**NAAN MUDHALVAN**

**DATA ANALYTICS**

**PHASE-4**

**PROJECT-1**

**PROJECT TITLE:COVID 19 CASES AND DEATH ANALYSIS**

**DOMAIN:DATA ANALYTICS**

**Introduction:**

Provide an opening to the project, introducing the context and importance of COVID-19 cases and death data analytics.

Briefly explain the significance of analyzing COVID-19 data in the context of public health.

Outline the objectives of the data analytics project and its potential impact on understanding the pandemic.

**Project Overview:**

Offer a broad perspective on the project, focusing on COVID-19 cases and death data analytics.

Describe the overall scope and purpose of the project.

Highlight the relevance of analyzing COVID-19 data for informed decision-making.

Provide a glimpse into the key components of the project.

**Project Development:**

Detail how the project was carried out, covering data collection, analysis, and visualization.

Describe the methodologies, techniques, and tools used for data analytics.

Specify the data sources and datasets that were employed for the analysis.

Explain the process of data cleaning, validation, and preprocessing.

# PROGRAM:

# 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.

## importing covid19 dataset

importing "Covid19\_Confirmed\_dataset.csv" from "./Dataset" folder.

In [2]:

df**=**pd**.**read\_csv("../input/covid19/covid19\_Confirmed\_dataset.csv") df**.**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 ... 4/21/20 4/2**

1. NaN Afghanistan 33.0000 65.0000 0 0 0 0 0 0 ... 1092 1
2. NaN Albania 41.1533 20.1683 0 0 0 0 0 0 ... 609
3. NaN Algeria 28.0339 1.6596 0 0 0 0 0 0 ... 2811 2
4. NaN Andorra 42.5063 1.5218 0 0 0 0 0 0 ... 717

-

1. NaN Angola 17.8739 0 0 0 0 0 0 ... 24

11.2027

1. rows × 104 columns

**Let's check the shape of the dataframe**

In [3]:

df**.**shape

Out[3]:

(266, 104)

## Delete the useless columns

In [4]: df**.**drop(["Lat","Long"],axis**=**1,inplace**=True**)

In [5]: df**.**head() Out[5]:

**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 ... 4/21/20 4/22**

1. NaN Afghanistan 0 0 0 0 0 0 0 0 ... 1092 11
2. NaN Albania 0 0 0 0 0 0 0 0 ... 609 6
3. NaN Algeria 0 0 0 0 0 0 0 0 ... 2811 29
4. NaN Andorra 0 0 0 0 0 0 0 0 ... 717 7
5. NaN Angola 0 0 0 0 0 0 0 0 ... 24
6. rows × 102 columns

## Aggregating the rows by the country

In [6]:

aggregating**=**df**.**groupby("Country/Region")**.**sum()

In [7]:

aggregating**.**head() Out[7]:

**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 ... 4/21/20 4/22/2**

**Country/Region**

**Afghanistan** 0 0 0 0 0 0 0 0 0 0 ... 1092 117

**Albania** 0 0 0 0 0 0 0 0 0 0 ... 609 63

**Algeria** 0 0 0 0 0 0 0 0 0 0 ... 2811 291

**Andorra** 0 0 0 0 0 0 0 0 0 0 ... 717 72

**Angola** 0 0 0 0 0 0 0 0 0 0 ... 24 2

5 rows × 100 columns

In [8]:

aggregating**.**shape

Out[8]:

(187, 100)

## Visualizing data related to a country for example China

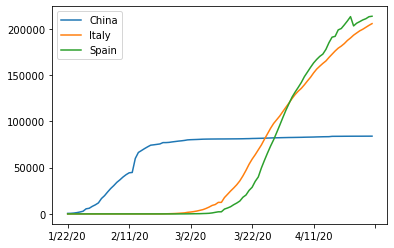
visualization always helps for better understanding of our data.

In [9]:

aggregating**.**loc["China"]**.**plot() aggregating**.**loc["Italy"]**.**plot() aggregating**.**loc["Spain"]**.**plot() plt**.**legend()

Out[9]:

<matplotlib.legend.Legend at 0x7f482e1e3990>



## Calculating a good measure

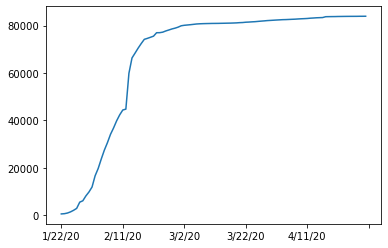
we need to find a good measure reperestend as a number, describing the spread of the virus in a country.

In [10]:

aggregating**.**loc['China']**.**plot()

Out[10]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482df94d90>

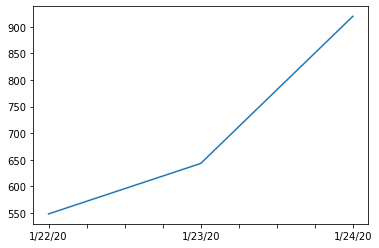


In [11]:

aggregating**.**loc['China'][:3]**.**plot()

Out[11]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482df83990>



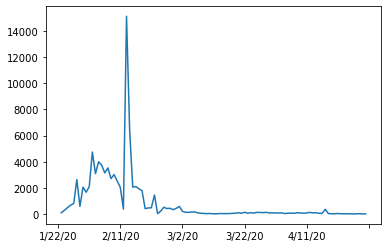
## caculating the first derivative of the curve

In [12]:

aggregating**.**loc['China']**.**diff()**.**plot()

Out[12]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482df09290>



## find maxmimum infection rate for China

In [13]:

aggregating**.**loc['China']**.**diff()**.**max()

Out[13]:

15136.0 In [14]:

aggregating**.**loc['Italy']**.**diff()**.**max()

Out[14]:

6557.0 In [15]:

aggregating**.**loc['Spain']**.**diff()**.**max()

Out[15]:

9630.0

**find maximum infection rate for all of the countries.**

In [16]:

countries**=**list(aggregating**.**index) max\_infection\_rates**=**[] **for** c **in** countries:

max\_infection\_rates**.**append(aggregating**.**loc[c]**.**diff()**.**max()) aggregating["max\_infection\_rates"]**=**max\_infection\_rates

In [17]:

aggregating**.**head()

Out[17]:

**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 ... 4/22/20 4/23/2**

**Country/Region**

**Afghanistan** 0 0 0 0 0 0 0 0 0 0 ... 1176 127

**Albania** 0 0 0 0 0 0 0 0 0 0 ... 634 66

**Algeria** 0 0 0 0 0 0 0 0 0 0 ... 2910 300

**Andorra** 0 0 0 0 0 0 0 0 0 0 ... 723 72

**Angola** 0 0 0 0 0 0 0 0 0 0 ... 25 2

5 rows × 101 columns

## create a new dataframe with only needed column

In [18]: data**=**pd**.**DataFrame(aggregating["max\_infection\_rates"])

In [19]: data**.**head() Out[19]:

**max\_infection\_rates Country/Region**

**Afghanistan** 232.0

**Albania** 34.0

**Algeria** 199.0

**Andorra** 43.0

**Angola** 5.0

## Task 4.1 : importing the dataset

In [20]: happiness**=**pd**.**read\_csv("../input/covid19/worldwide\_happiness\_report.csv") In [21]:

happiness**.**head() Out[21]:

**Overall Country or GDP per Social Healthy life Freedom to make Perceptions**

**Score Generosity**

**rank region capita support expectancy life choices of corruption**

1. 1 Finland 7.769 1.340 1.587 0.986 0.596 0.153 0.393
2. 2 Denmark 7.600 1.383 1.573 0.996 0.592 0.252 0.410
3. 3 Norway 7.554 1.488 1.582 1.028 0.603 0.271 0.341
4. 4 Iceland 7.494 1.380 1.624 1.026 0.591 0.354 0.118
5. 5 Netherlands 7.488 1.396 1.522 0.999 0.557 0.322 0.298

## Let's drop the useless columns

In [22]:

cols**=**["Overall rank","Score","Generosity","Perceptions of corruption"] In [23]:

happiness**.**drop(cols,axis**=**1,inplace**=True**) happiness**.**head()

Out[23]:

**GDP per Social Healthy life Freedom to make life**

**Country or region**

**capita support expectancy choices**

1. Finland 1.340 1.587 0.986 0.596
2. Denmark 1.383 1.573 0.996 0.592
3. Norway 1.488 1.582 1.028 0.603
4. Iceland 1.380 1.624 1.026 0.591
5. Netherlands 1.396 1.522 0.999 0.557

## changing the indices of the dataframe

In [24]:

happiness**.**set\_index("Country or region",inplace**=True**) happiness**.**head() Out[24]:

**GDP per Social Healthy life Freedom to make life**

**capita support expectancy 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

## now let's join two dataset we have prepared

**Corona Dataset :**

In [25]: data**.**head() Out[25]:

**max\_infection\_rates Country/Region**

**Afghanistan** 232.0

**Albania** 34.0

**Algeria** 199.0

**Andorra** 43.0

**Angola** 5.0

**wolrd happiness report Dataset :**

In [26]:

happiness**.**head()

Out[26]:

**GDP per Social Healthy life Freedom to make life**

**capita support expectancy 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 [27]:

final**=**data**.**join(happiness,how**=**"inner") final**.**head() Out[27]:

**GDP per Social Healthy life Freedom to make life**

**max\_infection\_rates**

**capita support expectancy 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

## correlation matrix

In [28]:

final**.**corr() Out[28]:

**GDP per Social Healthy life Freedom to make life**

**max\_infection\_rates**

**capita support expectancy choices**

**max\_infection\_rates** 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

**Freedom to make life**

0.078196 0.394603 0.456246 0.427892 1.000000

**choices**

## 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 [29]:

final**.**head()

Out[29]:

**GDP per Social Healthy life Freedom to make life**

**max\_infection\_rates**

**capita support expectancy 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

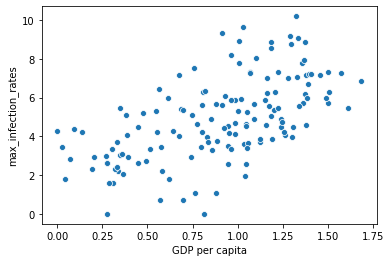
## Plotting GDP vs maximum Infection rate

In [30]:

x**=**final["GDP per capita"] y**=**final["max\_infection\_rates"] sns**.**scatterplot(x,np**.**log(y))

Out[30]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482de36590>

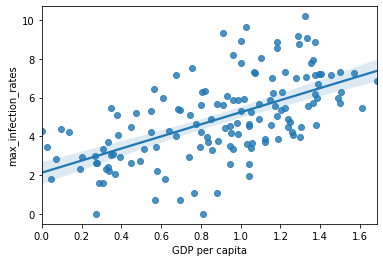


In [31]:

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

Out[31]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482dd8b3d0>

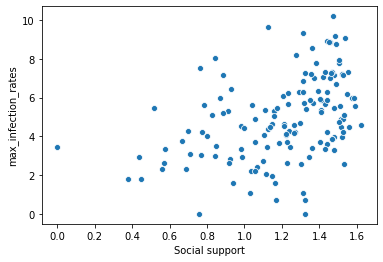


## Plotting Social support vs maximum Infection rate

In [32]:

x**=**final["Social support"] y**=**final["max\_infection\_rates"] sns**.**scatterplot(x,np**.**log(y)) Out[32]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482de1b210>

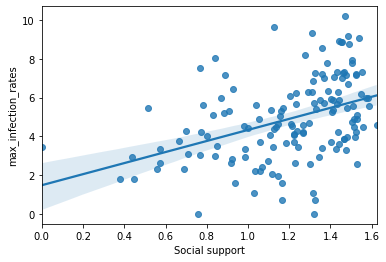


In [33]:

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

Out[33]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482b49a610>



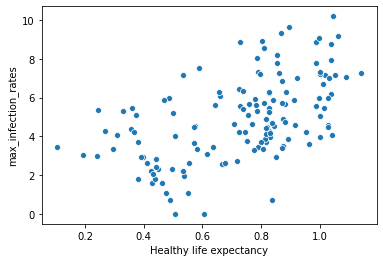
## Plotting Healthy life expectancy vs maximum Infection rate

In [34]:

x**=**final["Healthy life expectancy"] y**=**final["max\_infection\_rates"] sns**.**scatterplot(x,np**.**log(y))

Out[34]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482b3d8650>

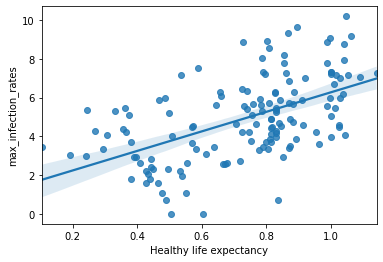


In [35]:

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

Out[35]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482b3be950>



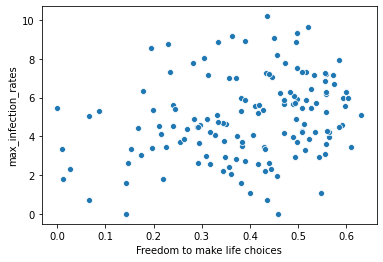
## Plotting Freedom to make life choices vs maximum Infection rate

In [36]:

x**=**final["Freedom to make life choices"] y**=**final["max\_infection\_rates"] sns**.**scatterplot(x,np**.**log(y))

Out[36]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482b328c90>

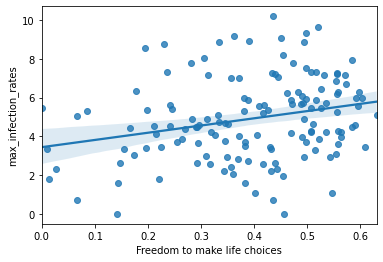


In [37]:

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

Out[37]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f482b2a2450>



**Conclusion:**

Summarize the key findings and conclusions of the COVID-19 data analytics project.

Highlight significant trends, correlations, or insights discovered during the analysis.

Discuss the implications of these findings for public health strategies or policy decisions.

Provide recommendations for future data analytics projects in the context of pandemics.

**THANK YOU**