CMSC 201 Section 60
Fall 2020
Project #30 - Using Jupyter Notebooks and Graphics Routines to Analyze Data

---

Value: 200 points

Due Date: December 4, 2020 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.

Details:

Step 1: Install Anaconda Navigator on your system

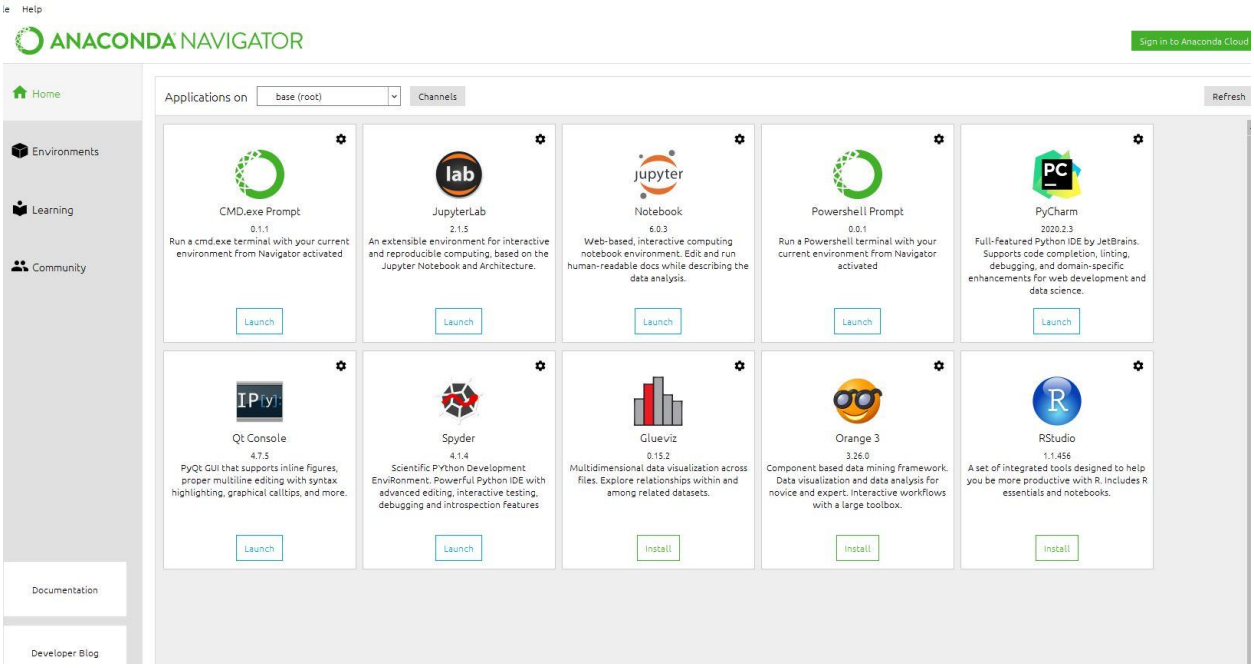
Anaconda Navigator is a desktop GUI that comes with Anaconda Individual Edition. It makes it easy to launch applications and manage packages and environments without using command-line commands. You will need the version of Anaconda Navigator that comes with Anaconda Individual Edition. DO NOT USE THE PROFESSIONAL OR ENTERPRISE EDITIONS.

In your favorite web browser, go to https://www.anaconda.com/products/individual

Download the current version for the operating system on your laptop – Windows, MacOS, or Linux. Make sure to download the current Python 3.8 version. Choose the 64-bit graphical installer as long as your computer will support it. Choose the 32-bit graphical installer only if your laptop can't support the 64-bit installer.

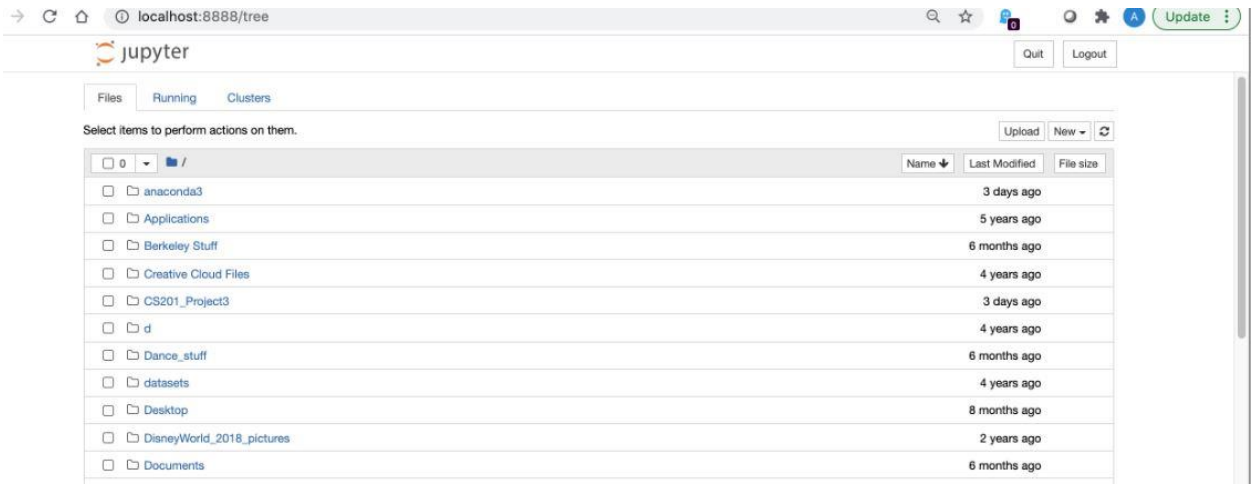
Step 2: Open a new Jupyter Notebook

Open Anaconda Navigator on your laptop. It should look something like the following (this screencap is from the Windows version; MacOS looks slightly different):



Find the box labeled “Jupyter Notebook” (the center application in the top row, in the above screencap), and click the button that says “Launch.”

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 and plotnine.

Pandas – "Panel Data" – is included with Anaconda Navigator. Open Anaconda Navigator and select "Environments" on the left side of the window. This will provide you with a list of all the Python modules included with Anaconda Navigator. Scroll down until you see "pandas" and make sure that it is installed.

Pandas documentation is available online at

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

Plotnine does NOT come with the standard Anaconda Navigator distribution. In the same list of modules where you found Pandas, search for plotnine. Presuming that you do NOT find it, you will have to use the command line to install it.

In a terminal window on MacOS, or in Command tool/PowerShell window on Windows, type

```
conda install -c conda-forge plotnine
```

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 from Anaconda Navigator, 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 file

The data file for Project 3 is in the class github site, as "Project3_data.csv". This file is a list of the results of all General Election races for the United States Senate, from 1976 through 2018. The original source for this file is the Harvard University Dataverse, at https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/PEJ5QU

The file can also be downloaded from Prof. Arsenault's pub directory on gl.umbc.edu at

/afs/umbc.edu/users/a/r/arsenaul/pub/Project3_data.csv.

Make sure that you have this file in a directory accessible to your Jupyter Notebook

This file is a "comma separated value" or "csv" file. It has each data item on a row separated 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. You can, at this time, write a very simple Python program to read in this data.

Step 8: read in the data file

You will need to read in this 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.

Here's the neat part: you don't actually have to do that work this time! Pandas has a "read_csv" command that reads the file and converts for you. Your statement should be something like

dataframe = pd.read_csv("Project3_data.csv")

Now, the entire file exists in the variable 'dataframe'. The problem is that this is not a 2D list like you're used to dealing with; it's a dataframe. So you need to convert it into a 2D list with the statement

```
df = dataframe.values.tolist()
```

"df" is now a python 2D list like you're used to dealing with.

Step 9: Sorting

We'll make a quick diversion into our computer science topics. Sort the 2D list, df, into descending values by candidate votes – that is, the value in column 14. Note: You may NOT use the pre-written pandas.DataFrame.sort_values function –you must write your own sorting routines.

Write a function to implement the selection sort algorithm, as discussed in class. Use it to sort df with the largest values first. Use the python time.time_ns() function to time how long it takes.

Also write a function to implement the quicksort algorithm, as discussed in class. Use it to sort df with the largest values first. Use the python time.time_ns() function to time how long it takes.

Insert a markdown cell to describe your results.

Step 10: Data analysis and visualization

Add another markdown cell that tells what you're going to do and why. Then, add a code cell that prints out the first ten lines of the 2D list you created in Step 8. DO NOT USE THE pandas "head" command to do this; write a loop to do it yourself.

This step is used to verify that you have properly read in the file.

Now, for our purposes, we only care about a few columns and a few rows in this 2D list. Specifically, we care about:

- Year (column 0)
- State post office code (column 2) – the two-letter abbreviation of the state
- Candidate name (column 10)
- Column party (column 11)
- Candidate votes (column 14)

Furthermore, we only carry about Republicans and Democrats, plus two Independents who we know won elections: Bernie Sanders in Vermont and Angus King in Maine.

So, your assignment is to create a new2D list, containing only those rows and the specific columns. Create a new, blank, 2D list. Give it a meaningful name – data_we_care_about or something. Write a loop that goes through each row of the dataframe. If column 10 in that row is 'Bernie Sanders' or 'Angus S. King, Jr.'; OR if column 11 in that row is 'republican' or 'democrat', then create a new row in your 2D list – data_we_care_about or whatever you call it. This new row will contain the data from columns 0, 2, 10, 11 and 14 of the original dataframe.

When you're done, print out the LAST 10 rows of this new list to prove that your code works correctly. Make sure to include a markdown cell to describe what your code does; then put the print code in a new cell. Again, write the loop yourself rather than using "head" or "tail" from Pandas.

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 – markdown and code.

Create a new ggplot object and draw a bar chart with country on the x axis, with the y-axis showing the year of the election, and the y axis showing the number of votes for republican candidates in that year. Your statement should involve something like "geom-col(aex(x="year", 'y=????") (replace the ???? with what you need to get the total number of votes for republican candidates in each year.)

Then repeat this to show the total votes for democratic candidates in each year. Include the votes for Bernie Sanders and Angus King in the democratic vote, since they caucus with Democrats in the Senate even though they're nominally independent.

Step 12: Votes per party, per election over time

Make a line plot(geom_line()) showing the percentage of votes for each party in each election for the states of Maryland, Virginia and Pennsylvania. Put all three of these lines on the same chart!

Step 13: Get creative

Develop three more graphics that provide useful information about election trends. Describe each one in a markdown cell, and then provide and execute python code in code cells.