

# Git and GitHub

A brief introduction

# Git

- version control system
- works offline (*repositories* exist on your computer)
- tracks changes via *commits*
- has a command-line interface and integrations with GUIs (like RStudio)

# GitHub

- web-based platform built around Git
- provides a remote location for hosting Git repositories
- enables collaboration
- offers other features for project management (pull requests, issue tracking)

# Our Git/GitHub goals

- For **you**: Keep track of progress on projects
  - Go back when you need to
  - Don't lose old work
  - Easily search the history of a project
- For **others**: Share your work
  - Have a place to store and link to code
  - Read and interact with others' code

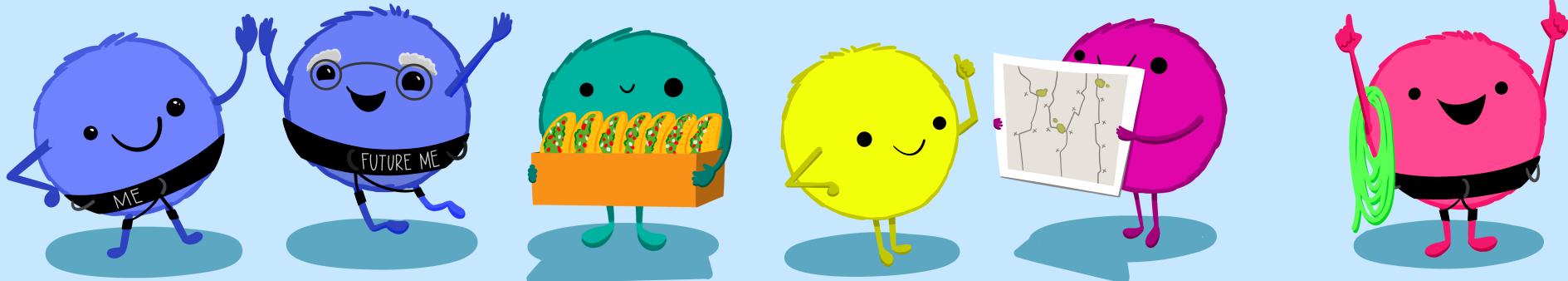
There is a *lot* to learn about this topic and I am *not* an expert on everything!

# What we won't cover

- Collaboration
  - When multiple people are working on the same GitHub project, things get a little more complex
  - I went though almost my whole PhD without working on shared GitHub projects and only now do I feel semi-confident collaborating!
  - I think it's best to figure things out in your own projects first
- Git on the command line
  - There are a lot of functions you might hear about (`git fetch`, `git merge`, etc.)
  - RStudio and GitHub will have everything we need!

**“Collaboration is the most compelling reason to manage a project with Git and GitHub.** My definition of collaboration includes hands-on participation by multiple people, including your past and future self, as well as an asymmetric model, in which some people are active makers and others only read or review.”

-JENNY BRYAN



Bryan, J. 2017. Excuse me, do you have a moment to talk about version control? PeerJ Preprints. 5:e3159v2. DOI: 10.7287/peerj.preprints.3159v2

Illustrations from the Openscapes blog **GitHub for supporting, contributing, and**



# Git

- version control system
- works offline (*repositories* exist on your computer)
- tracks changes via *commits*
- has a command-line interface and integrations with GUIs (like RStudio)

# GitHub

- web-based platform built around Git
- provides a remote location for hosting Git repositories
- enables collaboration
- offers other features for project management (pull requests, issue tracking)

# Workflow

Create a repository (clone from GitHub, or create on your computer and connect to GitHub)

1. Write some code!
2. When you complete “something”, **add** it to the **staging** area
3. Write a brief description of what you did (“added linear model”; “created table 1”) and **commit**
4. **Push** to GitHub
5. Repeat!

As long as you are working on your own, all on the same computer, you don’t need to worry about pulling



# What is a commit?

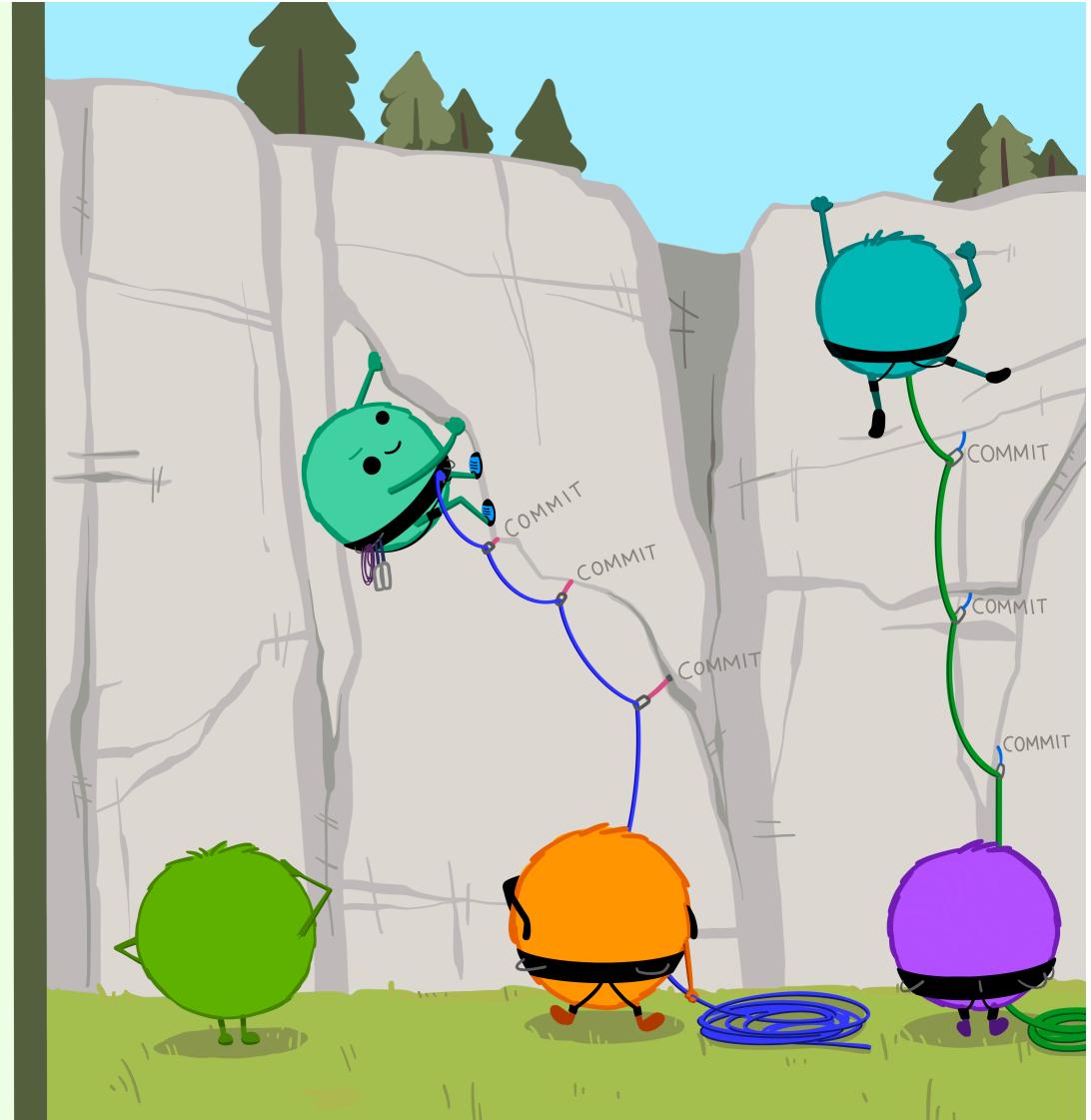
“

Using a Git commit is like using anchors and other protection when climbing...**if you make a mistake, you can't fall past the previous commit.**

Commits are also helpful to others, because **they show your journey, not just the destination.**”

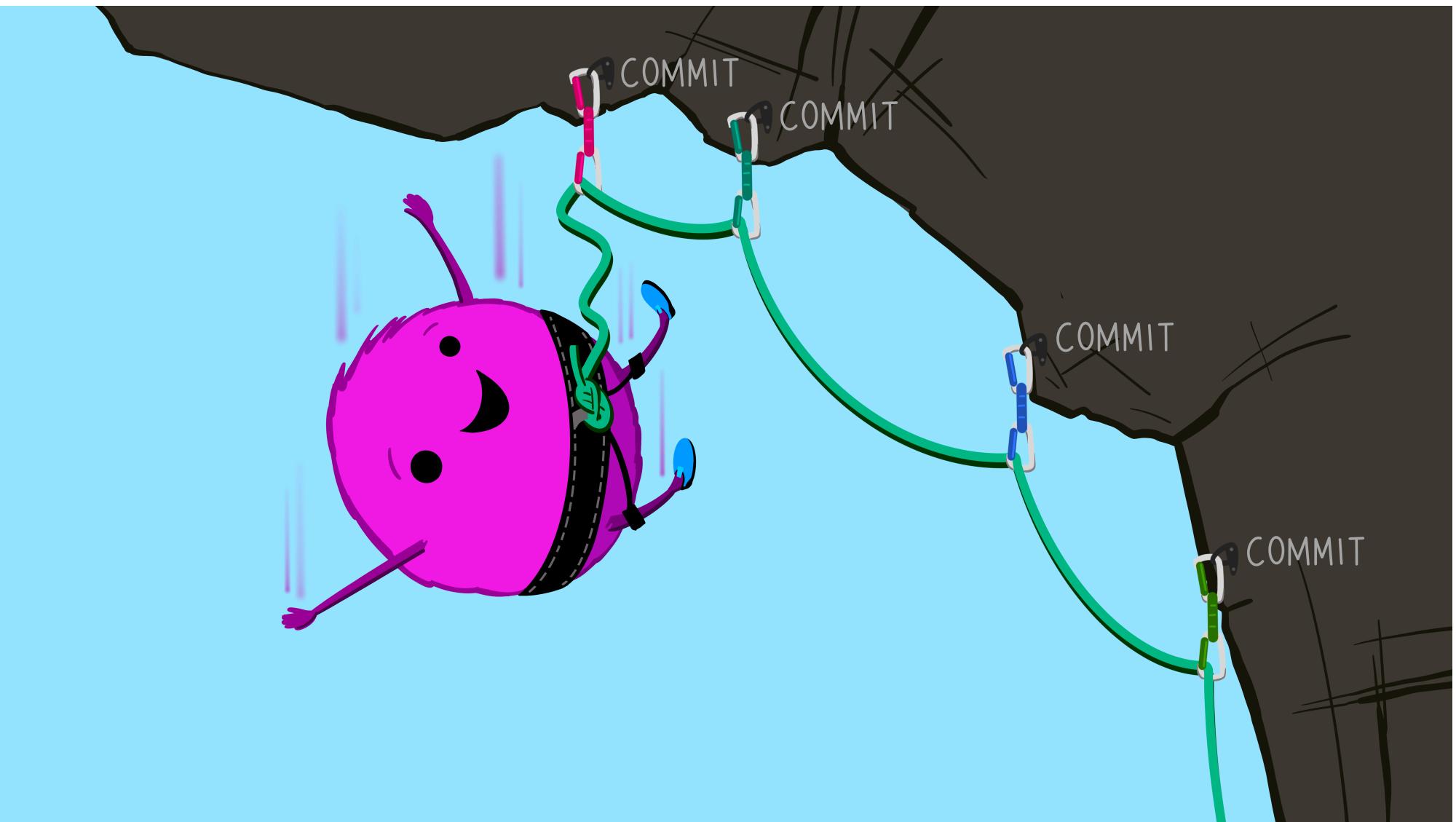
— HADLEY WICKHAM & JENNY BRYAN

Wickham & Bryan, R Packages (<https://r-pkgs.org/preface.html>)



Illustrations from the Openscapes blog **GitHub for supporting, contributing, and**

What should you commit? Whatever you  
don't want to lose!



Illustrations from the [Openscapes](#) blog **GitHub for supporting, contributing, and**

<b>additional sensitivity analyses, re-render</b>	 3a9a03f 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>figure out mice problem</b>	 4a75681 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>tried and failed imputation</b>	 21b38eb 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>add sensitivity analyses</b>	 945ca2c 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>rerender text</b>	 efb9476 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>updates to text</b>	 3d6da19 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>render doc</b>	 91ecf49 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>update pic</b>	 16a8b21 
 <b>louisahsmith</b> committed on Mar 11, 2021	
<b>add vertical line to pic</b>	 609e76a 
 <b>louisahsmith</b> committed on Mar 11, 2021	

If you know that your code worked at 10am on October 21, 2015, and now it doesn't, you can return!



# Exercises

1. Fork the repo (repository) at  
<https://github.com/louisahsmith/epi590r-in-class>
2. On **your** fork of the repository, click the green “Code” button. We are going to *clone* the repository to your computer using *HTTPS*.

## Forking

- **Purpose:** Used to create a personal copy of another user's repository on your GitHub account.
- **Ownership:** The forked repository is still on the original owner's account, and you get your own copy to work with.
- **Collaboration:** Allows you to make changes without affecting the original repository. You can make changes, commit them to your fork, and then propose these changes to the original repository through pull requests.
- **Relationship:** The forked repository remains connected to the original, but changes aren't automatically synced.
- **Use Case:** Commonly used when you want to contribute to a project that you don't have direct write access to.

## Cloning

- **Purpose:** Used to make a local copy of a GitHub repository on your computer.
- **Ownership:** You have a read-write copy on your local machine, but it's not automatically linked to your GitHub account (you can do so through RStudio).
- **Collaboration:** Allows you to work on the project locally and make changes, but these changes aren't automatically visible to others.
- **Relationship:** The cloned repository is a standalone copy, and changes won't automatically affect the original or other clones.
- **Use Case:** Useful when you want to work on a project locally and have full control over commits and pushes.

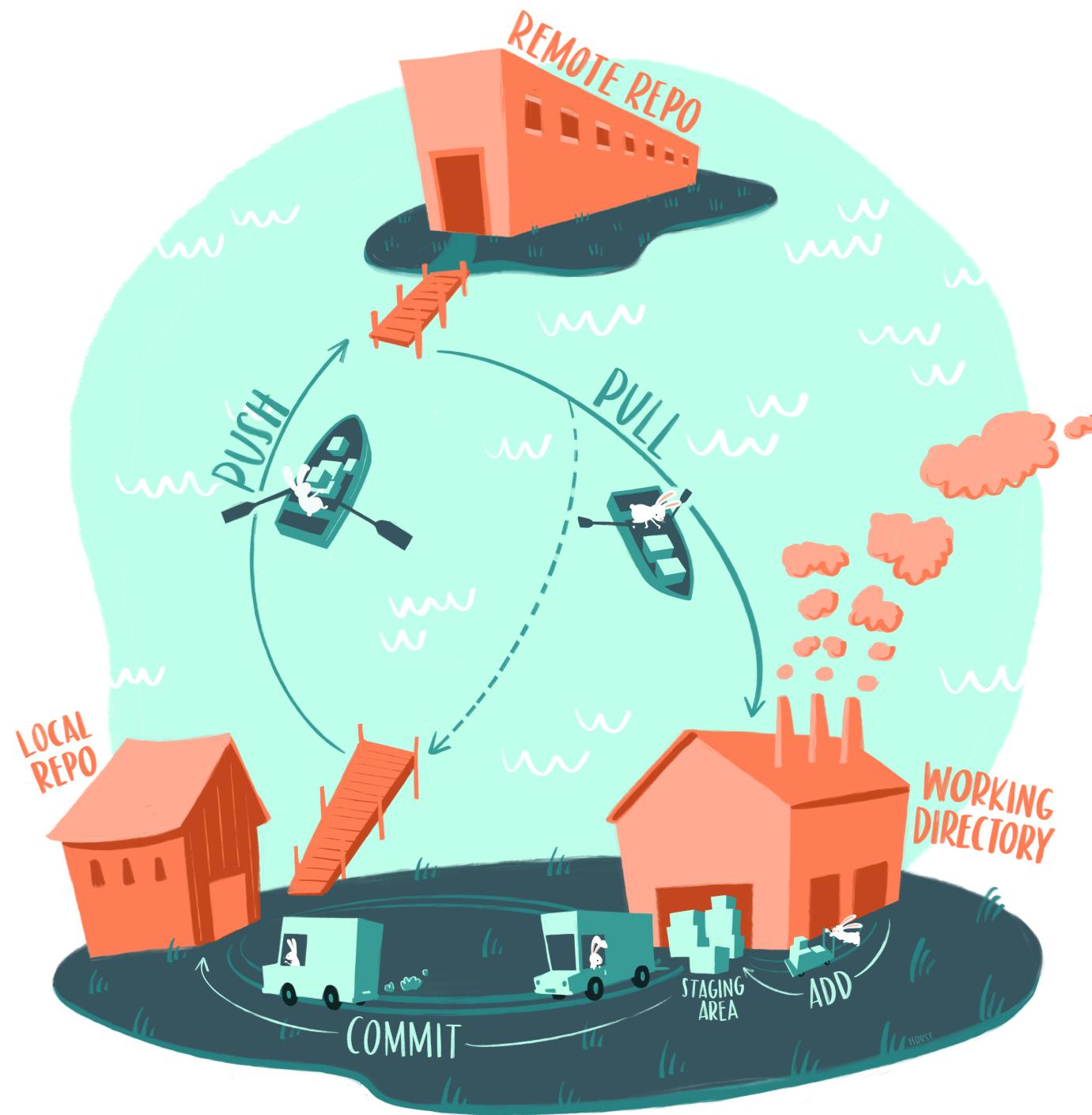
You have a forked repo on GitHub, now you are cloning that forked repo on your own computer

3. Open up RStudio.

- File > New Project > “Version Control” > “Git”

4. Paste the URL to your fork

- Name the project directory (easiest if it has the same name as the repo)
- Choose where you want to store the project (remember this spot!)
- Create Project!



# Practice making a change, staging, committing, pushing

5. From the filepane in RStudio, open `README.md`
  - Change the file and save your changes
6. In your Git pane, click on the checkbox to stage the file
  - Then click “Commit”

# Exercise: edit, commit, push readme file

15 : 00