

# Corona Virus Pandemic - In India

## An Exploratory Data Visualization and Analysis

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus.

Most people infected with the virus will experience mild to moderate respiratory illness and recover without requiring special treatment. However, some will become seriously ill and require medical attention. Older people and those with underlying medical conditions like cardiovascular disease, diabetes, chronic respiratory disease, or cancer are more likely to develop serious illness. Anyone can get sick with COVID-19 and become seriously ill or die at any age.

In this notebook, We will take a look at the current situation in India. We will take a look at the regions which are most hampered by the outbreak and how numbers have steadily climbed in the country.

## PROGRAMMING LANGUAGE AND MODULES INCLUDED IN THIS PROJECT :

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
%matplotlib notebook
```

## Data Sets Included in this Project :

```
In [39]: covid19_df = pd.read_csv(r"C:\Users\HARSHIT\Downloads\COVID19-EDA-INDIA-master\COVID19-ED
individuals_df = pd.read_csv(r"C:\Users\HARSHIT\Downloads\COVID19-EDA-INDIA-master\COVID
excel_file = pd.ExcelFile(r"C:\Users\HARSHIT\Downloads\COVID19-EDA-INDIA-master\COVID19-
covid_df = pd.ExcelFile(r"C:\Users\HARSHIT\Downloads\COVID19-EDA-INDIA-master\COVID19-ED
dbd_India=pd.read_excel(r"C:\Users\HARSHIT\Downloads\COVID19-EDA-INDIA-master\COVID19-ED
vaccine_df= pd.read_csv(r"C:\Users\HARSHIT\Downloads\COVID19-EDA-INDIA-master\COVID19-ED
indian_states_df = excel_file.parse('Sheet1')
```

```
In [ ]:
```

```
In [7]: covid19_df.head()
```

```
Out[7]:
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths
0	1	30/01/20	6:00 PM	Kerala	1	0	0	0
1	2	31/01/20	6:00 PM	Kerala	1	0	0	0
2	3	01/02/20	6:00 PM	Kerala	2	0	0	0

<b>3</b>	4	02/02/20	6:00 PM	Kerala	3	0	0	0
<b>4</b>	5	03/02/20	6:00 PM	Kerala	3	0	0	0

In [ ]:

In [8]: `covid19_df.tail()`

Out[8]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	De
<b>3634</b>	3635	01/07/20	8:00 AM	Tripura	-	-	1086	
<b>3635</b>	3636	01/07/20	8:00 AM	Uttarakhand	-	-	2231	
<b>3636</b>	3637	01/07/20	8:00 AM	Uttar Pradesh	-	-	16084	
<b>3637</b>	3638	01/07/20	8:00 AM	West Bengal	-	-	12130	
<b>3638</b>	3639	01/07/20	8:00 AM	Cases being reassigned to states	-	-	0	

In [ ]:

In [9]: `covid19_df.shape`

Out[9]: (3639, 9)

In [ ]:

In [10]: `covid19_df.isna().sum()`

Out[10]:

```

Sno          0
Date          0
Time          0
State/UnionTerritory  0
ConfirmedIndianNational  0
ConfirmedForeignNational  0
Cured         0
Deaths        0
Confirmed     0
dtype: int64

```

Here, we see that there are no missing values in this dataset which makes my job more easier. Let us now have a look at the most recent records for each state to gain an idea about where we stand currently.

In [ ]:

In [11]: `covid19_df_latest = covid19_df[covid19_df['Date']=="01/07/20"]`  
`covid19_df_latest.head()`

Out[11]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	De
<b>3603</b>	3604	01/07/20	8:00 AM	Andaman and Nicobar Islands	-	-	50	
<b>3604</b>	3605	01/07/20	8:00	Andhra Pradesh	-	-	6511	

			AM				
<b>3605</b>	3606	01/07/20	8:00 AM	Arunachal Pradesh	-	-	62
<b>3606</b>	3607	01/07/20	8:00 AM	Assam	-	-	5647
<b>3607</b>	3608	01/07/20	8:00 AM	Bihar	-	-	7687

In [ ]:

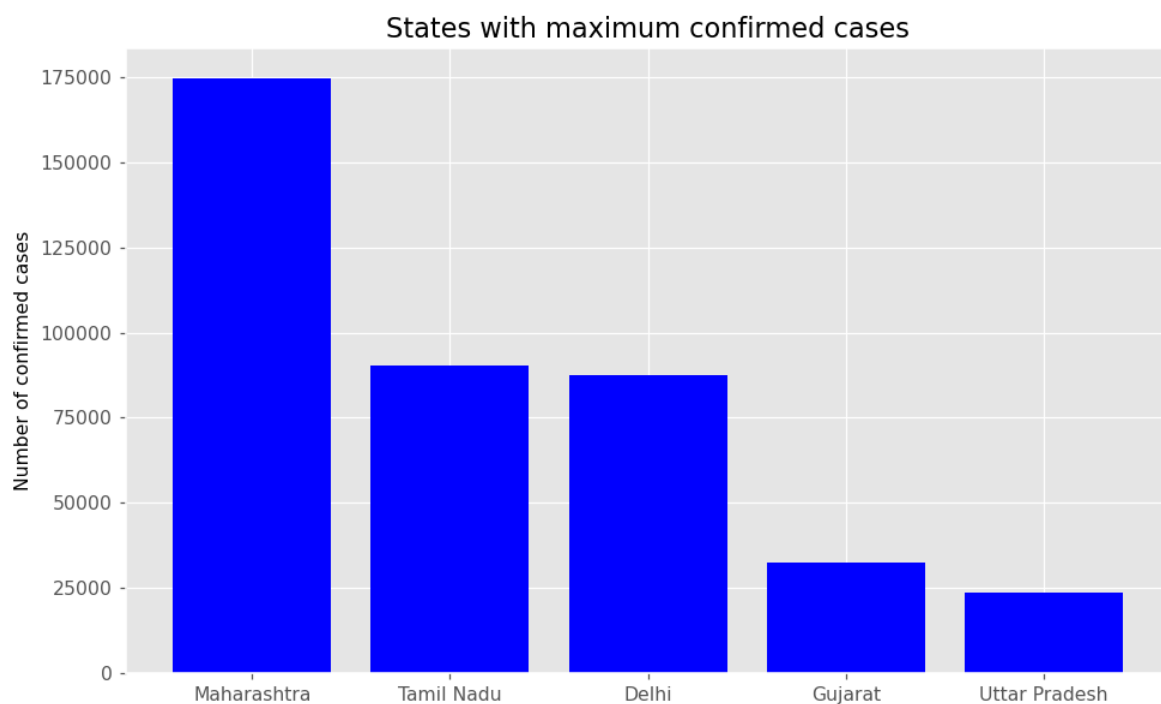
In [12]: `covid19_df_latest['Confirmed'].sum()`

Out[12]: 585493

In [ ]:

## STATEWISE FIGURES

```
In [21]: covid19_df_latest = covid19_df_latest.sort_values(by=['Confirmed'], ascending = False)
plt.figure(figsize=(10,6), dpi = 90)
plt.bar(covid19_df_latest['State/UnionTerritory'][:5], covid19_df_latest['Confirmed'][:5])
plt.ylabel('Number of confirmed cases')
plt.title('States with maximum confirmed cases')
plt.show()
```



On inspecting the above visualization, we see that Maharashtra has the most number of inspected cases as of now. The situation in Maharashtra is so grave that no other state in India has crossed even half that mark as per the data we have. Tamil Nadu and Delhi are about to touch the 100000 mark whereas Gujarat has around 32000 cases and Uttar Pradesh over 22000+ cases.

In [ ]:

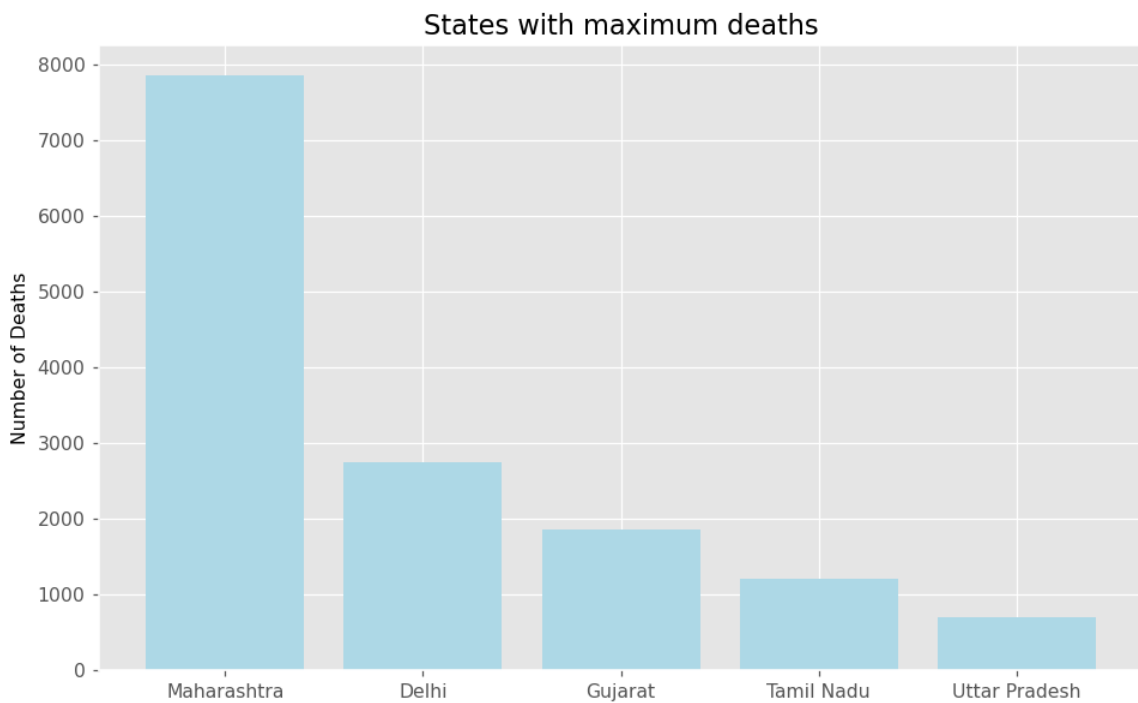
```
covid19_df_latest['Deaths'].sum()
```

Out[22]: 17400

As per the data in the dataset, India has had 17400 deaths across all states. We will now see which states have the most deaths.

In [ ]:

```
covid19_df_latest = covid19_df_latest.sort_values(by=['Deaths'], ascending = False)
plt.figure(figsize=(10,6), dpi=90)
plt.bar(covid19_df_latest['State/UnionTerritory'][:5], covid19_df_latest['Deaths'][:5],
plt.ylabel('Number of Deaths')
plt.title('States with maximum deaths')
plt.show()
```



Next up, I wanted to look at the number of deaths per confirmed cases in different Indian states to gain a better idea about the healthcare facilities available.

In [ ]:

```
covid19_df_latest['Deaths/Confirmed Cases'] = (covid19_df_latest['Confirmed']/covid19_df_latest['Deaths'])
covid19_df_latest['Deaths/Confirmed Cases'] = [np.nan if x==float("inf") else x for x in covid19_df_latest['Deaths/Confirmed Cases']]
covid19_df_latest = covid19_df_latest.sort_values(by=['Deaths/Confirmed Cases'], ascending = False)
covid19_df_latest.iloc[:10]
```

Out[24]:

Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	De
3613	3614	01/07/20	8:00 AM	Gujarat	-	-	23662
3622	3623	01/07/20	8:00	Maharashtra	-	-	90911

			AM				
<b>3621</b>	3622	01/07/20	8:00 AM	Madhya Pradesh	-	-	10395
<b>3637</b>	3638	01/07/20	8:00 AM	West Bengal	-	-	12130
<b>3611</b>	3612	01/07/20	8:00 AM	Delhi	-	-	58348
<b>3636</b>	3637	01/07/20	8:00 AM	Uttar Pradesh	-	-	16084
<b>3629</b>	3630	01/07/20	8:00 AM	Punjab	-	-	3867
<b>3630</b>	3631	01/07/20	8:00 AM	Rajasthan	-	-	14220
<b>3624</b>	3625	01/07/20	8:00 AM	Meghalaya	-	-	42
<b>3628</b>	3629	01/07/20	8:00 AM	Puducherry	-	-	272

So after creating this new measure and sorting the states based on this figure, I look at the ten worst states in this regard. We see that there are some states like Meghalaya, Puducherry, Punjab and Rajasthan where the number of cases and deaths are pretty low as of now and it appears things are in control. But other states like Gujarat, Maharashtra, Madhya Pradesh look well hit by the condition.

In [ ]:

## CASES PER 10 MILLION

In [ ]:

In [25]: `indian_states_df.head()`

Out[25]:

	State	Aadhaar assigned as of 2019	Area (per sq km)
<b>0</b>	Delhi	21763471	1483
<b>1</b>	Haryana	28941133	44212
<b>2</b>	Kerala	36475649	38852
<b>3</b>	Himachal Pradesh	7560770	55673
<b>4</b>	Punjab	30355185	50362

We will rename the number of Aadhaar cards assigned column as Population and discard the Area feature since We decided against using it due to recent updates in States and UTs in India.

In [ ]:

In [26]: `indian_states_df = indian_states_df[['State', 'Aadhaar assigned as of 2019']]  
indian_states_df.columns = ['State/UnionTerritory', 'Population']  
indian_states_df.head()`

Out[26]:

	State/UnionTerritory	Population
--	----------------------	------------

0	Delhi	21763471
1	Haryana	28941133
2	Kerala	36475649
3	Himachal Pradesh	7560770
4	Punjab	30355185

We will now merge the Population dataset with our main dataset and create a new feature called Cases/10 Million to gain some more idea on really which cases are more hit by the COVID-19 crisis. We feel this new measure is now a more level headed measure as it takes care of the population differences which exists between different states.

In [ ]:

```
covid19_df_latest.fillna(0, inplace=True)
covid19_df_latest.sort_values(by='Cases/10million', ascending=False)
```

Out[28]:

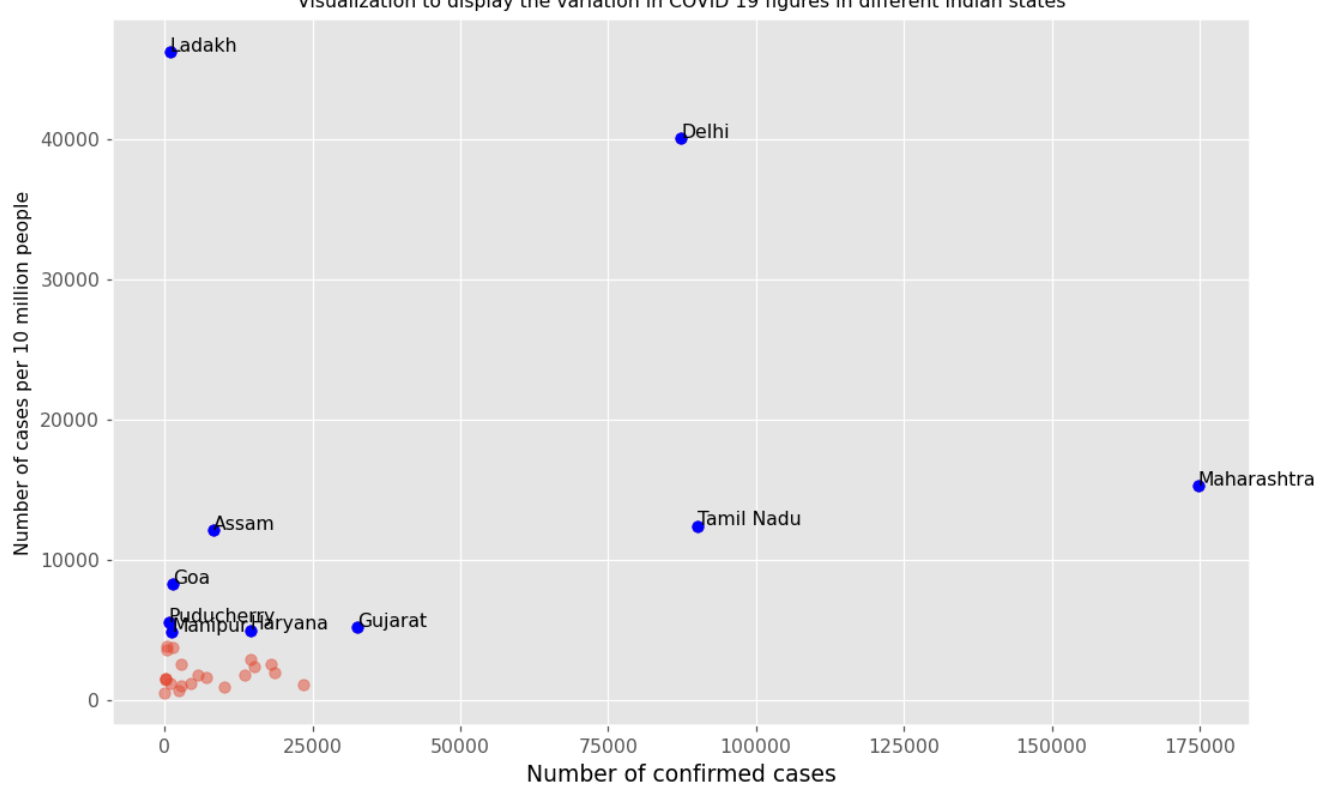
	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deat
25	3621	01/07/20	8:00 AM	Ladakh	-	-	648	
4	3612	01/07/20	8:00 AM	Delhi	-	-	58348	27
1	3623	01/07/20	8:00 AM	Maharashtra	-	-	90911	78
14	3633	01/07/20	8:00 AM	Tamil Nadu	-	-	50074	12
24	3607	01/07/20	8:00 AM	Assam	-	-	5647	
23	3613	01/07/20	8:00 AM	Goa	-	-	596	
9	3629	01/07/20	8:00 AM	Puducherry	-	-	272	
0	3614	01/07/20	8:00 AM	Gujarat	-	-	23662	18
10	3615	01/07/20	8:00 AM	Haryana	-	-	9972	2
28	3624	01/07/20	8:00 AM	Manipur	-	-	553	
13	3609	01/07/20	8:00 AM	Chandigarh	-	-	364	
26	3635	01/07/20	8:00 AM	Tripura	-	-	1086	
27	3627	01/07/20	8:00 AM	Nagaland	-	-	168	
15	3605	01/07/20	8:00 AM	Andhra Pradesh	-	-	6511	1
7	3631	01/07/20	8:00 AM	Rajasthan	-	-	14220	4

12	3636	01/07/20	8:00 AM	Uttarakhand	-	-	2231	
11	3619	01/07/20	8:00 AM	Karnataka	-	-	7918	2
3	3638	01/07/20	8:00 AM	West Bengal	-	-	12130	6
6	3630	01/07/20	8:00 AM	Punjab	-	-	3867	1
2	3622	01/07/20	8:00 AM	Madhya Pradesh	-	-	10395	5
22	3628	01/07/20	8:00 AM	Odisha	-	-	5189	
20	3606	01/07/20	8:00 AM	Arunachal Pradesh	-	-	62	
30	3632	01/07/20	8:00 AM	Sikkim	-	-	52	
29	3626	01/07/20	8:00 AM	Mizoram	-	-	122	
16	3616	01/07/20	8:00 AM	Himachal Pradesh	-	-	580	
19	3620	01/07/20	8:00 AM	Kerala	-	-	2306	
5	3637	01/07/20	8:00 AM	Uttar Pradesh	-	-	16084	6
21	3610	01/07/20	8:00 AM	Chhattisgarh	-	-	2250	
17	3608	01/07/20	8:00 AM	Bihar	-	-	7687	
18	3618	01/07/20	8:00 AM	Jharkhand	-	-	1884	
8	3625	01/07/20	8:00 AM	Meghalaya	-	-	42	

```
In [31]: df = covid19_df_latest[(covid19_df_latest['Confirmed']>=30000) | (covid19_df_latest['Case
plt.figure(figsize=(10,6),dpi=90)
plt.scatter(covid19_df_latest['Confirmed'],covid19_df_latest['Cases/10million'],alpha=0.
plt.xlabel('Number of confirmed cases',size=12)
plt.ylabel('Number of cases per 10 million people',size=10)
plt.scatter(df['Confirmed'],df['Cases/10million'],color='blue')

for i in range(df.shape[0]):
    plt.annotate(df['State/UnionTerritory'].tolist()[i], xy=(df['Confirmed'].tolist()[i]
    xytext = (df['Confirmed'].tolist()[i]+1.0, df['Cases/10million'].tolist()[i]+12.0),s

plt.tight_layout()
plt.title('Visualization to display the variation in COVID 19 figures in different India
plt.show()
```



In [ ]:

## INDIVIDUAL DATA

In [ ]:

In [33]: `individuals_df.iloc[0]`

Out[33]:

id	0
government_id	KL-TS-P1
diagnosed_date	30/01/2020
age	20
gender	F
detected_city	Thrissur
detected_district	Thrissur
detected_state	Kerala
nationality	India
current_status	Recovered
status_change_date	14/02/2020
notes	Travelled from Wuhan

Name: 0, dtype: object

In [ ]:

The first case in India due to COVID-19 was noticed on 30th January 2020. It was detected in the city of Thrissur in Kerala. The individual had a travel history in Wuhan.

In [34]:

```
individuals_grouped_district = individuals_df.groupby('detected_district')
individuals_grouped_district = individuals_grouped_district['id']
individuals_grouped_district.columns=['count']
individuals_grouped_district.count().sort_values(ascending=False).head()
```

Out[34]:

detected_district	
Mumbai	3149
Ahmedabad	2181
Indore	1176

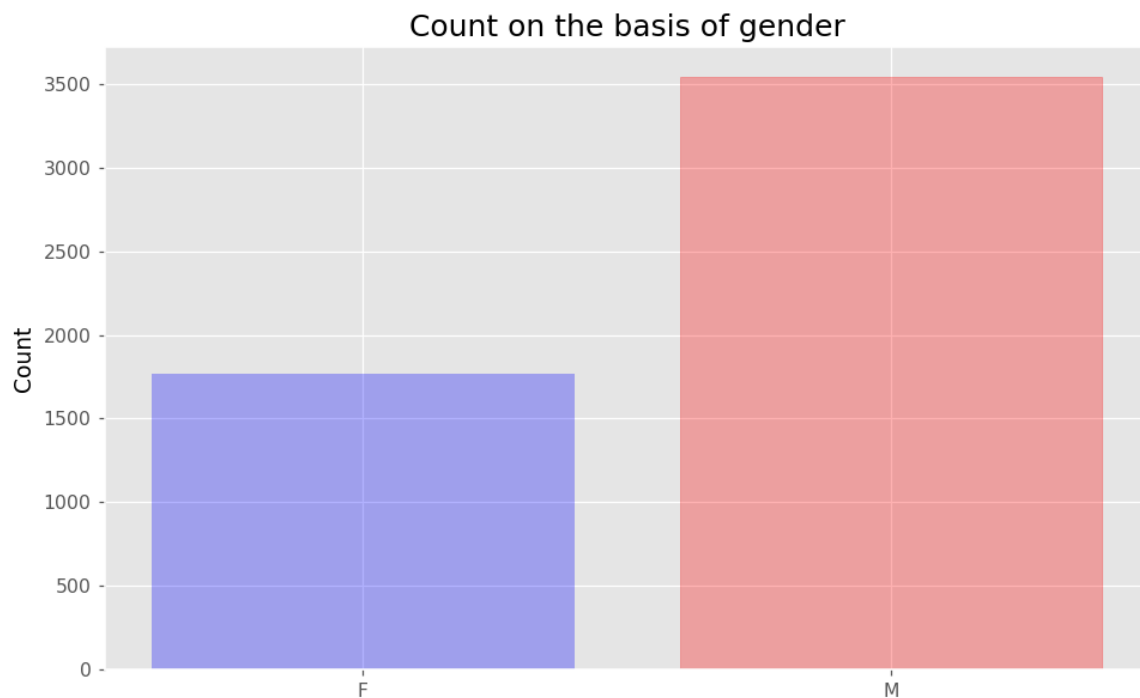


Jaipur 808  
Pune 706  
Name: id, dtype: int64

In [ ]:

```
In [44]: individuals_grouped_gender = individuals_df.groupby('gender')
individuals_grouped_gender = pd.DataFrame(individuals_grouped_gender.size().reset_index()
individuals_grouped_gender.head()

plt.figure(figsize=(10,6),dpi=90)
barlist= plt.bar(individuals_grouped_gender['gender'],individuals_grouped_gender['count']
barlist[1].set_color('r')
plt.ylabel('Count',size=12)
plt.title('Count on the basis of gender',size=16)
plt.show()
```



In [ ]:

From the data, it seems that the virus is affecting males more than females in India.

In [ ]:

## GRAPH TO SHOW COMPARISON BETWEEN 5 AFFECTED STATES

In [ ]:

```
In [52]: covid19_maharashtra = covid19_df[covid19_df['State/UnionTerritory']=="Maharashtra"]
covid19_maharashtra.head()
covid19_maharashtra.reset_index(inplace=True)
covid19_maharashtra= covid19_maharashtra.drop(['index', 'Sno', 'Time', 'ConfirmedIndianN
covid19_maharashtra.reset_index(inplace = True)
covid19_maharashtra.columns = ['Day Count', 'Date', 'State/UnionTerritory', 'Deaths', 'C
```

```

covid19_maharashtra['Day Count'] = covid19_maharashtra['Day Count'] + 8
missing_values = pd.DataFrame({"Day Count": [x for x in range(1,8)],
                                "Date": ["0" + str(x) + "/06/20" for x in range(2,9)],
                                "State/UnionTerritory": ["Maharashtra"]*7,
                                "Deaths": [0]*7,
                                "Confirmed": [0]*7})

covid19_maharashtra = covid19_maharashtra.append(missing_values, ignore_index=True)
covid19_maharashtra = covid19_maharashtra.sort_values(by="Day Count", ascending = True)

covid19_maharashtra.reset_index(drop=True, inplace=True)
print(covid19_maharashtra.shape)
covid19_maharashtra.head()

```

(122, 5)

C:\Users\HARSHIT\AppData\Local\Temp\ipykernel\_12332\928352712.py:13: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

Out[52]:

	Day Count	Date	State/UnionTerritory	Deaths	Confirmed
0	1	02/06/20	Maharashtra	0	0
1	2	03/06/20	Maharashtra	0	0
2	3	04/06/20	Maharashtra	0	0
3	4	05/06/20	Maharashtra	0	0
4	5	06/06/20	Maharashtra	0	0

In [ ]:

In [53]:

```

covid19_kerala = covid19_df[covid19_df['State/UnionTerritory'] == "Kerala"]
covid19_kerala = covid19_kerala.iloc[32:]
covid19_kerala.reset_index(inplace = True)
covid19_kerala = covid19_kerala.drop(['index', 'Sno', 'Time', 'ConfirmedIndianNational',
covid19_kerala.reset_index(inplace = True)
covid19_kerala.columns = ['Day Count', 'Date', 'State/UnionTerritory', 'Deaths', 'Confirmed']
covid19_kerala['Day Count'] = covid19_kerala['Day Count'] + 1
print(covid19_kerala.shape)
covid19_kerala.head()

```

(122, 5)

Out[53]:

	Day Count	Date	State/UnionTerritory	Deaths	Confirmed
0	1	02/03/20	Kerala	0	3
1	2	03/03/20	Kerala	0	3
2	3	04/03/20	Kerala	0	3
3	4	05/03/20	Kerala	0	3
4	5	06/03/20	Kerala	0	3

In [ ]:

In [54]:

```

covid19_delhi = covid19_df[covid19_df['State/UnionTerritory'] == "Delhi"]
covid19_delhi.reset_index(inplace = True)
covid19_delhi = covid19_delhi.drop(['index', 'Sno', 'Time', 'ConfirmedIndianNational', 'ConfirmedForeignNational'])
covid19_delhi.reset_index(inplace = True)
covid19_delhi.columns = ['Day Count', 'Date', 'State/UnionTerritory', 'Deaths', 'Confirmed']
covid19_delhi['Day Count'] = covid19_delhi['Day Count'] + 1

```

```
print(covid19_delhi.shape)
covid19_delhi.head()
```

```
(122, 5)
```

```
Out[54]:
```

	Day Count	Date	State/UnionTerritory	Deaths	Confirmed
0	1	02/03/20	Delhi	0	1
1	2	03/03/20	Delhi	0	1
2	3	04/03/20	Delhi	0	1
3	4	05/03/20	Delhi	0	2
4	5	06/03/20	Delhi	0	3

```
In [ ]:
```

```
In [55]: covid19_gujarat = covid19_df[covid19_df['State/UnionTerritory'] == "Gujarat"]
covid19_gujarat.reset_index(inplace = True)
covid19_gujarat = covid19_gujarat.drop(['index','Sno', 'Time', 'ConfirmedIndianNational'])
covid19_gujarat.reset_index(inplace = True)
covid19_gujarat.columns = ['Day Count', 'Date', 'State/UnionTerritory', 'Deaths', 'Confirmed']
covid19_gujarat['Day Count'] = covid19_gujarat['Day Count'] + 19
missing_values = pd.DataFrame({"Day Count": [x for x in range(1,19)],
                                "Date": [("0" + str(x) if x < 10 else str(x)) + "/03/20" for x
                                in range(1,19)],
                                "State/UnionTerritory": ["Gujarat"]*18,
                                "Deaths": [0]*18,
                                "Confirmed": [0]*18})
covid19_gujarat = covid19_gujarat.append(missing_values, ignore_index = True)
covid19_gujarat = covid19_gujarat.sort_values(by="Day Count", ascending = True)
covid19_gujarat.reset_index(drop=True, inplace=True)
print(covid19_gujarat.shape)
covid19_gujarat.head()
```

```
(122, 5)
```

```
C:\Users\HARSHIT\AppData\Local\Temp\ipykernel_12332\3533786358.py:12: FutureWarning:
```

```
The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
```

```
Out[55]:
```

	Day Count	Date	State/UnionTerritory	Deaths	Confirmed
0	1	02/03/20	Gujarat	0	0
1	2	03/03/20	Gujarat	0	0
2	3	04/03/20	Gujarat	0	0
3	4	05/03/20	Gujarat	0	0
4	5	06/03/20	Gujarat	0	0

```
In [ ]:
```

```
In [56]: covid19_tamilnadu = covid19_df[covid19_df['State/UnionTerritory'] == "Tamil Nadu"]
covid19_tamilnadu.reset_index(inplace = True)
covid19_tamilnadu = covid19_tamilnadu.drop(['index','Sno', 'Time', 'ConfirmedIndianNational'])
covid19_tamilnadu.reset_index(inplace = True)
covid19_tamilnadu.columns = ['Day Count', 'Date', 'State/UnionTerritory', 'Deaths', 'Confirmed']
covid19_tamilnadu['Day Count'] = covid19_delhi['Day Count'] + 1
print(covid19_tamilnadu.shape)
covid19_tamilnadu.head()
```

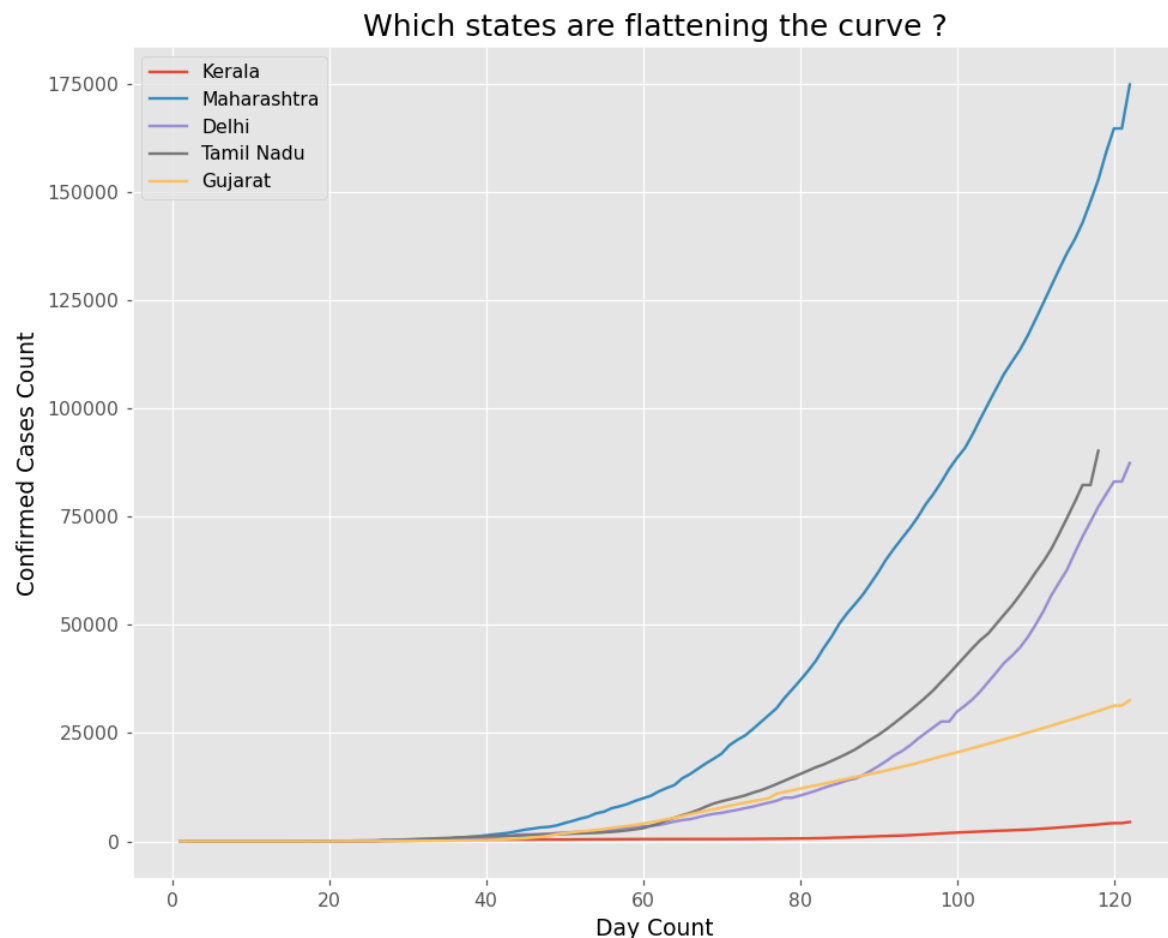
```
(117, 5)
```

Out[56]:

	Day Count	Date	State/UnionTerritory	Deaths	Confirmed
0	2	07/03/20	Tamil Nadu	0	1
1	3	08/03/20	Tamil Nadu	0	1
2	4	09/03/20	Tamil Nadu	0	1
3	5	10/03/20	Tamil Nadu	0	1
4	6	11/03/20	Tamil Nadu	0	1

In [ ]:

```
In [58]: plt.figure(figsize=(10,8), dpi=90)
plt.plot(covid19_kerala['Day Count'], covid19_kerala['Confirmed'])
plt.plot(covid19_maharashtra['Day Count'], covid19_maharashtra['Confirmed'])
plt.plot(covid19_delhi['Day Count'], covid19_delhi['Confirmed'])
plt.plot(covid19_tamilnadu['Day Count'], covid19_tamilnadu['Confirmed'])
plt.plot(covid19_gujarat['Day Count'], covid19_gujarat['Confirmed'])
plt.legend(['Kerala', 'Maharashtra', 'Delhi', 'Tamil Nadu', 'Gujarat'], loc='upper left')
plt.xlabel('Day Count', size=12)
plt.ylabel('Confirmed Cases Count', size=12)
plt.title('Which states are flattening the curve ?', size = 16)
plt.show()
```



In [ ]:

We see almost all the curves follow the curve which is displayed by the nation as a whole. The only anomaly

is that of Kerala. Kerala's curve saw the gradual incline in the period between 20-30 days as seen in other curves. But what Kerala managed to do was it did not let the curve incline further and manage to flatten the curve. As a result, the state has been able to contain the situation.

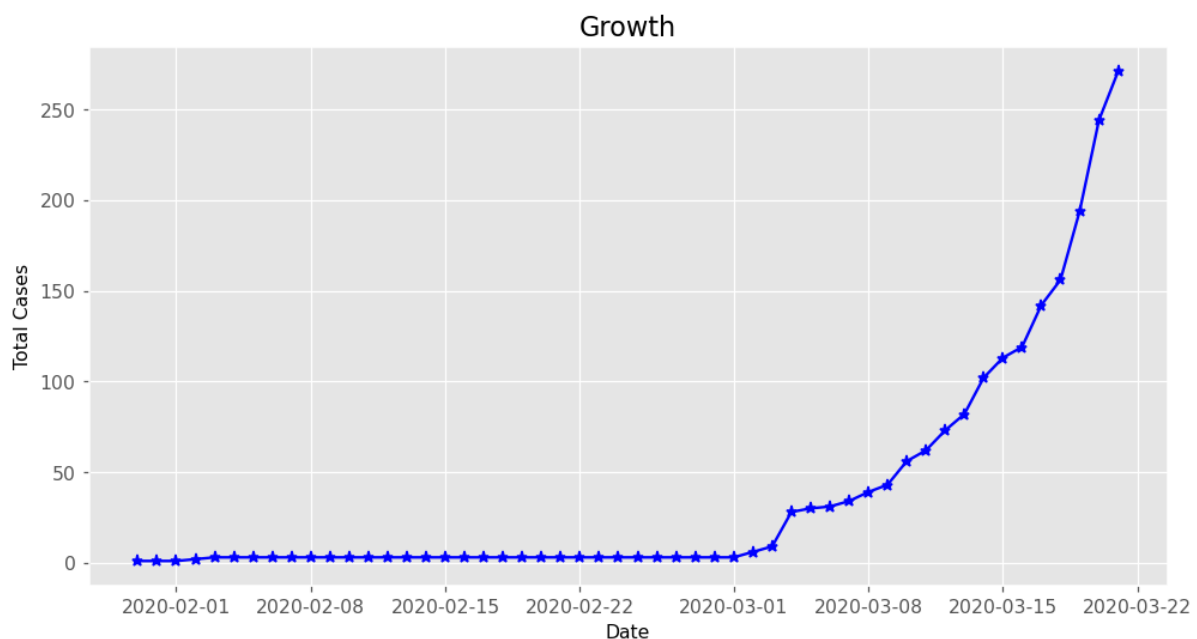
The situation in Maharashtra looks very grave indeed. The curve has had an immense steep incline and shows no signs of slowing down. Gujarat's curve steeped at a later time interval compared to the rest.

In [ ]:

## Growth of Virus in India

In [ ]:

```
In [61]: fig=plt.figure(figsize=(10,5),dpi=90)
axes=fig.add_axes([0.1,0.1,0.8,0.8])
axes.plot(dbd_India["Date"],dbd_India["Total Cases"],color='blue',marker='*')
axes.set_xlabel("Date")
axes.set_ylabel("Total Cases")
axes.set_title("Growth")
plt.show()
```



In [ ]:

## VACCINATION

In [ ]:

```
In [20]: vaccine_df.head()
```

Out[20]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN

2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN

5 rows × 24 columns

In [ ]:

In [21]: `vaccine_df.rename(columns = {'Updated On' : 'Vaccine_Date'}, inplace = True)`

In [ ]:

In [22]: `vaccine_df.head(10)`

Out[22]:

	Vaccine_Date	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN
5	21/01/2021	India	365965.0	32226.0	12600.0	365965.0	0.0	NaN	NaN
6	22/01/2021	India	549381.0	36988.0	14115.0	549381.0	0.0	NaN	NaN
7	23/01/2021	India	759008.0	43076.0	15605.0	759008.0	0.0	NaN	NaN
8	24/01/2021	India	835058.0	49851.0	18111.0	835058.0	0.0	NaN	NaN
9	25/01/2021	India	1277104.0	55151.0	19682.0	1277104.0	0.0	NaN	NaN

10 rows × 24 columns

In [ ]:

In [23]: `vaccine_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7845 entries, 0 to 7844
Data columns (total 24 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Vaccine_Date                             7845 non-null   object
1   State                                    7845 non-null   object
2   Total Doses Administered                 7621 non-null   float64
3   Sessions                                7621 non-null   float64
4   Sites                                   7621 non-null   float64
5   First Dose Administered                  7621 non-null   float64
6   Second Dose Administered                 7621 non-null   float64
7   Male (Doses Administered)                7461 non-null   float64
8   Female (Doses Administered)              7461 non-null   float64
9   Transgender (Doses Administered)          7461 non-null   float64
10  Covaxin (Doses Administered)              7621 non-null   float64
11  CoviShield (Doses Administered)            7621 non-null   float64
```

```

12  Sputnik V (Doses Administered)      2995 non-null    float64
13  AEFI                                5438 non-null    float64
14  18-44 Years (Doses Administered)    1702 non-null    float64
15  45-60 Years (Doses Administered)    1702 non-null    float64
16  60+ Years (Doses Administered)      1702 non-null    float64
17  18-44 Years(Individuals Vaccinated) 3733 non-null    float64
18  45-60 Years(Individuals Vaccinated) 3734 non-null    float64
19  60+ Years(Individuals Vaccinated)    3734 non-null    float64
20  Male(Individuals Vaccinated)         160 non-null     float64
21  Female(Individuals Vaccinated)       160 non-null     float64
22  Transgender(Individuals Vaccinated)  160 non-null     float64
23  Total Individuals Vaccinated         5919 non-null    float64
dtypes: float64(22), object(2)
memory usage: 1.4+ MB

```

```
In [24]: vaccine_df.isnull().sum()
```

```

Out[24]: Vaccine_Date      0
State      0
Total Doses Administered    224
Sessions      224
Sites      224
First Dose Administered    224
Second Dose Administered    224
Male (Doses Administered)   384
Female (Doses Administered) 384
Transgender (Doses Administered) 384
Covaxin (Doses Administered) 224
CoviShield (Doses Administered) 224
Sputnik V (Doses Administered) 4850
AEFI      2407
18-44 Years (Doses Administered) 6143
45-60 Years (Doses Administered) 6143
60+ Years (Doses Administered) 6143
18-44 Years(Individuals Vaccinated) 4112
45-60 Years(Individuals Vaccinated) 4111
60+ Years(Individuals Vaccinated) 4111
Male(Individuals Vaccinated) 7685
Female(Individuals Vaccinated) 7685
Transgender(Individuals Vaccinated) 7685
Total Individuals Vaccinated 1926
dtype: int64

```

```
In [ ]:
```

```
In [25]: vaccination = vaccine_df.drop(columns = ['Sputnik V (Doses Administered)', 'AEFI', '18-4
```

```
In [ ]:
```

```
In [26]: vaccination.head()
```

```

Out[26]:

```

	Vaccine_Date	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Fema (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	Na
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	Na
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	Na
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	Na
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	Na

In [ ]:

```
In [27]: # Remove rows where state = India

vaccine = vaccine_df[vaccine_df.State!='India']
vaccine
```

Out[27]:

	Vaccine_Date	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Admini
--	--------------	-------	--------------------------	----------	-------	-------------------------	--------------------------	---------------------------	--------

212	16/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	
213	17/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	
214	18/01/2021	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0.0	29.0	
215	19/01/2021	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0.0	53.0	
216	20/01/2021	Andaman and Nicobar Islands	124.0	16.0	3.0	124.0	0.0	67.0	
...	...	...	...	...	...	...	...	...	
7840	11/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	
7841	12/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	
7842	13/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	
7843	14/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	
7844	15/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	

7633 rows × 24 columns

In [ ]:

```
In [28]: vaccine.rename(columns = {"Total Individuals Vaccinated": "Total"}, inplace=True)
vaccine.head()
```

C:\Users\HARSHIT\AppData\Local\Temp\ipykernel\_1816\560008889.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_](https://pandas.pydata.org/pandas-docs/stable/user_)



Out[28]:

	Vaccine_Date	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)
212	16/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0
213	17/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0
214	18/01/2021	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0.0	29.0	13.0
215	19/01/2021	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0.0	53.0	36.0
216	20/01/2021	Andaman and Nicobar Islands	124.0	16.0	3.0	124.0	0.0	67.0	57.0

5 rows × 24 columns

In [29]:

```
# Most vaccinated State

max_vac = vaccine.groupby('State')['Total'].sum().to_frame('Total')
max_vac = max_vac.sort_values('Total', ascending = False)[:5]
max_vac
```

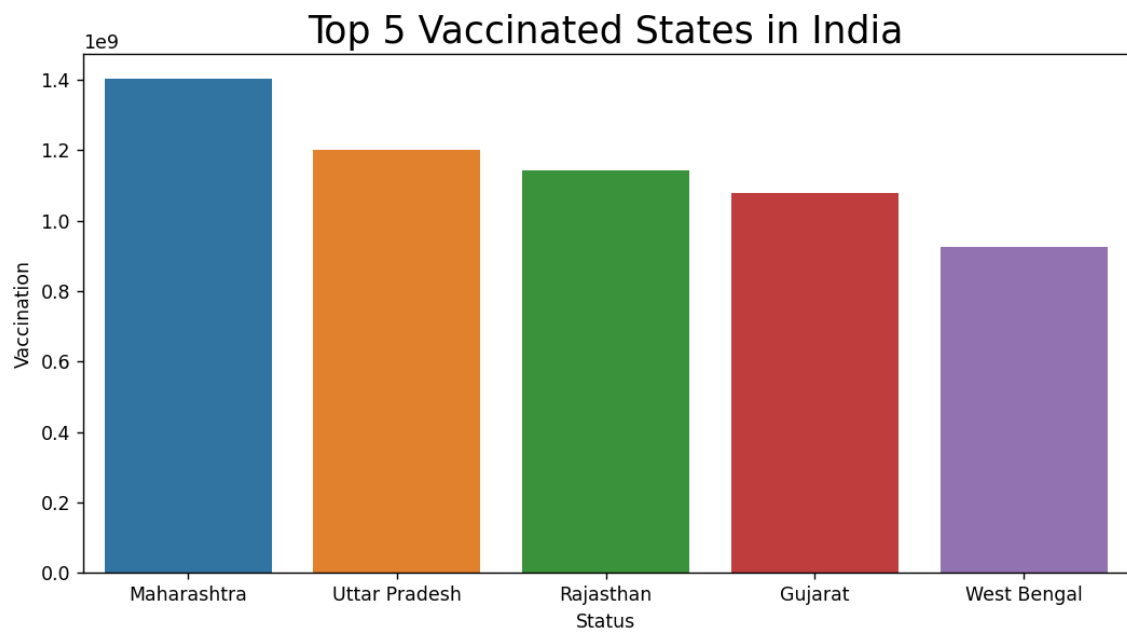
Out[29]:

	Total
State	
Maharashtra	1.403075e+09
Uttar Pradesh	1.200575e+09
Rajasthan	1.141163e+09
Gujarat	1.078261e+09
West Bengal	9.250227e+08

In [ ]:

In [30]:

```
fig = plt.figure(figsize = (10,5))
plt.title("Top 5 Vaccinated States in India", size=20)
x = sns.barplot(data = max_vac.iloc[:10], y= max_vac.Total, x = max_vac.index, linewidth=2)
plt.xlabel("Status")
plt.ylabel("Vaccination")
plt.show()
```



THANK YOU !!!