

Data Analytics of the Spread and Impact of COVID-19 Globally

Importing modules

```
In [4]: import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
print('modules are imported')
```

modules are imported

Loading the Dataset

```
In [7]: dataset_url = 'https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv'
df = pd.read_csv(dataset_url)
```

Check the dataframe

```
In [9]: df.head()
```

Out[9]:

	Date	Country	Confirmed	Recovered	Deaths
0	2020-01-22	Afghanistan	0	0	0
1	2020-01-23	Afghanistan	0	0	0
2	2020-01-24	Afghanistan	0	0	0
3	2020-01-25	Afghanistan	0	0	0
4	2020-01-26	Afghanistan	0	0	0

```
In [11]: df.tail()
```

Out[11]:

	Date	Country	Confirmed	Recovered	Deaths
161563	2022-04-12	Zimbabwe	247094	0	5460
161564	2022-04-13	Zimbabwe	247160	0	5460
161565	2022-04-14	Zimbabwe	247208	0	5462
161566	2022-04-15	Zimbabwe	247237	0	5462
161567	2022-04-16	Zimbabwe	247237	0	5462

Shape of the dataframe

```
In [13]: df.shape
```

Out[13]: (161568, 5)

Some preprocessing to reveiw from the dataset

```
In [16]: df = df[df.Confirmed > 0]
```

```
In [14]: df.head()
```

Out[14]:

	Date	Country	Confirmed	Recovered	Deaths
0	2020-01-22	Afghanistan	0	0	0
1	2020-01-23	Afghanistan	0	0	0

2	2020-01-24	Afghanistan	0	0	0
3	2020-01-25	Afghanistan	0	0	0
4	2020-01-26	Afghanistan	0	0	0

let's see data related to a country for example Italy

```
In [25]: df[df.Country == 'Italy']
```

Out[25]:

	Date	Country	Confirmed	Recovered	Deaths
70176	2020-01-22	Italy	0	0	0
70177	2020-01-23	Italy	0	0	0
70178	2020-01-24	Italy	0	0	0
70179	2020-01-25	Italy	0	0	0
70180	2020-01-26	Italy	0	0	0
...
70987	2022-04-12	Italy	15404809	0	161032
70988	2022-04-13	Italy	15467395	0	161187
70989	2022-04-14	Italy	15533012	0	161336
70990	2022-04-15	Italy	15595302	0	161469
70991	2022-04-16	Italy	15659835	0	161602

816 rows × 5 columns

Global spread of Covid19

```
In [19]: fig = px.choropleth(df , locations = 'Country' , locationmode='country
names',color='Confirmed',animation_frame='Date')
fig.update_layout(title_text = 'Global Spread of Covid-19')
fig.show()
```

Global Deaths of Covid19

```
In [26]: fig = px.choropleth(df , locations = 'Country' , locationmode='country
names',color='Deaths',animation_frame='Date')
fig.update_layout(title_text = 'Global Number of Death Cases')
fig.show()
```

Visualizing how intensive the Covid19 Transmission has been in each of the country

```
In [23]: df_China = df[df.Country == 'China']
df_China.head()
```

Out[23]:

Date	Country	Confirmed	Recovered	Deaths
------	---------	-----------	-----------	--------

30192	2020-01-22	China	548	28	17
30193	2020-01-23	China	643	30	18
30194	2020-01-24	China	920	36	26
30195	2020-01-25	China	1406	39	42
30196	2020-01-26	China	2075	49	56

Selecting for Number of Confirmed Cases in China

```
In [28]: df_China = df_China[['Date', 'Confirmed']]
In [30]: df_China.head()
```

Out[30]:

	Date	Confirmed
30192	2020-01-22	548
30193	2020-01-23	643
30194	2020-01-24	920
30195	2020-01-25	1406
30196	2020-01-26	2075

Calculating the first derivation of confrimed column

```
In [:]: df_China['Infection Rate'] = df_China['Confirmed'].diff()
In [42]: df_China.head()
```

Out[42]:

	Date	Confirmed	Infection Rate	Infection rate
30192	2020-01-22	548.0	NaN	548.0
30193	2020-01-23	643.0	95.0	643.0
30194	2020-01-24	920.0	277.0	920.0
30195	2020-01-25	1406.0	486.0	1406.0
30196	2020-01-26	2075.0	669.0	2075.0

```
In [50]: px.line(df_China , x = 'Date' , y = ['Confirmed','Infection Rate'])
```

```
In [46]: df_China['Infection Rate'].max()
```

Out[46]:77402.0
Calculating Maximum infection rate for all of the countries

```
In [51]: df.head()
```

Out[51]:

	Date	Country	Confirmed	Recovered	Deaths
33	2020-02-24	Afghanistan	5	0	0
34	2020-02-25	Afghanistan	5	0	0
35	2020-02-26	Afghanistan	5	0	0
36	2020-02-27	Afghanistan	5	0	0
37	2020-02-28	Afghanistan	5	0	0

```
In [64]: Countries = list(df['Country'].unique())
max_infection_rates = []
for c in Countries :
    MIR = df[df.Country == c].Confirmed.diff().max()
    max_infection_rates.append(MIR)
```

Creating a new Dataframe for the Maximum Infection Rate in each Country

```
In [66]: df_MIR = pd.DataFrame()
df_MIR['Country'] = Countries
df_MIR['Max Infection Rate'] = max_infection_rates
df_MIR.head()
```

Out[66]:

	Country	Max Infection Rate
0	Afghanistan	3243.0
1	Albania	4789.0
2	Algeria	2521.0
3	Andorra	2313.0
4	Angola	5035.0

Ploting the barchart : maximum infection rate of each country

```
In [68]: px.bar(df_MIR, x='Country' , y='Max Infection Rate', color='Country' , title='Global
Maximum Infection Rate', log_y=True)
```

How National Lockdowns Impacts Covid19 transmission in Italy

COVID19 pandemic lockdown in Italy

On 9 March 2020, the government of Italy under Prime Minister Giuseppe Conte imposed a national quarantine, restricting the movement of the population except for necessity, work, and health circumstances, in response to the growing pandemic of COVID-19 in the country. [source](#)

```
In [92]: Italy_lockdown_start_date = '2020-03-09'
Italy_lockdown_a_month_later = '2020-04-09'
```

```
In [69]: df.head()
```

Out[69]:

	Date	Country	Confirmed	Recovered	Deaths
33	2020-02-24	Afghanistan	5	0	0
34	2020-02-25	Afghanistan	5	0	0
35	2020-02-26	Afghanistan	5	0	0
36	2020-02-27	Afghanistan	5	0	0
37	2020-02-28	Afghanistan	5	0	0

Data related to Italy

```
In [72]: df_Italy = df[df.Country == 'Italy']  
lets check the dataframe  
In [73]: df_Italy.head()
```

Out[73]:

	Date	Country	Confirmed	Recovered	Deaths
70185	2020-01-31	Italy	2	0	0
70186	2020-02-01	Italy	2	0	0
70187	2020-02-02	Italy	2	0	0
70188	2020-02-03	Italy	2	0	0
70189	2020-02-04	Italy	2	0	0

Calculating the infection rate in Italy

```
In [74]: df_Italy['Infection rate'] = df_Italy.Confirmed.diff()  
df_Italy.head()
```

C:\Users\idowuay1\AppData\Local\Temp\1\ipykernel_8612\2459636895.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

Out[74]:

	Date	Country	Confirmed	Recovered	Deaths	Infection rate
70185	2020-01-31	Italy	2	0	0	NaN
70186	2020-02-01	Italy	2	0	0	0.0
70187	2020-02-02	Italy	2	0	0	0.0
70188	2020-02-03	Italy	2	0	0	0.0
70189	2020-02-04	Italy	2	0	0	0.0

visualization for Infection Rate in Italy

```
In [106]: fig = px.line(df_Italy , x = 'Date' , y = 'Infection rate' , title = "Before and  
After Lockdoown in Italy")  
fig.add_shape(  
    dict(  
        type="line",  
        x0=Italy_lockdown_start_date,  
        y0=0,  
        x1=Italy_lockdown_start_date,  
        y1= df_Italy['Infection rate'].max(),  
        line = dict(color='red' , width=2)  
    )  
)
```

```
)
fig.add_annotation(
    dict(
        x = Italy_lockdown_start_date,
        y = df_Italy['Infection rate'].max(),
        text = "Starting date of the lockdown"
    )
)
```

How National Lockdowns Impacts Covid19 active cases in Italy

```
In [108]: df_Italy.head()
```

Out[108]:

	Date	Country	Confirmed	Recovered	Deaths	Infection rate
70185	2020-01-31	Italy	2	0	0	NaN
70186	2020-02-01	Italy	2	0	0	0.0
70187	2020-02-02	Italy	2	0	0	0.0
70188	2020-02-03	Italy	2	0	0	0.0
70189	2020-02-04	Italy	2	0	0	0.0

Calculating the number of active cases day by day

```
In [109]:
```



```
df_Italy['Death Rate'] = df_Italy.Deaths.diff()
```

C:\Users\idowuay1\AppData\Local\Temp\1\ipykernel_8612\1984322640.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

let's check the dataframe again

```
In [110]: df_Italy.head()
```

Out[110]:

	Date	Country	Confirmed	Recovered	Deaths	Infection rate	Death Rate
70185	2020-01-31	Italy	2	0	0	NaN	NaN
70186	2020-02-01	Italy	2	0	0	0.0	0.0
70187	2020-02-02	Italy	2	0	0	0.0	0.0
70188	2020-02-03	Italy	2	0	0	0.0	0.0
70189	2020-02-04	Italy	2	0	0	0.0	0.0

Ploting a line chart to compare COVID19 national lockdowns impacts on spread of the virus and number of active cases in Italy

```
In [115]: fig = px.line(df_Italy,x='Date',y=['Infection rate', 'Death Rate'])  
fig.show()
```

```
In [117]: df_Italy['Infection rate'] = df_Italy['Infection rate']/df_Italy['Infection
rate'].max()
df_Italy['Death Rate'] = df_Italy['Death Rate']/df_Italy['Death Rate'].max()
```

C:\Users\idowuay1\AppData\Local\Temp\1\ipykernel_8612\1808481513.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\idowuay1\AppData\Local\Temp\1\ipykernel_8612\1808481513.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
In [118]: fig = px.line(df_Italy,x='Date',y=['Infection rate','Death Rate'])
fig.show()
```

Next Visualization to work on: COVID19 pandemic lockdown in Germany Lockdown was started in Freiburg, Baden-Württemberg and Bavaria on 20 March 2020. Three days later, it was expanded to the whole of Germany

```
In []: Germany_lockdown_start_date = '2020-03-23'
       Germany_lockdown_a_month_later = '2020-04-23'
```

let's select the data related to Germany

```
In []:
```

let's check the dataframe

```
In []:
```

selecting the needed column

```
In []:
```

let's check it again

```
In []:
```

let's calculate the infection rate in Germany

```
In []:
```

let's check the dataframe

```
In []:
```

now let's plot the line chart

```
In []:
```

```
In []:
```

let's do some scaling

```
In []:
```

let's plot the line chart

```
In []:
```