

```
In [44]: 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]: covid_df = pd.read_csv("covid_19_india.csv")
```

```
In [5]: covid_df.head(10)
```

```
Out[5]:
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1.0	2020-01-30	6:00 PM	Kerala	1	0	0.0	0.0	1.0
1	2.0	2020-01-31	6:00 PM	Kerala	1	0	0.0	0.0	1.0
2	3.0	2020-02-01	6:00 PM	Kerala	2	0	0.0	0.0	2.0
3	4.0	2020-02-02	6:00 PM	Kerala	3	0	0.0	0.0	3.0
4	5.0	2020-02-03	6:00 PM	Kerala	3	0	0.0	0.0	3.0
5	6.0	2020-02-04	6:00 PM	Kerala	3	0	0.0	0.0	3.0
6	7.0	2020-02-05	6:00 PM	Kerala	3	0	0.0	0.0	3.0
7	8.0	2020-02-06	6:00 PM	Kerala	3	0	0.0	0.0	3.0
8	9.0	2020-02-07	6:00 PM	Kerala	3	0	0.0	0.0	3.0
9	10.0	2020-02-08	6:00 PM	Kerala	3	0	0.0	0.0	3.0

```
In [6]: covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15114 entries, 0 to 15113
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Sno                                    15086 non-null  float64
1   Date                                  15086 non-null  object
2   Time                                  15086 non-null  object
3   State/UnionTerritory                  15086 non-null  object
4   ConfirmedIndianNational               15086 non-null  object
5   ConfirmedForeignNational              15086 non-null  object
6   Cured                                 15086 non-null  float64
7   Deaths                                15086 non-null  float64
8   Confirmed                             15086 non-null  float64
dtypes: float64(4), object(5)
memory usage: 1.0+ MB
```

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

```
Out[7]:
```

	Sno	Cured	Deaths	Confirmed
count	15086.000000	1.508600e+04	15086.000000	1.508600e+04
mean	7543.500000	1.747937e+05	2721.084449	1.942820e+05
std	4355.097416	3.648330e+05	7182.672358	4.095184e+05
min	1.000000	0.000000e+00	0.000000	0.000000e+00
25%	3772.250000	1.685000e+03	12.000000	2.935500e+03
50%	7543.500000	1.964700e+04	364.000000	2.608150e+04
75%	11314.750000	2.087552e+05	2170.000000	2.216012e+05
max	15086.000000	4.927480e+06	83777.000000	5.433506e+06

```
In [8]: vaccine_df = pd.read_csv("covid_vaccine_statewise.csv")
```

```
In [9]: vaccine_df
```

Out[9]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN	NaN
...
7840	11/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7841	12/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7842	13/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7843	14/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7844	15/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7845 rows × 24 columns

◀		▶
---	--	---

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

```
In [273]: covid_df.head()
```

	Date	State/UnionTerritory	Cured	Deaths	Confirmed
0	2020-01-30	Kerala	0.0	0.0	1.0
1	2020-01-31	Kerala	0.0	0.0	1.0
2	2020-02-01	Kerala	0.0	0.0	2.0
3	2020-02-02	Kerala	0.0	0.0	3.0
4	2020-02-03	Kerala	0.0	0.0	3.0

```
In [11]: covid_df['Date'] = pd.to_datetime(covid_df['Date'], format = '%Y-%m-%d')
```

```
In [12]: # active cases
covid_df['Active_Cases'] = covid_df['Confirmed'] - (covid_df['Cured'] + covid_df['Deaths'])
covid_df.head()
```

	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active_Cases
0	2020-01-30	Kerala	0.0	0.0	1.0	1.0
1	2020-01-31	Kerala	0.0	0.0	1.0	1.0
2	2020-02-01	Kerala	0.0	0.0	2.0	2.0
3	2020-02-02	Kerala	0.0	0.0	3.0	3.0
4	2020-02-03	Kerala	0.0	0.0	3.0	3.0

```
In [13]: statewise = pd.pivot_table(covid_df, values = ["Confirmed", "Deaths", "Cured"], index = "State/UnionTerritory",
```

C:\Users\Dell\AppData\Local\Temp\ipykernel_1916\2370900951.py:1: FutureWarning: The provided callable <built-in function max> is currently using DataFrameGroupBy.max. In a future version of pandas, the provided callable will be used directly. To keep current behavior pass the string "max" instead.

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

```
In [14]: statewise["Recovery Rate"] = statewise["Cured"]*100/statewise["Confirmed"]
statewise["Mortality Rate"] = statewise["Deaths"]*100/statewise["Confirmed"]
```

```
In [15]: statewise = statewise.sort_values(by = "Confirmed", ascending = False)
```

```
In [16]: statewise.style.background_gradient(cmap = "cubehelix")
```

Out[16]:

	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
State/UnionTerritory					
Maharashtra	5433506.000000	4927480.000000	83777.000000	90.686934	1.541859
Karnataka	2272374.000000	1674487.000000	22838.000000	73.688882	1.005028
Kerala	2200706.000000	1846105.000000	6612.000000	83.886944	0.300449
Tamil Nadu	1664350.000000	1403052.000000	18369.000000	84.300297	1.103674
Uttar Pradesh	1637663.000000	1483249.000000	18072.000000	90.571076	1.103524
Andhra Pradesh	1475372.000000	1254291.000000	9580.000000	85.015237	0.649328
Delhi	1402873.000000	1329899.000000	22111.000000	94.798246	1.576123
West Bengal	1171861.000000	1026492.000000	13576.000000	87.595030	1.158499
Chhattisgarh	925531.000000	823113.000000	12036.000000	88.934136	1.300443
Rajasthan	879664.000000	713129.000000	7080.000000	81.068340	0.804853
Gujarat	766201.000000	660489.000000	9269.000000	86.203098	1.209735
Madhya Pradesh	742718.000000	652612.000000	7139.000000	87.868074	0.961199
Haryana	709689.000000	626852.000000	6923.000000	88.327704	0.975498
Bihar	664115.000000	595377.000000	4039.000000	89.649684	0.608178
Odisha	633302.000000	536595.000000	2357.000000	84.729718	0.372176
Telangana	536766.000000	485644.000000	3012.000000	90.475924	0.561138
Punjab	511652.000000	427058.000000	12317.000000	83.466497	2.407300
Telengana	443360.000000	362160.000000	2312.000000	81.685312	0.521472
Assam	340858.000000	290774.000000	2344.000000	85.306491	0.687676
Jharkhand	320934.000000	284805.000000	4601.000000	88.742545	1.433628
Uttarakhand	295790.000000	214426.000000	5132.000000	72.492647	1.735015
Jammu and Kashmir	251919.000000	197701.000000	3293.000000	78.478003	1.307166
Himachal Pradesh	166678.000000	129330.000000	2460.000000	77.592724	1.475900
Goa	138776.000000	112633.000000	2197.000000	81.161728	1.583127
Puducherry	87749.000000	69060.000000	1212.000000	78.701752	1.381212
Chandigarh	56513.000000	48831.000000	647.000000	86.406667	1.144869
Tripura	42776.000000	36402.000000	450.000000	85.099121	1.051992
Manipur	40683.000000	33466.000000	612.000000	82.260404	1.504314
Meghalaya	24872.000000	19185.000000	355.000000	77.134931	1.427308
Arunachal Pradesh	22462.000000	19977.000000	88.000000	88.936871	0.391773
Nagaland	18714.000000	14079.000000	228.000000	75.232446	1.218339
Ladakh	16784.000000	15031.000000	170.000000	89.555529	1.012869
Sikkim	11689.000000	8427.000000	212.000000	72.093421	1.813671
Dadra and Nagar Haveli and Daman and Diu	9652.000000	8944.000000	4.000000	92.664733	0.041442
Cases being reassigned to states	9265.000000	0.000000	0.000000	0.000000	0.000000
Mizoram	9252.000000	7094.000000	29.000000	76.675313	0.313446
Andaman and Nicobar Islands	6674.000000	6359.000000	92.000000	95.280192	1.378484
Lakshadweep	5212.000000	3915.000000	15.000000	75.115119	0.287797
Unassigned	77.000000	0.000000	0.000000	0.000000	0.000000
Daman & Diu	2.000000	0.000000	0.000000	0.000000	0.000000

```
In [18]: # top 10 active cases states
top_10_active_cases = covid_df.groupby(by = 'State/UnionTerritory').max()[['Active_Cases', 'Date']].sort_values
```

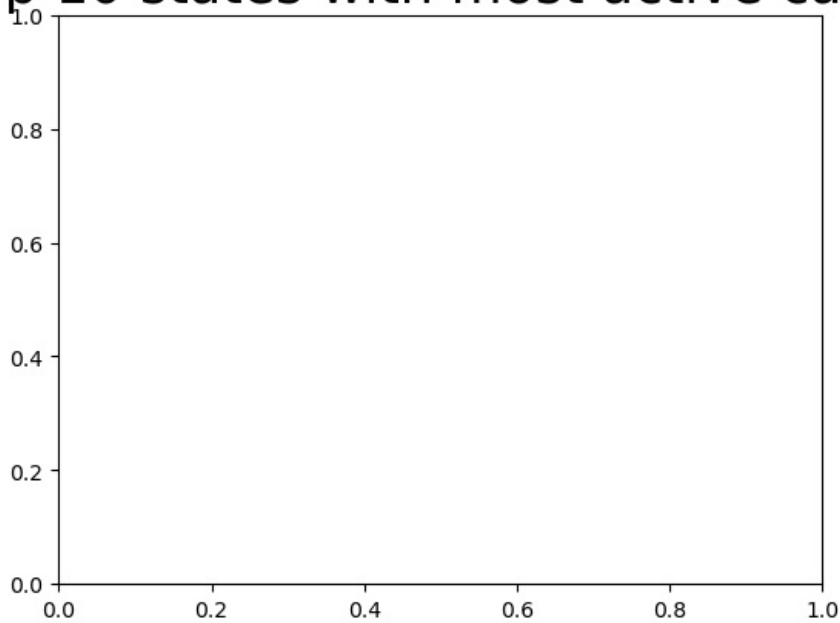
```
In [19]: fig = plt.figure(figsize=(16,9))

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

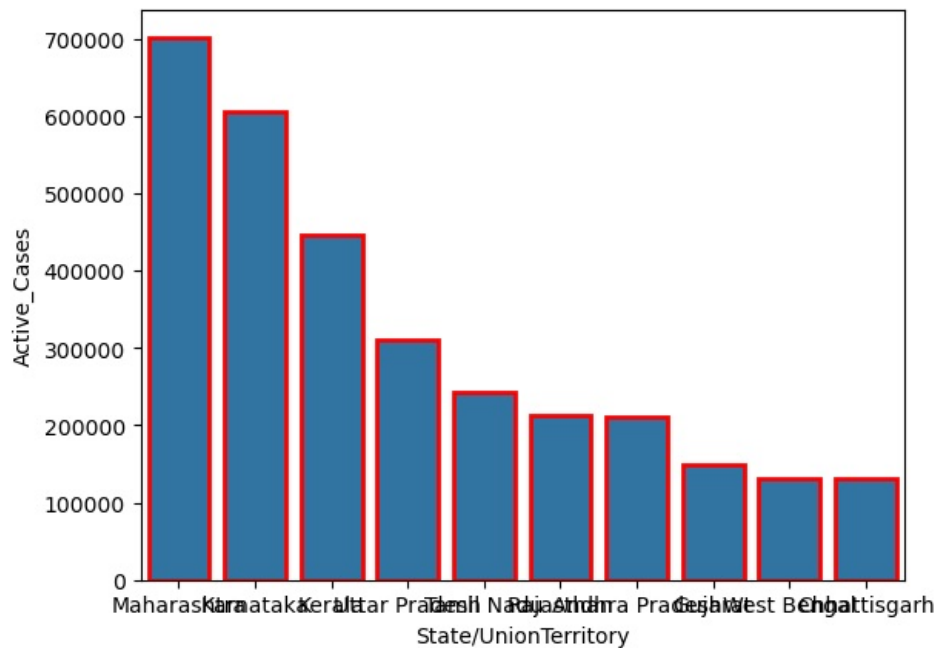
```
In [20]: plt.title("top 10 states with most active cases", size=25)
```

Out[20]: Text(0.5, 1.0, 'top 10 states with most active cases')

top 10 states with most active cases



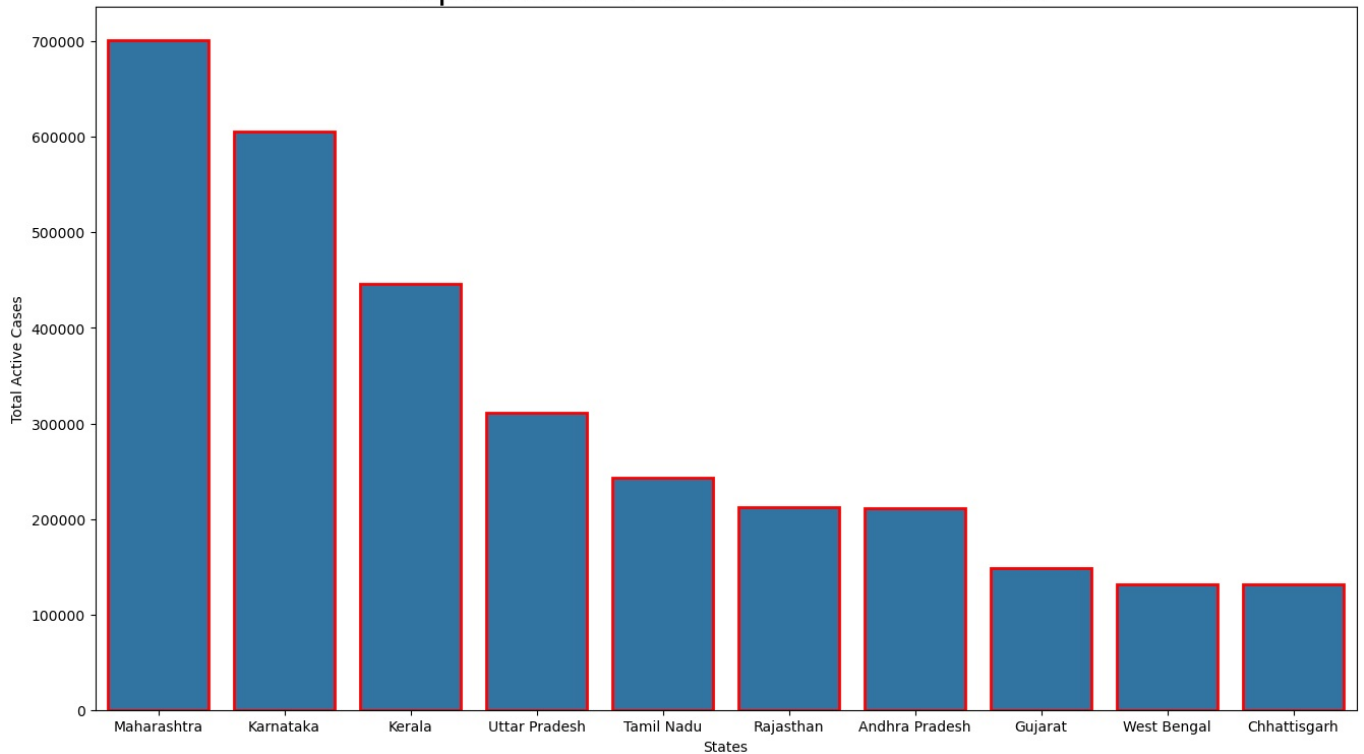
```
In [21]: ax = sns.barplot(data = top_10_active_cases.iloc[:10], y = "Active_Cases", x = "State/UnionTerritory", linewidth=
```



```
In [25]: # top 10 active cases states
top_10_active_cases = covid_df.groupby(by = 'State/UnionTerritory').max()[['Active_Cases', 'Date']].sort_values
fig = plt.figure(figsize=(16,9))
ax = sns.barplot(data = top_10_active_cases.iloc[:10], y = "Active_Cases", x = "State/UnionTerritory", linewidth=
plt.title("top 10 states with most active cases", size=25)
plt.xlabel("States")
plt.ylabel("Total Active Cases")
```

```
Out[25]: Text(0, 0.5, 'Total Active Cases')
```

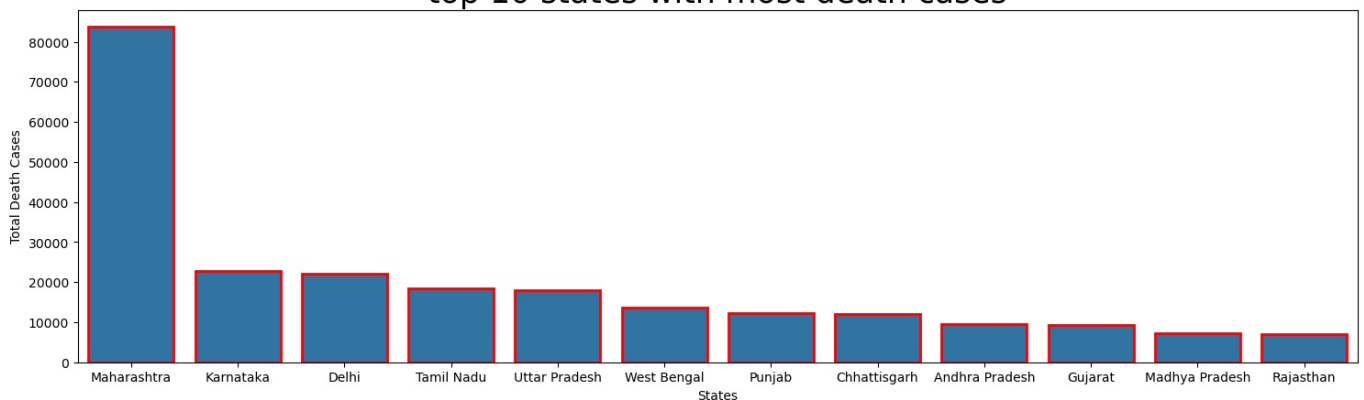
top 10 states with most active cases



```
In [23]: # Top states with highest Deaths
top_10_deaths = covid_df.groupby(by = 'State/UnionTerritory').max()['Deaths', 'Date'].sort_values(by = ['Deaths', 'Date'])
fig = plt.figure(figsize=(18,5))
ax = sns.barplot(data = top_10_deaths.iloc[:12], y = "Deaths", x = "State/UnionTerritory", linewidth = 2, edgecolor = 'red')
plt.title("top 10 states with most death cases", size=25)
plt.xlabel("States")
plt.ylabel("Total Death Cases")
```

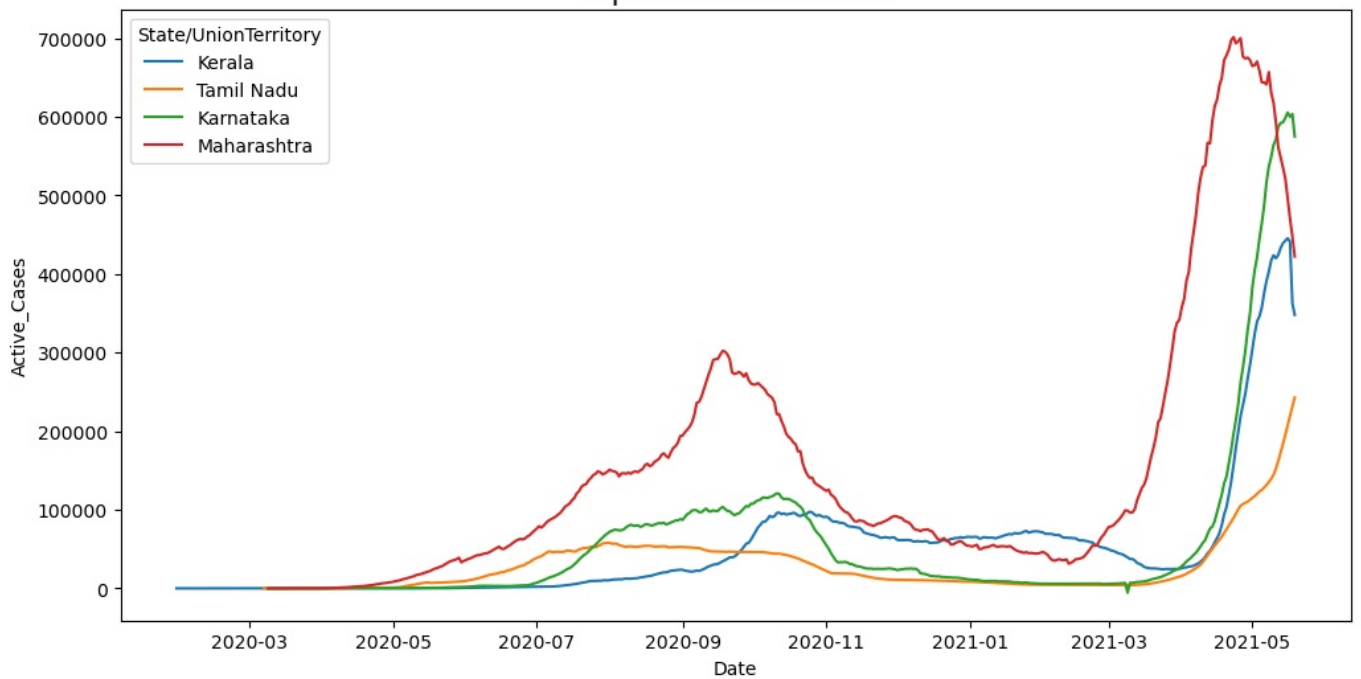
Out[23]: Text(0, 0.5, 'Total Death Cases')

top 10 states with most death cases



```
In [40]: fig = plt.figure(figsize=(12, 6))
ax = sns.lineplot(data=covid_df[covid_df['State/UnionTerritory'].isin(['Maharashtra', 'Karnataka', 'Kerala', 'Tamil Nadu', 'Uttar Pradesh', 'West Bengal', 'Punjab', 'Chhattisgarh', 'Andhra Pradesh', 'Gujarat', 'Madhya Pradesh', 'Rajasthan'])], x='Date', y='Active_Cases', hue='State/UnionTerritory')
ax.set_title("Top 5 affected States", size=16) # Comma added and correct keyword argument
plt.show()
```

Top 5 affected States



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

```
Out[28]:
```

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

	Vaccine_Date	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN	NaN
5	21/01/2021	India	365965.0	32226.0	12600.0	365965.0	0.0	NaN	NaN	NaN
6	22/01/2021	India	549381.0	36988.0	14115.0	549381.0	0.0	NaN	NaN	NaN
7	23/01/2021	India	759008.0	43076.0	15605.0	759008.0	0.0	NaN	NaN	NaN
8	24/01/2021	India	835058.0	49851.0	18111.0	835058.0	0.0	NaN	NaN	NaN
9	25/01/2021	India	1277104.0	55151.0	19682.0	1277104.0	0.0	NaN	NaN	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	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 [32]: vaccine_df.isnull().sum()
```

```
Out[32]: 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 [33]: vaccination = vaccine_df.drop(columns = ['Sputnik V (Doses Administered)', 'AEFI', '18-44 Years (Doses Administered)'])
```

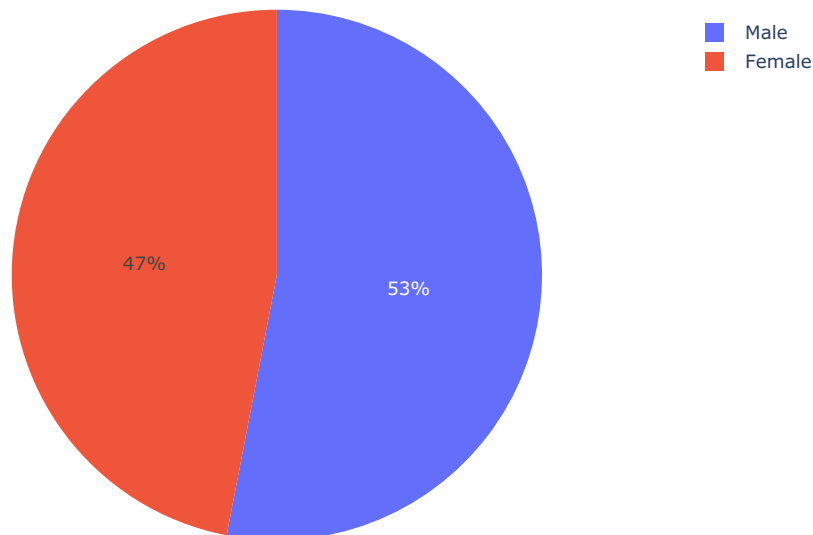
```
In [34]: vaccination.head()
```

```
Out[34]:
```

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

```
In [35]: # 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 Vaccination")
```

Male and Female Vaccination



```
In [36]: # remove rows where state= india
vaccine = vaccine_df[vaccine_df.State!='India']
```

```
In [37]: vaccine.rename(columns = {"Total Individuals Vaccinated": "Total"}, inplace = True)
```

C:\Users\Dell\AppData\Local\Temp\ipykernel_1916\2139085729.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

```
In [38]: vaccine.rename(columns = {"Total Individuals Vaccinated": "Total"}, inplace = True)
```

C:\Users\Dell\AppData\Local\Temp\ipykernel_1916\2139085729.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

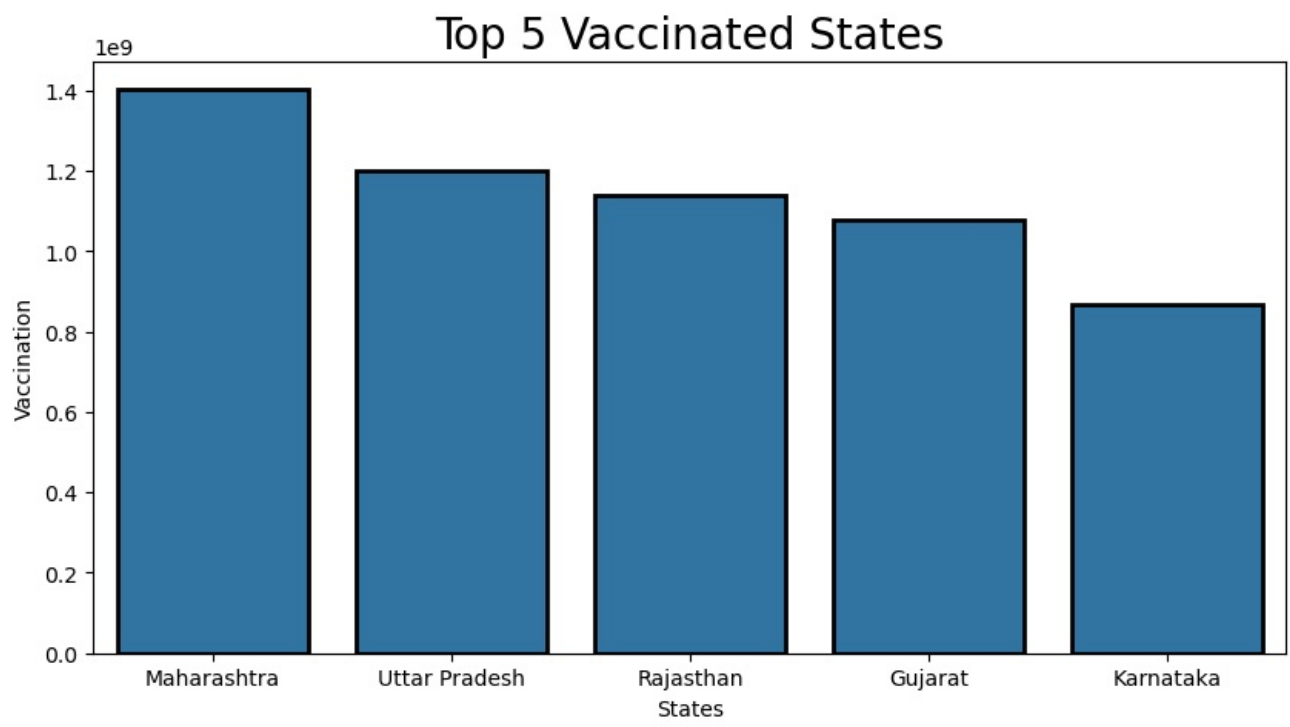
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
In [39]: # 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.head()
```

Out[39]:

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

In []: