In [1]:

```
!pip install plotly
!pip install matplotlib
!pip install seaborn
```

```
Requirement already satisfied: plotly in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (5.14.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages
(from plotly) (8.2.2)
Requirement already satisfied: packaging in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (from
plotly) (23.1)
Requirement already satisfied: matplotlib in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (3.
7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-package
s (from matplotlib) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (f
rom matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packag
es (from matplotlib) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packag
es (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (fr
om matplotlib) (1.24.2)
Requirement already satisfied: packaging>=20.0 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-package
s (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\swath\appdata\local\programs\python\python310\lib\site-pac
kages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (0.12.
2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\swath\appdata\local\programs\python\python310\lib\site-pac
kages (from seaborn) (1.24.2)
Requirement already satisfied: pandas>=0.25 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (f
rom seaborn) (2.0.0)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-
packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-package
s (from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (f
rom matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packag
es (from matplotlib!=3.6.1,>=3.1->seaborn) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packag
es (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.1->seaborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-package
s (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\swath\appdata\local\programs\python\python310\lib\site-pac
kages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (f
rom pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages
(from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
```

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
from datetime import datetime
```

In [5]:
```
pip install openpyxl
```

```
Collecting openpyxl
  Downloading openpyxl-3.1.2-py2.py3-none-any.whl (249 kB)
                                              0.0/250.0 kB ? eta -:--:--
     -------------------------------        204.8/250.0 kB 6.3 MB/s eta 0:00:01
     -------------------------------------- 250.0/250.0 kB 5.1 MB/s eta 0:00:00
Collecting et-xmlfile (from openpyxl)
  Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
Installing collected packages: et-xmlfile, openpyxl
Successfully installed et-xmlfile-1.1.0 openpyxl-3.1.2
Note: you may need to restart the kernel to use updated packages.
```

In [6]:
```
!pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (3.1.
2)
Requirement already satisfied: et-xmlfile in c:\users\swath\appdata\local\programs\python\python310\lib\site-packages (fro
m openpyxl) (1.1.0)
```

In [7]:
```
COVID_df=pd.read_excel(r'D:\Certificates\SQL Jupyter Notebook\India COVID-19 VACCINATION\covid_19_india DATA.xlsx')
```

In [8]:
```
COVID_df.head()
```

Out[8]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	2021-08-11	18:00:00	Kerala	1	0	0	0	1
1	2	2021-08-11	18:00:00	Kerala	1	0	0	0	1
2	3	2021-08-11	18:00:00	Kerala	2	0	0	0	2
3	4	2021-08-11	18:00:00	Kerala	3	0	0	0	3
4	5	2021-08-11	18:00:00	Kerala	3	0	0	0	3

In [9]:
```
COVID_df.head(10)
```

Out[9]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	2021-08-11	18:00:00	Kerala	1	0	0	0	1
1	2	2021-08-11	18:00:00	Kerala	1	0	0	0	1
2	3	2021-08-11	18:00:00	Kerala	2	0	0	0	2
3	4	2021-08-11	18:00:00	Kerala	3	0	0	0	3
4	5	2021-08-11	18:00:00	Kerala	3	0	0	0	3
5	6	2021-08-11	18:00:00	Kerala	3	0	0	0	3
6	7	2021-08-11	18:00:00	Kerala	3	0	0	0	3
7	8	2021-08-11	18:00:00	Kerala	3	0	0	0	3
8	9	2021-08-11	18:00:00	Kerala	3	0	0	0	3
9	10	2021-08-11	18:00:00	Kerala	3	0	0	0	3

In [10]:
```
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  datetime64[ns]
 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: datetime64[ns](1), int64(4), object(4)
memory usage: 1.2+ MB
```

In [11]:
```
COVID_df.describe()
```

Out[11]:

	Sno	Date	Cured	Deaths	Confirmed
count	18110.000000	18110	1.811000e+04	18110.000000	1.811000e+04
mean	9055.500000	2020-11-30 21:49:50.127001600	2.786375e+05	4052.402264	3.010314e+05
min	1.000000	2020-01-30 00:00:00	0.000000e+00	0.000000	0.000000e+00
25%	4528.250000	2020-07-26 00:00:00	3.360250e+03	32.000000	4.376750e+03
50%	9055.500000	2020-12-03 00:00:00	3.336400e+04	588.000000	3.977350e+04
75%	13582.750000	2021-04-08 00:00:00	2.788698e+05	3643.750000	3.001498e+05
max	18110.000000	2021-08-11 00:00:00	6.159676e+06	134201.000000	6.363442e+06
std	5228.051023	NaN	6.148909e+05	10919.076411	6.561489e+05

In [13]:
```
print(COVID_df.columns)
```

```
Index(['Sno', 'Date', 'Time', 'State/UnionTerritory',
       'ConfirmedIndianNational', 'ConfirmedForeignNational', 'Cured',
       'Deaths', 'Confirmed'],
      dtype='object')
```

In [14]:
```
COVID_df.drop(["Sno","Time","ConfirmedIndianNational", "ConfirmedForeignNational"],inplace = True,axis = 1 )
```

In [15]:
```
COVID_df.head()
```

Out[15]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed
0	2021-08-11	Kerala	0	0	1
1	2021-08-11	Kerala	0	0	1
2	2021-08-11	Kerala	0	0	2
3	2021-08-11	Kerala	0	0	3
4	2021-08-11	Kerala	0	0	3

In [16]:
```
COVID_df.head()
```

Out[16]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed
0	2021-08-11	Kerala	0	0	1
1	2021-08-11	Kerala	0	0	1
2	2021-08-11	Kerala	0	0	2
3	2021-08-11	Kerala	0	0	3
4	2021-08-11	Kerala	0	0	3

In [17]:
```
Covid_df=pd.read_excel(r'D:\Certificates\SQL Jupyter Notebook\India COVID-19 VACCINATION\covid_19_india DATA.xlsx')
```

In [18]:
```
Covid_df.head()
```

Out[18]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	2020-01-30	18:00:00	Kerala	1	0	0	0	1
1	2	2020-01-31	18:00:00	Kerala	1	0	0	0	1
2	3	2020-02-01	18:00:00	Kerala	2	0	0	0	2
3	4	2020-02-02	18:00:00	Kerala	3	0	0	0	3
4	5	2020-02-03	18:00:00	Kerala	3	0	0	0	3

In [20]:
```
Covid_df.drop(["Sno","Time","ConfirmedIndianNational", "ConfirmedForeignNational"],inplace = True,axis = 1 )
```

In [21]:
```
Covid_df.head()
```

Out[21]:

	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 [23]:
```
Covid_df['Date'] = pd.to_datetime(Covid_df['Date'],format ='%Y-%m-%d')
```

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

Out[24]:

	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 [26]:
```
# Active cases = Confirmed - cured +death

Covid_df['Active_Cases'] = Covid_df['Confirmed'] - (Covid_df['Cured'] + Covid_df['Deaths'])
Covid_df.tail()
```

Out[26]:

	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 [27]:
```
statewise = pd.pivot_table(Covid_df, values = ["Confirmed","Deaths","Cured"], index = "State/UnionTerritory", aggfunc = max
```

In [28]:
```
statewise["Recovery Rate"] = statewise["Cured"]*100/statewise["Confirmed"]
```

In [29]:
```
statewise["Mortality Rate"] = statewise["Deaths"]*100/statewise["Confirmed"]
```

In [30]:
```
statewise = statewise.sort_values(by = "Confirmed", ascending = False)
```

In [31]:
```
statewise.style.background_gradient(cmap = "cubehelix")
```

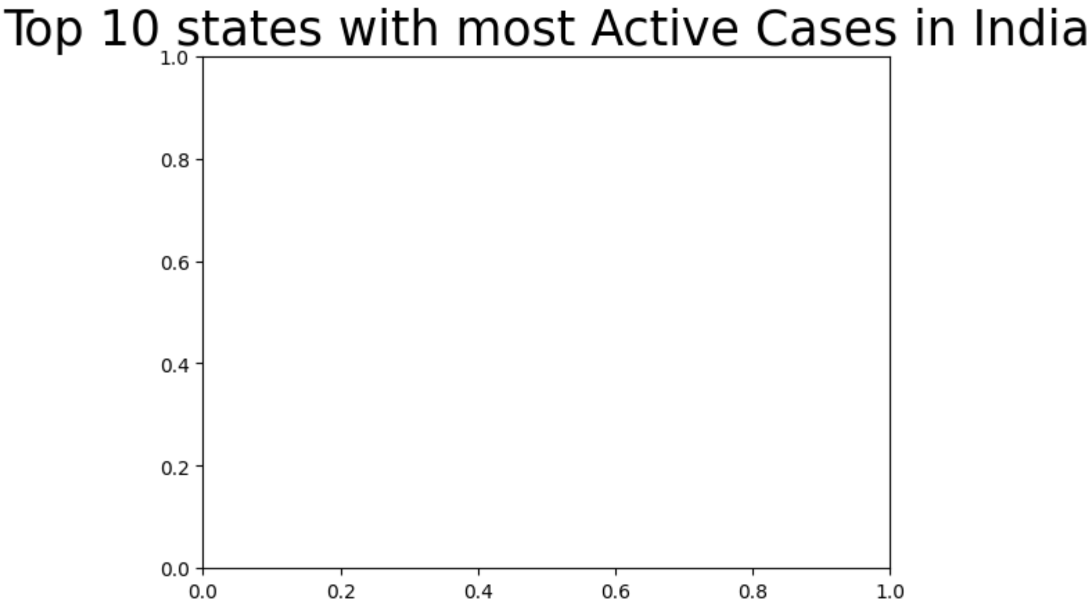
Out[31]:

	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
State/UnionTerritory					
Maharashtra	6363442	6159676	134201	96.797865	2.108937
Maharashtra* **	6229596	6000911	130753	96.329056	2.098900
Kerala	3586693	3396184	18004	94.688450	0.501967
Karnataka	2921049	2861499	36848	97.961349	1.261465
Karanataka	2885238	2821491	36197	97.790581	1.254559
Tamil Nadu	2579130	2524400	34367	97.877967	1.332504
Andhra Pradesh	1985182	1952736	13564	98.365591	0.683262
Uttar Pradesh	1708812	1685492	22775	98.635309	1.332797
West Bengal	1534999	1506532	18252	98.145471	1.189056
Delhi	1436852	1411280	25068	98.220276	1.744647
Chhattisgarh	1003356	988189	13544	98.488373	1.349870
Odisha	988997	972710	6565	98.353180	0.663804
Rajasthan	953851	944700	8954	99.040626	0.938721
Gujarat	825085	814802	10077	98.753704	1.221329
Madhya Pradesh	791980	781330	10514	98.655269	1.327559
Madhya Pradesh* **	791656	780735	10506	98.620487	1.327092
Haryana	770114	759790	9652	98.659419	1.253321
Bihar	725279	715352	9646	98.631285	1.329971
Bihar** **	715730	701234	9452	97.974655	1.320610
Telangana	650353	638410	3831	98.163613	0.589065
Punjab	599573	582791	16322	97.201008	2.722271
Assam	576149	559684	5420	97.142232	0.940729
Telengana	443360	362160	2312	81.685312	0.521472
Jharkhand	347440	342102	5130	98.463620	1.476514
Uttarakhand	342462	334650	7368	97.718871	2.151480
Jammu and Kashmir	322771	317081	4392	98.237140	1.360717
Himachal Pradesh	208616	202761	3537	97.193408	1.695460
Himanchal Pradesh	204516	200040	3507	97.811418	1.714780
Goa	172085	167978	3164	97.613389	1.838626
Puducherry	121766	119115	1800	97.822873	1.478245
Manipur	105424	96776	1664	91.796934	1.578388
Tripura	80660	77811	773	96.467890	0.958344
Meghalaya	69769	64157	1185	91.956313	1.698462
Chandigarh	61992	61150	811	98.641760	1.308233
Arunachal Pradesh	50605	47821	248	94.498567	0.490070
Mizoram	46320	33722	171	72.802245	0.369171
Nagaland	28811	26852	585	93.200514	2.030474
Sikkim	28018	25095	356	89.567421	1.270612
Ladakh	20411	20130	207	98.623291	1.014159
Dadra and Nagar Haveli and Daman and Diu	10654	10646	4	99.924911	0.037545
Dadra and Nagar Haveli	10377	10261	4	98.882143	0.038547
Lakshadweep	10263	10165	51	99.045114	0.496931
Cases being reassigned to states	9265	0	0	0.000000	0.000000
Andaman and Nicobar Islands	7548	7412	129	98.198198	1.709062
Unassigned	77	0	0	0.000000	0.000000
Daman & Diu	2	0	0	0.000000	0.000000

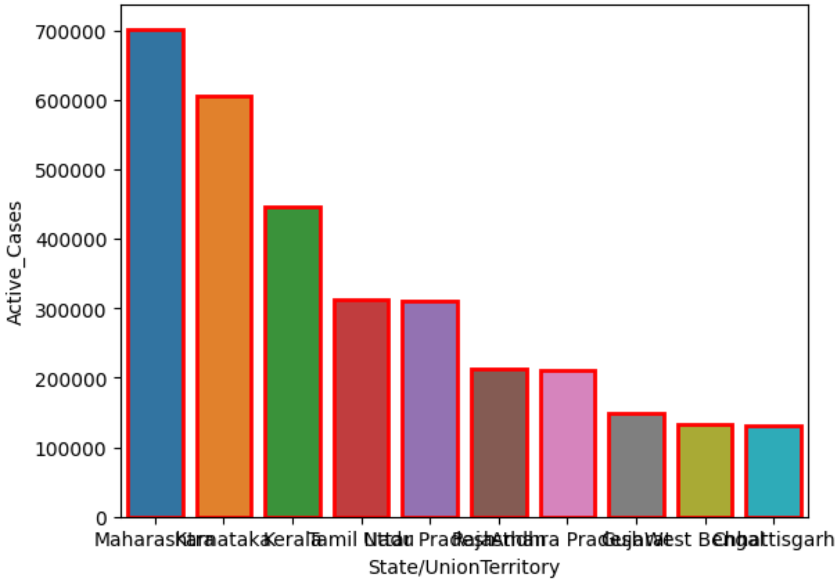
```
In [40]: top_10_active_cases = Covid_df.groupby(by='State/UnionTerritory').max()[['Active_Cases','Date']].sort_values(by=['Active_Ca
```

```
In [41]: fig= plt.figure(figsize=(16,9))

         <Figure size 1600x900 with 0 Axes>
```

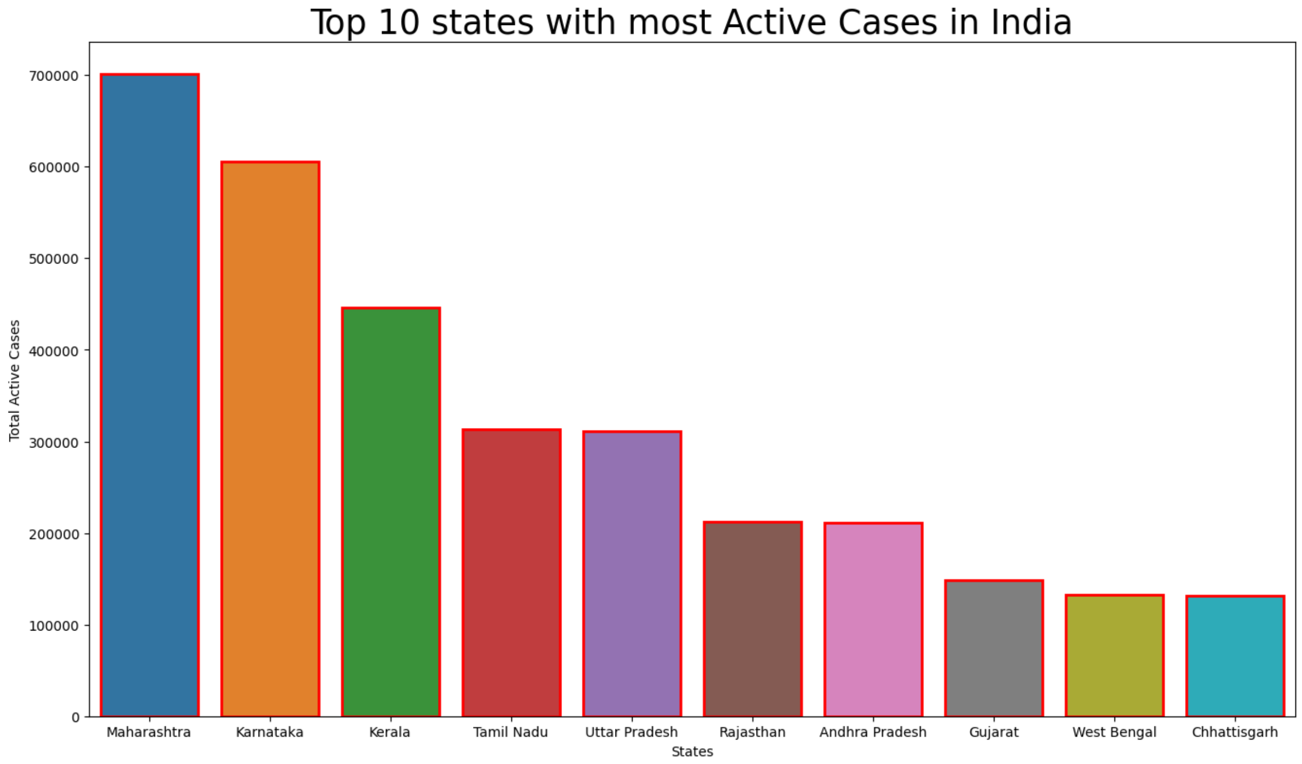
```
In [42]: plt.title("Top 10 states with most Active Cases in India", size = 25)

Out[42]: Text(0.5, 1.0, 'Top 10 states with most Active Cases in India')
```



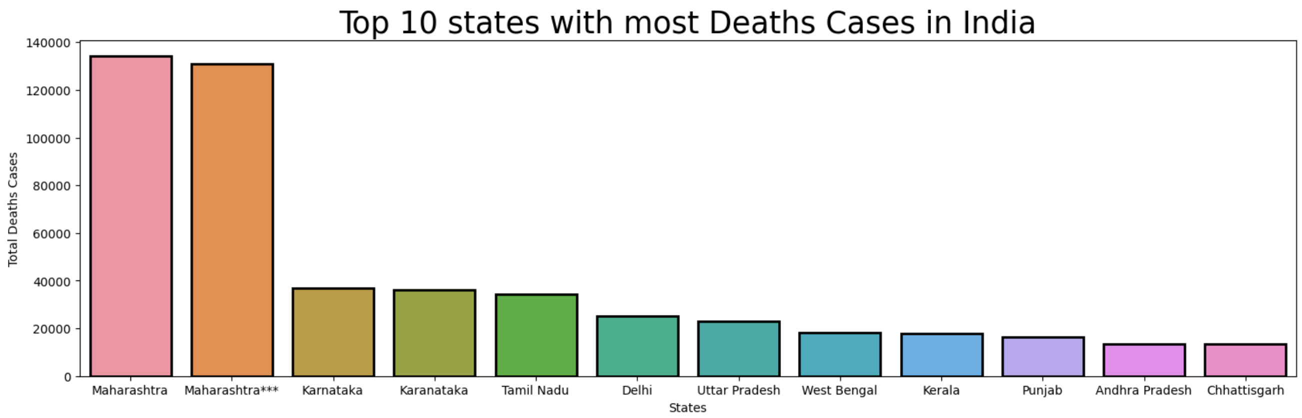
```
In [44]: ax=sns.barplot(data = top_10_active_cases.iloc[:10],y ="Active_Cases",x="State/UnionTerritory",linewidth=2, edgecolor = 're
```



```
In [45]: # "Top 10 states with most Active Cases in India

         top_10_active_cases = Covid_df.groupby(by='State/UnionTerritory').max()[['Active_Cases','Date']].sort_values(by=['Active_Ca

         fig= plt.figure(figsize=(16,9))

         plt.title("Top 10 states with most Active Cases in India", size = 25)

         ax=sns.barplot(data = top_10_active_cases.iloc[:10],y ="Active_Cases",x="State/UnionTerritory",linewidth=2, edgecolor = 're

         plt.xlabel("States")
         plt.ylabel("Total Active Cases")
         plt.show()
```

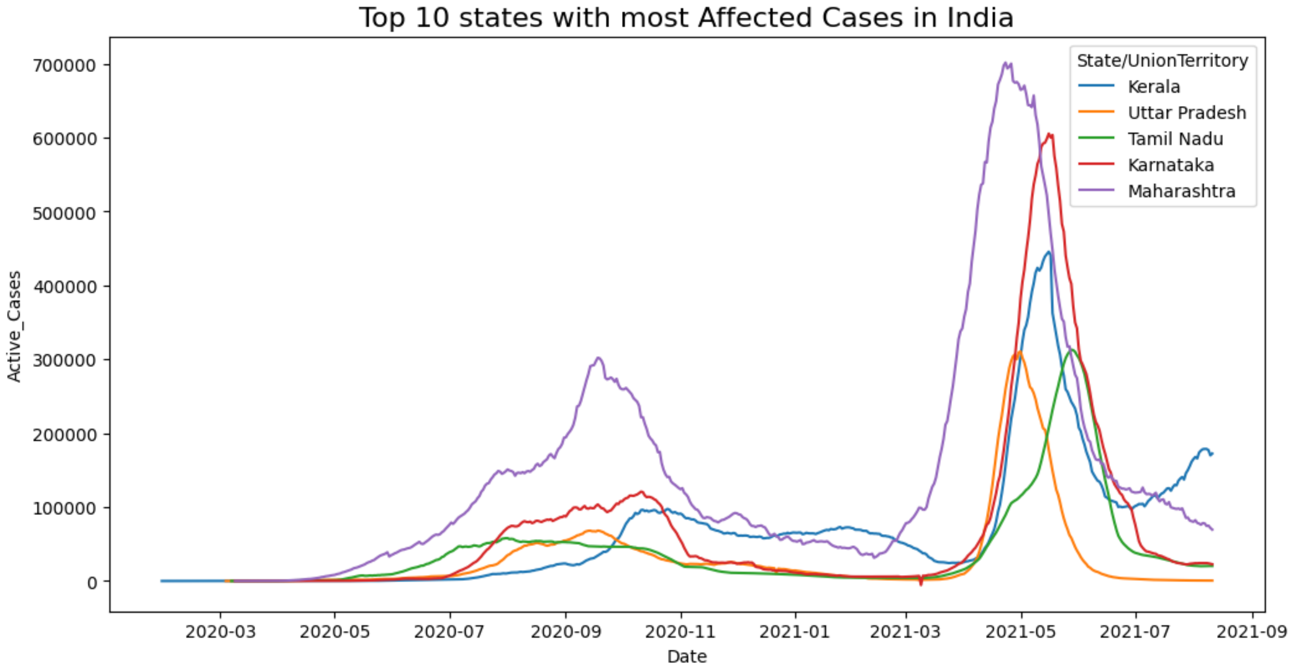


In [48]:
```
# Top States in Deaths

top_10_deaths = Covid_df.groupby(by='State/UnionTerritory').max()[['Deaths','Date']].sort_values(by=['Deaths'], ascending=F

fig= plt.figure(figsize=(18,5))

plt.title("Top 10 states with most Deaths Cases in India", size = 25)

ax=sns.barplot(data = top_10_deaths.iloc[:12],y ="Deaths",x="State/UnionTerritory",linewidth=2, edgecolor = 'black' )

plt.xlabel("States")
plt.ylabel("Total Deaths Cases")
plt.show()
```



In [54]:
```
# Growth Trend

fig= plt.figure(figsize=(12,6))

ax=sns.lineplot(data=Covid_df[Covid_df["State/UnionTerritory"].isin(['Maharashtra','Karnataka','Kerala','Tamil Nadu','Uttar
ax.set_title("Top 10 states with most Affected Cases in India", size = 16)
```

Out[54]:
```
Text(0.5, 1.0, 'Top 10 states with most Affected Cases in India')
```



In [55]:
```
Vaccine_df=pd.read_csv('D:\Certificates\SQL Jupyter Notebook\India COVID-19 VACCINATION\covid_vaccine_statewise.csv')
```

In [56]:
```
Vaccine_df.head()
```

Out[56]:

	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)	45- Admi
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 [59]:
```
Vaccine_df.rename(columns = { 'Updated On' : 'Vaccine_Date'}, inplace = True)
```

In [60]:
```
Vaccine_df.head(10)
```

Out[60]:

	Vaccine_Date	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)	Ad
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	23/01/2021	India	759008.0	43076.0	15605.0	759008.0	0.0	NaN	NaN	NaN	...	NaN	
8	24/01/2021	India	835058.0	49851.0	18111.0	835058.0	0.0	NaN	NaN	NaN	...	NaN	
9	25/01/2021	India	1277104.0	55151.0	19682.0	1277104.0	0.0	NaN	NaN	NaN	...	NaN	

10 rows × 24 columns

In [63]:
```
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 [64]:
```
Vaccine_df.isnull().sum()
```

Out[64]:
```
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
```

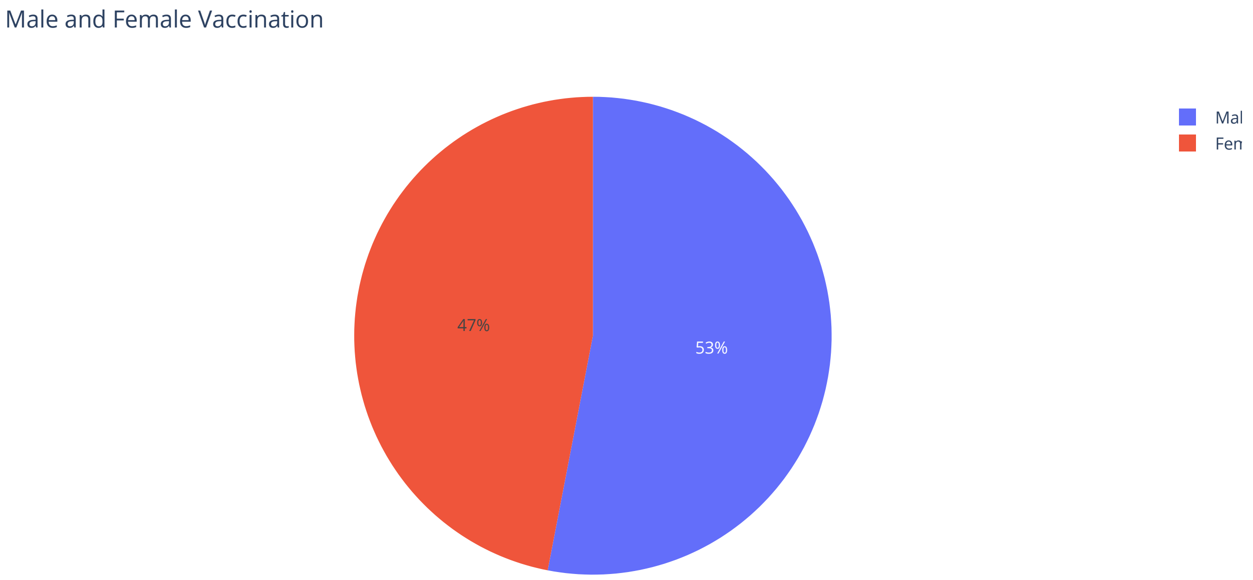
In [66]:
```
Vaccination =Vaccine_df.drop(columns = ['Sputnik V (Doses Administered)','AEFI','18-44 Years (Doses Administered)','45-60 Y
```

In [67]:
```
Vaccination.head()
```

Out[67]:

ıder ıses red)	Covaxin (Doses Administered)	CoviShield (Doses Administered)	18-44 Years(Individuals Vaccinated)	45-60 Years(Individuals Vaccinated)	60+ Years(Individuals Vaccinated)	Male(Individuals Vaccinated)	Female(Individuals Vaccinated)	Transgender(Individuals Vaccinated)	Indiv Vacc
NaN	579.0	47697.0	NaN	NaN	NaN	23757.0	24517.0	2.0	4
NaN	635.0	57969.0	NaN	NaN	NaN	27348.0	31252.0	4.0	5
NaN	1299.0	98150.0	NaN	NaN	NaN	41361.0	58083.0	5.0	9
NaN	3017.0	192508.0	NaN	NaN	NaN	81901.0	113613.0	11.0	19
NaN	3946.0	247334.0	NaN	NaN	NaN	98111.0	153145.0	24.0	25

In [71]:
```
# Male vs  Female

male= Vaccination["Male(Individuals Vaccinated)"].sum()
female = Vaccination["Female(Individuals Vaccinated)"].sum()
px.pie(names=["Male","Female"],values=[male,female],title="Male and Female Vaccination")
```



In [73]:
```
#Remove Rows State is india

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

Out[73]:

ears ses red)	45-60 Years (Doses Administered)	60+ Years (Doses Administered)	18-44 Years(Individuals Vaccinated)	45-60 Years(Individuals Vaccinated)	60+ Years(Individuals Vaccinated)	Male(Individuals Vaccinated)	Female(Individuals Vaccinated)	Transgender(Individuals Vaccinated)	Indiv Vacc
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [75]:
```
vaccine.rename(columns = {"Total Individuals Vaccinated" : "Totals"}, inplace = True)
vaccine.head()
```

```
C:\Users\swath\AppData\Local\Temp\ipykernel_28816\47294203.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-vi
ew-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

Out[75]:

	Vaccine_Date	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)
212	16/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0	0.0	...	NaN
213	17/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0	0.0	...	NaN
214	18/01/2021	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0.0	29.0	13.0	0.0	...	NaN
215	19/01/2021	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0.0	53.0	36.0	0.0	...	NaN
216	20/01/2021	Andaman and Nicobar Islands	124.0	16.0	3.0	124.0	0.0	67.0	57.0	0.0	...	NaN

5 rows × 24 columns

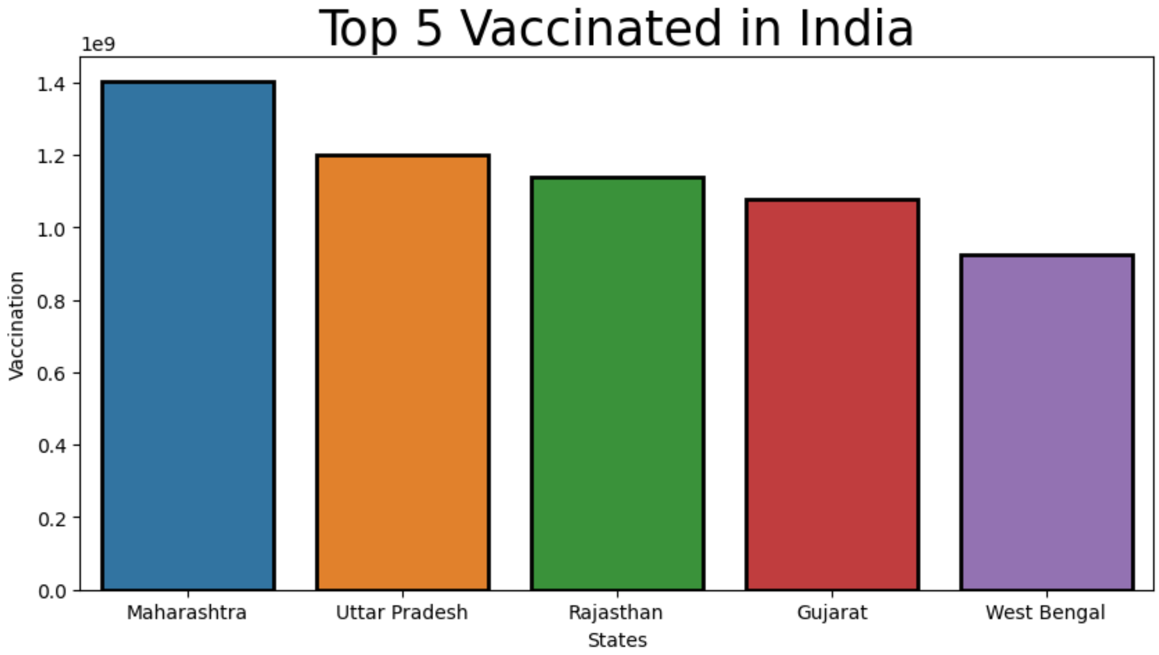
In [78]:
```
# Most Vaccinated State

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

Out[78]:

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

In [83]:

```
fig= plt.figure(figsize=(10,5))

plt.title("Top 5 Vaccinated in India", size = 25)

ax=sns.barplot(data = max_vac.iloc[:10],y =max_vac.Totals, x=max_vac.index,linewidth=2, edgecolor = 'black')

plt.xlabel("States")
plt.ylabel("Vaccination")
plt.show()
```



In [ ]: