Covid-19 Data Analysis

In [1]:
```
# Importing libraries
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
from datetime import datetime
```

In [2]:
```
# Load csv file of covid-19 dataset
covid_df = pd.read_csv(r"C:\Users\ameys\Downloads\archive (1)\covid_19_india.csv")
```

In [3]:
```
# Exploring data
covid_df.head(10)
```

Out[3]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	2020-01-30	6:00 PM	Kerala	1	0	0	0	1
1	2	2020-01-31	6:00 PM	Kerala	1	0	0	0	1
2	3	2020-02-01	6:00 PM	Kerala	2	0	0	0	2
3	4	2020-02-02	6:00 PM	Kerala	3	0	0	0	3
4	5	2020-02-03	6:00 PM	Kerala	3	0	0	0	3
5	6	2020-02-04	6:00 PM	Kerala	3	0	0	0	3
6	7	2020-02-05	6:00 PM	Kerala	3	0	0	0	3
7	8	2020-02-06	6:00 PM	Kerala	3	0	0	0	3
8	9	2020-02-07	6:00 PM	Kerala	3	0	0	0	3
9	10	2020-02-08	6:00 PM	Kerala	3	0	0	0	3

In [4]:
```
# Checking how many null values present in dataset
covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18110 entries, 0 to 18109
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Sno                      18110 non-null  int64 
 1   Date                     18110 non-null  object
 2   Time                     18110 non-null  object
 3   State/UnionTerritory     18110 non-null  object
 4   ConfirmedIndianNational  18110 non-null  object
 5   ConfirmedForeignNational  18110 non-null  object
 6   Cured                    18110 non-null  int64 
 7   Deaths                   18110 non-null  int64 
 8   Confirmed                18110 non-null  int64 
dtypes: int64(4), object(5)
memory usage: 1.2+ MB
```

In [5]:
```
covid_df.describe()
```

Out[5]:

	Sno	Cured	Deaths	Confirmed
count	18110.000000	1.811000e+04	18110.000000	1.811000e+04
mean	9055.500000	2.786375e+05	4052.402264	3.010314e+05
std	5228.051023	6.148909e+05	10919.076411	6.561489e+05
min	1.000000	0.000000e+00	0.000000	0.000000e+00
25%	4528.250000	3.360250e+03	32.000000	4.376750e+03
50%	9055.500000	3.336400e+04	588.000000	3.977350e+04
75%	13582.750000	2.788698e+05	3643.750000	3.001498e+05
max	18110.000000	6.159676e+06	134201.000000	6.363442e+06

In [6]:
```
# Load vaccination dataset
vaccine_df = pd.read_csv(r"C:\Users\ameys\Downloads\archive (1)\covid_vaccine_statewise.csv")
```

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

Out[7]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	...	18-44 Years (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN	NaN	...	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN	NaN	...	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN	NaN	...	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN	NaN	...	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN	NaN	...	NaN
5	21/01/2021	India	365965.0	32226.0	12600.0	365965.0	0.0	NaN	NaN	NaN	...	NaN
6	22/01/2021	India	549381.0	36988.0	14115.0	549381.0	0.0	NaN	NaN	NaN	...	NaN

7 rows × 24 columns

In [8]:
```
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   Updated On                            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 [9]:
```
covid_df.head(10)
```

Out[9]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	2020-01-30	6:00 PM	Kerala	1	0	0	0	1
1	2	2020-01-31	6:00 PM	Kerala	1	0	0	0	1
2	3	2020-02-01	6:00 PM	Kerala	2	0	0	0	2
3	4	2020-02-02	6:00 PM	Kerala	3	0	0	0	3
4	5	2020-02-03	6:00 PM	Kerala	3	0	0	0	3
5	6	2020-02-04	6:00 PM	Kerala	3	0	0	0	3
6	7	2020-02-05	6:00 PM	Kerala	3	0	0	0	3
7	8	2020-02-06	6:00 PM	Kerala	3	0	0	0	3
8	9	2020-02-07	6:00 PM	Kerala	3	0	0	0	3
9	10	2020-02-08	6:00 PM	Kerala	3	0	0	0	3

In [10]:
```
# Drop unwanted columns
covid_df.drop(["Sno","Time","ConfirmedIndianNational","ConfirmedForeignNational"],inplace = True, axis = 1)
```

In [11]:
```
covid_df.head()
```

Out[11]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed
0	2020-01-30	Kerala	0	0	1
1	2020-01-31	Kerala	0	0	1
2	2020-02-01	Kerala	0	0	2
3	2020-02-02	Kerala	0	0	3
4	2020-02-03	Kerala	0	0	3

In [12]:
```
covid_df['Date'] = pd.to_datetime(covid_df['Date'], format = '%Y/%m/%d')
```

In [13]:
```
covid_df.head(5)
```

Out[13]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed
0	2020-01-30	Kerala	0	0	1
1	2020-01-31	Kerala	0	0	1
2	2020-02-01	Kerala	0	0	2
3	2020-02-02	Kerala	0	0	3
4	2020-02-03	Kerala	0	0	3

In [14]:
```
# Add custom active Cases column by using datsets columns
covid_df["Active_cases"] = covid_df["Confirmed"] -  (covid_df["Cured"] + covid_df["Deaths"])
```

In [15]:
```
covid_df.tail()
```

Out[15]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases
18105	2021-08-11	Telangana	638410	3831	650353	8112
18106	2021-08-11	Tripura	77811	773	80660	2076
18107	2021-08-11	Uttarakhand	334650	7368	342462	444
18108	2021-08-11	Uttar Pradesh	1685492	22775	1708812	545
18109	2021-08-11	West Bengal	1506532	18252	1534999	10215

In [16]:
```
covid_df.shape
```

Out[16]:
```
(18110, 6)
```

In [17]:
```
covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18110 entries, 0 to 18109
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype         
---  ------                --------------  -----         
 0   Date                  18110 non-null  datetime64[ns]
 1   State/UnionTerritory  18110 non-null  object        
 2   Cured                 18110 non-null  int64         
 3   Deaths                18110 non-null  int64         
 4   Confirmed             18110 non-null  int64         
 5   Active_cases          18110 non-null  int64         
dtypes: datetime64[ns](1), int64(4), object(1)
memory usage: 849.0+ KB
```

In [18]:
```
# Adding column to uncover insights from datasets
statewise = covid_df
statewise["Recovery Rate"] = statewise["Cured"]*100/statewise["Confirmed"]

statewise["Mortality Rate"] = statewise["Deaths"]*100/statewise["Confirmed"]
```

In [19]:
```
# Create pivot table
statewise = pd.pivot_table(covid_df, values = ['Confirmed','Deaths','Cured'], index = 'State/UnionTerritory' ,aggfunc = 

statewise.style.background_gradient(cmap = "cubehelix")
```

Out[19]:

State/UnionTerritory	Confirmed	Cured	Deaths
Maharashtra	6363442	6159676	134201
Maharashtra***	6229596	6000911	130753
Kerala	3586693	3396184	18004
Karnataka	2921049	2861499	36848
Karanataka	2885238	2821491	36197
Tamil Nadu	2579130	2524400	34367
Andhra Pradesh	1985182	1952736	13564
Uttar Pradesh	1708812	1685492	22775
West Bengal	1534999	1506532	18252
Delhi	1436852	1411280	25068
Chhattisgarh	1003356	988189	13544
Odisha	988997	972710	6565
Rajasthan	953851	944700	8954
Gujarat	825085	814802	10077
Madhya Pradesh	791980	781330	10514
Madhya Pradesh***	791656	780735	10506
Haryana	770114	759790	9652
Bihar	725279	715352	9646
Bihar****	715730	701234	9452
Telangana	650353	638410	3831

State/UnionTerritory	Confirmed	Cured	Deaths
Punjab	599573	582791	16322
Assam	576149	559684	5420
Telengana	443360	362160	2312
Jharkhand	347440	342102	5130
Uttarakhand	342462	334650	7368
Jammu and Kashmir	322771	317081	4392
Himachal Pradesh	208616	202761	3537
Himanchal Pradesh	204516	200040	3507
Goa	172085	167978	3164
Puducherry	121766	119115	1800
Manipur	105424	96776	1664
Tripura	80660	77811	773
Meghalaya	69769	64157	1185
Chandigarh	61992	61150	811
Arunachal Pradesh	50605	47821	248
Mizoram	46320	33722	171
Nagaland	28811	26852	585
Sikkim	28018	25095	356
Ladakh	20411	20130	207
Dadra and Nagar Haveli and Daman and Diu	10654	10646	4
Dadra and Nagar Haveli	10377	10261	4
Lakshadweep	10263	10165	51
Cases being reassigned to states	9265	0	0
Andaman and Nicobar Islands	7548	7412	129
Unassigned	77	0	0

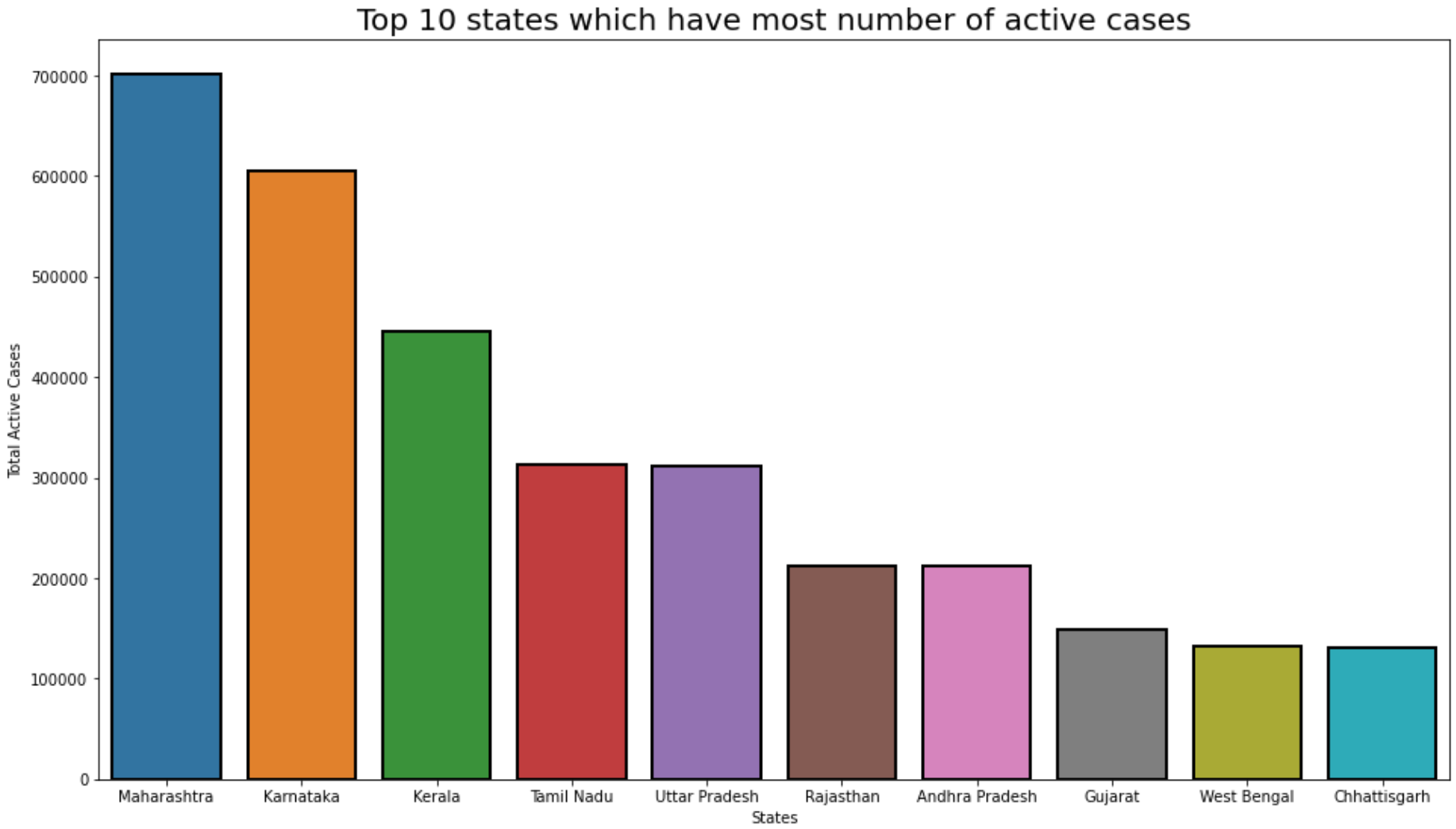
State/UnionTerritory	Confirmed	Cured	Deaths
Daman & Diu	2	0	0

In [20]:
```
# Top 10 Active Cases states
Top_active_cases = covid_df.groupby(by = "State/UnionTerritory").max()[["Active_cases", "Date"]].sort_values(by = "Activ
print(Top_active_cases)
```

	State/UnionTerritory	Active_cases	Date
0	Maharashtra	701614	2021-08-11
1	Karnataka	605515	2021-08-11
2	Kerala	445692	2021-08-11
3	Tamil Nadu	313048	2021-08-11
4	Uttar Pradesh	310783	2021-08-11
5	Rajasthan	212753	2021-08-11
6	Andhra Pradesh	211554	2021-08-11
7	Gujarat	148297	2021-08-11
8	West Bengal	132181	2021-08-11
9	Chhattisgarh	131245	2021-08-11
10	Haryana	116867	2021-08-11
11	Bihar	115152	2021-08-11
12	Madhya Pradesh	111366	2021-08-11
13	Odisha	106493	2021-08-11
14	Delhi	103424	2021-08-11
15	Maharashtra***	97932	2021-07-21
16	Telangana	80695	2021-08-11
17	Uttarakhand	80000	2021-08-11
18	Punjab	79963	2021-08-11
19	Telengana	78888	2021-05-01
20	Jharkhand	61195	2021-08-11
21	Assam	56295	2021-08-11
22	Jammu and Kashmir	52848	2021-08-11
23	Himachal Pradesh	40008	2021-08-11
24	Goa	32953	2021-08-11
25	Karanataka	27550	2021-07-20
26	Puducherry	18277	2021-08-11
27	Mizoram	13101	2021-08-11
28	Manipur	10922	2021-08-11
29	Cases being reassigned to states	9265	2020-07-18
30	Chandigarh	8653	2021-08-11
31	Tripura	8302	2021-08-11
32	Meghalaya	8255	2021-08-11
33	Nagaland	5049	2021-08-11
34	Bihar****	5044	2021-06-11

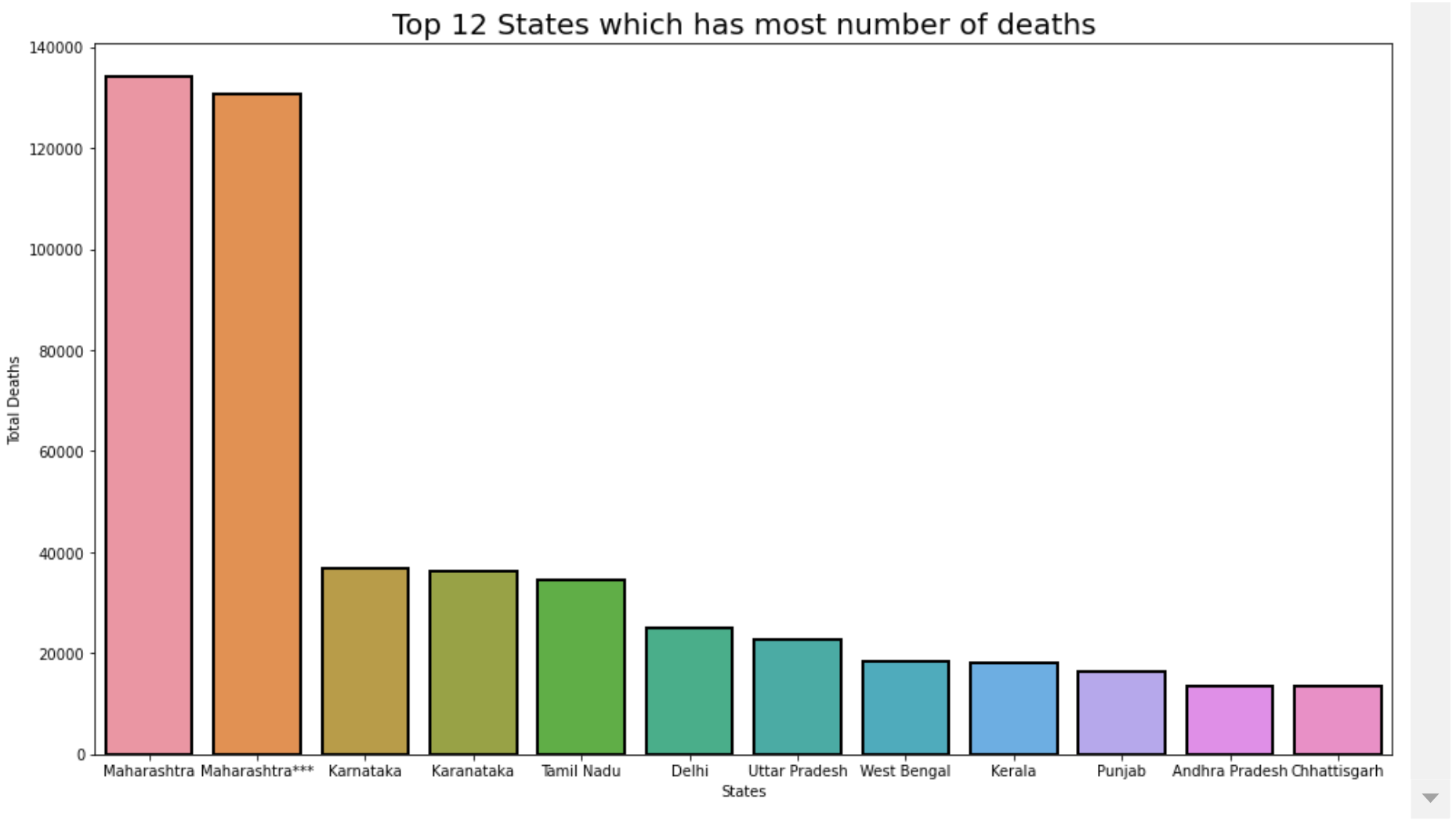
35	Arunachal Pradesh	4465	2021-08-11
36	Sikkim	4306	2021-08-11
37	Lakshadweep	2320	2021-08-11
38	Dadra and Nagar Haveli and Daman and Diu	2081	2021-08-11
39	Ladakh	2041	2021-08-11
40	Andaman and Nicobar Islands	1154	2021-08-11
41	Himanchal Pradesh	969	2021-07-20
42	Madhya Pradesh***	415	2021-07-13
43	Dadra and Nagar Haveli	250	2021-06-05
44	Unassigned	77	2020-04-03
45	Daman & Diu	2	2020-06-11

In [21]:
```
# visualizing data to find insights from datset
# Bar chart
fig = plt.figure(figsize=(16,9))
ax = sns.barplot(data = Top_active_cases.iloc[:10], x = "State/UnionTerritory" ,y ="Active_cases", linewidth = 2 , edgec
plt.xlabel("States")
plt.title("Top 10 states which have most number of active cases", size = 20)
plt.ylabel("Total Active Cases")
plt.show()
```



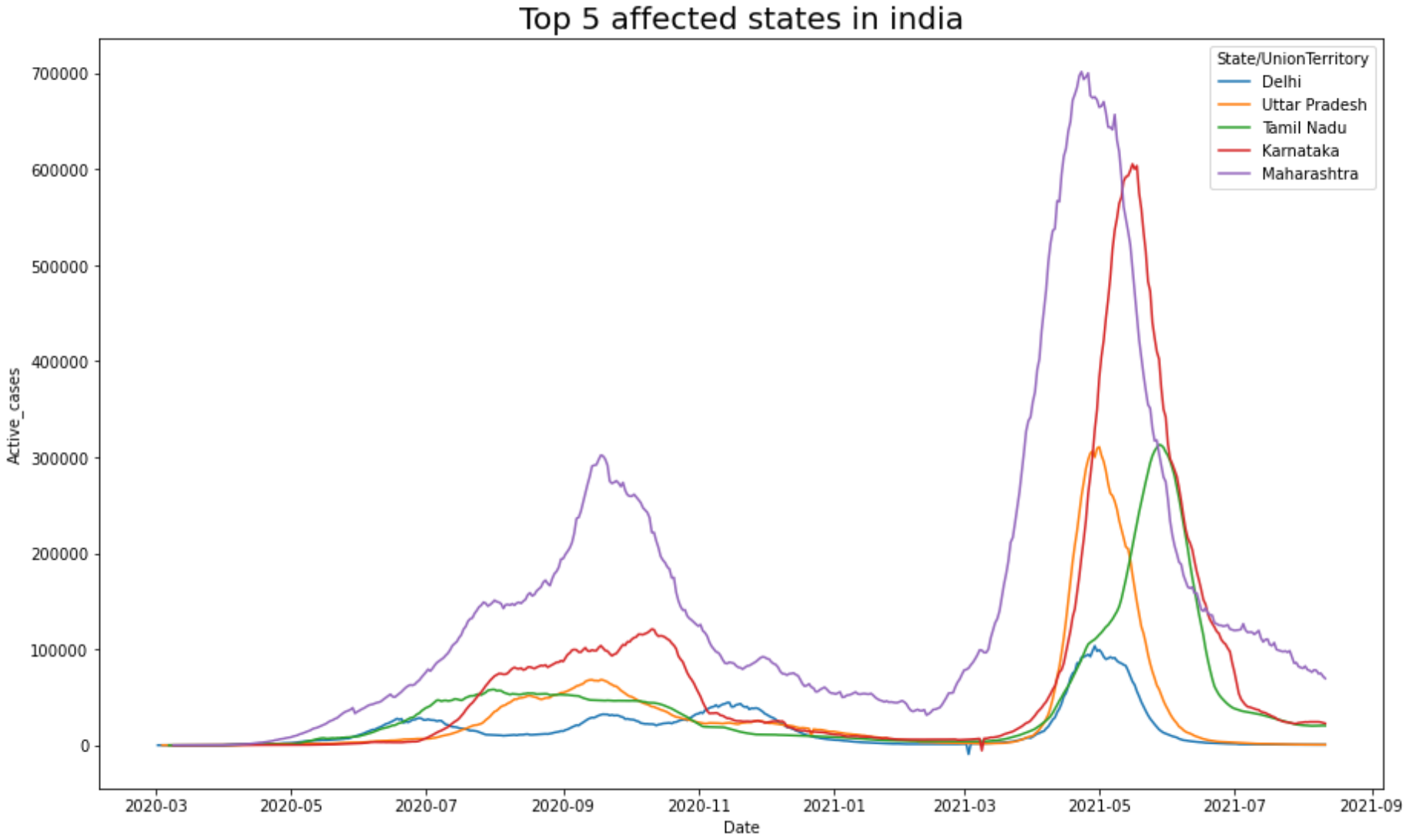
In [22]:
```
# top 10 states with most number deaths
top_states_deaths = covid_df.groupby("State/UnionTerritory").max()[["Deaths","Date"]].sort_values(by = "Deaths",ascendin
figure = plt.figure(figsize=(16,9))
ax = sns.barplot(data = top_states_deaths.iloc[:12], x ="State/UnionTerritory", y = "Deaths",linewidth = 2, edgecolor =

plt.title("Top 12 States which has most number of deaths",size = 20)
plt.xlabel("States")
plt.ylabel("Total Deaths")
plt.show()
```


Top 12 States which has most number of deaths
Total Deaths
140000
120000
100000
80000
60000
40000
20000
0
Maharashtra
Maharashtra***
Karnataka
Karanataka
Tamil Nadu
Delhi
Uttar Pradesh
West Bengal
Kerala
Punjab
Andhra Pradesh
Chhattisgarh
States

In [23]:
```
# Growth trend
figure = plt.figure(figsize = (15,9))
ax = sns.lineplot(data = covid_df[covid_df["State/UnionTerritory"].isin(["Maharashtra","Karnataka","Delhi","Tamil Nadu",
plt.title("Top 5 affected states in india", size = 20)
```

Out[23]: Text(0.5, 1.0, 'Top 5 affected states in india')



In [24]:
```
vaccine_df.head()
```

Out[24]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	...	18-44 Years (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN	NaN	...	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN	NaN	...	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN	NaN	...	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN	NaN	...	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN	NaN	...	NaN

5 rows × 24 columns

In [25]:
```
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   Updated On                            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 [26]:
```
vaccine_df.isnull().sum()
```

Out[26]:
```
Updated On                              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 [27]:
```
vaccine_df.drop(["Sputnik V (Doses Administered)","AEFI","18-44 Years (Doses Administered)","60+ Years (Doses Administer
```

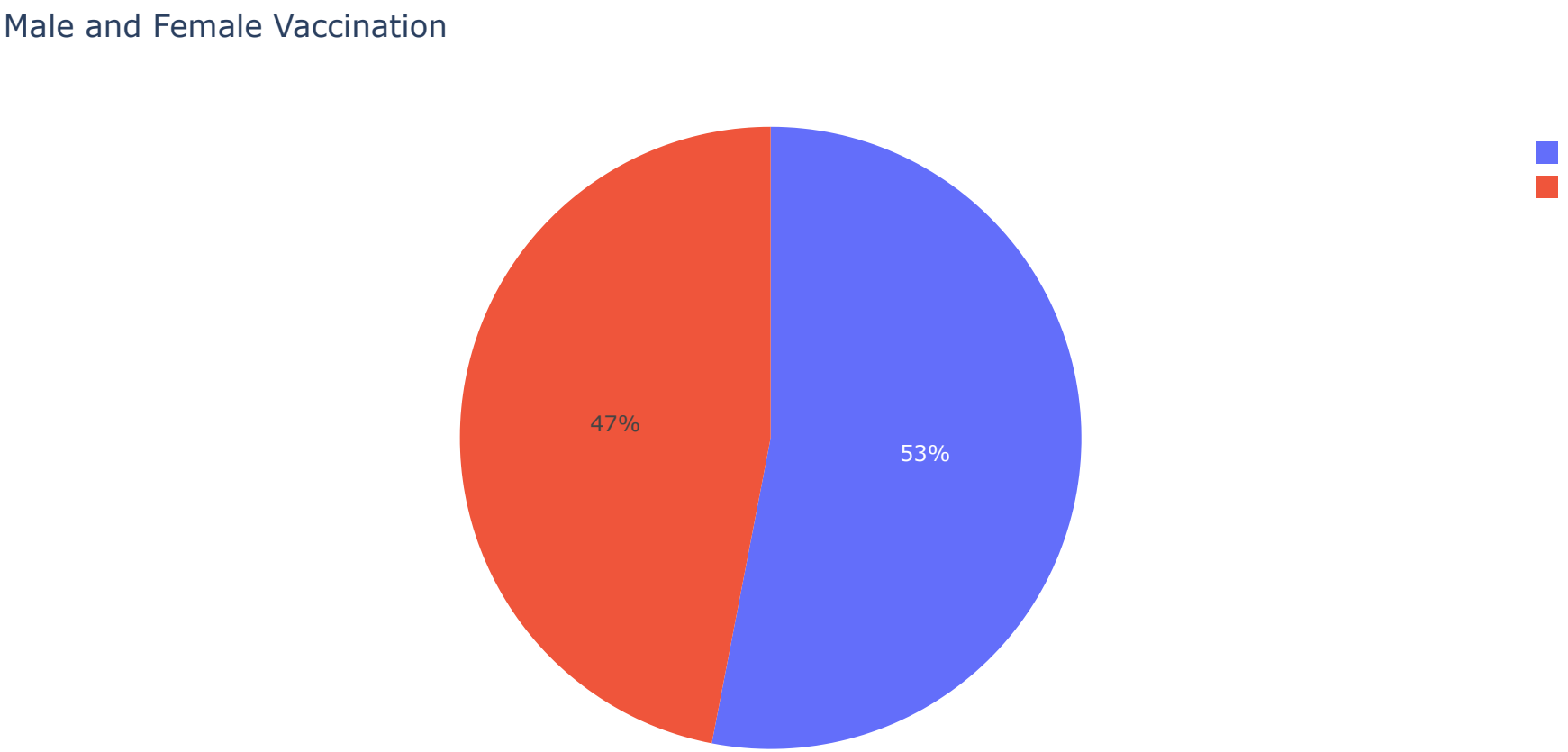
In [28]:
```
vaccine_df.head(5)
vaccine_df.rename({'vaccine date':'Vaccine Date'},inplace = True, axis =1)
```

In [29]:
```
vaccine_df.head(4)
```

Out[29]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	Covaxin (Doses Administered)	Co Admi
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN	NaN	579.0	
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN	NaN	635.0	
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN	NaN	1299.0	
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN	NaN	3017.0	

In [30]:
```
# Pie chart for male vs vaccination
male = vaccine_df["Male(Individuals Vaccinated)"].sum()
female = vaccine_df["Female(Individuals Vaccinated)"].sum()
px.pie(names = ["male","female"],values= [male,female],title = "Male and Female Vaccination")
```



In [31]:
```
# remove state name india
vaccine = vaccine_df[vaccine_df.State!="India"]
```

In [32]:
```
vaccine
```

Out[32]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	Covaxin (Doses Administered)
212	16/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0	0.0	0.0
213	17/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0	0.0	0.0
214	18/01/2021	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0.0	29.0	13.0	0.0	0.0
215	19/01/2021	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0.0	53.0	36.0	0.0	0.0
216	20/01/2021	Andaman and Nicobar Islands	124.0	16.0	3.0	124.0	0.0	67.0	57.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
7840	11/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7841	12/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7842	13/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7843	14/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7844	15/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7633 rows × 19 columns

In [33]:
```
vaccine.rename({"Total Individuals Vaccinated": "Total"},inplace = True, axis = 1)
```

```
C:\Users\ameys\AppData\Local\Temp\ipykernel_15732\1058107905.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_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

In [34]:
```
vaccine.head()
```

Out[34]:

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

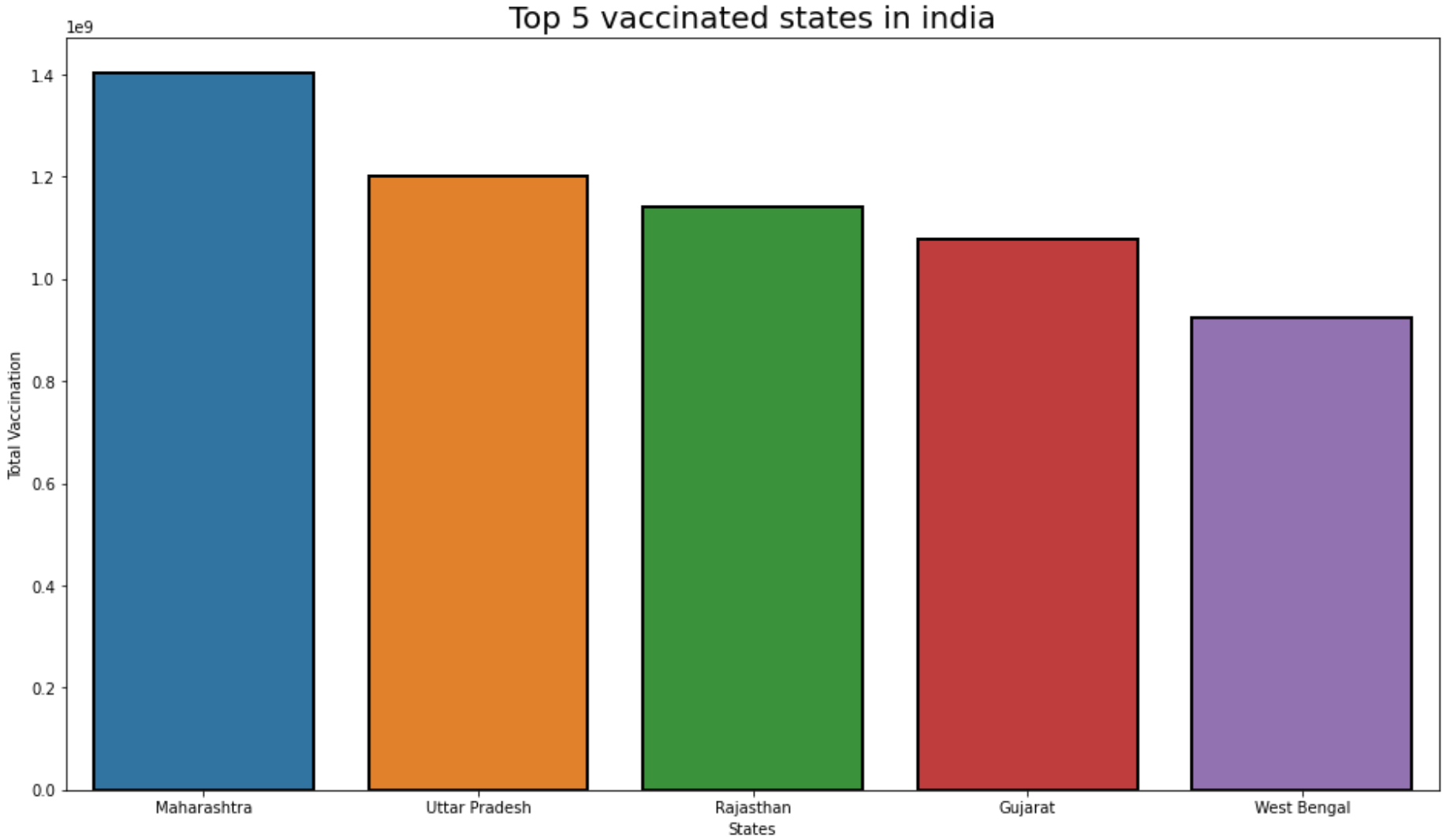
In [35]:
```
max_vac = vaccine.groupby('State')['Total'].sum().to_frame('Total').sort_values("Total",ascending = False)[:5]
```

In [36]:
```
max_vac
```

Out[36]:

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

In [37]:
```
figure = plt.figure(figsize=(16,9))
ax = sns.barplot(data = max_vac.iloc[:10], x = max_vac.index,y = max_vac.Total,linewidth = 2, edgecolor = "black")
plt.title("Top 5 vaccinated states in india",size = 20)
plt.xlabel("States")
plt.ylabel("Total Vaccination")
plt.show()
```



In [38]:
```
#  least number of individual vaccinated
vaccine.tail(5)
```

Out[38]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)	Covaxin (Doses Administered)	Ac
7840	11/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7841	12/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7842	13/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7843	14/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7844	15/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [39]:
```
least_vac = vaccine.groupby("State")["Total"].sum().to_frame('Total').sort_values("Total",ascending = True)[:5]
```

In [40]:
```
least_vac
```

Out[40]:

State	Total
Lakshadweep	2124715.0
Andaman and Nicobar Islands	8102125.0
Ladakh	9466289.0
Dadra and Nagar Haveli and Daman and Diu	11358600.0
Sikkim	16136752.0

In [41]:
```
figure = plt.figure(figsize=(16,9))
ax = sns.barplot(data = least_vac.iloc[:5],x = least_vac.index , y = least_vac.Total, linewidth  = 2 , edgecolor = "blacl
plt.title("Top 5 least vaccinated states", size = 20)
plt.xlabel("States")
plt.ylabel("Total Vaccination")
plt.show()
```