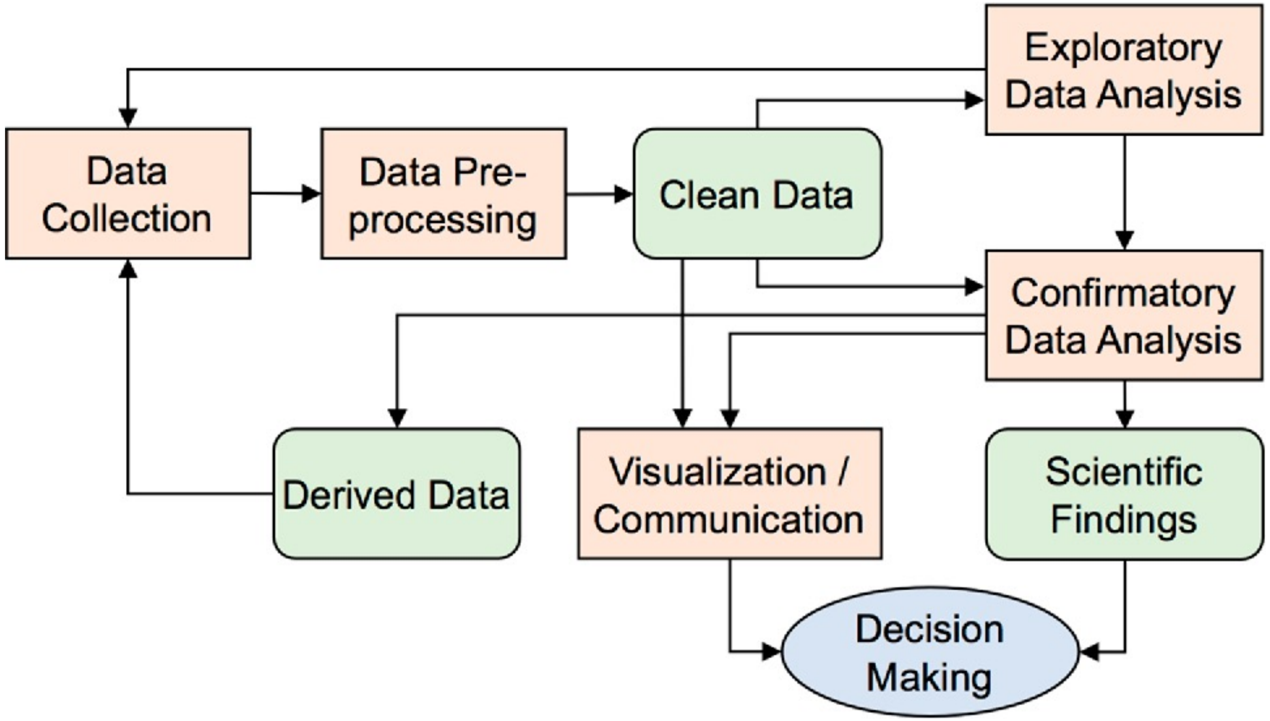
CSIT 558 DATA MINING

ASSIGNMENT 1: STATISTICAL ANALYSIS OF DATASET

RAMY OTHMAN

Exploratory Data Analysis Steps



OBJECTIVE

- Choosing dataset from real world
- Load the dataset
- Cleaning the dataset
- Describe the dataset using Descriptive Statistics methods
- Using Data Aggregation
- Visualize the dataset

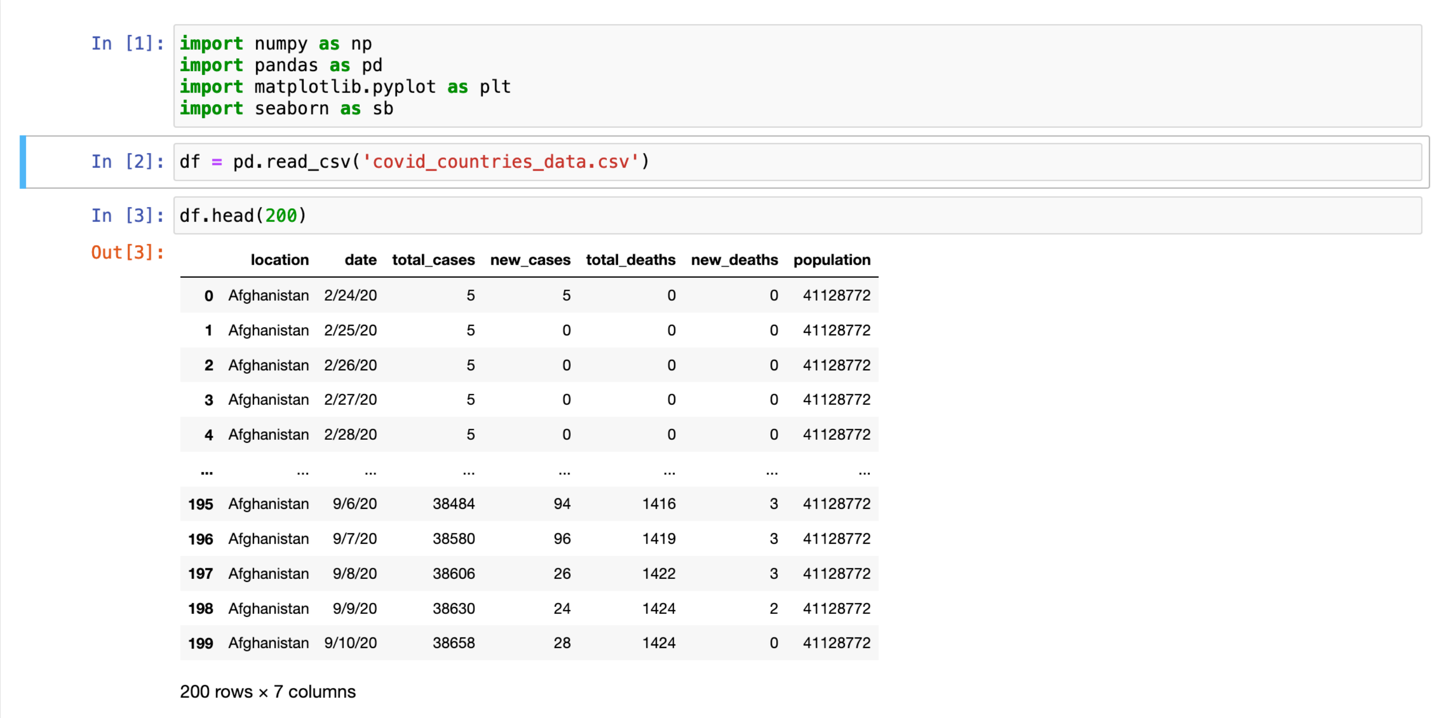
Dataset

- The dataset loaded in the assignment is COVID-19
- The dataset used has record of:
  - Total cases
  - Total deaths
  - new deaths
  - Countries population
  - Recorded Dates are between Feb. 24 2020 till Feb. 21 2023

Visualization Technique

- Using Matplotlib library to visualize the dataset:
  - Bar Chart
  - Pie Chart
  - Line Chart

Importing libraries and Loading the Dataset



In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [2]:

```
df = pd.read_csv('covid_countries_data.csv')
```

In [3]:

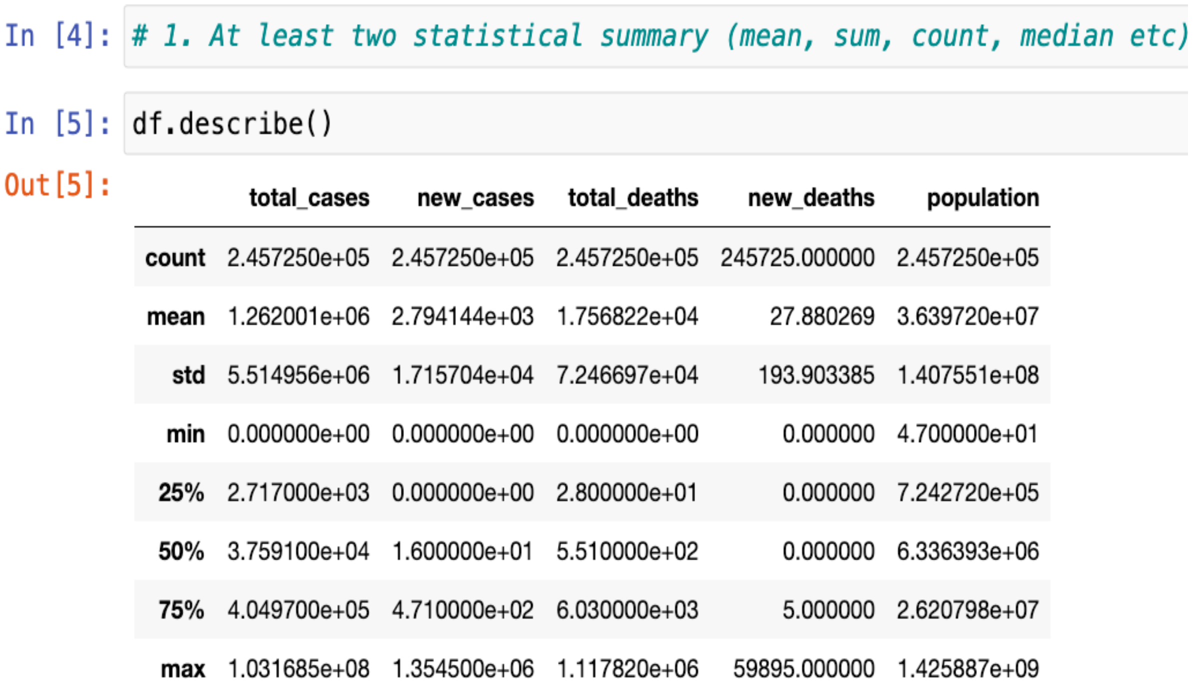
```
df.head(200)
```

Out[3]:

	location	date	total_cases	new_cases	total_deaths	new_deaths	population
0	Afghanistan	2/24/20	5	5	0	0	41128772
1	Afghanistan	2/25/20	5	0	0	0	41128772
2	Afghanistan	2/26/20	5	0	0	0	41128772
3	Afghanistan	2/27/20	5	0	0	0	41128772
4	Afghanistan	2/28/20	5	0	0	0	41128772
...	...	...	...	...	...	...	...
195	Afghanistan	9/6/20	38484	94	1416	3	41128772
196	Afghanistan	9/7/20	38580	96	1419	3	41128772
197	Afghanistan	9/8/20	38606	26	1422	3	41128772
198	Afghanistan	9/9/20	38630	24	1424	2	41128772
199	Afghanistan	9/10/20	38658	28	1424	0	41128772

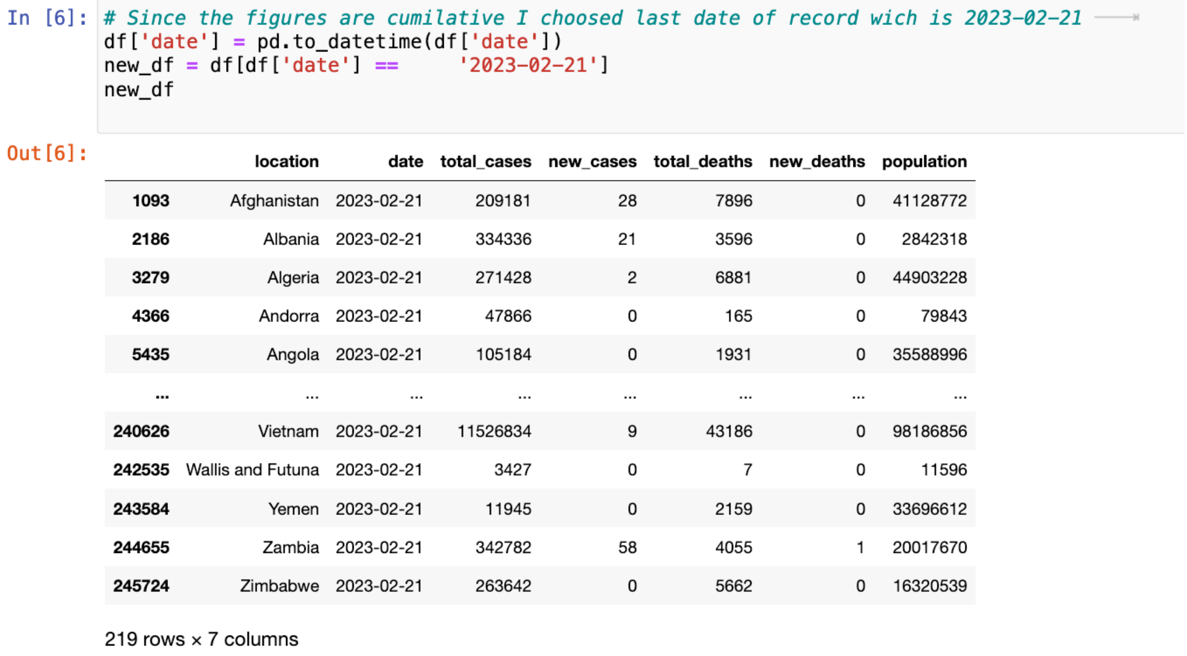
200 rows × 7 columns

Getting Statistical summary and Convert to Time Series



```
In [4]: # 1. At least two statistical summary (mean, sum, count, median etc)

In [5]: df.describe()

Out[5]:
```

	total_cases	new_cases	total_deaths	new_deaths	population
count	2.457250e+05	2.457250e+05	2.457250e+05	245725.000000	2.457250e+05
mean	1.262001e+06	2.794144e+03	1.756822e+04	27.880269	3.639720e+07
std	5.514956e+06	1.715704e+04	7.246697e+04	193.903385	1.407551e+08
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	4.700000e+01
25%	2.717000e+03	0.000000e+00	2.800000e+01	0.000000	7.242720e+05
50%	3.759100e+04	1.600000e+01	5.510000e+02	0.000000	6.336393e+06
75%	4.049700e+05	4.710000e+02	6.030000e+03	5.000000	2.620798e+07
max	1.031685e+08	1.354500e+06	1.117820e+06	59895.000000	1.425887e+09

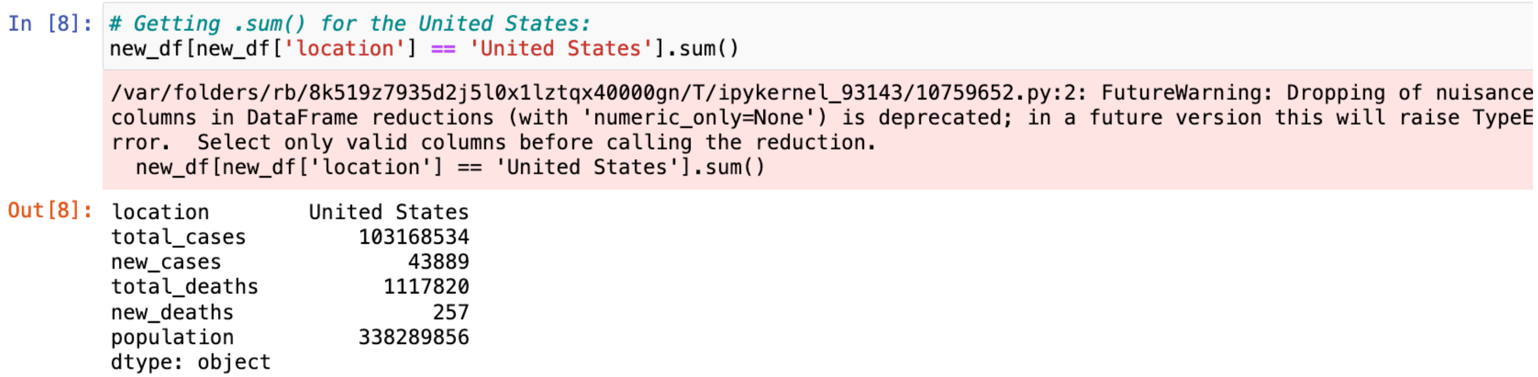


```
In [6]: # Since the figures are cumilative I choosed last date of record wich is 2023-02-21
        df['date'] = pd.to_datetime(df['date'])
        new_df = df[df['date'] ==     '2023-02-21']
        new_df

Out[6]:
```

	location	date	total_cases	new_cases	total_deaths	new_deaths	population
1093	Afghanistan	2023-02-21	209181	28	7896	0	41128772
2186	Albania	2023-02-21	334336	21	3596	0	2842318
3279	Algeria	2023-02-21	271428	2	6881	0	44903228
4366	Andorra	2023-02-21	47866	0	165	0	79843
5435	Angola	2023-02-21	105184	0	1931	0	35588996
...	...	...	...	...	...	...	...
240626	Vietnam	2023-02-21	11526834	9	43186	0	98186856
242535	Wallis and Futuna	2023-02-21	3427	0	7	0	11596
243584	Yemen	2023-02-21	11945	0	2159	0	33696612
244655	Zambia	2023-02-21	342782	58	4055	1	20017670
245724	Zimbabwe	2023-02-21	263642	0	5662	0	16320539

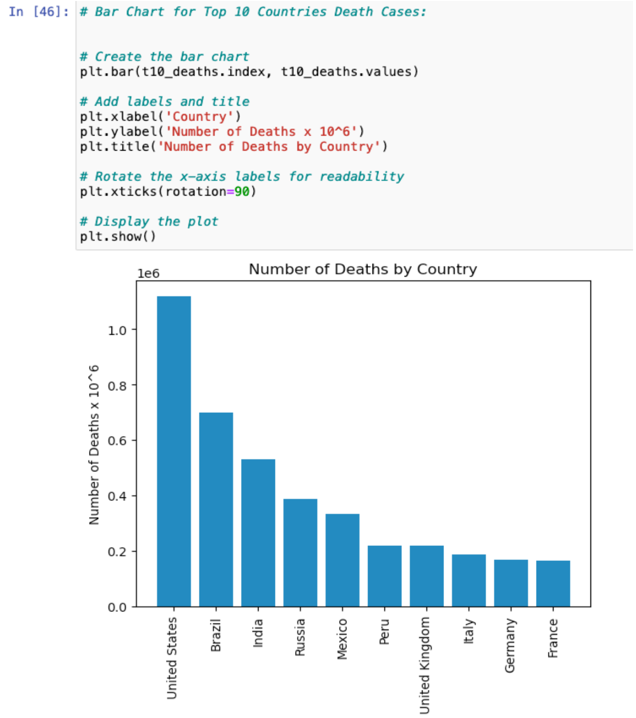
219 rows × 7 columns

Working on US records

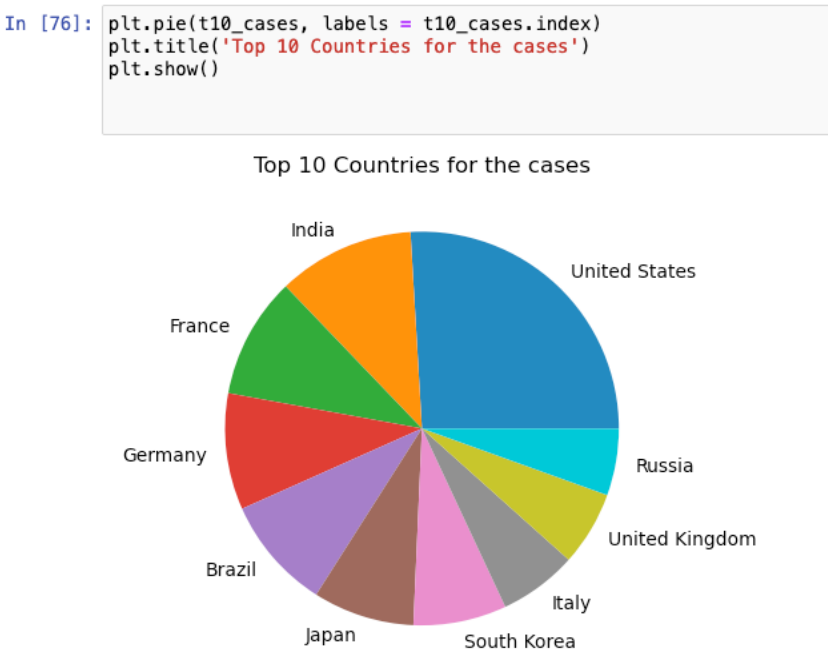


```
In [8]: # Getting .sum() for the United States:
        new_df[new_df['location'] == 'United States'].sum()

/var/folders/rb/8k519z7935d2j5l0x1lztqx40000gn/T/ipykernel_93143/10759652.py:2: FutureWarning: Dropping of nuisance
columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeE
rror.  Select only valid columns before calling the reduction.
  new_df[new_df['location'] == 'United States'].sum()

Out[8]: location        United States
        total_cases         103168534
        new_cases               43889
        total_deaths          1117820
        new_deaths                257
        population          338289856
        dtype: object
```

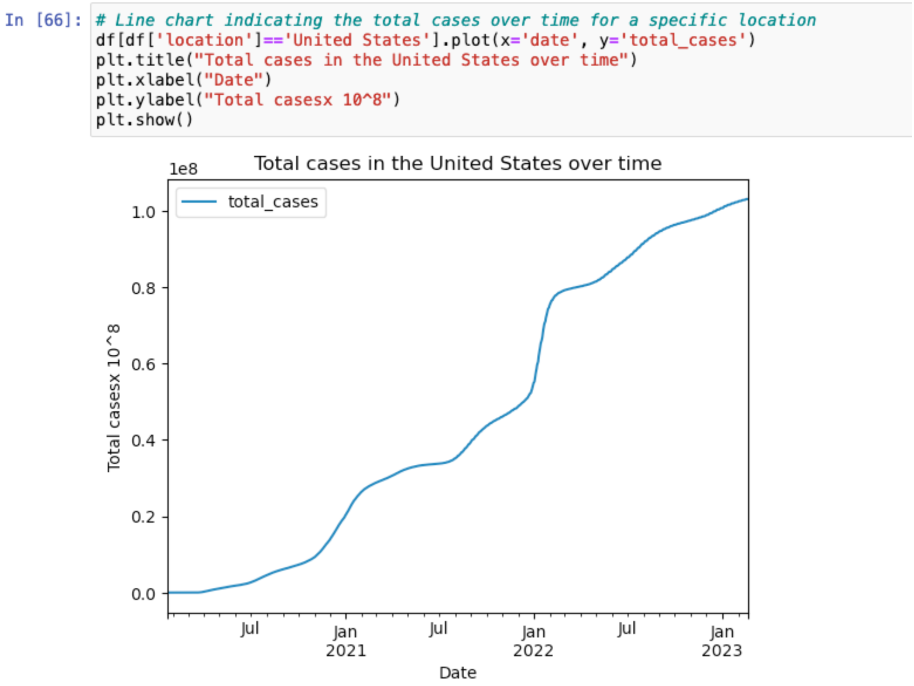
Visualize the dataset



```
In [46]: # Bar Chart for Top 10 Countries Death Cases:


         # Create the bar chart
         plt.bar(t10_deaths.index, t10_deaths.values)

         # Add labels and title
         plt.xlabel('Country')
         plt.ylabel('Number of Deaths x 10^6')
         plt.title('Number of Deaths by Country')

         # Rotate the x-axis labels for readability
         plt.xticks(rotation=90)

         # Display the plot
         plt.show()
```



```
In [76]: plt.pie(t10_cases, labels = t10_cases.index)
         plt.title('Top 10 Countries for the cases')
         plt.show()
```



```
In [66]: # Line chart indicating the total cases over time for a specific location
         df[df['location']=='United States'].plot(x='date', y='total_cases')
         plt.title("Total cases in the United States over time")
         plt.xlabel("Date")
         plt.ylabel("Total casesx 10^8")
         plt.show()
```

References

- Dataset source:
  - https://www.kaggle.com/datasets/imdevskp/corona-virus-report
- Software Tools:
  - Anaconda, Jupyter Notebook
- Text books:
  - Visualization Analysis and Design by Tamara Munzner
    ISBN-10: 9781466508910 • ISBN-13: 978-1466508910 ©2014
  - Python for Data Analysis 2/E by Wes McKinney
    ISBN-10: 1491957662 • ISBN-13: 1491957660 ©2018