```
#import the librarries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

#read the dataset
dataset=pd.read_csv("C:\\Users\\amitk\\OneDrive\\Desktop\\
covid19_Confirmed_dataset.csv")
dataset.head()

  Province/State Country/Region      Lat     Long  1/22/20  1/23/20
1/24/20  \
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

   1/25/20  1/26/20  1/27/20  ...  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 104 columns]

dataset.shape

(266, 104)
```

```
#delete the useless column
df=dataset.drop(["Lat","Long"],axis=1,inplace=True)
```

```
dataset.head()
```

```
  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]
```

```
#aggregate the rows by the country
corona_dataset_aggregated=dataset.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  \
Country/Region                                                         

Afghanistan           0        0        0        0        0        0        
```

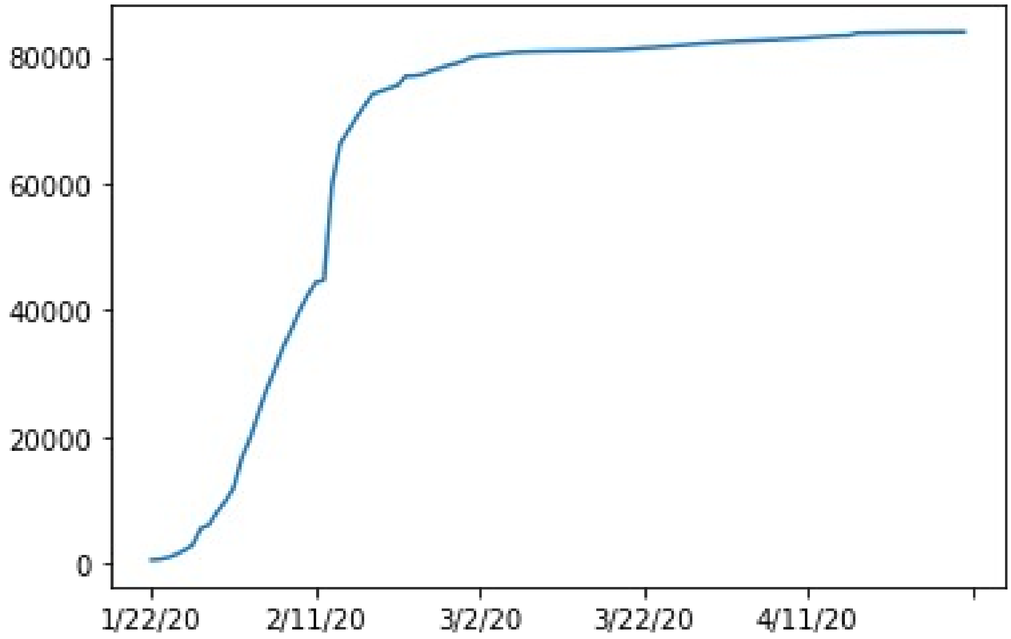
```
0
Albania               0        0        0        0        0        0
0
Algeria               0        0        0        0        0        0
0
Andorra               0        0        0        0        0        0
0
Angola                0        0        0        0        0        0
0

                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]
```

```
corona_dataset_aggregated.shape
```

```
(187, 100)
```

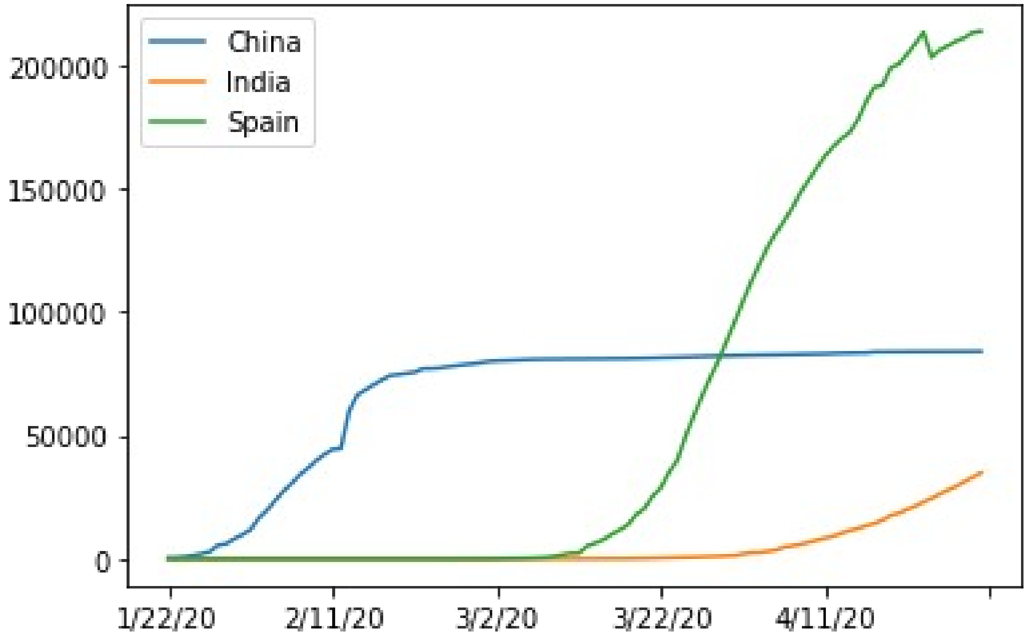
```
#visualize data related to a country
corona_dataset_aggregated.loc["China"].plot()
```

```
<AxesSubplot:>
```



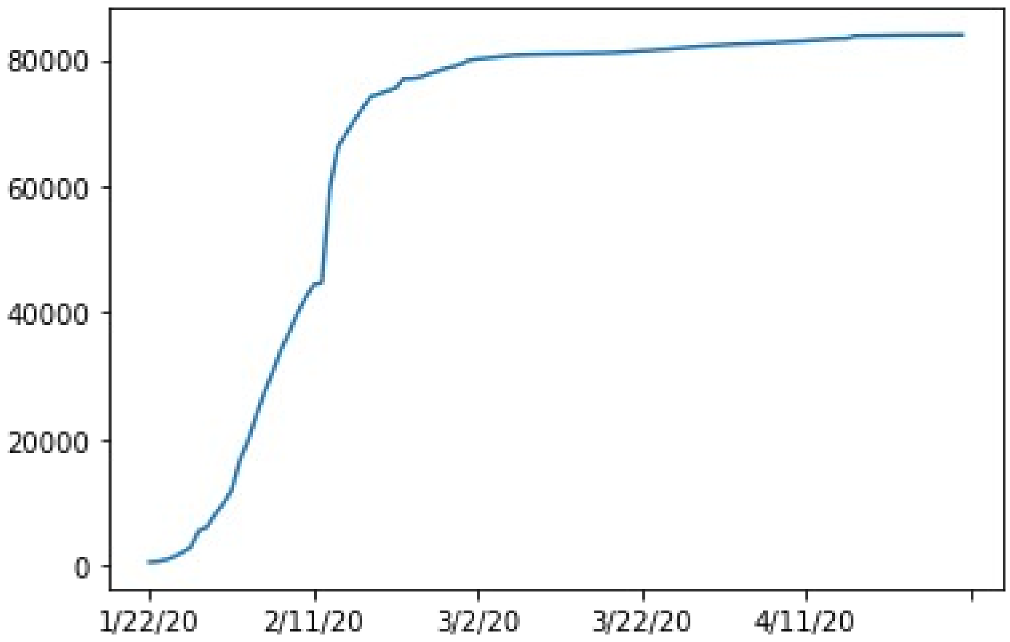
```
#visualize data related to a country
corona_dataset_aggregated.loc["China"].plot()
corona_dataset_aggregated.loc["India"].plot()
corona_dataset_aggregated.loc["Spain"].plot()
plt.legend()
```

```
<matplotlib.legend.Legend at 0x24cd52cefa0>
```



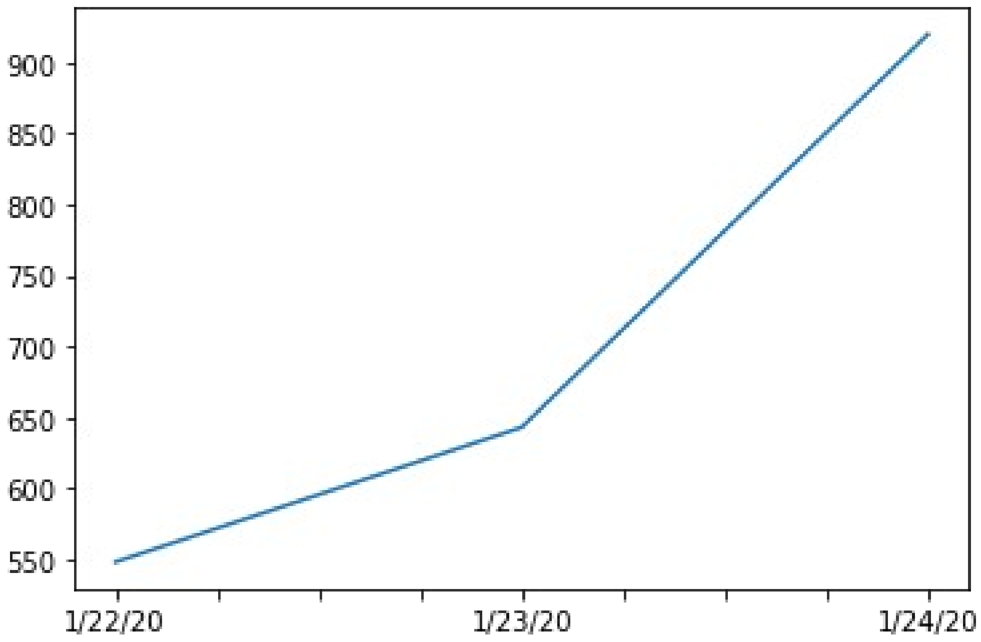
```
#calculate a good measure
corona_dataset_aggregated.loc["China"].plot()
```

```
<AxesSubplot:>
```



```
#corona virus spread in china in 3 days
corona_dataset_aggregated.loc["China"][:3].plot()
```

```
<AxesSubplot:>
```

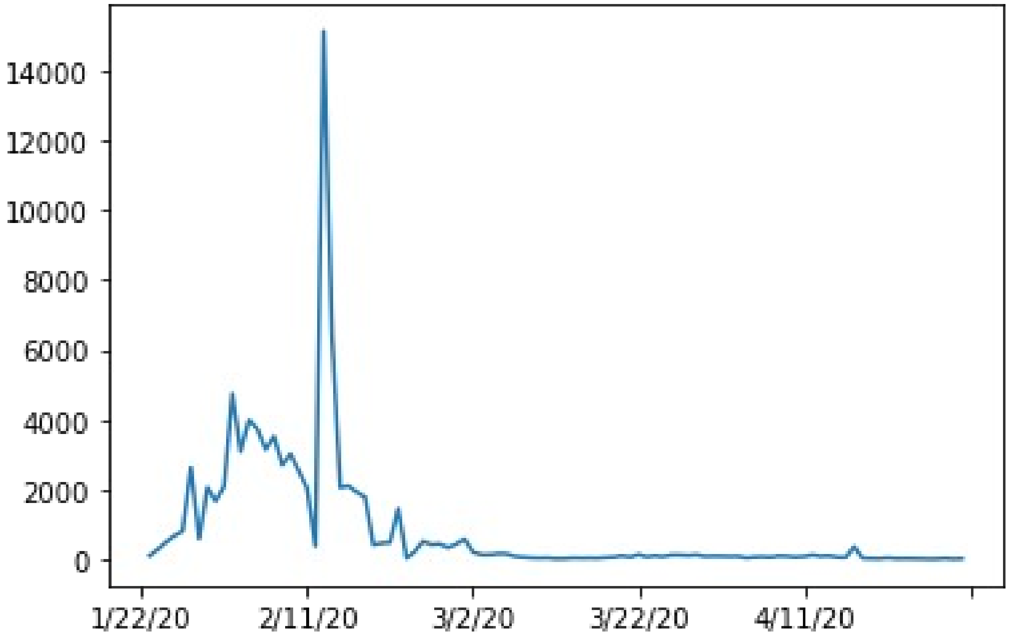


```
#calculate the first derivative of the curve

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

```
1/22/20      NaN
1/23/20     95.0
1/24/20    277.0
1/25/20    486.0
1/26/20    669.0
           ...
4/26/20      3.0
4/27/20      6.0
4/28/20     22.0
4/29/20      4.0
4/30/20     12.0
Name: China, Length: 100, dtype: float64
```

```
#calculate the first derivative of the curve

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

```
<AxesSubplot:>
```



```
#maximum infection rate
corona_dataset_aggregated.loc["China"].diff().max()
```

```
15136.0
```

```
#maximum infection rate

corona_dataset_aggregated.loc["India"].max()
```

```
34863
```

```
#maximum infection rate

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

```
9630.0
```

```
# for all countries infection rate
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_rates"]=max_infection_rates
```

```
#to print all the infection rate
corona_dataset_aggregated
```

```
                    1/22/20  1/23/20  1/24/20  1/25/20  1/26/20
1/27/20  \
Country/Region

Afghanistan               0        0        0        0        0
0
Albania                   0        0        0        0        0
0
Algeria                   0        0        0        0        0
0
Andorra                   0        0        0        0        0
0
Angola                    0        0        0        0        0
0
...                     ...      ...      ...      ...      ...      .
..
West Bank and Gaza        0        0        0        0        0
0
Western Sahara            0        0        0        0        0
0
Yemen                     0        0        0        0        0
0
Zambia                    0        0        0        0        0
0
Zimbabwe                  0        0        0        0        0
0


                    1/28/20  1/29/20  1/30/20  1/31/20  ...  4/22/20
4/23/20  \
Country/Region                                          ...

Afghanistan               0        0        0        0  ...     1176
1279
```

```
Albania                   0         0         0         0  ...       634
663
Algeria                   0         0         0         0  ...      2910
3007
Andorra                   0         0         0         0  ...       723
723
Angola                    0         0         0         0  ...        25
25
...                     ...       ...       ...       ...  ...       ...
...
West Bank and Gaza        0         0         0         0  ...       474
480
Western Sahara            0         0         0         0  ...         6
6
Yemen                     0         0         0         0  ...         1
1
Zambia                    0         0         0         0  ...        74
76
Zimbabwe                  0         0         0         0  ...        28
28

                    4/24/20  4/25/20  4/26/20  4/27/20  4/28/20
4/29/20  \
Country/Region

Afghanistan            1351     1463     1531     1703     1828
1939
Albania                 678      712      726      736      750
766
Algeria                3127     3256     3382     3517     3649
3848
Andorra                 731      738      738      743      743
743
Angola                   25       25       26       27       27
27
...                     ...      ...      ...      ...      ...      .
..
West Bank and Gaza      484      342      342      342      343
344
Western Sahara            6        6        6        6        6
6
Yemen                     1        1        1        1        1
6
Zambia                   84       84       88       88       95
97
Zimbabwe                 29       31       31       32       32
32

                    4/30/20  max_infection_rates
Country/Region
```

```
Afghanistan            2171                232.0
Albania                 773                 34.0
Algeria                4006                199.0
Andorra                 745                 43.0
Angola                   27                  5.0
...                     ...                  ...
West Bank and Gaza      344                 66.0
Western Sahara            6                  4.0
Yemen                     6                  5.0
Zambia                  106                  9.0
Zimbabwe                 40                  8.0

[187 rows x 101 columns]
```

```
#create a new data frame
corona_data=pd.DataFrame(corona_dataset_aggregated["max_infection_rate
s"])

corona_data
```

```
                    max_infection_rates
Country/Region
Afghanistan                       232.0
Albania                            34.0
Algeria                           199.0
Andorra                            43.0
Angola                              5.0
...                                 ...
West Bank and Gaza                 66.0
Western Sahara                      4.0
Yemen                               5.0
Zambia                              9.0
Zimbabwe                            8.0

[187 rows x 1 columns]
```

```
#importing the another  dataset

happiness_report=pd.read_csv("C:\\Users\\amitk\\OneDrive\\Desktop\\
worldwide_happiness_report.csv")

happiness_report
```

```
     Overall rank         Country or region  Score  GDP per capita  \
0               1                   Finland  7.769           1.340
1               2                   Denmark  7.600           1.383
2               3                    Norway  7.554           1.488
3               4                   Iceland  7.494           1.380
4               5               Netherlands  7.488           1.396
..            ...                       ...    ...             ...
151           152                    Rwanda  3.334           0.359
```

```
152           153                   Tanzania  3.231           0.476
153           154                Afghanistan  3.203           0.350
154           155  Central African Republic  3.083           0.026
155           156                South Sudan  2.853           0.306

     Social support  Healthy life expectancy  Freedom to make life
choices  \
0             1.587                    0.986
0.596
1             1.573                    0.996
0.592
2             1.582                    1.028
0.603
3             1.624                    1.026
0.591
4             1.522                    0.999
0.557
..              ...                      ...
...
151           0.711                    0.614
0.555
152           0.885                    0.499
0.417
153           0.517                    0.361
0.000
154           0.000                    0.105
0.225
155           0.575                    0.295
0.010

     Generosity  Perceptions of corruption
0         0.153                      0.393
1         0.252                      0.410
2         0.271                      0.341
3         0.354                      0.118
4         0.322                      0.298
..          ...                        ...
151       0.217                      0.411
152       0.276                      0.147
153       0.158                      0.025
154       0.235                      0.035
155       0.202                      0.091

[156 rows x 9 columns]
```

```
#drop the useless columns

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

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

```
  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
```

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

```
                   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
```

```
#join the dataset
```

```
corona_data.shape
```

```
(187, 1)
```

```
happiness_report.shape
```

```
(156, 4)
```

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

```
             max_infection_rates  GDP per capita  Social support  \
Afghanistan                232.0           0.350           0.517   
Albania                     34.0           0.947           0.848   
Algeria                    199.0           1.002           1.160   
Argentina                  291.0           1.092           1.432   
Armenia                    134.0           0.850           1.055   
...                          ...             ...             ...   
Venezuela                   29.0           0.960           1.427   
Vietnam                     19.0           0.741           1.346   
Yemen                        5.0           0.287           1.163   
Zambia                       9.0           0.578           1.058   
Zimbabwe                     8.0           0.366           1.114   

             Healthy life expectancy  Freedom to make life choices  
Afghanistan                    0.361                         0.000  
Albania                        0.874                         0.383  
Algeria                        0.785                         0.086  
Argentina                      0.881                         0.471  
Armenia                        0.815                         0.283  
...                              ...                           ...  
Venezuela                      0.805                         0.154  
Vietnam                        0.851                         0.543  
Yemen                          0.463                         0.143  
Zambia                         0.426                         0.431  
Zimbabwe                       0.433                         0.361  

[143 rows x 5 columns]
```

```
data.corr()
```

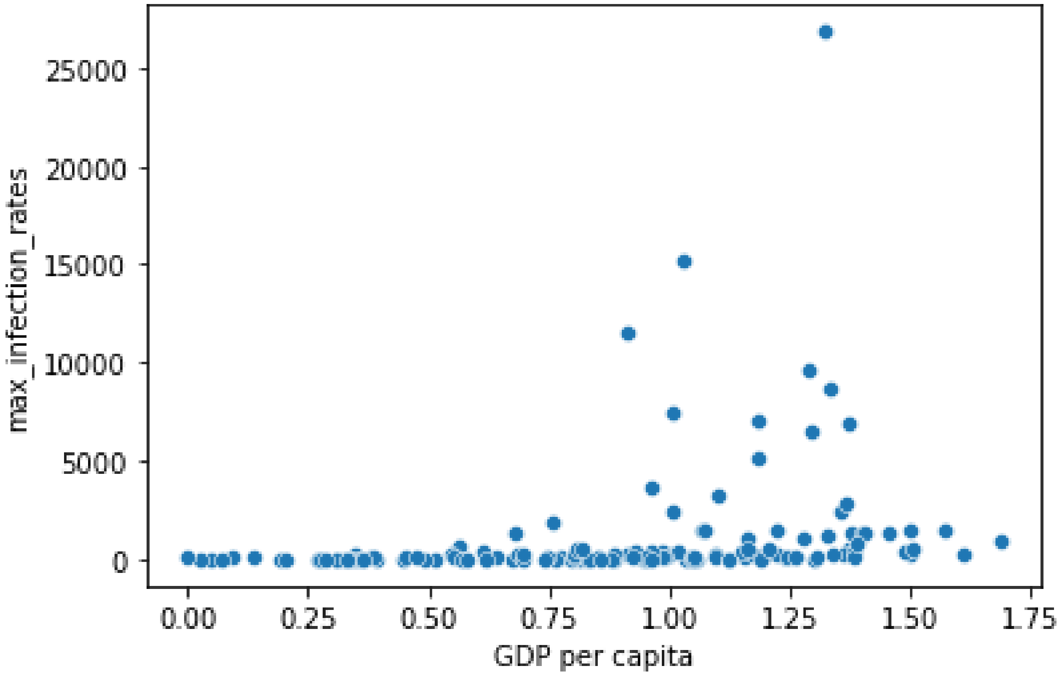
```
                              max_infection_rates  GDP per capita  \
max_infection_rates                      1.000000        0.250118   
GDP per capita                           0.250118        1.000000   
Social support                           0.191958        0.759468   
Healthy life expectancy                  0.289263        0.863062   
Freedom to make life choices             0.078196        0.394603   

                              Social support  Healthy life expectancy  
```

```
\
max_infection_rates                 0.191958                 0.289263

GDP per capita                      0.759468                 0.863062

Social support                      1.000000                 0.765286

Healthy life expectancy             0.765286                 1.000000

Freedom to make life choices        0.456246                 0.427892


                              Freedom to make life choices
max_infection_rates                               0.078196
GDP per capita                                    0.394603
Social support                                    0.456246
Healthy life expectancy                           0.427892
Freedom to make life choices                      1.000000

data

             max_infection_rates  GDP per capita  Social support  \
Afghanistan                232.0           0.350           0.517
Albania                     34.0           0.947           0.848
Algeria                    199.0           1.002           1.160
Argentina                  291.0           1.092           1.432
Armenia                    134.0           0.850           1.055
...                          ...             ...             ...
Venezuela                   29.0           0.960           1.427
Vietnam                     19.0           0.741           1.346
Yemen                        5.0           0.287           1.163
Zambia                       9.0           0.578           1.058
Zimbabwe                     8.0           0.366           1.114

             Healthy life expectancy  Freedom to make life choices
Afghanistan                    0.361                         0.000
Albania                        0.874                         0.383
Algeria                        0.785                         0.086
Argentina                      0.881                         0.471
Armenia                        0.815                         0.283
...                              ...                           ...
Venezuela                      0.805                         0.154
Vietnam                        0.851                         0.543
Yemen                          0.463                         0.143
Zambia                         0.426                         0.431
Zimbabwe                       0.433                         0.361

[143 rows x 5 columns]
```

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

```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

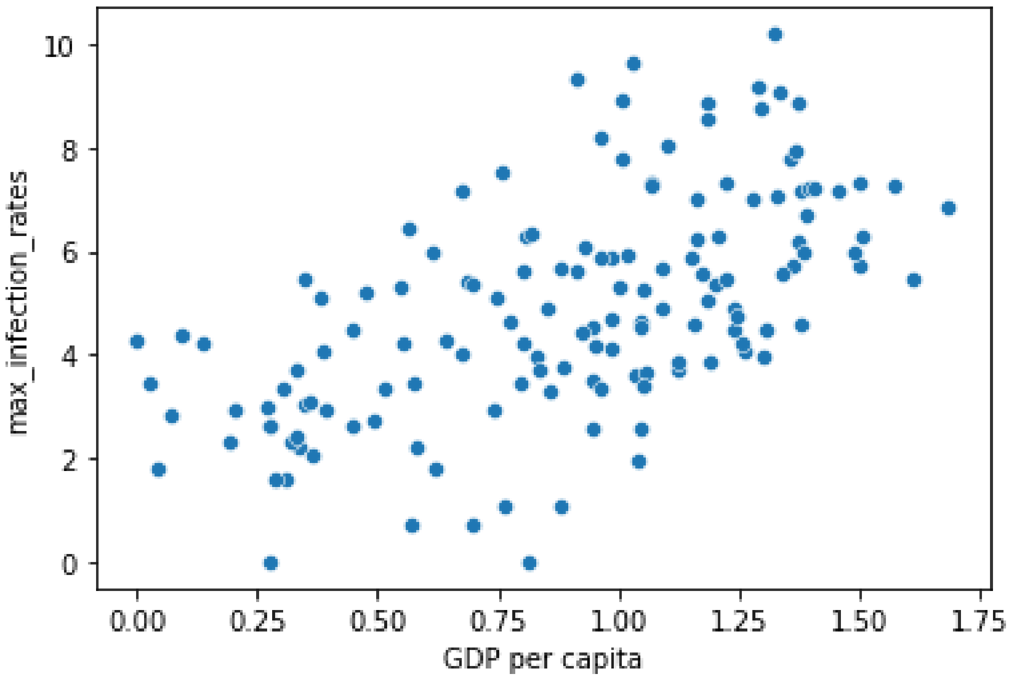
```
<AxesSubplot:xlabel='GDP per capita', ylabel='max_infection_rates'>
```



```
#visualization
x=data["GDP per capita"]
y=data["max_infection_rates"]
sns.scatterplot(x,np.log(y))
```

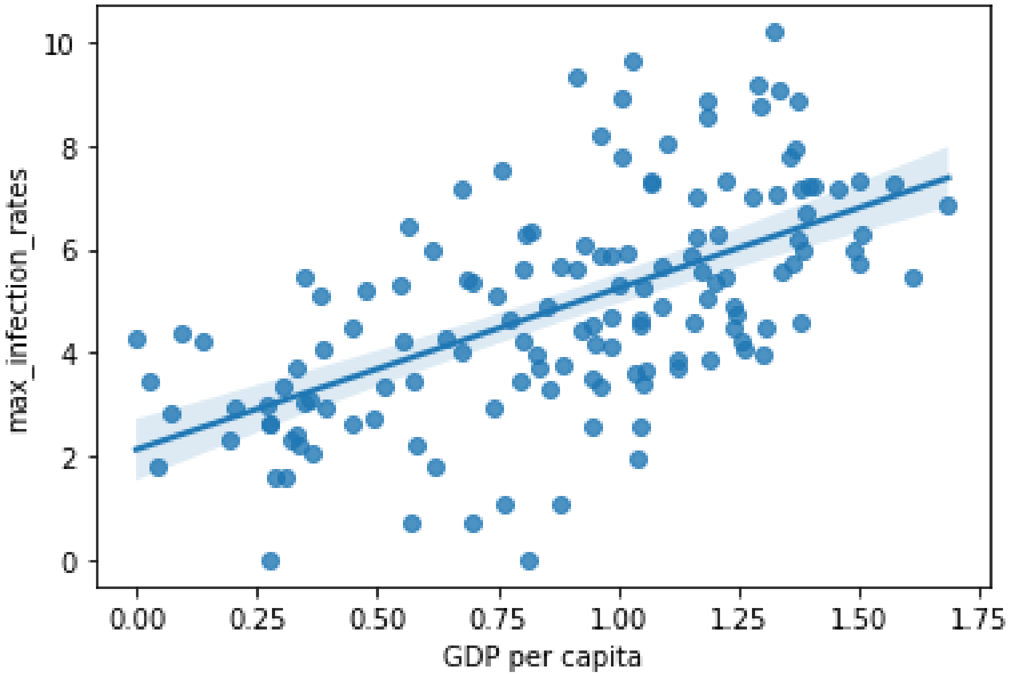
```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='GDP per capita', ylabel='max_infection_rates'>
```



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

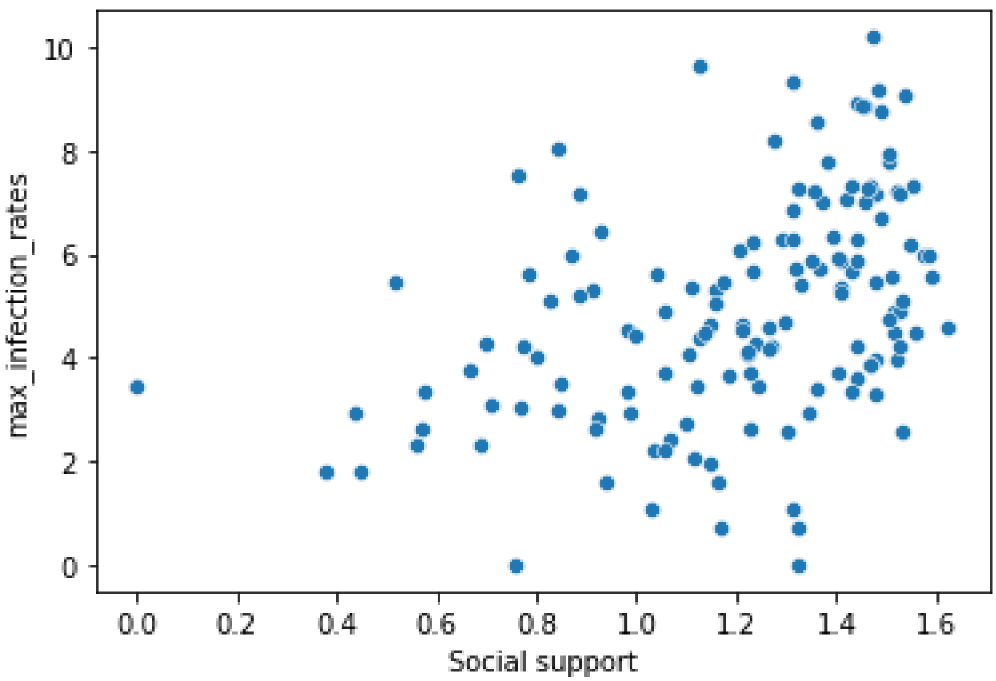
```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(

<AxesSubplot:xlabel='GDP per capita', ylabel='max_infection_rates'>
```



```
x=data["Social support"]
y=data["max_infection_rates"]
sns.scatterplot(x,np.log(y))
```

```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

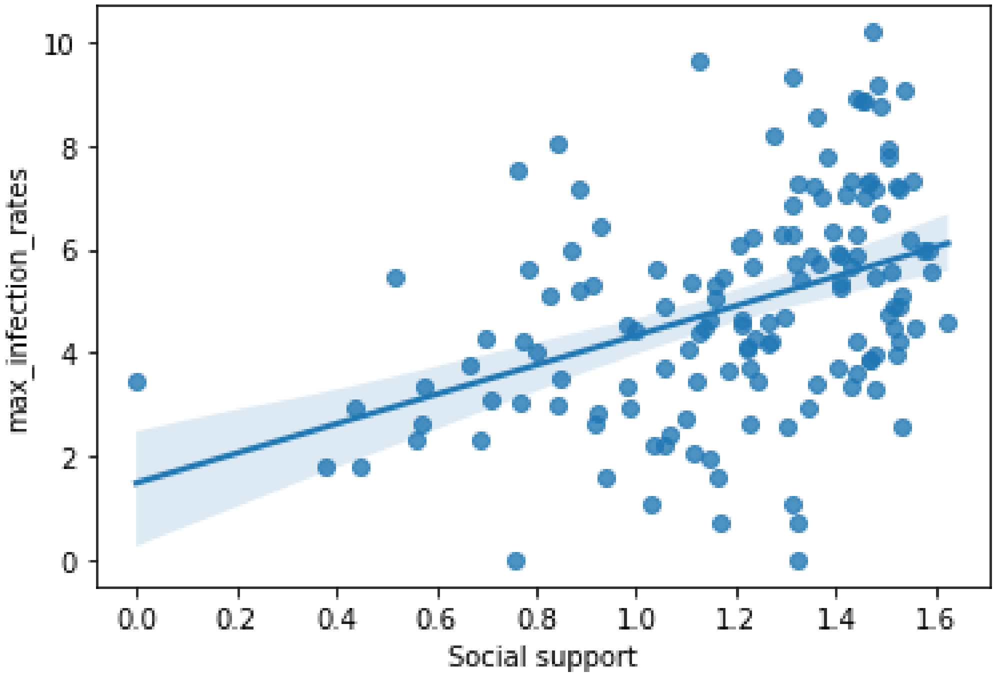
```
<AxesSubplot:xlabel='Social support', ylabel='max_infection_rates'>
```



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

```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

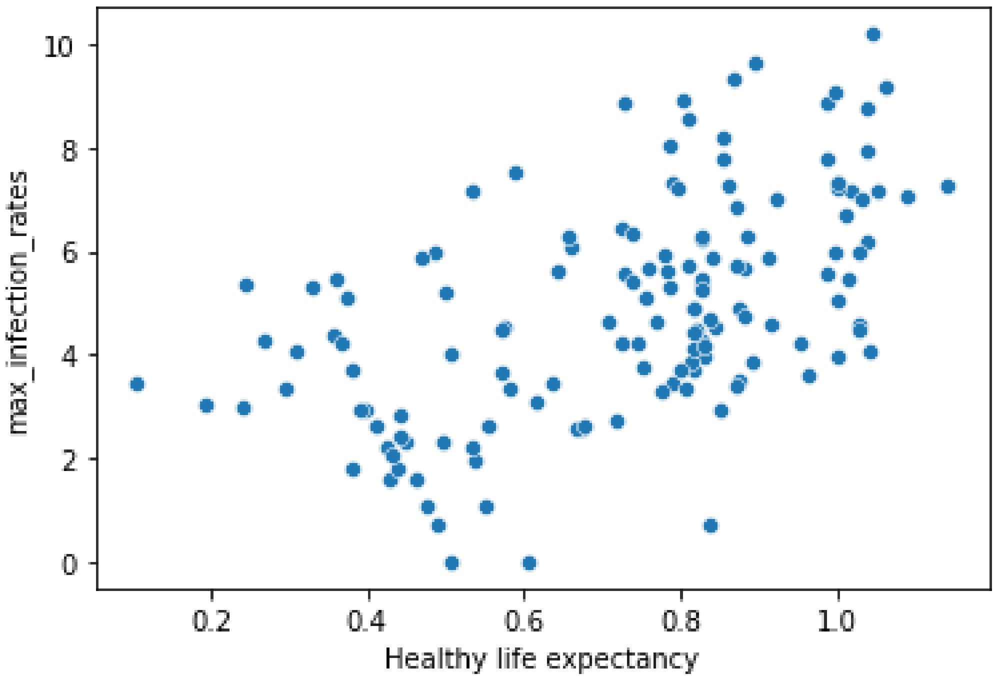
```
<AxesSubplot:xlabel='Social support', ylabel='max_infection_rates'>
```



```
x=data["Healthy life expectancy"]
y=data["max_infection_rates"]
sns.scatterplot(x,np.log(y))
```

```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

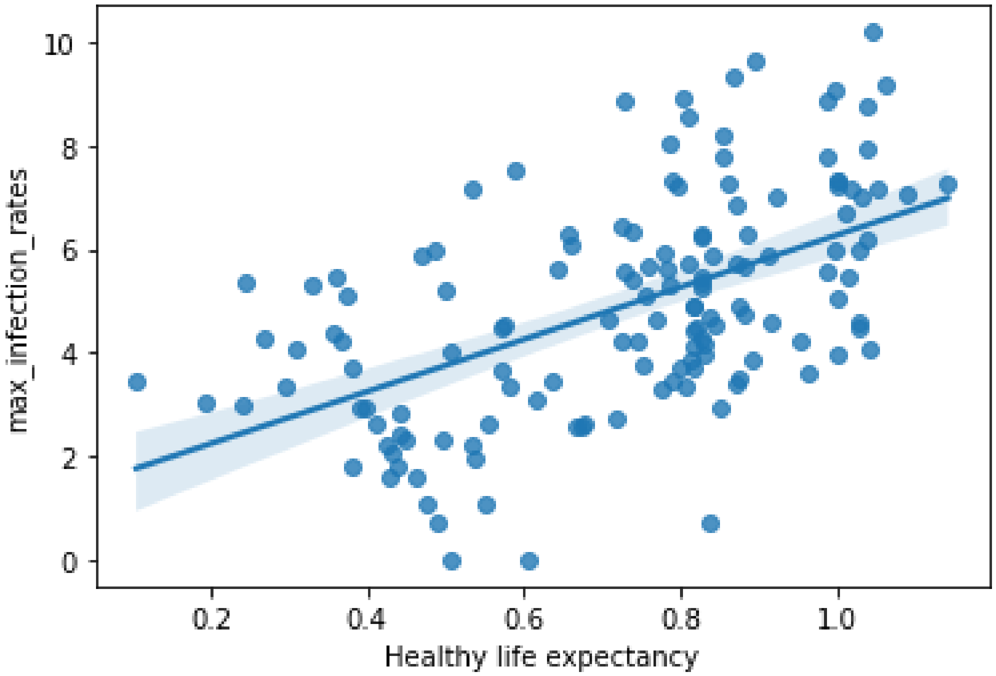
```
<AxesSubplot:xlabel='Healthy life expectancy',
ylabel='max_infection_rates'>
```



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

```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

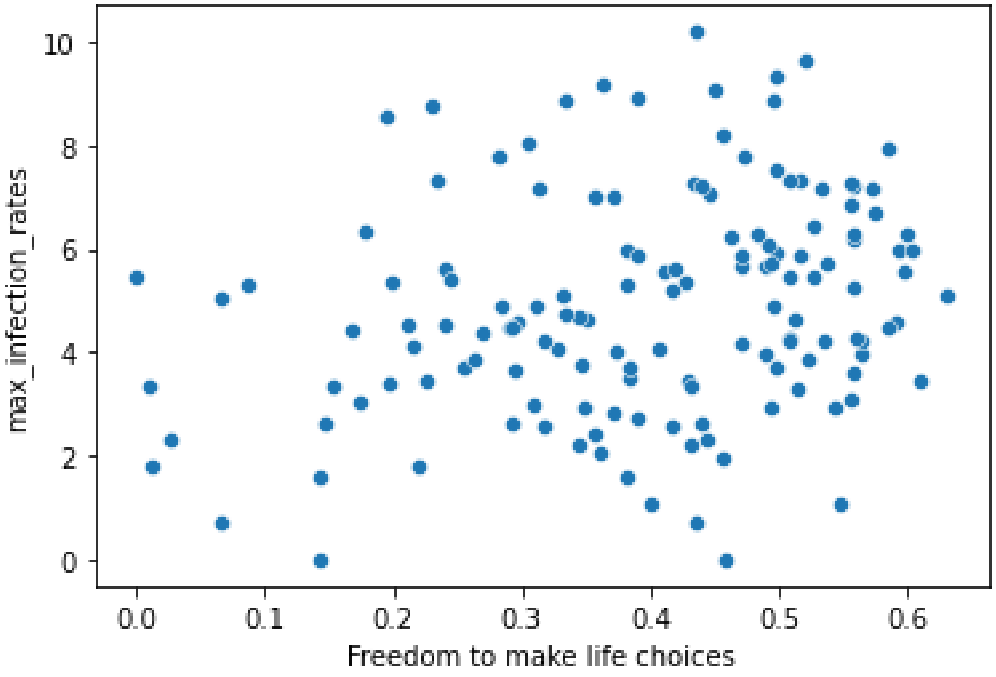
```
<AxesSubplot:xlabel='Healthy life expectancy',
ylabel='max_infection_rates'>
```



```
x=data["Freedom to make life choices"]
y=data["max_infection_rates"]
sns.scatterplot(x,np.log(y))
```

```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

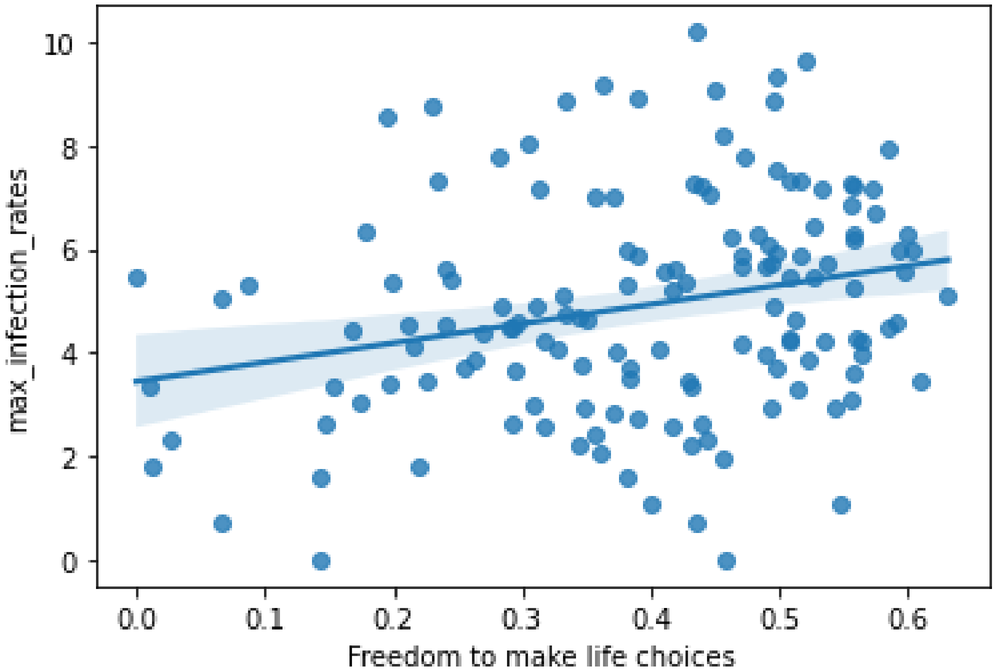
```
<AxesSubplot:xlabel='Freedom to make life choices',
ylabel='max_infection_rates'>
```



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

```
C:\Users\amitk\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Freedom to make life choices',
ylabel='max_infection_rates'>
```



10
8
6
4
2
0
max_infection_rates
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Freedom to make life choices