

COVID-19 DATA ANALYSIS

Abstract:

Data Analysis is the technique of collecting, transforming, and organizing data to make future predictions and informed data-driven decisions. It also helps to find possible solutions for a business problem.

In COVID-19,'CO' stands for "corona",'VI' stands for "virus",and 'D' stands for "disease".It was declared as a pandemic or health emergency for world health organization(WHO).

Symptoms and Diagnosis:

- ☒ Fever/chills
- ☒ Cough
- ☒ Difficulty breathing
- ☒ Fatigue
- ☒ Headache
- ☒ Muscle/Body aches
- ☒ Loss of taste or smell
- ☒ Sore throat

Libraries imported in Python :

- ☒ Numpy - Numerical Python
- ☒ SciPy - Scientific Python
- ☒ Pandas - Data Manipulation and Analysis
- ☒ Matplotlib - Plotting and Visualization
- ☒ Scikit-learn-Machine Learning and Data Mining
- ☒ Seaborn-For Statistical Data Visualization

Covid-19 Data Analysis :

Dataset consist of weekly confirmed cases and weekly cumulative confirmed cases for 35 weeks. Then the distribution of the data was examined using the most up-to-date Covid-19 weekly case data and

its parameters were obtained according to the statistical distributions.

DATASET :

Day	Month	year	cases	deaths	Countries And Territories
31	5	2021	366	5	Austria
30	5	2021	570	6	Austria
29	5	2021	538	11	Austria
28	5	2021	639	4	Austria
27	5	2021	405	19	Austria
31	5	2021	2017	14	Belgium
30	5	2021	589	18	Belgium
29	5	2021	1103	16	Belgium
28	5	2021	2004	11	Belgium
27	5	2021	2177	06	Belgium
31	5	2021	186	12	Croatia
30	5	2021	338	10	Croatia

29	5	2021	321	11	Croatia
28	5	2021	357	10	Croatia
27	5	2021	538	21	Croatia
31	5	2021	949	0	Denmark
30	5	2021	1038	4	Denmark
29	5	2021	997	1	Denmark
28	5	2021	1119	0	Denmark
27	5	2021	1073	1	Denmark
31	5	2021	8541	44	France
30	5	2021	10675	68	France
29	5	2021	11268	125	France
28	5	2021	13933	142	France
27	5	2021	12646	144	France

Program:

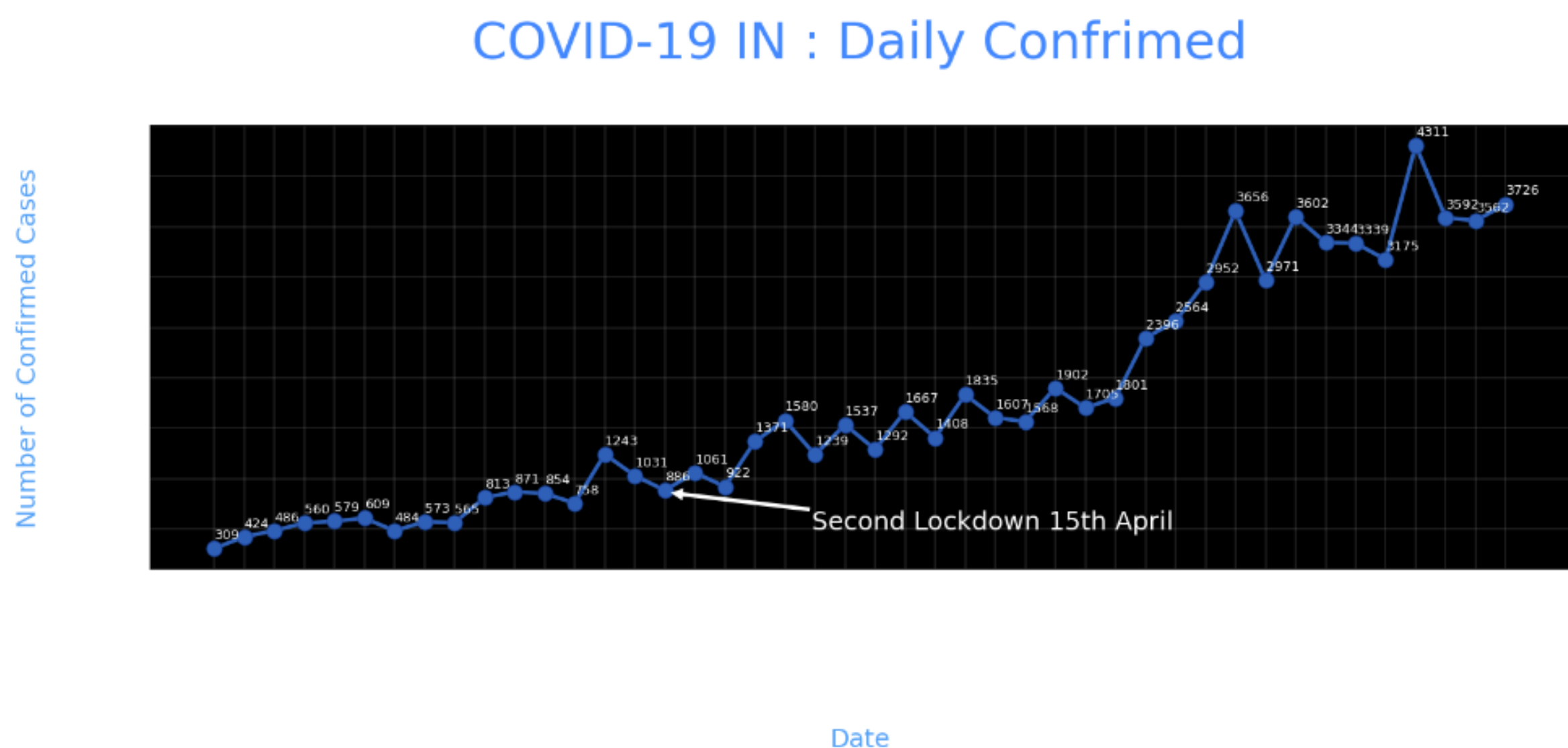
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('case_time_series.csv')
Y = data.iloc[61:,1].values
R = data.iloc[61:,3].values
D = data.iloc[61:,5].values
X = data.iloc[61:,0]
plt.figure (fig size = ( 25, 8 ))
ax = plt.axes()
ax.grid (linewidth=0.4, color='#8f8f8f')
ax.set_facecolor("black")
ax.set_xlabel('\nDate',size=25,color='#4bb4f2')
ax.set_ylabel('Number of Confirmed Cases\n',
size=25,color='#4bb4f2')
plt.xticks(rotation='vertical', size='20', color='white')
plt.yticks(size=20, color='white')
plt.tick_params(size=20,color='white')
for i, j in zip(X, Y):
ax.annotate (str (j), xy = (i, j+100), color='white',
size='13')ax.annotate('Second Lockdown 15th April', xy = (15.2,
860),
xy text = (19.9, 500),
```

```

color = 'white',
size = '25',
arrow props = dict(color='white', linewidth=0.025))
plt.title("COVID-19 IN: Daily Confirmed\n",
size = 50, color = '#28a9ff')
ax.plot(X, Y,
color = '#1F77B4',
marker = 'o',
linewidth = 4,
markersize = 15,
markeredgecolor = '#035E9B')

```

Output:



Now we will check the data for the 10 most affected states with covid-19 in India.

Statewise Data

Program :

```
df_latest = df_latest.sort_values(by=['Confirmed'], ascending = False)
```

```
plt.figure(figsize=(12,4), dpi=80)
```

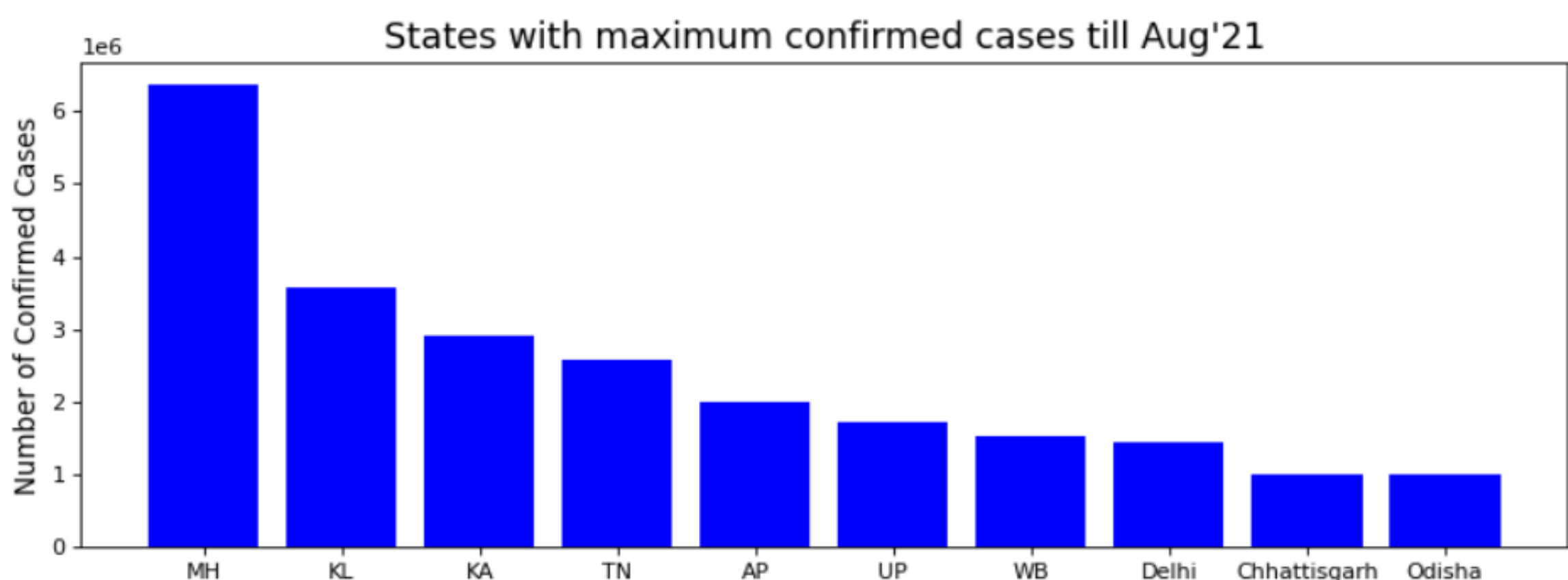
```
plt.bar(df_latest['States'][:10], df_latest['Confirmed'][:10],
```

```
align='center', color='blue')
```

```
plt.ylabel('Number of Confirmed Cases', size = 12)
```

```
plt.title("States with maximum confirmed cases till Aug'21", size = 16)
```

```
plt.show()
```



From the above figure, we can derive that Maharashtra has the maximum cases till Aug'21. On researching further about it, we found that Mumbai and Pune were the majorly affected cities in Maharashtra. Mumbai has a high population density hence an epidemic can easily spread. Pune has a climatic condition favourable for any disease to spread easily.