VARUVAN VADIVELAN INSTITUTE OF TECHNOLOGY

NAAN MUDHALVAN: IBM

PHASE - 4

TECHNOLOGY : DATA ANALYTICS

PROJRCT TITLE : COVID -19 CASES ANALYSIS

Topic:

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project.

Introduction:

Python is a highly powerful general-purpose programming language that can be easily learned and provides data scientists a wide variety of tools and packages. Amid this pandemic period, I decided to analyze this novel coronavirus

Steps involved in Data Analysis:

- Importing required packages
- Gathering Data
- Transforming Data to our needs (Data Wrangling)
- Exploratory Data Analysis (EDA) and Visualization

Importing required Packages:

Importing our required packages is the starting point of all data analysis programming in python. As I've said, python provides a wide variety of packages for data scientists and in this analysis, I used python's most popular data science

packages Pandas and NumPy for Data Wrangling and EDA. When coming to Data Visualization, I used python's interactive packages Plotly and Matplotlib.

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import numpy as np
import plotly
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

Gathering Data :

For a clean and perfect data analysis, the foremost important element is collecting quality Data. For this analysis, I've collected many data from various sources for better accuracy.

Workflow :

- Import libraries
- Load dataset
- Look for the missing values
- Perform Data visualization

Import all libraries :

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

Loading data :

```
data = pd.read_csv
data
```

output :

State/UTs	Total Cases	Active	Discharged	Deaths	Active Ratio	Discharge Ratio	Death Ratio	Population
Andaman and Nicobar	10747	0	10618	129	0.00	98.80	1.20	100896618
Andhra Pradesh	2339078	7	2324338	14733	0.00	99.37	0.63	128500364
Arunachal Pradesh	66891	0	66595	296	0.00	99.56	0.44	658019
Assam	746100	0	738065	8035	0.00	98.92	1.08	290492
Bihar	851404	1	839100	12303	0.00	98.55	1.45	40100376
Chandigarh	99358	3	98174	1181	0.00	98.81	1.19	30501026
Chhattisgarh	1177768	8	1163614	14146	0.00	98.80	1.20	28900667
Dadra and Nagar Haveli and Daman and Diu	11591	0	11587	4	0.00	99.97	0.03	231502578
Delhi	2007313	10	1980781	26522	0.00	98.68	1.32	773997
Goa	259110	15	255082	4013	0.01	98.45	1.55	3772103

Gujarat	1277615	11	1266561	11043	0.00	99.13	0.86	70400153
Haryana	1056655	38	1045903	10714	0.00	98.98	1.01	7503010
Himachal Pradesh	312692	14	308465	4213	0.00	98.65	1.35	3436948
Jammu and Kashmir	479444	10	474649	4785	0.00	99.00	1.00	66001
Jharkhand	442574	0	437243	5331	0.00	98.80	1.20	124904071
Karnataka	4072536	123	4032105	40308	0.00	99.01	0.99	1711947
Kerala	6829249	1300	6756379	71570	0.02	98.93	1.05	91702478
Ladakh	29417	1	29185	231	0.00	99.21	0.79	4184959
Lakshadweep	11415	0	11363	52	0.00	99.54	0.46	11700099
Madhya Pradesh	1054934	2	1044155	10777	0.00	98.98	1.02	14999397
Maharashtra	8136945	134	7988392	148419	0.00	98.17	1.82	399001
Manipur	139924	0	137775	2149	0.00	98.46	1.54	47099270
Meghalaya	96786	1	95161	1624	0.00	98.32	1.68	79502477
Mizoram	238964	0	238238	726	0.00	99.70	0.30	1308967
Nagaland	35986	0	35204	782	0.00	97.83	2.17	38157311

Odisha	1336595	84	1327306	9205	0.01	99.31	0.69	19301096
Puducherry	175636	73	173588	1975	0.04	98.83	1.12	2073074
Punjab	784282	29	764964	19289	0.00	97.54	2.46	34698876
Rajasthan	1315564	5	1305906	9653	0.00	99.27	0.73	1521992
Sikkim	44321	2	43820	499	0.00	98.87	1.13	83697770
Tamil Nadu	3594573	58	3556466	38049	0.00	98.94	1.06	35998752
Telengana	841453	27	837315	4111	0.00	99.51	0.49	69599762
Tripura	108034	0	107094	940	0.00	99.13	0.87	1646050
Uttar Pradesh	2128154	18	2104502	23634	0.00	98.89	1.11	1158040
Uttarakhand	449429	11	441665	7753	0.00	98.27	1.73	85002417
West Bengal	2118696	50	2097114	21532	0.00	98.98	1.02	32199722

Shape of Data :

```
data.shape # for show number of rows and columns
```

output :

```
(36, 9)
```

Top 5 rows :

linkcode

```
data.head() # top 5 rows
```

output :

State/UTs	Total Cases	Active	Discharged	Deaths	Active Ratio	Discharge Ratio	Death Ratio	Population
Andaman and Nicobar	10747	0	10618	129	0.0	98.80	1.20	100896618
Andhra Pradesh	2339078	7	2324338	14733	0.0	99.37	0.63	128500364
Arunachal Pradesh	66891	0	66595	296	0.0	99.56	0.44	658019
Assam	746100	0	738065	8035	0.0	98.92	1.08	290492
Bihar	851404	1	839100	12303	0.0	98.55	1.45	40100376

```
data.tail() # lasr 5 rows
```

output :

state/UTs	Total Cases	Active	Discharged	Deaths	Active Ratio	Discharge Ratio	Death Ratio	Population
Telengana	841453	27	837315	4111	0.0	99.51	0.49	69599762

Tripura	108034	0	107094	940	0.0	99.13	0.87	1646050
Uttar Pradesh	2128154	18	2104502	23634	0.0	98.89	1.11	1158040
Uttarakhand	449429	11	441665	7753	0.0	98.27	1.73	85002417
West Bengal	2118696	50	2097114	21532	0.0	98.98	1.02	32199722

Looking for summary :

data.columns

output :

Index(['State/UTs', 'Total Cases', 'Active', 'Discharged', 'Deaths 'Active Rat
io', 'Discharge Ratio', 'Death Ratio', 'Population'],
dtype='object')

Looking for missing values :

In [11]:

linkcode

data.isnull() #for cheking null values

output :

state/UTs	Total Cases	Active	Discharged	Deaths	Active Ratio	Discharge Ratio	Death Ratio	Population
False	False	False	False	False	False	False	False	False

False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False

```
data.isnull().sum()  # for number of null values
```

output :

```
State/UTs         0
Total Cases       0
Active            0
Discharged        0
Deaths            0
Active Ratio      0
Discharge Rati    0
Death Ratio       0
Population        0
dtype: int64
```

Data visualization :

In [13]:

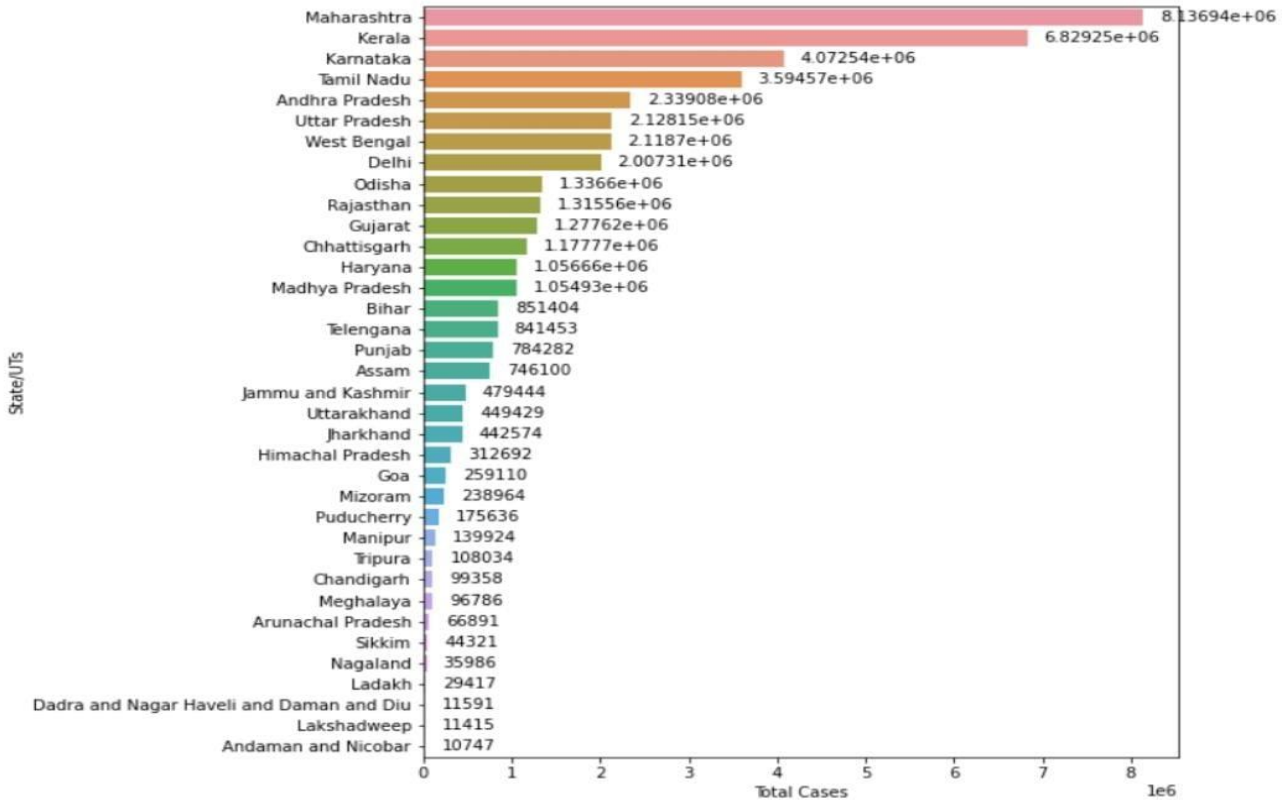
```
Total_cases = data[['State/UTs','Total Cases']].sort_values(by=['Total Cases'],ascending=False).reset_index(drop=True)
```

In [14]:

linkcode

```
# Draw barplot

plt.figure(figsize=(8,10))
ax = sns.barplot(x='Total Cases',y='State/UTs',data=Total_cases)
ax.bar_label(ax.containers[0],padding=10,fmt='%g');
plt.show()
```

output :

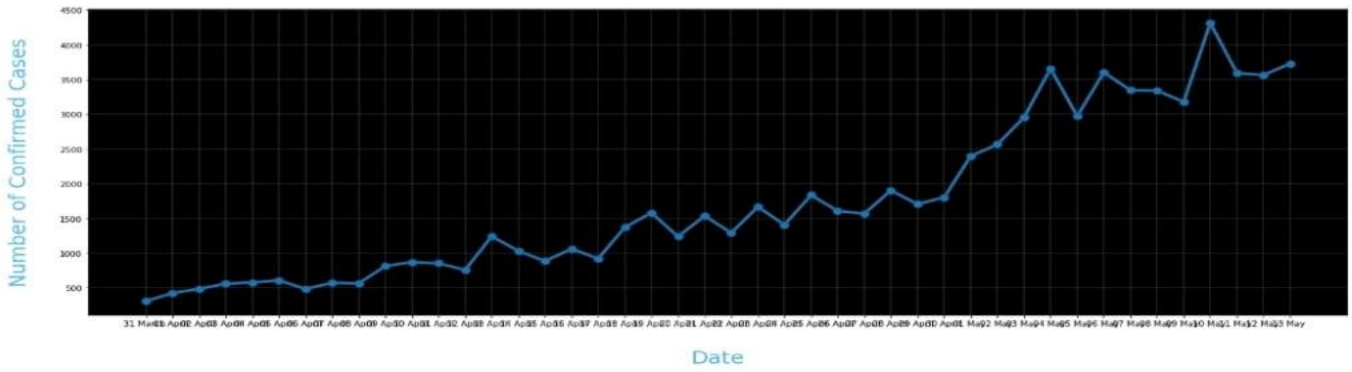


Complete Code:

```
 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(figsize=(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')

ax.plot(X,Y,
        color='#1F77B4',
        marker='o',
        linewidth=4,
        markersize=15,
        markeredgecolor='#035E9B')
```

Output :



Conclusion :

The COVID-19 pandemic has led to questions about many aspects in India—the quality of health care, the response of governments and institutions, and issues related to law and order.