

▼ Data Analysis on covid-19 data set

```
#import all required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
from datetime import datetime
import warnings
warnings.filterwarnings("ignore")

#load the data set
covid_df=pd.read_csv("/content/drive/MyDrive/Python project/covid_19_india.csv")
```

```
#check first 5 records
covid_df.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignN
0	1	2020-01-30	6:00 PM	Kerala	1	
1	2	2020-01-31	6:00 PM	Kerala	1	
2	3	2020-02-01	6:00 PM	Kerala	2	

```
#drop unwanted columns from data set
covid_df.drop(["Sno","Time","ConfirmedIndianNational","ConfirmedForeignNational"],inplace=
```

```
covid_df.head()
```

	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

```
#check data types of all the columns
covid_df.dtypes
```

```

Date                object
State/UnionTerritory object
Cured               int64
Deaths              int64
Confirmed            int64
dtype: object

```

```

#check null values in data set column
covid_df.isnull().sum()

```

```

Date                0
State/UnionTerritory 0
Cured               0
Deaths              0
Confirmed            0
dtype: int64

```

```

#check all states present in data set
covid_df["State/UnionTerritory"].unique()

```

```

array(['Kerala', 'Telengana', 'Delhi', 'Rajasthan', 'Uttar Pradesh',
       'Haryana', 'Ladakh', 'Tamil Nadu', 'Karnataka', 'Maharashtra',
       'Punjab', 'Jammu and Kashmir', 'Andhra Pradesh', 'Uttarakhand',
       'Odisha', 'Puducherry', 'West Bengal', 'Chhattisgarh',
       'Chandigarh', 'Gujarat', 'Himachal Pradesh', 'Madhya Pradesh',
       'Bihar', 'Manipur', 'Mizoram', 'Andaman and Nicobar Islands',
       'Goa', 'Unassigned', 'Assam', 'Jharkhand', 'Arunachal Pradesh',
       'Tripura', 'Nagaland', 'Meghalaya',
       'Dadra and Nagar Haveli and Daman and Diu',
       'Cases being reassigned to states', 'Sikkim', 'Daman & Diu',
       'Lakshadweep', 'Telangana', 'Dadra and Nagar Haveli', 'Bihar****',
       'Madhya Pradesh***', 'Himanchal Pradesh', 'Karnataka',
       'Maharashtra***'], dtype=object)

```

```

#counts the frequency of the state
covid_df["State/UnionTerritory"].value_counts()

```

```

Kerala                560
Delhi                 528
Rajasthan              527
Uttar Pradesh         526
Haryana                526
Ladakh                 523
Tamil Nadu             523
Punjab                 521
Jammu and Kashmir      521
Maharashtra            520
Karnataka              520
Andhra Pradesh         518
Uttarakhand            515
Odisha                 514
Puducherry             512
West Bengal            512
Chhattisgarh           511
Chandigarh             511
Gujarat                510

```

Himachal Pradesh	508
Madhya Pradesh	508
Bihar	506
Manipur	506
Mizoram	505
Andaman and Nicobar Islands	504
Goa	504
Assam	498
Jharkhand	498
Arunachal Pradesh	496
Tripura	492
Meghalaya	485
Dadra and Nagar Haveli and Daman and Diu	461
Nagaland	452
Sikkim	445
Telangana	426
Lakshadweep	244
Telangana	102
Cases being reassigned to states	60
Unassigned	3
Dadra and Nagar Haveli	2
Bihar****	2
Daman & Diu	1
Madhya Pradesh***	1
Himanchal Pradesh	1
Karnataka	1
Maharashtra***	1

Name: State/UnionTerritory, dtype: int64

```
#format the date column
```

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

```
#here in state column maharastra, Madhya pradesh and Bihar and all repeated value because o
covid_df.drop(index=covid_df[covid_df["State/UnionTerritory"]=="Maharashtra****"].index,inplace=
```

```
covid_df.drop(index=covid_df[covid_df["State/UnionTerritory"]=="Bihar****"].index,inplace=
```

```
covid_df.drop(index=covid_df[covid_df["State/UnionTerritory"]=="Madhya Pradesh***"].index,
```

```
covid_df.drop(index=covid_df[covid_df["State/UnionTerritory"]=="Dadra and Nagar Haveli"].i
```

```
covid_df.drop(index=covid_df[covid_df["State/UnionTerritory"]=="Daman & Diu"].index,inplace=
```

```
covid_df.drop(index=covid_df[covid_df["State/UnionTerritory"]=="Himanchal Pradesh"].index,
```

```
covid_df.drop(index=covid_df[covid_df["State/UnionTerritory"]=="Karnataka"].index,inplace=
```

```
#counts the frequency of the state
```

```
covid_df["State/UnionTerritory"].value_counts()
```

Kerala	560
--------	-----

Delhi	528
Rajasthan	527
Uttar Pradesh	526
Haryana	526
Ladakh	523
Tamil Nadu	523
Punjab	521
Jammu and Kashmir	521
Maharashtra	520
Karnataka	520
Andhra Pradesh	518
Uttarakhand	515
Odisha	514
Puducherry	512
West Bengal	512
Chandigarh	511
Chhattisgarh	511
Gujarat	510
Madhya Pradesh	508
Himachal Pradesh	508
Bihar	506
Manipur	506
Mizoram	505
Andaman and Nicobar Islands	504
Goa	504
Jharkhand	498
Assam	498
Arunachal Pradesh	496
Tripura	492
Meghalaya	485
Dadra and Nagar Haveli and Daman and Diu	461
Nagaland	452
Sikkim	445
Telangana	426
Lakshadweep	244
Telangana	102
Cases being reassigned to states	60
Unassigned	3

Name: State/UnionTerritory, dtype: int64

```
#check all states present in data set
covid_df["State/UnionTerritory"].unique()
```

```
array(['Kerala', 'Telengana', 'Delhi', 'Rajasthan', 'Uttar Pradesh',
      'Haryana', 'Ladakh', 'Tamil Nadu', 'Karnataka', 'Maharashtra',
      'Punjab', 'Jammu and Kashmir', 'Andhra Pradesh', 'Uttarakhand',
      'Odisha', 'Puducherry', 'West Bengal', 'Chhattisgarh',
      'Chandigarh', 'Gujarat', 'Himachal Pradesh', 'Madhya Pradesh',
      'Bihar', 'Manipur', 'Mizoram', 'Andaman and Nicobar Islands',
      'Goa', 'Unassigned', 'Assam', 'Jharkhand', 'Arunachal Pradesh',
      'Tripura', 'Nagaland', 'Meghalaya',
      'Dadra and Nagar Haveli and Daman and Diu',
      'Cases being reassigned to states', 'Sikkim', 'Lakshadweep',
      'Telangana'], dtype=object)
```

```
#show columns
covid_df.columns
```

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

```
#Add new column Active cases=Confirmed-Cured+Death
```

```
covid_df["Active_cases"]=covid_df["Confirmed"]-covid_df["Cured"]+covid_df['Deaths']
covid_df.tail()
```

	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases
18105	2021-08-11	Telangana	638410	3831	650353	15774
18106	2021-08-11	Tripura	77811	773	80660	3622
18107	2021-08-11	Uttarakhand	334650	7368	342462	15180
18108	2021-08-11	Uttar Pradesh	1685492	22775	1708812	46095
18109	2021-08-11	West Bengal	1506532	18252	1534999	46719

```
#information of the data set
```

```
covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18101 entries, 0 to 18109
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  18101 non-null  datetime64[ns]
1   State/UnionTerritory  18101 non-null  object
2   Cured                 18101 non-null  int64
3   Deaths               18101 non-null  int64
4   Confirmed             18101 non-null  int64
5   Active_cases          18101 non-null  int64
dtypes: datetime64[ns](1), int64(4), object(1)
memory usage: 989.9+ KB
```

```
#describe the dataset
```

```
covid_df.describe()
```

	Cured	Deaths	Confirmed	Active_cases	
count	1.810100e+04	18101.000000	1.810100e+04	18101.000000	
mean	2.781559e+05	4043.376222	3.005423e+05	26429.796475	
std	6.132422e+05	10878.106125	6.545130e+05	66861.654714	
min	0.000000e+00	0.000000	0.000000e+00	0.000000	
25%	3.357000e+03	32.000000	4.376000e+03	583.000000	
50%	3.335300e+04	588.000000	3.976300e+04	4954.000000	
75%	2.785230e+05	3643.000000	3.000110e+05	22955.000000	
max	6.159676e+06	134201.000000	6.363442e+06	829727.000000	



```
#create pivote table using pandas library it is sum of cured confirmed and death cases eac
statewise=pd.pivot_table(covid_df,values=["Confirmed","Deaths","Cured"],index="State/Union
```

```
#Recovery Rate statewise
statewise["Recovery Rate"]=statewise["Cured"]*100/statewise["Confirmed"]
```

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

```
#sort the value of confirmed cases acording statewise
statewise=statewise.sort_values(by="Confirmed",ascending=False)
```

```
#show top 8 records
statewise.head(8)
```

	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
State/UnionTerritory					
Maharashtra	6363442	6159676	134201	96.797865	2.108937
Kerala	3586693	3396184	18004	94.688450	0.501967
Karnataka	2921049	2861499	36848	97.961349	1.261465
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

```
#visualize the statewise records
statewise.style.background_gradient(cmap="cubehelix")
```

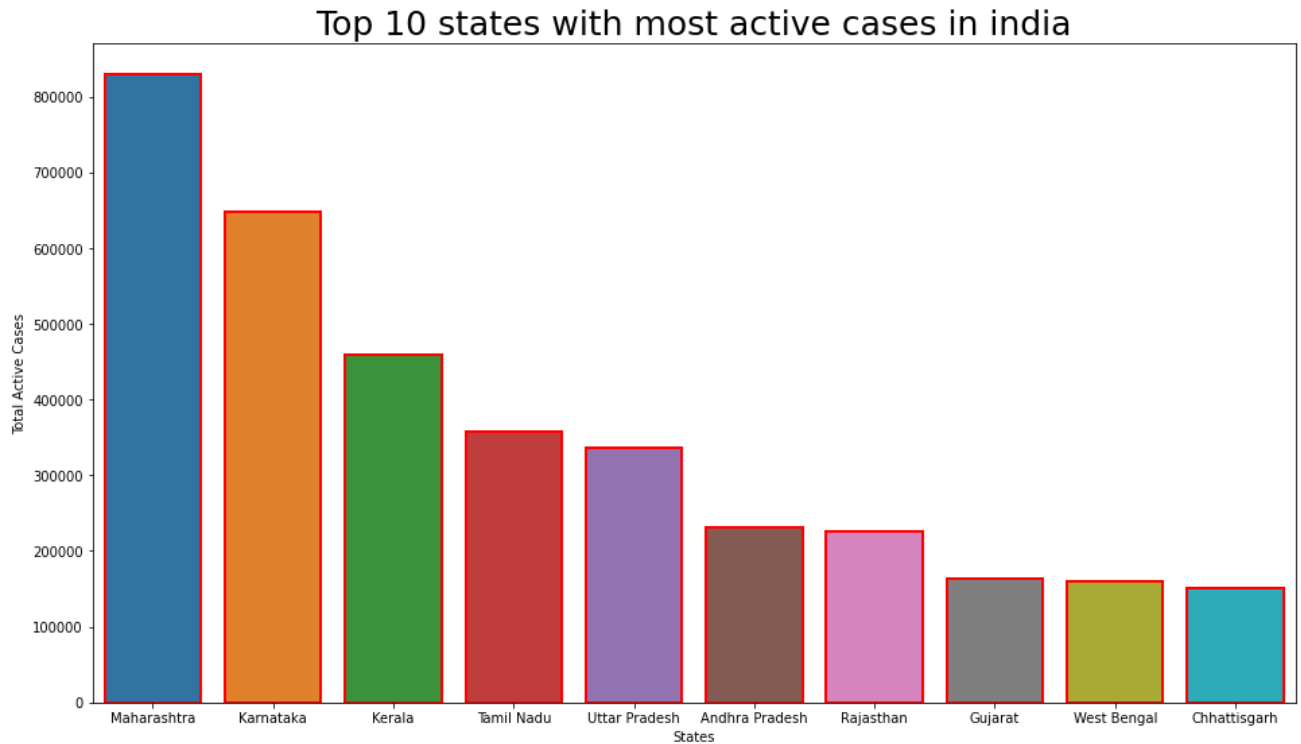
	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
State/UnionTerritory					
Maharashtra	6363442	6159676	134201	96.797865	2.108937
Kerala	3586693	3396184	18004	94.688450	0.501967
Karnataka	2921049	2861499	36848	97.961349	1.261465
Tamil Nadu	2579130	2524400	34367	97.877967	1.332504
Andhra Pradesh	1985182	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
Haryana	770114	759790	9652	98.659419	1.253321
Bihar	725279	715352	9646	98.631285	1.329971
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
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

#top 10 active cases by states

```

top_10_active_cases=covid_df.groupby(by="State/UnionTerritory").max()[["Active_cases","Date"]]
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")
plt.xlabel("States")
plt.ylabel("Total Active Cases")
plt.show()

```

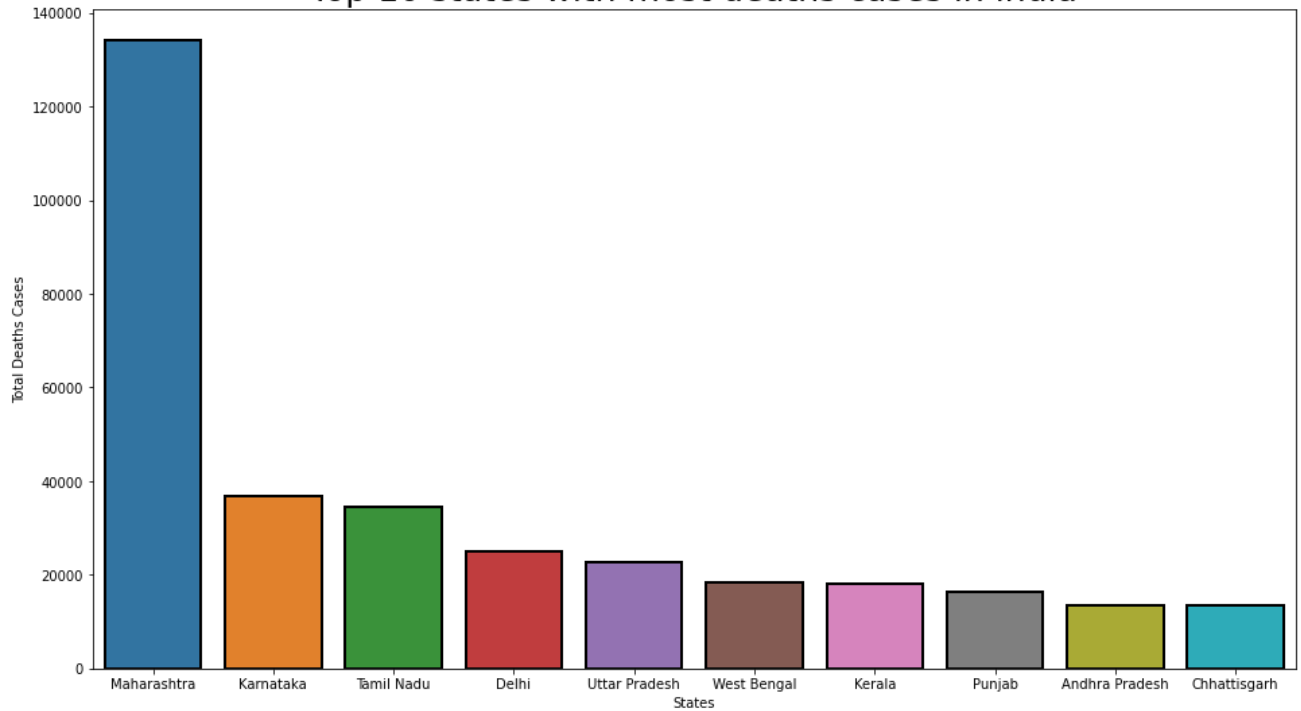


```

#to 10 state with highest death
top_10_deaths_states=covid_df.groupby(by="State/UnionTerritory").max()[["Deaths","Date"]].
fig=plt.figure(figsize=(16,9))
plt.title("Top 10 states with most deaths cases in india",size=25)
ax=sns.barplot(data=top_10_deaths_states.iloc[:10],y="Deaths",x="State/UnionTerritory",lin
plt.xlabel("States")
plt.ylabel("Total Deaths Cases")
plt.show()

```


Top 10 states with most deaths cases in india



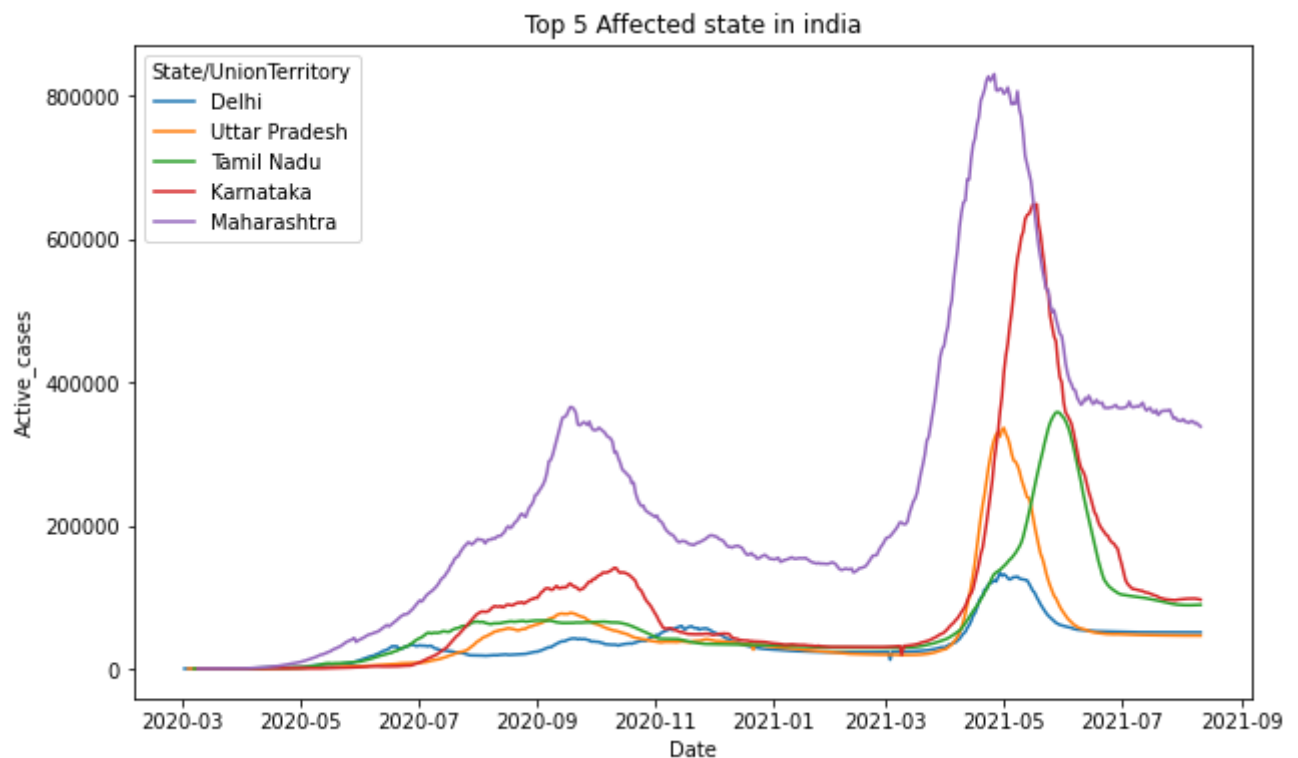
#Growth or the trend of active_cases to 5 state with most number of confirm cases

```
fig=plt.figure(figsize=(10,6))
```

```
ax=sns.lineplot(data = covid_df[covid_df["State/UnionTerritory"].isin(["Maharashtra","Karnataka","Tamil Nadu","Uttar Pradesh","Delhi"])]
```

```
ax.set_title("Top 5 Affected state in india",size=12)
```

```
Text(0.5, 1.0, 'Top 5 Affected state in india')
```



▼ Load Second Data Set covid_vaccine_statewise

```
vaccine_df=pd.read_csv("/content/drive/MyDrive/Python project/covid_vaccine_statewise.csv")
```

```
vaccine_df.head()
```

	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)	CoviShield (Doses Administered)	Sputnik V (Doses Administered)	AEFI	18-44 Years (Doses Administered)	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)	Total Individuals Vaccinated
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0																	
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0																	
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0																	
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0																	
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0																	

5 rows × 24 columns



```
#check size of the data set
vaccine_df.shape
```

```
(7845, 24)
```

```
#check columns present in data set
vaccine_df.columns
```

```
Index(['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)',
       'CoviShield (Doses Administered)', 'Sputnik V (Doses Administered)',
       'AEFI', '18-44 Years (Doses Administered)',
       '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)',
       'Total Individuals Vaccinated'],
      dtype='object')
```

```
#rename the updated on column to vaccine date
vaccine_df.rename(columns={"Updated On" : "Vaccine_Date"},inplace=True)
```

```
#Get information about data set
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

#chek null values present in each column

vaccine_df.isnull().sum()

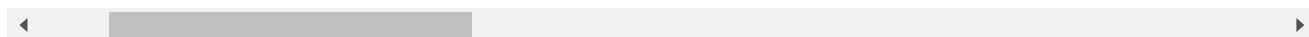
Vaccine_Date	0
State	0
Total Doses Administered	224
Sessions	224
Sites	224
First Dose Administered	224
Second Dose Administered	224
Male (Doses Administered)	384
Female (Doses Administered)	384
Transgender (Doses Administered)	384
Covaxin (Doses Administered)	224
CoviShield (Doses Administered)	224
Sputnik 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

```
#drop few unimportant columns of null value present in data set
vaccination=vaccine_df.drop(columns=["Sputnik V (Doses Administered)","AEFI","18-44 Years

vaccination.head()
```

State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)
India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	NaN
India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	NaN
India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	NaN
India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	NaN
India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	NaN



```
#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")
```

```
#drop all those rows wheren state= india
vaccine=vaccine_df[vaccine_df.State!="India"]
vaccine.head()
```

60+ Years (Doses Administered)	18-44 Years(Individuals Vaccinated)	45-60 Years(Individuals Vaccinated)	60+ Years(Individuals Vaccinated)	Male(Individ Vaccina
NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	

```
#rename the column total indidual vaccinated into total
vaccine.rename(columns={"Total Individuals Vaccinated":"Total"},inplace=True)

vaccine.head()
```

	Vaccine_Date	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered
212	16/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0
213	17/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0
214	18/01/2021	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0
215	19/01/2021	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0

```
#Most vaccinate state
```

```
max_vac=vaccine.groupby("State")["Total"].sum().to_frame("Total")
```

```
max_vac=max_vac.sort_values("Total",ascending=False)[:5]
```

```
max_vac
```

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

```
#visualize the top 5 vaccinated state
```

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

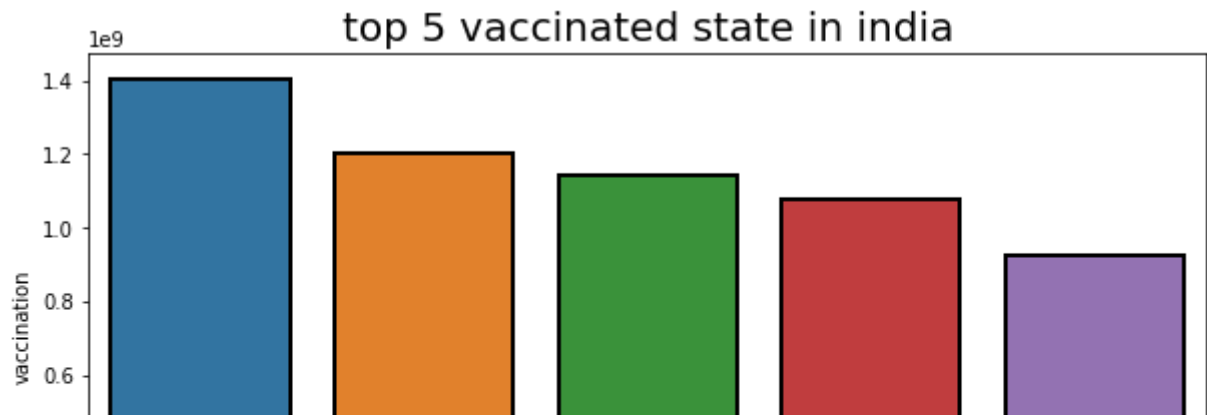
```
plt.title("top 5 vaccinated state in india",size=20)
```

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

```
plt.xlabel("States")
```

```
plt.ylabel("vaccination")
```

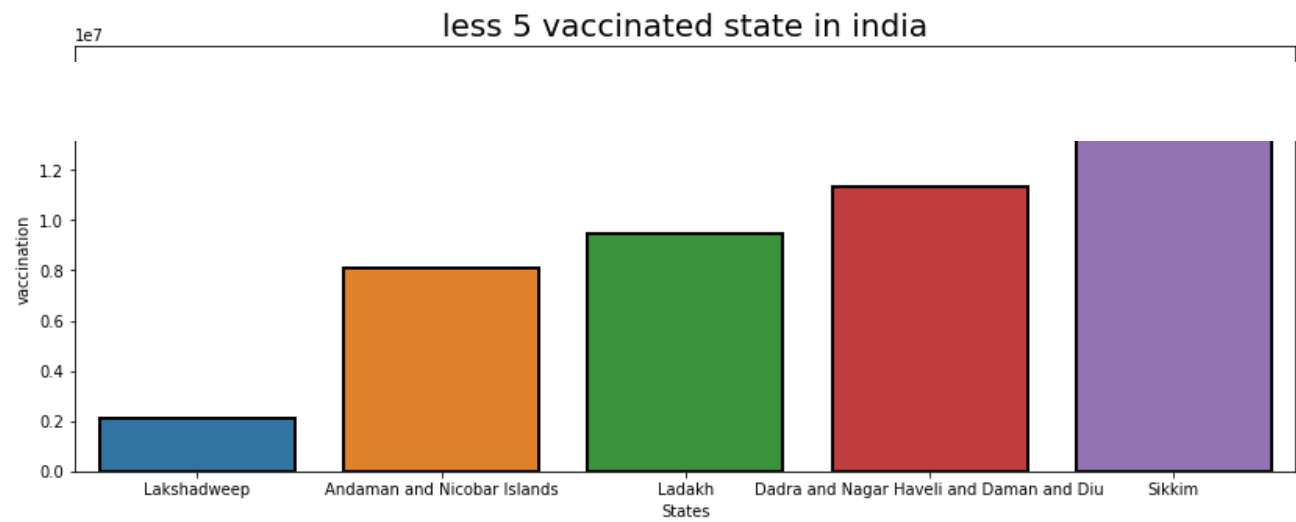
```
plt.show()
```



```
#less 5 vaccinate state
min_vac=vaccine.groupby("State")["Total"].sum().to_frame("Total")
min_vac=min_vac.sort_values("Total",ascending=False)[-1:-6:-1]
min_vac
```

	Total 
State	
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

```
#visualize less 5 vaccinated state
fig=plt.figure(figsize=(14,5))
plt.title("less 5 vaccinated state in india",size=20)
ax=sns.barplot(data=min_vac.iloc[:10],y=min_vac.Total,x=min_vac.index,linewidth=2,edgecolor='black')
plt.xlabel("States")
plt.ylabel("vaccination")
plt.show()
```



✓ 1s completed at 5:26 PM

