Welcome to Covid19 Data Analysis Notebook

Let's Import the modules

```
In [1]: 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.

```
In [2]: corona = pd.read_csv('covid19_Confirmed_dataset.csv')
        corona.head()
```

Out[2]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	4/31/20	4/32/20
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	1092	1176
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	609	634
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	2811	2910
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	717	723
4	NaN	Angola	11.2027	17.8739	0	0	0	0	0	0	...	24	25

5 rows × 104 columns

Let's check the shape of the dataframe

```
In [7]: corona.shape
```

```
Out[7]: (266, 104)
```

Task 2.2: Delete the useless columns

```
In [8]: corona.columns
```

```
Out[8]: Index(['Province/State', 'Country/Region', 'Lat', 'Long', '1/22/20', '1/23/20',
               '1/24/20', '1/25/20', '1/26/20', '1/27/20',
               ...
               '4/21/20', '4/22/20', '4/23/20', '4/24/20', '4/25/20', '4/26/20',
               '4/27/20', '4/28/20', '4/29/20', '4/30/20'],
              dtype="object", length=104)
```

```
In [9]: corona.drop(['Lat','Long'],axis=1,inplace=True)
```

```
In [10]: corona.head()
```

```
Out[10]:
```

	Province/State	Country/Region	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	...	4/31/20	4/32/20	4/3
0	NaN	Afghanistan	0	0	0	0	0	0	0	0	...	1092	1176	1
1	NaN	Albania	0	0	0	0	0	0	0	0	...	609	634	
2	NaN	Algeria	0	0	0	0	0	0	0	0	...	2811	2910	3
3	NaN	Andorra	0	0	0	0	0	0	0	0	...	717	723	
4	NaN	Angola	0	0	0	0	0	0	0	0	...	24	25	

5 rows × 100 columns

```
In [13]: df.shape
```

```
Out[13]: (187, 100)
```

Task 2.3: Aggregating the rows by the country

```
In [11]: df = corona.groupby("Country/Region").sum()
```

```
In [12]: df.head()
```

```
Out[12]:
```

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	...	4/31/20	4/32/20	4/3
Country/Region														
Afghanistan	0	0	0	0	0	0	0	0	0	0	...	1092	1176	1
Albania	0	0	0	0	0	0	0	0	0	0	...	609	634	
Algeria	0	0	0	0	0	0	0	0	0	0	...	2811	2910	3
Andorra	0	0	0	0	0	0	0	0	0	0	...	717	723	
Angola	0	0	0	0	0	0	0	0	0	0	...	24	25	

5 rows × 100 columns

```
In [13]: df.shape
```

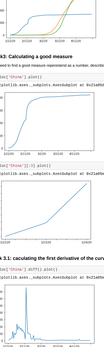
```
Out[13]: (187, 100)
```

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

visualization always helps for better understanding of our data.

```
In [14]: df.loc['China'].plot()
         df.loc['Italy'].plot()
         df.loc['Spain'].plot()
         plt.legend()
```

```
Out[14]: <matplotlib.legend.Legend at 0x21a6652dd08>
```



Task3: Calculating a good measure

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

```
In [15]: df.loc['China'].plot()
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x21a6856cb848>
```


```
In [16]: df.loc['China'][:3].plot()
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x21a85e3ec48>
```

task 3.1: caculating the first derivative of the curve

```
In [17]: df.loc['China'].diff().plot()
```

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x21a85e96f88>
```

task 3.2: find maximum infection rate for China

```
In [18]: df.loc['China'].diff().max()
```

```
Out[18]: 15136.0
```

```
In [19]: df.loc['Italy'].diff().max()
```

```
Out[19]: 6557.0
```

```
In [20]: df.loc['Spain'].diff().max()
```

```
Out[20]: 9630.0
```

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

```
In [21]: countries = list(df.index)
         max_infra = []
         for c in countries:
             max_infra.append(df.loc[c].diff().max())
         df['Max_infra'] = max_infra
```

```
In [22]: df.head()
```

```
Out[22]:
```

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	...	4/31/20	4/32/20	4/3
Country/Region														
Afghanistan	0	0	0	0	0	0	0	0	0	0	...	1176	1176	1
Albania	0	0	0	0	0	0	0	0	0	0	...	634	634	
Algeria	0	0	0	0	0	0	0	0	0	0	...	2910	3007	3
Andorra	0	0	0	0	0	0	0	0	0	0	...	723	723	
Angola	0	0	0	0	0	0	0	0	0	0	...	25	25	

5 rows × 101 columns

Task 3.4: create a new dataframe with only needed colum

```
In [23]: df1 = pd.DataFrame(df['Max_infra'])
```

```
In [24]: df1.head()
```

```
Out[24]:
```

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

Task4:

- importing the World Happiness report.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

```
In [27]: happy = pd.read_csv('worldwide_happiness_report.csv')
```

```
In [28]: happy.head()
```

```
Out[28]:
```

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	1	Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393
1	2	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410
2	3	Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341
3	4	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118
4	5	Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298

Task 4.2: let's drop the useless columns

```
In [29]: useless_col = ['Overall rank','Score', 'Generosity','Perceptions of corruption']
```

```
In [30]: happy.drop(useless_col,axis=1,inplace=True)
```

```
In [31]: happy.head()
```

```
Out[31]:
```

	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

```
In [32]: happy.set_index('Country or region',inplace=True)
         happy.head()
```

```
Out[32]:
```

	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
Netherlands	1.396	1.522	0.999	0.557

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

Corona Dataset :

```
In [33]: df1.head()
```

```
Out[33]:
```

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

```
In [34]: df1.shape
```

```
Out[34]: (187, 1)
```

world happiness report Dataset :

```
In [35]: happy.head()
```

```
Out[35]:
```

	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
Netherlands	1.396	1.522	0.999	0.557

```
In [36]: happy.shape
```

```
Out[36]: (156, 4)
```

```
In [37]: data = df1.join(happy, how='inner')
         data.head()
```

```
Out[37]:
```

	max_infec	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

```
In [38]: data.corr()
```

```
Out[38]:
```

	max_infec	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
max_infec	1.000000	0.250118	0.191958	0.289033	0.078196
GDP per capita	0.250118	1.000000	0.758492	0.862932	0.394603
Social support	0.191958	0.758492	1.000000	0.763200	0.456394
Healthy life expectancy	0.289033	0.862932	0.763200	1.000000	0.427662
Freedom to make life choices	0.078196	0.394603	0.456394	0.427662	1.000000

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

```
In [39]: data.head()
```

```
Out[39]:
```

	max_infec	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

```
In [40]: x = data['GDP per capita']
         y = data['max_infec']
         sns.scatterplot(x,np.log(y)) #without scaling (a lot of diffence nd we cant predict)
```

```
Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x21a85e3b188>
```



```
In [41]: sns.regplot(x,np.log(y))
```

```
Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x21a85fbc708>
```


Task 5.2: Plotting Social support vs maximum Infection rate

```
In [42]: x = data['Social support']
         y = data['max_infec']
         sns.scatterplot(x,np.log(y))
```

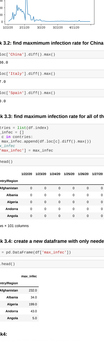
```
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x21a85faac88>
```


```
In [ ]:
```

Task 5.3: Plotting Healthy life expectancy vs maximum Infection rate

```
In [43]: x = data['Healthy life expectancy']
         y = data['max_infec']
         sns.scatterplot(x,np.log(y))
```

```
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x21a85f9ee48>
```

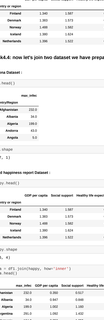



```
In [ ]:
```

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

```
In [44]: x = data['Freedom to make life choices']
         y = data['max_infec']
         sns.scatterplot(x,np.log(y))
```

```
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x21a8660c780>
```




```
In [ ]:
```