Covid19_Project

May 16, 2024

[ ]:
```
# TASK 1: IMPORT THE MODULES
```

[79]:
```
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
```

[80]:
```
df=pd.read_csv(r'C:\Users\user\Downloads\Covid19_Confirmed_dataset.csv')
```

[4]:
```
df.head()
```

[4]:

	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]
```

[ ]:
```
# The shape of the dataframe
```

[5]:
```
df.shape
```

[5]:
```
(266, 104)
```

[ ]:
```
# Task 2.2: Delete the useless columns
```

[10]:
```
df.drop(['Lat','Long'],axis=1, inplace=True)

df.head(10)
```

[10]:

	Province/State	Country/Region	1/22/20	1/23/20	\
0	NaN	Afghanistan	0	0	
1	NaN	Albania	0	0	
2	NaN	Algeria	0	0	
3	NaN	Andorra	0	0	
4	NaN	Angola	0	0	
5	NaN	Antigua and Barbuda	0	0	
6	NaN	Argentina	0	0	
7	NaN	Armenia	0	0	
8	Australian Capital Territory	Australia	0	0	
9	New South Wales	Australia	0	0	

	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	…	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	3	4	4	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 102 columns]
```

[ ]:
```
# Task 2.3: Aggregating the rows by the country
```

[13]:
```
df_aggregated = df.groupby('Country/Region').sum()

df_aggregated.head()
```

[13]:
```
                Province/State  1/22/20  1/23/20  1/24/20  1/25/20  1/26/20  \
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

                1/27/20  1/28/20  1/29/20  1/30/20  …  4/21/20  4/22/20  \
Country/Region                                      …
Afghanistan           0        0        0        0  …     1092     1176
Albania               0        0        0        0  …      609      634
Algeria               0        0        0        0  …     2811     2910
Andorra               0        0        0        0  …      717      723
Angola                0        0        0        0  …       24       25

                4/23/20  4/24/20  4/25/20  4/26/20  4/27/20  4/28/20  4/29/20  \
Country/Region
Afghanistan        1279     1351     1463     1531     1703     1828     1939
Albania             663      678      712      726      736      750      766
Algeria            3007     3127     3256     3382     3517     3649     3848
Andorra             723      731      738      738      743      743      743
Angola               25       25       25       26       27       27       27
```

```
                4/30/20
Country/Region
Afghanistan        2171
Albania             773
Algeria            4006
Andorra             745
Angola               27

[5 rows x 101 columns]
```

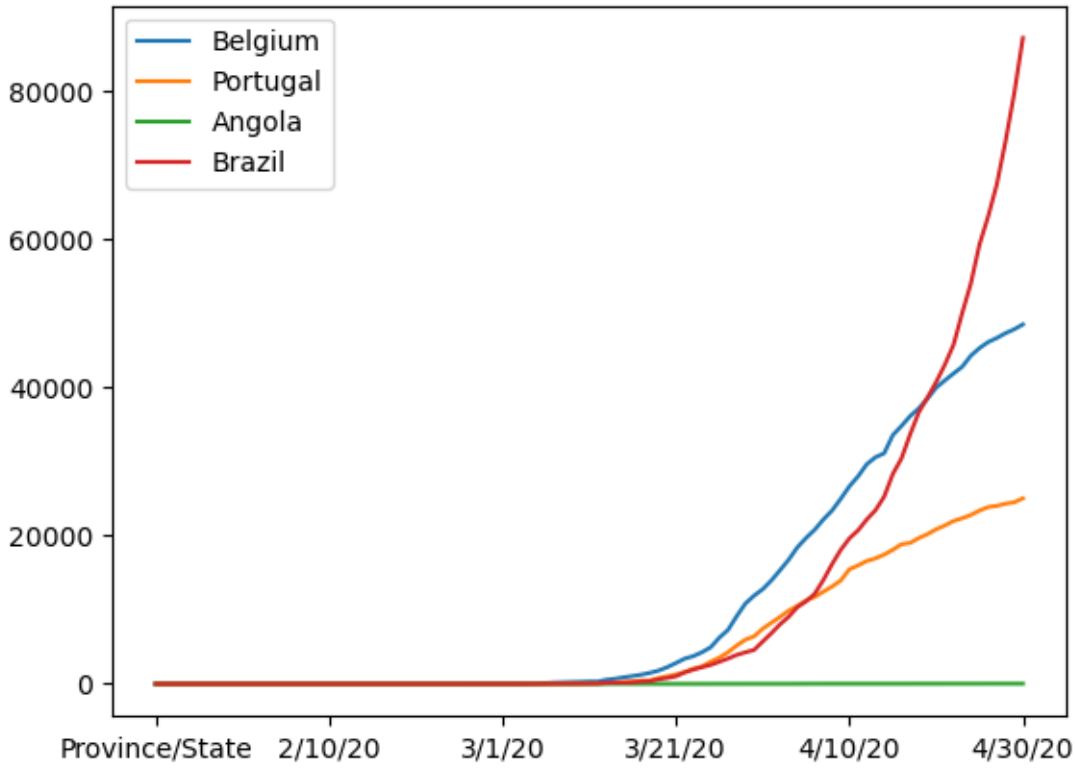
[14]:
```
df_aggregated.shape
```

[14]:
```
(187, 101)
```

[ ]:
```
# Task 2.4: Visualizing data related to a country => Belgium
```

[44]:
```
df_aggregated.loc["Belgium"].plot()
df_aggregated.loc["Portugal"].plot()
df_aggregated.loc["Angola"].plot()
df_aggregated.loc["Brazil"].plot()
plt.legend()
```

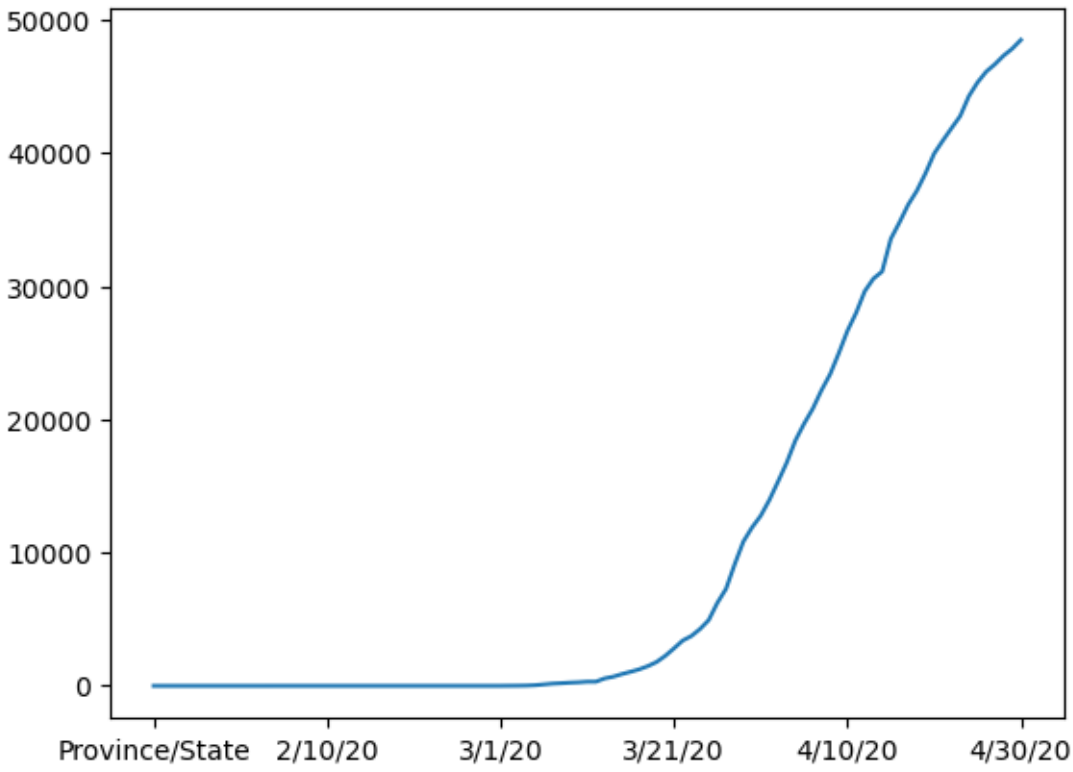
[44]:
```
<matplotlib.legend.Legend at 0x18528c59550>
```



[ ]:
```
# TASK 3 : CALCULATING A GOOD MEASURE
```

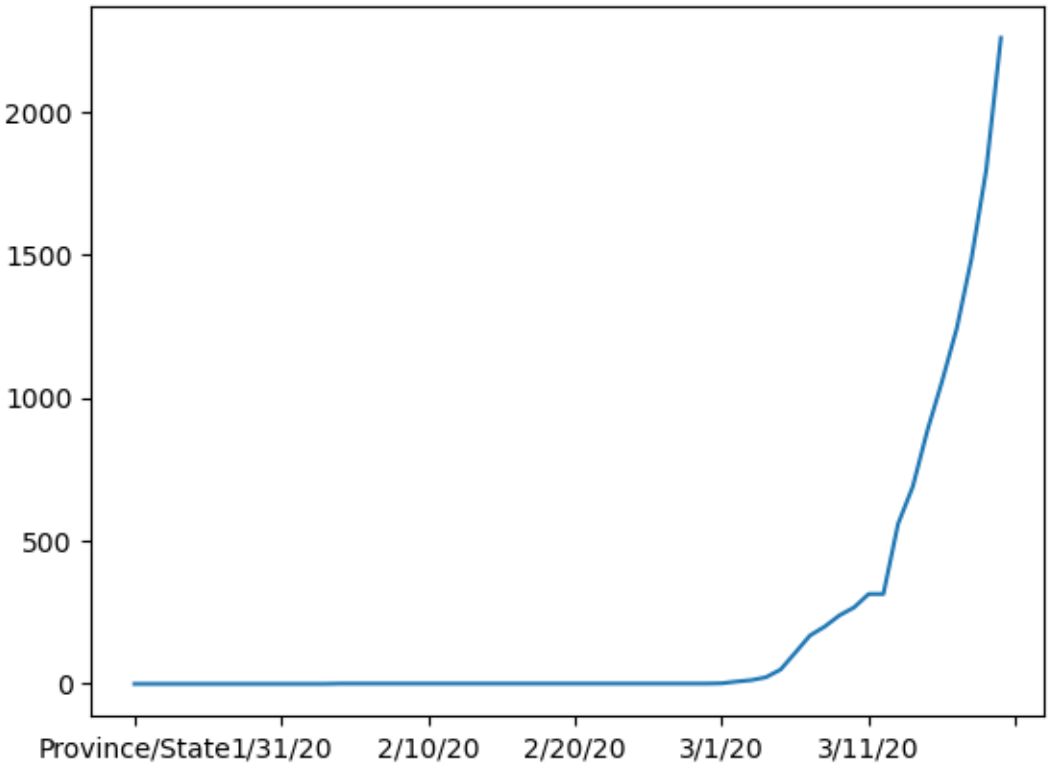
[40]:
```
df_aggregated.loc['Belgium'].plot()
```

[40]: <Axes: >



[41]:
```
df_aggregated.loc["Belgium"][:60].plot()
```

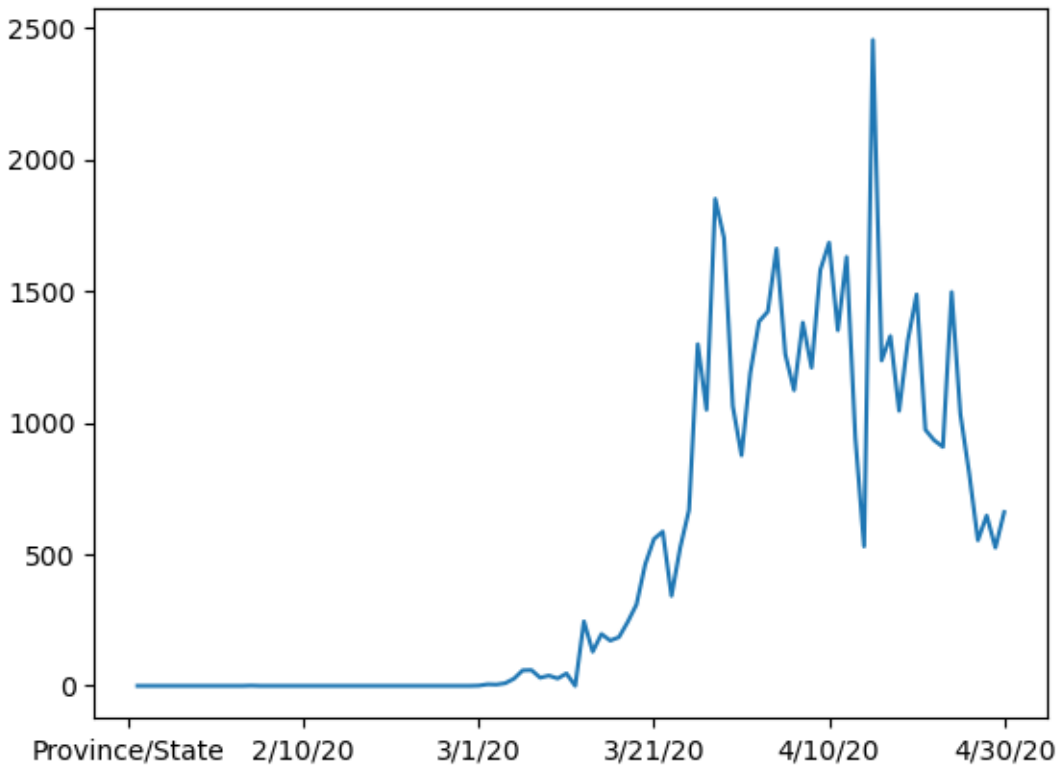
[41]: <Axes: >



[ ]:
```
# Task 3.1: Calculating and plotting the first derivative of the curve
```

[43]:
```
df_aggregated.loc["Belgium"].diff().plot()
```

[43]: <Axes: >



[ ]:
```
# Task 3.2: Find maximum infection rate for Belgium, Portugal, Angola and Brazil
```

[45]:
```
df_aggregated.loc["Belgium"].diff().max()
```

[45]: 2454

[46]:
```
df_aggregated.loc["Portugal"].diff().max()
```

[46]: 1516

[47]:
```
df_aggregated.loc["Angola"].diff().max()
```

[47]: 5

[48]:
```
df_aggregated.loc["Brazil"].diff().max()
```

[48]: 7502

[ ]:
```
# Task 3.3: Find maximum infection rate for all the coutries
```

[86]:
```
df_aggregated = df_aggregated.apply(pd.to_numeric, errors='coerce')
def calculate_max_infection_rate(row):
    return row.diff().max()
```

```
max_infection_rates = df_aggregated.apply(calculate_max_infection_rate, axis=1)
df_aggregated['max_infection_rates'] = max_infection_rates

df_aggregated.head()
```

[86]:

Country/Region	Province/State	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	\
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	

Country/Region	1/27/20	1/28/20	1/29/20	1/30/20	…	4/22/20	4/23/20	\
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	

Country/Region	4/24/20	4/25/20	4/26/20	4/27/20	4/28/20	4/29/20	4/30/20	\
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	

Country/Region	max_infection_rates
Afghanistan	232.0
Albania	34.0
Algeria	199.0
Andorra	43.0
Angola	5.0

[5 rows x 102 columns]

[ ]:
```
# Task 3.4: Create a new dataframe with only needed column
```

[89]:
```
corona_data = pd.DataFrame(df_aggregated['max_infection_rates'])
```

[90]:
```
corona_data.head()
```

[90]:

Country/Region	max_infection_rates

```
Afghanistan                 232.0
Albania                      34.0
Algeria                     199.0
Andorra                      43.0
Angola                        5.0
```

[ ]:
```
# TASK 4

# TASK 4.1: Importing dataset
```

[102]:
```
happiness_report_csv=pd.read_csv(r'C:
 ↪\Users\user\Downloads\worldwide_happiness_report.csv')
```

[103]:
```
happiness_report_csv.head()
```

[103]:

	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

[ ]:
```
# Task 4.2: Drop the useless columns
```

[125]:
```
print(happiness_report_csv.columns)
```

```
Index(['Country or region', 'GDP per capita', 'Social support',
       'Healthy life expectancy', 'Freedom to make life choices'],
      dtype='object')
```

[124]:
```
happiness_report_csv.reset_index(drop=True, inplace=True)
```

[126]:
```
happiness_report_csv = pd.read_csv(r'C:
 ↪\Users\user\Downloads\worldwide_happiness_report.csv', header=0)
```

[127]:
```
print(happiness_report_csv.columns)
```

```
Index(['Overall rank', 'Country or region', 'Score', 'GDP per capita',
       'Social support', 'Healthy life expectancy',
       'Freedom to make life choices', 'Generosity',
       'Perceptions of corruption'],
      dtype='object')
```

[128]:
```
useless_cols = ['Overall rank', 'Score', 'Generosity','Perceptions of␣
↪corruption']
```

[129]:
```
happiness_report_csv.drop(useless_cols, axis=1, inplace=True)
happiness_report_csv.head()
```

[129]:

	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
```

[130]:
```
happiness_report_csv.set_index("Country or region", inplace=True)

happiness_report_csv.head()
```

[130]:

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	

```
Netherlands                                0.557
```

[131]:
```
# Task 4.4: Join two dataset

# Corona Dataset

corona_data.head()
```

[131]:
```
                max_infection_rates
Country/Region
Afghanistan                   232.0
Albania                        34.0
Algeria                       199.0
Andorra                        43.0
Angola                          5.0
```

[132]:
```
corona_data.shape
```

[132]:
```
(187, 1)
```

[134]:
```
# World happiness report Dataset

happiness_report_csv.head()
```

[134]:
```
                    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
```

[135]:
```
happiness_report_csv.shape
```

[135]:
```
(156, 4)
```

[137]:
```
data = corona_data.join(happiness_report_csv, how="inner")
data.head()
```

[137]:
```
             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

             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
```

[138]:
```
# Task 4.5: Correlation matrix

data.corr()
```

[138]:
```
                              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
```

[ ]:
```
# TASK 5: VISUALIZATION OF THE RESULTS
```

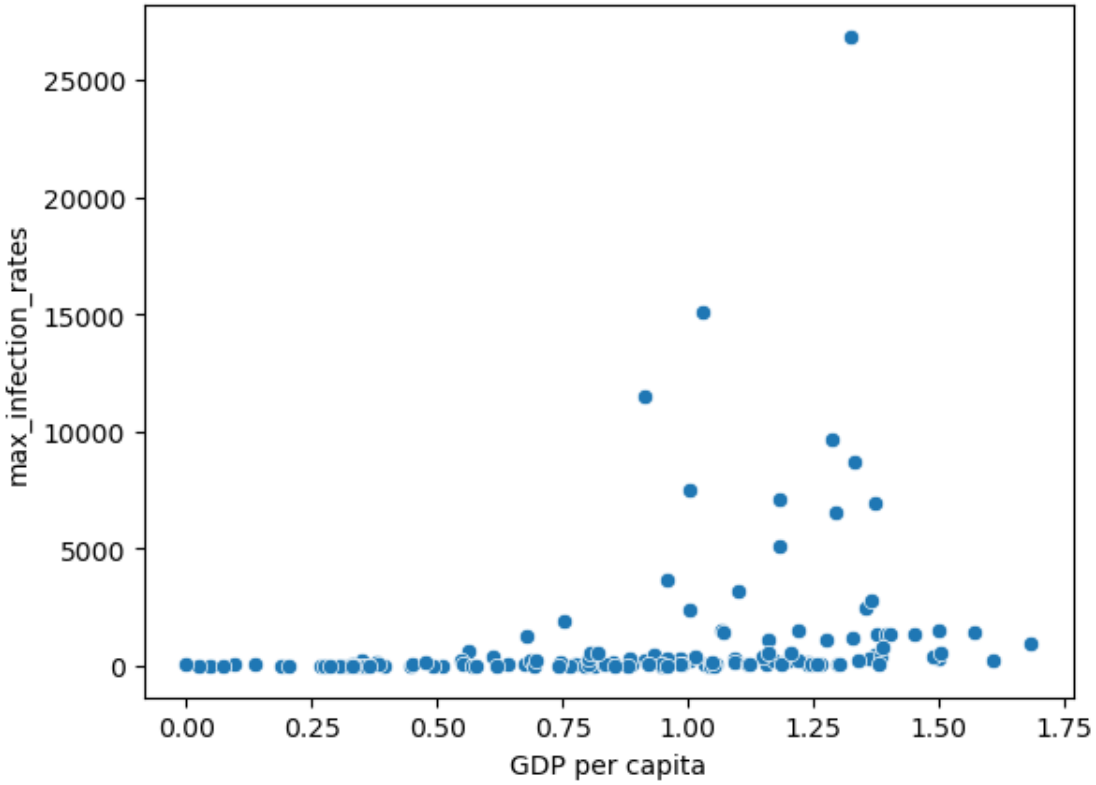
[139]:
```
data.head()
```

[139]:
```
             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

            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
```

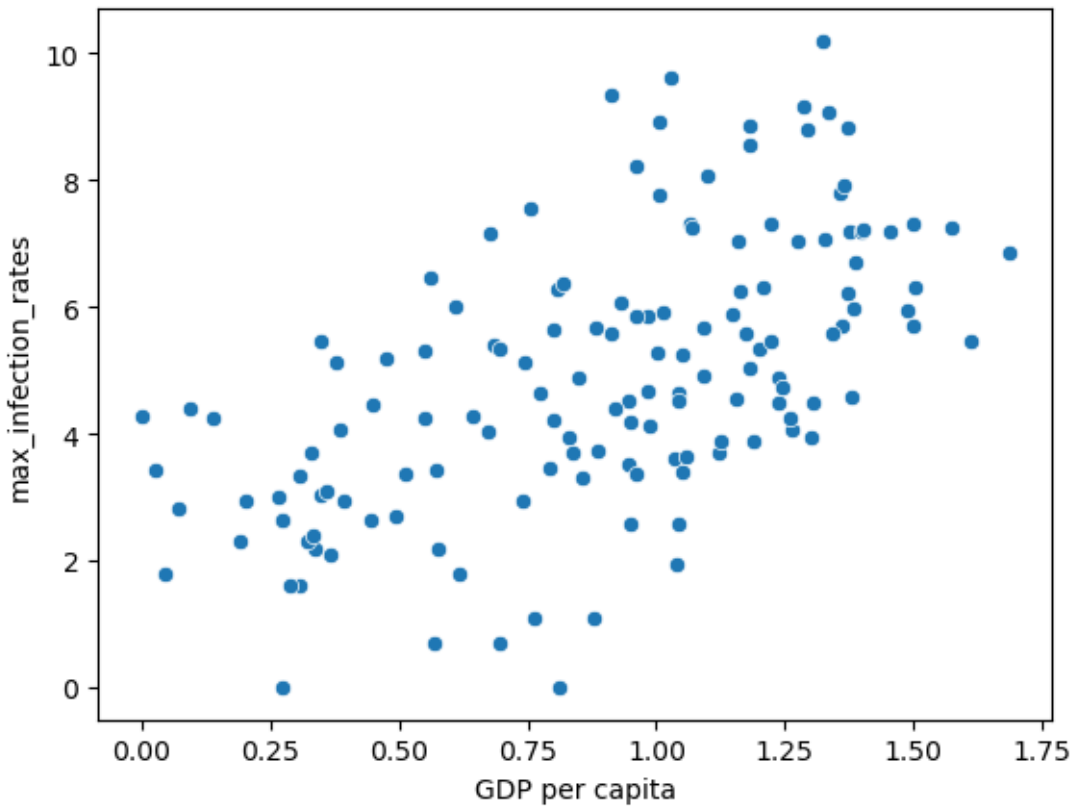
[142]:
```
# Task 5.1: Plotting GDP vs maximum infection rate

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

[142]: <Axes: xlabel='GDP per capita', ylabel='max_infection_rates'>



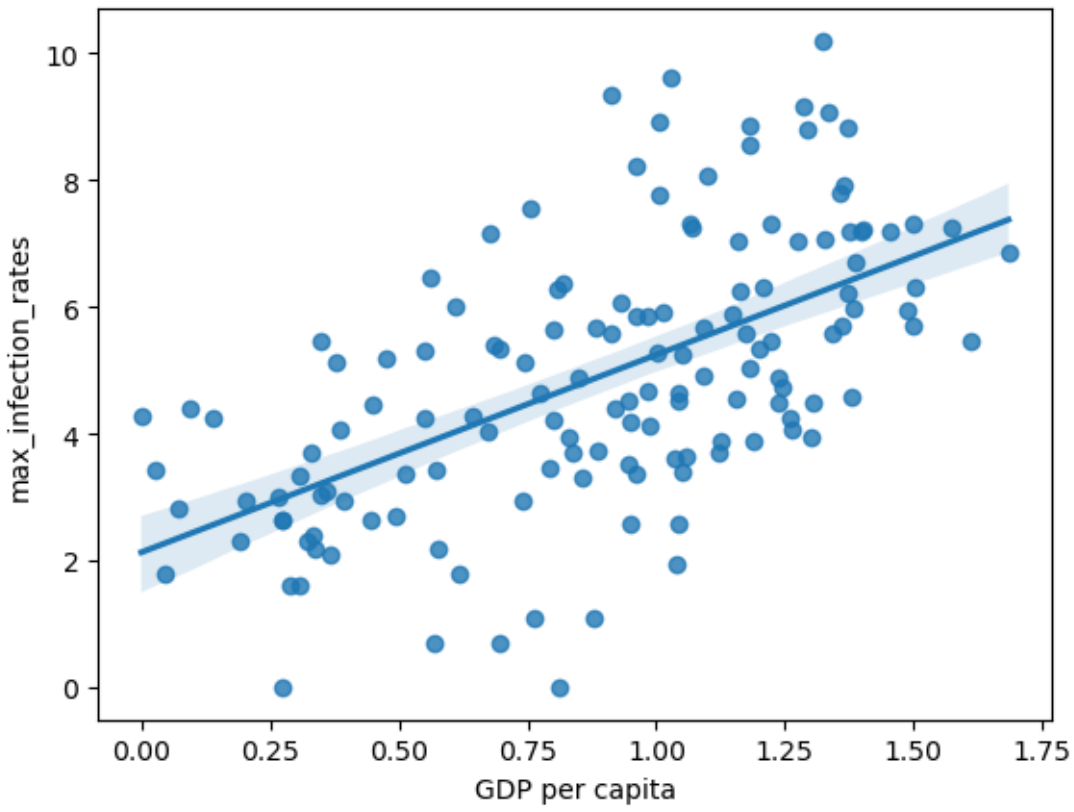
[143]:
```
y_log = np.log(data["max_infection_rates"])

sns.scatterplot(data=data, x="GDP per capita", y=y_log)
```

[143]: <Axes: xlabel='GDP per capita', ylabel='max_infection_rates'>



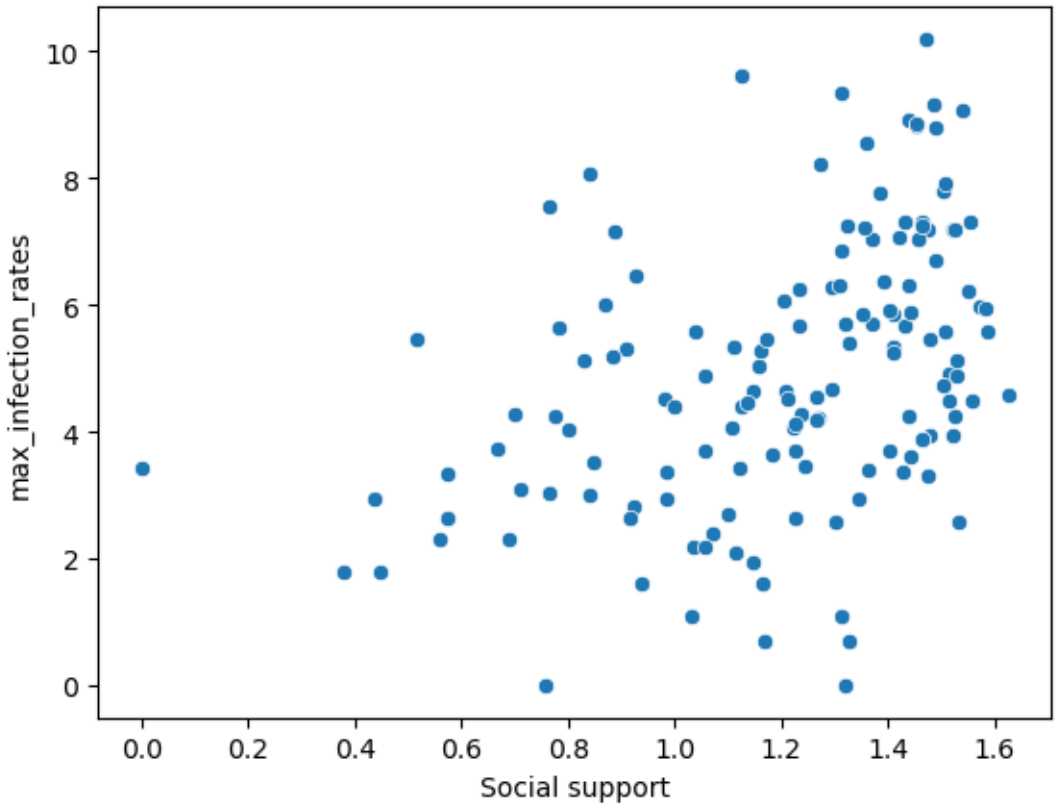
[145]:
```
sns.regplot(data=data, x="GDP per capita", y=np.
↪log(data["max_infection_rates"]))
```

[145]: <Axes: xlabel='GDP per capita', ylabel='max_infection_rates'>



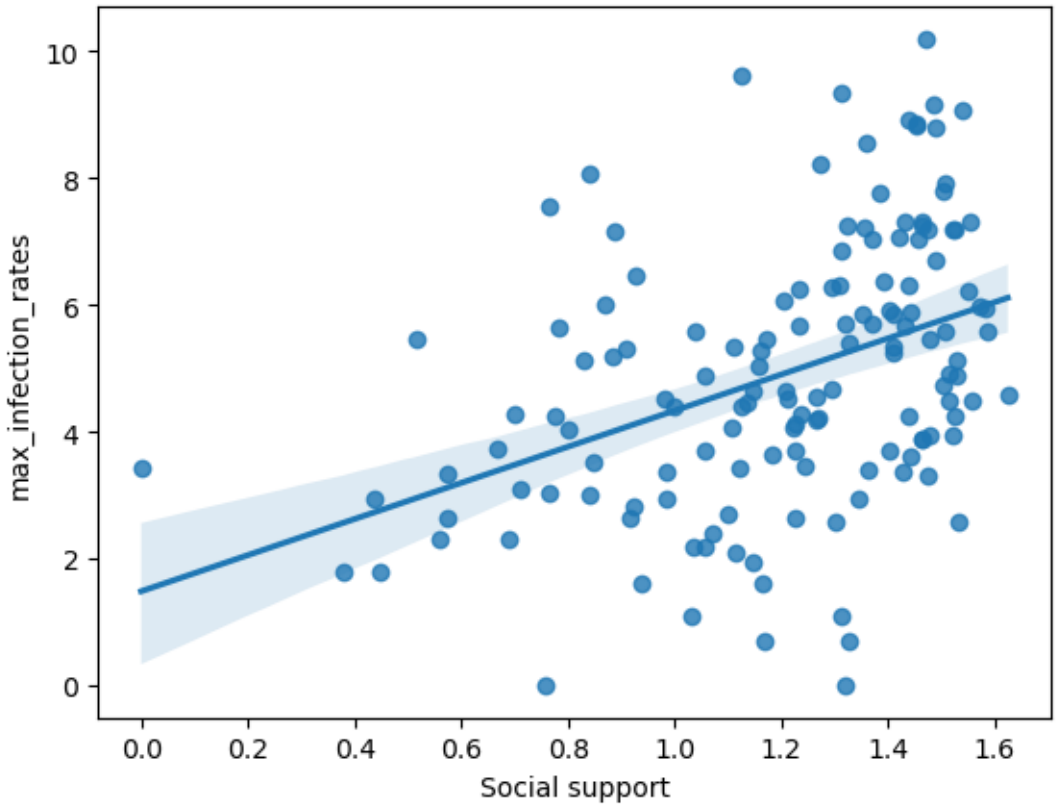
[146]:
```
# Task 5.2: Plotting Social support vs maximum infection rate


y_log = np.log(data["max_infection_rates"])

sns.scatterplot(data=data, x="Social support", y=y_log)
```

[146]: <Axes: xlabel='Social support', ylabel='max_infection_rates'>



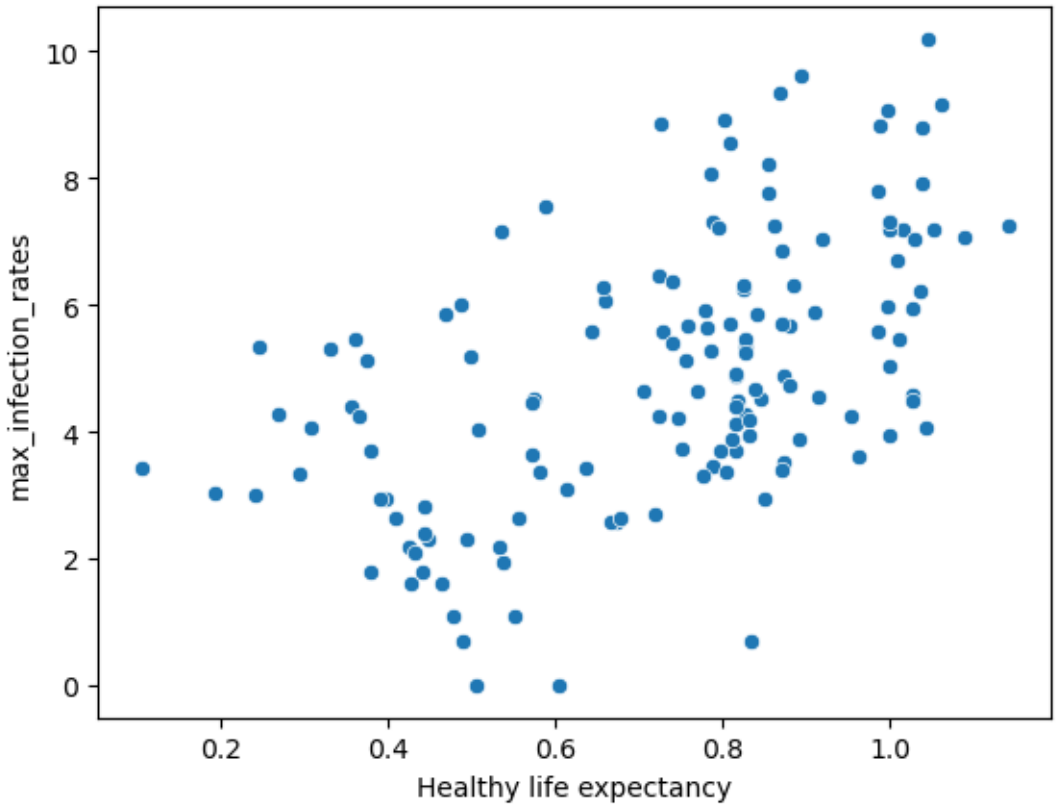
[148]:
```
sns.regplot(data=data, x="Social support", y=np.
↪log(data["max_infection_rates"]))
```

[148]: <Axes: xlabel='Social support', ylabel='max_infection_rates'>



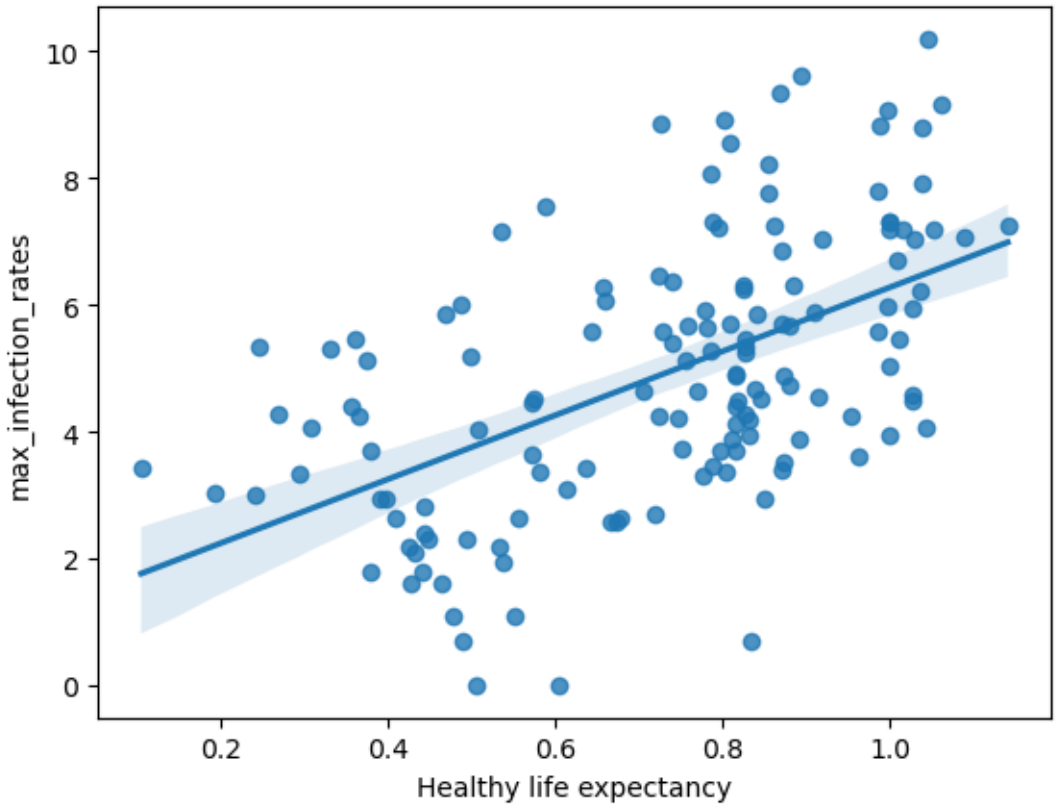
[150]:
```
# Task 5.3: Plotting Healty life expectancy vs maximum infection rate


y_log = np.log(data["max_infection_rates"])

sns.scatterplot(data=data, x="Healthy life expectancy", y=y_log)
```

[150]:
```
<Axes: xlabel='Healthy life expectancy', ylabel='max_infection_rates'>
```



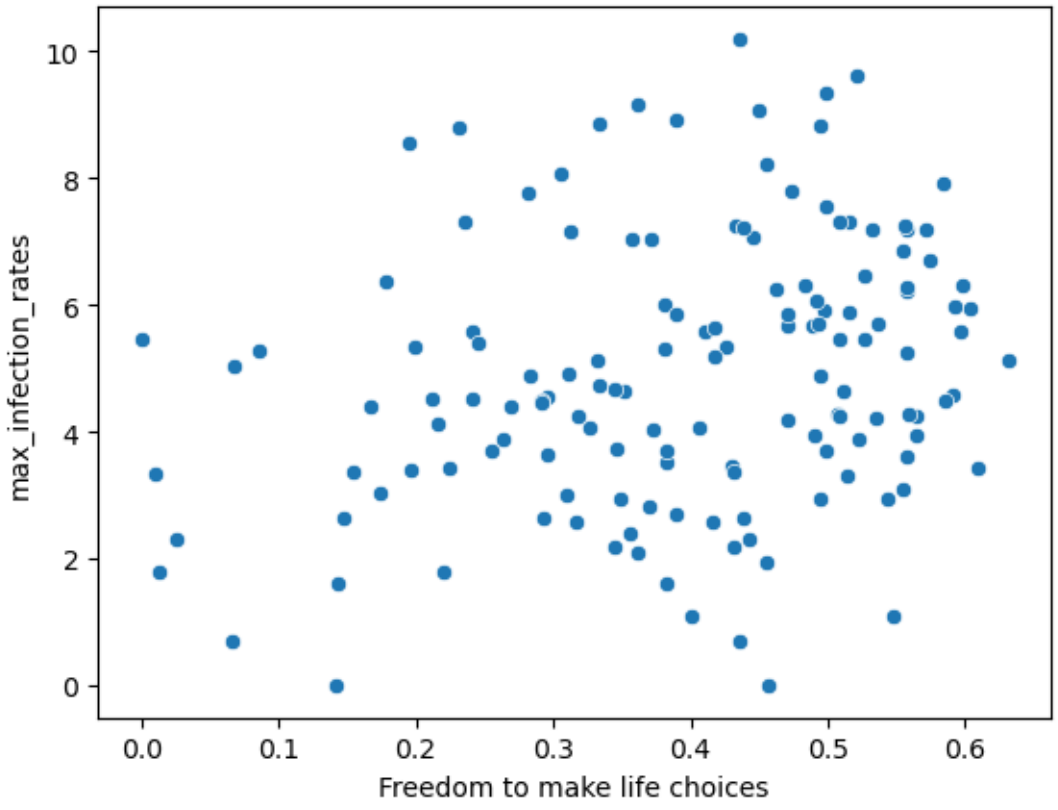
[153]:
```
sns.regplot(data=data, x="Healthy life expectancy", y=np.
↪log(data["max_infection_rates"]))
```

[153]: <Axes: xlabel='Healthy life expectancy', ylabel='max_infection_rates'>



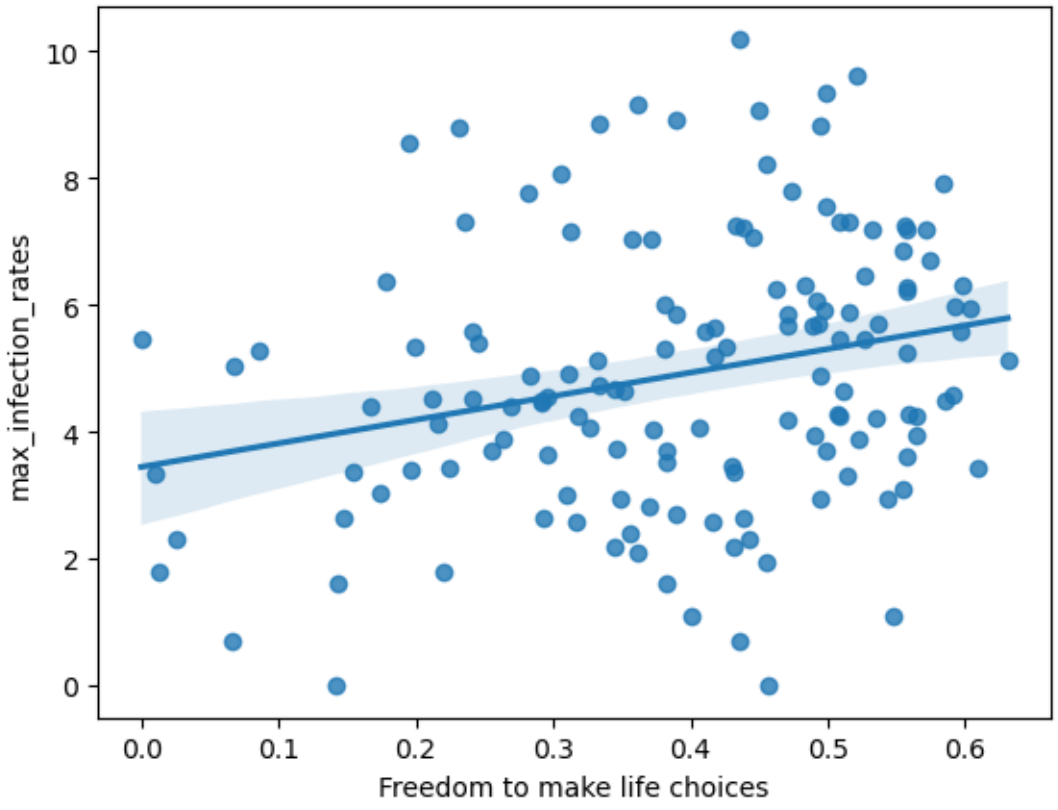
[151]:
```
# Task 5.4: Plotting Freedom to make life choices vs maximum infection rate


y_log = np.log(data["max_infection_rates"])

sns.scatterplot(data=data, x="Freedom to make life choices", y=y_log)
```

[151]:
```
<Axes: xlabel='Freedom to make life choices', ylabel='max_infection_rates'>
```



[154]:
```
sns.regplot(data=data, x="Freedom to make life choices", y=np.
↪log(data["max_infection_rates"]))
```

[154]: <Axes: xlabel='Freedom to make life choices', ylabel='max_infection_rates'>


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

[ ]: