

Week 1 Assignment

DATASCI 200: Introduction to Data Science Programming

This assignment is due at 11:59 PM PST the day before your second (week_02) live class. (So if you have class on Tuesday this is due Monday night at 11:59 PM PST)

Command line and bash scripting

Objectives:

- Build a Conda environment and make a requirements file
- Build a file tree using the command line
- Write a short shell script to make a file tree
- Demonstrate uploading files to Github

1.1. Quiz (30 points)

Login to gradescope using the link on bCourses. On gradescope, you will see an assignment named HW_Unit_01_pt1. This is the quiz section, please complete the quiz and submit when done. You can 'Re-Submit' as many times before the deadline if you want to change your answers.

- There are 10 multiple choice answers based on the first module each worth 2 points (total of 20)
- The last question asks for you to cut/paste your student github repo information into the text box (worth 10 points). You should have completed the github repo setup in async module 1.15.1.

1.2. Make a Conda environment and requirements.txt file (20 points)

For exercise 1.2, you will be creating a **conda virtual environment**. Virtual environments are important for programming projects because they manage package dependencies. We will discuss packages, modules and dependencies more later, but let's just say that each programming project requires a different set of Python packages (sets of programs), and these sets can be incompatible. An environment lets you run compatible sets of packages together and keep them away from other environments.

In the next steps:

- 1 - Build an environment
- 2 - Practice switching in and out of the environment
- 3 - List the component packages in the 'requirements.txt'

For this exercise, use the 'anaconda prompt' command line:

1. Create an environment:

```
conda create --name myenv
```

Note: Replace myenv with the environment name.

2. When conda asks you to proceed, type 'y':

```
proceed ([y]/n)?
```

This creates the myenv environment in /envs/. No packages will be installed in this environment.

3. Create an environment with a specific version of Python:

```
conda create -n myenv python=3.8
```

4. Create an environment with a specific package:

```
conda create -n myenv scipy
```

5. Create an environment with a specific version of a package:

```
conda create -n myenv scipy=1.5.0
```

6. To create an environment with a specific version of Python and multiple packages:

```
conda create -n myenv python=3.7 scipy=1.5.0
```

7. Activate the environment using:

```
conda activate myenv
```

8. Save the requirement to a file. The requirements.txt is a list of all the packages in that environment and it allows you to share your dependencies with consumers of your code.

Note: Make sure that you are activated the environment that you just made and use this command to save the file!

```
conda list -e > requirements.txt
```

8. Part to turn-in:

Make an environment with

- 1) jupyter 1.0.0
- 2) python 3.7 and
- 3) scipy 1.5.0

For this environment, don't specify the other package versions, let conda figure that out. **Make a "requirements.txt" file**, put this file in your repo in the main folder (not a sub-folder), and submit it in gradescope under **HW_Unit_01_pt2**. This page is the autograder screen and will take your uploaded repo, run them through an autograder and tell you the grade. It will grade both 1.2 and 1.3 so you'll need to upload them both at the same time to get your total grade. If you need a guide on how to upload into gradescope, see the bottom of this assignment.

1.3. Writing a Shell script (50 points)

For this part, you will be making a shell script that makes the file and folder structure as shown below. Shell scripts are files with lines of shell commands. They are designed to be run at the command line terminal and are a full featured language. Shell scripts are an automated way of programming tedious tasks like repeated sets of commands that need to be performed. For example, you might write one to run a series of python programs every morning or when you deploy a website. You might write one to automatically create a file structure for a project (like a project template).

```
|--s1
|   |--s3
|       |--conf.txt
|   |--s2
|       |--text_chunk1.txt
|       |--Advanced
|           |--text_chunk2.txt
```

- **s1, s2, s3, Advanced** are folder (or directory) names (for example: under the parent folder will be the s1 folder and then the rest of the folder & file structure as shown above)
- Make a file named **conf.txt** that contains the sentence (without the quotes):

```
"virtual (conda) environments are my favorite new technology"
```

- Make a file named **text_chunk1.txt** write a sentence describing what conda environments good for (without quotes) - the sentence needs to start like show below:

```
"virtual environments are good for ..."
```

- **text_chunk2.txt** is a **copy** of text_chunk1.txt with one extra text line appended about why you like virtual environments (without the quotes) - again the sentence needs to start like show below

```
"virtual environments are good for ..."
"I like them because ..."
```

Requirements:

- Write a bash script that generates the above folder and files as shown above and save it as **make_tree.sh**. To do so, you can simply save the commands you use to generate the folder and files in a .sh file.
- **Do NOT use the "cd" command.** For folder and file paths, please use "relative paths" instead of "absolute paths". If you are not sure what that means, this website might help: <https://www.computerhope.com/issues/ch001708.htm>
- text_chunk2.txt needs to be copied from text_chunk1.txt and then append to
- This needs to be done on a Mac/Linux machine because the autograder is on a Linux machine. If you are using a Windows PC, you can login to your virtual machine (as shown in module 1.8.1) and make/test the script there. (you can also install Linux for Windows here: <https://docs.microsoft.com/en-us/windows/wsl/install> but it isn't required)

- Hint: You should try running your script before you submit it to make sure it works before we grade it. There are several ways of executing bash files. Most simply, you can execute them via a command line call like `sh make_tree.sh`. In order to make this file executable, you should:
 - Add this to the top of your file: `#!/bin/bash`

This is called a "shebang" and it points to where your bash scripting is located (in this case bash is in `/bin/bash` which I can see by calling: **which bash** or **type -a bash**)
 - Make the file executable (do this on the command line NOT in the file) : `chmod +x make_tree.sh`

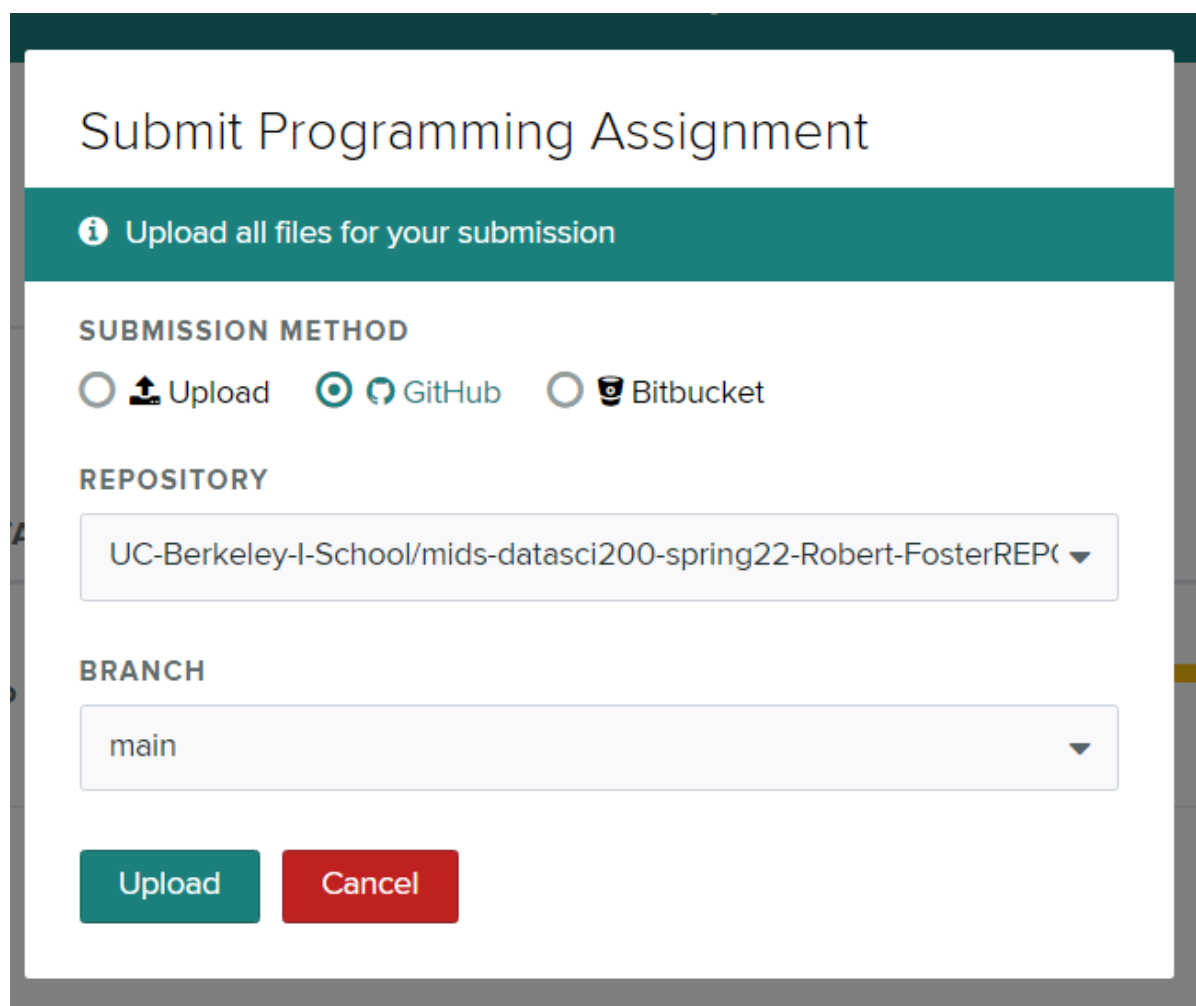
This is modifying the permissions to make the file executable (+x)
- If all of the autograder tests fail , make sure your `make_tree.sh` file runs on your machine. The server tries to run it and if it cannot or runs into an error all of the tests will fail.
- Put this file in the main folder (not a sub-folder) of your github repository.

Uploading files to gradescope

To upload your assignment files to gradescope, click on the assignment name (for this assignment **HW_Unit_01_pt2**).

Uploading from github:

You need to upload your files from github. To do that, link your repo and then select it as shown below:



The screenshot shows the 'Submit Programming Assignment' form in Gradescope. The submission method is set to GitHub. The repository is 'UC-Berkeley-I-School/mids-datasci200-spring22-Robert-FosterREPO' and the branch is 'main'. The 'Upload' button is highlighted in green.

Submit Programming Assignment

Upload all files for your submission

SUBMISSION METHOD

☐ Upload ☒ GitHub ☐ Bitbucket

REPOSITORY

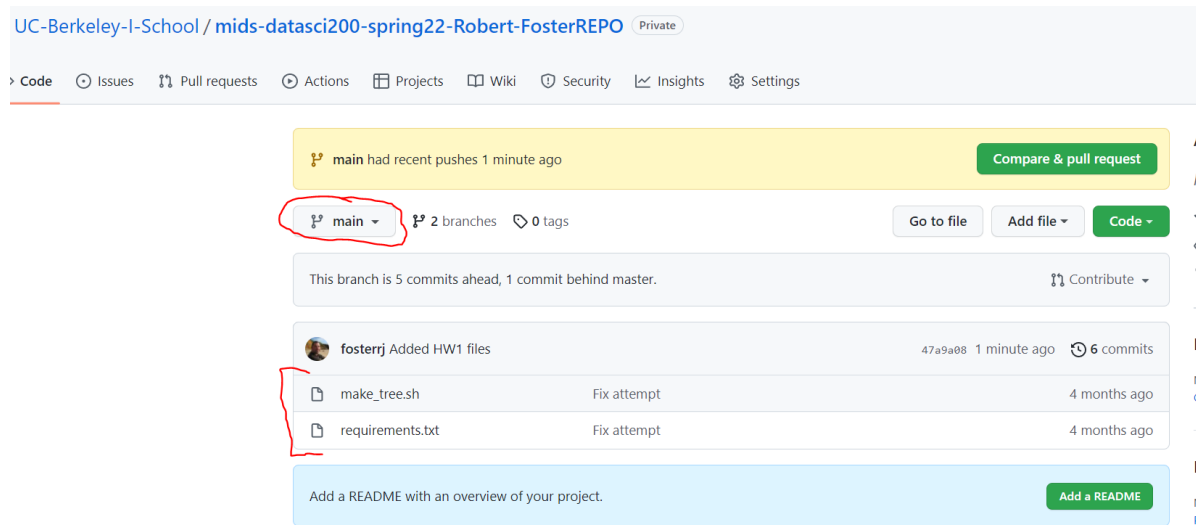
UC-Berkeley-I-School/mids-datasci200-spring22-Robert-FosterREPO ▼

BRANCH

main ▼

Upload **Cancel**

- Make sure its on the 'main' branch in your repo and also in the gradescope (the branches need to match!)
- In order for the autograder to find the files, the files need to be in the main folder of the repo (not in a sub-folder)
- My example repo looks like this:



Click the upload button and the autograder will automatically run:

Submit Programming Assignment

Upload all files for your submission

SUBMISSION METHOD

☒ Upload ☐ GitHub ☐ Bitbucket

REPOSITORY

UC-Berkeley-I-School/mids-datasci200-spring22-Robert-FosterREPO

BRANCH

main

[Upload](#) [Cancel](#)

Feedback

Homework feedback: If you have feedback for this homework, please submit it using the link below:

<http://goo.gl/forms/74yCiQTf6k>