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

In [2]:
```
corona_dataset_csv = pd.read_csv(r"C:\Users\Maj Mortuza\Downloads\covid19_Confirmed_dataset.csv")
corona_dataset_csv.head(10)
```

											...				
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	609	634	663	678
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	2811	2910	3007	3127
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	717	723	723	731
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	24	25	25	25
5	NaN	Antigua and Barbuda	17.0608	-61.7964	0	0	0	0	0	0	...	23	24	24	24
6	NaN	Argentina	-38.4161	-63.6167	0	0	0	0	0	0	...	3031	3144	3435	3607
7	NaN	Armenia	40.0691	45.0382	0	0	0	0	0	0	...	1401	1473	1523	1596
8	Australian Capital Territory	Australia	-35.4735	149.0124	0	0	0	0	0	0	...	104	104	104	105
9	New South Wales	Australia	-33.8688	151.2093	0	0	0	0	3	4	...	2969	2971	2976	2982

10 rows × 104 columns

In [3]:
```
corona_dataset_csv.shape
```

Out[3]:
```
(266, 104)
```

In [4]:
```
# Delete unnecessary columns
corona_dataset_csv.drop(["Lat","Long"], axis=1, inplace=True)
```

In [5]:
```
corona_dataset_csv.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	4/22/20	4/23/20	4/24/20	4/25/20
0	NaN	Afghanistan	0	0	0	0	0	0	0	0	...	1092	1176	1279	1351	1463
1	NaN	Albania	0	0	0	0	0	0	0	0	...	609	634	663	678	712
2	NaN	Algeria	0	0	0	0	0	0	0	0	...	2811	2910	3007	3127	3256
3	NaN	Andorra	0	0	0	0	0	0	0	0	...	717	723	723	731	738
4	NaN	Angola	0	0	0	0	0	0	0	0	...	24	25	25	25	25

5 rows × 102 columns

In [6]:
```
# Aggregating the rows by the country
corona_dataset_aggregated = corona_dataset_csv.groupby('Country/Region').sum('numeric_only')
corona_dataset_aggregated.head(10)
```

Out[6]:

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	1/30/20	1/31/20	...	4/21/20	4/22/20	4/23/20	4/24/20	4/25/20
Afghanistan	0	0	0	0	0	0	0	0	0	0	...	1092	1176	1279	1351	1463
Albania	0	0	0	0	0	0	0	0	0	0	...	609	634	663	678	712
Algeria	0	0	0	0	0	0	0	0	0	0	...	2811	2910	3007	3127	3256
Andorra	0	0	0	0	0	0	0	0	0	0	...	717	723	723	731	738
Angola	0	0	0	0	0	0	0	0	0	0	...	24	25	25	25	25
Antigua and Barbuda	0	0	0	0	0	0	0	0	0	0	...	23	24	24	24	24
Argentina	0	0	0	0	0	0	0	0	0	0	...	3031	3144	3435	3607	3780
Armenia	0	0	0	0	0	0	0	0	0	0	...	1401	1473	1523	1596	1677
Australia	0	0	0	0	4	5	5	6	9	9	...	6645	6652	6662	6677	6694
Austria	0	0	0	0	0	0	0	0	0	0	...	14873	14925	15002	15071	15148

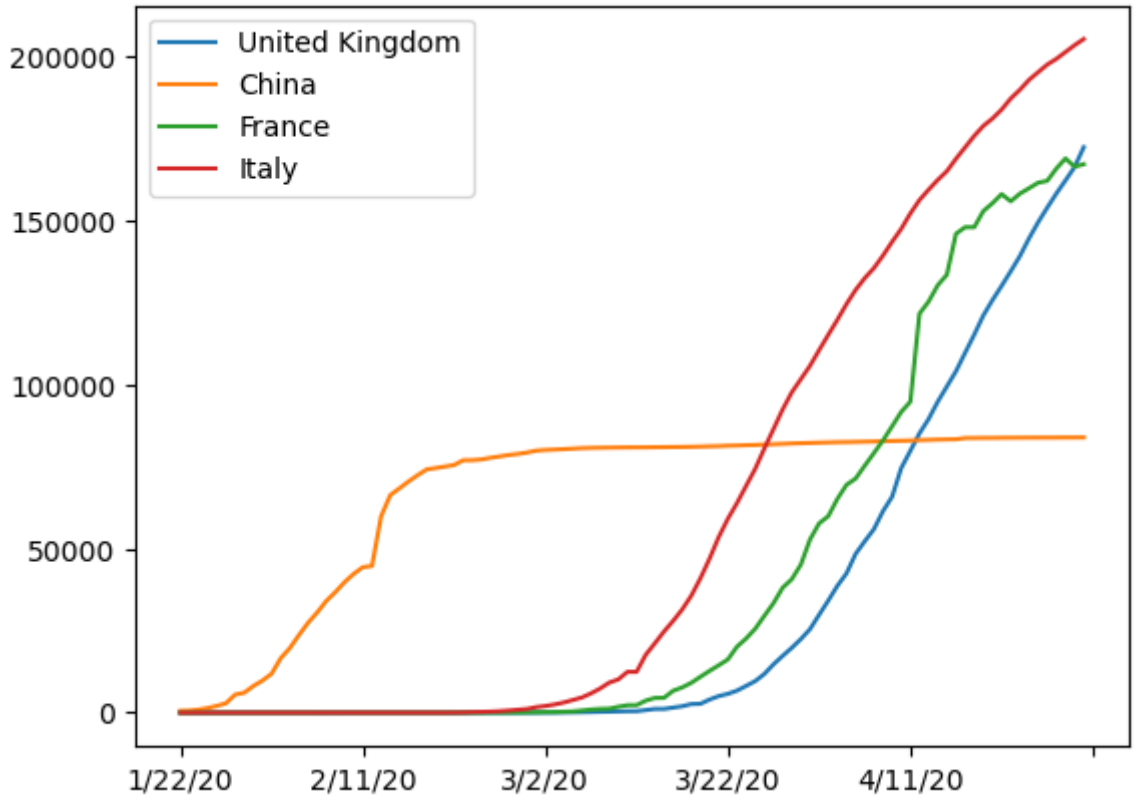
10 rows × 100 columns

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

Out[7]:
```
(187, 100)
```

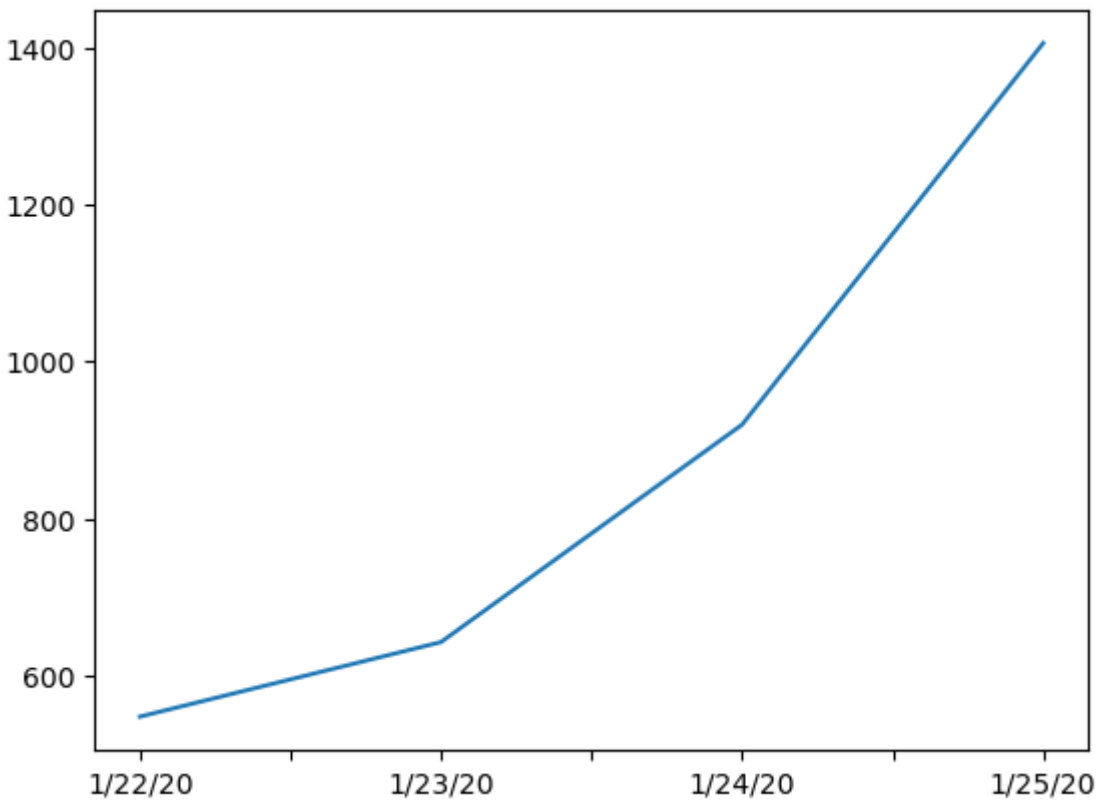
In [8]:
```
corona_dataset_aggregated.loc['United Kingdom'].plot()
corona_dataset_aggregated.loc['China'].plot()
corona_dataset_aggregated.loc['France'].plot()
corona_dataset_aggregated.loc['Italy'].plot()
plt.legend()
```

Out[8]:
```
<matplotlib.legend.Legend at 0x1e1987e3970>
```



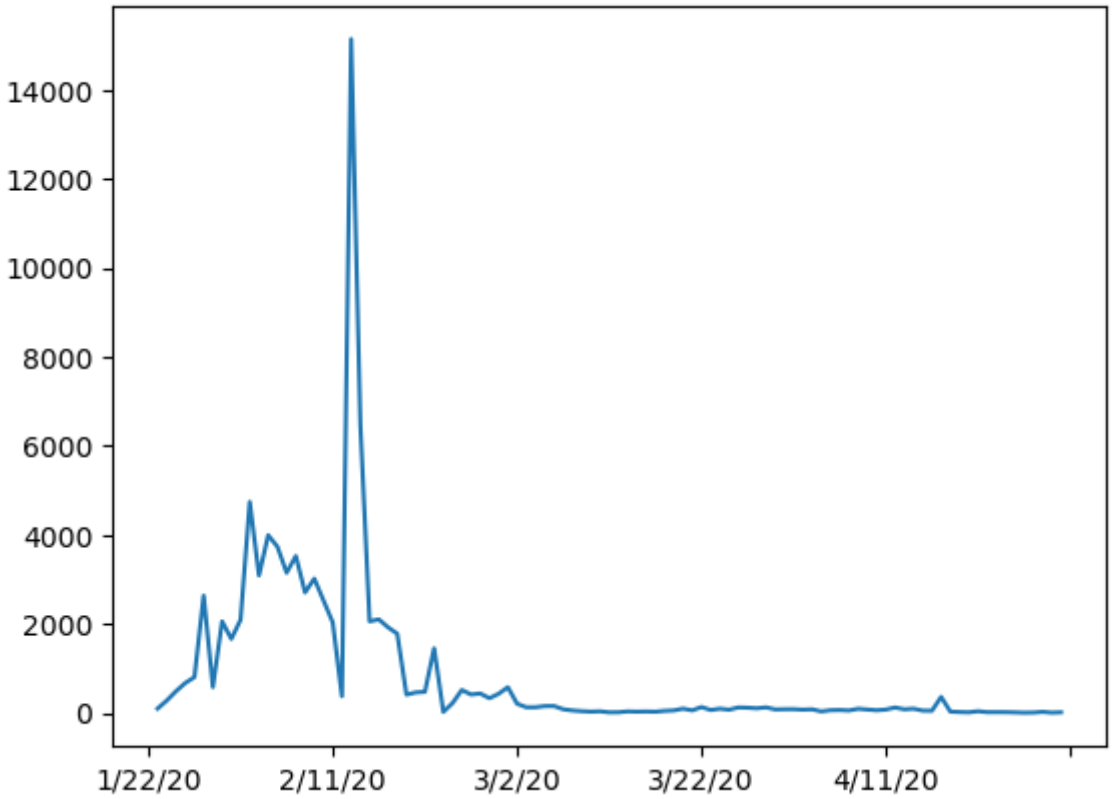
In [9]:
```
corona_dataset_aggregated.loc['China'][:4].plot()
```

Out[9]: <Axes: >



In [10]:
```
# Calculating and ploting first the derivative of the curve
corona_dataset_aggregated.loc['China'].diff().plot()
```

Out[10]: <Axes: >



In [11]:
```
corona_dataset_aggregated.loc['China'].diff().max()
```

Out[11]: 15136.0

In [12]:
```
corona_dataset_aggregated.loc['United Kingdom'].diff().max()
```

Out[12]: 8733.0

In [13]:
```
corona_dataset_aggregated.loc['Italy'].diff().max()
```

Out[13]: 6557.0

In [14]:
```
corona_dataset_aggregated.loc['France'].diff().max()
```

Out[14]: 26849.0

In [15]:
```
# Finding max infection rate for each country
countries = list(corona_dataset_aggregated.index)
max_infection_rate = []
for c in countries:
    max_infection_rate.append(corona_dataset_aggregated.loc[c].diff().max())
corona_dataset_aggregated['max_infection_rate'] = max_infection_rate
corona_dataset_aggregated.head()
```

Out[15]:

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	1/30/20	1/31/20	...	4/22/20	4/23/20	4/24/20	4/25/20	4/26/20
Afghanistan	0	0	0	0	0	0	0	0	0	0	...	1176	1279	1351	1463	1531
Albania	0	0	0	0	0	0	0	0	0	0	...	634	663	678	712	726
Algeria	0	0	0	0	0	0	0	0	0	0	...	2910	3007	3127	3256	3382
Andorra	0	0	0	0	0	0	0	0	0	0	...	723	723	731	738	738
Angola	0	0	0	0	0	0	0	0	0	0	...	25	25	25	25	26

5 rows × 101 columns

In [16]:
```
# Create a dataframe with only needed columns
corona_data = pd.DataFrame(corona_dataset_aggregated['max_infection_rate']).sort_values(by='max_infection_rate',
                                                                                        ascending = False)
corona_data.head()
```

Out[16]:

Country/Region	max_infection_rate
US	36188.0
France	26849.0
China	15136.0
Ecuador	11536.0
Spain	9630.0

In [17]:
```
happiness_report_csv = pd.read_csv(r"C:\Users\Maj Mortuza\Downloads\worldwide_happiness_report.csv")
happiness_report_csv.head()
```

Out[17]:

	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

In [18]:
```
# Remove unwanted columns
useless_cols = ['Overall rank', 'Score', 'Generosity', 'Perceptions of corruption']
happiness_report_csv.drop(useless_cols, axis = 1, inplace = True)
happiness_report_csv.head()
```

Out[18]:

	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

In [19]:
```
# Putting the name of the countries as the indices of the matrix
happiness_report_csv.set_index('Country or region', inplace = True)
happiness_report_csv.head()
```

Out[19]:

Country or region	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
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 [20]:
```
happiness_report_csv.shape
```

Out[20]: (156, 4)

In [21]:
```
corona_data.shape
```

Out[21]: (187, 1)

In [22]:
```
# As the row numbers are not same for both sets, we will need the common countries. SO we will use inner join
data = corona_data.join(happiness_report_csv, how = 'inner')
data.head()
```

Out[22]:

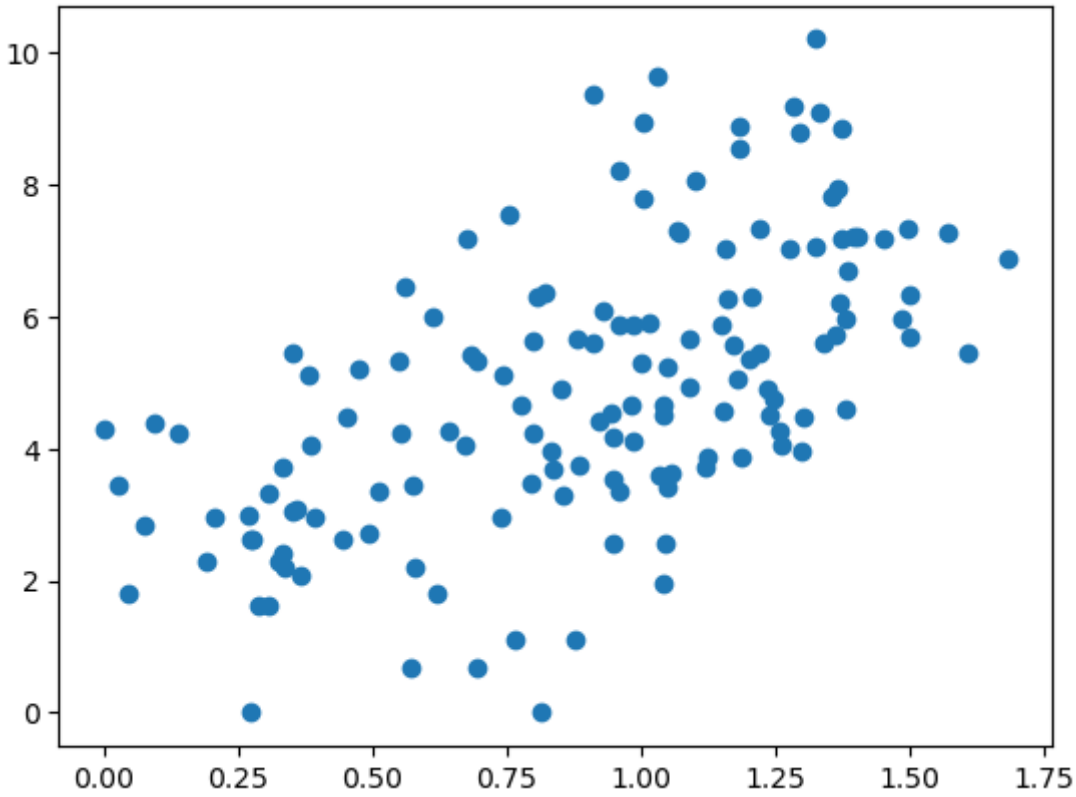
	max_infection_rate	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
France	26849.0	1.324	1.472	1.045	0.436
China	15136.0	1.029	1.125	0.893	0.521
Ecuador	11536.0	0.912	1.312	0.868	0.498
Spain	9630.0	1.286	1.484	1.062	0.362
United Kingdom	8733.0	1.333	1.538	0.996	0.450

In [23]:
```
data.shape
```

Out[23]: (143, 5)

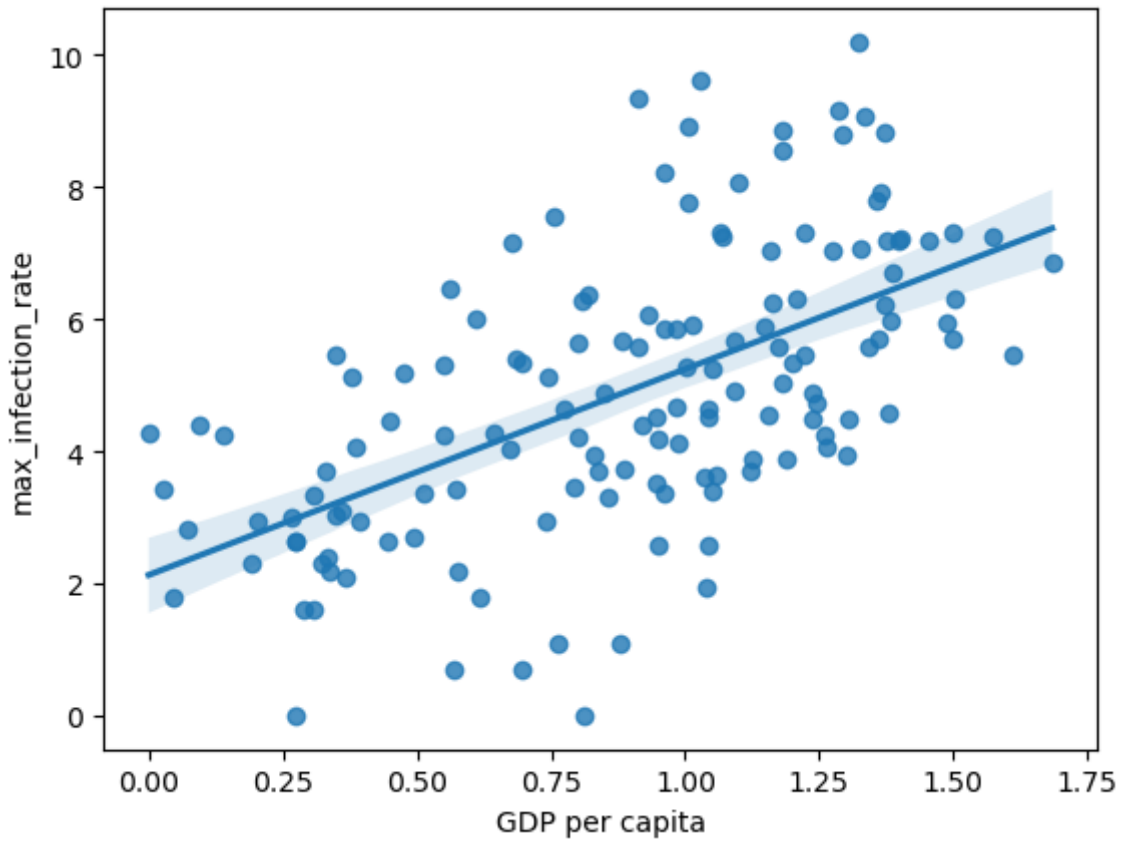
In [24]:
```
x = data['GDP per capita']
y = data['max_infection_rate']
plt.scatter(x, np.log(y))
```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x1e198b968f0>
```



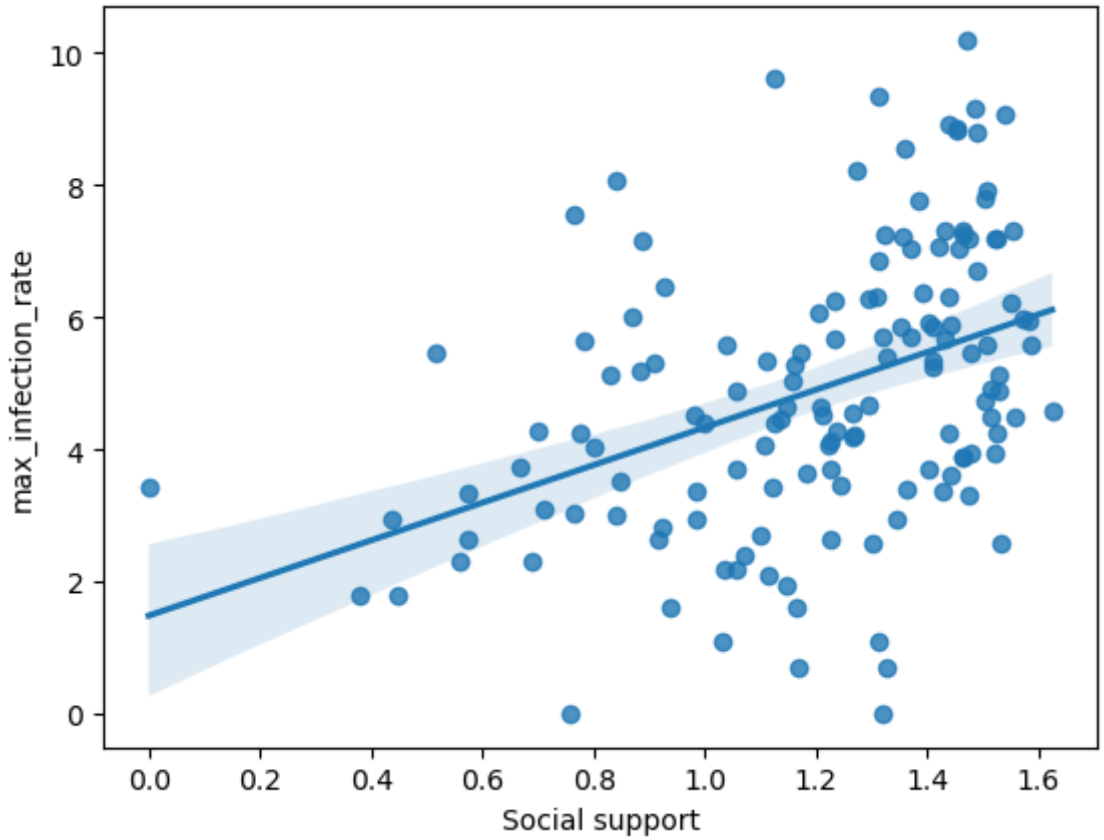
In [31]:
```
sns.regplot(x=x, y=np.log(y))
```

Out[31]:
```
<Axes: xlabel='GDP per capita', ylabel='max_infection_rate'>
```



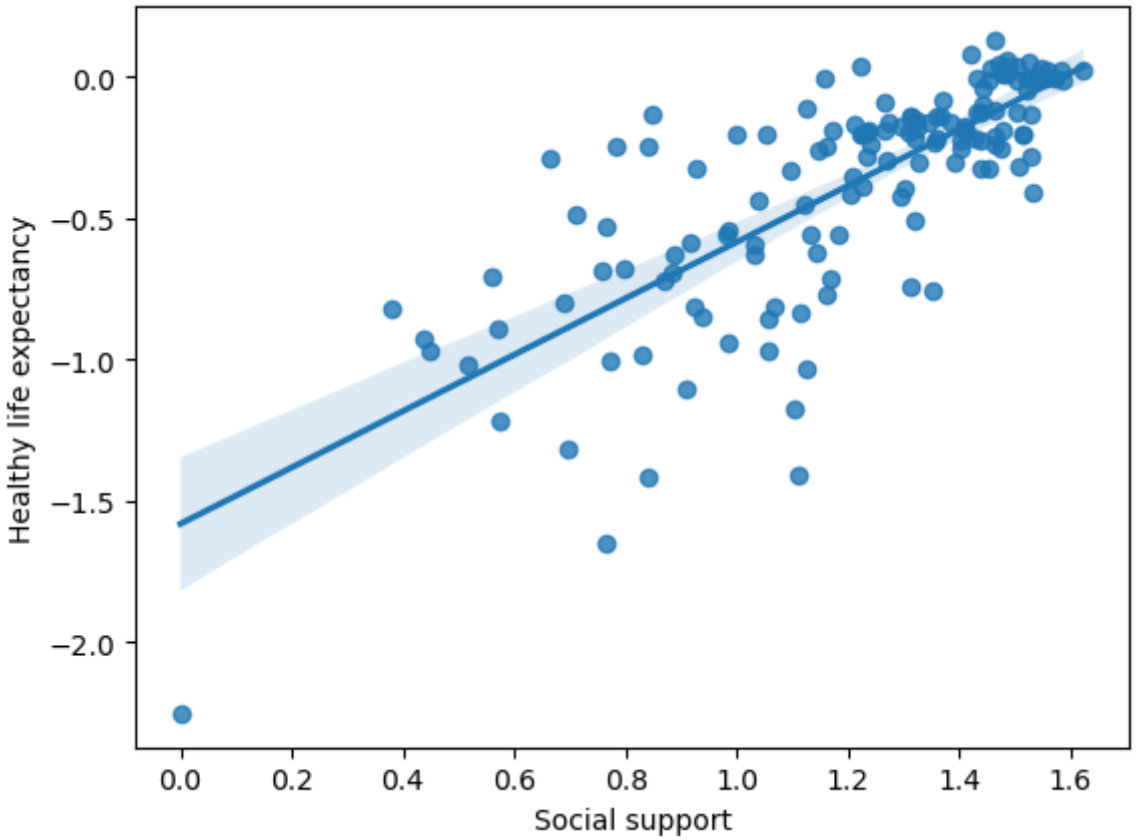
In [33]:
```
x = data['Social support']
y = data['max_infection_rate']
sns.regplot(x=x, y=np.log(y))
```

Out[33]: <Axes: xlabel='Social support', ylabel='max_infection_rate'>



In [34]:
```
x = data['Social support']
y = data['Healthy life expectancy']
sns.regplot(x=x, y=np.log(y))
```

Out[34]: <Axes: xlabel='Social support', ylabel='Healthy life expectancy'>



In [ ]:
```
# This results show that people who are living in develop countries were more prone to getting the infection
#compared to less developed country.
```