Covid data analysis

1) Analyzed the spread of covid in different countries over a period of time.

2) calculated max infection rates in different countries.
3) Analyzed the dependency of other factors from a happiness report on the max infection rates.

Skills used -

cleaning the data

plotting the data and finding maxima

EDA

joining two data sets

correlation analysis

used seaborn to plot different types of graphs - scatter plot and regression plot

regression analysis - shows the best fit line and the spread of data accordingly.

importing all the necessary libraries

```
#case study
#eda is about asking questions

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

importing the covid dataset and viewing the first 5 entries.

```
#importing the datasets
dataset = pd.read_csv("covid19_Confirmed_dataset[1].csv")
dataset.head(5)
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	4/21/20	4/22/20	4/23/20	4/24/20	4/25/20	4/2
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	1092	1176	1279	1351	1463	
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	609	634	663	678	712	
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	2811	2910	3007	3127	3256	
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	717	723	723	731	738	
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	24	25	25	25	25	

checking the shape of the data and dropping some unnecessary columns.

```
#checking the shape of the data
dataset.shape
```

```
(266, 104)
```

```
#deleting the useless columns  i.e. lat and long
df = dataset.drop(["Lat",'Long'],axis= 1 , inplace=True) #axis - 0 for deleting the row , 1 for deleting the column
dataset.head()
```

	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	4/
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

making the first row i.e. the index as the country to classify data based on that, also no state or province given so to eliminate that column. Then aggregating the dataset to get sum values for plotting.

[5]:
```
#aggregate the rows by country
corona_dataset_aggregated = dataset.groupby("Country/Region").sum() #grouping the data based on certain attribute
```

[6]:
```
corona_dataset_aggregated.head() #shows the dataset with resoect to the countries
```

[6]:

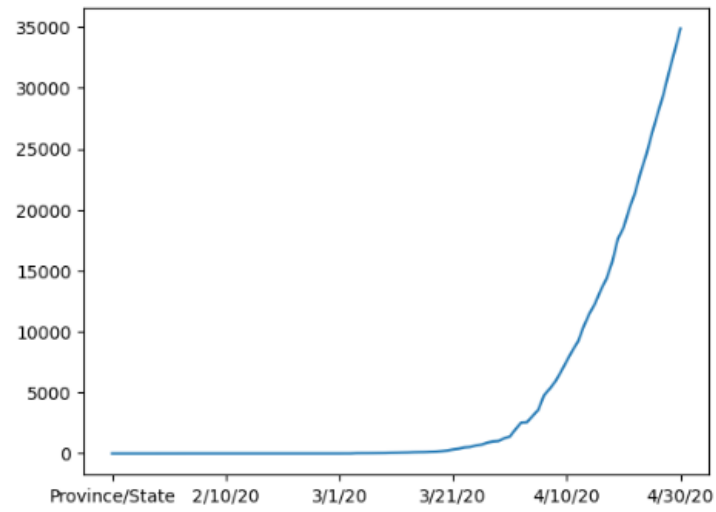
Country/Region	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	...	4/21/20	4/22/20	4/23/20	4/24/20	4/25/20	4/
Afghanistan	0	0	0	0	0	0	0	0	0	0	...	1092	1176	1279	1351	1463	
Albania	0	0	0	0	0	0	0	0	0	0	...	609	634	663	678	712	
Algeria	0	0	0	0	0	0	0	0	0	0	...	2811	2910	3007	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	

5 rows × 100 columns

Plotting the infection rates based on dates for India.

```
corona_dataset_aggregated.loc["India"].plot()
```

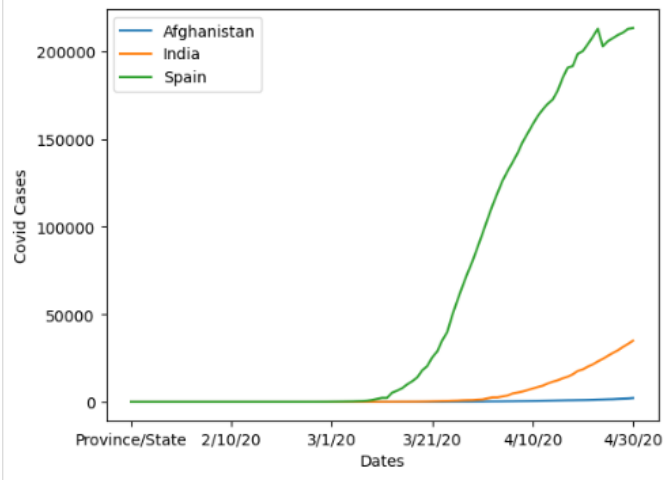
```
<Axes: >
```



Plotting for multiple countries in the same plot.

```
corona_dataset_aggregated.loc["Afghanistan"].plot()
corona_dataset_aggregated.loc["India"].plot()
corona_dataset_aggregated.loc["Spain"].plot()
plt.xlabel("Dates")
plt.ylabel("Covid Cases")
plt.legend()
```

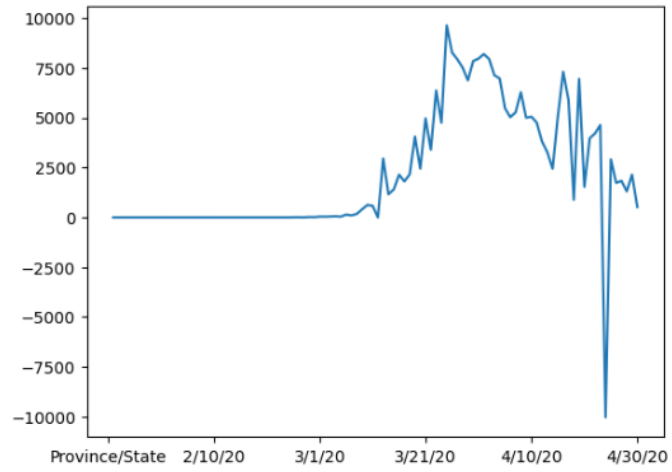
```
<matplotlib.legend.Legend at 0x14e94019990>
```



plotting first derivative of Spain to see maxima (peak) and calculating the maxima of different countries.

```
#calculating the first derivative
corona_dataset_aggregated.loc["Spain"].diff().plot()
```

```
<Axes: >
```




```
#maximum infection rate
corona_dataset_aggregated.loc["Afghanistan"].diff().max()
```

```
232
```

```
corona_dataset_aggregated.loc["India"].diff().max()
```

```
1893
```

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

```
9630
```

creating a new data frame that stores the max infection rates and printing it. (to be used later for correlation and other analysis.)

```
[18]: #create a new dataframe
      corona_data = pd.DataFrame(corona_dataset_aggregated["Max_infection_rates"])

[19]: corona_data
```

[19]:

Country/Region	Max_infection_rates
Afghanistan	232.0
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

importing a happiness report dataset using pandas.

```
[20]: #importing the world happiness data
      happiness_report = pd.read_csv("worldwide_happiness_report[1].csv")

[21]: happiness_report
```

[21]:

	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
...	...	...	...	...	...	...	...	...	...
151	152	Rwanda	3.334	0.359	0.711	0.614	0.555	0.217	0.411
152	153	Tanzania	3.231	0.476	0.885	0.499	0.417	0.276	0.147
153	154	Afghanistan	3.203	0.350	0.517	0.361	0.000	0.158	0.025
154	155	Central African Republic	3.083	0.026	0.000	0.105	0.225	0.235	0.035
155	156	South Sudan	2.853	0.306	0.575	0.295	0.010	0.202	0.091

156 rows × 9 columns

dropping the unnecessary columns.

[22]:
```
#dropping the useless columns
useless_columns = ["Overall rank","Generosity", "Perceptions of corruption"]
happiness_report.drop(useless_columns, axis = 1  , inplace = True)
happiness_report
```

[22]:

	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
0	Finland	7.769	1.340	1.587	0.986	0.596
1	Denmark	7.600	1.383	1.573	0.996	0.592
2	Norway	7.554	1.488	1.582	1.028	0.603
3	Iceland	7.494	1.380	1.624	1.026	0.591
4	Netherlands	7.488	1.396	1.522	0.999	0.557
...	...	...	...	...	...	...
151	Rwanda	3.334	0.359	0.711	0.614	0.555
152	Tanzania	3.231	0.476	0.885	0.499	0.417
153	Afghanistan	3.203	0.350	0.517	0.361	0.000
154	Central African Republic	3.083	0.026	0.000	0.105	0.225
155	South Sudan	2.853	0.306	0.575	0.295	0.010

156 rows × 6 columns

[23]:
```
happiness_report
happiness_report.drop(["Score"],axis = 1 , inplace=True)
happiness_report
```

[23]:

	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
...	...	...	...	...	...
151	Rwanda	0.359	0.711	0.614	0.555
152	Tanzania	0.476	0.885	0.499	0.417
153	Afghanistan	0.350	0.517	0.361	0.000
154	Central African Republic	0.026	0.000	0.105	0.225
155	South Sudan	0.306	0.575	0.295	0.010

checking the shape of both of the datasets and joining them

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

[25]: #joining the datasets
      corona_data.shape

[25]: (187, 1)

[26]: happiness_report.shape

[26]: (156, 4)

[27]: data = corona_data.join(happiness_report, how = "inner")
      data
```

[27]:

	Max_infection_rates	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
...	...	...	...	...	...
Venezuela	29.0	0.960	1.427	0.805	0.154
Vietnam	19.0	0.741	1.346	0.851	0.543
Yemen	5.0	0.287	1.163	0.463	0.143
Zambia	9.0	0.578	1.058	0.426	0.431
Zimbabwe	8.0	0.366	1.114	0.433	0.361

143 rows × 5 columns

got a new dataset with max_infection_rates as a column.

stored this in data

creating a correlation matrix to understand how these variables depend on each other

```
[28]: #creating a correlation matrix to see the correlation using carl pearson
      data.corr()
```

[28]:

	Max_infection_rates	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices
Max_infection_rates	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

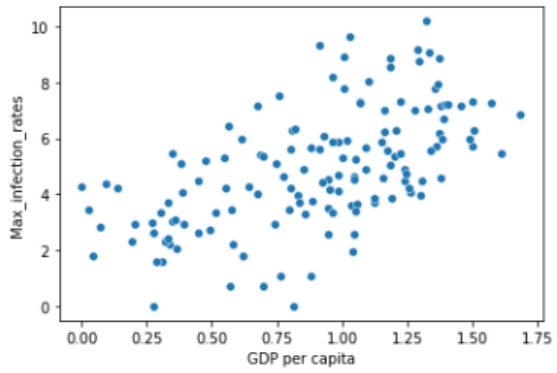
as observed - healthy life expectancy and GDP are highly correlated.

max_infection_rates has highest correlation with healthy life expectancy and least correlation with freedom to make choices.

Visualizing the spread of the data by plotting scatter plots and regression plots for different parameters.

```
#visualising the data
#plotting graph between gdp and max inflation rates
x = data["GDP per capita"]
y = data["Max_infection_rates"]
sns.scatterplot(x=x,y =np.log(y)) #i.e. log of the max_inflection_rate values in order to have better estimation
```

```
<AxesSubplot:xlabel='GDP per capita', ylabel='Max_infection_rates'>
```



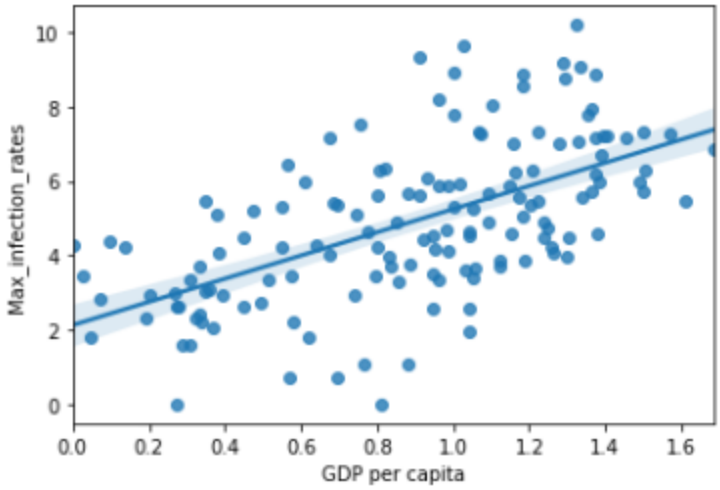
The graph shows a positive correlation between GDP and max infection rate,

could be due to higher tests performed due to better facilities or population density etc.

Note - correlation does not mean causation. That is it does not directly indicate that these factors have a cause and effect relationship.

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

[30]: <AxesSubplot:xlabel='GDP per capita', ylabel='Max_infection_rates'>
```



Regression line - shows the general trend of the data points with the help of the best fit line, also shows that the data is widely scattered.

Similar graph can be plotted for all the factors to understand the correlation between them to make some basic speculations.