Project 1
Reproducibility and GitHub
Sociology 273L: Computational Social Science

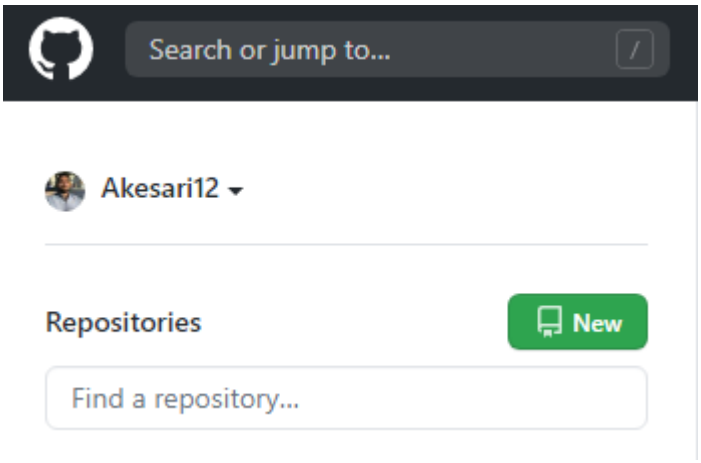
1 Introduction

In this project, you will learn the fundamentals of creating a reproducible data science workflow. You will use GitHub to collaborate with others on code and create your own website. Throughout the project, make sure you copy all code you write in the terminal to a markdown or plain text file as we'll need it later. As with all projects, get started early, take breaks often, and consult your peers and the instructor regularly! I find that putting in a little time each day tends to lead to the best experience.

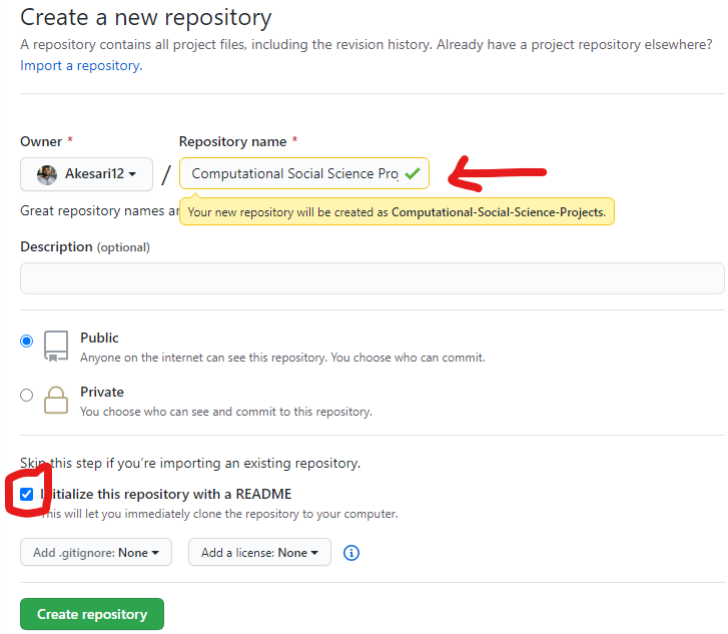
2 Creating a GitHub Repository

2.1 Initialize a New Repository

In class, we learned how to clone an existing repository to our local machines, and we went through some of the common git commands like pulls, commits, and pushes. In this project, we'll practice creating your own repositories to host your code throughout the year. The first step is to initialize a new repository that will host your code. First navigate to GitHub, and then find the button to create a new repository.



Once you do this, you will be brought to a screen where you can name and create your repo. Be sure to give it the name: "Computational Social Science Projects." Then, make sure you click the button to initialize the repository with a README (note that this will not be automatically checked off).



Now that you have initialized a repository, you're ready to practice some of the common commands we learned in class. Do the following steps entirely with code, and make sure to copy and paste the code output to a markdown or text file.

1. Clone your repository to your local machine.
2. Create a folder called "Project 1".

3. Create a text file in the "Project 1" folder.

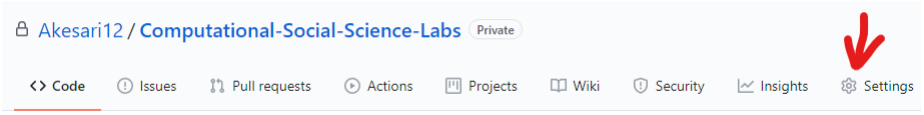
4. Edit the text file to include all of the commands you have run so far, as well as commands to add, commit, and push changes to your github repo.

5. Run the commands to add, commit, and push changes to your github repo.

3 Collaborate with Teammates

In this section, you will learn to use GitHub's collaborative tools. We will need these tools throughout the year to work together on projects, so practicing them now will be helpful later on. Do the following:

- 1. Have one group member initialize a new repository titled "GitHub Collaboration Practice".
- 2. Invite other group members to collaborate. To do this, navigate to the "Settings" tab on the web interface, navigate to "Manage Access," and then add your teammates by their GitHub handles.



- 3. Create markdown or text file with the text "Group Project" on the first line.
- 4. Each group member should then do the following in code except point (c), which is done only by the member who initialized the repository:
  - a. Clone the repository.
  - b. Checkout a new branch.
  - c. The member who initialized the repository should add this text: "Degree Programs & Research Interests of Members". They should then add, commit, and push changes to the repo. Important: This step should ONLY be done by the member who initialized the repository.
  - d. After the member who initialized the repository completes step (4c), everyone should merge the main branch into your branch to ensure your branch is up to date with the main branch.
  - c. Edit the text file to add your degree program and some keywords for your research interests.
  - d. Create a new text file with all of the commands you ran, and the commands you will run to add, commit, and push changes to the repo.
  - e. Add, commit, and push your changes to the repo.

- 5. Then in the GitHub web interface, each group member should do the following:
  - a. Make a pull request.
  - b. Assign someone to review the pull request.
  - c. Look at the requests your teammates have assigned to you. Use the "compare" feature to see differences between branches. Review and merge them into the main branch. Resolve any merge conflicts. If two members edit the same line (which is likely in this situation) and try to merge to the main branch, there will be conflict.

4 Create Your Own Website

Our last task will be to create a personal website using GitHub Pages. Having a personal web presence can be useful in academia and industry for a variety of reasons. In this case, we will use your personal webpages to showcase your work over the year, and you can build on this however you wish with your teaching, research, and other interests. You can find detailed steps for creating your own personal webpage in the above GitHub Pages link. Note that you can create as many project pages as you like, but only one personal page per GitHub account. The steps for creating a personal page are as follows:

- 1. Navigate to GitHub, create a new repo and name is with the convention "username.github.io". For example, Aniket's repo would be "akesari12.github.io".
- 2. Clone the repository.
- 3. Create a file called "index.md"
- 4. Populate "index.md" with some content about yourself. Try including text, images, and anything else you would like.
- 5. Create another page by creating a file called "projects.md" and populate it with a short description of this first project.
- 6. Add, commit, and push changes to your repo.
- 7. Navigate to the website to check and make sure everything worked!