**What is EDA and why is this important?**

EDA is an approach to understand the various aspect of the data. The primary aim with EDA is to examine the distribution of the data, reveal an underlying structure, detect outliers, find the hidden correlation and relationships among the data, among others. EDA is an initial step in the data analysis process after data collection where the data is summarised, visualised, plotted to get insight into the dataset.

*The goal of EDA is to discover patterns in data.*

**How to perform EDA?**

1. *Summary Statistics*

These are measurements to describe data. We can use df.describe() function to get various summary statistics. However, this process excludes NaN values and categorical values. To include description for categorical data we need to use df.describe(include= "all").

2. *Visualise*

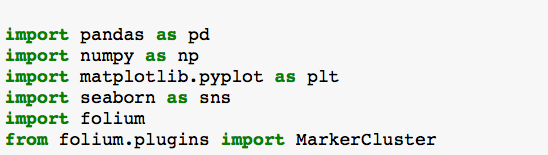
Different visualisation techniques such can histogram, scatter plots, box plot, distribution plots, maps, etc can be used to perform EDA.

In this article, we will do EDA using the COVID-19 dataset. We will use different commonly used EDA techniques to discover patterns in the COVID-19 dataset. The complete code can be found on my [github](https://github.com/pragpaudyal/covid).

**Datasets**

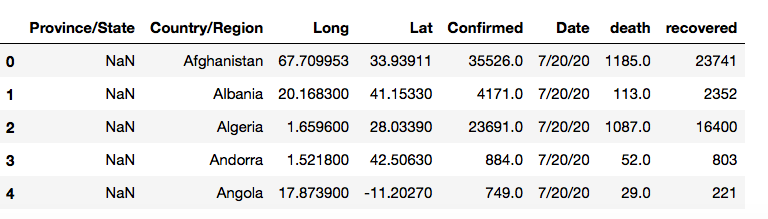
We will be working with the publicly available dataset from [JHU CSSE’s COVID-19 data repository](https://github.com/CSSEGISandData/COVID-19). The data repository consists of different datasets related to covid-19. I merged and concat the relevant data. You can find modified dataset on my [github](https://github.com/pragpaudyal/covid).

**Import Libraries and loading data**

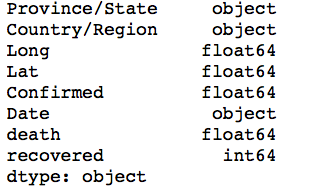


data=pd.read\_csv(“country\_vaccination.csv”)

We will use data.shape to check shape of the dataset. Using .head() function, we get a glance to the dataset. covid19\_csv consists of following row:



Using .dtypes() function, we can check the datatype. We can see data type are either object or float. When the data type is object, it is a categorical data.



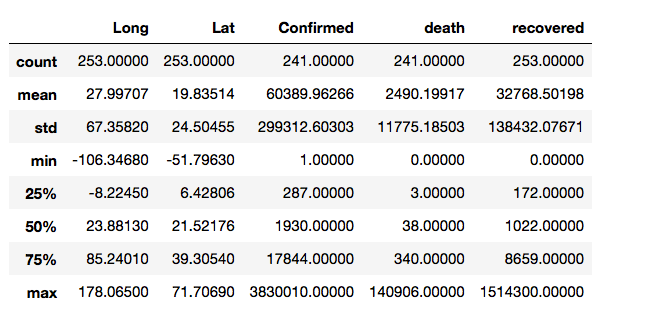
**Using EDA approach to get more insight of the data**

1. **Summary Statistic**

The describe functions give us descriptive statistics that summarise the count, mean, standard deviation, minimum, maximum, quartile (lower, middle and upper) values. Sometimes, you will get your result in scientific notation, to suppress scientific notation in Pandas we will use:

pd.set\_option (‘display. float format’, lambda x: ‘%.5f’ % x)

We can use data.describe() function to get summary statistic of the dataframe. This function will give us summary only for numerical datasets.

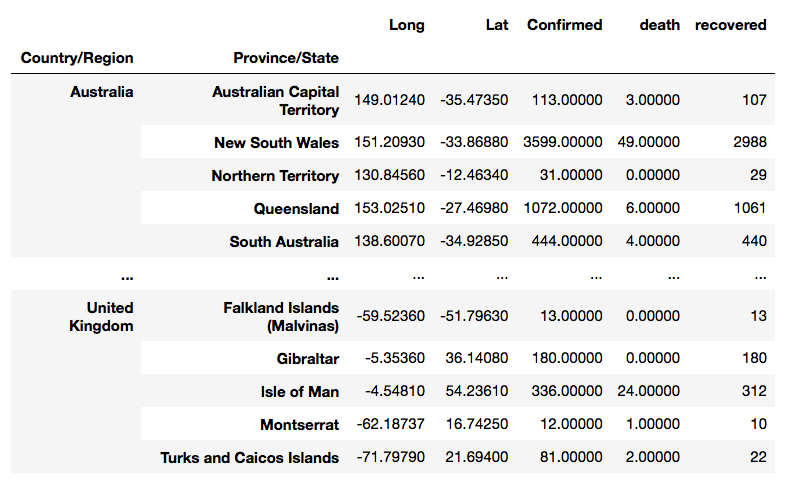


However, .describe(include=”all”) describes all columns of a dataframe regardless of data type. We can also observe if there are any missing values for any features. For instant, we can see **confirmed** and **death** have missing values. We can get more insight for confirmed, death and recovered data such as what is the maximum, minimum number for each feature.

**2. Visualisation**

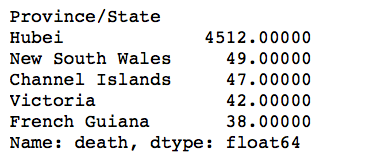
To group confirmed, death and recovery by Country/Region we use groupby function.

data. groupby([“Country/Region”, “Province/State”]).sum()



We can also group the data based on Province/State and view top 5 Province with highest death.

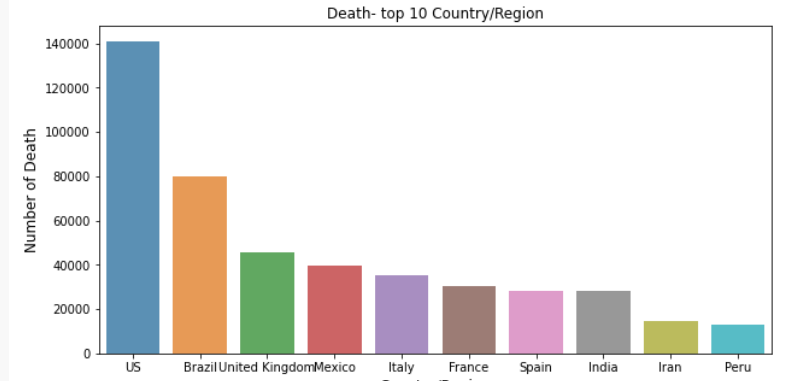
data. groupby(“Province/State”)[“death”]. sum().sort\_values(ascending=False) [:5]



This will show only top 5 Province with highest death. This information provides lots of useful insight.

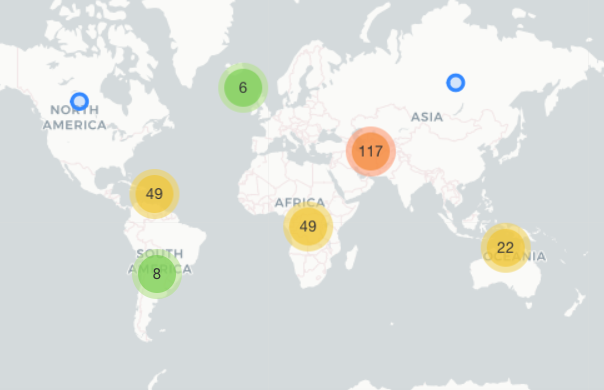
We can get a bar graph for Province based on death rate.

Country=data.groupby("Country/Region")["death"].sum().sort\_values(ascending=False)[:10]  
plt.figure(figsize=(10,5))  
sns.barplot(Country.index, Country.values, alpha=0.8)  
plt.title('Death- top 10 Country/Region')  
plt.ylabel('Number of Death', fontsize=12)  
plt.xlabel('Country/Region', fontsize=12)  
plt.show()



In addition to these approaches, we also used map to explore geographical analysis on Covid-19. There is different python package to create visual and informative maps. Here we used folium to create a world map to show effect of covid-19 around the world. Here we used longitude and latitude data from the dataset.

world map= folium. Map(tiles="cartodbpositron")  
marker\_cluster = MarkerCluster().add\_to(world\_map)for i in range(len(data)):  
 Lat = data.iloc[i]['Lat']  
 Long = data.iloc[i]['Long']  
 radius=6  
 popup\_text = """Country/Region : {}<br>  
 Confirmed : {}<br>  
 death: {} <br>  
 recovered:{} <br>  
 Date:{} <br>"""  
 popup\_text = popup\_text.format(data.iloc[i]['Country/Region'],  
 data.iloc[i]['Confirmed'],  
 data.iloc[i]['death'],  
 data.iloc[i]['recovered'],  
 data.iloc[i]['Date']  
 )  
 folium.CircleMarker(location = [Lat, Long], radius=radius, popup= popup\_text, fill =True).add\_to(marker\_cluster)world\_map



When we click on the bubble, the map will zoom in to show more detail for country.