6401: Visualization of Complex Data

Spring 2022

Instructor: R. Jafari, Ph.D

Lab #3

Student: Lydia Teinfalt

Initials: LT

Last updated: February 24, 2022

Contents

Abstract 4

Introduction 4

Methodology 5

Dataset 5

Lab Answers 6

Part 1 6

Question 1 6

Question 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” 7

Question 3: Repeat step 2 for the “United Kingdom” 7

Question 4: Plot the COVID confirmed cases for the following US versus the time. 8

Question 5: Repeat step 4 for the “United Kingdom”, “China”, “Germany”, ”Brazil”, “India” and “Italy” 9

Question 6: Plot the histogram plot of the graph in Question 4 10

Question 7: Plot the histogram plot of the graph in Question 5 11

Question 8: Which country (from the list above) has the highest mean, variance and median of # of COVID confirmed cases? 11

Part 2 12

Question 2: Write a python program that plot the pie chart and shows the number of male and female on the titanic dataset 13

Question 3: Write a python program that plot the pie chart and shows the percentage of male and female on the titanic dataset 13

Question 4: Write a python program that plot the pie chart showing the percentage of males who survived versus the percentage of males who did not survive 14

Question 5: Write a python program that plot the pie chart showing the percentage of females who survived versus the percentage of females who did not survive 14

Question 6: Write a python program that plot the pie chart showing the percentage passengers with first class, second class and third-class tickets 15

Question 7: Write a python program that plot the pie chart showing the survival percentage rate based on the ticket class. The final answer should look like bellow. 15

Question 8: Write a python program that plot the pie chart showing the percentage passengers who survived versus the percentage of passengers who did not survive with the first-class ticket category. The final answer should look like bellow. 16

Question 9: Write a python program that plot the pie chart showing the percentage passengers who survived versus the percentage of passengers who did not survive with the second-class ticket category. The final answer should look like bellow. 16

Question 10: Write a python program that plot the pie chart showing the percentage passengers who survived versus the percentage of passengers who did not survive in the third-class ticket category. 17

Question 11: Using the matplotlib and plt.subplots create a dashboard which includes all the pie charts above. Note: Use the figure size = (16,8). The final answer should look like the following. ... 17

Conclusion 18

Appendix A: Code 19

References 26

Abstract

First part of the lab examined COVID19 cases globally in 2020. The dataset was provided showing daily confirmed COVID19 cases by country in 2020. Analysis and visualization focused on the following countries: US, United Kingdom, China, Germany, Brazil, India, and Italy and using line plots.

Next, we filtered a subset of Seaborn library's titanic dataset to explore the data in terms of on gender, survival, and ticket class and created visualization of the analysis using pie charts using Matplotlib package.

Introduction

The first half of the lab analyzed data of confirmed COVID19 cases during the pandemic in 2020. The data showed the daily number of COVID19 infections by country and state/province. Line plots showing rising number of infections were created using Matplotlib from January 23 – November 22,2020. A sample of countries were selected in this comparison and to answer which country had the highest mean, median, and variance of confirmed COVID19 cases. The countries considered in this lab were: US, United Kingdom, China, Germany, Brazil, India, and Italy. Analysis and visualizations were conducted by aggregating state /province level data up to country level.

The mean is the average value found in a sample and is calculated using a formula displayed in Figure 1 where you sum up all values in the sample and divide it by the total number of values (via StatisticsHowTo).

$$\bar{x} = (\Sigma x_i) / n$$

Figure 1: Sample Mean

The variance tells you how spread out the data is from the mean in the sample and can be calculated per Figure 2 by taking the square of standard deviation of a random variable (Wikipedia):

$$\frac{1}{n-1}\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2$$

Figure 2: Variance

The median reflects the middle value of an ordered list. If the data x's values is sorted in numerical order with n elements, then median can be calculated with the following formula in Figure 3 (Wikipedia):

$$\text{if } n \text{ is odd, } \operatorname{median}(x) = x_{(n+1)/2}$$

$$\text{if } n \text{ is even, } \operatorname{median}(x) = \frac{x_{(n/2)} + x_{(n/2)+1}}{2}$$

Figure 3: Median

In the second half of the lab utilized Titanic dataset provided by Seaborn. A subset of Titanic dataset was constructed based on gender, survival, and ticket class. The number of males versus female passengers,

the percentage of male/female passengers versus total number of passengers, the percentage of male/female passengers' survival rate, percentage of passengers holding first, second- and third-class tickets, and survival rate of passengers for each ticket class. Tickets from class 1 was the most expensive and class 3 was least expensive. Pie charts built using matplotlib library were used to visualize the findings.

Methodology

Dataset

The first dataset was obtained from Professor R. Jafari's Complex Visualization repository on GitHub called "CONVENIENT_global_confirmed_cases.csv". The Pandas package read_csv method was given the URL directly to the data's raw format from the repository. This dataset contained 306 rows and 270 columns. The first row of the dataset was used as the dataframe's headers. The second row of the dataset was also a secondary header containing names of provinces/states of each country. We were asked to clean the dataset by removing any nan and missing data. This only occurred in the second header row for province and state information because many countries did not have this information, so row with names of province/state was removed because it was not needed in this analysis.

The first column the reporting date in the format eg "1/23/20" for first day, to the last day "11/22/20" for confirmed COVID19 cases. The state / province secondary header row was not pertinent to this lab because we were looking at country-level data. The rest of columns represented country names in alphabetical order starting from Afghanistan to Zimbabwe.

To create a line graph of number of COVID19 cases versus time for the following countries: US, United Kingdom, China, Germany, Brazil, India, and Italy. China and United Kingdom's numbers were further broken down by state / province, so we created a new column called "China_sum" and "United Kingdom_sum" to combining all the individual columns of each country. China's provincial data encompassed column indices 58 – 89. United Kingdom had column indices 249-259. The rest of the countries US, Germany, Brazil, India, and Italy did not state or provincial level data, so no aggregation was necessary. A new date column was created based on the first column labelled "Country/Region" using pandas method to_datetime() method.

1	Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Australia	Australia	Australia	Australia
2	Province/State	nan	nan	nan	nan	nan	nan	nan	nan	Australian Capital Territory	New South Wales	Northern Territory	Queensland	South Australia
3	1/23/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1/24/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1/25/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	1/26/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0

The second part of the lab loaded at the Titanic dataset from the Seaborn library using the following Python command:

```
import seaborn as sns
df = sns.load_dataset('titanic')
```

We examined explored Titanic passenger survival rate based on gender and/or ticket class with column names ‘sex’, ‘survived’, and ‘pclass’. The raw data has 182 rows and 15 columns.

	survived	pclass	sex	age	...	deck	embark_town	alive	alone
1	1	1	female	38.0	...	C	Cherbourg	yes	False
3	1	1	female	35.0	...	C	Southampton	yes	False
6	0	1	male	54.0	...	E	Southampton	no	True
10	1	3	female	4.0	...	G	Southampton	yes	False
11	1	1	female	58.0	...	C	Southampton	yes	True

Survived has 1 if the passenger lived through the tragedy and 0 if passenger died. The pclass column is the ticket class and can have values 1,2, and 3. Tickets prices are in descending from class 1 to class 3.

	sex	survived	pclass
1	female	1	1
3	female	1	1
6	male	0	1
10	female	1	3
11	female	1	1

Several pie charts were constructed using matplotlib comparing of number of males versus female passengers, the percentage of male/female passengers versus total number of passengers, the percentage of male/female passengers’ survival rate, percentage of passengers holding first, second- and third-class tickets, and survival rate of passengers for each ticket class.

Lab Answers

Part 1

Question 1

Load the dataset using Pandas. Clean the dataset by removing the ‘nan’ and missing data.

	Country/Region	Afghanistan	Albania	...	Yemen	Zambia	Zimbabwe
301	11/18/20	223.0	711.0	...	2.0	37.0	36.0
302	11/19/20	377.0	786.0	...	3.0	70.0	65.0
303	11/20/20	215.0	836.0	...	4.0	23.0	74.0
304	11/21/20	60.0	737.0	...	3.0	21.0	52.0
305	11/22/20	203.0	565.0	...	6.0	30.0	48.0

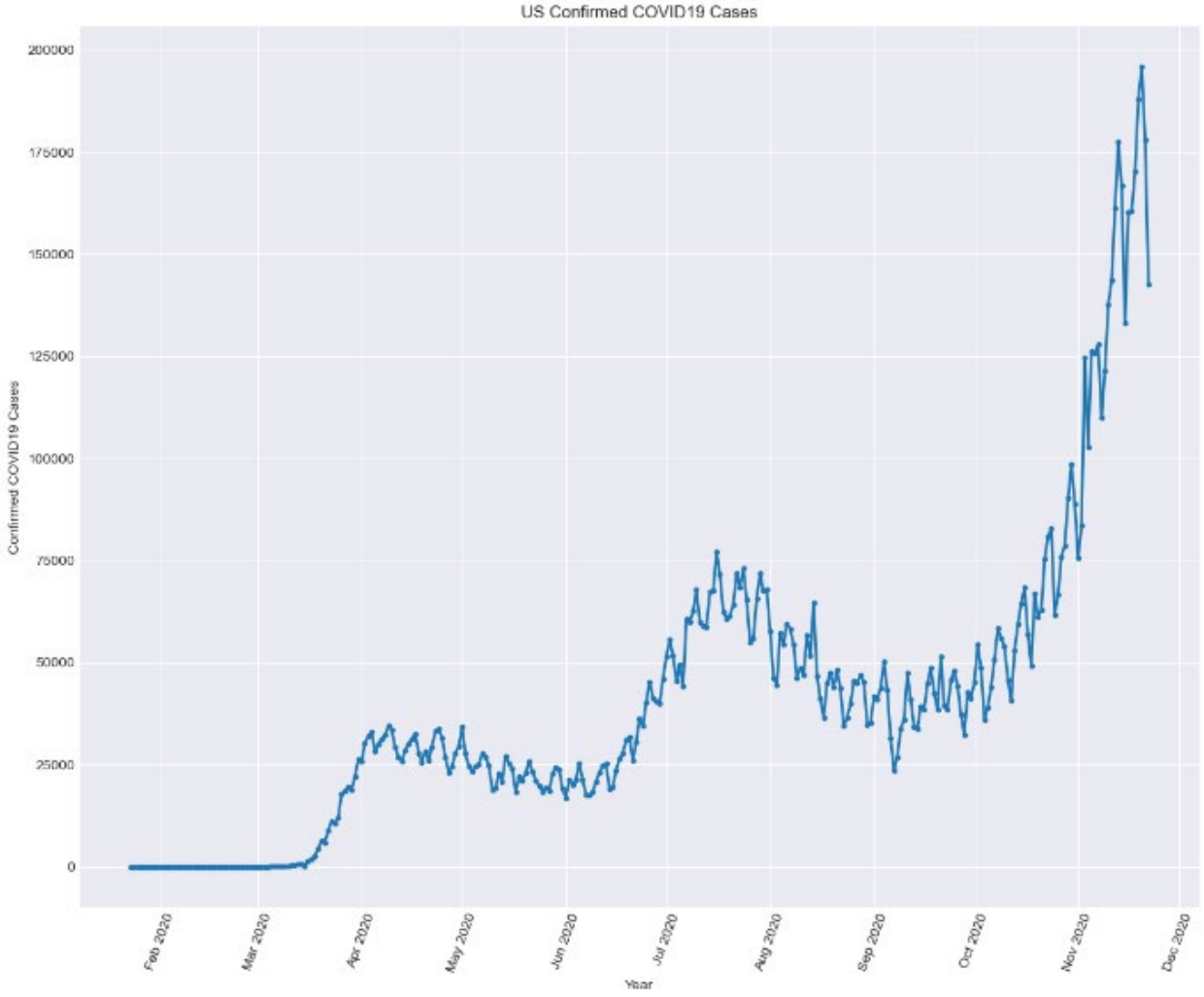
Question 2: The country “China” has multiple columns (“ China.1”, “China.2”, …) . Create a new column name it “China_sum” which contains the sum of “China.1” + “China.2”

Date	China_sum
2020-01-23 00:00:00	78.00000
2020-01-24 00:00:00	261.00000
2020-01-25 00:00:00	467.00000
2020-01-26 00:00:00	627.00000
2020-01-27 00:00:00	778.00000
2020-01-28 00:00:00	2587.00000

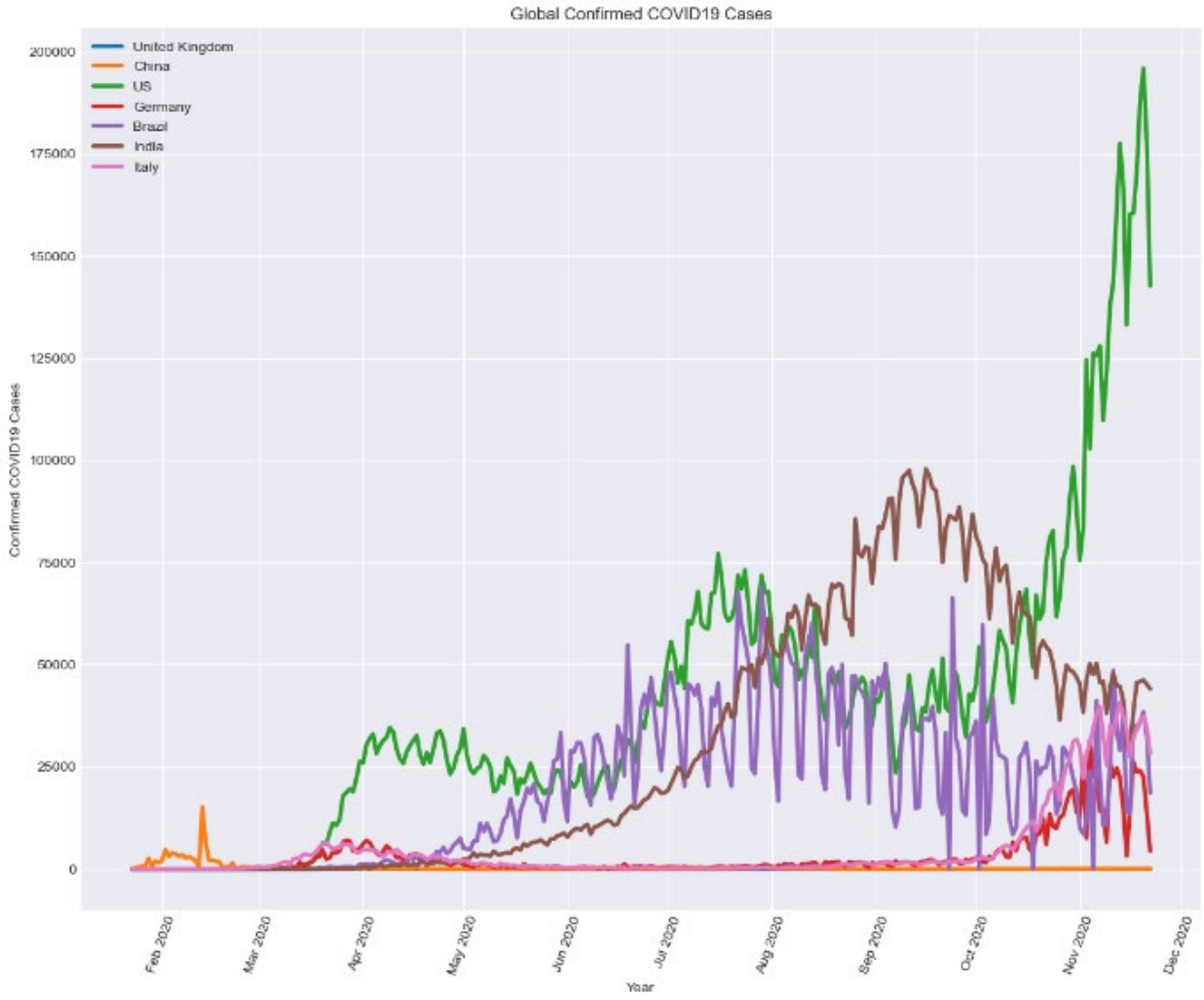
Question 3: Repeat step 2 for the “United Kingdom”

Date	China_sum	United Kingdom_sum
2020-01-23 00:00:00	78.00000	0.00000
2020-01-24 00:00:00	261.00000	0.00000
2020-01-25 00:00:00	467.00000	0.00000
2020-01-26 00:00:00	627.00000	0.00000
2020-01-27 00:00:00	778.00000	0.00000
2020-01-28 00:00:00	2587.00000	0.00000

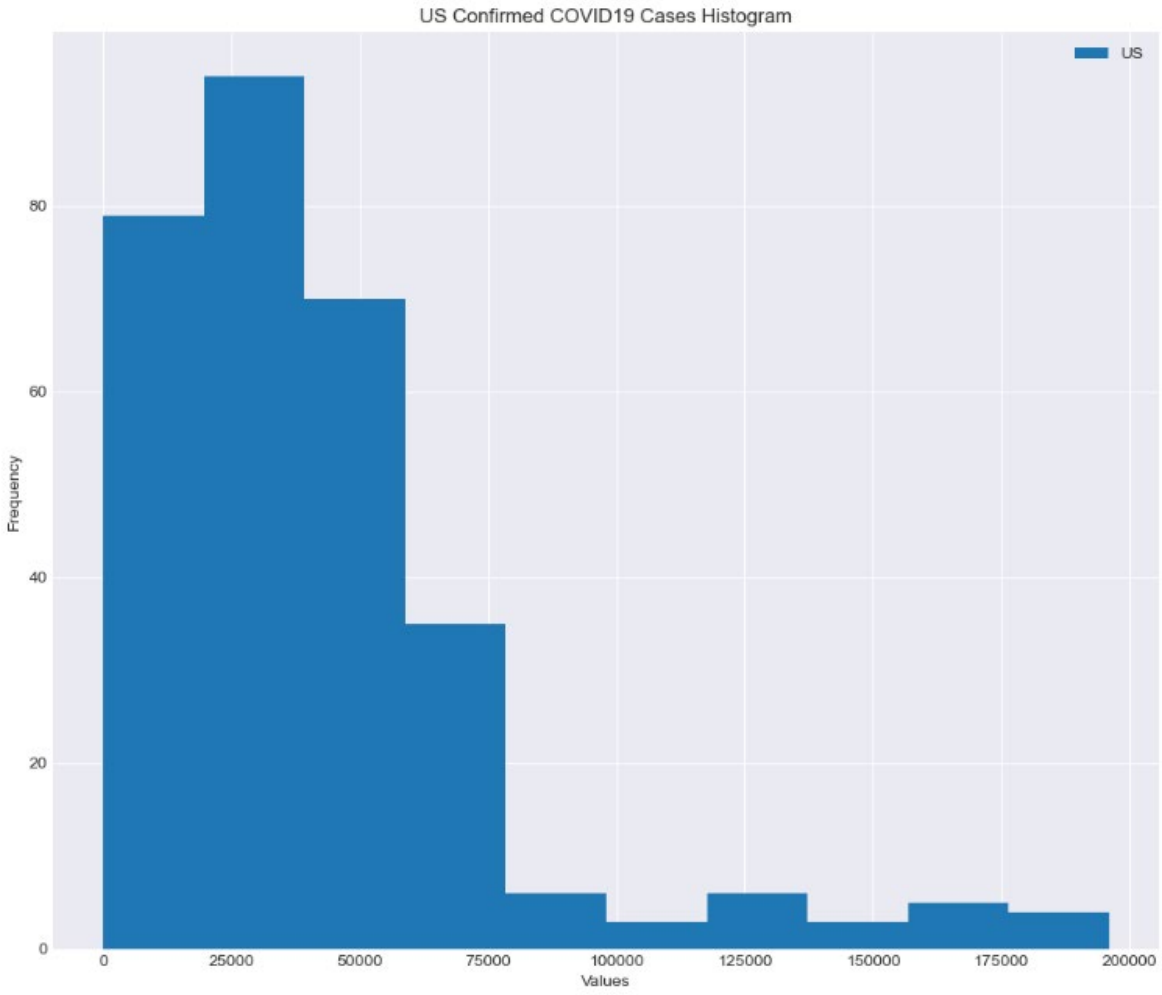
Question 4: Plot the COVID confirmed cases for the following US versus the time.



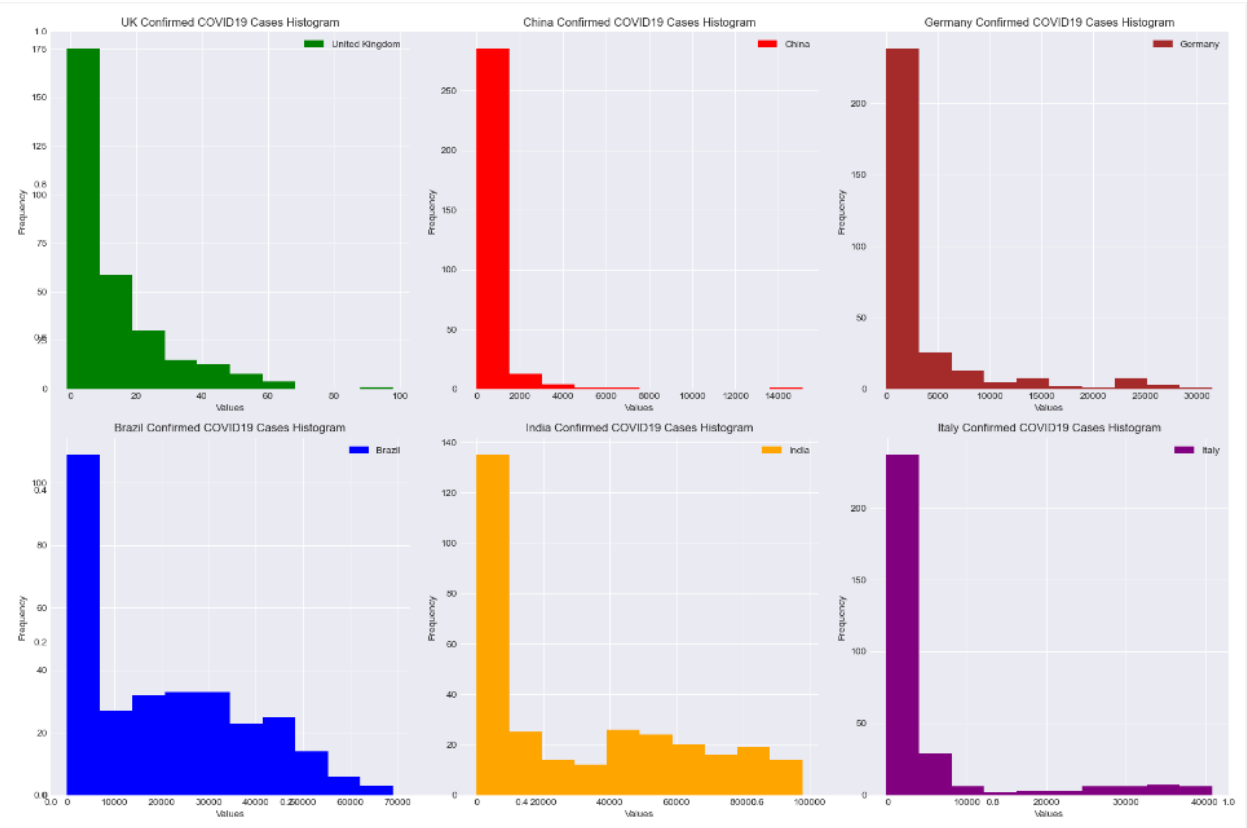
Question 5: Repeat step 4 for the “United Kingdom”, “China”, “Germany”, ”Brazil”, “India” and “Italy”



Question 6: Plot the histogram plot of the graph in Question 4



Question 7: Plot the histogram plot of the graph in Question 5



Question 8: Which country (from the list above) has the highest mean, variance and median of # of COVID confirmed cases?

Highest mean, variance and median of number of COVID confirmed cases was US

```
Mean Confirmed COVID19 Cases
United Kingdom_sum       12.318033
China_sum               296.016393
Germany                3056.940984
Italy                  4619.239344
Brazil                19906.232787
India                 29966.770492
US                    40153.600000
dtype: float64
US had highest mean
India had highest mean
```

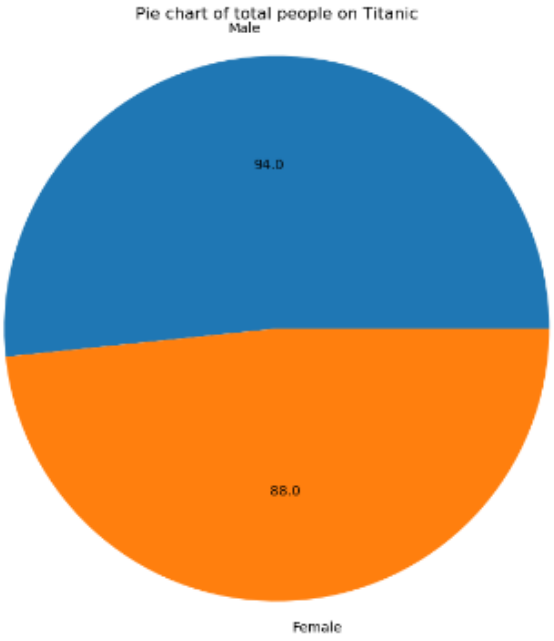
```
Variance Confirmed COVID19 Cases
United Kingdom_sum    2.366189e+02
China_sum             1.283579e+06
Germany               3.038726e+07
Italy                 8.493878e+07
Brazil                3.282161e+08
India                 9.775760e+08
US                    1.350611e+09
dtype: float64
US had highest variance
India had second highest variance

Median Confirmed COVID19 Cases
United Kingdom_sum        7.0
China_sum                31.0
Germany                 988.0
Italy                  1008.0
India                 15968.0
Brazil                16517.0
US                    33485.0
dtype: float64
US had highest median
Brazil had highest median
```

Part 2

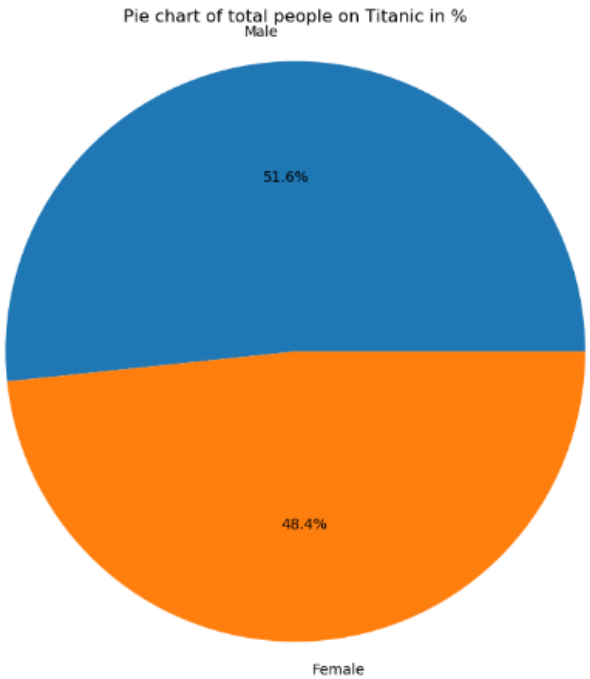
Question 1: The titanic dataset needs to be cleaned due to nan entries. Remove all the nan in the dataset using ""dropna()" method. Display the first 5 row of the dataset.

	survived	pclass	sex	age	...	deck	embark_town	alive	alone
1	1	1	female	38.0	...	C	Cherbourg	yes	False
3	1	1	female	35.0	...	C	Southampton	yes	False
6	0	1	male	54.0	...	E	Southampton	no	True
10	1	3	female	4.0	...	G	Southampton	yes	False
11	1	1	female	58.0	...	C	Southampton	yes	True

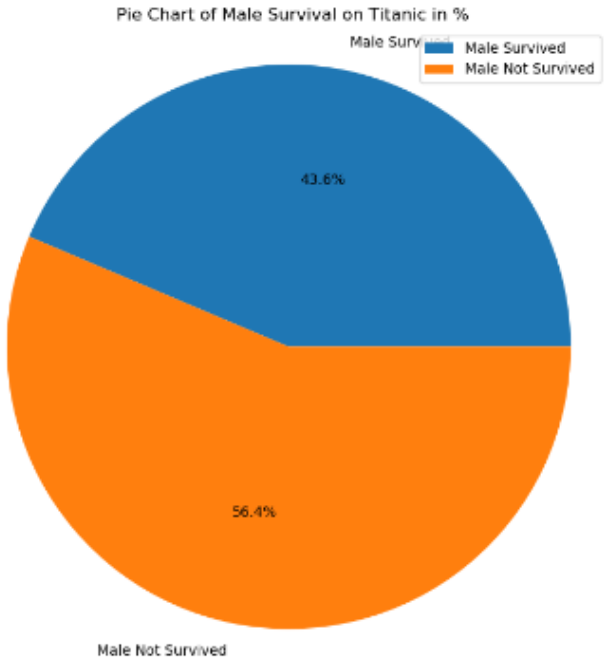
Question 2: Write a python program that plot the pie chart and shows the number of male and female on the titanic dataset



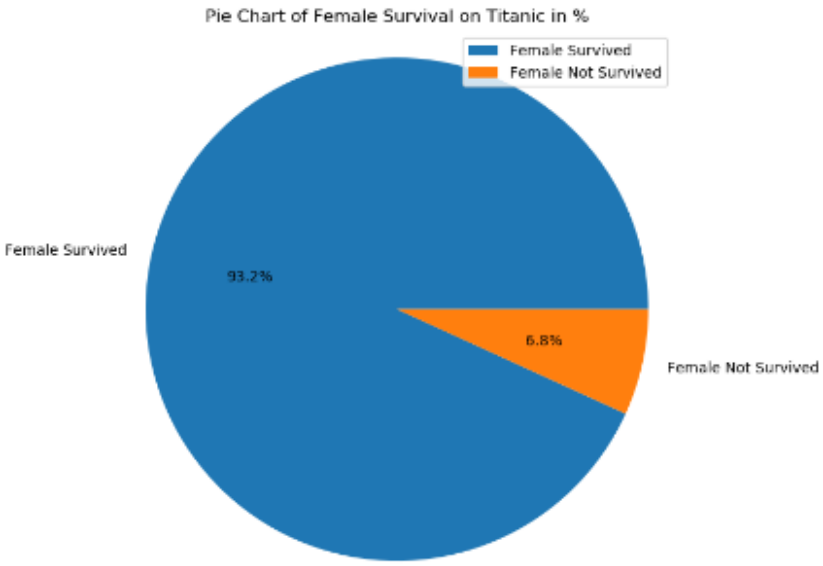
Question 3: Write a python program that plot the pie chart and shows the percentage of male and female on the titanic dataset



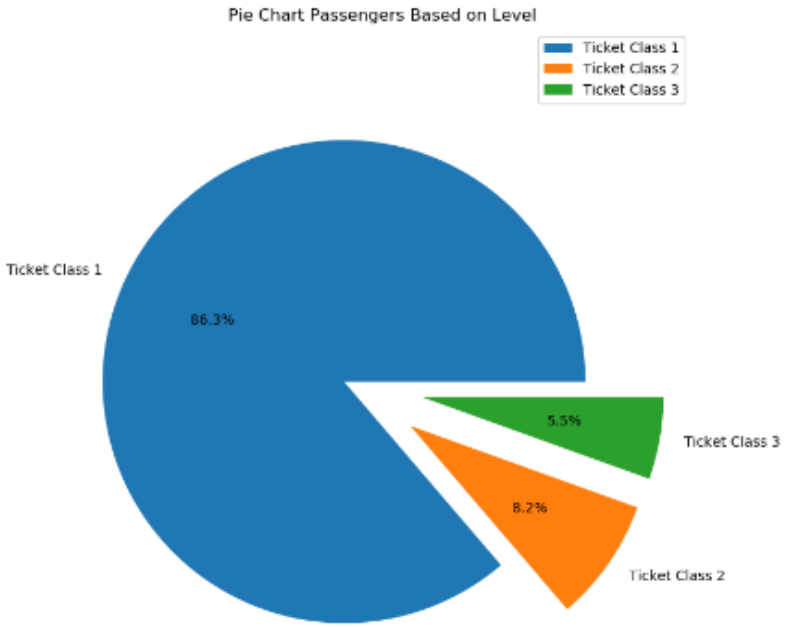
Question 4: Write a python program that plot the pie chart showing the percentage of males who survived versus the percentage of males who did not survive.



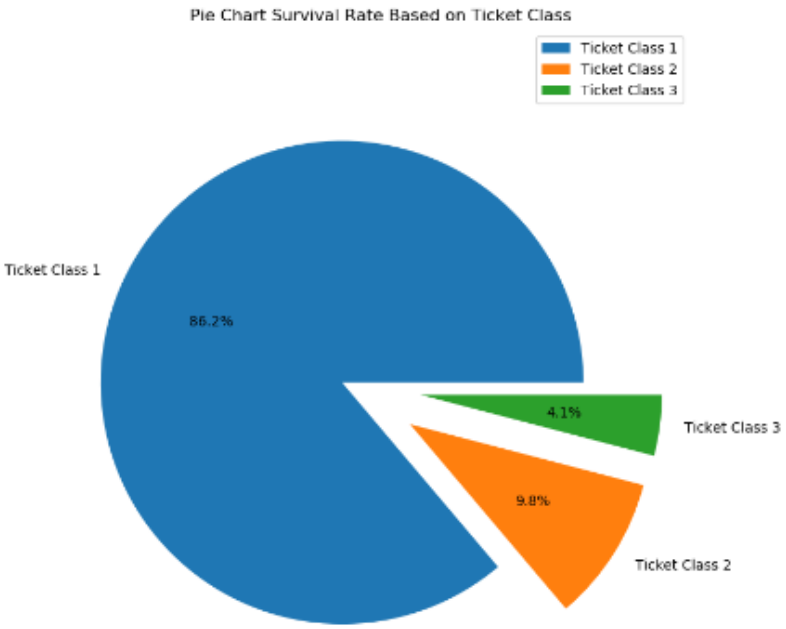
Question 5: Write a python program that plot the pie chart showing the percentage of females who survived versus the percentage of females who did not survive



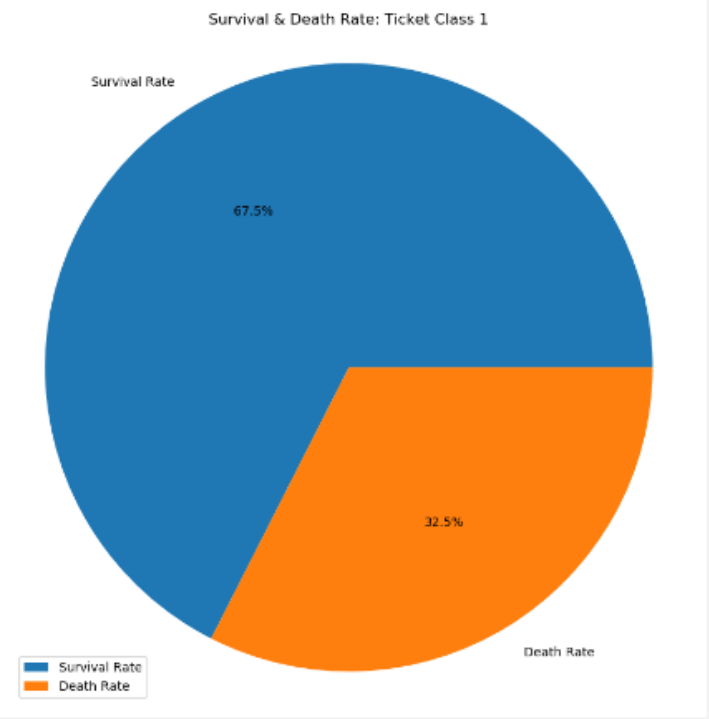
Question 6: Write a python program that plot the pie chart showing the percentage passengers with first class, second class and third-class tickets



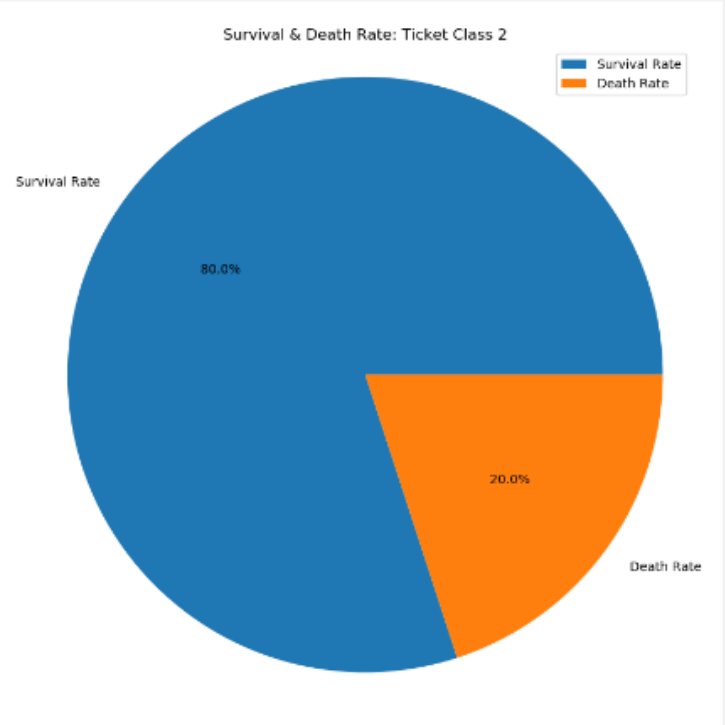
Question 7: Write a python program that plot the pie chart showing the survival percentage rate based on the ticket class. The final answer should look like bellow.



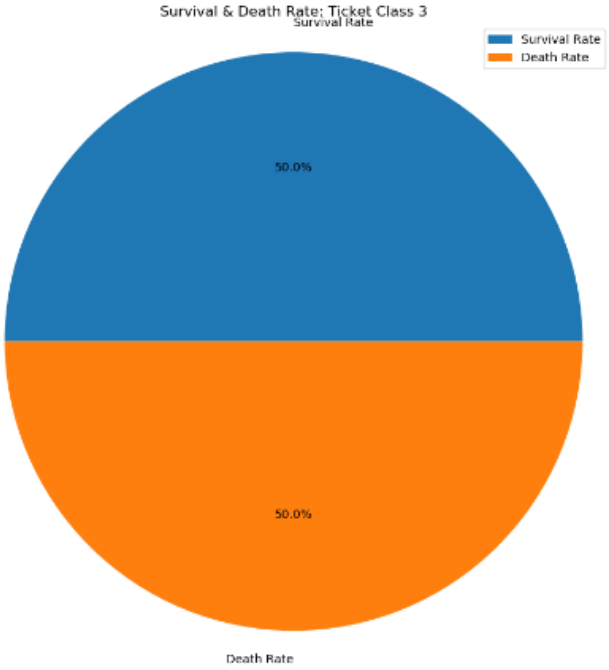
Question 8: Write a python program that plot the pie chart showing the percentage passengers who survived versus the percentage of passengers who did not survive with the first-class ticket category. The final answer should look like bellow.



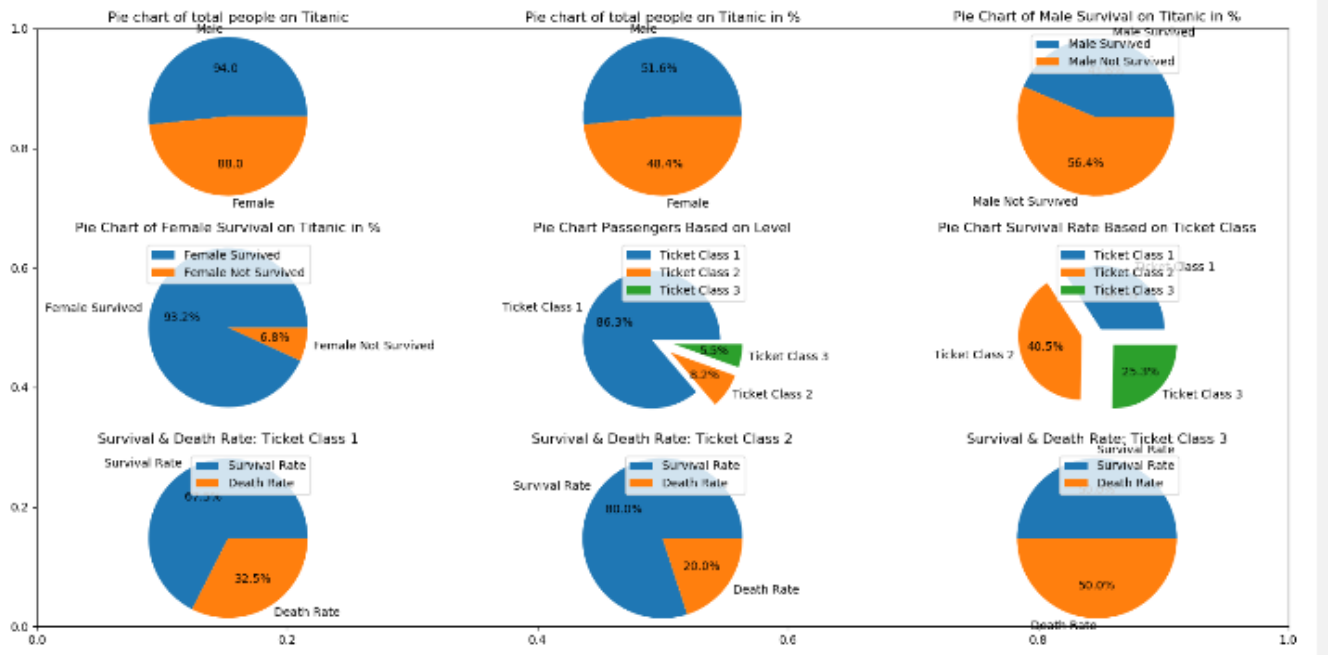
Question 9: Write a python program that plot the pie chart showing the percentage passengers who survived versus the percentage of passengers who did not survive with the second-class ticket category. The final answer should look like bellow.



Question 10: Write a python program that plot the pie chart showing the percentage passengers who survived versus the percentage of passengers who did not survive in the third-class ticket category.



Question 11: Using the matplotlib and plt.subplots create a dashboard which includes all the pie charts above. Note: Use the figure size = (16,8). The final answer should look like the following.



Conclusion

The figure showing confirmed COVID19 cases in the US shows that starting mid-March marks the beginning of the pandemic with increase of infections having first spike of 25,000 by beginning of April. From day to day, the COVID19 cases had minor fluctuations up and down but month to month but overall, it was increasing trend. By mid-summer in July in the US there was a new peak with values three times as much Spring with ~ 75,000 cases. Between September and October, there was a slight plateau in the numbers. However, after October through end of November, the number of confirmed COVID19 cases rose exponentially. There were approximately 50,000 cases beginning of October and by the end of the reporting date, November 23, there were close to 2 million confirmed cases in the US.

The plot combining results for United Kingdom, China, US, Germany, Brazil, India, and Italy. The highest number of cases were in the US followed by India and Brazil. This is confirmed when we took statistics of the confirmed COVID19 cases. US had the highest mean of 40153.60 and variance with 1.35e^9, India had the second highest mean of 29966.77 and variance 9.78e^08. In terms of the median, US was highest with value of 33485 and Brazil was second highest with value of 16517. United Kingdom had the lowest mean at 12.32 and median value of 7. China was second lowest at 296.02 mean and 31 for median. Looking at the COVID19 vs time, different countries saw COVID19 infections occurring in waves. Brazil's numbers peaked from mid-July to August. India's peak occurred from September to October and US had several waves but dramatic rise from October to November. The histogram plots show that distribution is not normal for the countries we examined.

For the titanic dataset, the pie charts were employed to look at categorical data -- gender, survival, and ticket class. There were 94 (51.6%) men and 88(48.4%) women on the Titanic. This is surprising because but perhaps it is not counting crew which would have presumably been all or predominantly men. Looking at survival from the point of view of gender, higher percentage of men died rather than survive. In contrast, 93.2% of female passengers survived versus 6.8% who died. This shows that more women than men survived the Titanic and is consistent with the social mores of the time. In terms of ticket class, great majority of passengers holding class 1 tickets survived, 86.3% whereas only 8.2% of second-class passengers survived and 5.5% of third class. These results make sense with the first-class ticket passengers located on the top level of the ship, the second-class ticket passenger located in the middle and the third class in the bottom of the ship. When the Titanic hit the iceberg, water would have started filling up the ship from bottom to up. Higher percentage of ticket class 2 passengers (80%) survived versus 67.75% of first-class ticket. For third-class ticket passengers, there was a 50% survival rate.

Appendix A: Code

```
#Lydia Teinfalt
#Lab 3
#DATS 6450 Spring '22
#02/23/2022

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
plt.style.use('seaborn-darkgrid')

df = pd.read_csv("https://raw.githubusercontent.com/rjafari979/Complex-Data-
Visualization-/main/CONVENIENT_global_confirmed_cases.csv")
print(df.describe())
print(df.head(10))
print(df.tail())
print("Shape before dropping NA from dataframe")
print(f"rows = {df.shape[0]}")
print(f"columns = {df.shape[1]}")

#First row of dataframe is State/Province
df1 = df.iloc[0]
dfnull = df.iloc[1:-1].isnull().values.any()
if dfnull:
    df = df.dropna(df.iloc[1:-1].isnull(), axis=0, inplace=True)

columns = df.columns

df1 = df.copy()
dfnull = df.isnull().values
df1 = df1.loc[1:]
df1['Date'] = pd.to_datetime(df1['Country/Region'])
countries = ['China', "United Kingdom", "Germany", "Brazil", "India",
"Italy"]

for i in countries:
    col_names = df.columns[df.columns.str.startswith(i)]
    idx = [df.columns.get_loc(col) for col in col_names]
    # print(idx)
    new_column= i+"_sum"
    first_index = idx[0]
    last_index = idx[-1]
#china 57:89
#united Kingdom 249:259


df1['China_sum'] = df1.iloc[:,57:89].astype(float).sum(axis=1)
df1['United Kingdom_sum'] = df1.iloc[:,249:259].astype(float).sum(axis=1)

#us covid cases - Line plot
fig, ax = plt.subplots(figsize=(12,10))
ax.plot(df1['Date'], df1['US'], linewidth = 2, marker = '.')
ax.grid()
ax.set_ylabel("Confirmed COVID19 Cases")
plt.xticks(rotation=70)
```

```
ax.set_xlabel("Year")
ax.set_title("US Confirmed COVID19 Cases")
ax.xaxis.set_major_locator(mdates.MonthLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %Y'))
plt.grid()
plt.tight_layout()
plt.show()

#Question 5 Global Confirmed COVID19 Cases
#"United Kingdom", "China", "Germany", "Brazil", "India" and "Italy"
fig, ax = plt.subplots(figsize=(12,10))
ax.plot(df1['Date'], df1['United Kingdom_sum'], label = 'United Kingdom',
linewidth = 3)
ax.plot(df1['Date'], df1['China_sum'], label = 'China',linewidth = 3)
ax.plot(df1['Date'], df1['US'], label = 'US',linewidth = 3)
ax.plot(df1['Date'], df1['Germany'], label = 'Germany', linewidth = 3)
ax.plot(df1['Date'], df1['Brazil'], label = 'Brazil', linewidth = 3)
ax.plot(df1['Date'], df1['India'], label = 'India', linewidth = 3)
ax.plot(df1['Date'], df1['Italy'], label = 'Italy', linewidth = 3)
ax.grid()
ax.set_ylabel("Confirmed COVID19 Cases")
plt.legend()
plt.xticks(rotation=70)
ax.set_xlabel("Year")
ax.set_title("Global Confirmed COVID19 Cases")
ax.xaxis.set_major_locator(mdates.MonthLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %Y'))
plt.grid()
plt.tight_layout()
plt.show()



#Question 6 Plot histogram for US Covid19 Cases
fig, ax = plt.subplots(figsize=(12,10))
us = df1['US']
ax.hist(df1['US'], label = 'US')
ax.set_ylabel("Frequency")
ax.legend()
ax.grid()
ax.set_xlabel("Values")
ax.set_title("US Confirmed COVID19 Cases Histogram")
plt.grid()
plt.show()


#"United Kingdom", "China", "Germany", "Brazil", "India" and "Italy"
fig, ax = plt.subplots(figsize=(18,12))
plt.grid()
ax = fig.add_subplot(2,3,1)
ax.hist(df1['United Kingdom_sum'], label ='United Kingdom', color = 'green' )
ax.set_title("UK Confirmed COVID19 Cases Histogram")
ax.set_ylabel("Frequency")
ax.set_xlabel("Values")
ax.legend()
ax = fig.add_subplot(2,3,2)
ax.hist(df1['China_sum'], label = 'China', color = 'red')
```

```
ax.set_title("China Confirmed COVID19 Cases Histogram")
ax.set_ylabel("Frequency")
ax.set_xlabel("Values")
ax.legend()
ax = fig.add_subplot(2,3,3)
ax.hist(df1['Germany'], label = 'Germany', color = 'brown')
ax.set_title("Germany Confirmed COVID19 Cases Histogram")
ax.set_ylabel("Frequency")
ax.set_xlabel("Values")
ax.legend()
ax = fig.add_subplot(2,3,4)
ax.hist(df1['Brazil'], label = 'Brazil', color = 'blue')
ax.set_title("Brazil Confirmed COVID19 Cases Histogram")
ax.set_ylabel("Frequency")
ax.set_xlabel("Values")
ax.legend()
ax = fig.add_subplot(2,3,5)
ax.hist(df1['India'], label = 'India', color = 'orange')
ax.set_title("India Confirmed COVID19 Cases Histogram")
ax.set_ylabel("Frequency")
ax.set_xlabel("Values")
ax.legend()
ax = fig.add_subplot(2,3,6)
ax.hist(df1['Italy'], label = 'Italy', color = 'purple')
ax.set_title("Italy Confirmed COVID19 Cases Histogram")
ax.set_ylabel("Frequency")
ax.set_xlabel("Values")
plt.legend()
plt.tight_layout()
plt.show()

#Which country (from the list above) has the highest mean, variance and
median of # of COVID confirmed cases?
df_mean = df1[['China_sum', 'United Kingdom_sum', 'US', 'Germany', 'Brazil',
'India', 'Italy']].mean()
df_mean.columns = "Mean"
df_mean = df_mean.sort_values()
print("Mean Confirmed COVID19 Cases")
print(df_mean)
print("US had highest mean")
print("India had highest mean")

df_var = df1[['China_sum', 'United Kingdom_sum', 'US', 'Germany', 'Brazil',
'India', 'Italy']].var()
df_var.columns = "Variance"
df_var = df_var.sort_values()
print("Variance Confirmed COVID19 Cases")
print(df_var)
print("US had highest variance")
print("India had second highest variance")

df_med = df1[['China_sum', 'United Kingdom_sum', 'US', 'Germany', 'Brazil',
'India', 'Italy']].median()
df_med.columns = "Median"
df_med = df_med.sort_values()
print("Median Confirmed COVID19 Cases")
print(df_med)
```

```
print("US had highest median")
print("Brazil had highest median")

print("#"*80)
import seaborn as sns
df = sns.load_dataset('titanic')
df1 = df.dropna(how='any')
print(df1.head(5))
print(df1.describe())

columns = df1.columns
#print(df1)

print(f"rows = {df1.shape[0]}")
print(f"columns = {df1.shape[1]}")
print(df1)
df1 =df1[['sex','survived', 'pclass']]

male_count = df1['sex'].value_counts()['male']
female_count = df1['sex'].value_counts()['female']
gender_count = np.array([male_count, female_count])

def absolute_value(val):
    a  = np.round(val/100.*gender_count.sum(), 0)
    return a

fig, ax = plt.subplots(figsize=(8,8))
ax.pie(gender_count, labels=['Male', 'Female'], autopct=absolute_value)
ax.axis('square')
ax.set_title("Pie chart of total people on Titanic")
plt.show()

percent_male = len(df1[df1['sex'] == 'male'])/len(df1)
percent_female = len(df1[df1['sex'] == 'female'])/len(df1)
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([percent_male, percent_female], labels=['Male',
'Female'],autopct='%.1f%%' )
ax.axis('square')
ax.set_title("Pie chart of total people on Titanic in %")
plt.show()
#
#
# #Write a python program that plot the pie chart showing the percentage of
males who survived versus the percentage of males who did not survive. The
final answer should look like bellow.
percent_male_survived = len(df1[(df1['sex'] == 'male')&(df1['survived'] ==
1)])/len(df1[df1['sex'] == 'male'])
percent_male_not_survived = len(df1[(df1['sex'] == 'male')&(df1['survived']
== 0)])/len(df1[df1['sex'] == 'male'])
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([percent_male_survived, percent_male_not_survived ], labels=['Male
Survived', 'Male Not Survived'],autopct='%.1f%%' )
ax.axis('square')
ax.legend()
ax.set_title("Pie Chart of Male Survival on Titanic in %")
plt.show()
#
```

```
#
# #Write a python program that plot the pie chart showing the percentage of
females who survived versus the percentage of males who did not survive. The
final answer should look like bellow.
percent_female_survived = len(df1[(df1['sex'] == 'female')&(df1['survived']
== 1)])/len(df1[df1['sex'] == 'female'])
percent_female_not_survived = len(df1[(df1['sex'] ==
'female')&(df1['survived'] == 0)])/len(df1[df1['sex'] == 'female'])
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([percent_female_survived, percent_female_not_survived ],
labels=['Female Survived', 'Female Not Survived'],autopct='%.1f%%' )
ax.axis('square')
ax.legend()
ax.set_title("Pie Chart of Female Survival on Titanic in %")
plt.tight_layout()
plt.show()
#

#Write a python program that plot the pie chart showing the percentage
passengers with first class, second class and third-class tickets
#
first_class = len(df1[(df1['pclass'] == 1)])/len(df1)
second_class = len(df1[(df1['pclass'] == 2)])/len(df1)
third_class = len(df1[(df1['pclass'] == 3)])/len(df1)
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([first_class, second_class, third_class], labels=['Ticket Class 1',
'Ticket Class 2', 'Ticket Class 3'],explode = (.03, .3, .3),autopct='%.1f%%'
)
ax.axis('square')
ax.legend()
ax.set_title("Pie Chart Passengers Based on Level")
plt.tight_layout()
plt.show()
#
#
# #Write a python program that plot the pie chart showing the percentage
passengers with first class, second class and third-class tickets
#
first_class_survived = len(df1[(df1['pclass'] == 1)&(df1['survived'] ==
1)])/len(df1[df1['survived'] == 1])
second_class_survived = len(df1[(df1['pclass'] == 2)&(df1['survived'] ==
1)])/len(df1[df1['survived'] == 1])
third_class_survived = len(df1[(df1['pclass'] == 3)&(df1['survived'] ==
1)])/len(df1[df1['survived'] == 1])
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([first_class_survived, second_class_survived, third_class_survived],
labels=['Ticket Class 1', 'Ticket Class 2', 'Ticket Class 3'],explode = (.03,
.3, .3),autopct='%.1f%%' )
ax.axis('square')
ax.legend()
ax.set_title("Pie Chart Survival Rate Based on Ticket Class")
plt.tight_layout()
plt.show()
#
# #Write a python program that plot the pie chart showing the percentage
passengers who survived versus the percentage of passengers who did not
survive with the first-class ticket category. The final answer should look
```

```
like bellow.
first_class_survived = len(df1[(df1['pclass'] == 1)&(df1['survived'] ==
1)])/len(df1[df1['pclass'] == 1])
first_class_not_survived = len(df1[(df1['pclass'] == 1)&(df1['survived'] ==
0)])/len(df1[df1['pclass'] == 1])
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([first_class_survived, first_class_not_survived], labels=['Survival
Rate', 'Death Rate'],autopct='%.1f%%' )
ax.axis('square')
ax.legend()
ax.set_title("Survival & Death Rate: Ticket Class 1")
plt.tight_layout()
plt.show()

# Write a python program that plot the pie chart showing the percentage
passengers who survived versus the percentage of passengers who did not
survive with the second-class ticket category. The final answer should look
like bellow.
second_class_survived = len(df1[(df1['pclass'] == 2)&(df1['survived'] ==
1)])/len(df1[df1['pclass'] == 2])
second_class_not_survived = len(df1[(df1['pclass'] == 2)&(df1['survived'] ==
0)])/len(df1[df1['pclass'] == 2])
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([second_class_survived, second_class_not_survived], labels=['Survival
Rate', 'Death Rate'],autopct='%.1f%%' )
ax.axis('square')
ax.legend()
ax.set_title("Survival & Death Rate: Ticket Class 2")
plt.tight_layout()
plt.show()


#Write a python program that plot the pie chart showing the percentage
passengers who survived versus the percentage of passengers who did not
survive in the third-class ticket category.
third_class_survived = len(df1[(df1['pclass'] == 3)&(df1['survived'] ==
1)])/len(df1[df1['pclass'] == 3])
third_class_not_survived = len(df1[(df1['pclass'] == 3)&(df1['survived'] ==
0)])/len(df1[df1['pclass'] == 3])
fig, ax = plt.subplots(figsize=(8,8))
ax.pie([third_class_survived, third_class_not_survived], labels=['Survival
Rate', 'Death Rate'],autopct='%.1f%%' )
ax.axis('square')
ax.legend()
ax.set_title("Survival & Death Rate: Ticket Class 3")
plt.tight_layout()
plt.show()

fig, ax = plt.subplots(figsize=(16,8))
ax = fig.add_subplot(3,3,1)
ax.pie(gender_count, labels=['Male', 'Female'], autopct=absolute_value)
ax.axis('square')
ax.grid()
ax.set_title("Pie chart of total people on Titanic")
ax = fig.add_subplot(3,3,2)
ax.pie([percent_male, percent_female], labels=['Male',
'Female'],autopct='%.1f%%' )
```

```
ax.axis('square')
ax.grid()
ax.set_title("Pie chart of total people on Titanic in %")
ax = fig.add_subplot(3,3,3)
ax.pie([percent_male_survived, percent_male_not_survived ], labels=['Male
Survived', 'Male Not Survived'],autopct='%.1f%%' )
ax.axis('square')
ax.grid()
ax.legend()
ax.set_title("Pie Chart of Male Survival on Titanic in %")
ax = fig.add_subplot(3,3,4)
ax.pie([percent_female_survived, percent_female_not_survived ],
labels=['Female Survived', 'Female Not Survived'],autopct='%.1f%%' )
ax.axis('square')
ax.grid()
ax.legend()
ax.set_title("Pie Chart of Female Survival on Titanic in %")
ax = fig.add_subplot(3,3,5)
ax.pie([first_class, second_class, third_class], labels=['Ticket Class 1',
'Ticket Class 2', 'Ticket Class 3'],explode = (.03, .3, .3),autopct='%.1f%%'
)
ax.axis('square')
ax.grid()
ax.legend()
ax.set_title("Pie Chart Passengers Based on Level")
ax = fig.add_subplot(3,3,6)
ax.pie([first_class_survived, second_class_survived, third_class_survived],
labels=['Ticket Class 1', 'Ticket Class 2', 'Ticket Class 3'],explode = (.03,
.3, .3),autopct='%.1f%%' )
ax.axis('square')
ax.grid()
ax.legend()
ax.set_title("Pie Chart Survival Rate Based on Ticket Class")
ax = fig.add_subplot(3,3,7)
ax.pie([first_class_survived, first_class_not_survived], labels=['Survival
Rate', 'Death Rate'],autopct='%.1f%%' )
ax.axis('square')
ax.grid()
ax.legend()
ax.set_title("Survival & Death Rate: Ticket Class 1")
ax = fig.add_subplot(3,3,8)
ax.pie([second_class_survived, second_class_not_survived], labels=['Survival
Rate', 'Death Rate'],autopct='%.1f%%' )
ax.axis('square')
ax.grid()
ax.legend()
ax.set_title("Survival & Death Rate: Ticket Class 2")
ax = fig.add_subplot(3,3,9)
ax.pie([third_class_survived, third_class_not_survived], labels=['Survival
Rate', 'Death Rate'],autopct='%.1f%%' )
ax.axis('square')
ax.legend()
ax.set_title("Survival & Death Rate: Ticket Class 3")
ax.grid()
plt.show()
```

References

Sample Mean: Symbol (X Bar), Definition, Standard Error. (n.d.). Statistics How To. https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/sample-mean/

Wikipedia Contributors. (2019, January 8). Variance. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Variance

Wikipedia Contributors. (2022). Median. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Median