COVID 19 Data Analysis

Fahim Alam

16th July, 2020

Abstract

The aim of the project is to illustrate how to pre-process and merge datasets to calculate needed measures and prepare them for an analysis. We are going to use COVID19 dataset, prepared by John Hopkins University, which consist of the data related to cumulative number of confirmed cases, per day, in each country. In addition, we also have a different dataset that consist of various life factors. We are going to merge both datas and try to establish relationship between them.

Covid19 Data Analysis

Let's Import the modules

First we import modules that we need for the whole analysis,

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing datas Using read csv method from pandas we can import the data set and to show some of the data, we use head method.

[2]:
```
corona_dataset_csv = pd.read_csv('covid19_Confirmed_dataset.csv')
corona_dataset_csv.head(10)
```

[2]:

	Province/State	Country/Region	Lat	Long	\
0	NaN	Afghanistan	33.0000	65.0000	
1	NaN	Albania	41.1533	20.1683	
2	NaN	Algeria	28.0339	1.6596	
3	NaN	Andorra	42.5063	1.5218	
4	NaN	Angola	-11.2027	17.8739	
5	NaN	Antigua and Barbuda	17.0608	-61.7964	
6	NaN	Argentina	-38.4161	-63.6167	
7	NaN	Armenia	40.0691	45.0382	
8	Australian Capital Territory	Australia	-35.4735	149.0124	
9	New South Wales	Australia	-33.8688	151.2093	

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	4/21/20	\

```
0        0        0        0        0        0        0  ...     1092
1        0        0        0        0        0        0  ...      609
2        0        0        0        0        0        0  ...     2811
3        0        0        0        0        0        0  ...      717
4        0        0        0        0        0        0  ...       24
5        0        0        0        0        0        0  ...       23
6        0        0        0        0        0        0  ...     3031
7        0        0        0        0        0        0  ...     1401
8        0        0        0        0        0        0  ...      104
9        0        0        0        0        3        4  ...     2969

   4/22/20  4/23/20  4/24/20  4/25/20  4/26/20  4/27/20  4/28/20  4/29/20  \
0     1176     1279     1351     1463     1531     1703     1828     1939
1      634      663      678      712      726      736      750      766
2     2910     3007     3127     3256     3382     3517     3649     3848
3      723      723      731      738      738      743      743      743
4       25       25       25       25       26       27       27       27
5       24       24       24       24       24       24       24       24
6     3144     3435     3607     3780     3892     4003     4127     4285
7     1473     1523     1596     1677     1746     1808     1867     1932
8      104      104      105      106      106      106      106      106
9     2971     2976     2982     2994     3002     3004     3016     3016

   4/30/20
0     2171
1      773
2     4006
3      745
4       27
5       24
6     4428
7     2066
8      106
9     3025

[10 rows x 104 columns]
```

Let's check the shape of the dataframe

[4]:
```
corona_dataset_csv.shape
```

[4]:
```
(266, 104)
```

Remove the useless columns When we imported our datas, we saw that some of the columns were not really important for our study, so we intend to drop them. And to replace the current data set with the new data set, we use inplace = "True".

[5]:
```
corona_dataset_csv.drop(['Lat','Long'],axis=1,inplace=True)
```

[6]:
```
corona_dataset_csv.head(5)
```

[6]:
```
  Province/State Country/Region  1/22/20  1/23/20  1/24/20  1/25/20  1/26/20  \
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

   1/27/20  1/28/20  1/29/20  ...  4/21/20  4/22/20  4/23/20  4/24/20  \
0        0        0        0  ...     1092     1176     1279     1351
1        0        0        0  ...      609      634      663      678
2        0        0        0  ...     2811     2910     3007     3127
3        0        0        0  ...      717      723      723      731
4        0        0        0  ...       24       25       25       25

   4/25/20  4/26/20  4/27/20  4/28/20  4/29/20  4/30/20
0     1463     1531     1703     1828     1939     2171
1      712      726      736      750      766      773
2     3256     3382     3517     3649     3848     4006
3      738      738      743      743      743      745
4       25       26       27       27       27       27

[5 rows x 102 columns]
```

Aggregating the rows by the country

We can use groupby method to aggregate the rows by the countries with the help of sum method.

[11]:
```
corona_dataset_aggregated = corona_dataset_csv.groupby("Country/Region").sum()
```

[12]:
```
corona_dataset_aggregated.head(5)
```

[12]:
```
                1/22/20  1/23/20  1/24/20  1/25/20  1/26/20  1/27/20  1/28/20  \
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

                1/29/20  1/30/20  1/31/20  ...  4/21/20  4/22/20  4/23/20  \
Country/Region                             ...
Afghanistan           0        0        0  ...     1092     1176     1279
Albania               0        0        0  ...      609      634      663
```

```
Algeria                0        0        0  ...     2811     2910     3007
Andorra                0        0        0  ...      717      723      723
Angola                 0        0        0  ...       24       25       25

                4/24/20  4/25/20  4/26/20  4/27/20  4/28/20  4/29/20  4/30/20
Country/Region
Afghanistan        1351     1463     1531     1703     1828     1939     2171
Albania             678      712      726      736      750      766      773
Algeria            3127     3256     3382     3517     3649     3848     4006
Andorra             731      738      738      743      743      743      745
Angola               25       25       26       27       27       27       27

[5 rows x 100 columns]
```

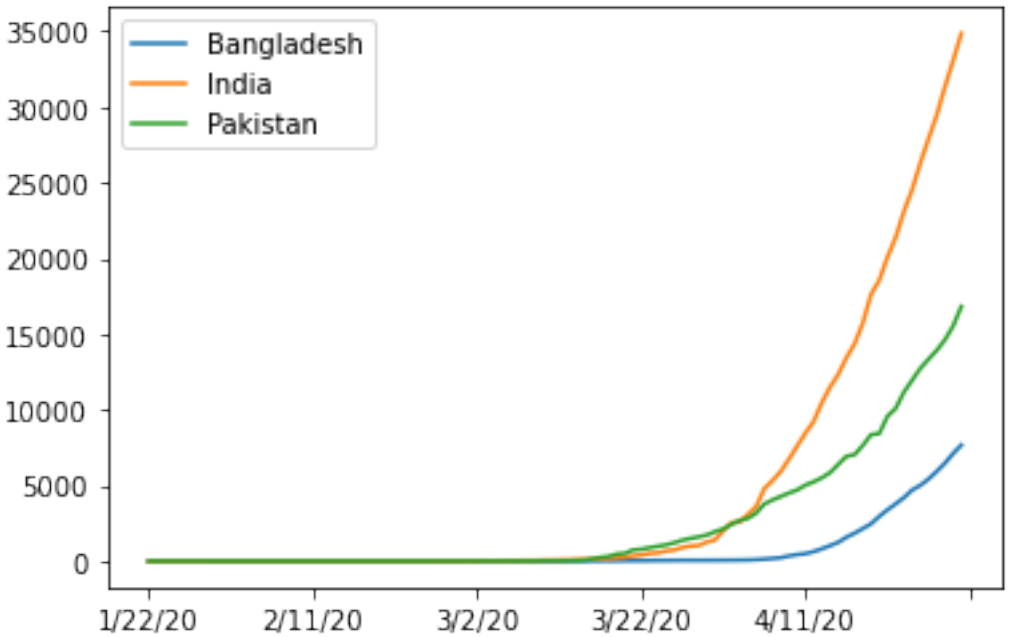
[13]:
```
corona_dataset_aggregated.shape
```

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

Visualizing data related to some specific countries

[15]:
```
corona_dataset_aggregated.loc['Bangladesh'].plot()
corona_dataset_aggregated.loc['India'].plot()
corona_dataset_aggregated.loc['Pakistan'].plot()
plt.legend()
```

[15]:
```
<matplotlib.legend.Legend at 0x212dbd7c188>
```

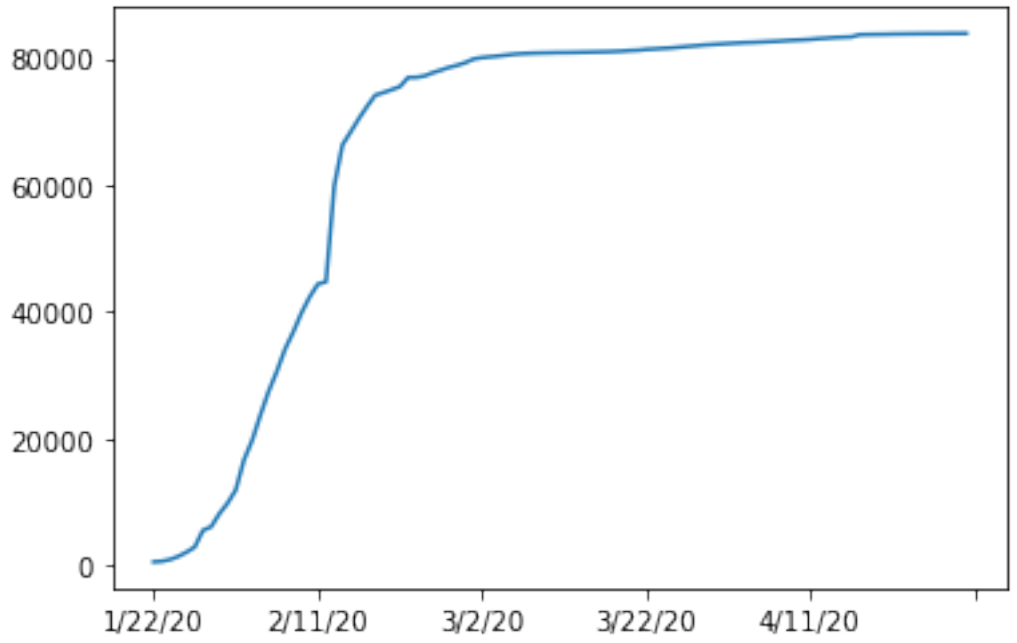


Calculating a good measure

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

[16]:
```
corona_dataset_aggregated.loc['China'].plot()
```

[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x212dbdfdfc8>
```

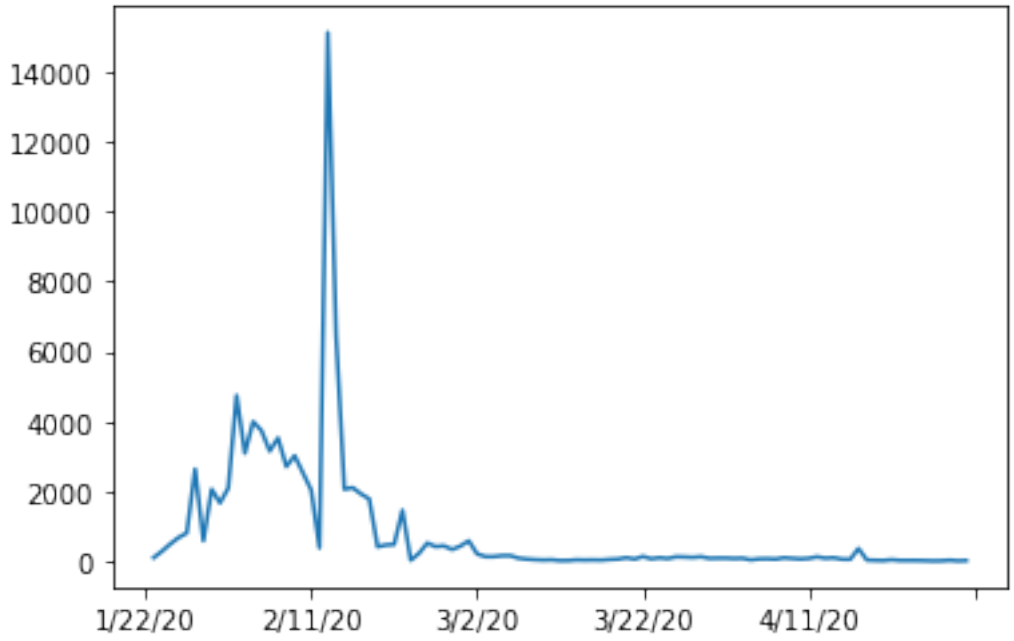


We can see that the affected numbers began to rise after 3/2/2020. To find the rate,we can calculate the derivative.

caculating the first derivative of the curve

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

[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x212dbe63648>
```



Maximum infection rate for China

In 24 hours, the maximum infection rate for China can be measured using max method.

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

[18]: 15136.0

[19]:
```
corona_dataset_aggregated.loc['Bangladesh'].diff().max()
```

[19]: 641.0

Maximum infection rate for all of the countries.

[20]:
```
countries = list(corona_dataset_aggregated.index)
max_infection_rates = []
for country in countries :
    max_infection_rates.append(corona_dataset_aggregated.loc[country].diff().
 ↪max())
corona_dataset_aggregated['max infection rate'] = max_infection_rates
```

[21]:
```
corona_dataset_aggregated.head()
```

[21]:
```
                1/22/20  1/23/20  1/24/20  1/25/20  1/26/20  1/27/20  1/28/20  \
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

                1/29/20  1/30/20  1/31/20  ...  4/22/20  4/23/20  4/24/20  \
Country/Region                             ...
Afghanistan           0        0        0  ...     1176     1279     1351
Albania               0        0        0  ...      634      663      678
Algeria               0        0        0  ...     2910     3007     3127
Andorra               0        0        0  ...      723      723      731
Angola                0        0        0  ...       25       25       25

                4/25/20  4/26/20  4/27/20  4/28/20  4/29/20  4/30/20  \
Country/Region
Afghanistan        1463     1531     1703     1828     1939     2171
Albania             712      726      736      750      766      773
Algeria            3256     3382     3517     3649     3848     4006
Andorra             738      738      743      743      743      745
Angola               25       26       27       27       27       27

                max infection rate
Country/Region
Afghanistan                  232.0
Albania                       34.0
Algeria                      199.0
Andorra                       43.0
Angola                         5.0

[5 rows x 101 columns]
```

Create a new dataframe with only needed column

[23]:
```
corona_data = pd.DataFrame(corona_dataset_aggregated['max infection rate'])
```

[25]:
```
corona_data.tail()
```

[25]:
```
                    max infection rate
Country/Region
West Bank and Gaza                66.0
Western Sahara                     4.0
Yemen                              5.0
Zambia                             9.0
Zimbabwe                           8.0
```

Now we will focus on our 2nd data set and our task will be,

- Importing the WorldHappinessReport.csv dataset
- selecting needed columns for our analysis
- join the datasets
- calculate the correlations as the result of our analysis

Importing the dataset

[27]:
```
world_happiness_report = pd.read_csv("worldwide_happiness_report.csv")
world_happiness_report.head()
```

[27]:
```
   Overall rank Country or region  Score  GDP per capita  Social support  \
0             1           Finland  7.769           1.340           1.587
1             2           Denmark  7.600           1.383           1.573
2             3            Norway  7.554           1.488           1.582
3             4           Iceland  7.494           1.380           1.624
4             5       Netherlands  7.488           1.396           1.522

   Healthy life expectancy  Freedom to make life choices  Generosity  \
0                    0.986                         0.596       0.153
1                    0.996                         0.592       0.252
2                    1.028                         0.603       0.271
3                    1.026                         0.591       0.354
4                    0.999                         0.557       0.322

   Perceptions of corruption
0                      0.393
1                      0.410
2                      0.341
3                      0.118
4                      0.298
```

[28]:
```
world_happiness_report.shape
```

[28]:
```
(156, 9)
```

Let's drop the useless columns

[29]:
```
columns_to_dropped = ['Overall rank','Score','Generosity','Perceptions of␣
 ↪corruption']
world_happiness_report.drop(columns_to_dropped,axis=1 , inplace=True)
```

[30]:
```
world_happiness_report.head()
```

[30]:
```
  Country or region  GDP per capita  Social support  Healthy life expectancy  \
0           Finland           1.340           1.587                    0.986
1           Denmark           1.383           1.573                    0.996
2            Norway           1.488           1.582                    1.028
```

```
3           Iceland           1.380           1.624                     1.026
4       Netherlands           1.396           1.522                     0.999

   Freedom to make life choices
0                         0.596
1                         0.592
2                         0.603
3                         0.591
4                         0.557
```

Changing the indices of the dataframe

[31]:
```
world_happiness_report.set_index(['Country or region'],inplace=True)
world_happiness_report.head()
```

[31]:
```
                  GDP per capita  Social support  Healthy life expectancy  \
Country or region
Finland                     1.340           1.587                    0.986
Denmark                     1.383           1.573                    0.996
Norway                      1.488           1.582                    1.028
Iceland                     1.380           1.624                    1.026
Netherlands                 1.396           1.522                    0.999

                  Freedom to make life choices
Country or region
Finland                                   0.596
Denmark                                   0.592
Norway                                    0.603
Iceland                                   0.591
Netherlands                               0.557
```

Merge the datasets:

Corona Dataset :

[32]:
```
corona_data.head()
```

[32]:
```
                max infection rate
Country/Region
Afghanistan                  232.0
Albania                       34.0
Algeria                      199.0
Andorra                       43.0
Angola                         5.0
```

World happiness report Dataset :

[33]:
```
world_happiness_report.head()
```

[33]:
```
                   GDP per capita  Social support  Healthy life expectancy  \
Country or region
Finland                     1.340           1.587                    0.986
Denmark                     1.383           1.573                    0.996
Norway                      1.488           1.582                    1.028
Iceland                     1.380           1.624                    1.026
Netherlands                 1.396           1.522                    0.999

                   Freedom to make life choices
Country or region
Finland                                   0.596
Denmark                                   0.592
Norway                                    0.603
Iceland                                   0.591
Netherlands                               0.557
```

[34]:
```
data = world_happiness_report.join(corona_data).copy()
data.head()
```

[34]:
```
                   GDP per capita  Social support  Healthy life expectancy  \
Country or region
Finland                     1.340           1.587                    0.986
Denmark                     1.383           1.573                    0.996
Norway                      1.488           1.582                    1.028
Iceland                     1.380           1.624                    1.026
Netherlands                 1.396           1.522                    0.999

                   Freedom to make life choices  max infection rate
Country or region
Finland                                   0.596               267.0
Denmark                                   0.592               391.0
Norway                                    0.603               386.0
Iceland                                   0.591                99.0
Netherlands                               0.557              1346.0
```

Correlation matrix

[35]:
```
data.corr()
# it is representing the currelation between every two columns of our dataset
```

[35]:
```
                            GDP per capita  Social support  \
GDP per capita                    1.000000        0.754906
Social support                    0.754906        1.000000
Healthy life expectancy           0.835462        0.719009
Freedom to make life choices      0.379079        0.447333
```

```
max infection rate                  0.250118        0.191958

                              Healthy life expectancy  \
GDP per capita                               0.835462
Social support                               0.719009
Healthy life expectancy                      1.000000
Freedom to make life choices                 0.390395
max infection rate                           0.289263

                              Freedom to make life choices  max infection rate
GDP per capita                                    0.379079            0.250118
Social support                                    0.447333            0.191958
Healthy life expectancy                           0.390395            0.289263
Freedom to make life choices                      1.000000            0.078196
max infection rate                                0.078196            1.000000
```

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.

[36]:
```
data.head()
```

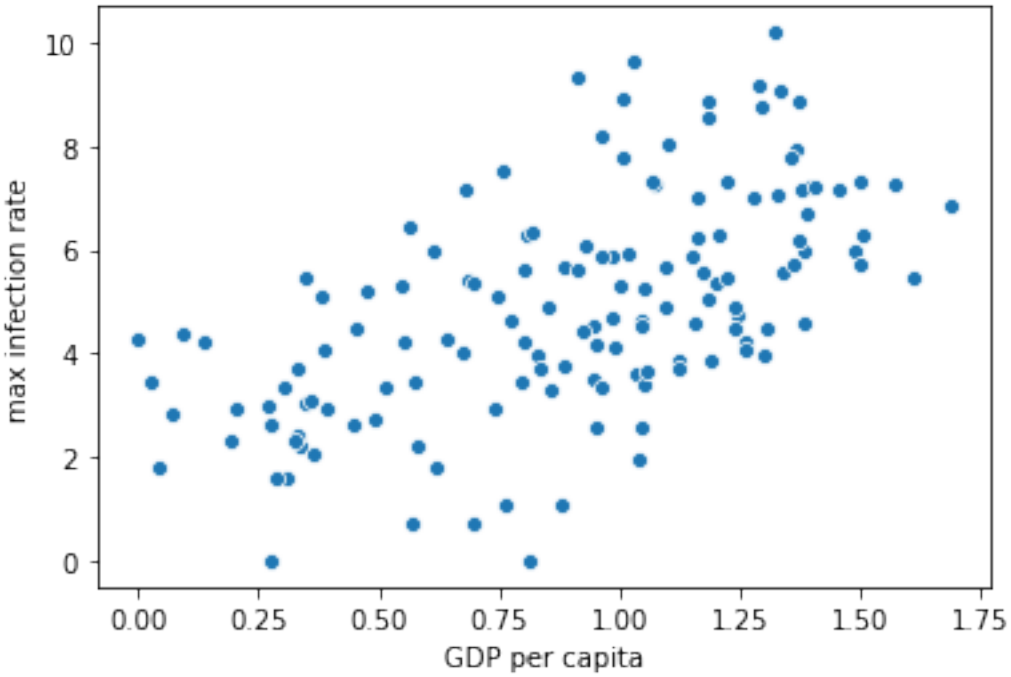
[36]:
```
                    GDP per capita  Social support  Healthy life expectancy  \
Country or region
Finland                      1.340           1.587                    0.986
Denmark                      1.383           1.573                    0.996
Norway                       1.488           1.582                    1.028
Iceland                      1.380           1.624                    1.026
Netherlands                  1.396           1.522                    0.999

                    Freedom to make life choices  max infection rate
Country or region
Finland                                    0.596               267.0
Denmark                                    0.592               391.0
Norway                                     0.603               386.0
Iceland                                    0.591                99.0
Netherlands                                0.557              1346.0
```

Plotting GDP vs maximum Infection rate

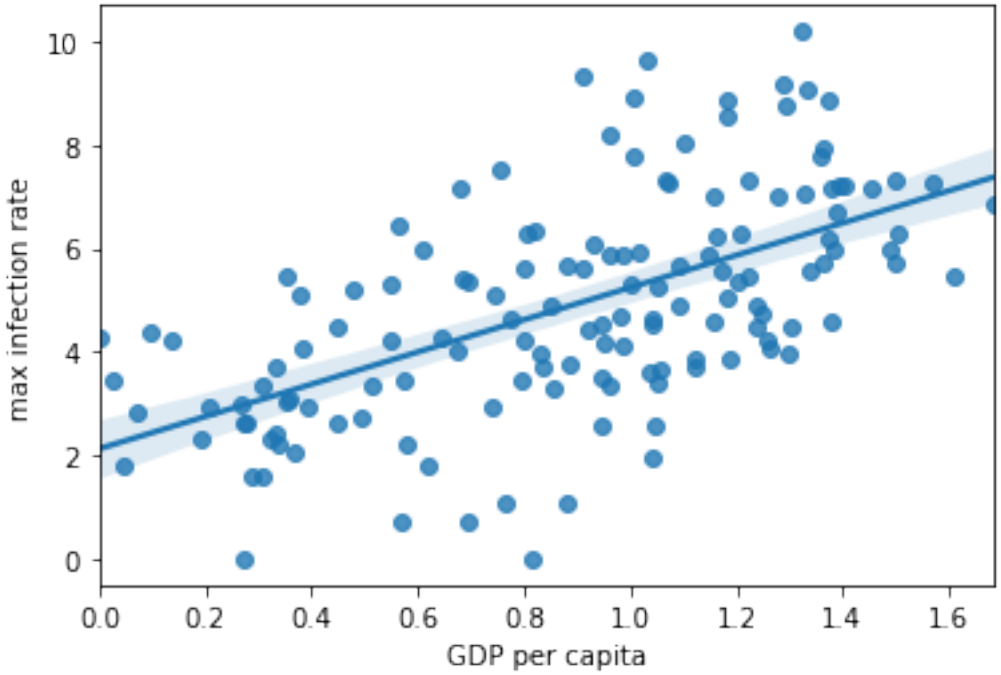
[37]:
```
x = data['GDP per capita']
y = data['max infection rate']
sns.scatterplot(x,np.log(y))
```

[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x212dbeefec8>
```



[38]:
```
sns.regplot(x,np.log(y))
```

[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x212dbfc23c8>
```



Result:

We have found a very interesting result in this analysis. We found that people living in developed countries (more GDP cases) are prone to affect by the virus more.