

# HOW TO: GitHub Desktop

# Introduction

Source: Daniel van Strien. 2018. An Introduction to version Control Using GitHub Desktop. Retrieved from <https://programminghistorian.org/en/lessons/getting-started-with-github-desktop>

## Getting Started

<https://desktop.github.com/>



Overview Release Notes Help

# The new native

Extend your GitHub workflow beyond your browser with GitHub Desktop, completely redesigned with Electron. Get a unified cross-platform experience that's completely open source and ready to customize.

## **GitHub Desktop**

is a Graphical User Interface (GUI) for Git (a version control command-line system)

Advantages:

- Track changes in your documents, code, data uploads
- Revert changes
- Online storage
- Keeping track of collaborative projects
- Open source with community support

## Terminology

Online Glossary - <https://help.github.com/en/articles/github-glossary>

### Cloning a Repo

- creating a local copy of the code provided by a developer

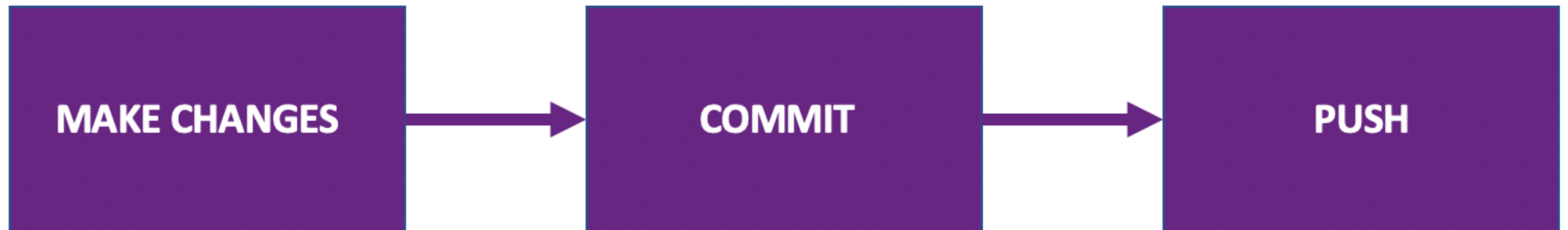
### Commit

- Saving individual changes
- Commits usually contain a commit message - a brief description of changes

### Push

- Sending your committed changes to a remote repository on GitHub

STEP 1: Make changes > STEP 2:  
Commit with a description about these  
changes > STEP 3: Push



## **Register for a GitHub Account**

Register for an account at <https://github.com/>. For this class you will be creating a PRIVATE repository to prevent your work/code from being inappropriately shared.

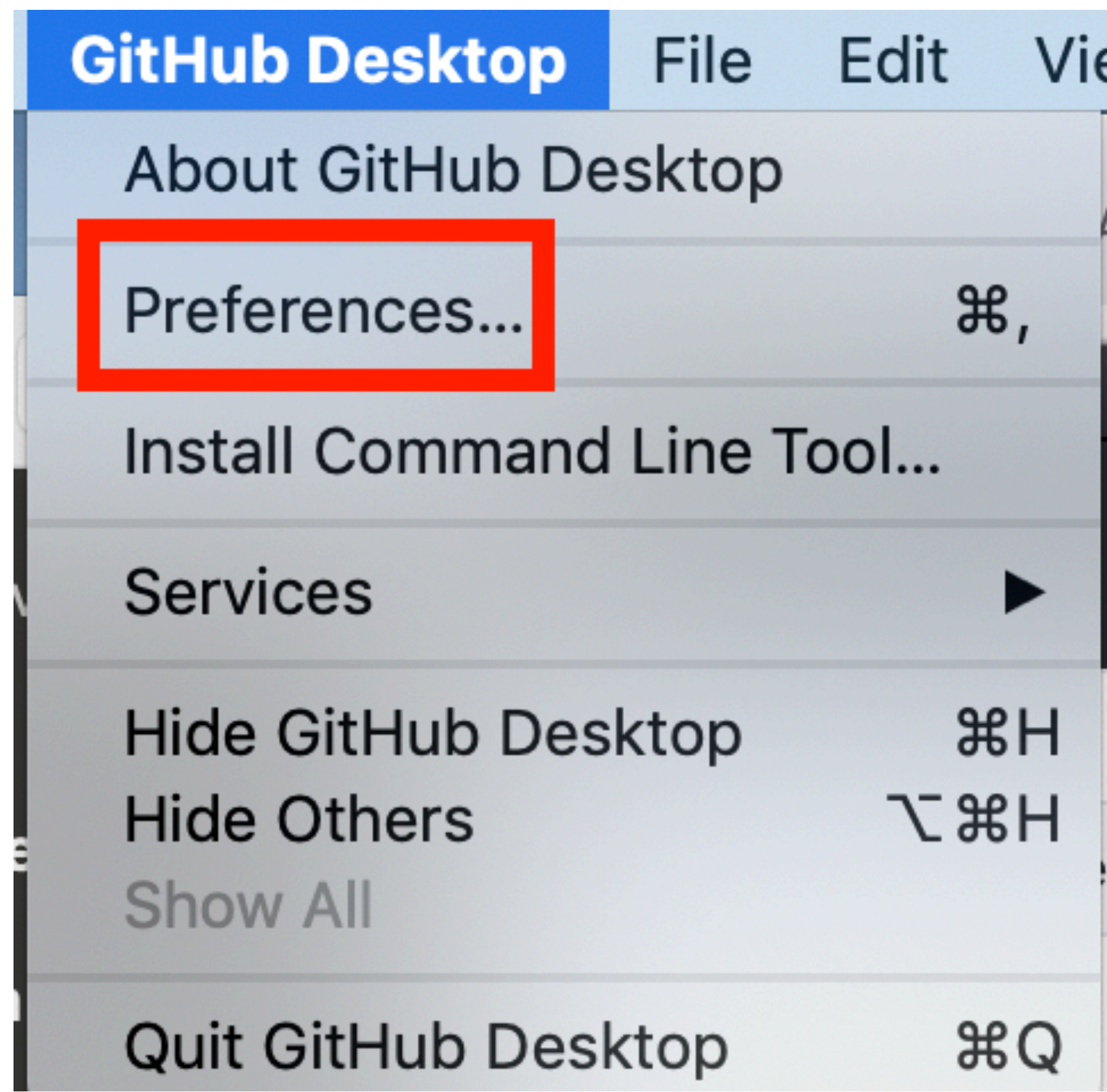


## Install GitHub Desktop

Download GitHub Desktop

<https://desktop.github.com/>. Once you have installed the app, open the app and link it to your GitHub account.

**Step 1:** GitHub Desktop Menu > Preferences





**Step 2:** Preferences > Accounts > Sign In with [GitHub.com](https://github.com)

Preferences

Accounts

Git

Appearance

Advanced

GitHub User name  
and password

×

GitHub.com

Sign in to your GitHub.com account to access your repositories.

Sign In

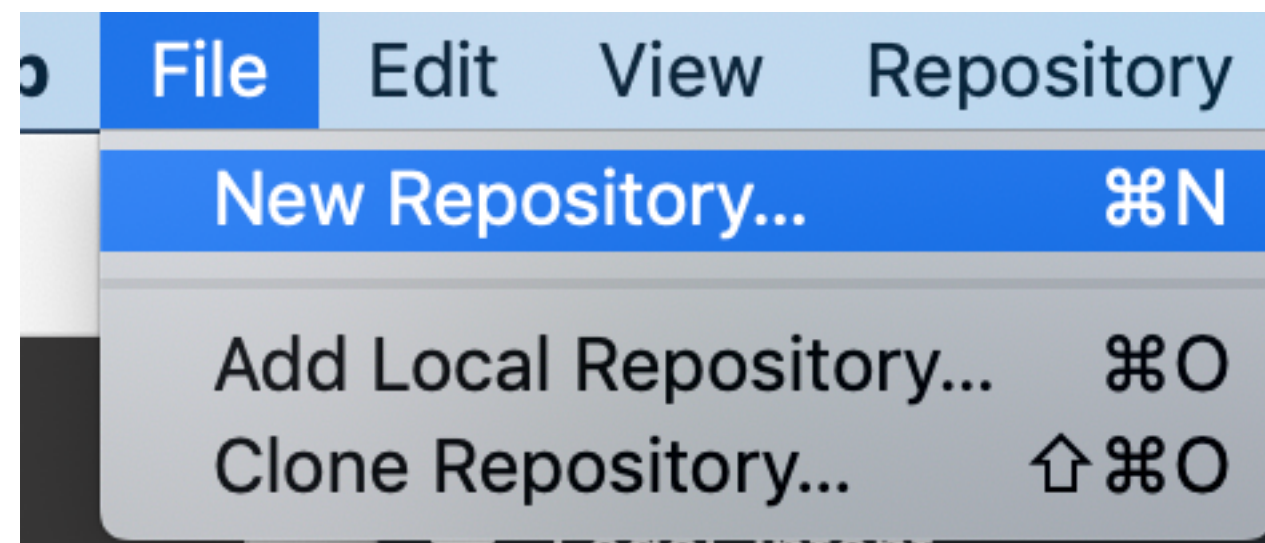
Enterprise

If you have a GitHub Enterprise account at work, sign in to it to get access to your repositories.

Sign In

## Create a New Repository

**Step 1:** File > New Repository



## Step 2: Create a New Repository

Create a New Repository

Type a name for your new folder

Name

dataanalytics

Optional

Description

the repository for my final project

Local Path

/Users/olgascrivner/Documents/IU-DataScienc

Choose...

Select where to add you new repository

☒ Initialize this repository with a README

Git Ignore

None

Select

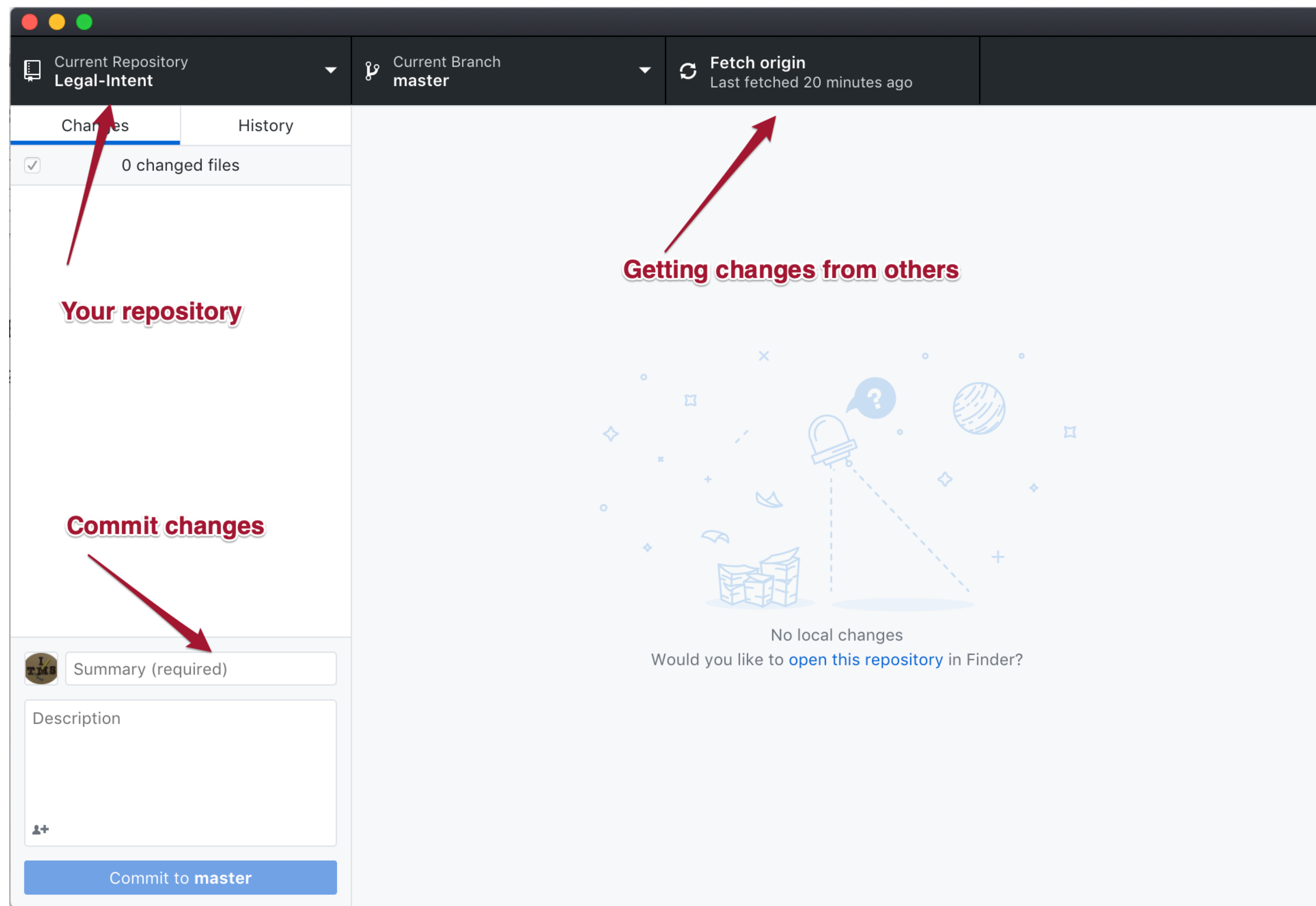
License

None

Cancel

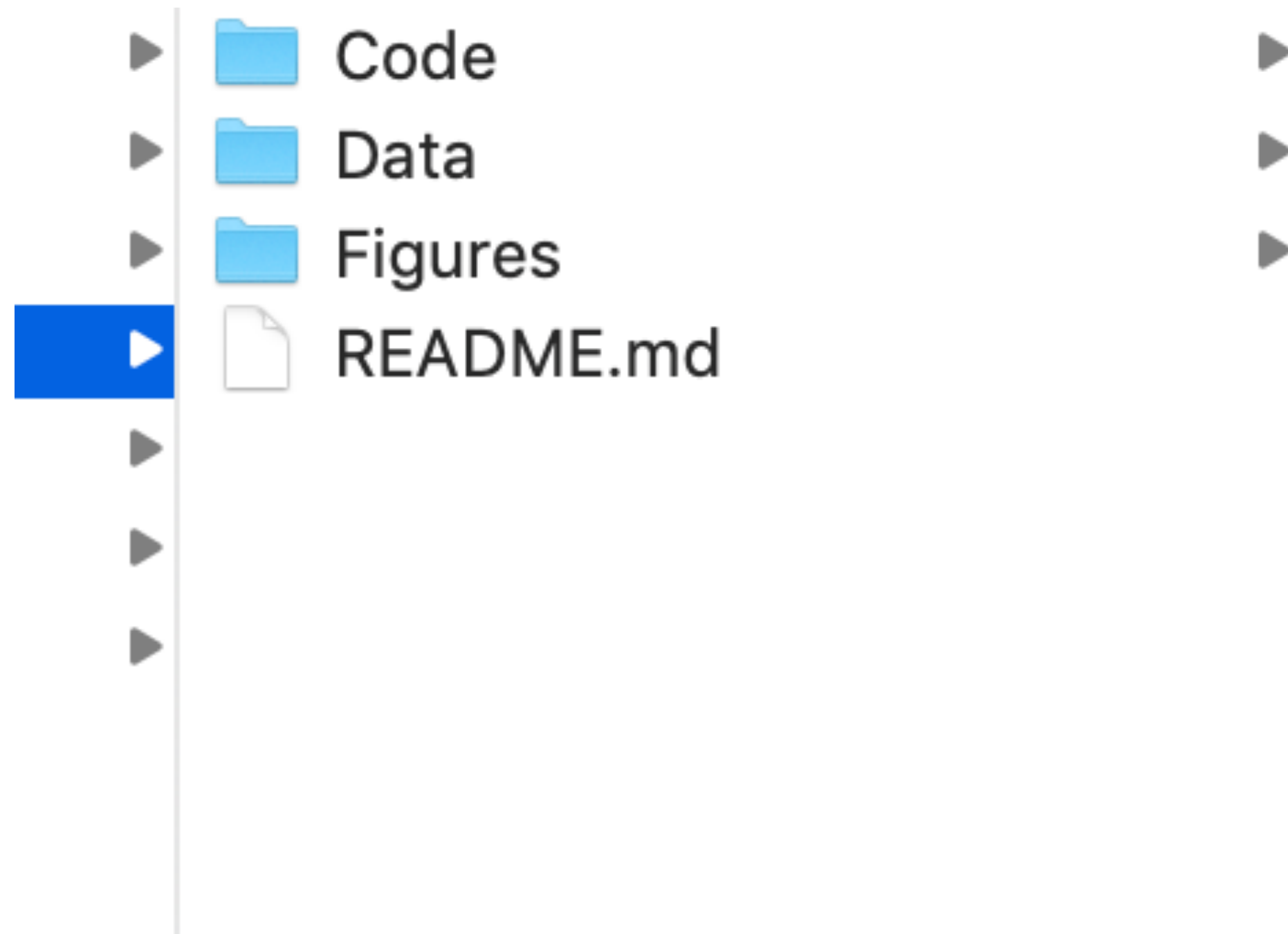
Create Repository

Click



Layout of your GitHub Desktop

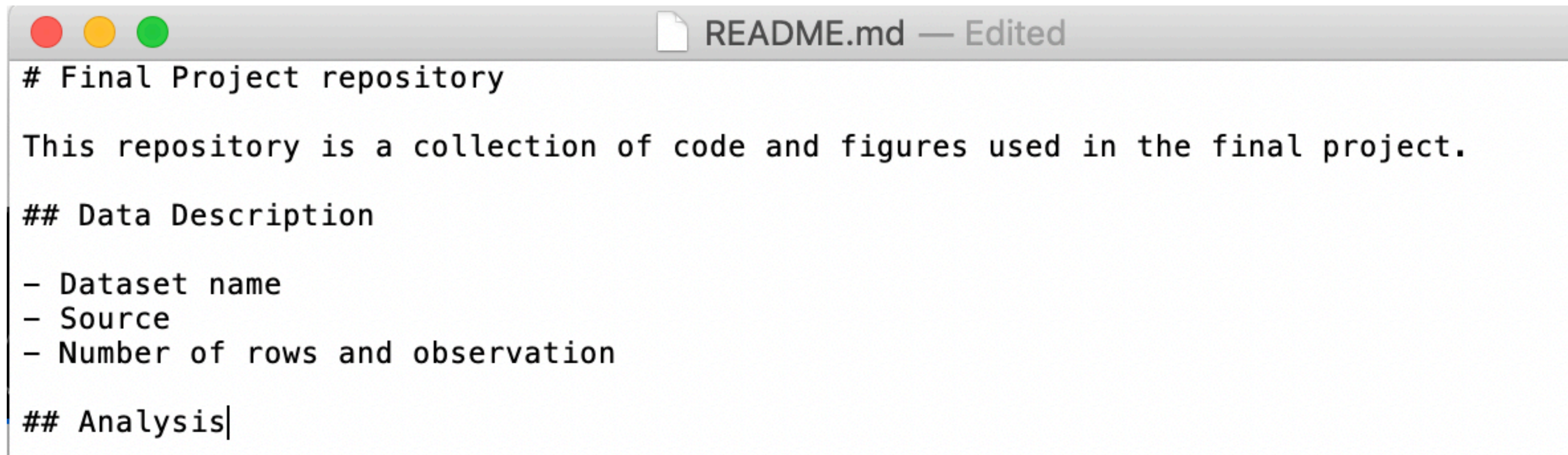
**Step 3:** You can add subfolders to your local repository, for example Code, Data, Project etc



Note: Empty folders will not be shown in a remote GitHub repository



**Step 4:** Add description to your Repository > Open README.md (use any text editor). Add a brief description (You can modify it later)



```
# Final Project repository

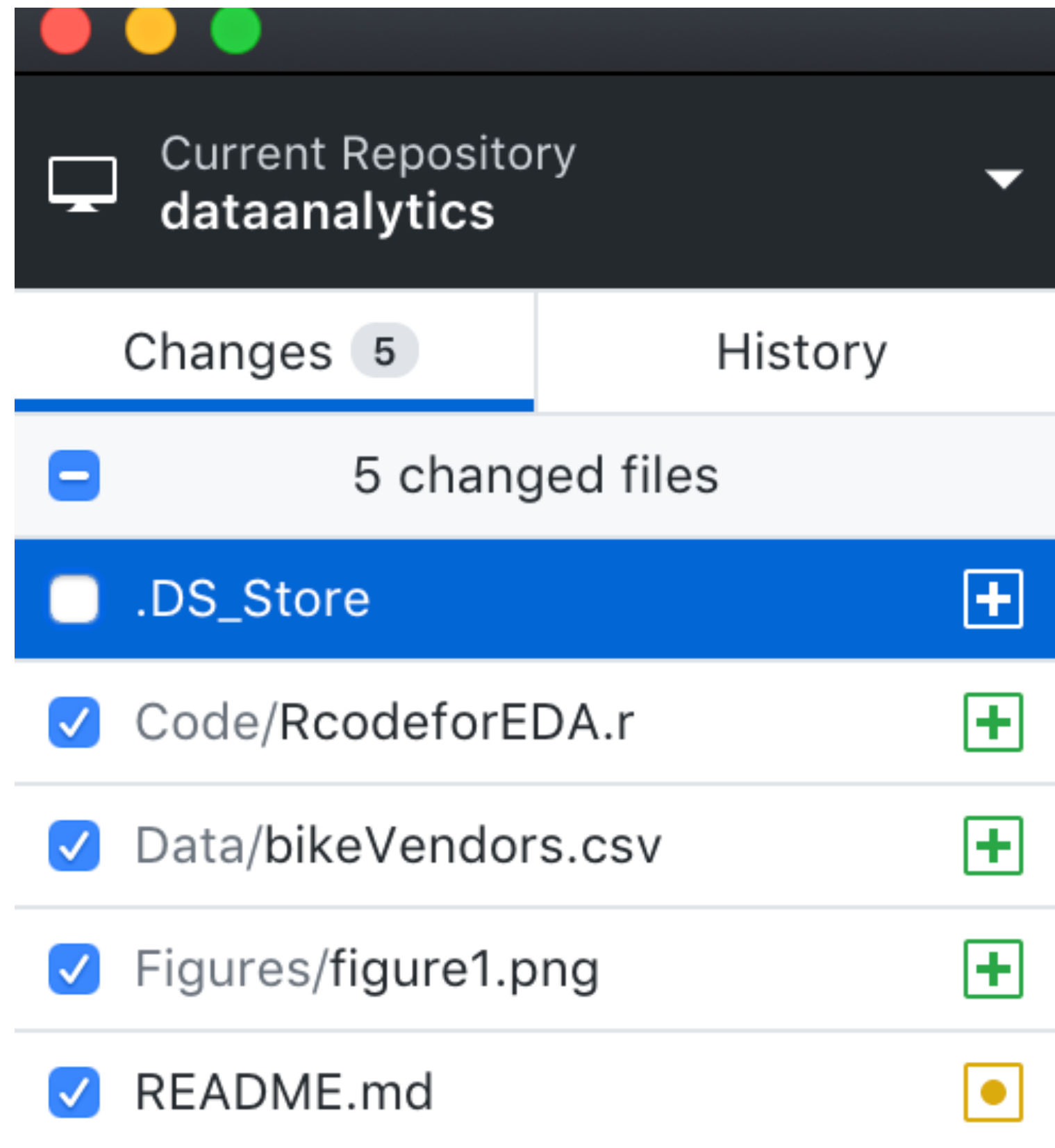
This repository is a collection of code and figures used in the final project.

## Data Description

- Dataset name
- Source
- Number of rows and observation

## Analysis|
```

## Step 5: Select and Commit



**You can select specific  
Files for repository**



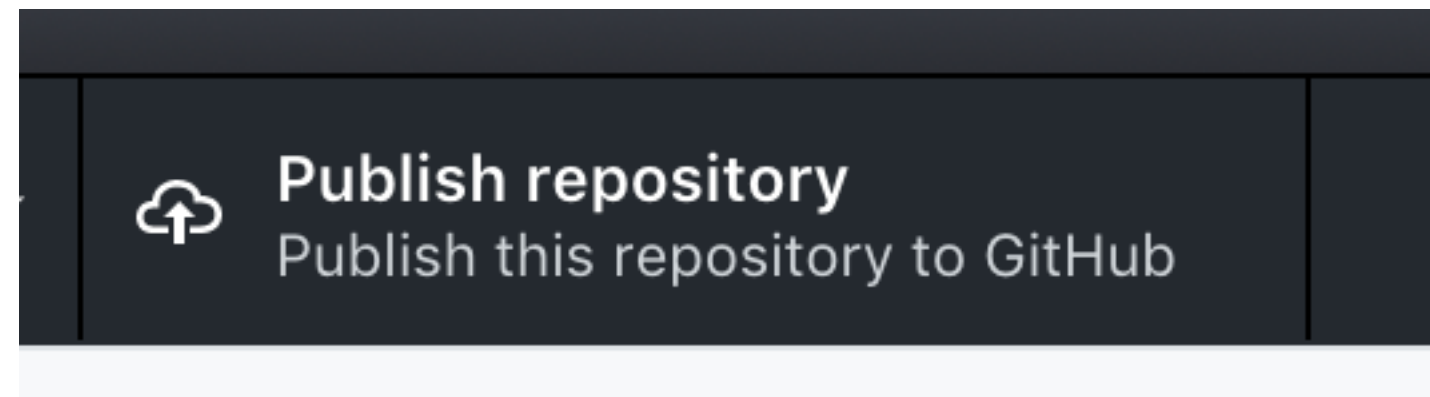
first commit

**Type your  
brief  
message**

Description

Commit to master

## Step 6: Publish



# Check your online Repository (GitHub my repositories)

2 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

obsrivn

first commit

Latest commit a25aaa2 4 minutes ago

Code	first commit	4 minutes ago
Data	first commit	4 minutes ago
Figures	first commit	4 minutes ago
.gitattributes	Initial commit	5 hours ago
README.md	first commit	4 minutes ago

README.md

## Final Project repository

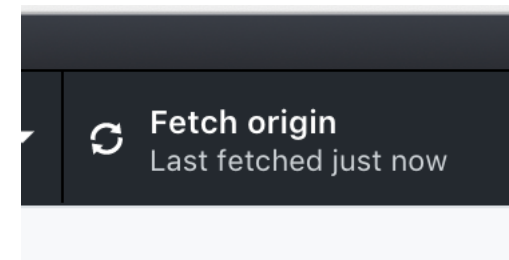
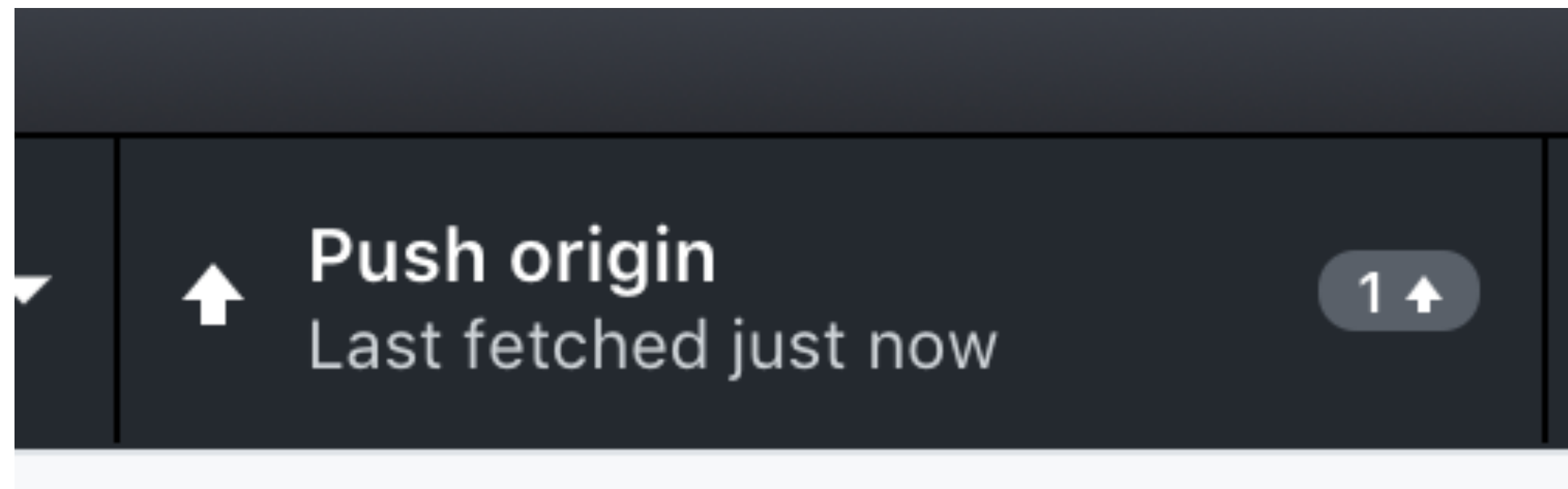
This repository is a collection of code and figures used in the final project.

### Data Description

- Dataset name
- Source
- Number of rows and observation



Once it is published, you will be using **Push** to send to a remote GitHub repository and **Fetch** to synchronize your local files with the remote repository (if changes are made online).



## Cloning Remote Repository

When you need to save a remote repository locally, clone it and select Open in Desktop.

