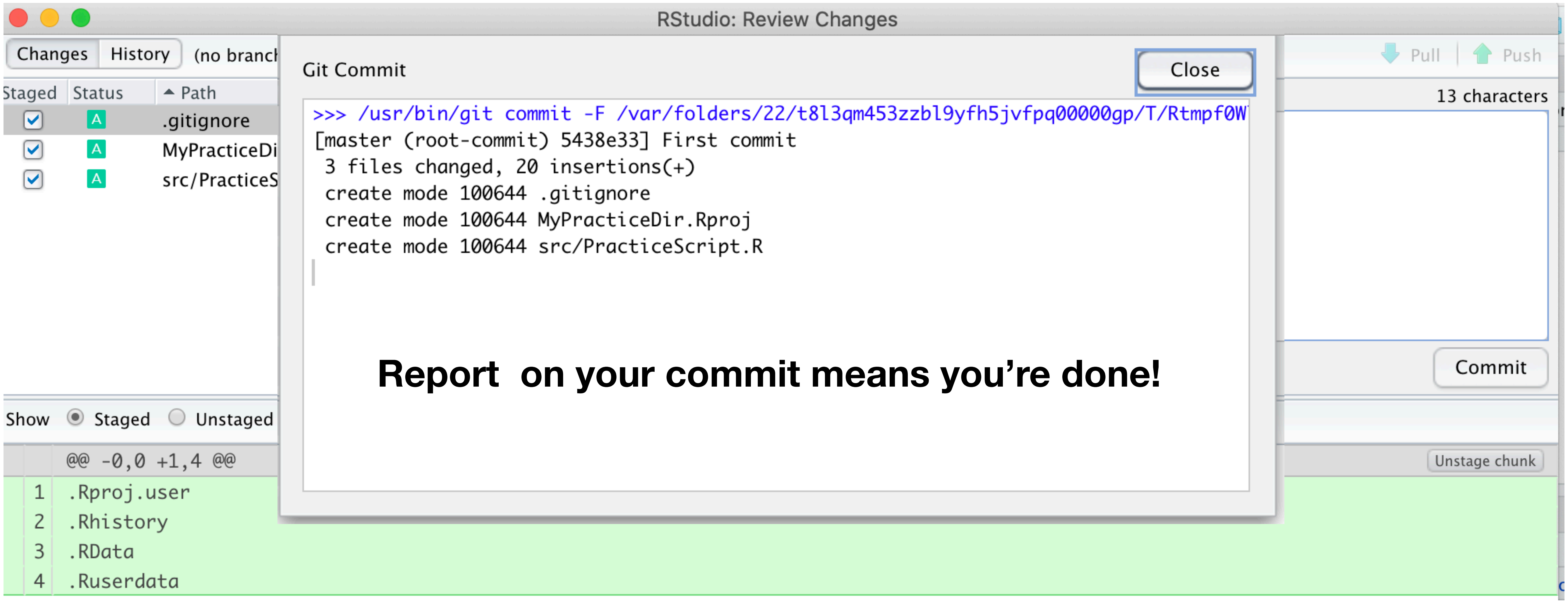
You can review changes to each document here.

Basics of Operating git in RStudio

3. Commit your changes.



Staged	Status	Path
✓	A	.gitignore
✓	A	MyPracticeDir.Rproj
✓	A	src/practice-script.R



```
>>> /usr/bin/git commit -F /var/folders/22/t8l3qm453zzbl9yfh5jvfpq00000gp/T/RtmpfOW
[master (root-commit) 5438e33] First commit
 3 files changed, 20 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 MyPracticeDir.Rproj
 create mode 100644 src/PracticeScript.R
```

Report on your commit means you're done!

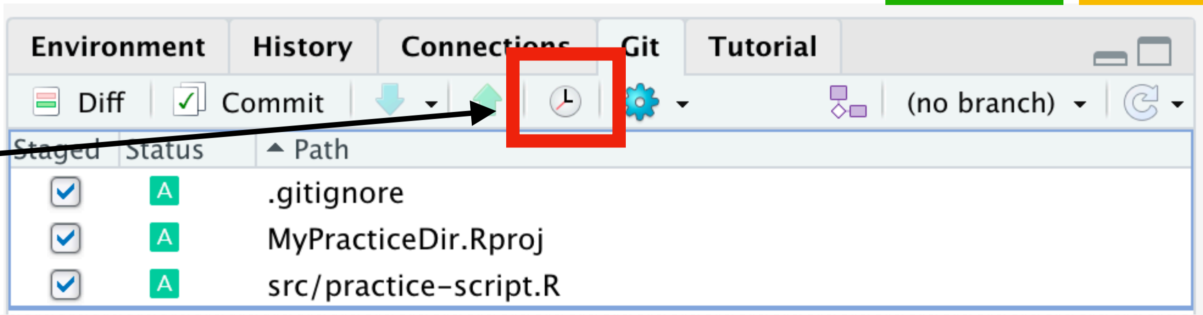
```
@@ -0,0 +1,4 @@
.Rproj.user
.Rhistory
.RData
.Ruserdata
```

Check Your Understanding

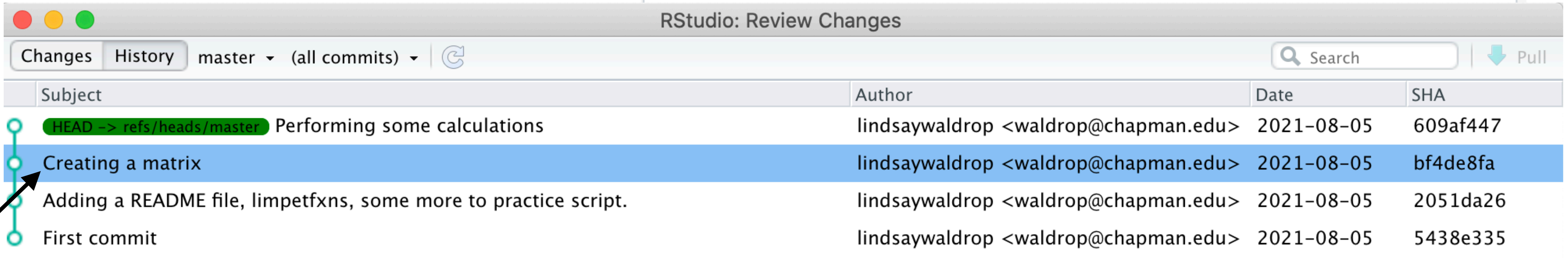
Create a few more files, and make at least two more commits to your repository!

Examining Your Commit History

- Access your commit history by clicking the clock.

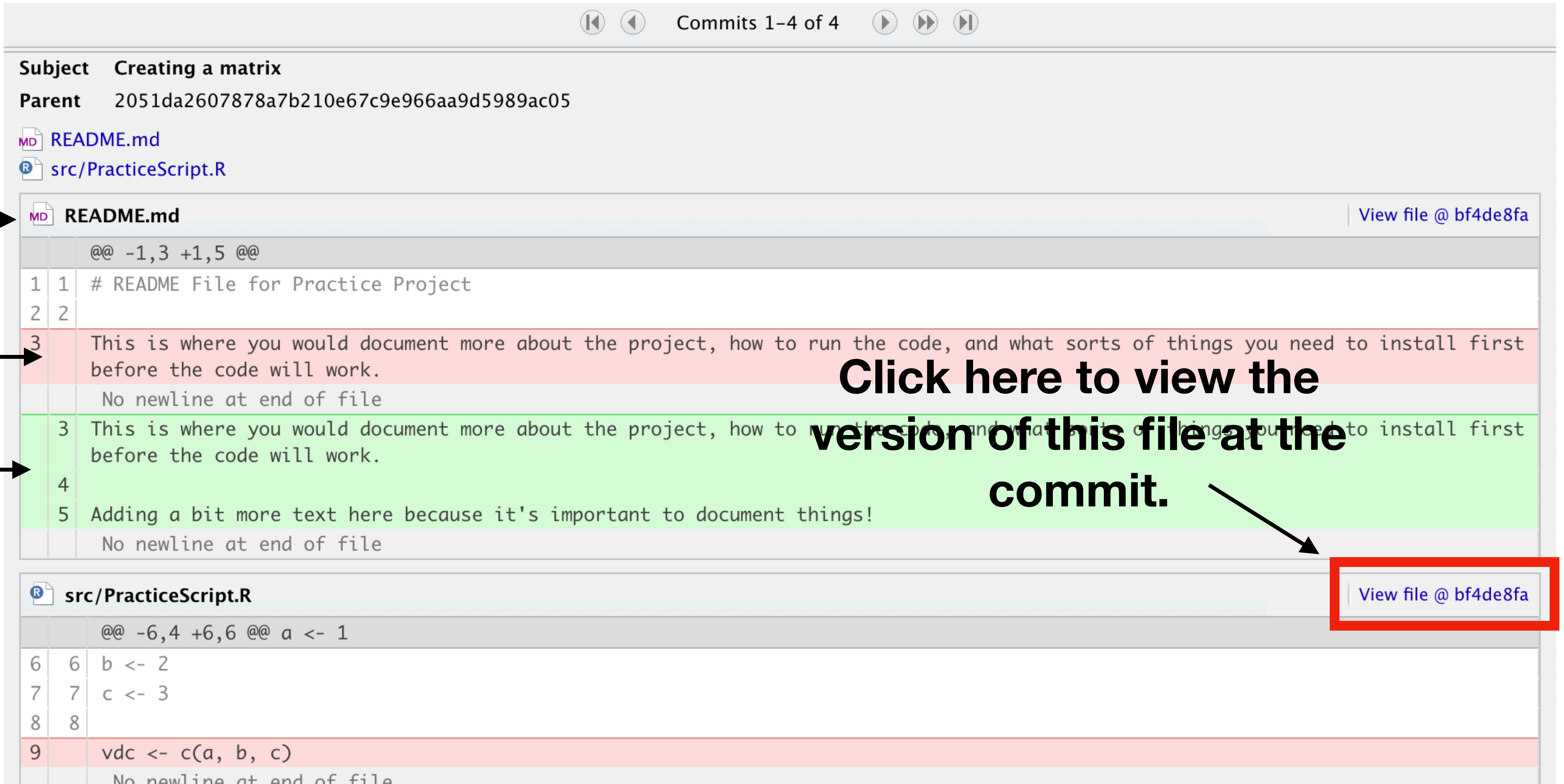


Staged	Status	Path
✓	A	.gitignore
✓	A	MyPracticeDir.Rproj
✓	A	src/practice-script.R



Subject	Author	Date	SHA
HEAD -> refs/heads/master Performing some calculations	lindsaywaldrop <waldrop@chapman.edu>	2021-08-05	609af447
Creating a matrix	lindsaywaldrop <waldrop@chapman.edu>	2021-08-05	bf4de8fa
Adding a README file, limpetfxns, some more to practice script.	lindsaywaldrop <waldrop@chapman.edu>	2021-08-05	2051da26
First commit	lindsaywaldrop <waldrop@chapman.edu>	2021-08-05	5438e335

Click on different commits to view history.



Subject Creating a matrix
Parent 2051da2607878a7b210e67c9e966aa9d5989ac05

README.md
src/PracticeScript.R

Filename → README.md — View file @ bf4de8fa

```
@@ -1,3 +1,5 @@
# README File for Practice Project

This is where you would document more about the project, how to run the code, and what sorts of things you need to install first
before the code will work.
 No newline at end of file
This is where you would document more about the project, how to run the code, and what sorts of things you need to install first
before the code will work.

Adding a bit more text here because it's important to document things!
 No newline at end of file
```

Removals →

Additions →

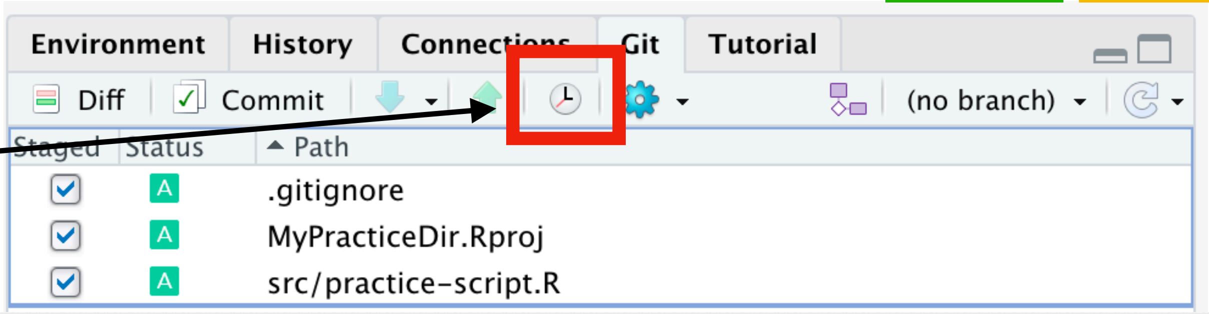
Click here to view the version of this file at the commit. → View file @ bf4de8fa

src/PracticeScript.R — View file @ bf4de8fa

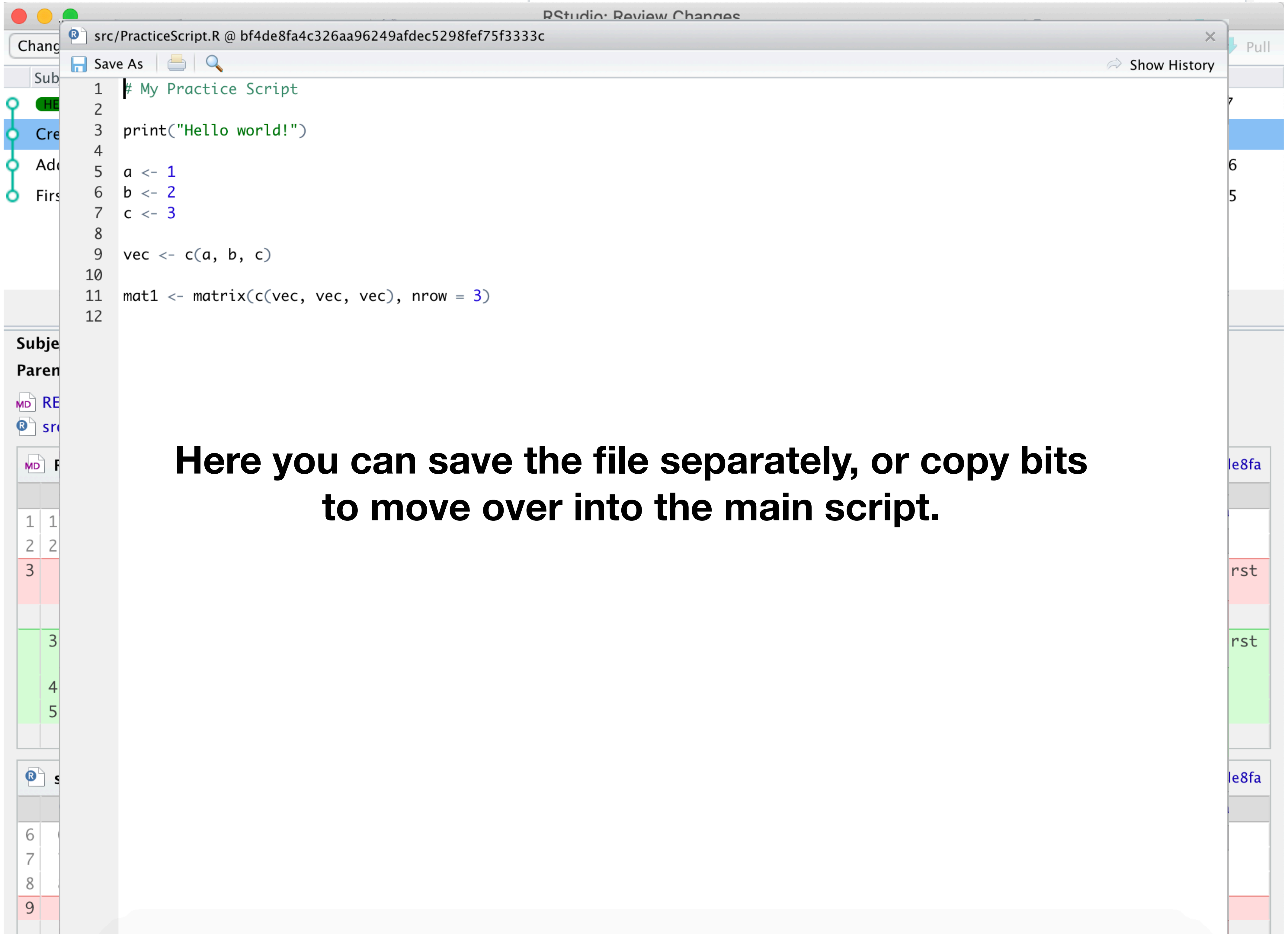
```
@@ -6,4 +6,6 @@ a <- 1
b <- 2
c <- 3

vdc <- c(a, b, c)
 No newline at end of file
```

Examining Your Commit History

- Access your commit history by clicking the clock.



Staged	Status	Path
✓	A	.gitignore
✓	A	MyPracticeDir.Rproj
✓	A	src/practice-script.R


RStudio: Review Changes
src/PracticeScript.R @ bf4de8fa4c326aa96249afdec5298fef75f3333c
Save As
Show History
Pull

```r
# My Practice Script

print("Hello world!")

a <- 1
b <- 2
c <- 3

vec <- c(a, b, c)

mat1 <- matrix(c(vec, vec, vec), nrow = 3)

```

Here you can save the file separately, or copy bits to move over into the main script.

Check Your Understanding

Can you access the first version of your practice script through history?

Online Code Repositories

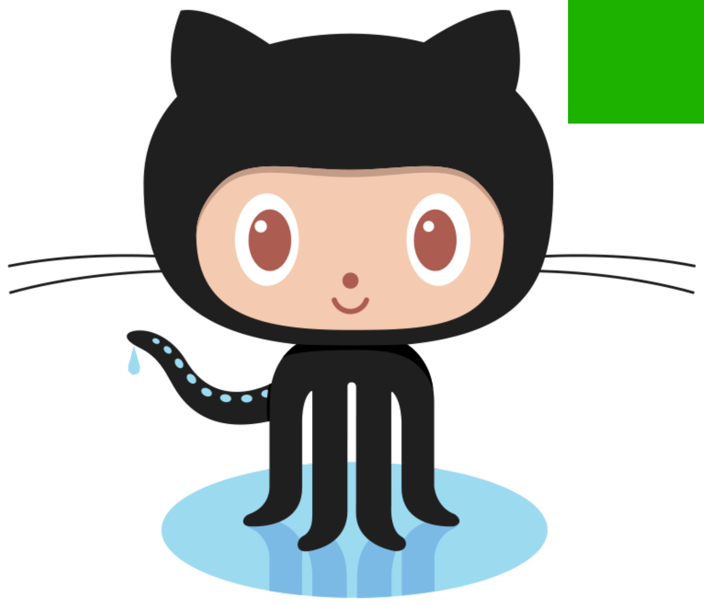
- Why put your code online?
  - Backs up your code in case of computer crash. (private & public)
  - Makes collaborations easier. (private & public)
  - Makes code available to others who wish to replicate your work or use your code. (public)
  - Shows off your projects and skills to future employers! (public)

- **Sign up for Github:**

Github: https://github.com

Use your Chapman email address!

Obtaining a Personal Access Token



- Personal Access Tokens (PATs) are like passwords but more secure. You can select specific administrative rights and access for each PAT you generate.

  - Generate a PAT:
    - Install **`usethis`** package in RStudio
    - Run the line: **`usethis::create_github_token()`**
    - Click "generate token" and WRITE IT DOWN!
  - Save the PAT in RStudio:
    - Install **`gitcreds`** package in RStudio
    - Run the line: **`gitcreds::gitcreds_set()`**
    - Respond to the prompt by entering your PAT.

Start a New Project in RStudio

Go to Github and in “Repositories” click the green “New Repository” button.

Pick a unique repository name (this will also be a folder name on your computer).

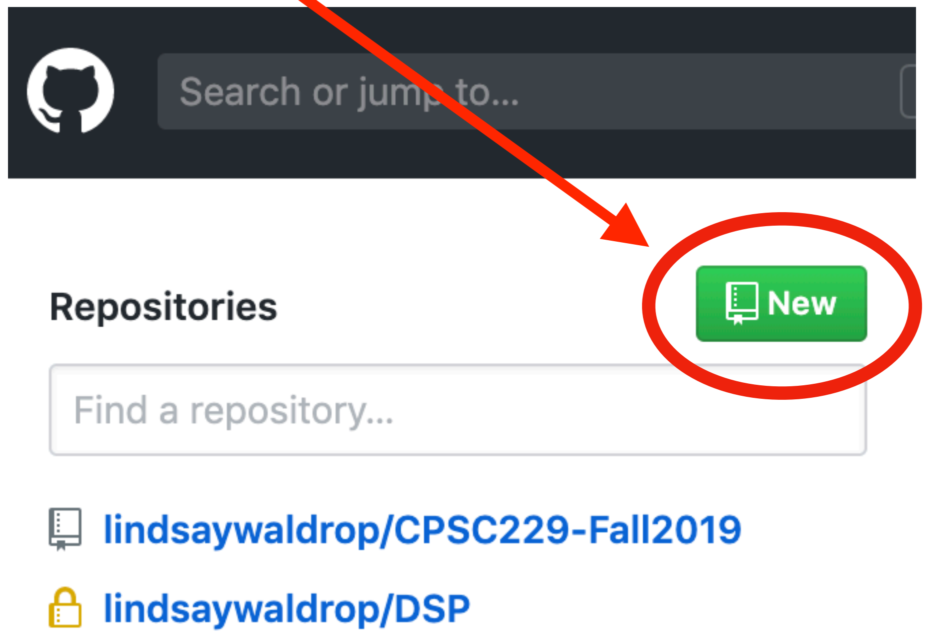
Click “Initialize this repository with a README file” then “Create Repository”

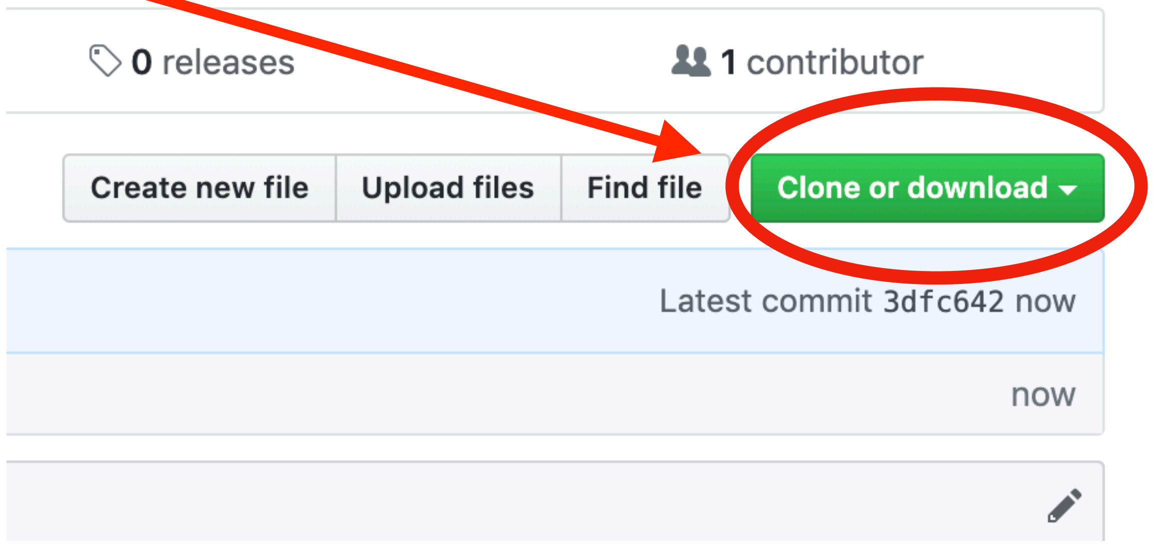
Click the “Clone or download” button and copy the URL to your clipboard.

Return to RStudio and from Files select “New Project…”

Select “Version Control” option and then “git”

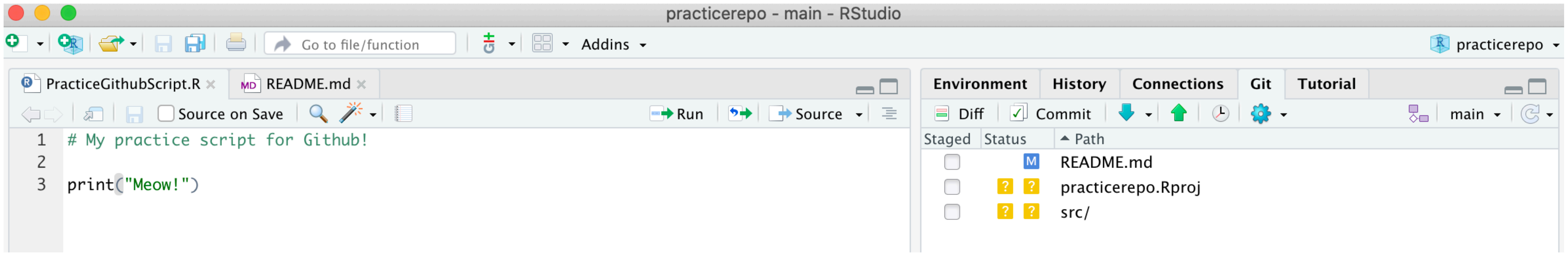
Copy the URL from Github into the repository URL box and then click through to create a new project!





Interacting with Github through RStudio

1. Save a new script and changes to other documents.

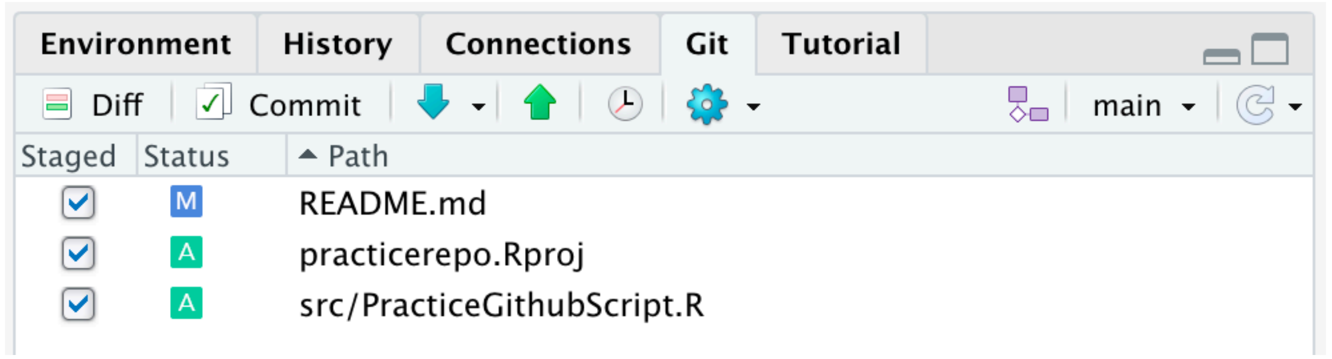


practicerepo - main - RStudio

```
# My practice script for Github!

print("Meow!")
```

Staged	Status	Path
☐	M	README.md
☐	? ?	practicerepo.Rproj
☐	? ?	src/

2. Stage your changes in all documents you want to track.

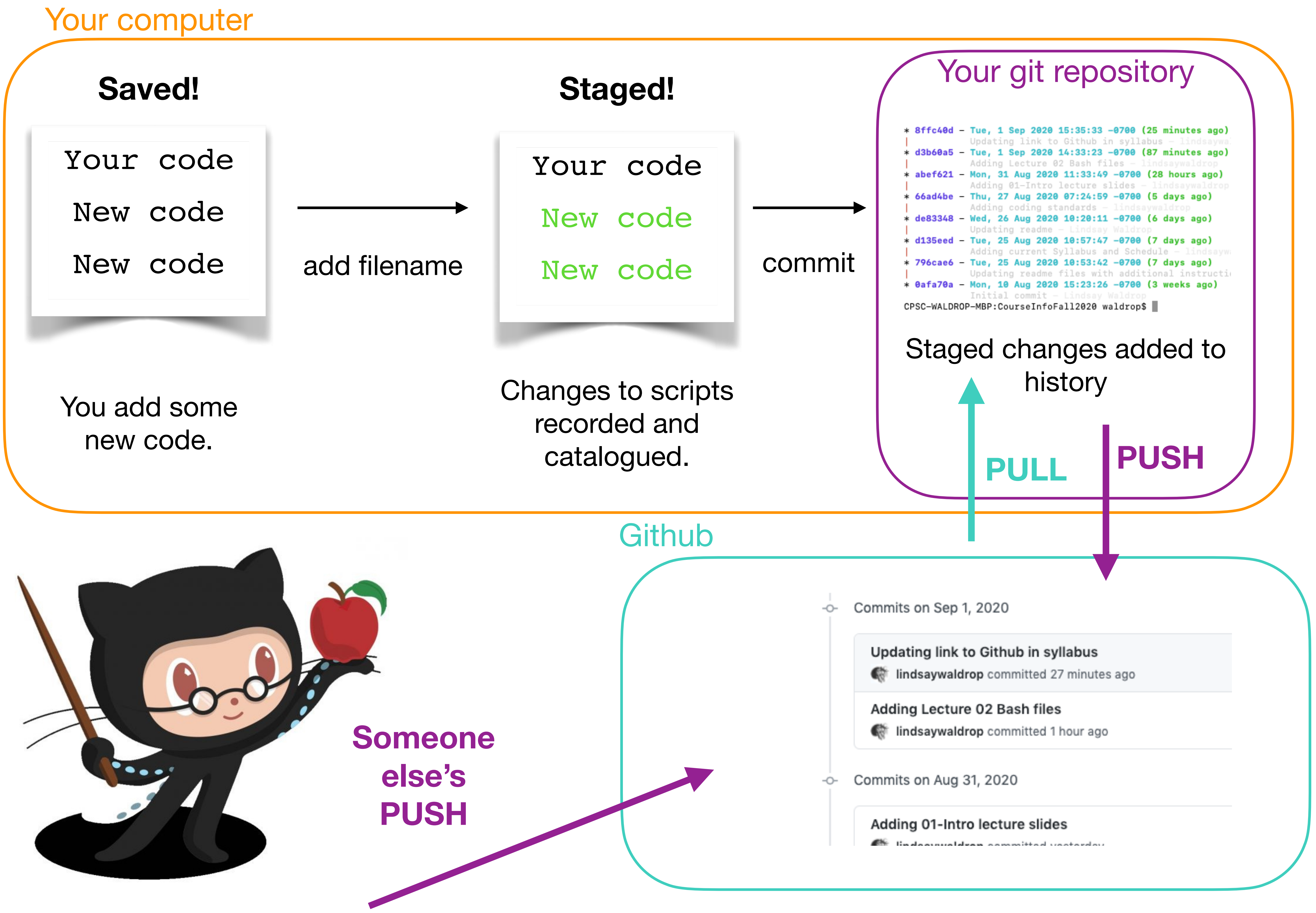


Staged	Status	Path
✓	M	README.md
✓	A	practicerepo.Rproj
✓	A	src/PracticeGithubScript.R

3. Commit your changes!

Repeat these steps one or two more times, adding a bit to the README and your practice script!

Basics of git and Github Online Repositories



```
* 8ffc40d - Tue, 1 Sep 2020 15:35:33 -0700 (25 minutes ago)
|           Updating link to Github in syllabus - lindsaywa
* d3b60a5 - Tue, 1 Sep 2020 14:33:23 -0700 (87 minutes ago)
|           Adding Lecture 02 Bash files - lindsaywaldrop
* abef621 - Mon, 31 Aug 2020 11:33:49 -0700 (28 hours ago)
|           Adding 01-Intro lecture slides - lindsaywaldrop
* 66ad4be - Thu, 27 Aug 2020 07:24:59 -0700 (5 days ago)
|           Adding coding standards - lindsaywaldrop
* de83348 - Wed, 26 Aug 2020 10:20:11 -0700 (6 days ago)
|           Updating readme - Lindsay Waldrop
* d135eed - Tue, 25 Aug 2020 10:57:47 -0700 (7 days ago)
|           Adding current Syllabus and Schedule - lindsayw
* 796cae6 - Tue, 25 Aug 2020 10:53:42 -0700 (7 days ago)
|           Updating readme files with additional instructi
* 0afa70a - Mon, 10 Aug 2020 15:23:26 -0700 (3 weeks ago)
            Initial commit - Lindsay Waldrop
CPSC-WALDROP-MBP:CourseInfoFall2020 waldrop$
```

Commits on Sep 1, 2020

Updating link to Github in syllabus
lindsaywaldrop committed 27 minutes ago

Adding Lecture 02 Bash files
lindsaywaldrop committed 1 hour ago

Commits on Aug 31, 2020

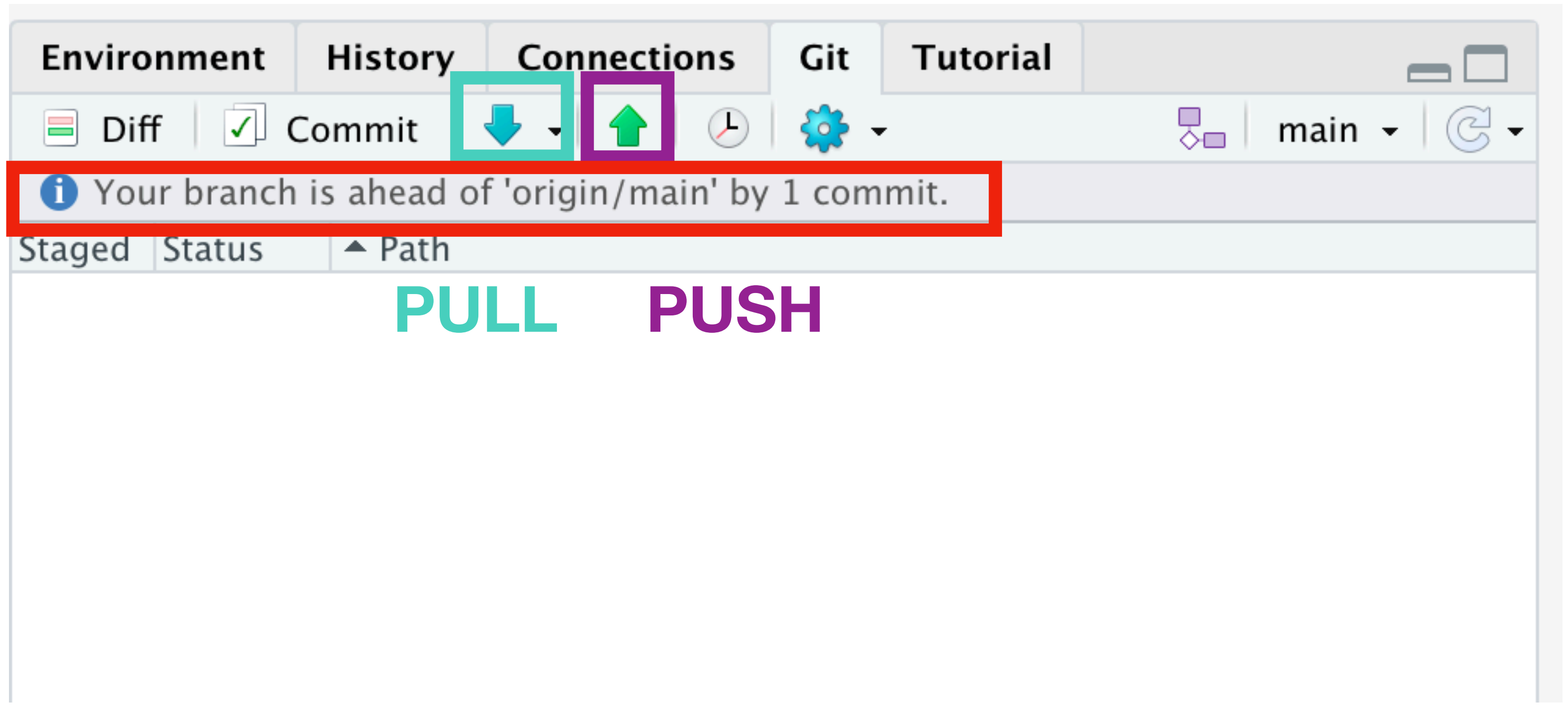
Adding 01-Intro lecture slides

Push Changes to Github through RStudio

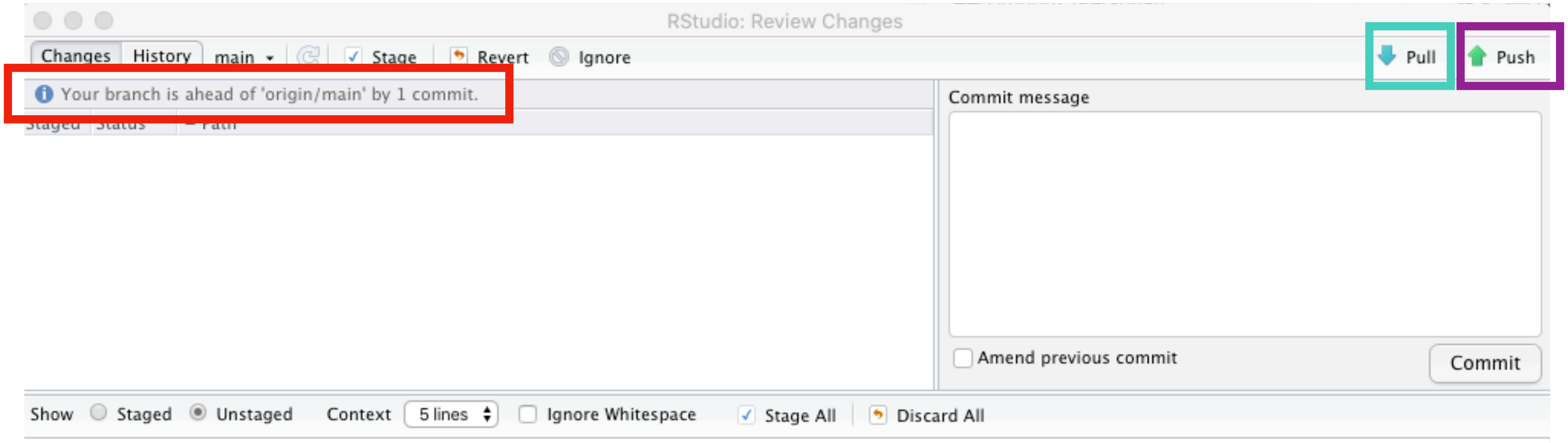
- RStudio will warn you that your branch is not the same as origin/master (Github).

- Look for the push/pull icons in the git window or review changes window!

- Click the Push button to put your local changes onto Github!



Your branch is ahead of 'origin/main' by 1 commit.



Your branch is ahead of 'origin/main' by 1 commit.

Check Your Understanding

Updating your Github repository by PUSHing!

Go to Github and make a change on the website to the README file. Then, go back to RStudio and PULL down that change!

Action Items