Welcome to the Covid 19 Data Analysis Note book

Author : Poshan Pandey

Date : 6/5/2020

Let's import the modules

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
print("All modules imported!")
```

```
All modules imported!
```

Let's import the covid 19 datasets by John Hopkins University

https://github.com/CSSEGISandData/COVID-19 (https://github.com/CSSEGISandData/COVID-19)

In [3]:

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

In [4]:

```
corona_dataset_csv.head()
```

Out[4]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	5
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	

5 rows × 139 columns

Checking the shape of the data

In [5]:

```
corona_dataset_csv.shape
```

Out[5]:

```
(266, 139)
```

Deleting unnecessary columns

In [6]:
```
corona_dataset_csv.drop(["Lat", "Long"], axis = 1, inplace = True)
corona_dataset_csv.head(15)
```

Out[6]:

	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	...	5/
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	Northern Territory	Australia	0	0	0	0	0	0	0	0	...	
11	Queensland	Australia	0	0	0	0	0	0	0	1	...	
12	South Australia	Australia	0	0	0	0	0	0	0	0	...	
13	Tasmania	Australia	0	0	0	0	0	0	0	0	...	
14	Victoria	Australia	0	0	0	0	1	1	1	1	...	

15 rows × 137 columns

Aggregating the data of all province/state of similar country

In [7]:

```
aggregated_corona_dataset = corona_dataset_csv.groupby("Country/Regio
n").sum()
aggregated_corona_dataset.head(10)
```

Out[7]:

	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	...	5/26/
Country/Region												
Afghanistan	0	0	0	0	0	0	0	0	0	0	...	118
Albania	0	0	0	0	0	0	0	0	0	0	...	10
Algeria	0	0	0	0	0	0	0	0	0	0	...	86
Andorra	0	0	0	0	0	0	0	0	0	0	...	7
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	...	132
Armenia	0	0	0	0	0	0	0	0	0	0	...	74
Australia	0	0	0	0	4	5	5	6	9	9	...	71
Austria	0	0	0	0	0	0	0	0	0	0	...	165

10 rows × 135 columns

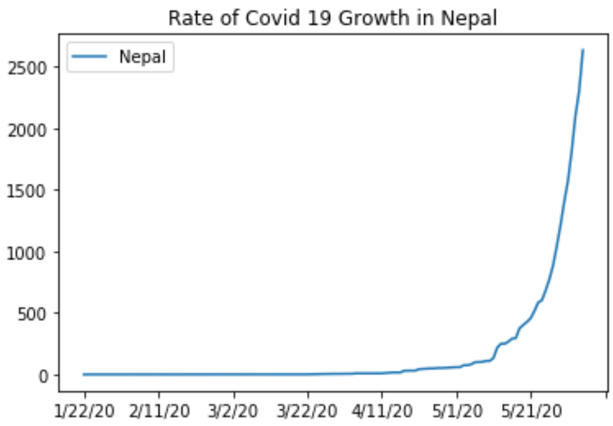
Visualizing the Corona Infection data of Nepal

In [8]:

```
aggregated_corona_dataset.loc["Nepal"].plot()
plt.title("Rate of Covid 19 Growth in Nepal")
plt.legend()
```

Out[8]:

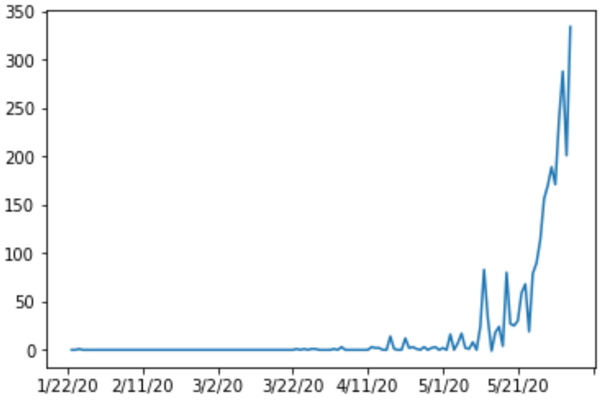
```
<matplotlib.legend.Legend at 0x21a2bd54e88>
```



Calculating derivative of above curve and finding the maximum infection rate

```
In [10]: aggregated_corona_dataset.loc["Nepal"].diff().plot()
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x21a2be86408>
```



```
In [11]: aggregated_corona_dataset.loc["Nepal"].diff().max()
```

```
Out[11]: 334.0
```

Finding maximum infection rate for all the countries and Adding it to new column in Dataframe

```
In [12]: countries = list(aggregated_corona_dataset.index)
         max_infection_rates = []
         for c in countries:
             max_infection_rates.append(aggregated_corona_dataset.loc[c].diff().
         max())
         aggregated_corona_dataset["max_infection_rates"] = max_infection_rates
         aggregated_corona_dataset.head()
```

Out[12]:

	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	...	5/27/
Country/Region												
Afghanistan	0	0	0	0	0	0	0	0	0	0	...	124
Albania	0	0	0	0	0	0	0	0	0	0	...	10
Algeria	0	0	0	0	0	0	0	0	0	0	...	88
Andorra	0	0	0	0	0	0	0	0	0	0	...	7
Angola	0	0	0	0	0	0	0	0	0	0	...	

5 rows × 136 columns

Creating new dataframe with countries and maximum infection rate only

In [13]:

```
corona_data = pd.DataFrame(aggregated_corona_dataset["max_infection_rates"])
corona_data.head()
```

Out[13]:

	max_infection_rates
Country/Region	
Afghanistan	866.0
Albania	34.0
Algeria	199.0
Andorra	79.0
Angola	8.0

Importing the World Happiness Report dataset

In [16]:

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

Out[16]:

	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

Deleting the unnecessary columns and changing indices to Country or region

In [17]:

```
useless_cols = ["Overall rank", "Score", "Generosity", "Perceptions of
corruption"]
happiness_report_csv.drop(useless_cols, axis = 1, inplace = True)
happiness_report_csv.set_index("Country or region", inplace= True)
happiness_report_csv.head()
```

Out[17]:

	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

Comparing Number of countries in Happiness and Covid 19 datasets

In [21]:

```
corona_data.shape
```

Out[21]:

```
(188, 1)
```

In [20]:

```
happiness_report_csv.shape
```

Out[20]:

```
(156, 4)
```

Number of countries in Corina dataset is more than World Happiness Report Dataset

So, We have to join them ussing Inner join

In [22]:

```
final_data = corona_data.join(happiness_report_csv, how = "inner")
final_data.head()
```

Out[22]:

	max_infection_rates	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Afghanistan	866.0	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	949.0	1.092	1.432	0.881	0.471
Armenia	697.0	0.850	1.055	0.815	0.283

Calculating Correleation Matrix for the final Data

In [23]:
```
final_data.corr()
```

Out[23]:

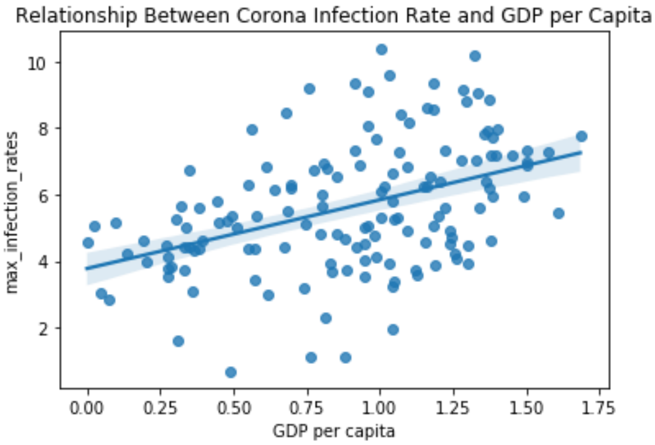
	max_infection_rates	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
max_infection_rates	1.000000	0.207071	0.158977	0.218118	0.071825
GDP per capita	0.207071	1.000000	0.757521	0.859431	0.394799
Social support	0.158977	0.757521	1.000000	0.751632	0.456317
Healthy life expectancy	0.218118	0.859431	0.751632	1.000000	0.423146
Freedom to make life choices	0.071825	0.394799	0.456317	0.423146	1.000000

Visualizing our final result

Plotting GDP vs Maximum Infection Rate

In [24]:
```
x = final_data["GDP per capita"]
y = final_data["max_infection_rates"]
sns.regplot(x,np.log(y)).set_title("Relationship Between Corona Infection Rate and GDP per Capita")
```

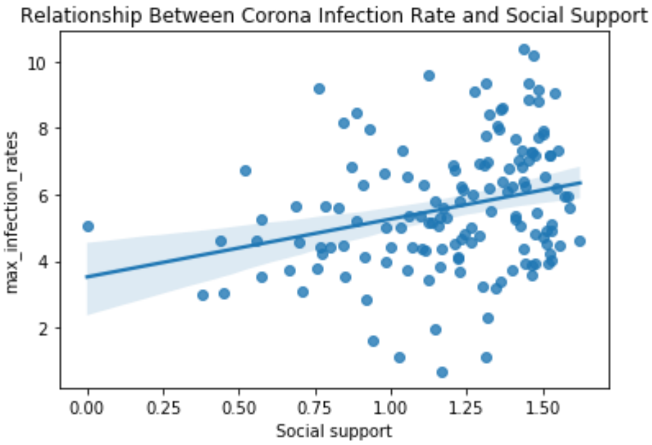
Out[24]:
```
Text(0.5, 1.0, 'Relationship Between Corona Infection Rate and GDP per
Capita')
```



Plotting Social support vs Maximum Infection Rate

```
In [25]: x = final_data["Social support"]
         y = final_data["max_infection_rates"]
         sns.regplot(x,np.log(y)).set_title("Relationship Between Corona Infecti
         on Rate and Social Support")
```

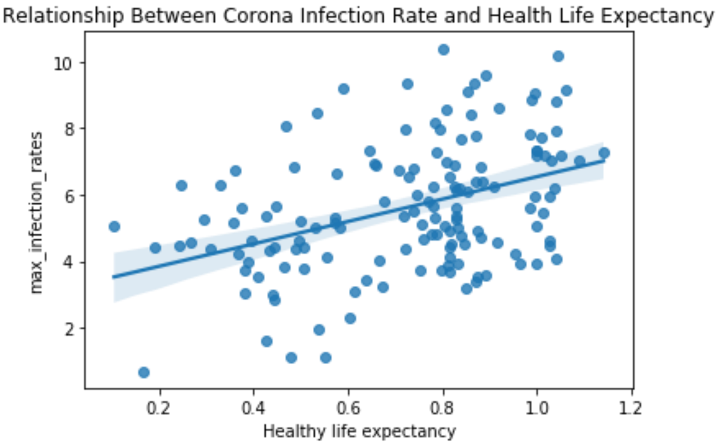
```
Out[25]: Text(0.5, 1.0, 'Relationship Between Corona Infection Rate and Social S
         upport')
```



Plotting Social support vs Health Life Expectancy

```
In [26]: x = final_data["Healthy life expectancy"]
         y = final_data["max_infection_rates"]
         sns.regplot(x,np.log(y)).set_title("Relationship Between Corona Infecti
         on Rate and Health Life Expectancy")
```

```
Out[26]: Text(0.5, 1.0, 'Relationship Between Corona Infection Rate and Health L
         ife Expectancy')
```



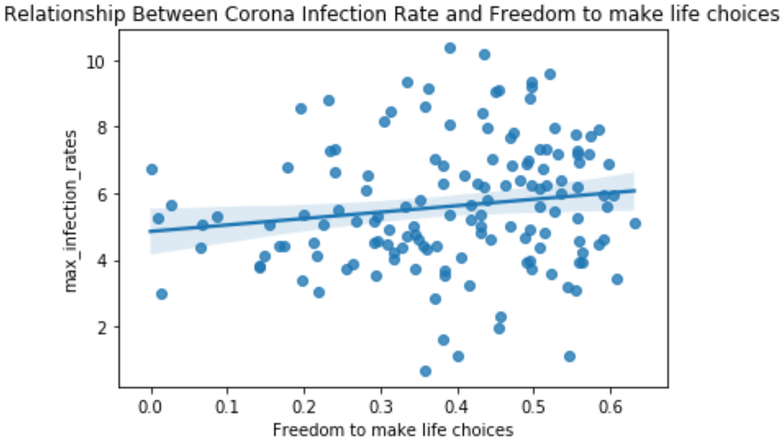
Plotting Social support vs Freedom to make life choices

In [27]:

```
x = final_data["Freedom to make life choices"]
y = final_data["max_infection_rates"]
sns.regplot(x,np.log(y)).set_title("Relationship Between Corona Infecti
on Rate and Freedom to make life choices")
```

Out[27]:

```
Text(0.5, 1.0, 'Relationship Between Corona Infection Rate and Freedom
to make life choices')
```



------------------------------------The End--------------------------------

In [ ]:

```
```