In [4]:
```
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 [4]:
```
```

In [5]:
```
covid_df =pd.read_csv("C:/Users/nanav/Downloads/COVID_ POrtfolio_project/covid_19_indi
```

In [6]:
```
covid_df.head(10)
```

Out[6]:

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

In [7]:
```
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 [8]:
```
covid_df.describe()
```

Out[8]:

	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 [9]:
```
vaccine_df = pd.read_csv("C:/Users/nanav/Downloads/COVID_ POrtfolio_project/covid_vacc
```

In [10]:
```
vaccine_df.head(7)
```

Out[10]:

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

7 rows × 24 columns

In [11]:
```
covid_df.drop(["Sno", "Time", "ConfirmedIndianNational", "ConfirmedForeignNational"],
```

In [12]:
```
covid_df.head()
```

Out[12]:

	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 [13]:
```
covid_df['Date'] = pd.to_datetime(covid_df['Date'], format = '%Y-%m-%d')
```

In [14]:
```
covid_df.head()
```

Out[14]:

	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 [15]:
```
# Active cases

covid_df['Active_cases'] = covid_df['Confirmed'] - (covid_df['Cured'] + covid_df['Deat
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]:
```
statewise = pd.pivot_table(covid_df, values = ["Confirmed", "Deaths", "Cured"], index
```

In [17]:
```
statewise["Recovery Rate"] = statewise["Cured"]*100/statewise["Confirmed"]
```

In [18]:
```
statewise["Mortality Rate"] = statewise["Deaths"]*100/statewise["Confirmed"]
```

In [19]:
```
statewise = statewise.sort_values(by = "Confirmed", ascending = False)
```

In [20]:
```
statewise.style.background_gradient(cmap = "rainbow")
```

Out[20]:

State/UnionTerritory	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
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

State/UnionTerritory	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
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

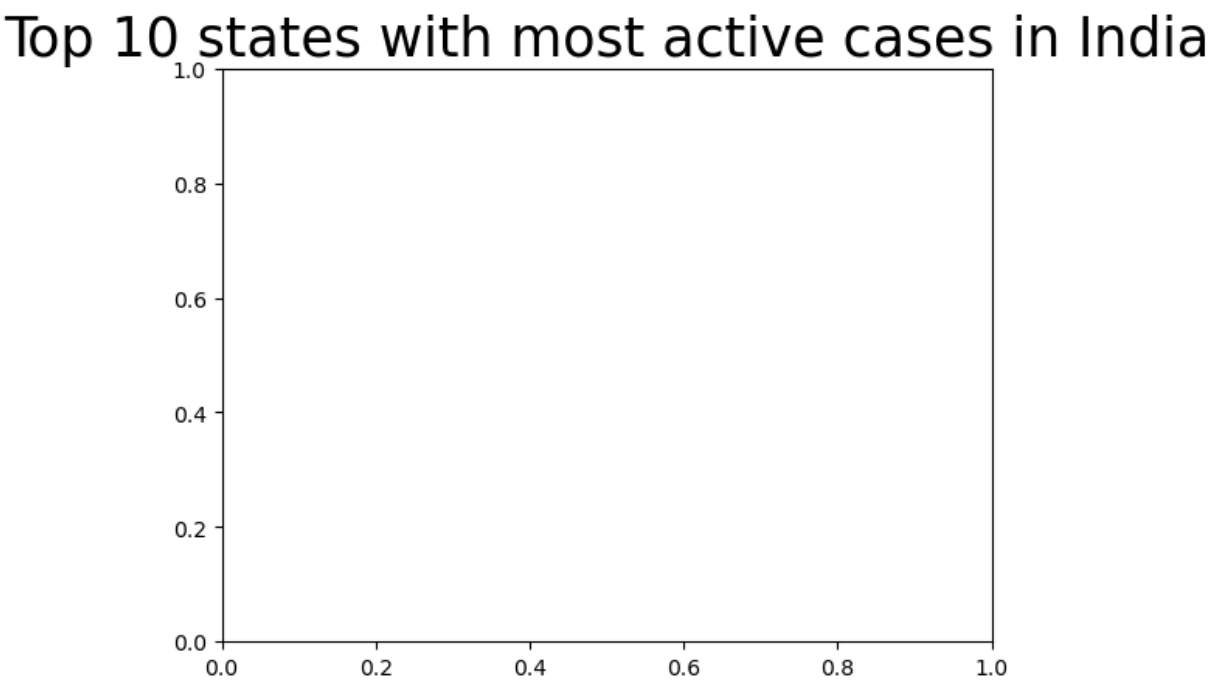
In [21]:
```
#  Top 10 active cases states

top_10_active_cases = covid_df.groupby(by = 'State/UnionTerritory').max()[['Active_cas
```

In [22]:
```
fig = plt.figure(figsize = (16, 9))
```

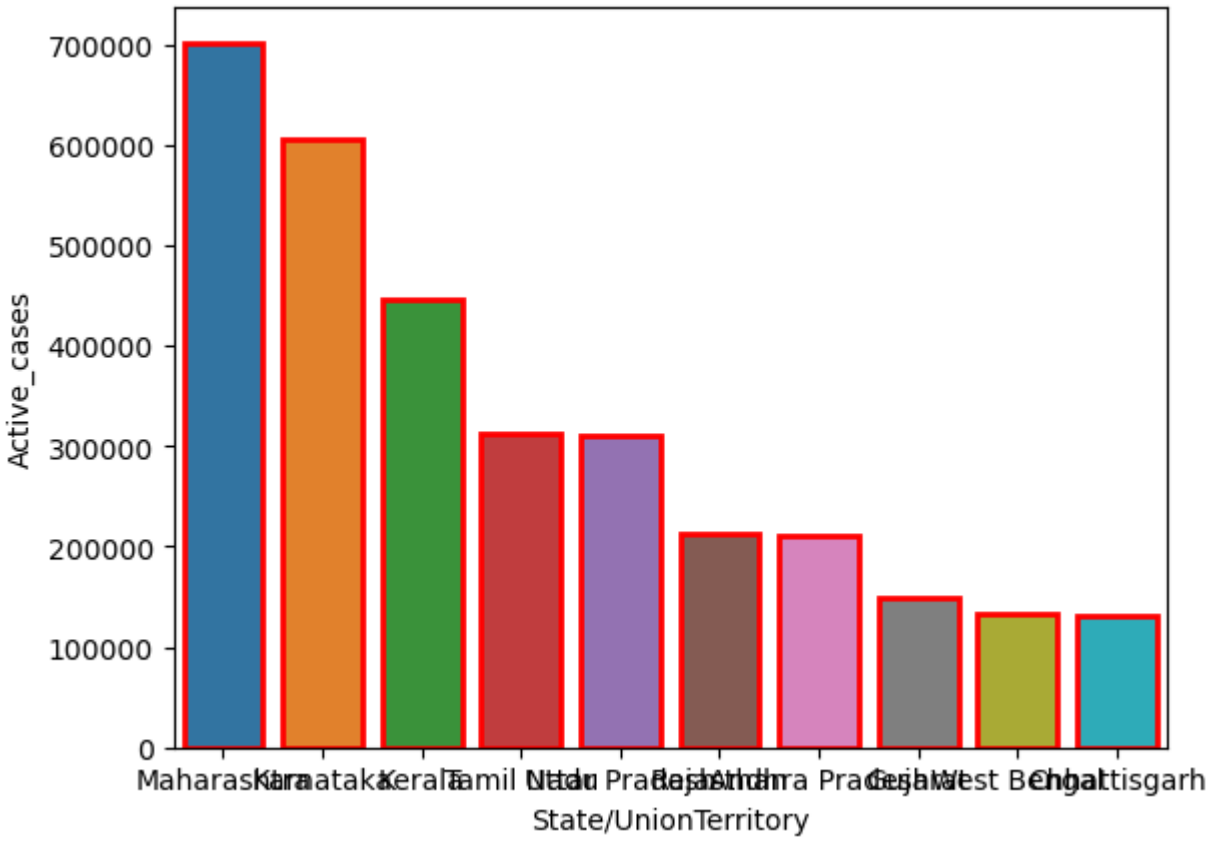
```
<Figure size 1600x900 with 0 Axes>
```

In [23]:
```
plt.title("Top 10 states with most active cases in India", size = 25)
```

Out[23]:
```
Text(0.5, 1.0, 'Top 10 states with most active cases in India')
```

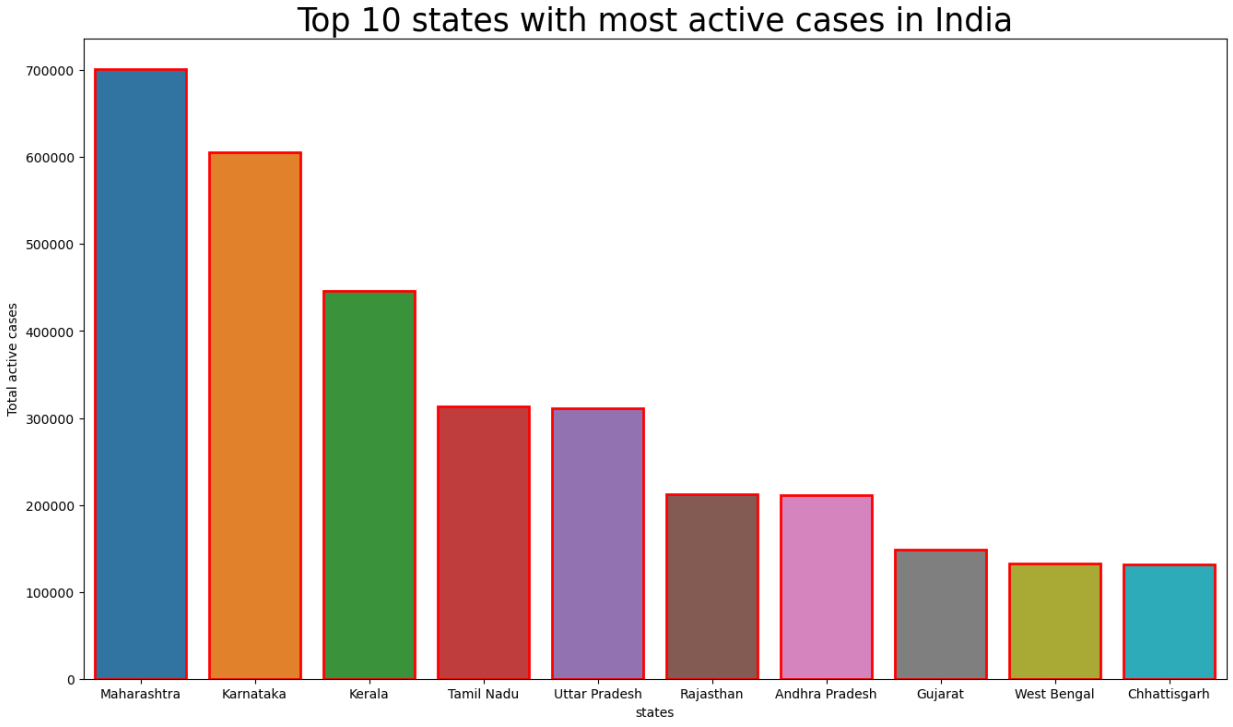


In [24]:
```
ax = sns.barplot(data = top_10_active_cases.iloc[:10], y = "Active_cases", x = "State/
```

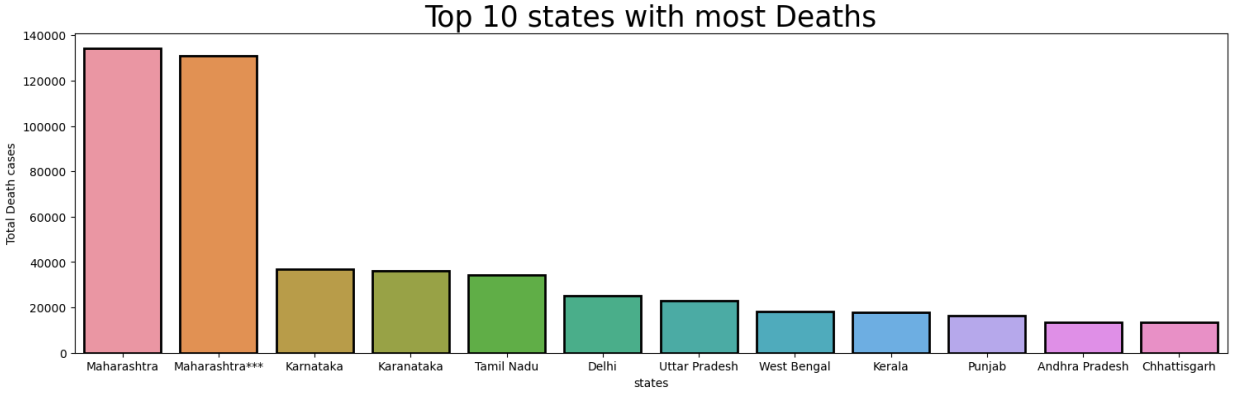


In [25]:
```
#  Top 10 active cases states

top_10_active_cases = covid_df.groupby(by = 'State/UnionTerritory').max()[['Active_cas

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/

plt.xlabel("states")
plt.ylabel("Total active cases")
plt.show()
```



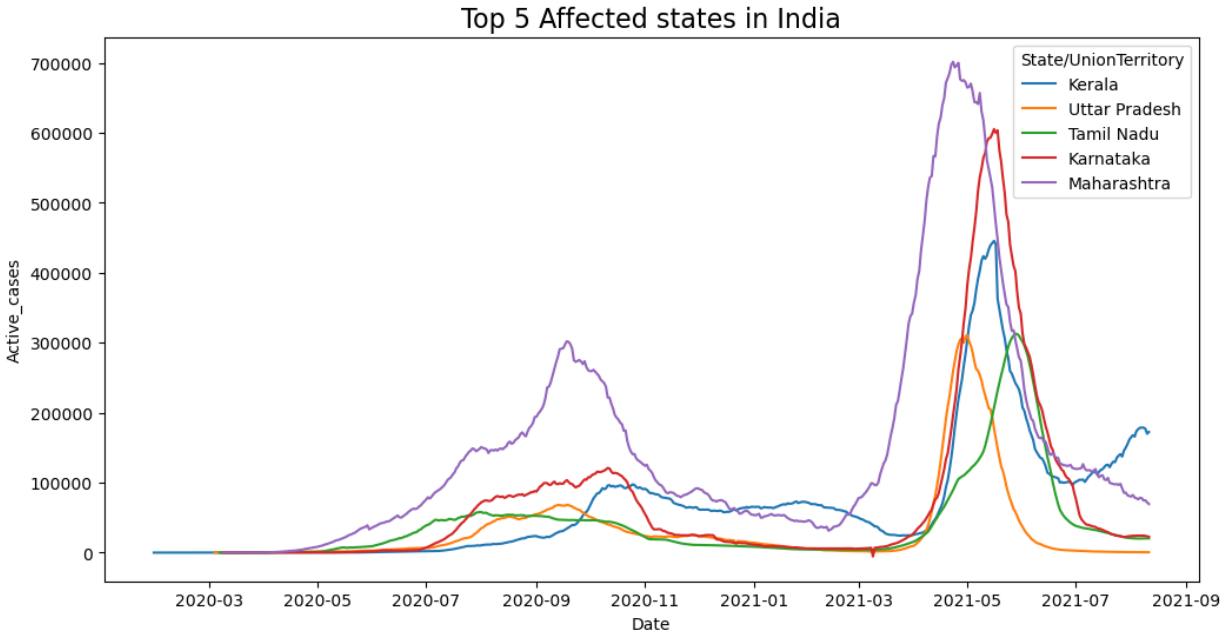
In [26]:
```
# Top states with Highest deaths

top_10_deaths = covid_df.groupby(by='State/UnionTerritory').max()[['Deaths', 'Date']].

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

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

ax = sns.barplot(data = top_10_deaths.iloc[:12], y = "Deaths", x = "State/UnionTerritc

plt.xlabel("states")
plt.ylabel("Total Death cases")
plt.show()
```



In [27]:
```
fig = plt.figure(figsize=(12, 6))

ax = sns.lineplot(data=covid_df[covid_df['State/UnionTerritory'].isin(['Maharashtra',
```

```
ax.set_title("Top 5 Affected states in India", size=16)
```

Out[27]:
```
Text(0.5, 1.0, 'Top 5 Affected states in India')
```



In [28]:
```
vaccine_df.head()
```

Out[28]:

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

5 rows × 24 columns

In [29]:
```
vaccine_df.rename(columns = {'updated on' : 'Vaccine_Date'}, inplace = True)
```

In [30]:
```
vaccine_df.head(10)
```

Out[30]:

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

10 rows × 24 columns

In [31]:
```
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
```

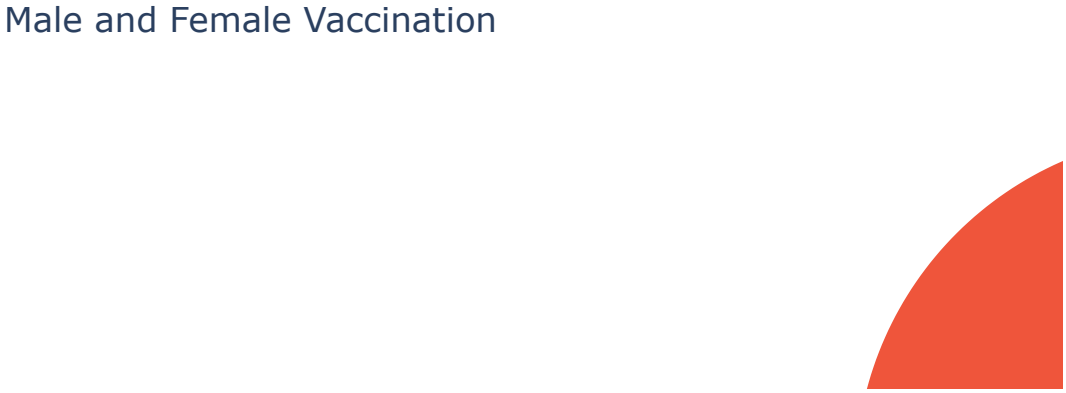
In [32]:
```
vaccination = vaccine_df.drop(columns = ['Sputnik V (Doses Administered)', 'AEFI', '18
```

In [33]:
```
vaccination.head()
```

Out[33]:

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

In [34]:
```
# Male vs Female Vaccination

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



In [35]:
```
#remove rows Where state = India

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

Out[35]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Dose Administered
212	16/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.(
213	17/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.(
214	18/01/2021	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0.0	29.(
215	19/01/2021	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0.0	53.(
216	20/01/2021	Andaman and Nicobar Islands	124.0	16.0	3.0	124.0	0.0	67.(
...	...	...	...	...	...	...	...	.
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 [36]:
```
vaccine.rename(columns = {"Total Individuals Vaccinated": 'Total'}, inplace = True)
vaccine.head()
```

```
C:\Users\nanav\AppData\Local\Temp\ipykernel_4448\3799401627.py:1: SettingWithCopyWarn
ing:


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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[36]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)
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

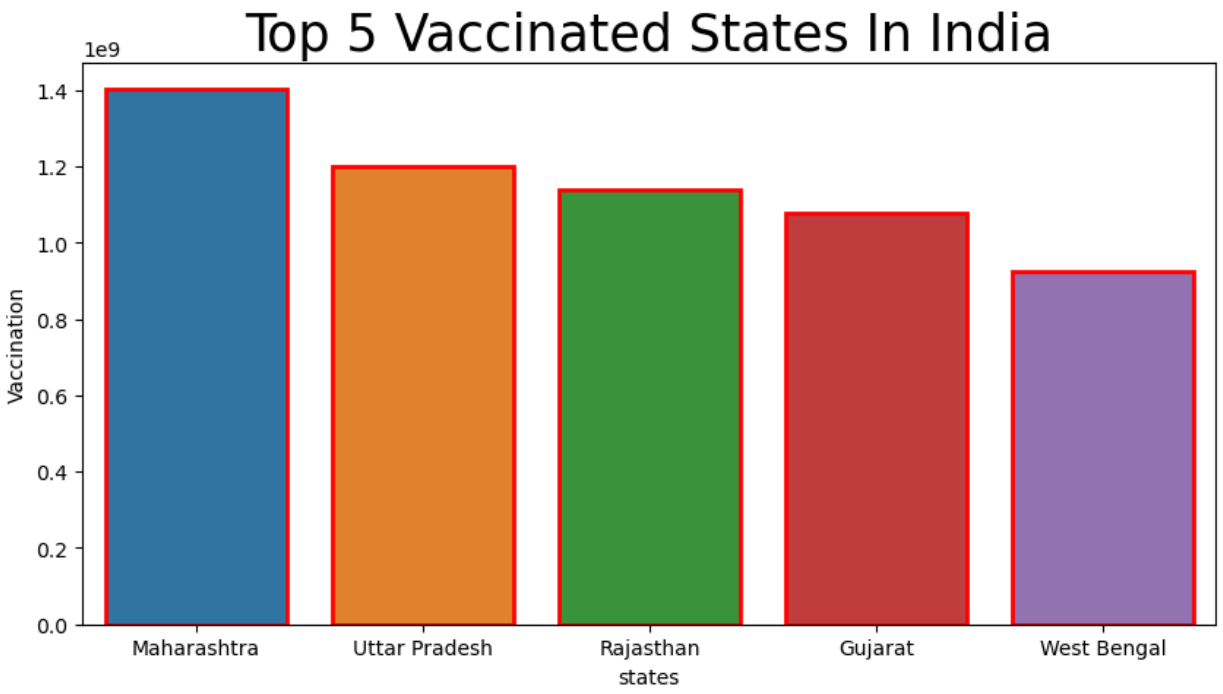
5 rows × 24 columns

In [37]:
```
# most vaccinated States

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

Out[37]:

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 [38]:
```
fig = plt.figure(figsize = (10, 5))

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

x = sns.barplot(data = max_vac.iloc[:10], y = max_vac.Total, x = max_vac.index, linewi

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



In [ ]:

In [ ]: