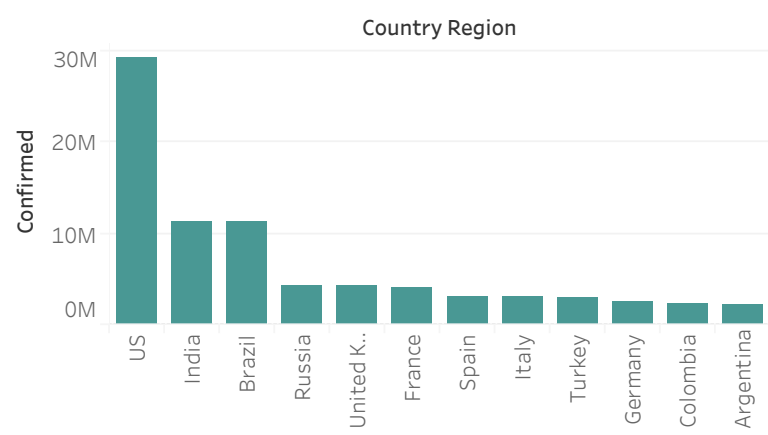
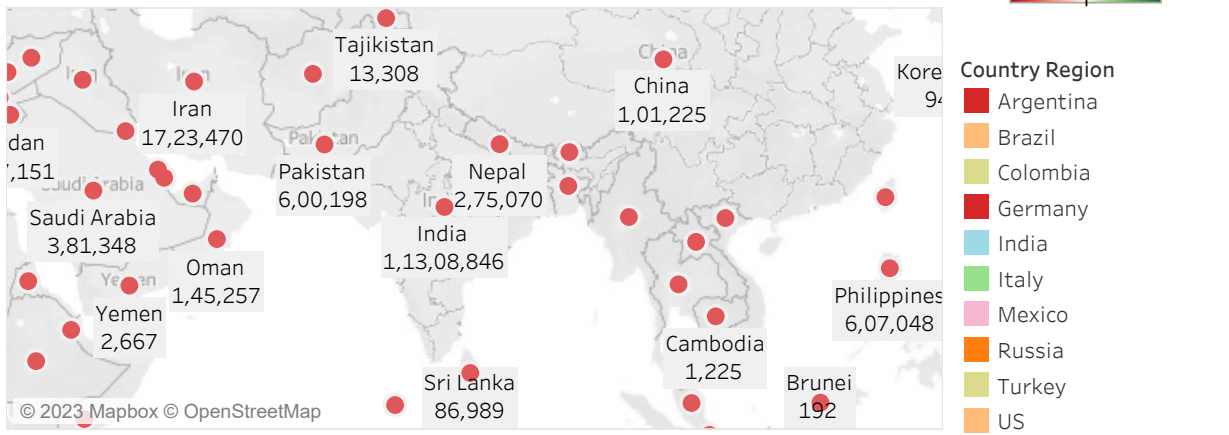


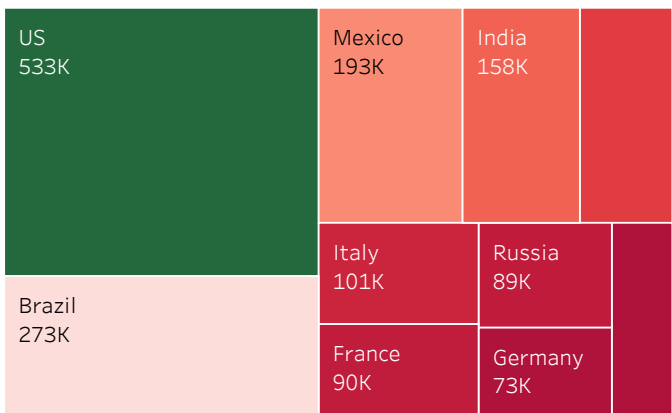
## Top 10 Confirmed Countries



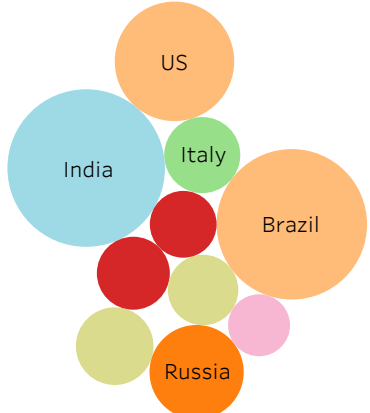
## Global Cases



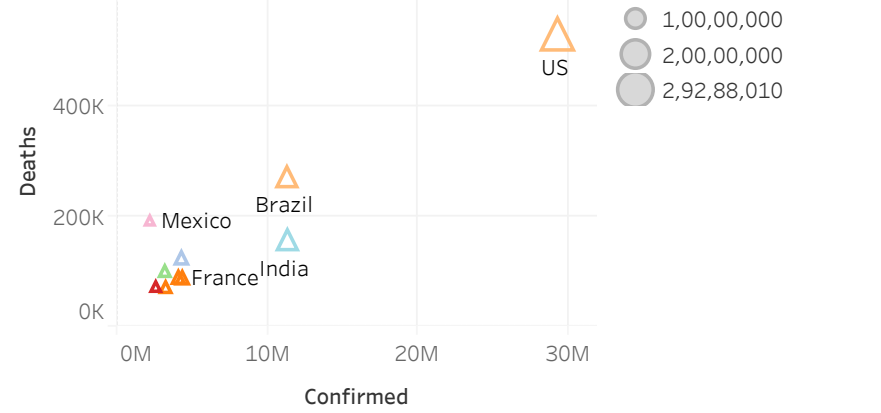
## Top 10 Deaths Cases



## Top 10 Recoveries



## Confirmed vs Deaths



```
In [2]: 1 import pandas as pd
        2 import matplotlib.pyplot as plt
        3 import seaborn as sns
        4 import plotly.express as px
```

```
In [4]: 1 covid_df = pd.read_csv("covid_19_india.csv")
        2 vaccine_df = pd.read_csv("covid_vaccine_statewise.csv")
```

```
In [5]: 1 covid_df.head()
```

Out[5]:

	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

```
In [6]: 1 covid_df.drop(['Sno', 'Time', 'ConfirmedIndianNational', 'ConfirmedForeignNational'], axis = 1, inplace = True)
```

```
In [7]: 1 covid_df.head()
```

Out[7]:

	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 [8]: 1 covid_df['Active_cases'] = covid_df['Confirmed'] - (covid_df['Cured'] + covid_df['Deaths'])
```

In [9]:

```
1 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  object
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: int64(4), object(2)
memory usage: 849.0+ KB
```

In [10]:

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

In [11]:

```
1 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 [13]:

```
1 #Create Pivot Table
2 statewise = pd.pivot_table(covid_df, values = ['Confirmed', 'Cured', 'Deaths'], index = 'State/UnionTerritory', aggfunc = m
```



In [15]:

```
1 statewise['Recovery Rate'] = statewise['Cured']*100 / statewise['Confirmed']
2 statewise['Mortality Rate'] = statewise['Deaths']*100 / statewise['Confirmed']
3 statewise = statewise.sort_values('Confirmed', ascending = False)
4 statewise.style.background_gradient(cmap = 'cubehelix')
```

Out[15]:

	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

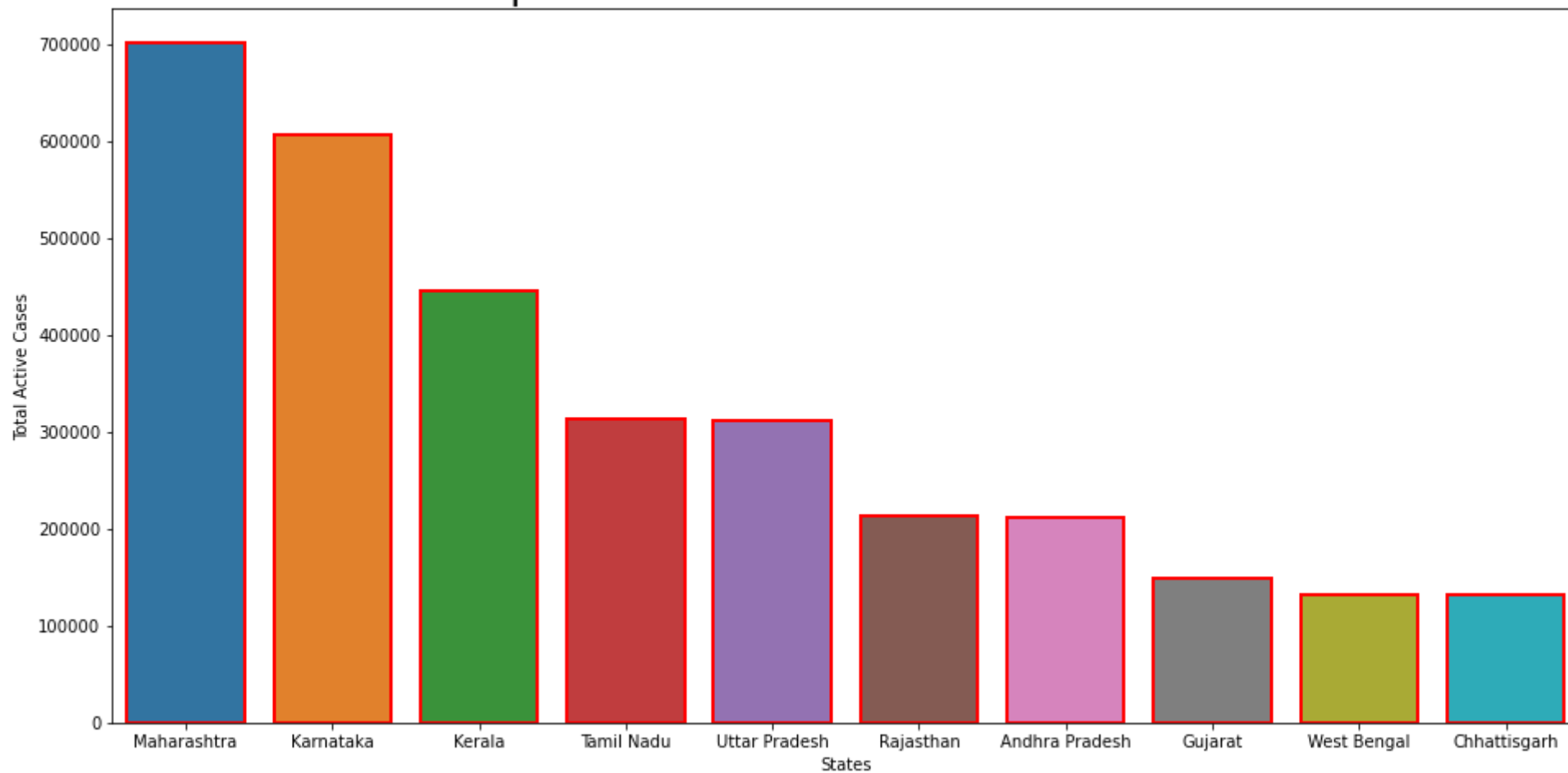
	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
State/UnionTerritory					
Himanchal Pradesh	204516	200040	3507	97.811418	1.714780
Goa	172085	167978	3164	97.613389	1.838626
Puducherry	121766	119115	1800	97.822873	1.478245
Manipur	105424	96776	1664	91.796934	1.578388
Tripura	80660	77811	773	96.467890	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 [22]: 1 #Top 10 states with most active cases
2 top_10_active_cases = covid_df.groupby('State/UnionTerritory')['Active_cases', 'Date'].max().sort_values('Active_cases', asc
3
4 fig = plt.figure(figsize = (16,8))
5 ax = sns.barplot(top_10_active_cases.iloc[:10], x = 'State/UnionTerritory', y = 'Active_cases', linewidth = 2, edgecolor = '
6 plt.title("Top 10 states with most active cases", size = 25)
7 plt.xlabel("States")
8 plt.ylabel("Total Active Cases")
9 plt.show()
```

C:\Users\NOBODY\AppData\Local\Temp\ipykernel\_8920\391276493.py:1: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
top_10_active_cases = covid_df.groupby('State/UnionTerritory')['Active_cases', 'Date'].max().sort_values('Active_cases', ascending = False).reset_index()
```

Top 10 states with most active cases

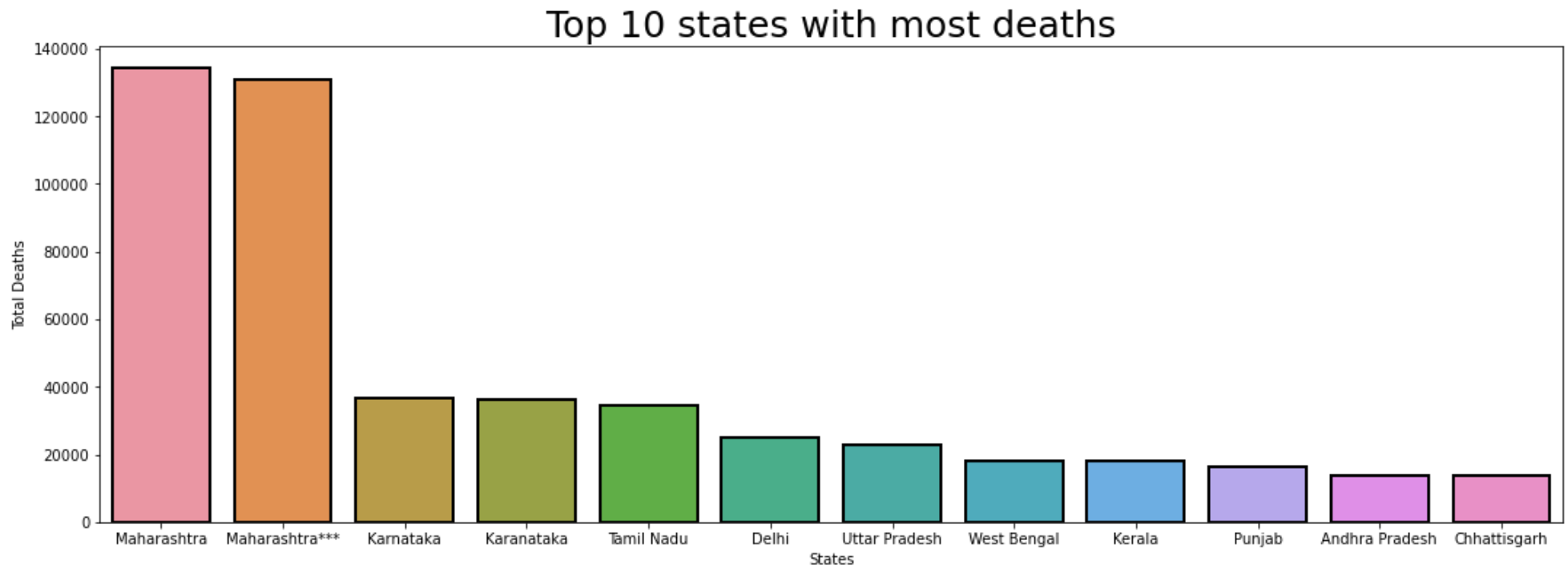




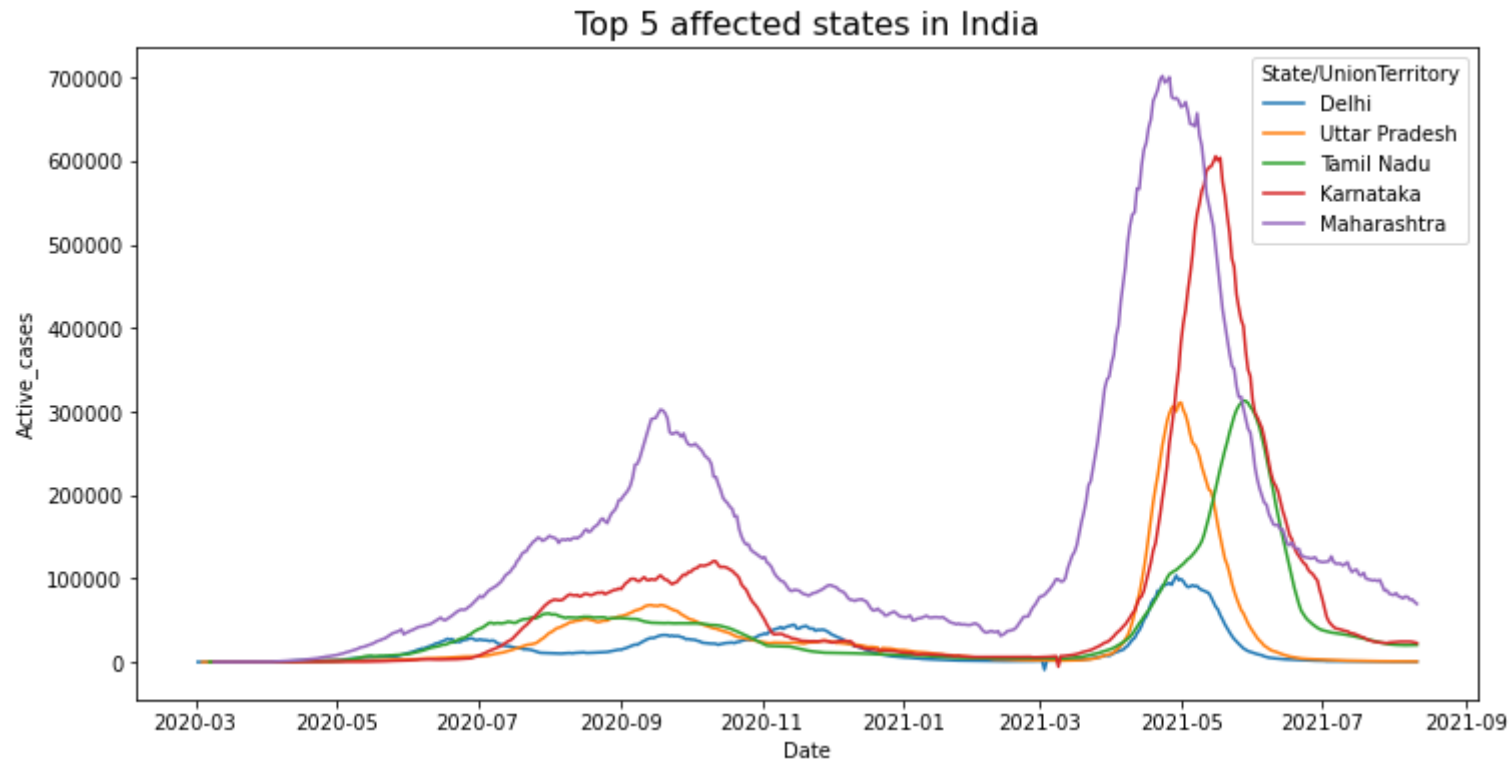
```
In [25]: 1 #Top 10 states with most deaths
2 top_10_deaths = covid_df.groupby('State/UnionTerritory')['Deaths', 'Date'].max().sort_values('Deaths', ascending = False).re
3
4 fig = plt.figure(figsize = (18,6))
5 ax = sns.barplot(top_10_deaths.iloc[:12], x = 'State/UnionTerritory', y = 'Deaths', linewidth = 2, edgecolor = 'black')
6 plt.title("Top 10 states with most deaths", size = 25)
7 plt.xlabel("States")
8 plt.ylabel("Total Deaths")
9 plt.show()
```

C:\Users\NOBODY\AppData\Local\Temp\ipykernel\_8920\186563100.py:2: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
top_10_deaths = covid_df.groupby('State/UnionTerritory')['Deaths', 'Date'].max().sort_values('Deaths', ascending = False).reset_index()
```



```
In [29]: 1 #Top 5 affected states in India
2 fig = plt.figure(figsize = (12,6))
3 ax = sns.lineplot(covid_df[covid_df['State/UnionTerritory'].isin(['Maharashtra', 'Karnataka', 'Tamil Nadu', 'Delhi', 'Uttar
4 plt.title("Top 5 affected states in India", size = 16)
5 plt.xlabel("Date")
6 plt.ylabel("Active_cases")
7 plt.show()
```

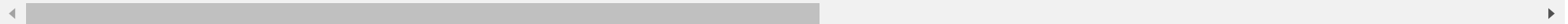


```
In [31]: 1 vaccine_df.head()
```

Out[31]:

	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)	45-60 Yea (Dose
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN	NaN	...	NaN	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN	NaN	...	NaN	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN	NaN	...	NaN	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN	NaN	...	NaN	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN	NaN	...	NaN	NaN

5 rows × 24 columns



```
In [33]: 1 #Removing Invalid State
2 vaccine_df = vaccine_df[vaccine_df['State'] != 'India']
```

In [34]:

1 vaccine\_df.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)	...	18-44 Years (Doses Administered)	45-60 (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	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	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	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	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	NaN

5 rows × 24 columns

In [36]:

1 vaccine = vaccine\_df[['Updated On', 'State', 'Male (Doses Administered)', 'Female (Doses Administered)', 'Total Individuals Administered']

In [38]:

```
1 vaccine.rename(columns = {'Updated On':'Date', 'Total Individuals Vaccinated':'Total'}, inplace = True)
2 vaccine.head()
```

C:\Users\NOBODY\AppData\Local\Temp\ipykernel\_8920\1612112090.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) ([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))

```
vaccine.rename(columns = {'Updated On':'Date', 'Total Individuals Vaccinated':'Total'}, inplace = True)
```

Out[38]:

	Date	State	Male (Doses Administered)	Female (Doses Administered)	Total
212	16/01/2021	Andaman and Nicobar Islands	12.0	11.0	23.0
213	17/01/2021	Andaman and Nicobar Islands	12.0	11.0	23.0
214	18/01/2021	Andaman and Nicobar Islands	29.0	13.0	42.0
215	19/01/2021	Andaman and Nicobar Islands	53.0	36.0	89.0
216	20/01/2021	Andaman and Nicobar Islands	67.0	57.0	124.0

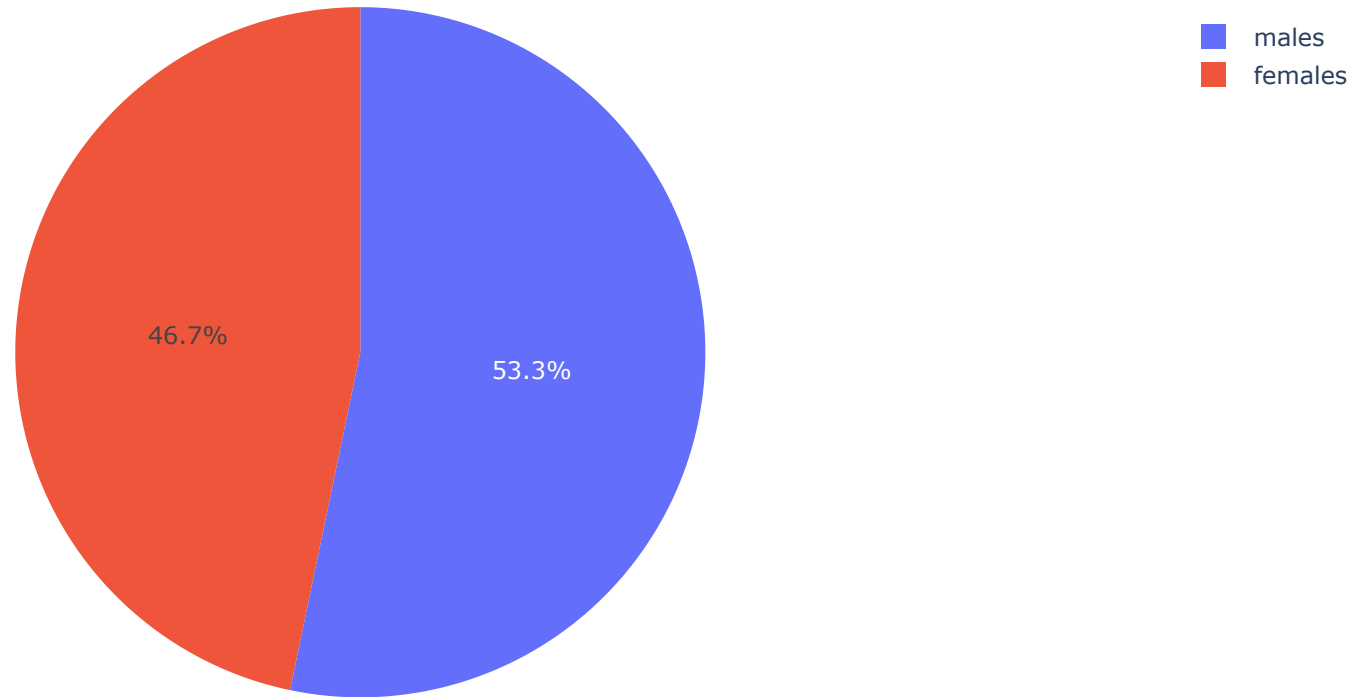
In [39]:

```
1 males = vaccine['Male (Doses Administered)'].sum()
2 females = vaccine['Female (Doses Administered)'].sum()
```

In [40]:

```
1 #Males vs Females vaccinated  
2 px.pie(names = ['males', 'females'], values = [males, females], title = 'Males vs Females vaccinated')
```

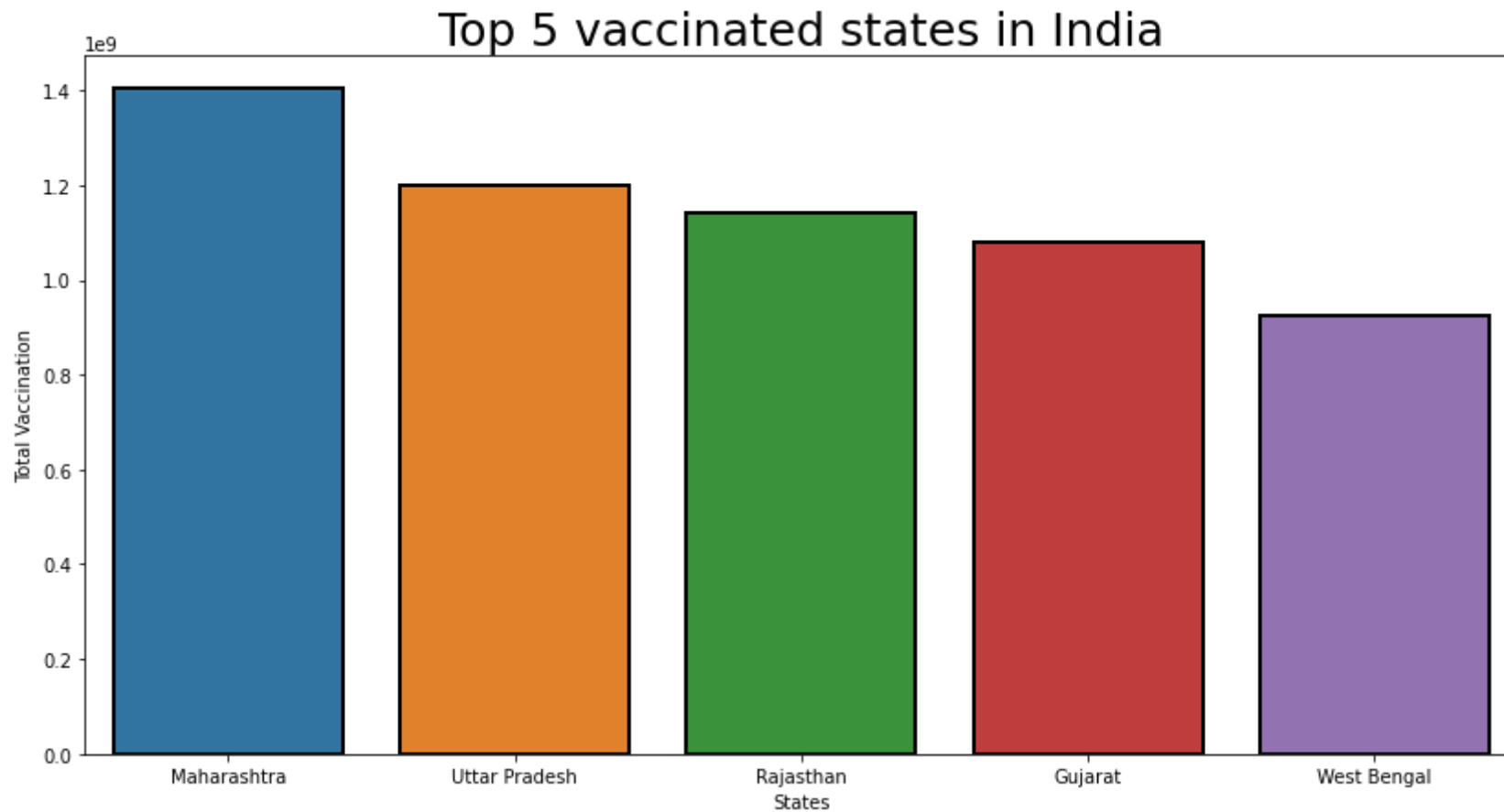
### Males vs Females vaccinated



```
In [47]: 1 #Top 5 vaccinated states in India
2 top_5_vac_states = vaccine.groupby('State')['Total', 'Date'].sum().sort_values('Total', ascending = False).reset_index()
3 fig = plt.figure(figsize = (14,7))
4 ax = sns.barplot(top_5_vac_states[:5], x = 'State', y = 'Total', linewidth = 2, edgecolor = 'black')
5 plt.title("Top 5 vaccinated states in India", size = 25)
6 plt.xlabel("States")
7 plt.ylabel("Total Vaccination")
8 plt.show()
```

C:\Users\NOBODY\AppData\Local\Temp\ipykernel\_8920\134765216.py:2: FutureWarning:

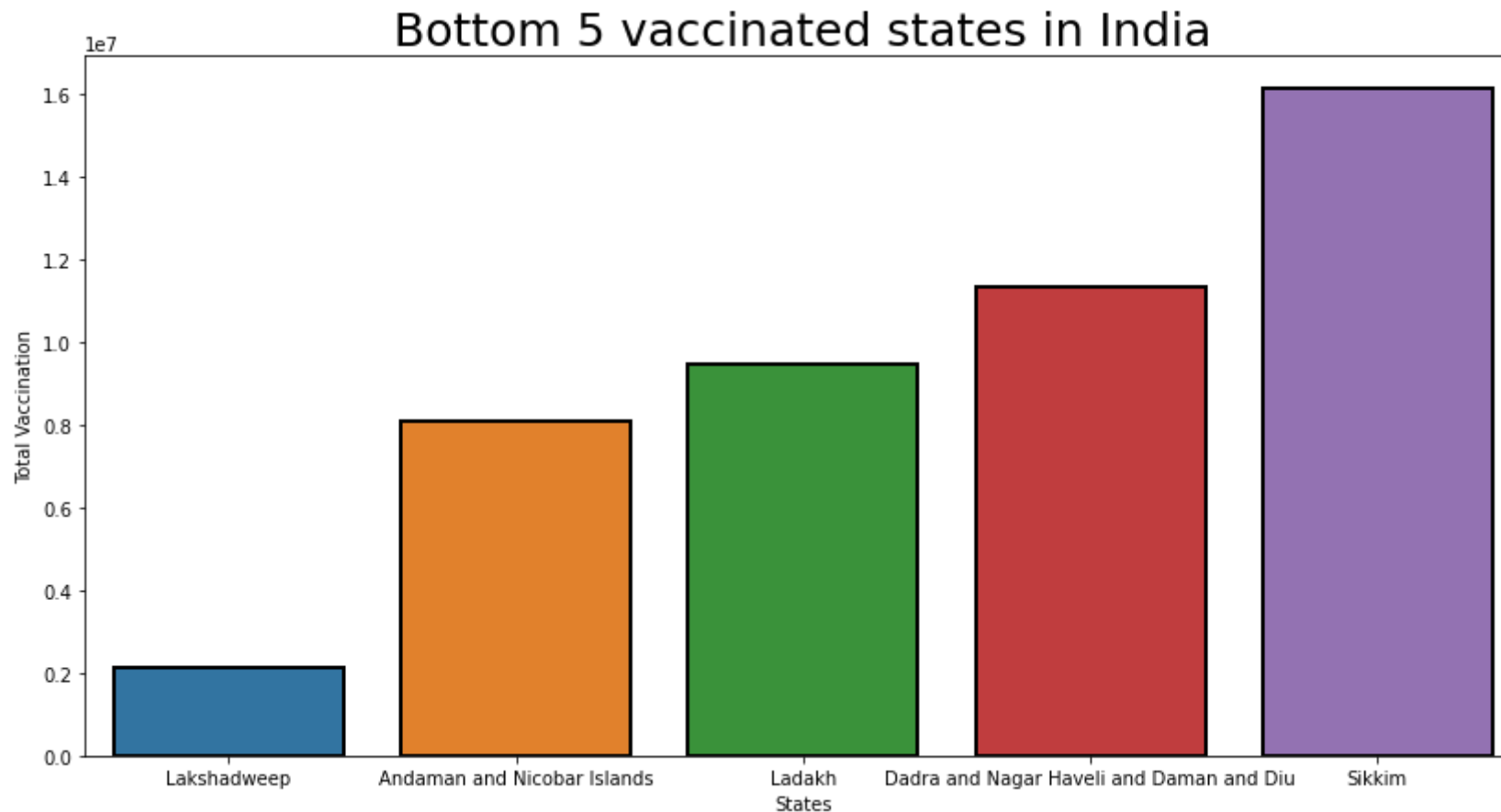
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.



```
In [48]: 1 #Bottom 5 vaccinated states in India
2 bottom_5_vac_states = vaccine.groupby('State')['Total', 'Date'].sum().sort_values('Total', ascending = True).reset_index()
3 fig = plt.figure(figsize = (14,7))
4 ax = sns.barplot(bottom_5_vac_states[:5], x = 'State', y = 'Total', linewidth = 2, edgecolor = 'black')
5 plt.title("Bottom 5 vaccinated states in India", size = 25)
6 plt.xlabel("States")
7 plt.ylabel("Total Vaccination")
8 plt.show()
```

C:\Users\NOBODY\AppData\Local\Temp\ipykernel\_8920\2113886462.py:2: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.



In [ ]: 1