CMSC 201 Section 60
Fall 2021
Project #3 - Using Jupyter Notebooks and Graphics Routines to Analyze Data

---

Value: 80 points

Due Date: December 13, 2021 at 11:59:59 pm

High level description:

This project is different from the previous two in that you will do the programming on your own computer. If you do not have access to a computer that is capable of running Jupyter notebooks, please let Prof. Arsenault know immediately so that arrangements can be made.

Assignment:

Several data files are provided that contain current data from the US States and Territories regarding the current status of COVID-19. Read in those data files, process the data, provide useful graphs that show trends in selected parts of the US, and provide analyses of your graphs.

Details:

You will be doing this project using Jupyter Notebooks. There are several ways to get a Jupyter Notebook on your laptop.

IF YOU HAVE THE PROFESSIONAL EDITION OF PYCHARM ON YOUR LAPTOP, skip Steps 1 and 2 below. To create a new notebook in Pycharm on your laptop, follow the instructions here:

https://www.jetbrains.com/help/pycharm/jupyter-notebook-support.html#get-started

Then go to Step 3.

IF YOU DO NOT HAVE THE PROFESSION EDITION OF PYCHARM, YOU CAN FOLLOW THE STEPS BELOW:

Step 1: Install Jupyter Notebooks on your computer

Go to https://jupyter.org/install.html and follow the instructions there. We recommend that you use pip (The Package Installer for Python) to install the program.

On a Mac, open a terminal window and type

```
pip install notebook
```

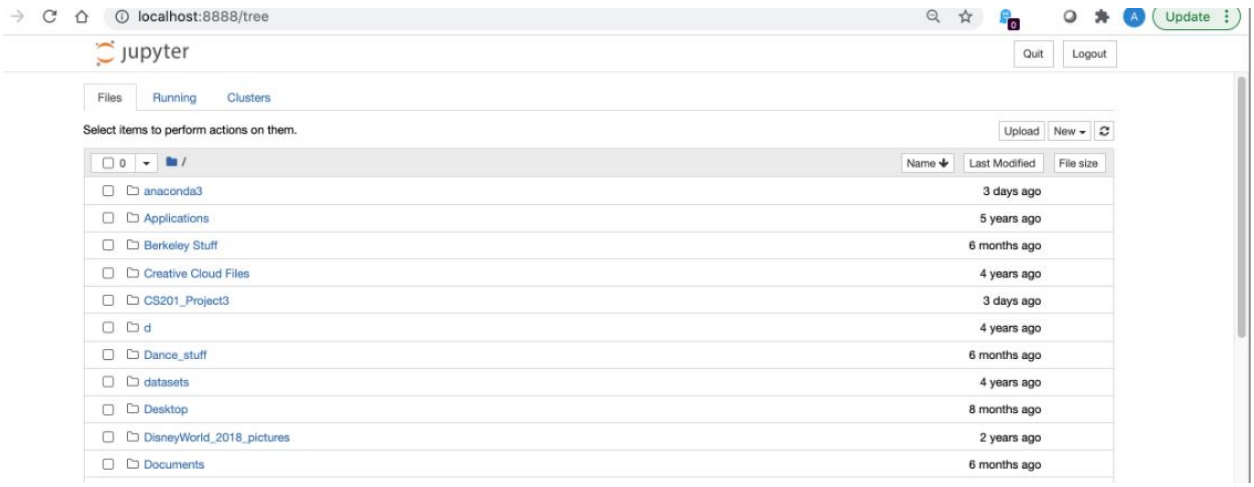
If you have a Windows 10 laptop, open a PowerShell window and type that command. If your laptop is running a version of Windows earlier than version 10, please contact Prof. Arsenault

Step 2: Open a new Jupyter Notebook

In a terminal (Mac) or Powershell (Windows 10) window, type

```
jupyter notebook
```

This should open the Jupyter Notebook application in your default browser. A new tab or window should open. It should looks something like the image below. (This screencap came from a Chrome browser running on MacOS Catalina)



The URL should be “localhost:8888/tree” or similar. This indicates that Jupyter Notebook is running on YOUR laptop; it is not going to some other webserver to get content.

This list of files will be everything that Jupyter Notebook can find on your computer that it thinks can be a Python 3.8-based Notebook. Do not worry if this list is initially empty; it depends on what you’ve done with your computer previously. Regardless, you do not need any previous notebooks.

Step 3: Install plotnine and pandas

You will need at least two Python modules to complete this program – pandas plotnine.

One or both may already be installed on your computer, or they may not be. The simplest way to find out is to create a code cell in a Jupyter notebook, type

import pandas

import plotnine

And run the cell. If it runs, move on to the next step. Otherwise, you have to install it.

Create a new code cell in your notebook and type

```
!pip install pandas
!pip install plotnine
```

Run this cell. After it successfully completes, you can then import pandas and plotnine in your program using the import commands described above.

Pandas documentation is available online at

https://pandas.pydata.org/pandas-docs/stable/index.html

Plotnine documentation is available online at

https://plotnine.readthedocs.io/en/stable/

You should now have all the needed code – except for the code you're going to write - on your laptop computer.

Step 4: Create a new Jupyter notebook.

Launch the Jupyter Notebook application, if it is not already running. In the browser tab that opens, look at the content. Towards the upper right corner of this tab, you will see a button labeled "New." Click this; it will start a new Jupyter Notebook. This is what you need. Select "Python 3" from the list of choices for the type of notebook. A new tab (or new window) should be created in your browser. Congratulations – this is a new, blank, Jupyter notebook and you are ready to begin work.

Step 5: Title

Title this notebook "Project 3" by clicking on the name field and typing "Project 3" when you're asked for a new name.

Click "File', then "Save as" and then give this file a name you will remember on your computer.

Step 6: start your notebook!!

Click in the one cell that shows by default. Click "Cell" on the menu above and click "Type". Then click "Markdown."

Now it's time to write your program heading as Markdown. Here's a tip-sheet on how to do that: **https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html**
Using Markdown, in that cell, write CMSC 201, your section number, Fall 2020, Project 3; the file name; and the date.

Then add another cell below. It should be a code cell. Start by importing the plotnine module that you installed. Then import pandas as pd.

Step 7: Get the data files

The data files for this project consist of 14 days of COVID data for the states and territories of the United States. The original data source is the Center for Systems Science and Engineering at the Johns Hopkins University, who make the data available on their public github repo at https://github.com/CSSEGISandData/COVID-19

You do not need to fetch the files from the JHU site; they have been made available on this class's github repo at https://github.com/Al-Arsenault201/Fall_2021_projects

The data files are all "comma separated value" or "csv" files. Each data item on a row is separated from others by a comma; with each row separated by a newline ('\n\) character. This is a very common file type; it can be produced by Microsoft Excel; Google sheets; or any similar program. Thanks to project 2, you already have the ability to write a very simple Python program to read in this data if you wanted to. But see the next step!!

Step 8: read in the data files

You will need to read in each file to process it.

Add a markdown cell that explains what you're going to do. Then add a code cell and write a function to read in the file and create a 2D list from it.

There are 14 data files. You will need to read in each one and add it to a project "database." Here's good way to do this:

Create an empty, 2D list called

```
database = []

date = 11  # you'll need to append the date to each line of data
```

Let's suppose you have all of the 14 datafiles in a directory on your computer. In my case, the datafiles are in the directory "/Users/alfredarsenault/Documents/Fall_21_project_3_data" on my mac. To read in all the files, I set up a loop like this:

```
import os   # this imports some operating specific calls you'll need

for file in os.listdir("/Users/alfredarsenault/Documents/Spring_21_project_3_data"):

        #os.listdir() is an operating system call that returns a list of all the files in the directory you name.  Replace the string above with the name of the directory on your computer

        print(file)  # this is a debugging line; it prints out the file names. Comment this out or deleting when you're sure that the code is working correctly

        if ".csv" in file:   # ONLY read in the actual data files. There may be some extra stuff in your directory; this guards against accidentally using it. If you're sure there's nothing extra in your directory you can omit this

                with open (file, "r") as infile:
                        infile.readline()   # read and throw away the header line from each file
                        d = infile.read()   # read in each file
                        d_list = d.split('\n') # split each data item into lines
                        for I in range(len(d_list)):   # for every line of data
                                d_list[i] = d_list[i].split(',') #split it on the commas
                                d_list[i].append(date) # add a date column to the row
                                database.append(d_list[i])  # add it to the database

        date += 1  # increment the date for the next file
```

When you're done, print out the first 10 lines of data and the last 10 lines of data to show that your code is working correctly. Just use two separate loops – one loop to print out the first 10 lines of the list; the other loop to print out the last 10 lines of the list

Step 9: create a simpler list

We don't care about all of the data in this field – like, for instance, the latitude and longitude of each state. What we care about are the following columns:

- The state name (column 0)
- The number of confirmed cases (column 5)
- The number of deaths (column 6)
- The incidence rate (column 10)
- The total test results (column 11)
- The case fatality ration (column13)
- The date (column 18; the one you added in step 8)

So write code that creates a new database as a 2D list. Then go through the original database and copy across ONLY those columns from the original that we care about. When you're done, again print out the

first 10 lines of the list and the last 10 lines of this new list to show that it has been copied over correctly.

Step 10: convert your 2D list into a dataframe

Data scientists generally use datastructures called “dataframes” for analysis. Plotnine requires them for graphics, and they are native to pandas.

To turn your 2D list from Step 9 into a dataframe, use the fact that you’ve imported pandas and type

df = pandas.dataframe(list) #replace “list” with the name of the variable you created in Step 9

df.columns = [“State”, “Confirmed”, “Deaths”, “I_rate”, “Tests”, “Fatality_ratio”,”Date”]

this assigns column headings to the dataframe that you will use in your graphics

Step 11: graphical analysis

Now it’s time for some graphical analysis of the data. You already imported the plotnine module in the code cell from Step 6. Now it’s time to use it.

New cells time – create a markdown cell and a code cell.

We’ll start with a simple chart of the states that begin with “A” on the 11th of April. Create a new dataframe where the state begins with “A” – that is, Arizona, Alaska, Alabama, and Arkansas – and the date is the 11th.

Create a new ggplot object and draw a bar chart with the state on the x axis, and the y axis being the number of Confirmed cases of COVID. Your statement should involve something like:

geom-col(aes(x=”State”, y=”Confirmed”))

Then do the same thing to produce a similar graphic for states beginning with “M” on the 16th.

Step 12: trend lines

Next, create a multiline graph showing deaths over the entre 14-day period for the states of Maryland, New York, Michigan, and California. What you want to see is how the cases went up and/or down; and see if all the states were similar or whether some states were different than others.

Step 13: get creative

Produce at least 2 more graphs based on the data. Use markdown cells to explain what you’re showing and why. Use graphics that clearly illustrate your point