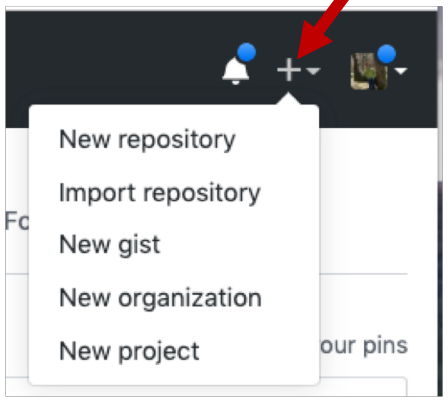
GitHub Projects

Moving from the NSS Data Analytics Cohort 1 account
to your personal account

Step 1: Create an empty repository with a short descriptive name for the project



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template
Start your repository with a template repository's contents.

No template

Owner MaryLvV / **Repository name** * scooter-availability-analysis

Great repository names are short and memorable. Need inspiration? How about super-doodle?

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

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

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: Python | Add a license: None

Create repository

Short name that tells what the project is about

Make it public

***Always* initialize with a README and choose a gitignore**

Step 2: Clone the repo and move *your work* from the classroom repo to the new repo on your account.

***If it is not possible to separate your work from your teammates' work, you must acknowledge them as contributors in the README**

Step 3: Build out your README.md file (I like using Visual Studio Code for this because of the preview feature!)

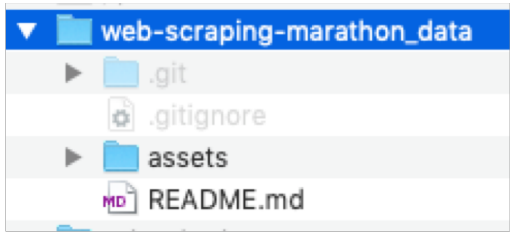
- **Introduce and describe your project**
- **Have a data section that describes the data that you used**
- **It's fine to build on the existing README from class OR create your own**

Here are some examples of excellent READMEs:

https://github.com/jroth006/The-future-of-music
https://github.com/jroth006/tn_achievement_analysis
https://github.com/taylorperkins/New-Constructs-NLP-Text-Classification
https://github.com/taylorperkins/Top-100-Spotify-Analysis

Step 4: Put all data files in a folder named data

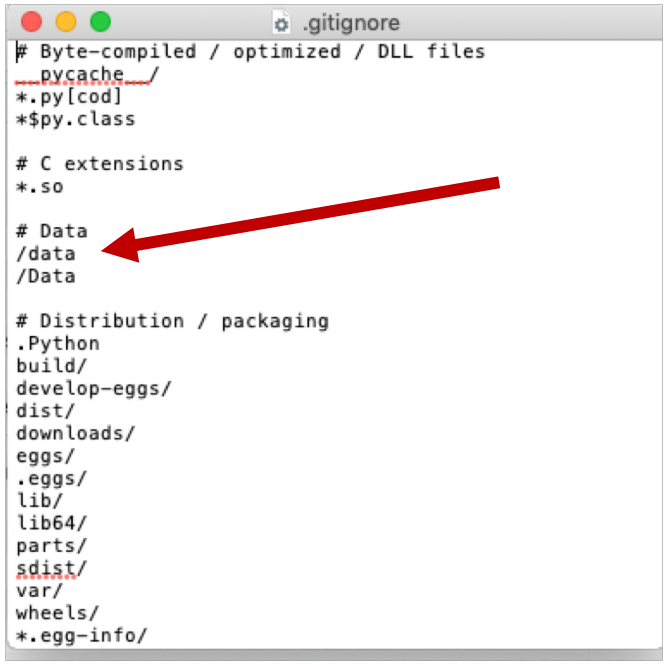
Step 5: Open the .gitignore file for your repo with a text editor.



- The .gitignore file is a hidden file, so you will need to change the view properties for your file explorer (Finder in Mac, File Explorer in Windows) to show hidden files

Step 6: Add a data section to the .gitignore and tell git to ignore everything in your data folder

- This keeps all data files local. Company data is shared for class purposes only, and should not be made publicly available!



Step 7: Add, commit, and push your work! Iterate as needed.