Welcome to Covid19 Data Analysis Notebook

Update : 01 July 2020

Let's Import the modules

In [127]:

```
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 [44]:

```
#corona_dataset_csv =pd.read_csv("Datasets/covid19_Confirmed_dataset.csv")
corona_dataset_csv =pd.read_csv("Datasets/time_series_covid19_confirmed_global.csv")
corona_dataset_csv.head()
```

Out[44]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/2
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	
1	NaN	Albania	41.1533	20.1683	0	0	0	0	
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	

5 rows × 165 columns

Let's check the shape of the dataframe

In [45]:

```
corona_dataset_csv.shape
```

Out[45]:

```
(266, 165)
```

Task 2.2: Delete the useless columns

In [46]:

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

In [47]:

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

Out[47]:

	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
0	NaN	Afghanistan	0	0	0	0	0	0	0
1	NaN	Albania	0	0	0	0	0	0	0
2	NaN	Algeria	0	0	0	0	0	0	0
3	NaN	Andorra	0	0	0	0	0	0	0
4	NaN	Angola	0	0	0	0	0	0	0
5	NaN	Antigua and Barbuda	0	0	0	0	0	0	0
6	NaN	Argentina	0	0	0	0	0	0	0
7	NaN	Armenia	0	0	0	0	0	0	0
8	Australian Capital Territory	Australia	0	0	0	0	0	0	0
9	New South Wales	Australia	0	0	0	0	3	4	4

10 rows × 163 columns

Task 2.3: Aggregating the rows by the country

In [48]:

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

In [49]:

```
corona_dataset_aggregated.head()
```

Out[49]:

	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
Country/Region									
Afghanistan	0	0	0	0	0	0	0	0	0
Albania	0	0	0	0	0	0	0	0	0
Algeria	0	0	0	0	0	0	0	0	0
Andorra	0	0	0	0	0	0	0	0	0
Angola	0	0	0	0	0	0	0	0	0

5 rows × 161 columns

In [50]:

```
corona_dataset_aggregated.shape
```

Out[50]:

```
(188, 161)
```

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

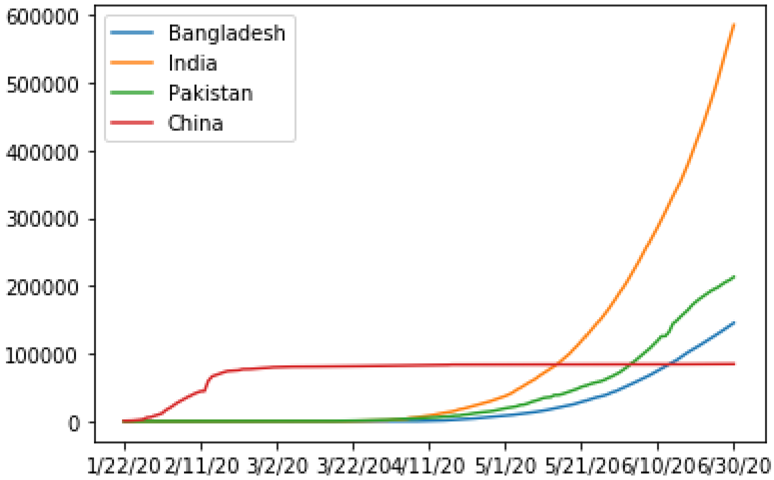
visualization always helps for better understanding of our data.

In [51]:

```
corona_dataset_aggregated.loc["Bangladesh"].plot()
corona_dataset_aggregated.loc["India"].plot()
corona_dataset_aggregated.loc["Pakistan"].plot()
corona_dataset_aggregated.loc["China"].plot()
plt.legend()
```

Out[51]:

```
<matplotlib.legend.Legend at 0x25bb4b00408>
```



Task3: Calculating a good measure

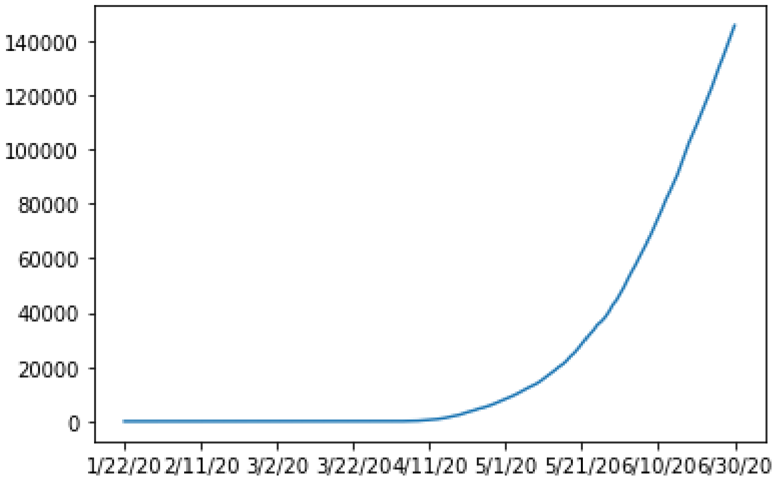
we need to find a good measure reperestend as a number, describing the spread of the virus in a country.

In [54]:

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

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb4c3c688>
```

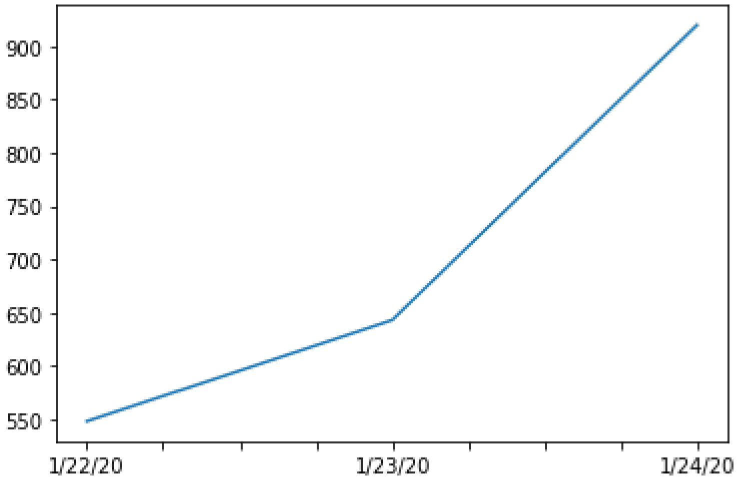


In [63]:

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

Out[63]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb4fcfa48>
```



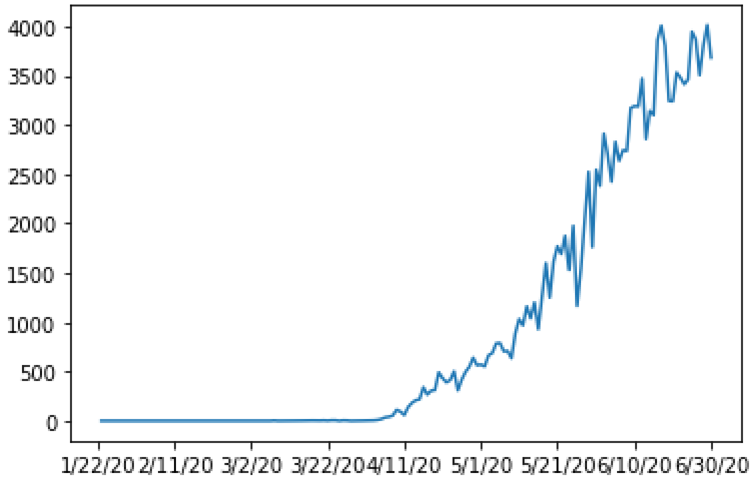
task 3.1: caculating the first derivative of the curve

In [64]:

```
corona_dataset_aggregated.loc['Bangladesh'].diff().plot()
```

Out[64]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb5044d88>
```



task 3.2: find maxmimum infection rate for China

In [67]:

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

Out[67]:

```
4014.0
```

In [66]:

```
corona_dataset_aggregated.loc['Pakistan'].diff().max()
```

Out[66]:

```
12073.0
```

In [68]:

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

Out[68]:

```
19906.0
```

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

In [71]:

```
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
```

In [85]:

```
corona_dataset_aggregated.head()
```

Out[85]:

	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
Country/Region									
Afghanistan	0	0	0	0	0	0	0	0	0
Albania	0	0	0	0	0	0	0	0	0
Algeria	0	0	0	0	0	0	0	0	0
Andorra	0	0	0	0	0	0	0	0	0
Angola	0	0	0	0	0	0	0	0	0

5 rows × 162 columns

Task 3.4: create a new dataframe with only needed column

In [86]:

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

In [87]:

```
corona_data.head()
```

Out[87]:

	max_infection_rate
Country/Region	
Afghanistan	915.0
Albania	82.0
Algeria	336.0
Andorra	79.0
Angola	47.0

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

In [116]:

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

In [117]:

```
happiness_report_csv.head()
```

Out[117]:

	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 [110]:

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

In [118]:

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

Out[118]:

	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 [119]:

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

Out[119]:

	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 [120]:

```
corona_data.head()
```

Out[120]:

	max_infection_rate
Country/Region	
Afghanistan	915.0
Albania	82.0
Algeria	336.0
Andorra	79.0
Angola	47.0

wolrd happiness report Dataset :

In [122]:

```
happiness_report_csv.head()
```

Out[122]:

	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 [123]:

```
data = happiness_report_csv.join(corona_data).copy()
data.head()
```

Out[123]:

	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	max_infection_rate
Country or region					
Finland	1.340	1.587	0.986	0.596	267.0
Denmark	1.383	1.573	0.996	0.592	391.0
Norway	1.488	1.582	1.028	0.603	386.0
Iceland	1.380	1.624	1.026	0.591	99.0
Netherlands	1.396	1.522	0.999	0.557	1346.0

Task 4.5: correlation matrix

In [124]:

```
data.corr()
```

Out[124]:

	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	max_infection_rate
GDP per capita	1.000000	0.754906	0.835462	0.379079	0.161995
Social support	0.754906	1.000000	0.719009	0.447333	0.124306
Healthy life expectancy	0.835462	0.719009	1.000000	0.390395	0.165086
Freedom to make life choices	0.379079	0.447333	0.390395	1.000000	0.048141
max_infection_rate	0.161995	0.124306	0.165086	0.048141	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 [125]:

```
data.head()
```

Out[125]:

	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	max_infection_rate
Country or region					
Finland	1.340	1.587	0.986	0.596	267.0
Denmark	1.383	1.573	0.996	0.592	391.0
Norway	1.488	1.582	1.028	0.603	386.0
Iceland	1.380	1.624	1.026	0.591	99.0
Netherlands	1.396	1.522	0.999	0.557	1346.0

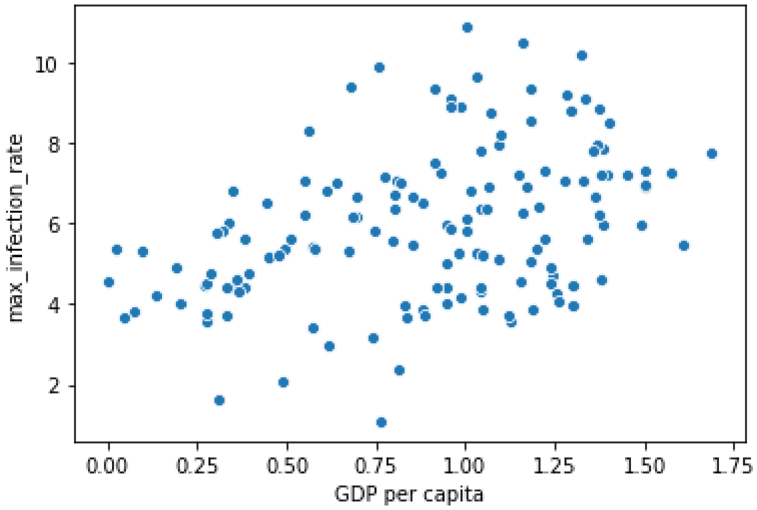
Task 5.1: Plotting GDP vs maximum Infection rate

In [133]:

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

Out[133]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb6244908>
```

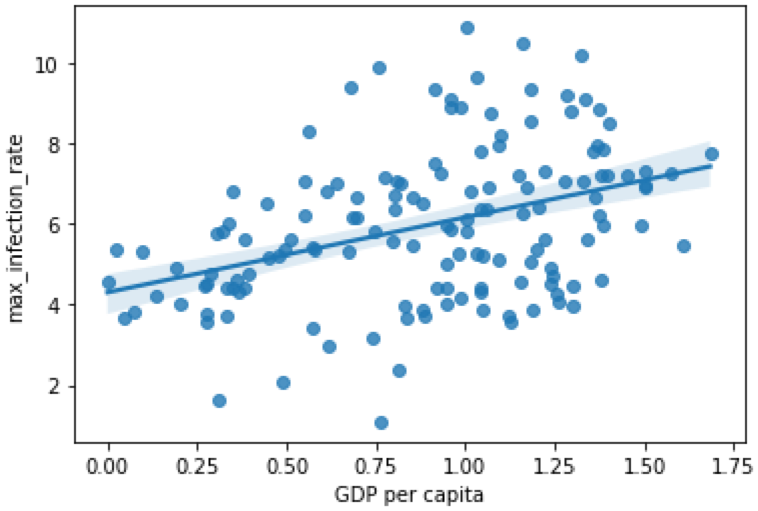


In [134]:

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

Out[134]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb6278308>
```



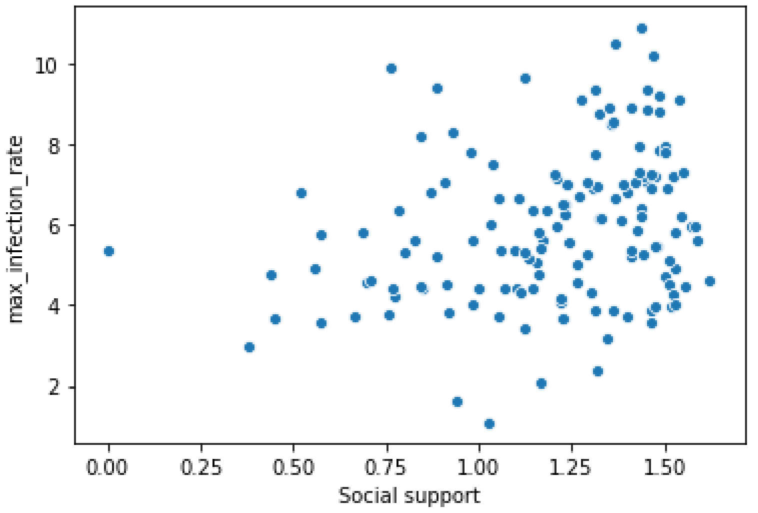
Task 5.2: Plotting Social support vs maximum Infection rate

In [135]:

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

Out[135]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb6292308>
```

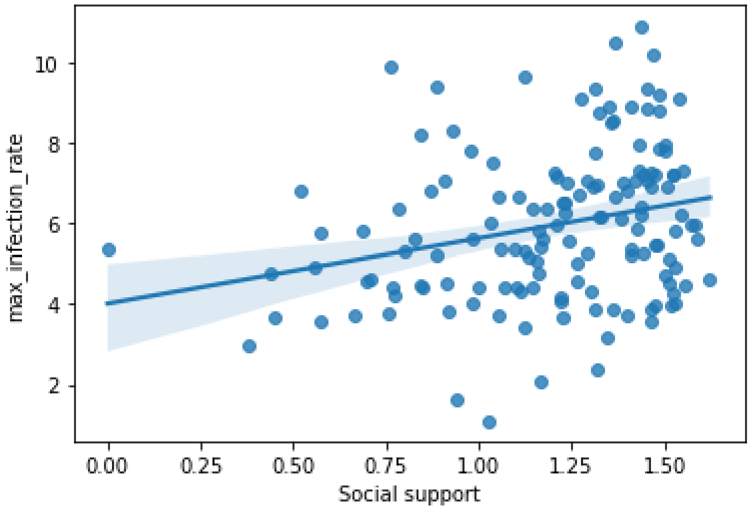


In [136]:

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

Out[136]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb6367648>
```



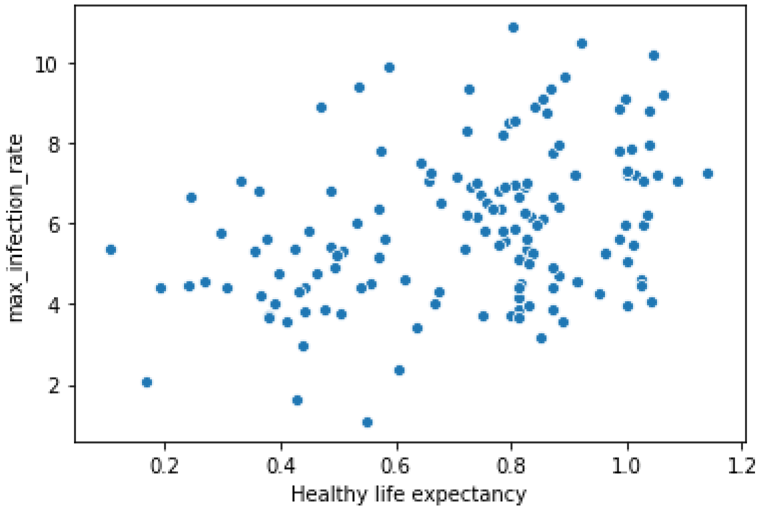
Task 5.3: Plotting Healthy life expectancy vs maximum Infection rate

In [137]:

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

Out[137]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb63d9748>
```

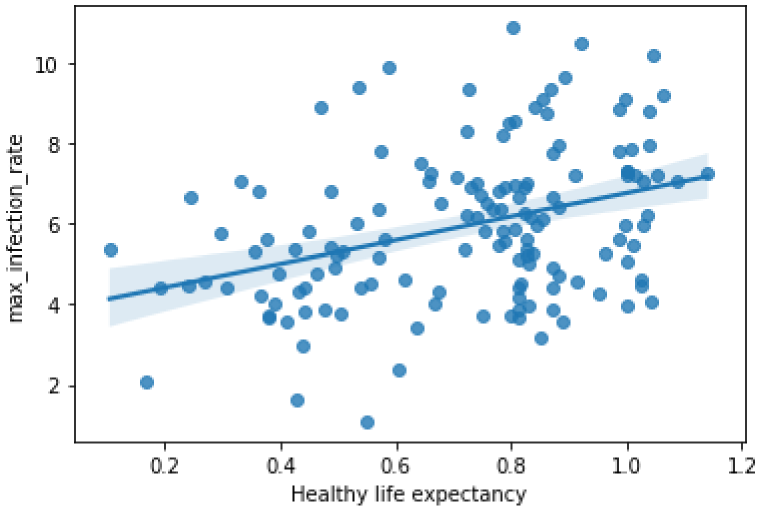


In [138]:

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

Out[138]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb63d4888>
```



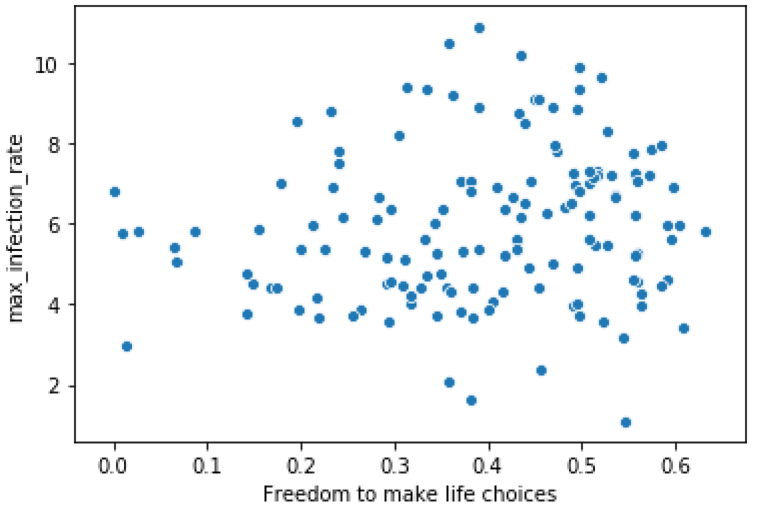
Task 5.4: Plotting Freedom to make life choices vs maximum Infection rate

In [139]:

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

Out[139]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb63d4108>
```

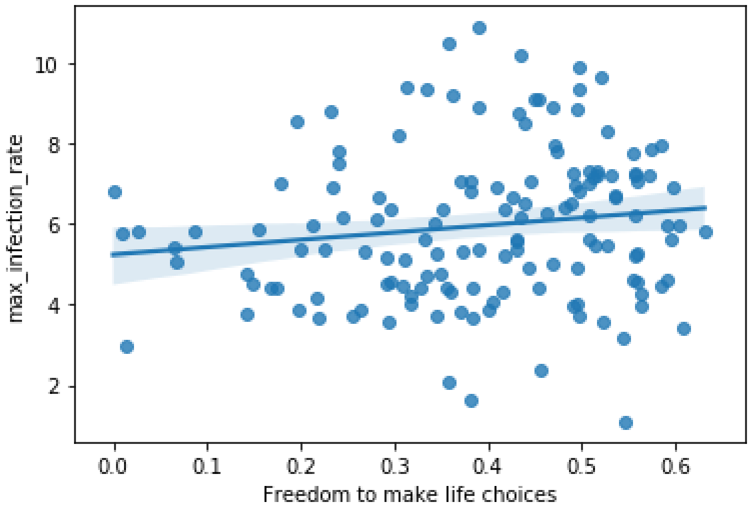


In [140]:

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

Out[140]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25bb64fcc88>
```


max_infection_rate
Freedom to make life choices