

GETTING STARTED WITH GIT AND GITHUB

April 29th, 2024

Objectives

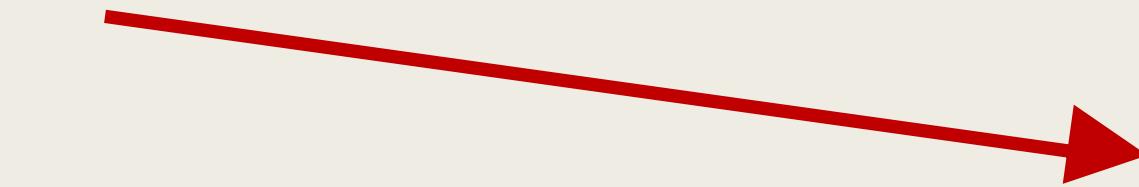
- Learn basics of version control with git
 - *Cycle of work*
 - *Terminology*
 - *Commands*
- Learn basics of github
 - *Remote repositories*
 - *Setting up keys*
 - *Sharing code*
- Level up
 - *Collaborating with forks and pull requests*
 - *Github pages – portfolio for your code*
 - *Conflicts and merging*
 - *Starting over (if all else fails)*

Terminology: *remote* versus *local* repository



[Github.com hosts git repositories](#)

`https://github.com/your-username/your-repo-name`



You clone the remote repo to your computer



Steps to working on a repo: step 0. Clone repo.

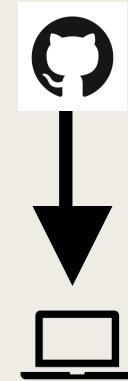
Clone Remote (1st step, done once)

R Studio® → New project → Clone existing repo



Downloads files
and git data

Creates new folder on your computer.
Working tree (your files)



Downloads files
and git data

Creates new folder on your computer.
Repository files are in “File” tab in
RStudio

R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

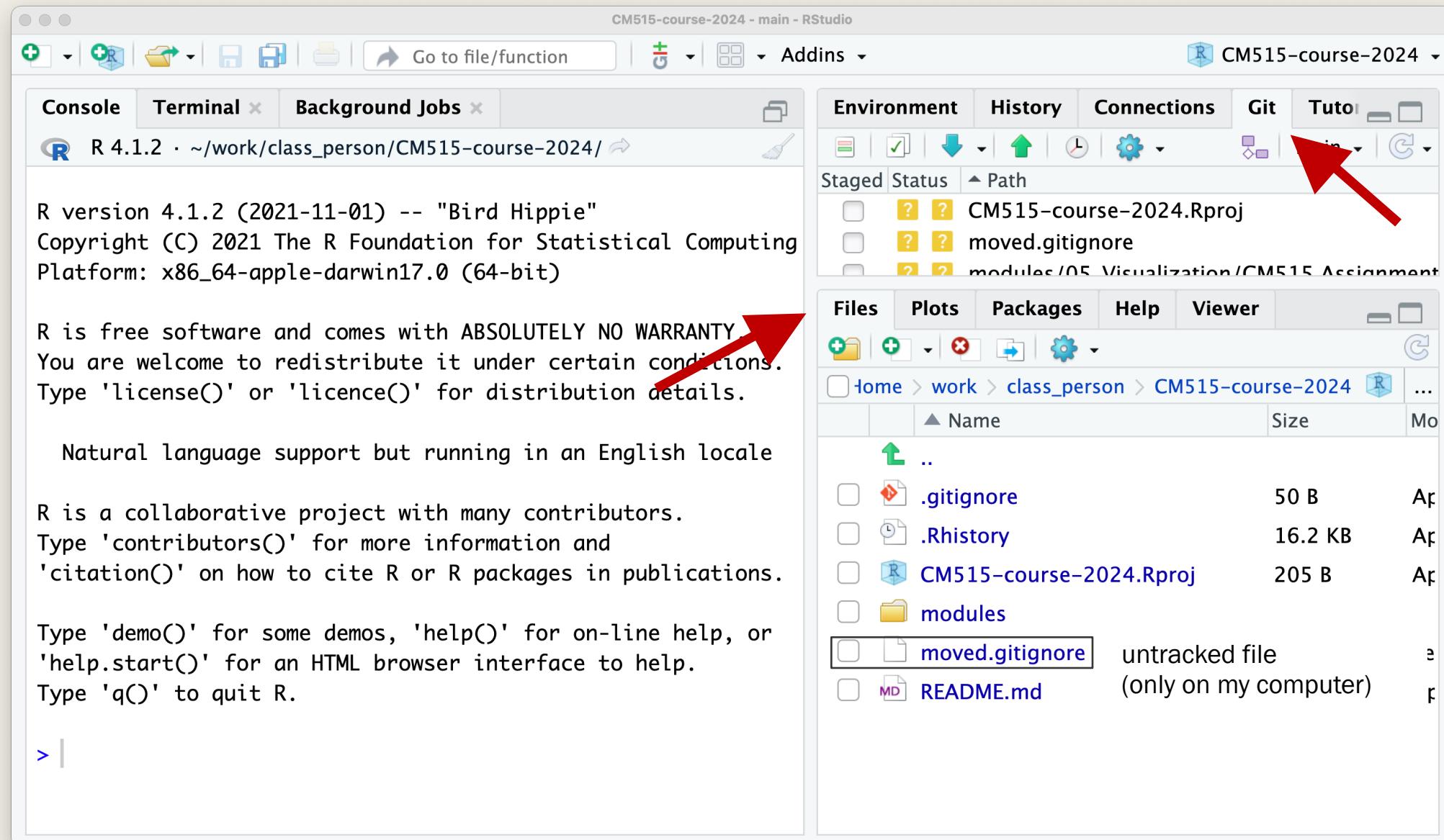
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

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
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

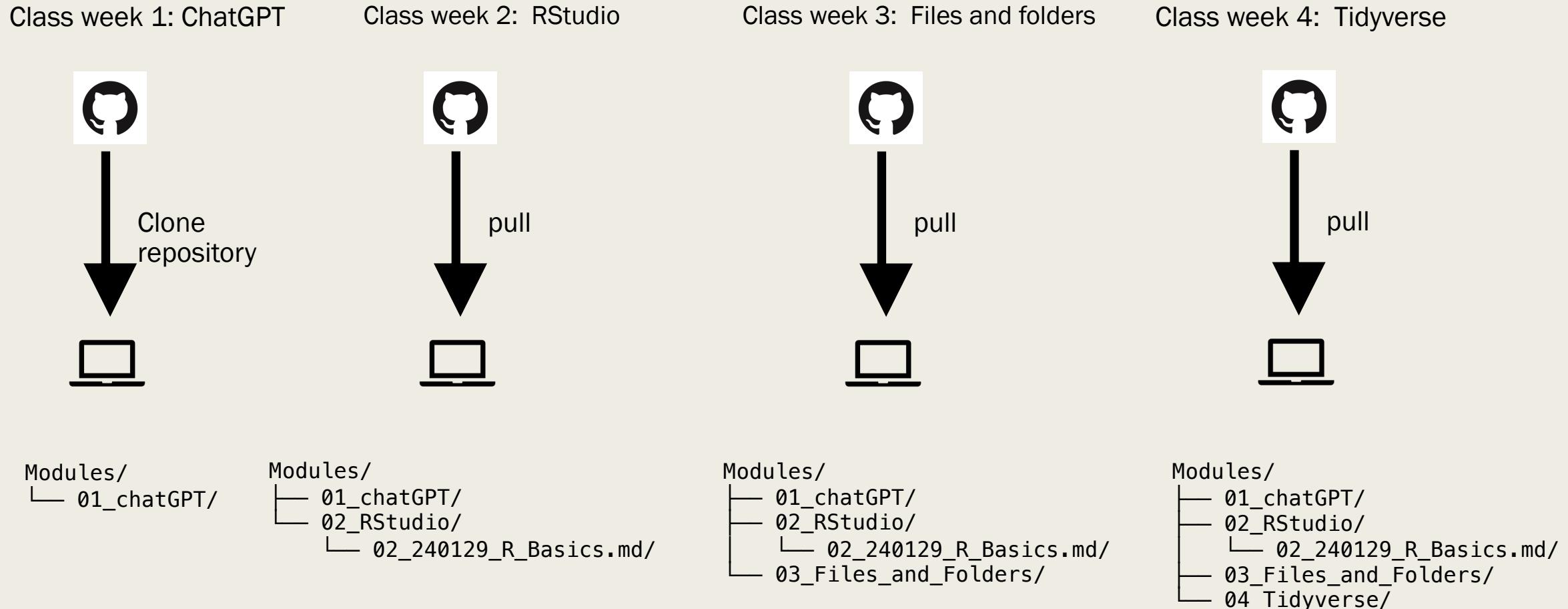


The screenshot shows the RStudio interface with the following details:

- Environment Tab:** Shows the current workspace with objects like CM515-course-2024.Rproj, moved.gitignore, and modules.
- History Tab:** Shows the history of R commands run in the session.
- Connections Tab:** Shows network connections.
- Git Tab:** Active tab, showing the local repository status with files .gitignore, .Rhistory, CM515-course-2024.Rproj, modules, moved.gitignore, and README.md.
- Tutorials Tab:** Shows available tutorials.

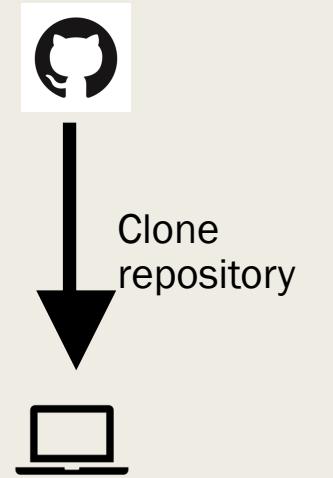
A red arrow points to the **Git** tab in the top navigation bar.

Staying up-to-date with remote



It's OK to add your own files alongside the tracked ones.

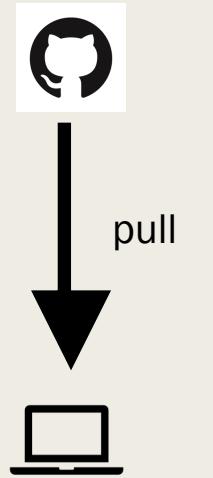
Class week 1: ChatGPT



Modules/
└── 01_chatGPT/

■ - untracked, added by you

Class week 2: RStudio



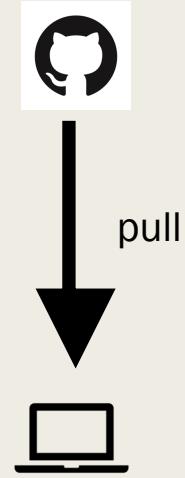
Modules/
└── 01_chatGPT/
└── 02_RStudio/
└── 02_240129_R_Basics.md
└── assignment1.html

Class week 3: Files and folders



Modules/
└── 01_chatGPT/
└── 02_RStudio/
└── 02_240129_R_Basics.md
└── assignment1.html
└── knitted.html
└── 03_Files_and_Folders/
└── assignment2.html

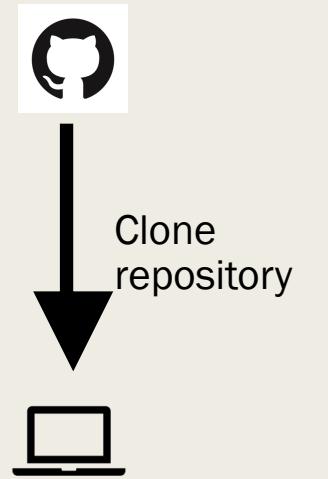
Class week 4: Tidyverse



Modules/
└── 01_chatGPT/
└── 02_RStudio/
└── 02_240129_R_Basics.md
└── assignment1.html
└── knitted.html
└── 03_Files_and_Folders/
└── assignment2.html
└── 04_Tidyverse/
└── assignment3.html

But modifying a tracked file causes a divergence from the repo

Class week 1: ChatGPT



- - untracked , added by you
- - Modified, changed by you

Class week 2: RStudio



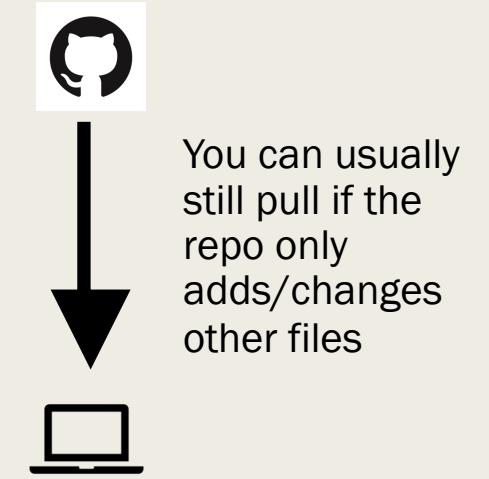
pull

Class week 3: Files and folders



pull

Class week 4: Tidyverse



You can usually still pull if the repo only adds/changes other files

Using Git with RStudio (1)

Exercise: We will use RStudio's “git” tab to track the changes we make to the repository

- **Find the right windows, tabs and files**
 1. Go to the *File tab* and navigate to *modules/02_RStudio*.
 2. Click on *02_240129_R_Basics.Rmd* to open it in the editor.
 3. Click on the “git” tab in the upper right.
- **Make a change to the tracked file**
 1. Modify *02_240129_R_Basics.Rmd* and save it. I changed “let’s” to “let us”.
 2. Notice the “modified” status appear.
- **“Diff”**
 1. Select the file and click “diff” on the left side of the window.
 2. See the difference between your working copy and the repository.

		@@ -1,8 +1,8 @@
1	1	
2	2	
3		# 2. Let's learn about R: it's history and how it biological research
	3	# 2. Let us learn about R: it's history and how it biological research
4	4	
5	5	## References
6	6	
7	7	* [R-intro](https://cran.r-project.org/doc/manuals/intro.pdf) - .pdf documentation. R's manual

Using Git with RStudio (2)

We will use RStudio's "git" tab to track the changes we make to the repository

1. Make another change and save it.
2. I changed "References" on line 5 to "References to read"
3. Now two lines are diff'd.

```
@@ -1,10 +1,10 @@
1 1 
2 2
3 # 2. Let's learn about R: it's history and how it
   biological research
4 3 # 2. Let us learn about R: it's history and how
   biological research
5 4
6 5 ## References
7 5 ## References to read
8 6
9 7 * [R-intro](https://cran.r-project.org/doc/manuals/intro.pdf) - .pdf documentation. R's manual
```

Git/RStudio (3)

We will use RStudio's "git" tab to track the changes we make to the repository

1. I deleted the R-intro reference, line 7.
2. Now the line numbers are different
 1. Repo - still contains line 7.
 2. Working tree- line 7 is now the reference to Brooke's youtube channel.

1	1	
2	2	
3		# 2. Let's learn about R: it's history and how i- research
	3	# 2. Let us learn about R: it's history and how research
4	4	
5		## References
	5	## References to read
6	6	
7		* [R-intro](https://cran.r-project.org/doc/man- .pdf documentation. R's manual
8	7	* [Brooke Anderson's R youtube channel]

Git/RStudio (4)

Reverting your changes, safely

- You can revert to the repository's version of a file
- But it overwrites your changes
- So, use with caution- make a copy first
- Exercise
 - 1. *Make a copy of 02_240129_R_Basics.Rmd by checking the box next to the filename and go to the “gear” icon to copy it.*
 - 2. *Then, in the Git tab, click box by the modified file, click the gear icon and choose “Revert”*
 - 3. *Confirm in the dialog box.*

Creating/working on your own repository

- Create a remote repository
 - Go to your `github.com/accountname`
 - Click on the Repositories tab and find the New button
 - Type ‘**my-resources**’



A screenshot of the GitHub repository creation form. It shows the following fields:

- Owner ***: A dropdown menu showing "meekrob".
- Repository name ***: A text input field containing "example-repo". Below it, a message says "example-repo is available."
- Description (optional)**: An empty text area.
- Visibility**: A radio button group with "Public" selected (indicated by a blue outline) and "Private" unselected (indicated by a grey outline). The "Public" option has a description: "Anyone on the internet can see this repository. You choose who can commit".

Create a new repo (continued)

The screenshot shows the GitHub interface for creating a new repository. A large gray arrow points from the left towards the 'Public' and 'Private' options at the top. Another gray arrow points down from the 'Add .gitignore' section towards the search bar. A third gray arrow points right from the bottom of the 'Add .gitignore' section towards the 'Create repository' button.

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None ▾

.gitignore template
Search: RI

- Processing
- PureScript
- R
- ROS
- Racket
- Rails
- Raku
- RhodesRhomobile
- Ruby

do with your code. [Learn more about licenses](#).

ch. Change the default name in your [settings](#).

in your personal account.

 **Create repository**

The screenshot shows a GitHub repository page for 'meekrob/my-resources'. The repository is public and contains one branch ('main') and two files: '.gitignore' and 'README.md'. The 'Code' tab is selected. The 'About' section indicates no description, website, or topics provided. It also shows 1 watching, 0 stars, and 0 forks. The 'Releases' and 'Packages' sections are both empty.

New repository

meekrob/my-resources

<https://github.com/meekrob/my-resources>

Repos Microsoft Office... Classes At CSU Data management DavidK_Notebook UCSC browser on... Track Hubs Slurm Flags, Part... Home | Modern St... Home | Modern St...

meekrob / my-resources

Type ⌘ to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

my-resources Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file

Code

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

<https://github.com/meekrob/my-resources/blob/main/README.md>

meekrob / my-resources

Code Issues Pull requests Actions Projects Wiki Security

my-resources / README.md

Go to file

meekrob Initial commit b6cd11f · 6 minutes ago History

1 lines (1 loc) · 14 Bytes

Preview Code Blame

Raw

my-resources

<https://github.com/meekrob/my-resources/blob/main/.gitignore>

meekrob / my-resources

Code Issues Pull requests Actions Projects Wiki Security

my-resources / .gitignore

Go to file

meekrob Initial commit b6cd11f · 3 minutes ago

49 lines (35 loc) · 671 Bytes

Code Blame

```
1 # History files
2 .Rhistory
3 .Rapp.history
4
5 # Session Data files
6 .RData
7 .RDataTmp
8
9 # User-specific files
10 .Ruserdata
11
12 # Example code in package build process
13 *-Ex.R
14
15 # Output files from R CMD build
16 /*.tar.gz
17
18 # Output files from R CMD check
19 /*.Rcheck/
20
21 # RStudio files
```

1) Click preview



1 lines (1 loc) · 14 Bytes

Preview Code Blame Raw

my-resources

2) Click to edit

my-resources / README.md in main

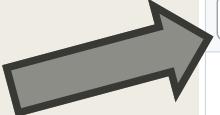
Cancel changes Commit changes...

Edit Preview Spaces 2 Soft wrap

```
1 # my-resources
```



3) Type in here (see next slide)

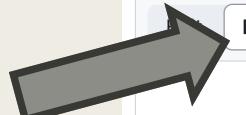


my-resources / README.md in main

Cancel changes Commit changes...

Edit Preview Spaces 2 Soft wrap

```
1 # my-resources
2
3 This is a collection of my computational resources
4
5 ## R
6
7 ### Plotting
8
9 ## Genomics
10
11 ## Python
```



Preview Show Diff

my-resources

This is a collection of my computational resources

R

Plotting

Genomics

Python

A screenshot of a GitHub repository page for 'my-resources'. The repository is owned by 'meekrob' and is public. The 'Code' tab is selected. The repository has 1 branch and 0 tags. The commit history shows two commits:

File	Commit Message	Date
.gitignore	Initial commit	yesterday
README.md	Update README.md	20 hours ago

Annotations:

- A large grey arrow points to the 'Code' tab in the top navigation bar.
- A large grey arrow points to the '2 Commits' link in the commit history.
- A callout box on the right side contains the text: "Notice ‘commits’ and different info For the two files".

Examine your repo's history

The screenshot shows a GitHub repository named 'meekrob / my-resources'. The 'Commits' page is displayed, showing two commits:

- Update README.md**: Committed by meekrob yesterday. It has a green 'Verified' badge, a unique ID of 7254326, and copy/collapse buttons.
- Initial commit**: Committed by meekrob yesterday. It has a green 'Verified' badge, a unique ID of b6cd11f, and copy/collapse buttons.

Arrows from the surrounding text point to specific elements:

- An arrow points to the commit message 'Initial commit'.
- An arrow points to the first six digits of the commit ID 'b6cd11f'.
- An arrow points to the 'Copy full SHA' button next to the commit ID.
- An arrow points to the three-dot menu icon in the top right corner.

The commit message. Click to get **commit details**: the diff of each file change for the commit.

First 6 digits of unique ID for the commit. Click to get the **commit details**.

Copy full SHA (for command line use)

Browse repository at this point

Exercise – clicking on the commit page

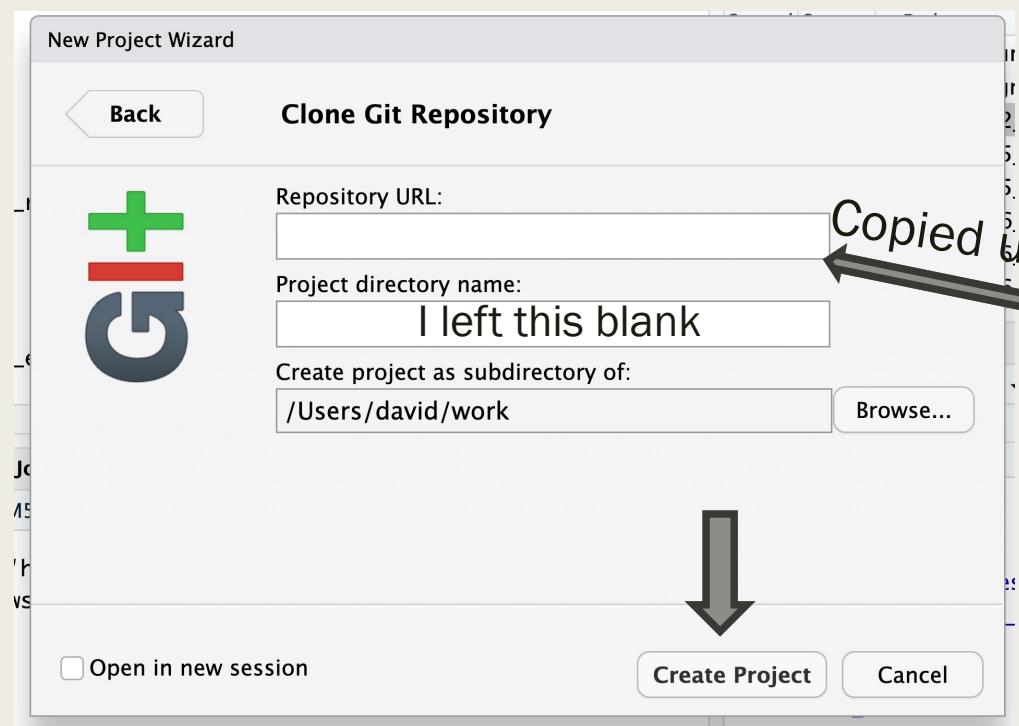
- Click the ID (*view commit details*) for the first commit (all the way at the bottom).
This was the initial commit. What is displayed?
- Click the commit message (also *view commit details*) for the second commit. **We changed the README.** What is displayed?
- ANSWER Initial commit: All diffs show the addition of the files.
- ANSWER Second commit shows the added lines of the README file.

Next steps: pull, edit, push

- Clone your example repo using RStudio
- Make changes, stage, commit, push.
- More involved- Set up SSH keys to authenticate without typing your password each time

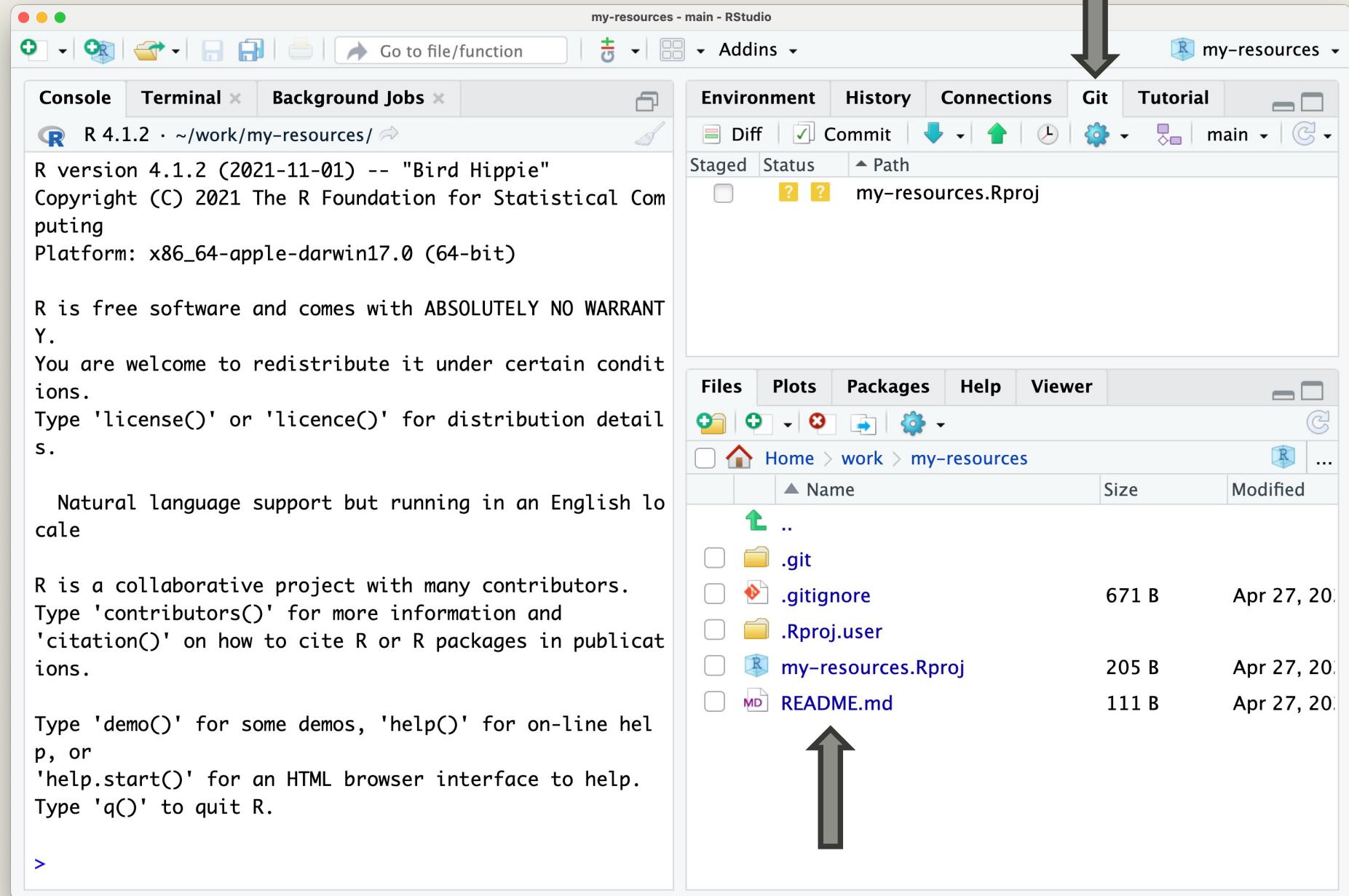
In RStudio

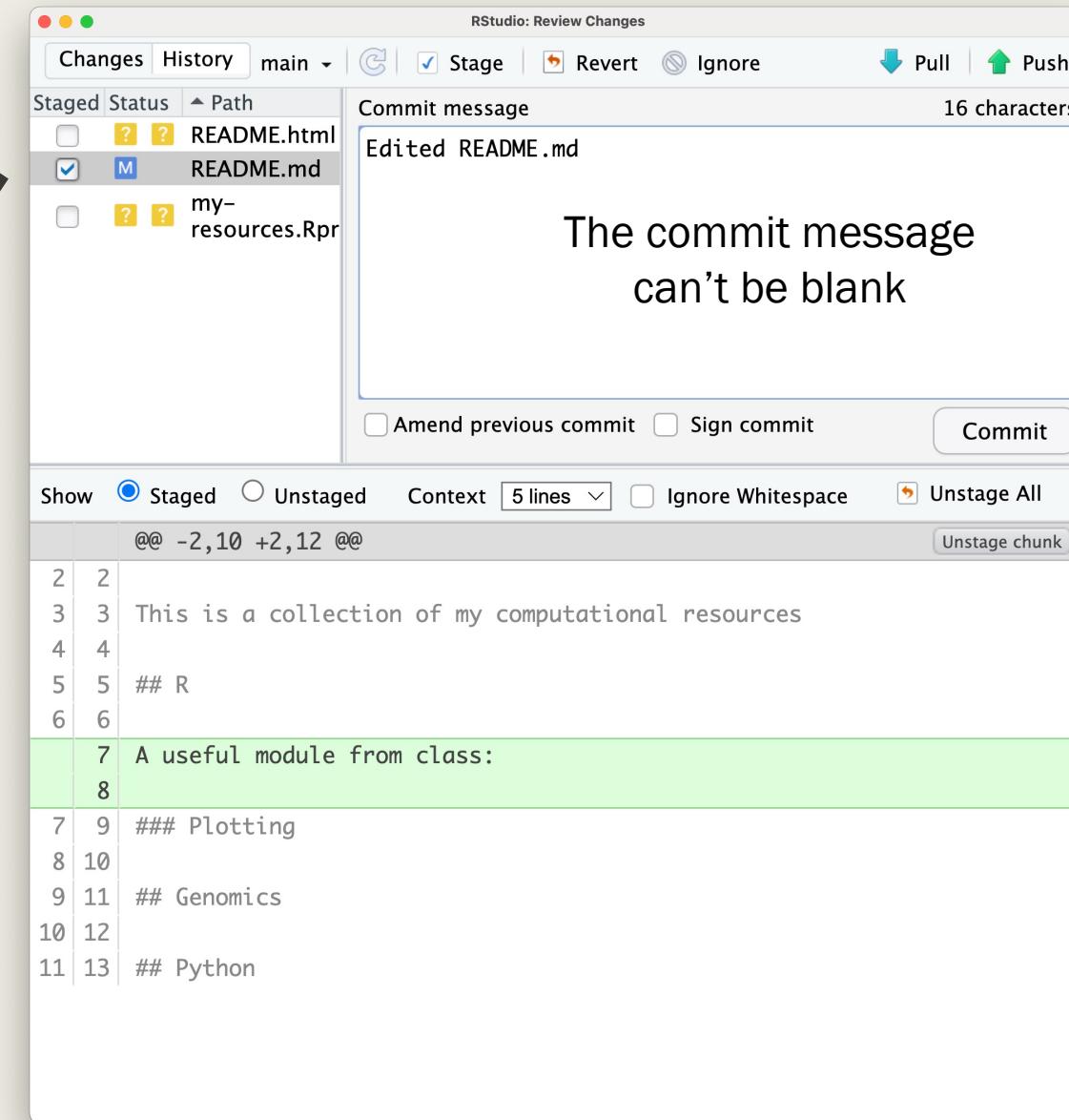
- New Project
- Version Control
- Git

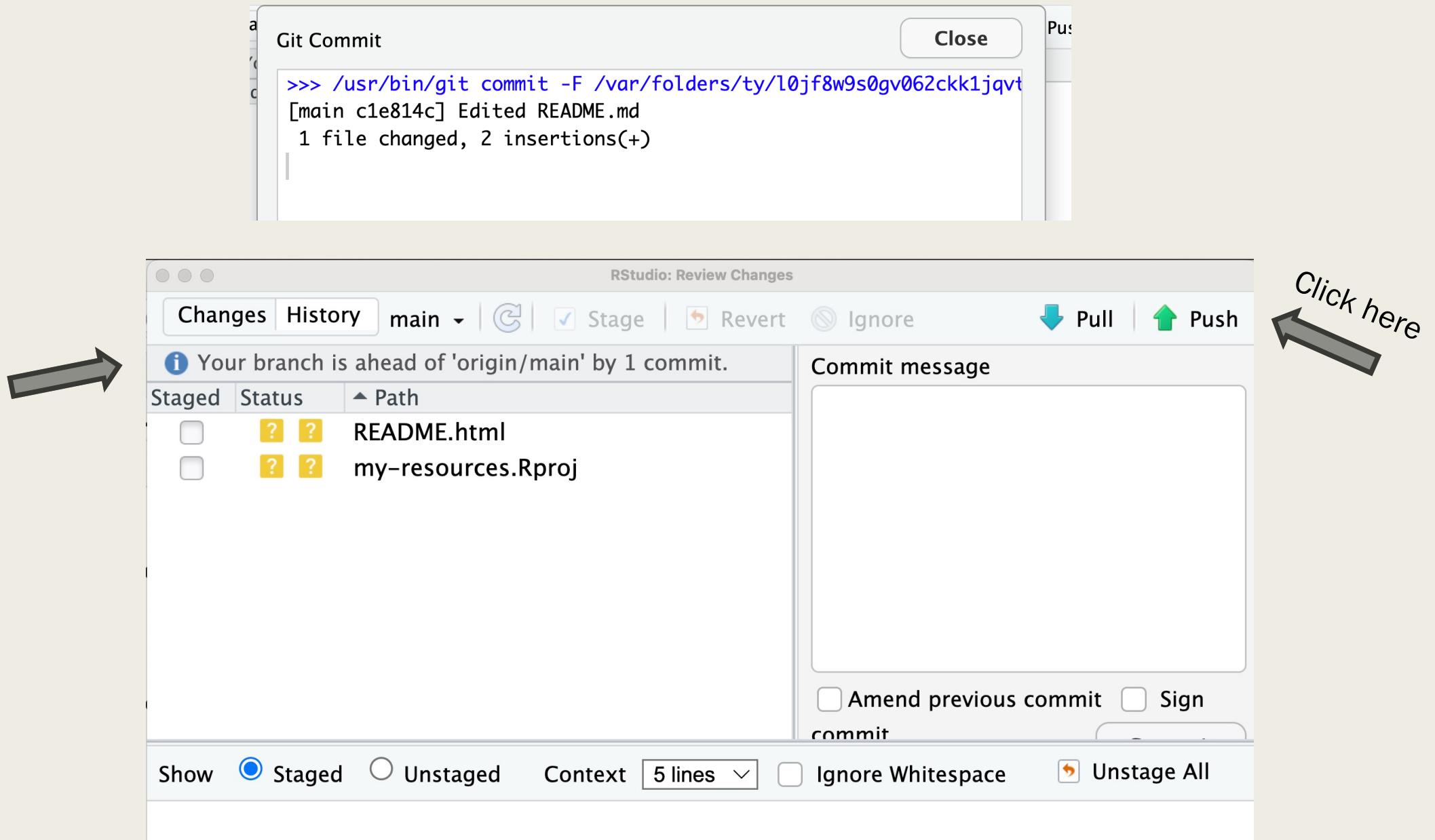


On github.com/yourusername/my-resources

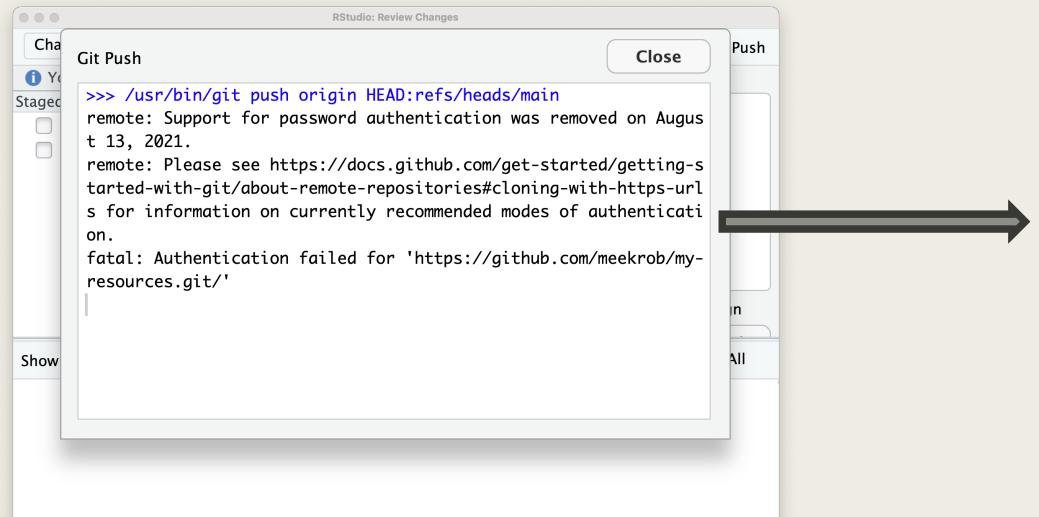
The screenshot shows a GitHub repository page for 'meekrob / my-resources'. A context menu is open over the repository name, showing options like 'Clone', 'HTTPS', 'SSH', and 'GitHub CLI'. The 'Copy url to clipboard' option is highlighted. A large arrow from the 'Copied url' text in the RStudio screenshot points to this menu item.







Error



Cloning with HTTPS URLs [🔗](#)

The `https://` clone URLs are available on all repositories, regardless of visibility. `https://` clone URLs work even if you are behind a firewall or proxy.

When you `git clone`, `git fetch`, `git pull`, or `git push` to a remote repository using HTTPS URLs on the command line, Git will ask for your GitHub username and password. When Git prompts you for your password, enter your personal access token. Alternatively, you can use a credential helper like [Git Credential Manager](#). Password-based authentication for Git has been removed in favor of more secure authentication methods. For more information, see "[Managing your personal access tokens](#)."

If you are accessing an organization that uses SAML SSO and you are using a personal access token (classic), you must also authorize your personal access token to access the organization before you authenticate. For more information, see "[About authentication with SAML single sign-on](#)" and "[Authorizing a personal access token for use with SAML single sign-on](#)."

Tips:

- You can use a credential helper so Git will remember your GitHub credentials every time it talks to GitHub. For more information, see "[Caching your GitHub credentials in Git](#)."
- To clone a repository without authenticating to GitHub on the command line, you can use GitHub Desktop to clone instead. For more information, see "[Cloning a repository from GitHub to GitHub Desktop](#)."

If you'd rather use SSH but cannot connect over port 22, you might be able to use SSH over the HTTPS port. For more information, see "[Using SSH over the HTTPS port](#)."

Cloning with SSH URLs [🔗](#)

SSH URLs provide access to a Git repository via SSH, a secure protocol. To use these URLs, you must generate an SSH keypair on your computer and add the **public** key to your account on GitHub.com. For more information, see "[Connecting to GitHub with SSH](#)."

When you `git clone`, `git fetch`, `git pull`, or `git push` to a remote repository using SSH URLs, you'll be prompted for a password and must provide your SSH key passphrase. For more information, see "[Working with SSH key passphrases](#)."

Using SSH keys

- What are they?
- Generation
- Add them on github. (Should work with all repos on your computer)
- In RStudio
 - Change the remote to the ssh url (<git@github.com:meekrob/my-resources.git>)
 - See if you can push
- Deal with other possible problems

Requirements:

Mac: Terminal (already installed). Go to Finder->Go->Utilities. Scroll down and double click on **Terminal**.

Windows:

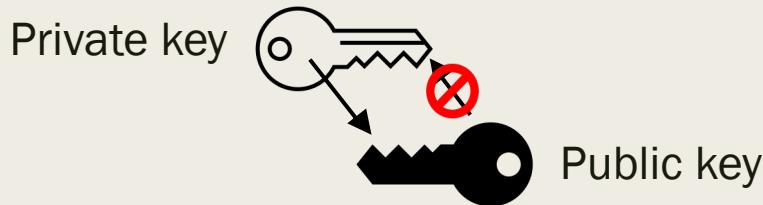
Powershell (already installed)

[Starting windows powershell \(in user and admin mode\)](#)

git-bash (need to install from <https://gitforwindows.org/>)

SSH keys for password-less login

ssh-keygen produces two large cryptographic strings



They are numbers. The public key is computed from the private key.
The private key cannot be mathematically derived from the public key



*Log in securely, then provide server with your public key if you want to authenticate this way.

Exercise for rest of class. Work in groups of your own OS
On the links below, make sure you click the tab for your OS.

Check for existing keys

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/checking-for-existing-ssh-keys>

Generate new keys

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Adding the new key to your account

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>

Testing the connection

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/testing-your-ssh-connection>

Go back to RStudio

In RStudio Terminal tab, do:

```
git remote set-url origin git@github.com:meekrob/my-resources.git
```

Now click the up-arrow to push!!!