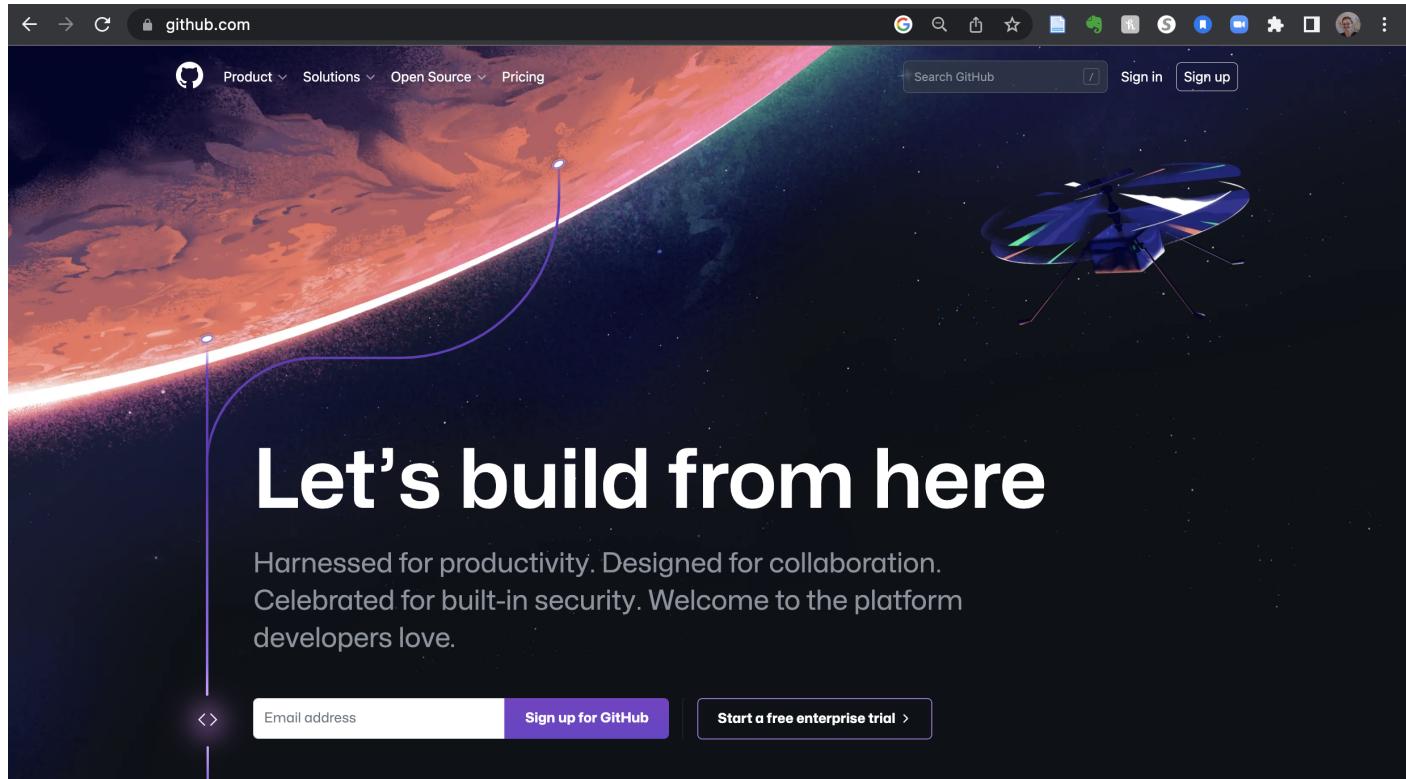


git_tutorial_notes

Getting started with github, git and R-studio.

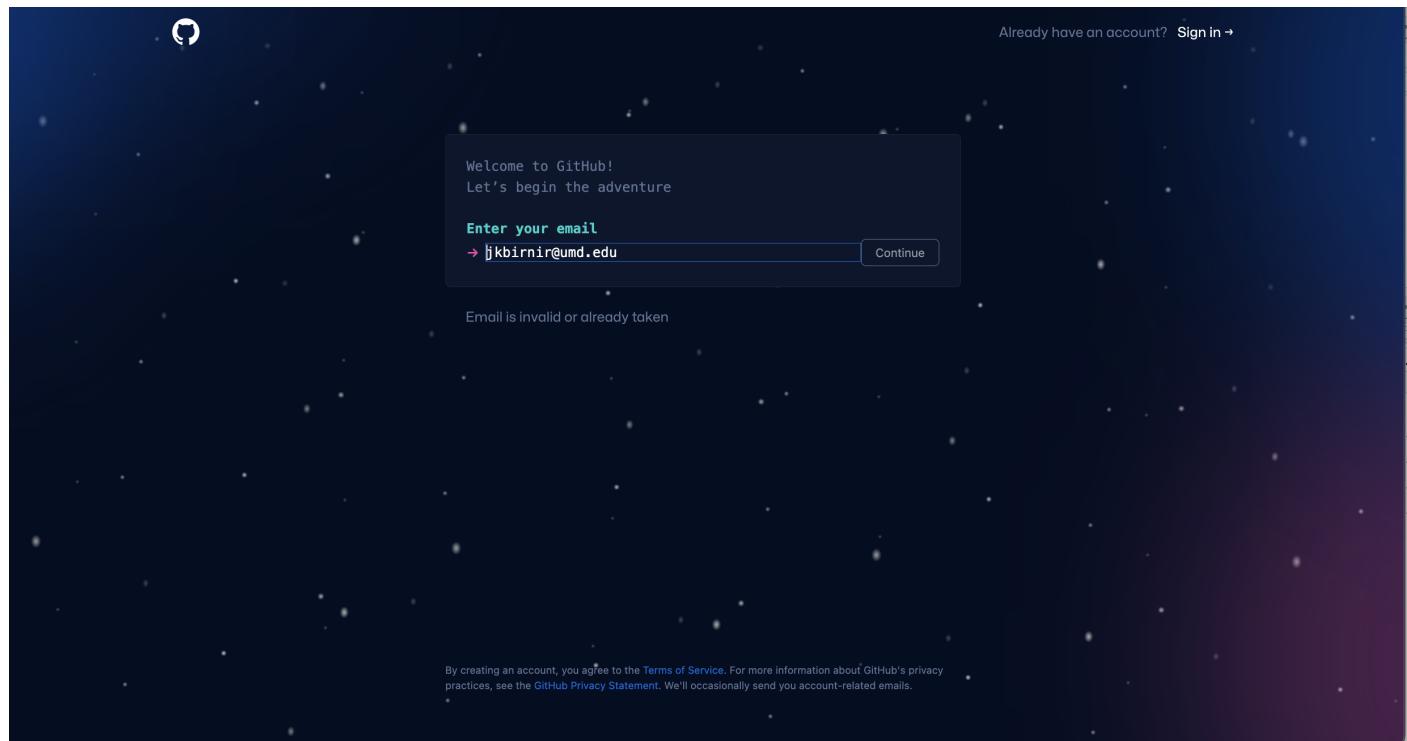
Creating a github account

Go to the github [website](#)



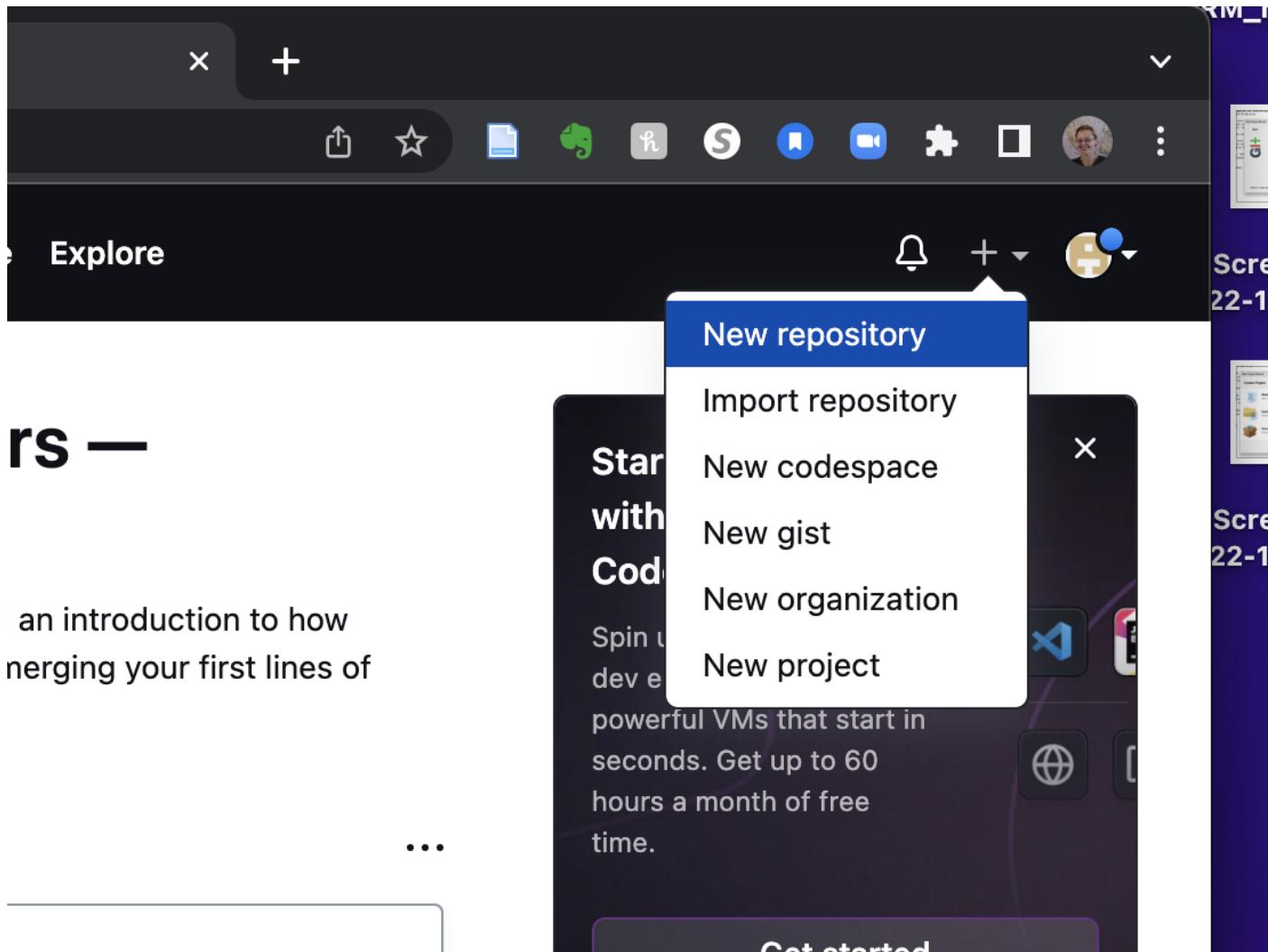
Github website

and signup with your email and password.



Github signup

Once you have a github site, in the upper left hand corner on your github site create a repository for where you want files from your pending project to go. Name the repository whatever you will be calling your project. Select to have a readme file where you can post notes about the project. Select to make it private while you are working on it (this can be changed later).



Github new repository

Once you hit create you new repository should look something like this:

The screenshot shows a GitHub repository page for the user 'jkbirnir' with the repository name 'git_tutorial'. The repository is marked as private. The main area displays a single commit by 'jkbirnir' with the message 'Initial commit' and timestamp '4 minutes ago'. The commit hash is '94ca305'. Below the commit, there is a file named 'README.md' with the content 'git_tutorial'. The sidebar on the right contains sections for 'About', 'Releases', and 'Packages', each of which is currently empty. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings.

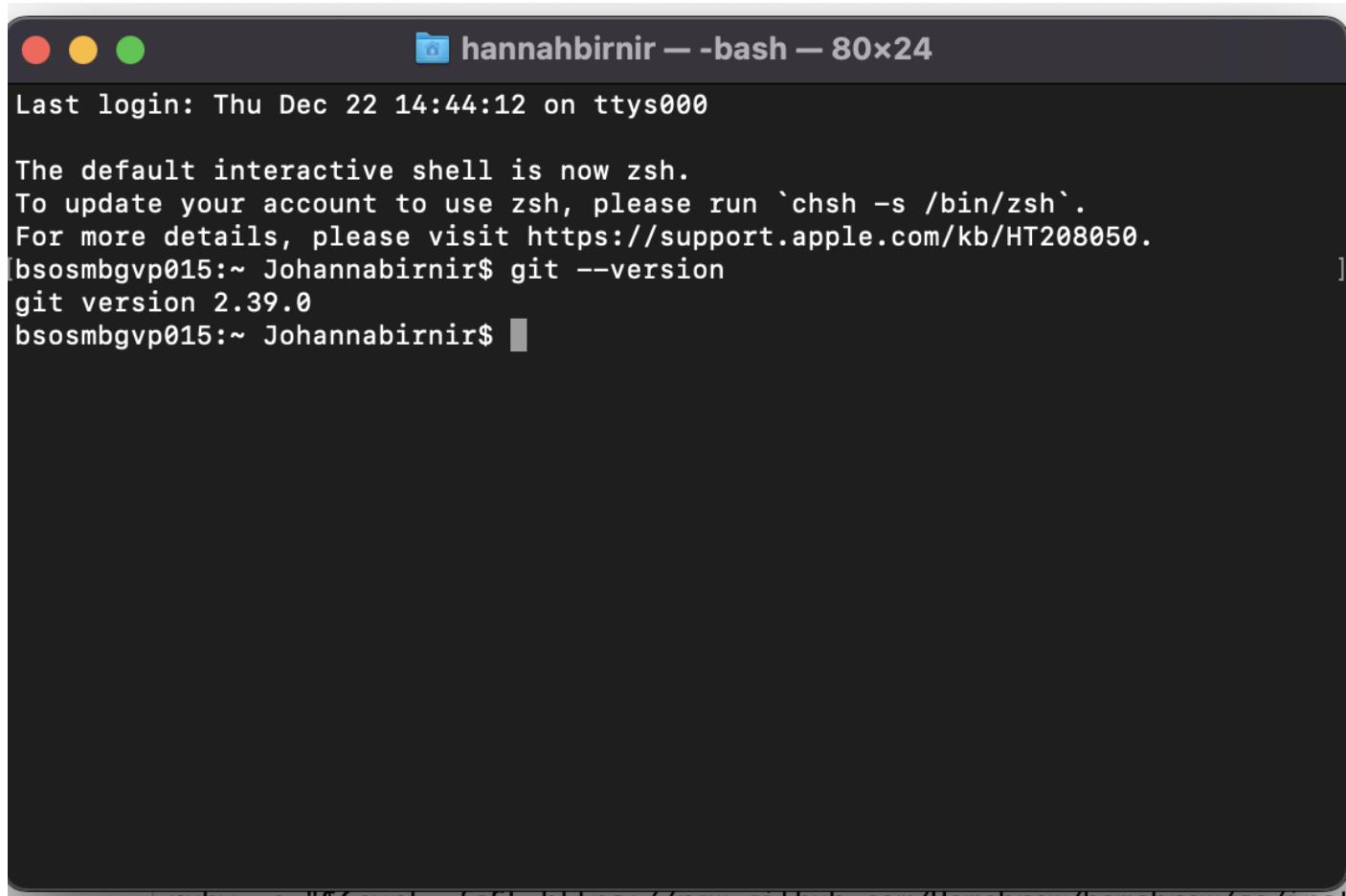
terminal git version

Getting started with git on your computer

Check if you have git installed on your computer. For this you have to use terminal to check. In your terminal write

```
git --version
```

and you should get back your version number.



```
Last login: Thu Dec 22 14:44:12 on ttys000
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[bsosmbgvp015:~ Johannabirnir$ git --version
git version 2.39.0
[bsosmbgvp015:~ Johannabirnir$ ]
```

terminal git version

If you do not have git on your computer you may have to install it. For instructions on how to check and install git on your computer see this very helpful [website](#) or better yet this [manual](#)

Setting up your project in R studio

Now that you have a github account and a local git on your computer you are ready to start working with git through R studio.

Open R-studio and start a new project. Choose a project with version control:

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

Creating a version control project

Select the option to clone a repository from git. What R studio will then do is to clone the repository you created on github locally on your computer.

New Project Wizard

 Back

Create Project from Version Control

**Git**Clone a project from a Git repository **Subversion**Checkout a project from a Subversion repository  Cancel

Cloning the git repository

So that R-studio knows which github local repository to clone you have to specify an external url that matches your username and the name of the new repository that you just created on github (Repository URL).

You also have to specify where your local git files are going to be located see (Create project as subdirectory of:).

New Project Wizard

 Back

Clone Git Repository

Repository URL:

https://github.com/jkbirnir/git_tutorial.git.

Project directory name:

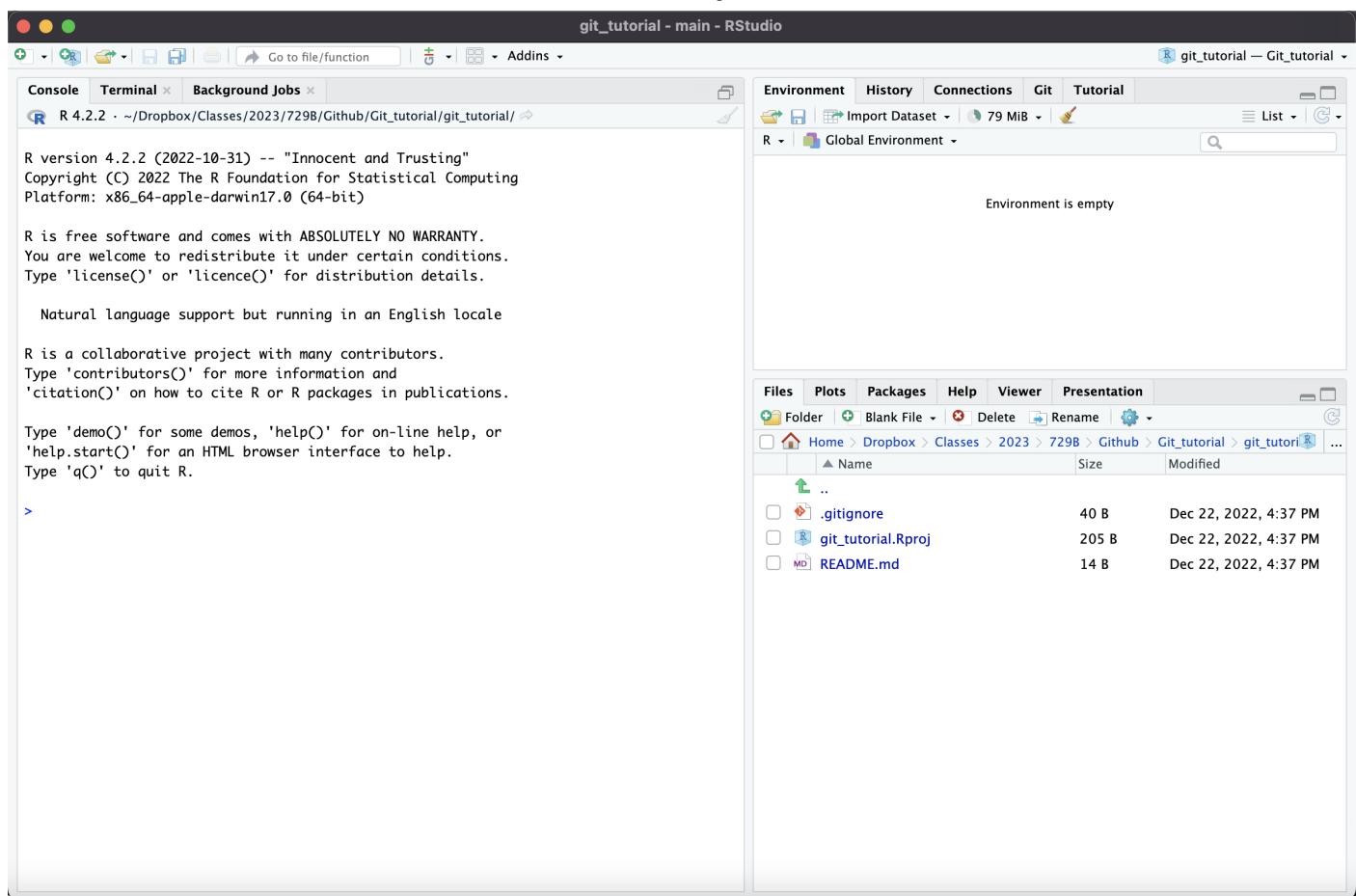
gitTutorial.git.

Create project as subdirectory of:

 Open in new session

Specifying location of external and local repositories

Hit the create button and R studio will create a project site that should look something like this. Notice how the files replicate what is in



local files replicating external repository files

In case you run into trouble at this stage – and are not able to connect your files make sure that your local credentials (signup email matches the email you used to signup with github). To check this you can use:

```
library(usethis)
edit_git_config()
```

- Edit '/Users/hannahbirnir/.gitconfig'

Remember that you have to install the "usethis" package if it is not already on your computer.

If your git and hub have no problems communicating you can set about modifying your local files at will, adding files and changing them.

	Name	Size	Modified
	..		
<input type="checkbox"/>	.gitignore	40 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	gitTutorial.Rproj	205 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	README.md	14 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	figures		
<input type="checkbox"/>	tutorial_notes.qmd	4.5 KB	Dec 22, 2022, 4:24 PM

Modifying local files

Each time you add a new file in your local directory or change it in some way it will appear in your git tab like so:

The screenshot shows the RStudio interface with the 'Git' tab selected. The 'Staged' section displays the following files:

<input type="checkbox"/>	?	?	.DS_Store
<input type="checkbox"/>	?	?	.gitignore
<input type="checkbox"/>	?	?	figures/
<input type="checkbox"/>	?	?	gitTutorial.Rproj
<input type="checkbox"/>	?	?	tutorial_notes.qmd

The screenshot shows the RStudio interface with the 'Files' tab selected. The file list includes:

	Name	Size	Modified
	..		
<input type="checkbox"/>	.gitignore	40 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	gitTutorial.Rproj	205 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	README.md	14 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	figures		
<input type="checkbox"/>	tutorial_notes.qmd	4.5 KB	Dec 22, 2022, 4:24 PM

Local files in Git tab

Notice that because I have not modified the README file that was imported from github this file does not appear in my git tab.

Pushing files

The final step is to commit the changes I have made to my local files to the github repository. For this purpose I have to commit the files I want to update (I only commit the files I wish to update) and then I need to push them to the external repository.

To do this I first select the files that I want to commit. Next I hit the commit button.

The screenshot shows the RStudio interface with the Git tab selected. The top navigation bar includes tabs for Environment, History, Connections, Git, and Tutorial. The Git tab has sub-options for Diff, Commit (which is circled in red), Pull, Push, History, and Settings. Below the tabs is a toolbar with icons for Staged, Status, and Path. The main workspace displays a list of files under the 'Staged' tab, with many files checked (indicated by a green checkmark icon). The list includes: .DS_Store, .gitignore, gitTutorial.Rproj, tutorial_notes.qmd, figures/.DS_Store, figures/1.png, figures/10.png, figures/11.png, figures/12.png, figures/2.png, and figures/3.png. A large red circle highlights the 'Commit' button in the Git tab bar. The bottom section of the interface shows the 'Files' tab selected, displaying a file browser with the following contents:

	Name	Size	Modified
	..		
<input type="checkbox"/>	.gitignore	40 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	gitTutorial.Rproj	205 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	README.md	14 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	figures		
<input type="checkbox"/>	tutorial_notes.qmd	4.5 KB	Dec 22, 2022, 4:24 PM

Local files in Git tab

When I hit the commit button another window pops up where I can write myself notes about the changes I am committing.

The screenshot shows the RStudio interface with the 'Review Changes' tab selected. The left pane lists files in the project directory, with 'tutorial_notes.qmd' highlighted. The right pane contains a 'Commit message' field with the text: 'This is where I can write notes about the changes I am committing.' Below the message is a checkbox for 'Amend previous commit' and a 'Commit' button. At the bottom, there's a code diff view showing the contents of 'tutorial_notes.qmd'. The code includes sections for a title, editor settings, and notes on missing material, along with instructions for creating a GitHub account and signing up.

```

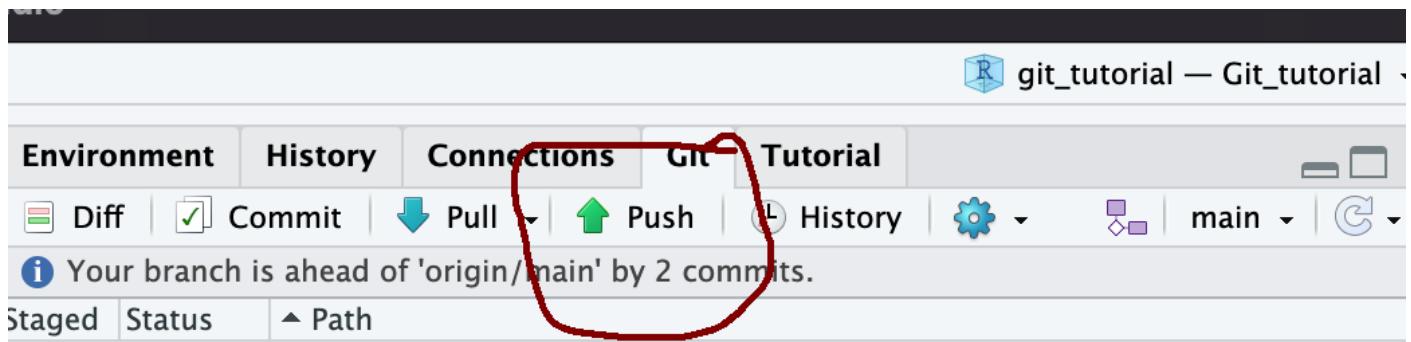
@@ -0,0 +1,108 @@
1 ---
2 title: "git_tutorial_notes"
3 format: html
4 editor: visual
5 ---
6
7 ## Notes on missing material
8
9 The objective of this document is to go through Jordan's tutorial and add notes about where I run into problems or think some additional information would be helpful for the students.
10
11 First, I think that for a comprehensive tutorial it would be helpful to add a few more items at the beginning.
12
13 ### Creating a github account
14
15 Go to the github [website](http://github.com)
16
17 ![[Github website]](figures/1.PNG)
18
19 and signup with your email and password.
20
21 ![[Github signup]](figures/2.PNG)
22

```

Commit notes

Once I hit commit R-studio knows which local files I want to change in my external depository. Note that I can work locally and commit many files and then work on other files and commit them later. **Commit only commits changes to my files locally.** The last step then is to push all my locally committed files to my external repository.

From within my project I simply push the push button in my now empty git tab and my external repository is updated.



A screenshot of the RStudio file browser. The top menu bar includes Files, Plots, Packages, Help, Viewer, and Presentation. Below the menu, there are buttons for Folder, Blank File, Delete, Rename, and settings. The path is displayed as Home > Dropbox > Classes > 2023 > 729B > Github > GitTutorial > gitTutorial. The file list table has columns for Name, Size, and Modified. The files listed are .gitignore, gitTutorial.Rproj, README.md, figures, and tutorial_notes.qmd.

	Name	Size	Modified
	..		
<input type="checkbox"/>	.gitignore	40 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	gitTutorial.Rproj	205 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	README.md	14 B	Dec 22, 2022, 4:37 PM
<input type="checkbox"/>	figures		
<input type="checkbox"/>	tutorial_notes.qmd	4.5 KB	Dec 22, 2022, 4:24 PM

Pushing changes to github

The screenshot shows a GitHub repository page for 'git_tutorial'. At the top, it displays 'main' branch, '1 branch', and '0 tags'. Below this is a commit history table:

Author	File	Message	Time Ago
jkbirnir	figures	message	3 minutes ago
	.DS_Store	This is where I can write notes about the changes I am committing.	9 minutes ago
	.gitignore	This is where I can write notes about the changes I am committing.	9 minutes ago
	README.md	Initial commit	1 hour ago
	git_tutorial.Rproj	This is where I can write notes about the changes I am committing.	9 minutes ago
	tutorial_notes.qmd	This is where I can write notes about the changes I am committing.	9 minutes ago

Below the commit history is the 'README.md' file content:

```
git_tutorial
```

About
No description, website, or topics provided.

[Readme](#)
[0 stars](#)
[1 watching](#)
[0 forks](#)

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Updated external repository

Github then tracks all of the changes in each file and new files added in each commit while also updating the main file.

This is very nice if you are working on a project - like a website that you might want to change frequently. You can simply open the project - make a change to any one part of the project and commit those changes to github. Imagine, for example, a website where you post new data and other information as it becomes available.

Exercise

Install git on your computer

Create a github account

Create a version control project in R studio and upload it to Github

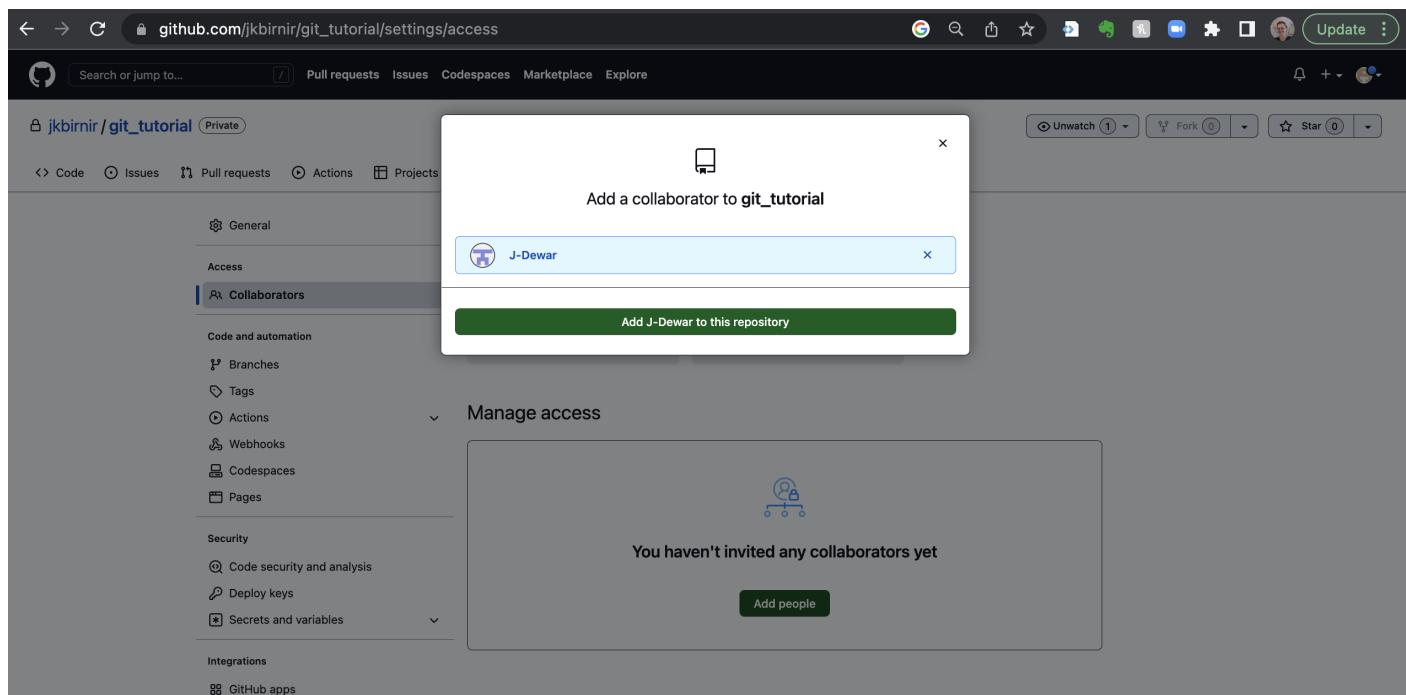
Working together in git.

One of the most useful things about github is the ability to work collaboratively.

Associating your RStudio with a collaborative project

First, in order to work collaboratively, you may need to associate your RStudio with a project in GitHub that you did not create. If you created the project, do the following to add collaborators:

Go on the Github website to Settings > Manage Access > Invite a collaborator.



Inviting Github Collaborations

Your teammate should accept the invite in their email.

Once this is complete, you can use the steps above to associate your RStudio with the GitHub project.

Pulling Changes

One important aspect of collaboration in GitHub is the ability to pull changes. This allows you to update your code to align with changes pushed by collaborators.

Using the down arrow button, RStudio goes to the GitHub repo, grabs the most recent code and brings it into your local editor. (Pulling regularly is extremely important if you're collaborating, though if you're the only one working on an RStudio project and associated GitHub repo, you know your local code matches what's on GitHub so it's less important.)

To pull, click the blue down arrow on your Git tab to see if you have changes to pull. If collaborating, you might run into merge conflicts.

When you pull your project updates to show the changes your collaborator has made to the project. Look at the dates.

The screenshot shows the RStudio interface with two main panes: the Git pane at the top and the Files pane below it.

Git Pane:

- Header: Environment, History, Connections, Git, Tutorial.
- Buttons: Diff, Commit, Pull, Push, Refresh, Gear, Branches, main.
- States: Staged, Status, Path.

Files Pane:

- Header: Files, Plots, Packages, Help, Viewer, Presentation.
- Toolbar: Folder, Blank File, Delete, Rename, Gear.
- Path: Home > Dropbox > Classes > 2023 > 729B > Github > Git_tutorial > git_tutorial.
- Table:

	Name	Size	Modified
	..		
<input type="checkbox"/>	.gitignore	69 B	Jan 14, 2023, 10:41 AM
<input type="checkbox"/>	.RData	2.5 KB	Dec 27, 2022, 11:15 AM
<input type="checkbox"/>	.Rhistory	35 B	Jan 16, 2023, 2:03 PM
<input type="checkbox"/>	figures		
<input type="checkbox"/>	git_tutorial_notes1_files		
<input type="checkbox"/>	git_tutorial_notes1.html	24.4 KB	Jan 14, 2023, 12:27 PM
<input type="checkbox"/>	git_tutorial_notes1.pdf	3.7 MB	Jan 14, 2023, 12:28 PM
<input type="checkbox"/>	git_tutorial_notes1.qmd	8 KB	Jan 14, 2023, 12:28 PM
<input type="checkbox"/>	git_tutorial.Rproj	205 B	Jan 24, 2023, 12:12 PM
<input type="checkbox"/>	README.md	89 B	Jan 14, 2023, 11:41 AM
<input type="checkbox"/>	tutorial_notes.qmd	7.1 KB	Dec 22, 2022, 6:01 PM

and

The screenshot shows the RStudio interface with a 'git_tutorial - main - RStudio' tab at the top. On the left, a 'Git Pull' dialog box displays the command run in the terminal:

```
>>> /usr/bin/git pull
From https://github.com/jkbirnir/git_tutorial
  bb3f293..1df0ac3  main      -> origin/main
Updating bb3f293..1df0ac3
Fast-forward
  .DS_Store           | Bin 6148 -> 6148 bytes
  figures/.DS_Store   | Bin 6148 -> 6148 bytes
  figures/git branch picture.png | Bin 0 -> 35722 bytes
  git_tutorial_notes1.qmd    |  8 +++++++-
  4 files changed, 7 insertions(+), 1 deletion(-)
  create mode 100644 figures/git branch picture.png
```

Below the dialog, the terminal output continues:

```
ng in
ny co
matio
ackages in publications.

)' for on-line help, or
nterface to help.

sses/2023/729B/Github/Git_tutorial/git_tutorial/.RData]
```

On the right, a file browser window titled 'git_tutorial — Git_tutorial' shows the directory structure:

Name	Size	Modified
..		
.gitignore	69 B	Jan 14, 2023, 10:41 AM
.RData	2.5 KB	Dec 27, 2022, 11:15 AM
.Rhistory	35 B	Jan 16, 2023, 2:03 PM
figures		
git_tutorial_notes1_files		
git_tutorial_notes1.html	24.4 KB	Jan 14, 2023, 12:27 PM
git_tutorial_notes1.pdf	3.7 MB	Jan 14, 2023, 12:28 PM
git_tutorial_notes1.qmd	8.1 KB	Jan 24, 2023, 12:13 PM
git_tutorial.Rproj	205 B	Jan 24, 2023, 12:12 PM
README.md	89 B	Jan 14, 2023, 11:41 AM
tutorial_notes.qmd	7.1 KB	Dec 22, 2022, 6:01 PM

You can also track the changes on github if you want more details.

Go to your github project:

The GitHub repository page for `jkbirnir/git_tutorial` is shown. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings.

The repository summary shows:

- Branch: main
- Branches: 1 branch
- Tags: 0 tags
- Last commit: 1df0ac3 12 minutes ago
- Commits: 10 commits

The commit history table lists the following commits:

Commit	Description	Time
J-Dewar test	Edits to the github tutorial to include branches	17 hours ago
figures	Most recent changes Jan 14.	last week
git_tutorial_notes1_files/libs	Edits to the github tutorial to include branches	17 hours ago
.DS_Store	I'm committing changes January 14	last week
.gitignore	checking order of commit and push	last week
README.md	This is where I can write notes about the changes I am committing.	last month
git_tutorial.Rproj	Most recent changes Jan 14.	last week
git_tutorial_notes1.html	Most recent changes Jan 14.	last week
git_tutorial_notes1.pdf	test	12 minutes ago
git_tutorial_notes1.qmd	I'm committing changes January 14	last week
tutorial_notes.qmd		

The README.md file content is:

```
#Adding some new notes January 14. #Checking the order of commit and push
```

On the right side of the page, there are sections for About, Releases, Packages, and Contributors.

There you see who are the collaborators and when each item was updated. For even more information

click on any of the files (here the qmd file)

git_tutorial / git_tutorial (Private)

Code Issues Pull requests Actions Projects Security Insights Settings

git_tutorial / git_tutorial_notes1.qmd

J-Dewar test

At 2 contributors

191 lines (105 sloc) | 8.14 KB

```

1 ----
2 title: "git_tutorial_notes"
3 format: html
4 editor: visual
5 ----
6
7 # Getting started with github, git and R-studio.
8
9 ### Creating a github account
10
11 Go to the github [website](http://github.com)
12
13 ![Github website](figures/1.PNG)
14
15 and signup with your email and password.
16
17 !![Github signup](figures/2.PNG)
18
19 Once you have a github site, in the upper left hand corner on your github site create a repository for where you want files from your pending project to go. Nam

```

Raw Blame

Here you see the history of the development of the project and if you want to see who made what changes when push the blame button.

git_tutorial / git_tutorial (Private)

Code Issues Pull requests Actions Projects Security Insights Settings

git_tutorial / git_tutorial_notes1.qmd

I'm committing changes January 14 last week

test 13 minutes ago

I'm committing changes January 14 last week

100644 | 191 lines (105 sloc) | 8.14 KB

Newer Older

			Raw Normal view History

```

1 ----
2 title: "git_tutorial_notes"
3 format: html
4 editor: visual
5 ----
6
7 # Getting started with github, git and R-studio.
8
9 ### Creating a github account
10
11 Go to the github [website](http://github.com)
12
13 ![Github website](figures/1.PNG)
14
15 and signup with your email and password.
16
17 !![Github signup](figures/2.PNG)
18

```

The blame allows you to blame whoever - mostly yourself ;) is responsible for making changes to your project.

If you want more of an overview - then push the history button and you get a summary of changes:

History for [git_tutorial / git_tutorial_notes1.qmd](#)

- o Commits on Jan 24, 2023
 - [test](#)
J-Dewar committed 13 minutes ago
- o Commits on Jan 23, 2023
 - [Edits to tutorial notes](#)
J-Dewar committed 17 hours ago
 - [Edits to the github tutorial to include branches](#)
J-Dewar committed 17 hours ago
- o Commits on Jan 14, 2023
 - [Most recent changes Jan 14.](#)
jkbirnir committed last week
 - [I'm committing changes January 14](#)
jkbirnir committed last week
- o End of commit history for this file

[Newer](#) [Older](#)

In sum this is the workflow when you and your collaborator are both working on the main project and either one of you can make changes to the project.

When someone invites you to a project and to work on the main as here:

You now have push access to the J-Dewar/729B repository. X

[J-Dewar / 729B](#) Public

[Watch \(1\)](#) [Fork \(0\)](#) [Star \(0\)](#)

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

[master](#) [branch](#) [tags](#)

		Last Commit	Commits
Quarto and Github Module.qmd	Edited to include RStudio-centric information	705c369 on Dec 19, 2022	3
Quarto-and-Github-Module.pdf	Edited to include RStudio-centric information	last month	
README.md	Here is the module for Github and Quarto	2 months ago	
	A commit from my local computer	2 months ago	

[README.md](#)

This is the repository for the GVPT class 729B

About

No description, website, or topics provided.

[Readme](#) [0 stars](#) [1 watching](#) [0 forks](#)

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

You can open the project locally as you would any other version control project (see earlier steps in the tutorial)

This screenshot shows the RStudio interface. On the left, a Quarto document titled "Quarto and Github Module.qmd" is open, containing the following code:

```
---
title: "GitHub"
format: pdf
editor: visual
---
```

The text "This is an old tutorial that we are checking is you can work on from another project" is displayed below the code.

In the top right, the "Git" tab is selected in the toolbar. A dropdown menu shows options like "R Console...", "Getting St...", "Pushing C...", etc.

On the right, a file browser window titled "729B - master - RStudio" shows the directory structure:

Name	Size	Modified
..		
.gitignore	40 B	Jan 24, 2023, 12:56 PM
729B.Rproj	205 B	Jan 24, 2023, 12:56 PM
Quarto and Github Module.qmd	5.4 KB	Jan 24, 2023, 12:57 PM
Quarto-and-Github-Module.pdf	413.6 KB	Jan 24, 2023, 12:56 PM
README.md	47 B	Jan 24, 2023, 12:56 PM

Once you are done changing your files locally - you then go through the same steps of committing and pushing. Which results in this message telling you the process was successful.

This screenshot shows the RStudio interface after a git push operation. A modal dialog titled "Git Push" displays the command run and its output:

```
>>> /usr/bin/git push origin HEAD:refs/heads/master
To https://github.com/J-Dewar/729B
 30181c4..62151f4  HEAD -> master
```

The file browser on the right shows the updated directory structure:

Name	Size	Modified
..		
.gitignore	40 B	Jan 24, 2023, 12:56 PM
729B.Rproj	205 B	Jan 24, 2023, 12:56 PM
Quarto and Github Module.qmd	5.5 KB	Jan 24, 2023, 1:06 PM
Quarto-and-Github-Module.pdf	413.6 KB	Jan 24, 2023, 12:56 PM
README.md	47 B	Jan 24, 2023, 12:56 PM

and your changes will show up in the remote directory on github.

The screenshot shows a GitHub repository page for 'git_tutorial_notes'. The 'Code' tab is selected. The master branch has 1 branch and 0 tags. There are 5 commits from 'jkbirnir' made some change. The commits are:

- .gitignore: Jordan and I are checking if you need git desktop (9 minutes ago)
- 729B.Rproj: Jordan and I are checking if you need git desktop (9 minutes ago)
- Quarto and Github Module.qmd: made some change (now)
- Quarto-and-Github-Module.pdf: Here is the module for Github and Quarto (2 months ago)
- README.md: A commit from my local computer (2 months ago)

The README.md file contains:

This is the repository for the GVPT class 729B

About
No description, website, or topics provided.

Readme

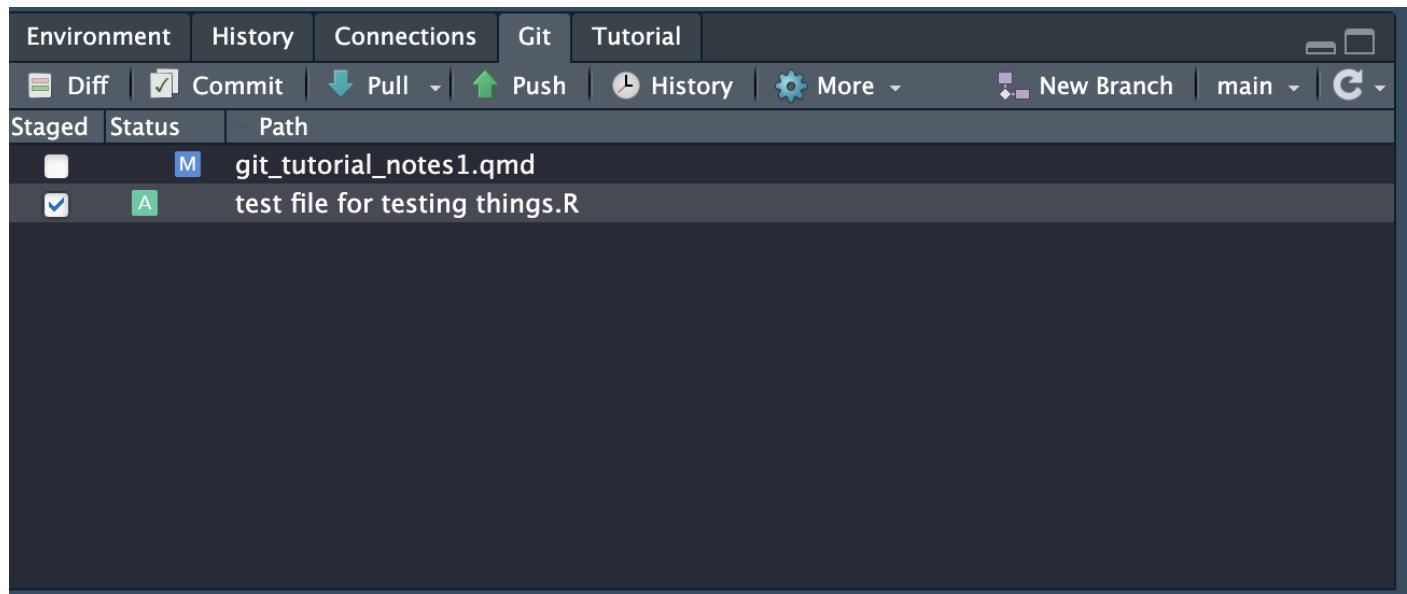
Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Branches

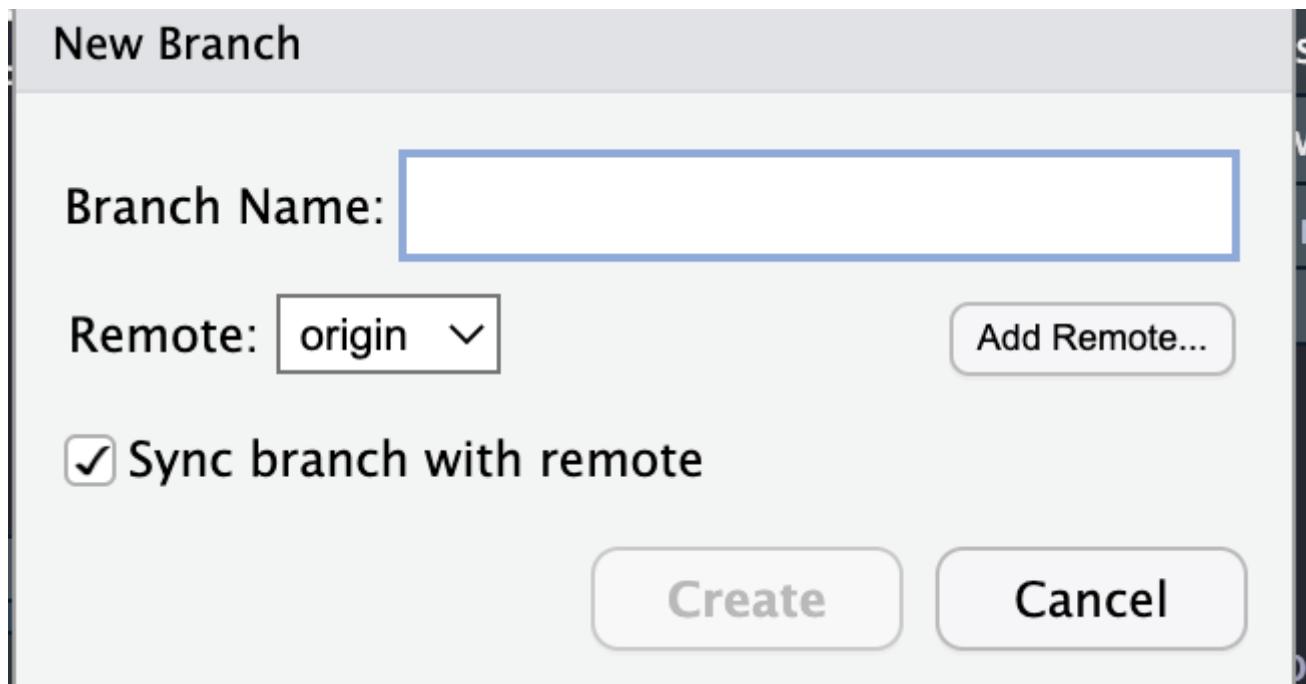
Sometimes you have a hierarchy - when one of you is the lead (author for example, or if you are working with an RA etc). where you want to review and approve any changes before they are made. In those cases you work with branches.

In order to create new branches, go to the 'git' tab on your RStudio console and click 'new branch'.



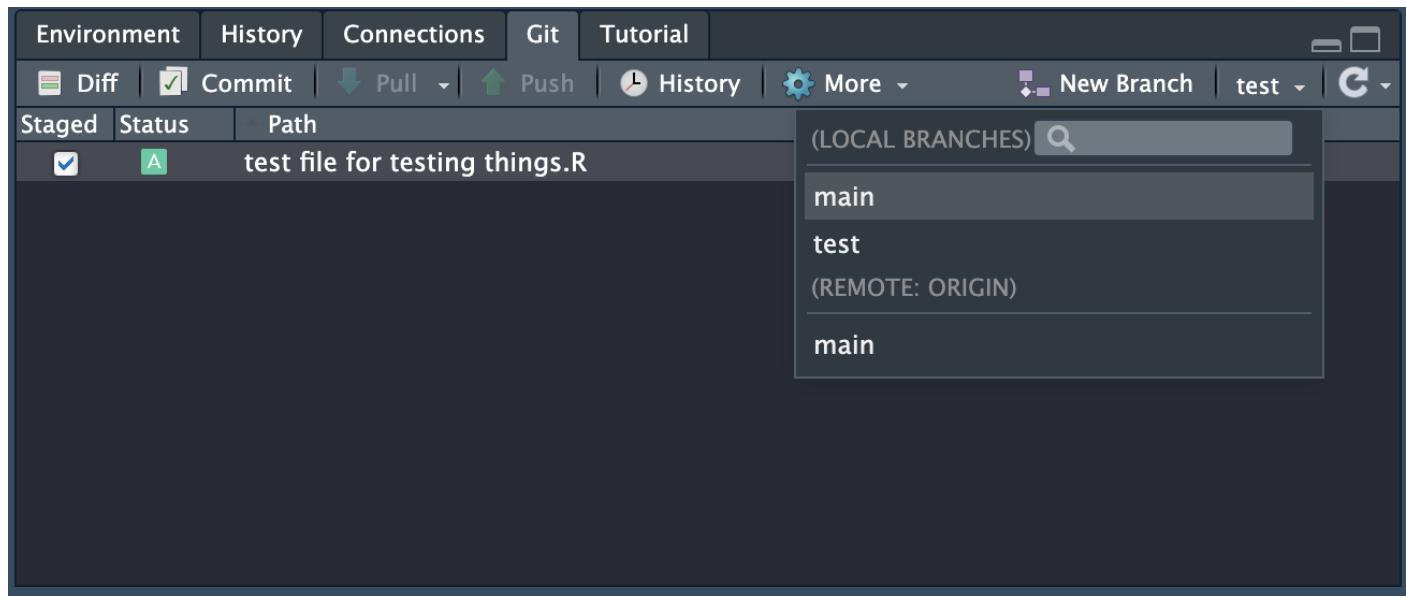
Creating a New Branch

You can then populate that branch the way you want and ask your associate to work on that branch only (your associate can also create a branch to be reviewed later)



Then click create and you have made the new branch.

You can see your branches like so:



Seeing Branches

Then simply create the new branch and you're done.

If you click on the new branch, you will then see this as you switch:

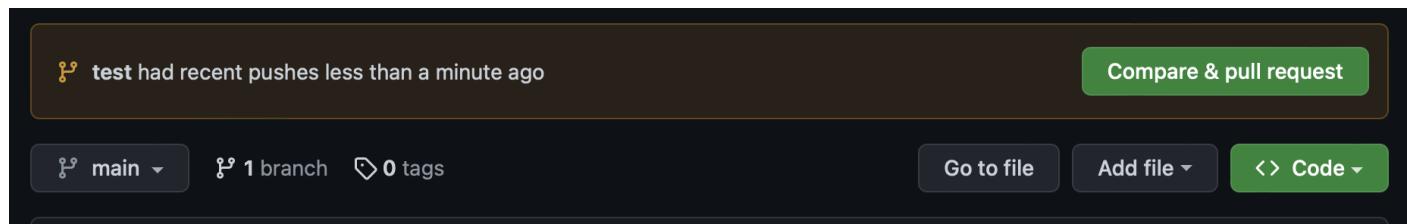
Git Checkout test

[Close](#)

```
>>> /usr/bin/git checkout test
Switched to branch 'test'
A       test file for testing things.R
```

This branch will initially not be published to the repo. You can publish it via the github website or the github desktop client.

You will then see this on your github page:



Exercises

Start a github project with some documents

Invite your partner to join

Pull your partner's documents,

Edit the documents, commit the changes and push changes to their project

Troubleshooting unintended branches

If you try to push when you have not pulled changes that your collaborator has made on a main branch you get the following message:

Git Push

Close

```
>>> /usr/bin/git push origin HEAD:refs/heads/master
To https://github.com/J-Dewar/729B.git
! [rejected]      HEAD -> master (fetch first)
error: failed to push some refs to 'https://github.com/J-Dewar/729B.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

If you try to pull before pushing changes that you have made to documents that your collaborator has changed you get the following warning

Git Pull

Close

```
>>> /usr/bin/git pull
From https://github.com/J-Dewar/729B
    705c369..62151f4  master      -> origin/master
error: Your local changes to the following files would be overwritten by merge:
      Quarto and Github Module.qmd
Please commit your changes or stash them before you merge.
error: The following untracked working tree files would be overwritten by merge:
      .gitignore
      729B.Rproj
Please move or remove them before you merge.
Aborting
Updating 705c369..62151f4
```

Here the program is creating branches for you so that changes are not lost. The solution to #1 is to pull before you start making any changes to make sure you are working on the most up to date version. The solution to #2 when you both have made changes that need to be reconciled is to

- a. Commit your local changes then you receive the following message:

Git Pull

[Close](#)

```
>>> /usr/bin/git pull
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only      # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
```

then you pick from the options merge rebase and fast-forward

for a detailed explanation of the differences between the different options see for example this [website](#)

To employ the merge solution go to terminal and write:

The screenshot shows the RStudio interface. The top panel displays a Quarto document titled "Quarto and Github Module.qmd". The code in the document includes R console instructions, Git setup, and a section on connecting R to GitHub. The right side of the interface features a sidebar with a tree view of "R Console Instructions" and "Getting Started" sections. The bottom panel shows a terminal window titled "Terminal 1" with the command `git config pull.rebase false # merge` entered.

```
1 ---  
2 title: "GitHub"  
3 format: pdf  
4 editor: visual  
5 ---  
6  
7 A quick brown fox jumped over the lazy dog  
8 |  
9 # R Console Instructions  
10  
11 ## Getting Started with Git  
12  
13 Git should likely come pre-installed on your computer. In order to  
set up git connected to R, you can use the usethis package like so:  
14  
15 ```{r, eval=FALSE}  
16 #install.packages("usethis")  
17 ##This is for if you don't yet have the package  
18 library(usethis)  
19 edit_ait.confia()  
8:1 (Top Level) Quarto
```

Console Terminal × Background Jobs ×

Terminal 1 ~ /Library/Mobile Documents/com~apple~CloudDocs/Documents/GVPT 622/729B

wireless-10-105-61-23:729B jordandewar\$ git config pull.rebase false # merge
wireless-10-105-61-23:729B jordandewar\$

then you can pull the document. Once you pull the document you can scroll through and see where your merger conflict occurred.

Git Pull

[Close](#)

```
>>> /usr/bin/git pull
Auto-merging Quarto and Github Module.qmd
CONFLICT (content): Merge conflict in Quarto and Github Module.qmd
Automatic merge failed; fix conflicts and then commit the result.
```

You then have to resolve this conflict save and now you can push.

Good git hygiene

You can look to this [article](#) for other useful information for keeping good Git hygiene when collaborating.

Briefly the first four rules of thumb are:

Always Pull Before a Push

Pull frequently

Push infrequently

Commit Frequently

Additionally the [article](#) discusses optimal git branch for working together.

Merge “forward” frequently

Create Pull Requests Infrequently

To better understand these

Associating existing r studio projects with git

In order to associate an existing RStudio project with Git you will need to create a Git repository as described above and then follow the steps below.

✓ Setting active project to '/Users/hannahbirnir/Dropbox/Classes/2023/729B/Github/Git_tutorial/git_tutorial'

You will then get a prompt asking if you want to commit the files you've already created to your repo. Select yes (option 1). You should then also see the git tab.

- Call `gitcreds::gitcreds_set()` to register this token in the local Git credential store
It is also a great idea to store this token in any password-management software that you use
- Open URL '<https://github.com/settings/tokens/new?scopes=repo,user,gist,workflow&description=DESCRIBE THE TOKEN\\'S USE CASE>'

You will then get a number of options to select about what your token use case will be. This will be project-dependent.

You can learn more about the selections [here](<https://docs.github.com/en/developers/apps/building-oauth-apps/scopes-for-oauth-apps>) to help guide you in your process.

```
#library(gitcreds)
#gitcreds_set()
```

When prompted to enter a token or password, enter the token you just created. If you have entered one previously, you will be prompted to choose if you'd like to keep your credentials. If nothing has changed, select 1 and keep things as they are. If they have, follow the most applicable selection.

Optional: Github Desktop Client

A lot of problems can be resolved by examining the Github desktop client. It's free and you can download it from the Github website.