THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Visualization of Complex Data

DATS 6401

Homework # 2

In this LAB, you will practice different plots from the Matplotlib package. The dataset for this LAB is "CONVENIENT_global_confirmed_cases" which can be found on the course GitHub. The dataset contains the confirmed COVID cases globally from Jan 23rd, 2020, till November 23rd, 2020. Some countries have multiple columns due to different reporting agencies.

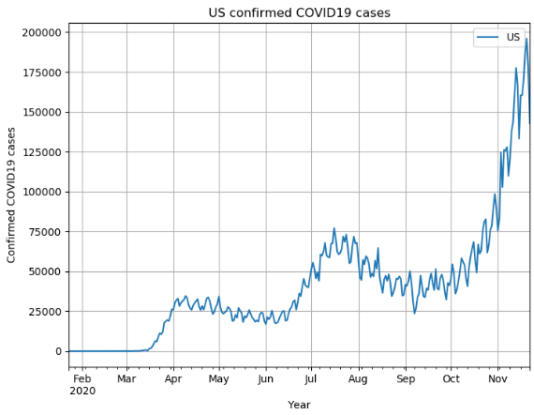
[Except loading the dataset, the rest of questions must be answered using **matplotlib package**]

1. Load the dataset using pandas package. Clean the dataset by removing the 'nan' and missing data.
2. The country "China" has multiple columns ("China.1", "China.2", …). Create a new column name it "China_sum" which contains the sum of "China.1" + "China.2", … column wise. [numbers may not be accurate]

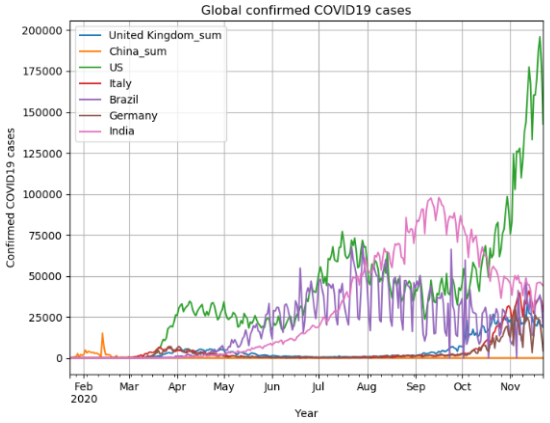
China.1	China.2	China.3	China.4	China.5	China.6
8.0	3.0	4.0	2.0	6.0	3.0
14.0	18.0	5.0	0.0	21.0	18.0
5.0	30.0	8.0	2.0	25.0	0.0
27.0	18.0	17.0	3.0	33.0	13.0
12.0	35.0	24.0	7.0	40.0	10.0
11.0	22.0	21.0	5.0	56.0	5.0

China_sum
95.00000
277.00000
486.00000
669.00000
802.00000
2632.00000

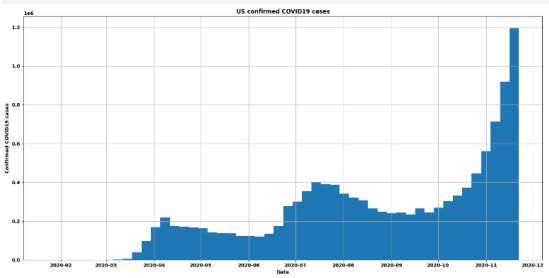
3. Repeat step 2 for the "United Kingdom".
4. Plot the COVID confirmed cases for the following US versus the time. The final plot should look like the following.
5. Repeat step 4 for the "United Kingdom", "China"," Germany"," Brazil", "India" and "Italy".
6. Plot the histogram plot of the graph in Question 4 versus time.
7. Plot the histogram plot of the graph in Question 5 versus time. Use subplot three-by-two. Not a shared axis.
8. Which country (from the list above) has the highest mean, variance and median of # of COVID confirmed cases?



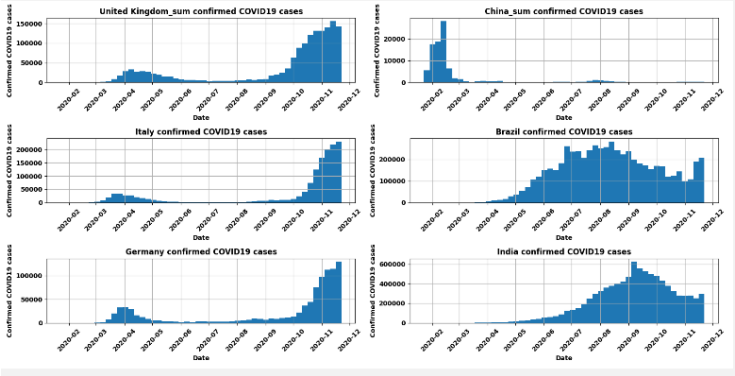
Question 4



Question 5



Question 6



Question 7

The dataset for this section of the LAB will is 'titanic.' To access the 'titanic' dataset you need to connect to the seaborn repository.

The list of features in the dataset and the explanation is as followed:

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1st = Upper, 2nd = Middle, 3rd = Lower
sex	Sex	
Age	Age in years	
sibsp	# of siblings/ spouses abroad the Titanic	
parch	# of parents/ children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	

1- The titanic dataset needs to be cleaned due to nan entries. Remove all the nan in the dataset using "dropna()" method. Show the dataset is cleaned and display the first five rows of the dataset.

2- Develop a python program that plot the pie chart and shows the number of male and female on the titanic dataset. Display the total number of males and females on the console.

3- Develop a python program that plot the pie chart and shows the percentage of male and female on the titanic dataset. Display the percentage of males and females on the console.

4- Develop a python program that plot the pie chart showing the percentage of males who survived versus the percentage of males who did not survive. Display the numbers of the console.

5- Develop a python program that plot the pie chart showing the percentage of females who survived versus the percentage of females who did not survive. Display the numbers of the console.

6- Develop a python program that plots the pie chart showing the percentage passengers with first class, second class and third-class tickets. Display the numbers of the console.

7- Develop a python program that plots the pie chart showing the survival percentage rate based on the ticket class. Display the numbers of the console.

8- Develop a python program that plots the pie charts showing the percentage passengers who survived versus the percentage of passengers who did not survive with the first-class, second- and third-class ticket category. Display the numbers of the console.

9- Using the matplotlib and plt.subplots [3x3] create a dashboard which includes all the pie charts above. Note: Use the figure size = (16,8).

All the figures should have the appropriate title and legend with the correct label.

Submission guidelines:

- The softcopy of the developed Python code .py must also be submitted separately. Please make sure the developed python code runs without any error by evaluating it through PyCharm software. The developed python code with any error will subject to 50% points penalty.
- Add an appropriate x-label, y-label, legend, and title to each graph.
- Develop a report and answer all the above questions. Include the required graphs in your report.
- Submission: report (pdf format) + .py file. The python file is a supporting file and will not replace the solution. A report that includes the solution to all questions is required and will be graded only.
- The python file must regenerate the provided results inside the report.