Welcome to Covid19 Data Analysis Notebook

Let's Import the modules

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
print('Modules are imported.')
```

```
Modules are imported.
```

Task 2

Task 2.1: importing covid19 dataset

importing "Covid19_Confirmed_dataset.csv" from "./Dataset" folder.

```
corona_dataset_csv = pd.read_csv("../covid19_Confirmed_dataset.csv")
corona_dataset_csv.head()
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	
1	NaN	Albania	41.1533	20.1683	0	0	0	
2	NaN	Algeria	28.0339	1.6596	0	0	0	
3	NaN	Andorra	42.5063	1.5218	0	0	0	
4	NaN	Angola	-11.2027	17.8739	0	0	0	

5 rows × 104 columns

Let's check the shape of the dataframe

```
corona_dataset_csv.shape
```

```
(266, 104)
```

Task 2.2: Delete the useless columns

```
corona_dataset_csv.drop(["Lat","Long"],axis=1,inplace=True)
```

```
corona_dataset_csv.head(10)
```

	Province/State	Country/Region	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/
0	NaN	Afghanistan	0	0	0	0	0	
1	NaN	Albania	0	0	0	0	0	
2	NaN	Algeria	0	0	0	0	0	
3	NaN	Andorra	0	0	0	0	0	
4	NaN	Angola	0	0	0	0	0	
5	NaN	Antigua and Barbuda	0	0	0	0	0	
6	NaN	Argentina	0	0	0	0	0	
7	NaN	Armenia	0	0	0	0	0	
8	Australian Capital Territory	Australia	0	0	0	0	0	
9	New South Wales	Australia	0	0	0	0	3	

10 rows × 102 columns

Task 2.3: Aggregating the rows by the country

```
corona_dataset_aggregated=corona_dataset_csv.groupby("Country/Region").sum()
```

```
corona_dataset_aggregated.head()
```

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/2
Country/Region								
Afghanistan	0	0	0	0	0	0	0	
Albania	0	0	0	0	0	0	0	
Algeria	0	0	0	0	0	0	0	
Andorra	0	0	0	0	0	0	0	
Angola	0	0	0	0	0	0	0	

5 rows × 100 columns

```
corona_dataset_aggregated.shape
corona_dataset_aggregated.shape
```

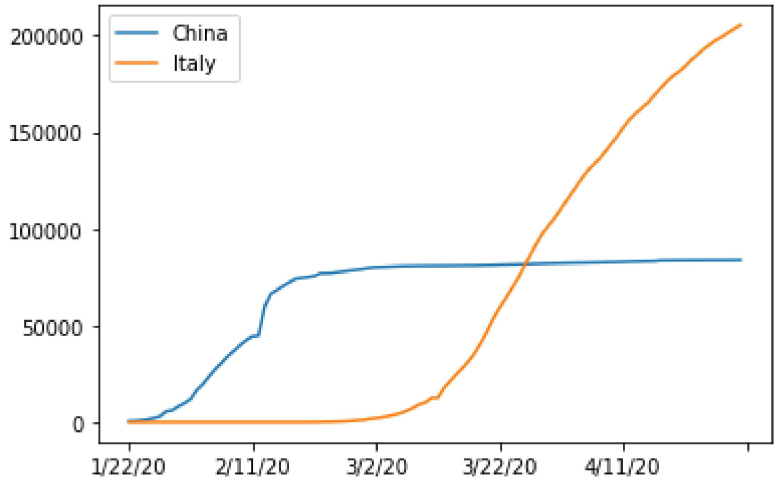
```
(187, 100)
```

Task 2.4: Visualizing data related to a country for example China

visualization always helps for better understanding of our data.

```
corona_dataset_aggregated.loc["China"].plot()
corona_dataset_aggregated.loc["Italy"].plot()
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fb8587e9320>
```

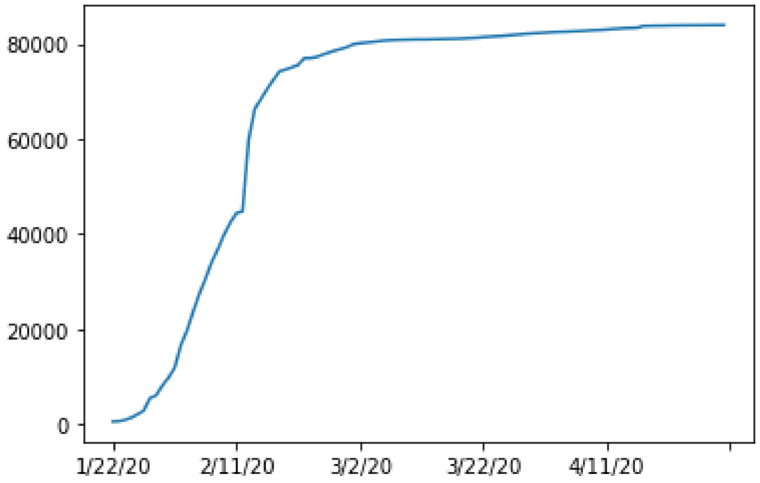


Task3: Calculating a good measure

we need to find a good measure reperestend as a number, describing the spread of the virus in a country.

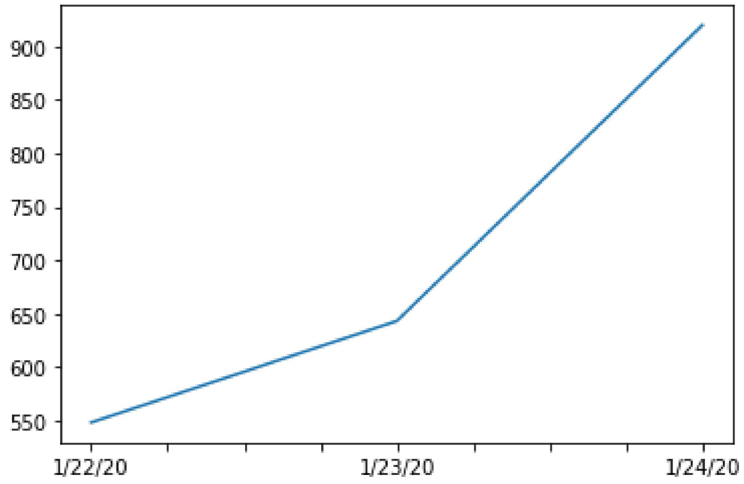
```
corona_dataset_aggregated.loc['China'].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb87b72b400>
```



```
corona_dataset_aggregated.loc["China"][:3].plot()
```

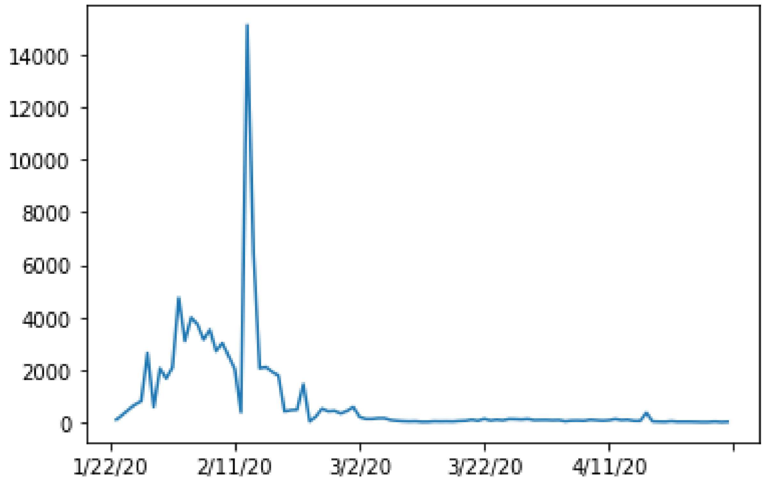
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8582c32b0>
```



task 3.1: caculating the first derivative of the curve

```
corona_dataset_aggregated.loc["China"].diff().plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb85827ba90>
```



task 3.2: find maxmimum infection rate for China

```
corona_dataset_aggregated.loc["China"].diff().max()
```

```
15136.0
```

```
corona_dataset_aggregated.loc["Italy"].diff().max()
```

```
6557.0
```

```
corona_dataset_aggregated.loc["Spain"].diff().max()
```

```
9630.0
```

Task 3.3: find maximum infection rate for all of the countries.

```
countries=list(corona_dataset_aggregated.index)
max_infection_rates=[]
for c in countries:
    max_infection_rates.append(corona_dataset_aggregated.loc[c].diff().max())
corona_dataset_aggregated["max_infection_rate"]=max_infection_rates
```

```
corona_dataset_aggregated.head()
```

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/2
Country/Region								
Afghanistan	0	0	0	0	0	0	0	
Albania	0	0	0	0	0	0	0	
Algeria	0	0	0	0	0	0	0	
Andorra	0	0	0	0	0	0	0	
Angola	0	0	0	0	0	0	0	

5 rows × 101 columns

Task 3.4: create a new dataframe with only needed column

```
corona_data=pd.DataFrame(corona_dataset_aggregated["max_infection_rate"])
```

```
corona_data.head()
```

	max_infection_rate
Country/Region	
Afghanistan	232.0
Albania	34.0
Algeria	199.0
Andorra	43.0
Angola	5.0

Task4:

- Importing the WorldHappinessReport.csv dataset
- selecting needed columns for our analysis
- join the datasets

- calculate the correlations as the result of our analysis

Task 4.1 : importing the dataset

```
happiness_report_csv=pd.read_csv("../worldwide_happiness_report.csv")
```

```
happiness_report_csv.head()
```

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perc cor
0	1	Finland	7.769	1.340	1.587	0.986	0.596	0.153	
1	2	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	
2	3	Norway	7.554	1.488	1.582	1.028	0.603	0.271	
3	4	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	

Task 4.2: let's drop the useless columns

```
useless_cols=["Overall rank","Score","Generosity","Perceptions of corruption"]
```

```
happiness_report_csv.drop(useless_cols,axis=1,inplace=True)
happiness_report_csv.head()
```

	Country or region	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
0	Finland	1.340	1.587	0.986	0.596
1	Denmark	1.383	1.573	0.996	0.592
2	Norway	1.488	1.582	1.028	0.603
3	Iceland	1.380	1.624	1.026	0.591
4	Netherlands	1.396	1.522	0.999	0.557

Task 4.3: changing the indices of the dataframe

```
happiness_report_csv.set_index("Country or region",inplace=True)
```

```
happiness_report_csv.head()
```

	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Country or region				
Finland	1.340	1.587	0.986	0.596
Denmark	1.383	1.573	0.996	0.592

Task4.4: now let's join two dataset we have prepared

Corona Dataset :

```
corona_data.head()
```

	max_infection_rate
Country/Region	
Afghanistan	232.0
Albania	34.0
Algeria	199.0
Andorra	43.0
Angola	5.0

wolrd happiness report Dataset :

```
happiness_report_csv.head()
```

	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Country or region				
Finland	1.340	1.587	0.986	0.596
Denmark	1.383	1.573	0.996	0.592
Norway	1.488	1.582	1.028	0.603
Iceland	1.380	1.624	1.026	0.591

```
happiness_report_csv.shape
```

```
(156, 4)
```

```
data=corona_data.join(happiness_report_csv,how="inner")
```

```
data=corona_data.join(happiness_report_csv,how="inner")
data.head()
```

	max_infection_rate	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Afghanistan	232.0	0.350	0.517	0.361	0.000
Albania	34.0	0.947	0.848	0.874	0.383
Algeria	199.0	1.002	1.160	0.785	0.086
Argentina	291.0	1.092	1.432	0.881	0.471
Armenia	134.0	0.850	1.055	0.815	0.283

Task 4.5: correlation matrix

```
data.corr()
```

	max_infection_rate	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
max_infection_rate	1.000000	0.250118	0.191958	0.289263	0.078196
GDP per capita	0.250118	1.000000	0.759468	0.863062	0.394603
Social support	0.191958	0.759468	1.000000	0.765286	0.456246
Healthy life expectancy	0.289263	0.863062	0.765286	1.000000	0.427892

Task 5: Visualization of the results

our Analysis is not finished unless we visualize the results in terms figures and graphs so that everyone can understand what you get out of our analysis

```
data.head()
```

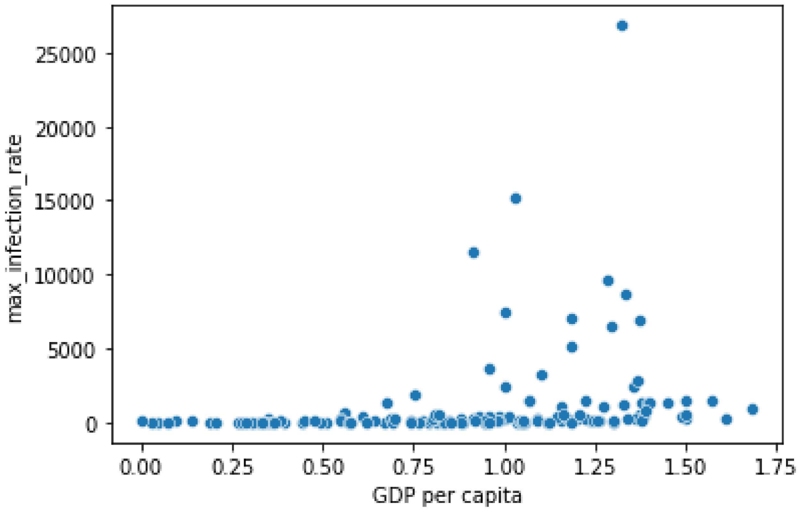
	max_infection_rate	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Afghanistan	232.0	0.350	0.517	0.361	0.000
Albania	34.0	0.947	0.848	0.874	0.383
Algeria	199.0	1.002	1.160	0.785	0.086
Argentina	291.0	1.092	1.432	0.881	0.471
Armenia	134.0	0.850	1.055	0.815	0.283

Task 5.1: Plotting GDP vs maximum Infection rate

```
x=data["GDP per capita"]
```

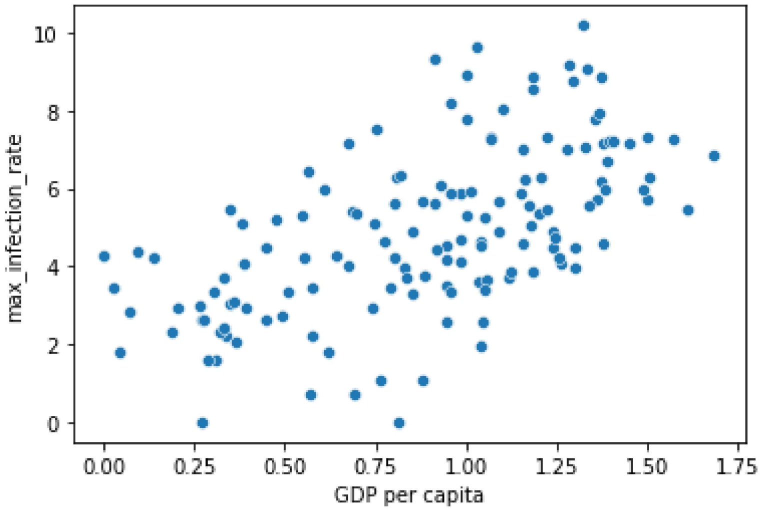
```
x=data["GDP per capita"]
y=data["max_infection_rate"]
sns.scatterplot(x,y)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8582116a0>
```



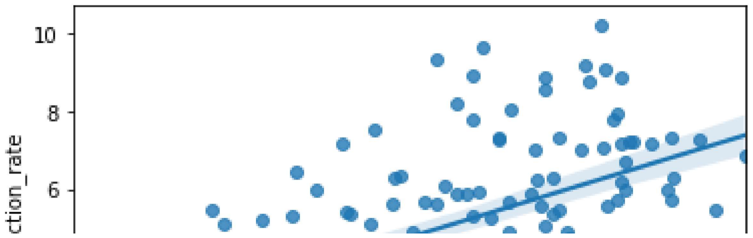
```
sns.scatterplot(x,np.log(y))
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb85816a160>
```



```
sns.regplot(x,np.log(y))
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb856e6dda0>
```



Task 5.2: Plotting Social support vs maximum Infection rate

Task 5.3: Plotting Healthy life expectancy vs maximum Infection rate

GDP per capita

Task 5.4: Plotting Freedom to make life choices vs maximum Infection rate