

Covid19 Data Analysis

```
In [35]: #import required modules
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
print("Imported!")
```

Imported!

```
In [36]: #read csv file
covid19_dataset=pd.read_csv("covid19_Confirmed_dataset.csv")
covid19_dataset.head(10)
```

Out[36]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0
5	NaN	Antigua and Barbuda	17.0608	-61.7964	0	0	0	0	0
6	NaN	Argentina	-38.4161	-63.6167	0	0	0	0	0
7	NaN	Armenia	40.0691	45.0382	0	0	0	0	0
8	Australian Capital Territory	Australia	-35.4735	149.0124	0	0	0	0	0

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
9	New South Wales	Australia	-33.8688	151.2093	0	0	0	0	3

10 rows × 104 columns



In [37]: `covid19_dataset.shape`

Out[37]: (266, 104)

In [38]: `#remove columns not in use`
`covid19_dataset.drop(["Lat", "Long"], axis=1, inplace=True)`
`covid19_dataset.head(10)`

Out[38]:

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

10 rows × 102 columns

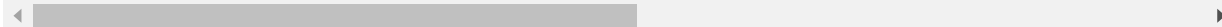


```
In [39]: covid19_aggregate=covid19_dataset.groupby("Country/Region").sum()  
covid19_aggregate.head()
```

Out[39]:

	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
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 × 100 columns

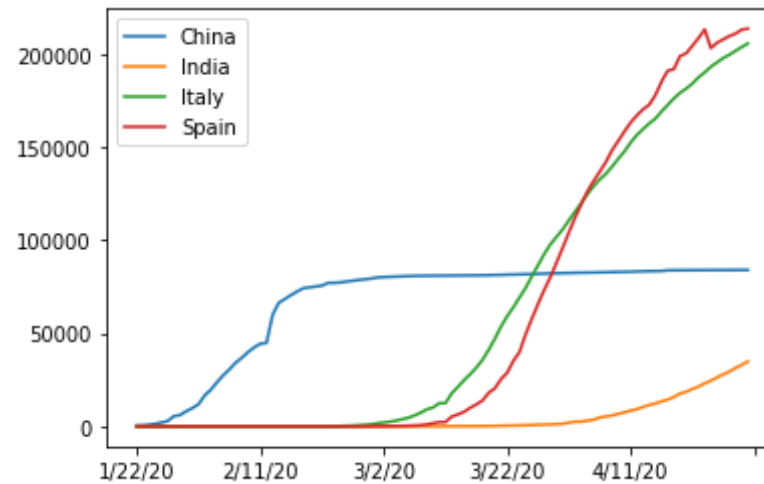


```
In [40]: covid19_aggregate.shape
```

Out[40]: (187, 100)

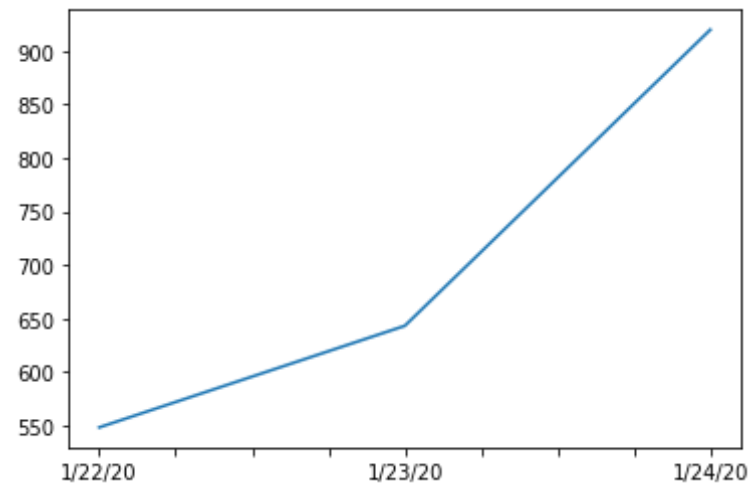
```
In [41]: covid19_aggregate.loc['China'].plot()  
covid19_aggregate.loc['India'].plot()  
covid19_aggregate.loc['Italy'].plot()  
covid19_aggregate.loc['Spain'].plot()  
plt.legend()
```

Out[41]: <matplotlib.legend.Legend at 0x20817da0948>



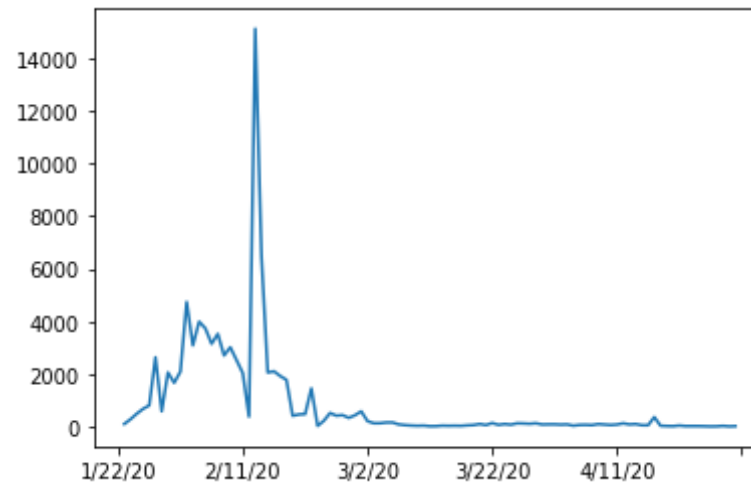
```
In [42]: covid19_aggregate.loc['China'][:3].plot()
```

```
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x20817e094c8>
```



```
In [43]: covid19_aggregate.loc["China"].diff().plot()
```

```
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x20817e8dfc8>
```



```
In [44]: #high infection rate in China,India,Italy,Spain  
covid19_aggregate.loc['China'].diff().max()
```

```
Out[44]: 15136.0
```

```
In [45]: covid19_aggregate.loc['India'].diff().max()
```

```
Out[45]: 1893.0
```

```
In [46]: covid19_aggregate.loc['Spain'].diff().max()
```

```
Out[46]: 9630.0
```

```
In [47]: covid19_aggregate.loc['Italy'].diff().max()
```

```
Out[47]: 6557.0
```

```
In [48]: #high infectionn rate in all countries  
all_countries=list(covid19_aggregate.index)  
max_infection=[]  
for c in all_countries:
```

```
max_infection.append(covid19_aggregate.loc[c].diff().max())
covid19_aggregate['max_infections']=max_infection
```

In [49]: covid19_aggregate.head(10)

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	1/31
Country/Region										
Afghanistan	0	0	0	0	0	0	0	0	0	0
Albania	0	0	0	0	0	0	0	0	0	0
Algeria	0	0	0	0	0	0	0	0	0	0
Andorra	0	0	0	0	0	0	0	0	0	0
Angola	0	0	0	0	0	0	0	0	0	0
Antigua and Barbuda	0	0	0	0	0	0	0	0	0	0
Argentina	0	0	0	0	0	0	0	0	0	0
Armenia	0	0	0	0	0	0	0	0	0	0
Australia	0	0	0	0	4	5	5	6	9	
Austria	0	0	0	0	0	0	0	0	0	0

10 rows × 101 columns



In [50]: *#dataframe with required columns*
covid19_data=pd.DataFrame(covid19_aggregate['max_infections'])
covid19_data

Out[50]:

	max_infections
Country/Region	
Afghanistan	232.0

max_infections	
Country/Region	
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 × 1 columns

```
In [51]: happiness_report=pd.read_csv("worldwide_happiness_report.csv")
happiness_report.head()
```

Out[51]:

	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 [52]: useless_col=["Overall rank","Score","Generosity","Perceptions of corrup
```

```
tion"]
happiness_report.drop(useless_col,axis=1,inplace=True)
happiness_report.head()
```

Out[52]:

	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 [55]: #changing data frames
happiness_report.set_index("Country or region",inplace=True)
happiness_report.head()
```

Out[55]:

	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 [57]: covid19_data.head()
```

Out[57]:

	max_infections
Country/Region	

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

In [58]: covid19_data.shape

Out[58]: (187, 1)

In [59]: happiness_report.head()

Out[59]:

	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 [60]: happiness_report.shape

Out[60]: (156, 4)

In [62]: data=covid19_data.join(happiness_report,how="inner")
data.head()

Out[62]:

	max_infections	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Afghanistan	232.0	0.350	0.517	0.361	0.000
Albania	34.0	0.947	0.848	0.874	0.383
Algeria	199.0	1.002	1.160	0.785	0.086
Argentina	291.0	1.092	1.432	0.881	0.471
Armenia	134.0	0.850	1.055	0.815	0.283

In [63]: `data.corr()`*#correlation of the matrix*

Out[63]:

	max_infections	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
max_infections	1.000000	0.250118	0.191958	0.289263	0.078196
GDP per capita	0.250118	1.000000	0.759468	0.863062	0.394603
Social support	0.191958	0.759468	1.000000	0.765286	0.456246
Healthy life expectancy	0.289263	0.863062	0.765286	1.000000	0.427892
Freedom to make life choices	0.078196	0.394603	0.456246	0.427892	1.000000

Result

In [64]: `data.head()`

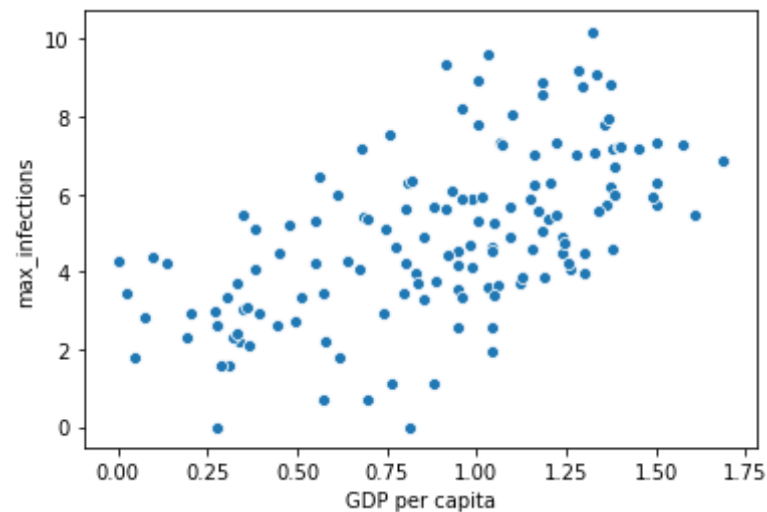
Out[64]:

	max_infections	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Afghanistan	232.0	0.350	0.517	0.361	0.000
Albania	34.0	0.947	0.848	0.874	0.383

	max_infections	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Algeria	199.0	1.002	1.160	0.785	0.086
Argentina	291.0	1.092	1.432	0.881	0.471
Armenia	134.0	0.850	1.055	0.815	0.283

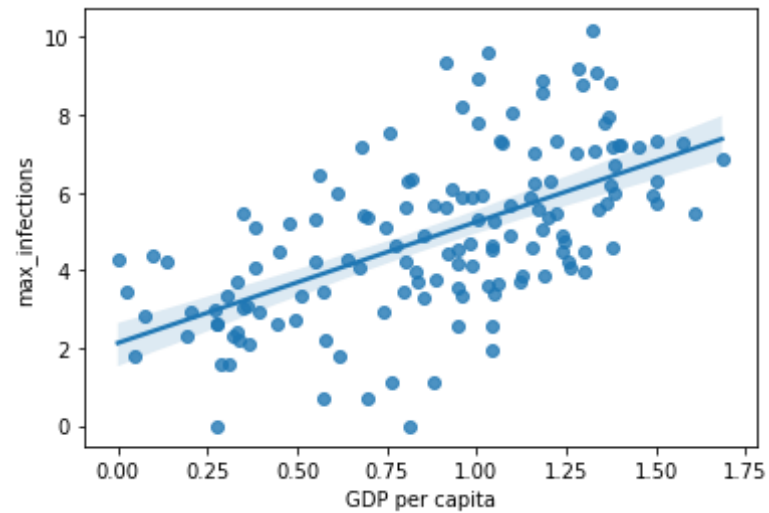
```
In [67]: x_axis=data["GDP per capita"]
y_axis=data["max_infections"]
sns.scatterplot(x_axis,np.log(y_axis))
```

```
Out[67]: <matplotlib.axes._subplots.AxesSubplot at 0x20817f8be08>
```



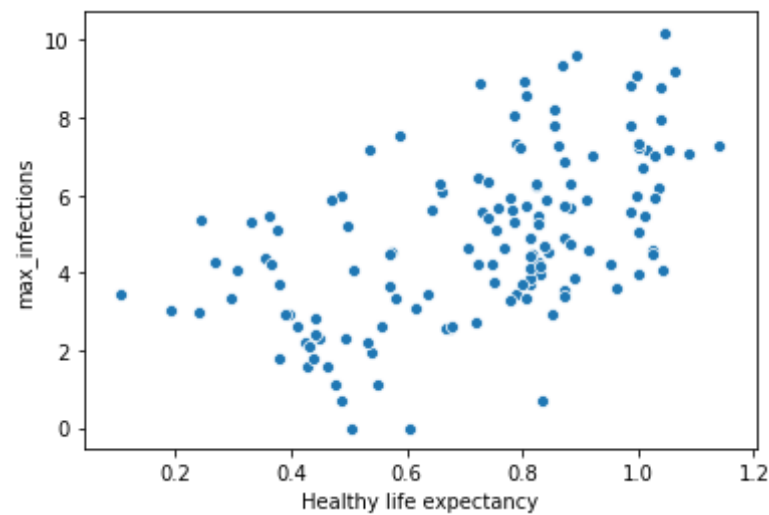
```
In [70]: sns.regplot(x_axis,np.log(y_axis))
```

```
Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x20818048948>
```



```
In [71]: x=data["Healthy life expectancy"]  
y=data["max_infections"]  
sns.scatterplot(x,np.log(y))
```

```
Out[71]: <matplotlib.axes._subplots.AxesSubplot at 0x208180b0c08>
```



```
In [77]: sns.regplot(x,np.log(y))
```

```
Out[77]: <matplotlib.axes._subplots.AxesSubplot at 0x208181417c8>
```

