Project Procedures

Sociology 273L/M: Computational Social Science

1 Introduction

This document outlines the procedure for working on and submitting projects for Computational Social Science. Please read these instructions carefully. Feel free to contact the instructors with any questions about these procedures.

2 Initialize the Project Repository

1. You will be assigned to teams of 3-4 for each project. All team members are expected to contribute equally. Please let the instructors know as early as possible if there are any issues or concerns.

2. One of the team members should initialize a new GitHub repo for the project. Name it with the convention "CSS_Project #_Team #_Fall 2022" and substitute the Project number and Team number accordingly.

3. Add the other teammates and the lab instructor (tomvannuenen for Fall 2022; prsnt1 for Spring 2023) as collaborators.

4. Add the relevant project files from the course repository to your group's repo (e.g., for Project 2, copy materials from the "Project 2" folder). With the exception of Project 1, we will often provide you with a .ipynb or .Rmd template to get started.

5. In general, it is good practice to use git branches if you will all be working on the same file. This will ensure that you do not overwrite each other or run into merge conflicts as you're working.

3 Asking for Help

Learning how to code is a challenging process with a steep learning curve for beginners. One of our goals in this course is to equip you with the skills to troubleshoot your own code by reading documentation, searching online to see if people have had similar problems, and collaborating with team members to get help. While the instructors are more than happy to help you, we also want

you to develop the confidence to solve your own problems as they arise so that you are ready to use these methods in your own research. Therefore, we ask that when a technical question arises with regards to code or modeling, you take the following steps:

1. Make sure to check the documentation for the libraries that you are using. For beginners especially, syntax errors can be quite common, so make sure the code is written the way Python expects. The other most common error is a TypeError, meaning you passed an object type that a function was not expecting. For example, you might have passed a dictionary to sklearn when it was expecting to see a dataframe.

2. If you still can't figure out a solution, try using resources like Google, GitHub, and Stack Overflow. Stack Overflow in particular is a good resource because people with similar questions will have posted their questions and code, and someone else will have provided an answer. Learning how to find a similar example and adapting it to your own code is one of the most important skills you will need as a programmer.

3. Check with your teammates to see if they can help. Try using pull requests here!

4. If after these steps, you still cannot solve your problem, then you may ask the instructors for help. When you do, make sure you:

   - Checkout a new branch through git or on GitHub.
   - Push the branch with the error to the repository.
   - Open a pull request and assign the instructor (tomvannuenen) as the reviewer.

- In the pull request comments, detail what the error/problem is, what your question is, and any steps you took to try to resolve it. Include links to stackexchange/stackoverflow threads, documentation, or anything else that would be helpful.
- We will then interact through GitHub where I may suggest changes. Once we come to a solution, you can then merge into the main branch and the request will be closed.

The goal of this procedure is to help you get familiar with the typical workflow for dealing with errors in code. It also helps ensure that when you come to the instructors with a question, we can work with a fully reproducible example. Being able to work with a reproducible example reduces much of the back and forth between you and the instructor, and helps you solve your problem faster.

4 Submission

To submit the assignment, make sure that your final notebook is in the **main** branch. Make sure any other notebooks or unnecessary files are in other folders. Also make sure any supporting files that are necessary to run the notebook (data, external functions, etc.) are in the main branch. At 11:59 PM on the due date, we will pull all of the repositories and grade the notebooks. As part of the process, we will run the notebooks again. Make sure file paths are consistent, random seeds are set, and do anything else that you need to do to ensure that the notebook will run the way that you intended. The goal is that we should be able to reproduce your notebook entirely just by using the files on the GitHub repo.