

# Project – Analyzing the trends of COVID-19 with Python

Problem Statement: Given data about COVID 19 patients, write code to visualize the impact and analyze the trend of rate of infection and recovery as well as make predictions about the number of cases expected a week in future based on the current trends Dataset: CSV and Excel files containing data about Number of COVID 19 Confirmed, Deaths and Recovered patients both around the World and in India.

Download Link Guidelines: • Use pandas to accumulate data from multiple data files • Use plotly (visualization library) to create interactive visualizations • Use Facebook prophet library to make time series models • Visualize the prediction by combining these technologies

```
In [1]: import os
os.chdir('C:\\Users\\veena\\OneDrive\\Desktop\\intellipaath assignment pdf s')
```

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.express as px
```

```
In [3]: data=pd.read_csv('covid-19 file.csv')
data.head()
```

```
Out[3]:
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered
0	NaN	Thailand	15.0000	101.0000	1/22/20	2.0	0.0	0.0
1	NaN	Japan	36.0000	138.0000	1/22/20	2.0	0.0	0.0
2	NaN	Singapore	1.2833	103.8333	1/22/20	0.0	0.0	0.0
3	NaN	Nepal	28.1667	84.2500	1/22/20	0.0	0.0	0.0
4	NaN	Malaysia	2.5000	112.5000	1/22/20	0.0	0.0	0.0

```
In [4]: data.shape
```

```
Out[4]: (19220, 8)
```

```
In [5]: data.isnull().sum()
```

```
Out[5]: Province/State    10788
Country/Region          0
Lat                      0
Long                     0
Date                     0
Confirmed                 1
Deaths                   1
Recovered                 1
dtype: int64
```

```
In [6]: data=data.dropna()
```

```
In [7]: data.isnull().sum()
```

```
Out[7]: Province/State    0
Country/Region          0
Lat                      0
Long                     0
Date                     0
Confirmed                 0
Deaths                   0
Recovered                 0
dtype: int64
```

```
In [8]: data1=data.drop(['Province/State', 'Lat', 'Long'],axis=1)
data1.head()
```

```
Out[8]:
```

	Country/Region	Date	Confirmed	Deaths	Recovered
5	Canada	1/22/20	0.0	0.0	0.0
6	Australia	1/22/20	0.0	0.0	0.0
7	Australia	1/22/20	0.0	0.0	0.0
8	Australia	1/22/20	0.0	0.0	0.0
19	Australia	1/22/20	0.0	0.0	0.0

```
In [9]: data2= data1.groupby('Country/Region')
data2.first()
```

Out[9]:

	Date	Confirmed	Deaths	Recovered
Country/Region				
Australia	1/22/20	0.0	0.0	0.0
Canada	1/22/20	0.0	0.0	0.0
China	1/22/20	444.0	17.0	28.0
Cruise Ship	1/22/20	0.0	0.0	0.0
Denmark	1/22/20	0.0	0.0	0.0
France	1/22/20	0.0	0.0	0.0
Netherlands	1/22/20	0.0	0.0	0.0
US	1/22/20	0.0	0.0	0.0
United Kingdom	1/22/20	0.0	0.0	0.0

In [10]: data2

Out[10]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000218C99226D0>

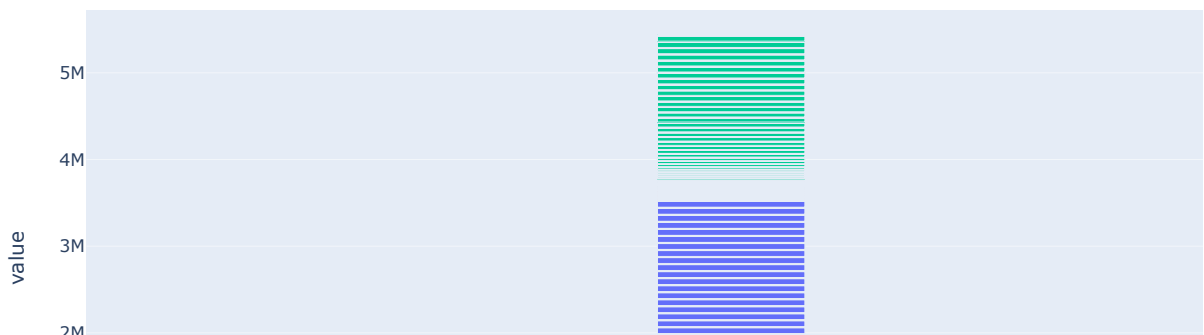
In [11]: import plotly.express as px

In [12]: import plotly.express as px

```
plot = px.data.medals_wide()
```

```
fig = px.bar(data1, x="Country/Region", y=["Confirmed", "Deaths", "Recovered"], title="Covid -19 Death,confirm, Recovery r", fig.show()
```

Covid -19 Death,confirm, Recovery rate



In [13]: df=pd.read\_csv('covid-19 file.csv')  
df.head()

Out[13]:

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered
0	NaN	Thailand	15.0000	101.0000	1/22/20	2.0	0.0	0.0
1	NaN	Japan	36.0000	138.0000	1/22/20	2.0	0.0	0.0
2	NaN	Singapore	1.2833	103.8333	1/22/20	0.0	0.0	0.0
3	NaN	Nepal	28.1667	84.2500	1/22/20	0.0	0.0	0.0
4	NaN	Malaysia	2.5000	112.5000	1/22/20	0.0	0.0	0.0

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

```
Out[14]:
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered
0	NaN	Thailand	15.0000	101.0000	1/22/20	2.0	0.0	0.0
1	NaN	Japan	36.0000	138.0000	1/22/20	2.0	0.0	0.0
2	NaN	Singapore	1.2833	103.8333	1/22/20	0.0	0.0	0.0
3	NaN	Nepal	28.1667	84.2500	1/22/20	0.0	0.0	0.0
4	NaN	Malaysia	2.5000	112.5000	1/22/20	0.0	0.0	0.0

```
In [15]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19220 entries, 0 to 19219
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Province/State        8432 non-null   object  
1   Country/Region        19220 non-null  object  
2   Lat                   19220 non-null  float64  
3   Long                  19220 non-null  float64  
4   Date                  19220 non-null  object  
5   Confirmed             19219 non-null  float64  
6   Deaths               19219 non-null  float64  
7   Recovered             19219 non-null  float64  
dtypes: float64(5), object(3)
memory usage: 1.2+ MB
```

```
In [16]: df=df[['Date','Confirmed']]
df.head()
```

```
Out[16]:
```

	Date	Confirmed
0	1/22/20	2.0
1	1/22/20	2.0
2	1/22/20	0.0
3	1/22/20	0.0
4	1/22/20	0.0

```
In [17]: df.columns=['ds','y']
```

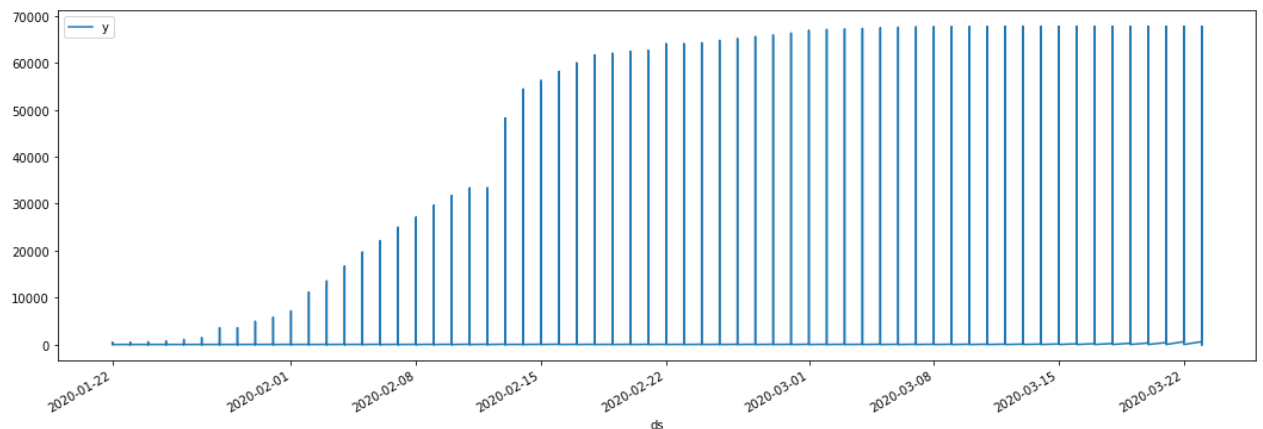
```
In [18]: df['ds']=pd.to_datetime(df['ds'])
df.head()
```

```
Out[18]:
```

	ds	y
0	2020-01-22	2.0
1	2020-01-22	2.0
2	2020-01-22	0.0
3	2020-01-22	0.0
4	2020-01-22	0.0

```
In [19]: df.plot(x='ds',y='y',figsize=(18,6))
```

```
Out[19]: <AxesSubplot: xlabel='ds'>
```



```
In [20]: train=df.iloc[:len(df)-365]
test=df.iloc[len(df)-365:]
```

```
In [21]: from prophet import Prophet

In [22]: from prophet.plot import plot_plotly, plot_components_plotly

In [23]: m=Prophet()
m.fit(train)
future=m.make_future_dataframe(periods=30)
forecast=m.predict(future)

21:18:26 - cmdstanpy - INFO - Chain [1] start processing
21:18:34 - cmdstanpy - INFO - Chain [1] done processing

In [24]: forecast.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	weel
86	2020-04-17	2321.154931	-1714.523810	6279.621946	2138.882812	2505.290117	-0.245291	-0.245291	-0.245291	-0.2452
87	2020-04-18	2372.794793	-1708.455451	6235.879901	2180.735146	2573.780325	3.430022	3.430022	3.430022	3.4300
88	2020-04-19	2424.434655	-1624.357521	6148.926758	2225.069925	2637.692162	27.495413	27.495413	27.495413	27.4954
89	2020-04-20	2476.074518	-1841.455659	6452.061750	2263.962034	2700.103255	-7.663861	-7.663861	-7.663861	-7.6638
90	2020-04-21	2527.714380	-1726.482405	6893.420003	2303.834640	2765.321117	-8.370178	-8.370178	-8.370178	-8.3701

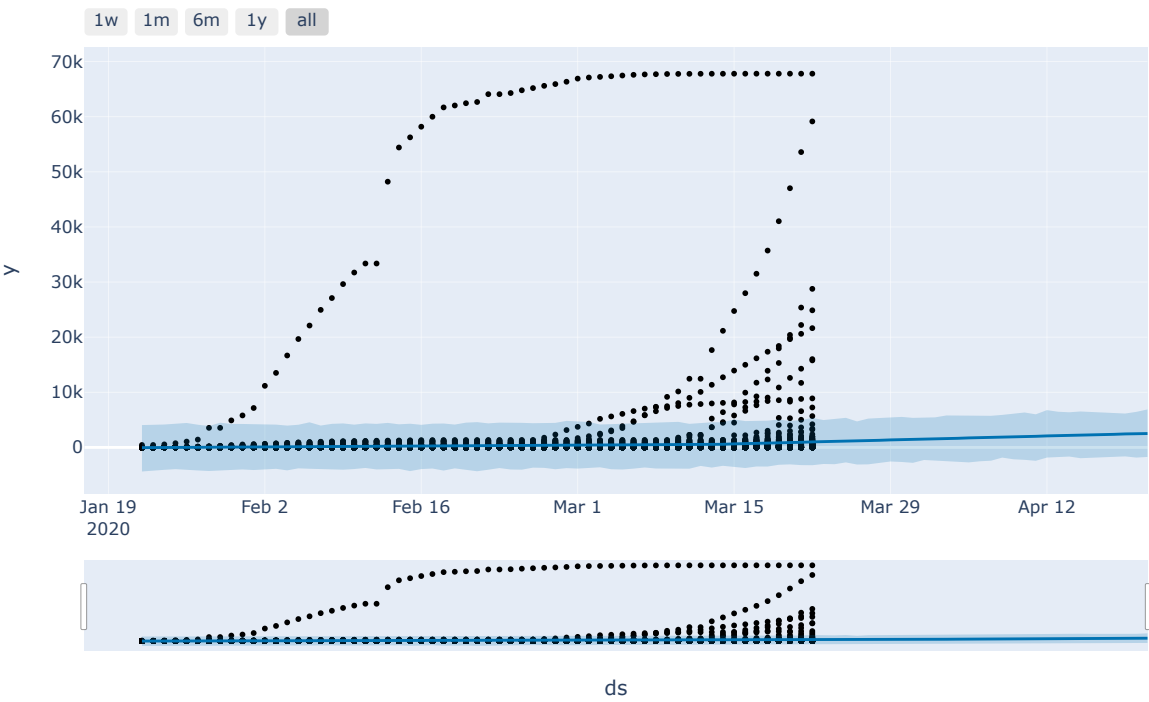
```
In [25]: forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
86	2020-04-17	2320.909640	-1714.523810	6279.621946
87	2020-04-18	2376.224815	-1708.455451	6235.879901
88	2020-04-19	2451.930069	-1624.357521	6148.926758
89	2020-04-20	2468.410656	-1841.455659	6452.061750
90	2020-04-21	2519.344202	-1726.482405	6893.420003

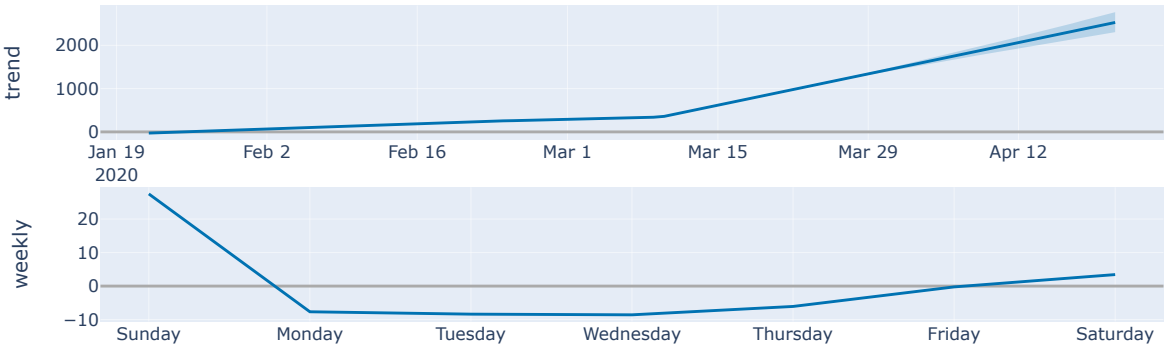
```
In [26]: test.tail()
```

	ds	y
19215	2020-03-23	0.0
19216	2020-03-23	0.0
19217	2020-03-23	0.0
19218	2020-03-23	0.0
19219	2020-03-23	0.0

```
In [27]: plot_plotly(m, forecast)
```



```
In [28]: plot_components_plotly(m,forecast)
```



```
In [ ]:
```