

COVID 19

CONTEXT:

The COVID-19 pandemic was caused by the emergence and global spread of the novel coronavirus, SARS-CoV-2. The virus likely originated in bats and might have been transmitted to humans through an intermediate host species, possibly at a seafood market in Wuhan, China, where the early cases were initially identified in late 2019. The exact origins are still under investigation. SARS-CoV-2 is primarily transmitted through respiratory droplets when an infected person coughs, sneezes, or talks. It can also spread by touching surfaces contaminated with the virus and then touching the face. The virus has a relatively high transmission rate, contributing to its rapid spread worldwide.

The global nature of modern travel facilitated the virus's swift international transmission, leading to a pandemic. As the virus spread, various countries implemented measures such as lockdowns, social distancing, and travel restrictions to contain its transmission. The lack of pre-existing immunity to the novel virus, combined with its ability to cause a range of symptoms from mild to severe, posed significant challenges for public health systems globally, contributing to the scale and impact of the pandemic.

ATTRIBUTE INFORMATION:

1: covid data frame(covid_df)

Date:

State/UnionTerritory: states in India

Cured: number of cured count

Deaths: number of deaths

Confirmed: number of deaths

2: vaccination data frame(vaccine_df)

State: states globally

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

Importing libraries:

```
In [14]: 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
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: covid_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_19_india (1).csv")
```

```
In [3]: covid_df.head()
```

```
Out[3]:
```

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

```
In [4]: vaccine_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_vaccine_statewise.csv")
```

```
In [5]: vaccine_df.head()
```

```
Out[5]:
```

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

```
In [7]: covid_df.columns
```

```
Out[7]: Index(['Date', 'State/UnionTerritory', 'Cured', 'Deaths', 'Confirmed'], dtype='object')
```

```
In [8]: vaccine_df.columns
```

```
Out[8]: 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')
```

```
In [9]: covid_df.describe()
```

```
Out[9]:      Cured    Deaths   Confirmed
count  1.508600e+04  15086.000000  1.508600e+04
mean   1.747937e+05  2721.084449  1.942820e+05
std    3.648330e+05  7182.672358  4.095184e+05
min    0.000000e+00  0.000000  0.000000e+00
25%   1.685000e+03  12.000000  2.935500e+03
50%   1.964700e+04  364.000000  2.608150e+04
75%   2.087552e+05  2170.000000  2.216012e+05
max   4.927480e+06  83777.000000  5.433506e+06
```

```
In [10]: vaccine_df.describe()
```

```
Out[10]:      Total Doses Administered    Sessions     Sites  First Dose Administered  Second Dose Administered  Male (Doses Administered)  Female (Doses Administered)  Transgender (Doses Administered)  Admir
count  7.621000e+03  7.621000e+03  7621.000000  7.621000e+03  7.621000e+03  7.461000e+03  7.461000e+03  7461.000000  7.621000e+03
mean   9.188171e+06  4.792358e+05  2282.872064  7.414415e+06  1.773755e+06  3.620156e+06  3.168416e+06  1162.978019  1.044000e+07
std    3.746180e+07  1.911511e+06  7275.973730  2.995209e+07  7.570382e+06  1.737938e+07  1.515310e+07  5931.353995  4.452000e+06
min    7.000000e+00  0.000000e+00  0.000000  7.000000e+00  0.000000e+00  0.000000e+00  2.000000e+00  0.000000  0.000000e+00
25%   1.356570e+05  6.004000e+03  69.000000  1.166320e+05  1.283100e+04  5.655500e+04  5.210700e+04  8.000000  0.000000e+00
50%   8.182020e+05  4.547000e+04  597.000000  6.614590e+05  1.388180e+05  3.897850e+05  3.342380e+05  113.000000  1.185000e+05
75%   6.625243e+06  3.428690e+05  1708.000000  5.387805e+06  1.166434e+06  2.735777e+06  2.561513e+06  800.000000  7.579000e+05
max   5.132284e+08  3.501031e+07  73933.000000  4.001504e+08  1.130780e+08  2.701636e+08  2.395186e+08  98275.000000  6.236000e+08
```

8 rows × 22 columns

```
In [11]: covid_df.describe(include='object')
```

```
Out[11]:      Date  State/UnionTerritory
count        15086          15086
unique       476            40
top         2020-06-11        Kerala
freq         37             476
```

```
In [12]: vaccine_df.describe(include='object')
```

```
Out[12]:      Updated On  State
count        7845          7845
unique       213            37
top         16/01/2021        Delhi
freq         37             213
```

```
In [13]: covid_df.nunique()
```

```
Out[13]: Date                  476
State/UnionTerritory          40
Cured                   11520
Deaths                  5026
Confirmed                12073
dtype: int64
```

```
In [14]: vaccine_df.nunique()
```

```
Out[14]: Updated On          213  
State                  37  
Total Doses Administered    7376  
Sessions                6464  
Sites                  3044  
First Dose Administered     7367  
Second Dose Administered    6275  
Male (Doses Administered)   7170  
Female (Doses Administered) 7155  
Transgender (Doses Administered) 2117  
Covaxin (Doses Administered) 4353  
CoviShield (Doses Administered) 7375  
Sputnik V (Doses Administered) 1040  
AEFI                   1548  
18-44 Years (Doses Administered) 1694  
45-60 Years (Doses Administered) 1693  
60+ Years (Doses Administered) 1692  
18-44 Years(Individuals Vaccinated) 3696  
45-60 Years(Individuals Vaccinated) 3700  
60+ Years(Individuals Vaccinated) 3684  
Male(Individuals Vaccinated) 159  
Female(Individuals Vaccinated) 159  
Transgender(Individuals Vaccinated) 156  
Total Individuals Vaccinated 5676  
dtype: int64
```

```
In [15]: covid_df.isna().sum()
```

```
Out[15]: Date          28  
State/UnionTerritory  28  
Cured               28  
Deaths               28  
Confirmed            28  
dtype: int64
```

```
In [16]: vaccine_df.isna().sum()
```

```
Out[16]: Updated On          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 [18]: #Active cases  
covid_df['Active_cases'] = covid_df['Confirmed'] - (covid_df['Cured'] + covid_df['Deaths'])  
covid_df.head()
```

```
Out[18]:   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 [19]: statewise = pd.pivot_table(covid_df, values = ["Confirmed", "Deaths", "Cured"], index = "State/UnionTerritory",
```

```
In [20]: statewise["Recovery Rate"] = statewise["Cured"]*100/statewise["Confirmed"]
```

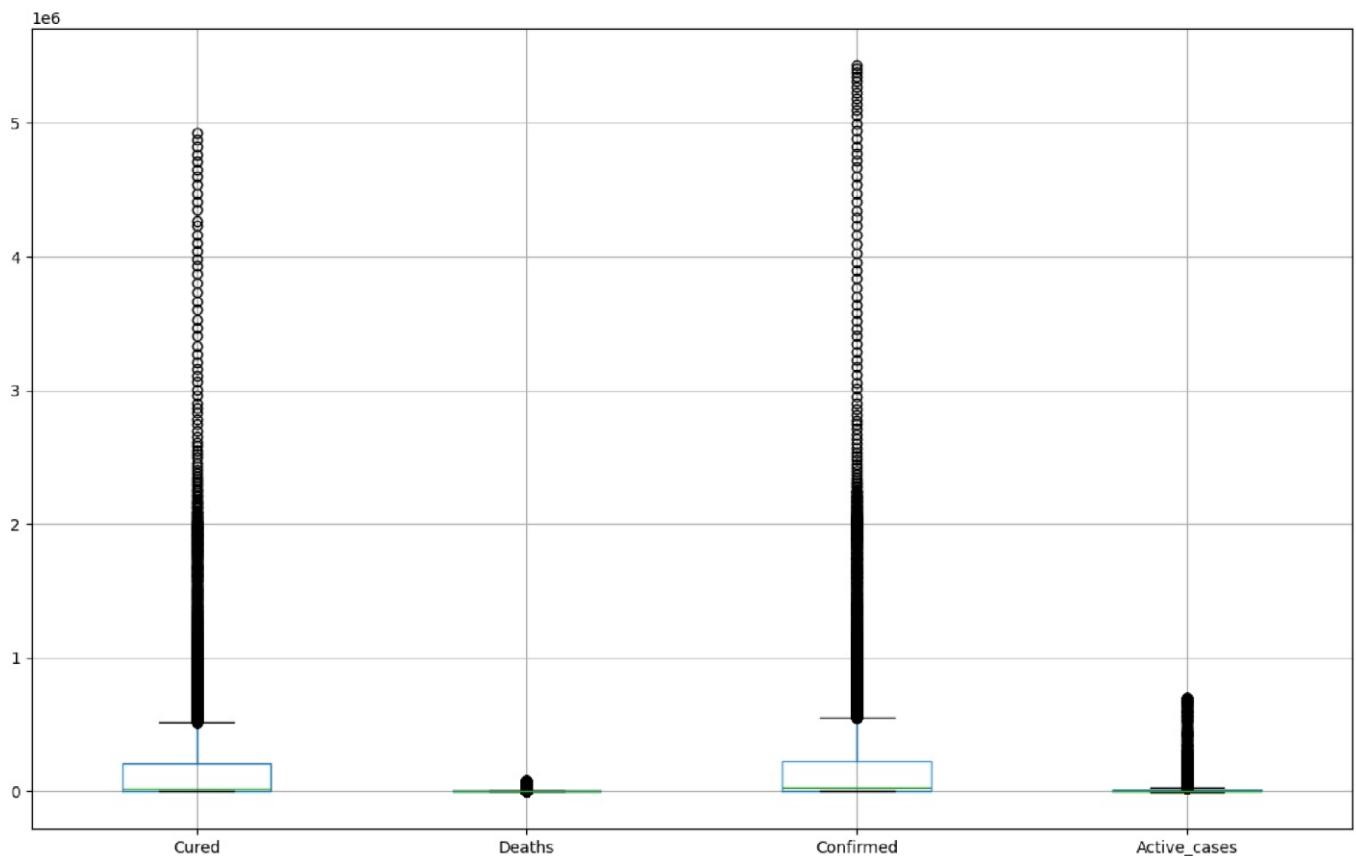
statewise["Mortality Rate"]	=	statewise["Deaths"]*100/statewise["Confirmed"]				
statewise	=	statewise.sort_values(by	=	"Confirmed",	ascending	=
		Maharashtra	5433506.000000	4927480.000000	83777.000000	90.686934
		statewise.style.background_gradient(cmap	=	"cubebehelix")		1.541859
		Karnataka	2272374.000000	1674487.000000	22838.000000	73.688882
		Kerala	2200706.000000	1846105.000000	6612.000000	83.886944
		Tamil Nadu	1664350.000000	1403052.000000	18369.000000	84.300297
		Uttar Pradesh	1637663.000000	1483249.000000	18072.000000	90.571076
		Andhra Pradesh	1475372.000000	1254291.000000	9580.000000	85.015237
		Delhi	1402873.000000	1329899.000000	22111.000000	94.798246
		West Bengal	1171861.000000	1026492.000000	13576.000000	87.595030
		Chhattisgarh	925531.000000	823113.000000	12036.000000	88.934136
		Rajasthan	879664.000000	713129.000000	7080.000000	81.068340
		Gujarat	766201.000000	660489.000000	9269.000000	86.203098
		Madhya Pradesh	742718.000000	652612.000000	7139.000000	87.868074
		Haryana	709689.000000	626852.000000	6923.000000	88.327704
		Bihar	664115.000000	595377.000000	4039.000000	89.649684
		Odisha	633302.000000	536595.000000	2357.000000	84.729718
		Telangana	536766.000000	485644.000000	3012.000000	90.475924
		Punjab	511652.000000	427058.000000	12317.000000	83.466497
		Telengana	443360.000000	362160.000000	2312.000000	81.685312
		Assam	340858.000000	290774.000000	2344.000000	85.306491
		Jharkhand	320934.000000	284805.000000	4601.000000	88.742545
		Uttarakhand	295790.000000	214426.000000	5132.000000	72.492647
		Jammu and Kashmir	251919.000000	197701.000000	3293.000000	78.478003
		Himachal Pradesh	166678.000000	129330.000000	2460.000000	77.592724
		Goa	138776.000000	112633.000000	2197.000000	81.161728
		Puducherry	87749.000000	69060.000000	1212.000000	78.701752
		Chandigarh	56513.000000	48831.000000	647.000000	86.406667
		Tripura	42776.000000	36402.000000	450.000000	85.099121
		Manipur	40683.000000	33466.000000	612.000000	82.260404
		Meghalaya	24872.000000	19185.000000	355.000000	77.134931
		Arunachal Pradesh	22462.000000	19977.000000	88.000000	88.936871
		Nagaland	18714.000000	14079.000000	228.000000	75.232446
		Ladakh	16784.000000	15031.000000	170.000000	89.555529
		Sikkim	11689.000000	8427.000000	212.000000	72.093421
Dadra and Nagar Haveli and Diu			9652.000000	8944.000000	4.000000	92.664733
Cases being reassigned to states			9265.000000	0.000000	0.000000	0.000000
		Mizoram				76.675313
						0.313446

Andaman and Nicobar Islands	Nicobar Islands	9252.000000	7094.000000	29.000000	95.280192	1.378484	In [21]:
	Lakshadweep	6674.000000	6359.000000	92.000000	75.115119	0.287797	In [22]:
	Unassigned	5212.000000	3915.000000	15.000000			In [23]:
Daman & Diu		77.000000 2.000000	0.000000	0.000000	0.000000	0.000000	Out[23]:
			Confirmed Mortality	Cured	Deaths	Recovery	Rate

State/UnionTerritory

In [24]:

Features BoxPlot

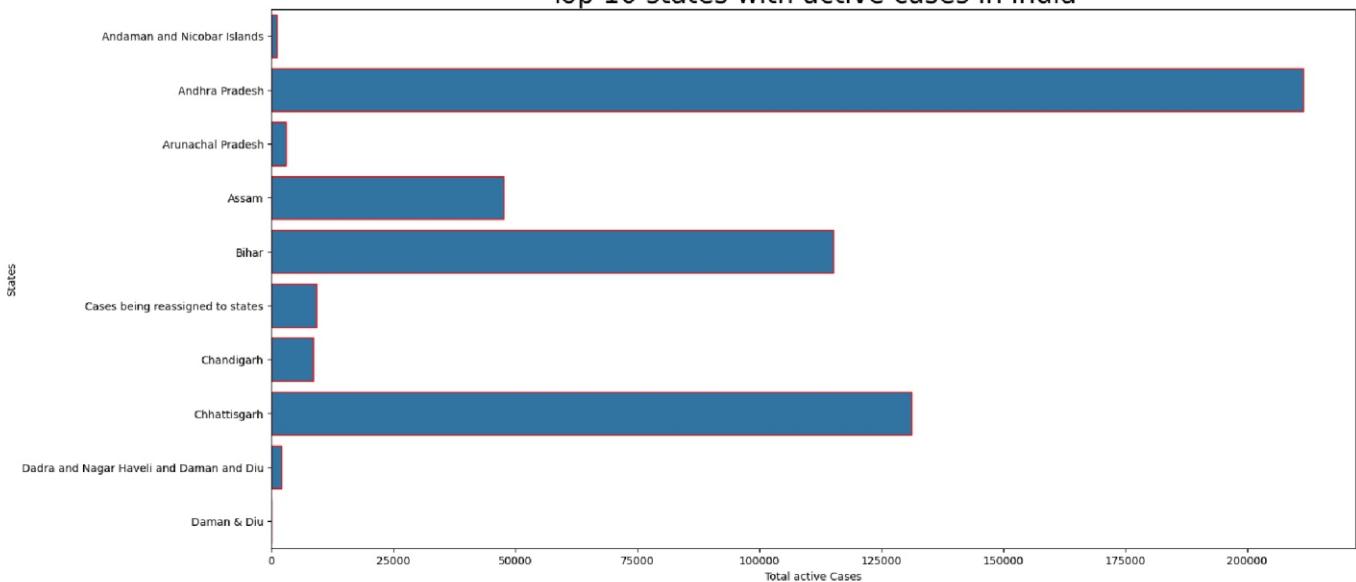


```
In [27]: #making boxplot for all features
vaccine_df.boxplot(figsize=(12,8))
#adding title
plt.suptitle('Features BoxPlot',x=0.5,y=1.02,ha='center',fontsize='large')
#adjust the plots gaps
plt.tight_layout()
```

```
In [28]: Top_10_active_cases = covid_df.groupby(by='State/UnionTerritory').max()[['Active_cases', 'Date']]
```

```
In [34]: Top_10_active_cases = covid_df.groupby(by='State/UnionTerritory').max()[['Active_cases', 'Date']]
fig = plt.figure(figsize=(18,9))
plt.title("Top 10 states with active cases in india", size = 25)
ax = sns.barplot(data = Top_10_active_cases.iloc[:10], y = "State/UnionTerritory", x = "Active_cases", linewidth=2)
plt.xlabel("Total active Cases")
plt.ylabel("States")
plt.show()
```

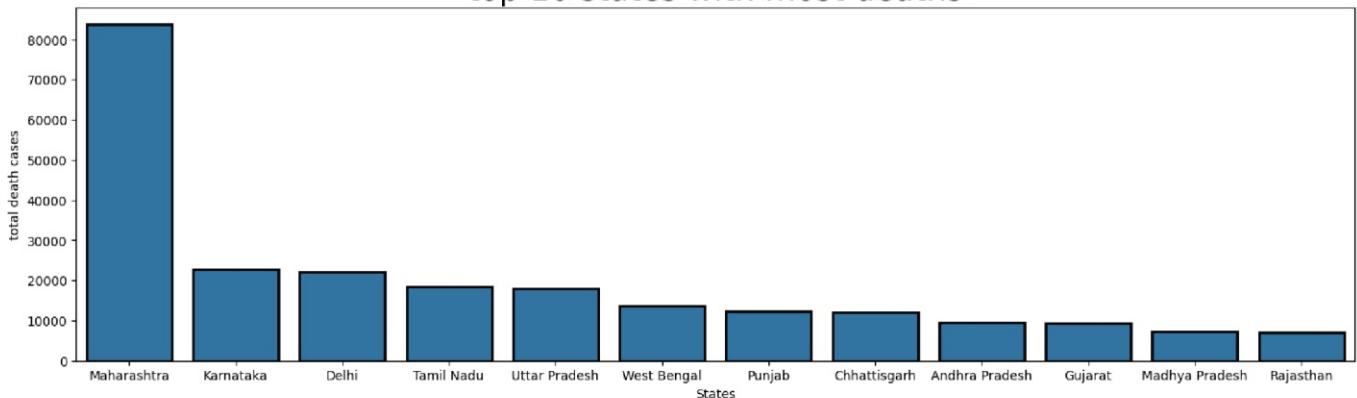
Top 10 states with active cases in india



```
In [35]: top_10_deaths = covid_df.groupby(by = 'State/UnionTerritory').max()[['Deaths','Date']].sort_values(by = ['Deaths'], ascending=False)
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/UnionTerritory", linewidth = 2, edgecolor="black")
```

```
plt.xlabel("States")
plt.ylabel("total death cases")
plt.show()
```

top 10 states with most deaths



```
In [36]: #growth trend
fig = plt.figure(figsize = (12,6))
ax = sns.lineplot(data = covid_df[covid_df['State/UnionTerritory'].isin(['Maharashtra', 'Karnataka', 'Kerala'])], ax.set_title("top 5 affected states in india", size=16)
```

```
Out[36]: Text(0.5, 1.0, 'top 5 affected states in india')
```

```
In [37]: vaccine_df.rename(columns = {'Updated On' : 'Vaccine_Date'}, inplace = True)
```

```
In [38]: vaccine_df.head()
```

```
Out[38]:
```

	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 rows × 11 columns

```
In [39]: vaccine_df.isnull().sum()
```

```
Out[39]:
```

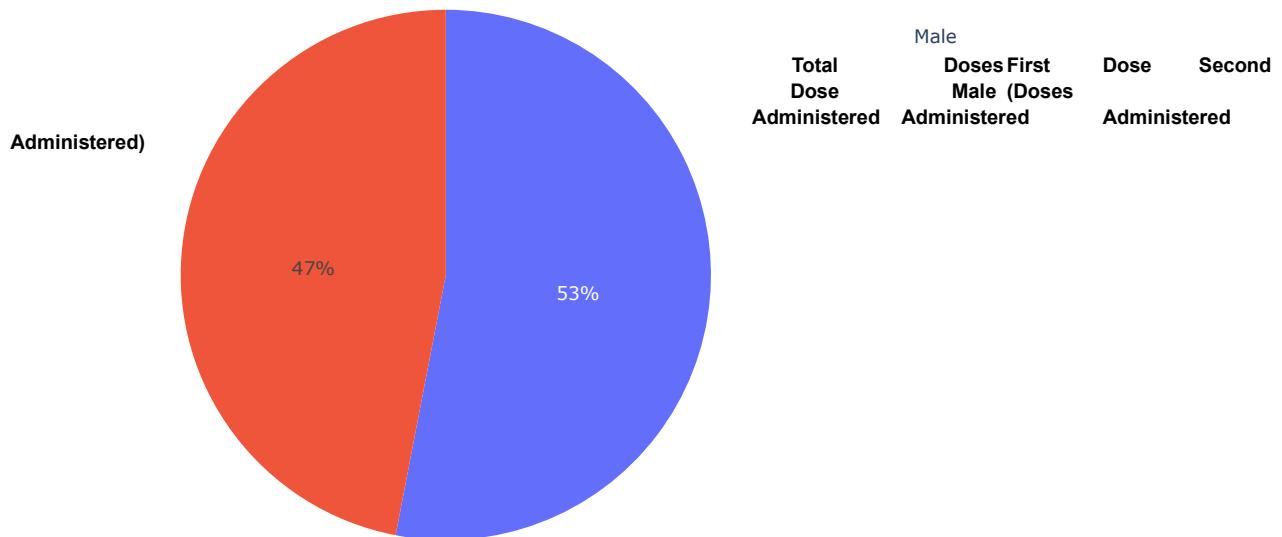
Vaccine_Date	0
State	0
Total Doses Administered	224
Sessions	224
Sites	224
First Dose Administered	224
Second Dose Administered	224
Male (Doses Administered)	384
Female (Doses Administered)	384
Transgender (Doses Administered)	384
Covaxin (Doses Administered)	224
CoviShield (Doses Administered)	224
Sputnik 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 [40]: vaccination = vaccine_df.drop(columns = ['Sputnik V (Doses Administered)', 'AEFI', '18-44 Years (Doses Administe
```

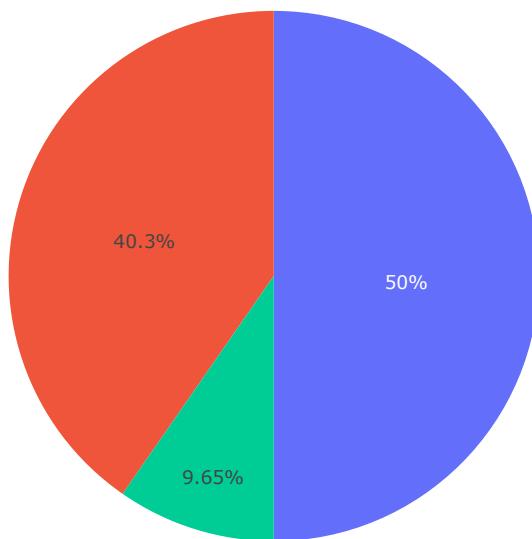
```
In [41]: # Male vs Female vaccination
```

```
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")
```



```
Import plotly.express as px
```

```
In [49]: #           "Total"      "Doses"      "Administered",      "First"      "Dose"      "Administered",      "Second"      "Dose"      "Administered"  
total=vaccination["Total Doses Administered"].sum() first_dose=vaccination["First Dose Administered"].sum() second_dose=vaccination["Second Dose Administered"].sum()  
  
fig=px.pie(names=["Total", "First Dose", "Second Dose"],  
values=[total, first_dose, second_dose],  
title="Total Number of Vaccination Doses Administered") fig.show()
```



```
In [50]:
```

```
[50]: cured = covid_df[\"Cured\"].sum()
deaths = covid_df[\"Deaths\"].sum()
confirmed = covid_df[\"Confirmed\"].sum()
active_cases = covid_df[\"Active_cases\"].sum()

fig = px.pie(names=[\"CURED\", \"DEATHS\", \"CONFIRMED\", \"ACTIVE_CASES\"],
              values=[cured, deaths, confirmed, active_cases],
              title=\"classification of cases\")

fig.show()
```

classification of cases



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

	Vaccine_Date	State	Sessions	Sites			Female (Doses Administered)	Transgender (Doses Administered)
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
...
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

```
In [54]: vaccine.rename(columns = {"Total Individuals Vaccinates": "Total"}, inplace = True)  
vaccine.head()
```

	Vaccine_Date	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Female (Doses Administered)	Transgender (Doses Administered)
212	16/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0	
213	17/01/2021	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	11.0	
214	18/01/2021	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0.0	29.0	13.0	
215	19/01/2021	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0.0	53.0	36.0	
216	20/01/2021	Andaman and Nicobar Islands	124.0	16.0	3.0	124.0	0.0	67.0	57.0	

5 rows × 24 columns

```
In [55]: max_vac = vaccine.groupby('State')['Total Individuals Vaccinated'].sum().to_frame('Total')  
max_vac = max_vac.sort_values('Total', ascending = False)[:5]  
max_vac
```

Out[55]:

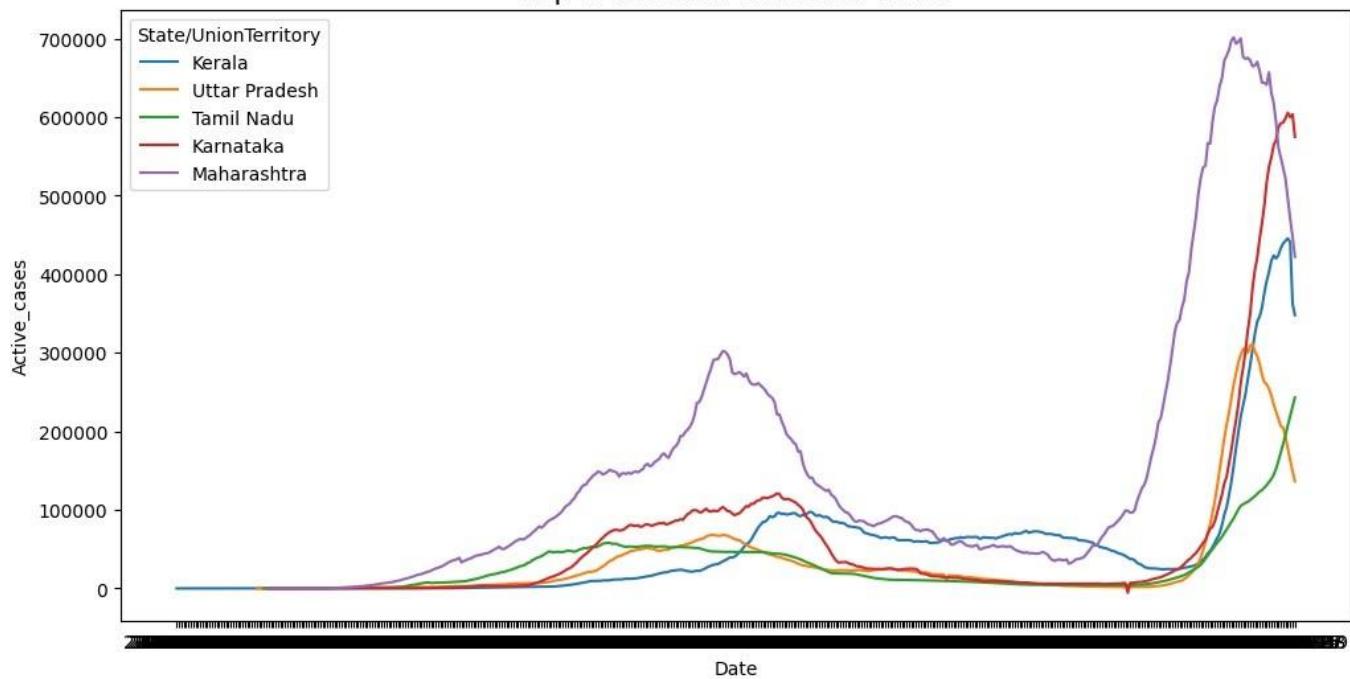
Total

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

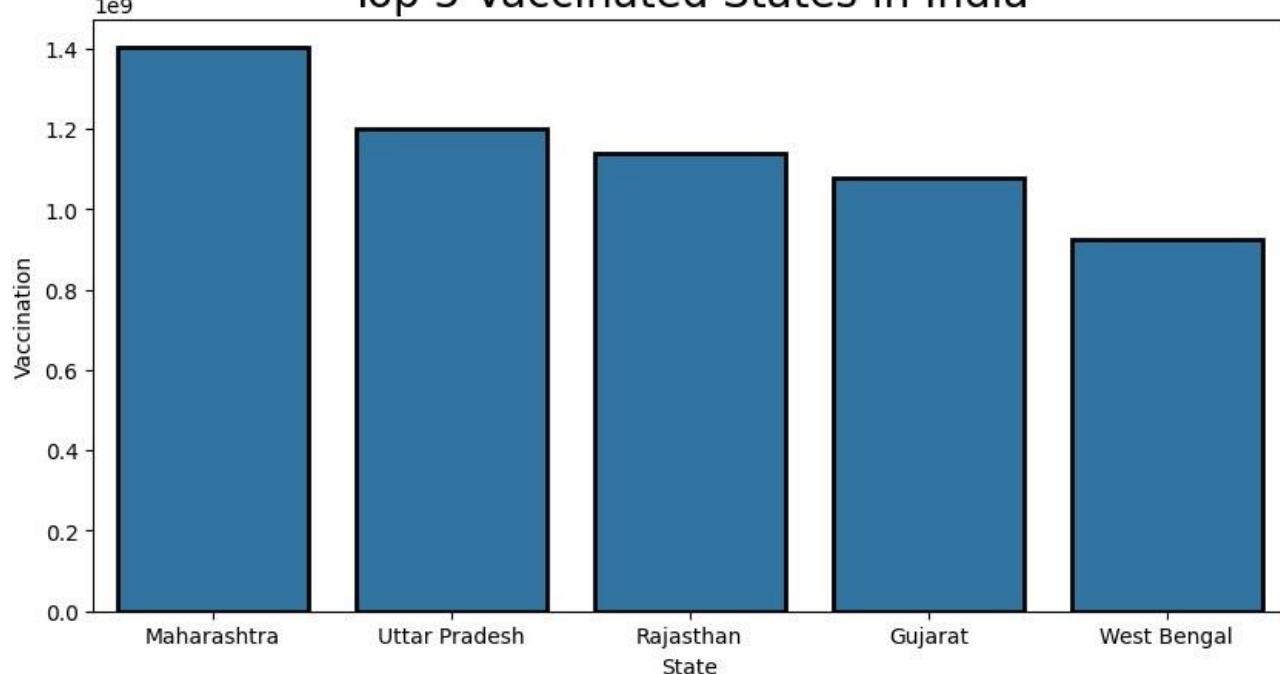
In [56]:

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

top 5 affected states in india



Top 5 Vaccinated States in India



In [2]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Load the CSV data into
vaccine_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_vaccine_statewise.csv")
```

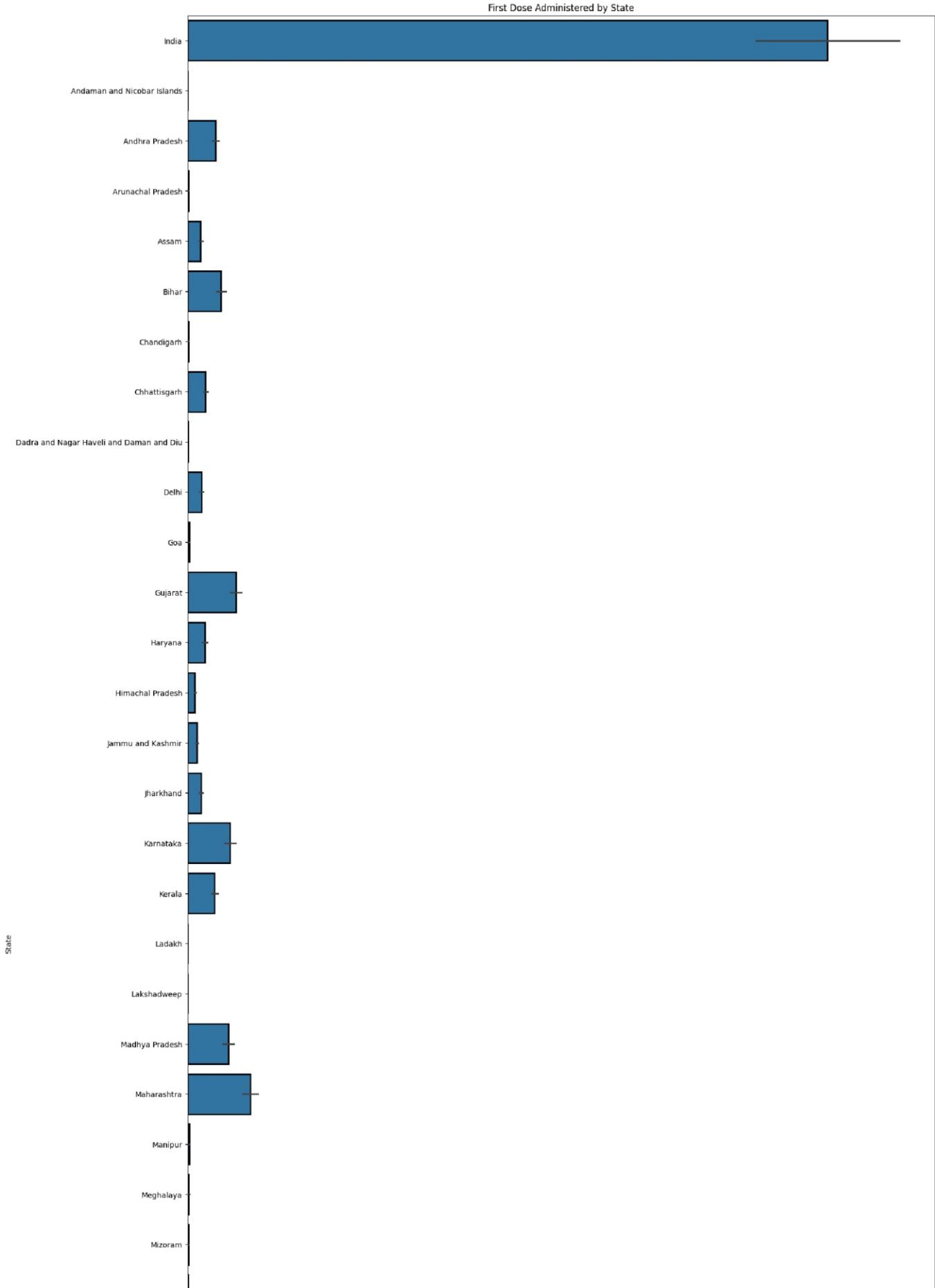
```

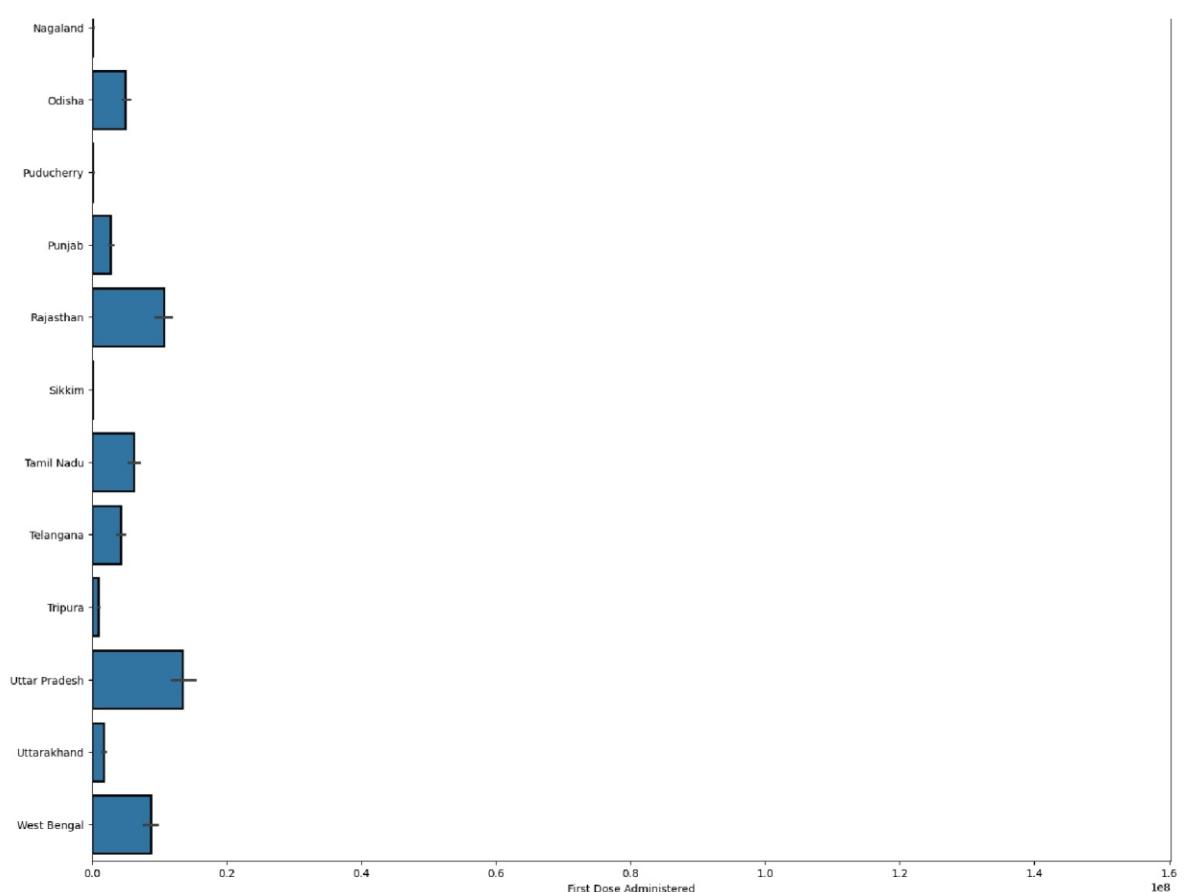
# Create a horizontal
plt.figure(figsize=(18, 45))
sns.barplot(x='First Dose Administered', y='State', data=vaccine_df, linewidth=2, edgecolor='black')

# Set plot title and
plt.title('First Dose Administered by')
plt.xlabel('First Dose Administered')
plt.ylabel('State')

# Show the
plt.show()

```

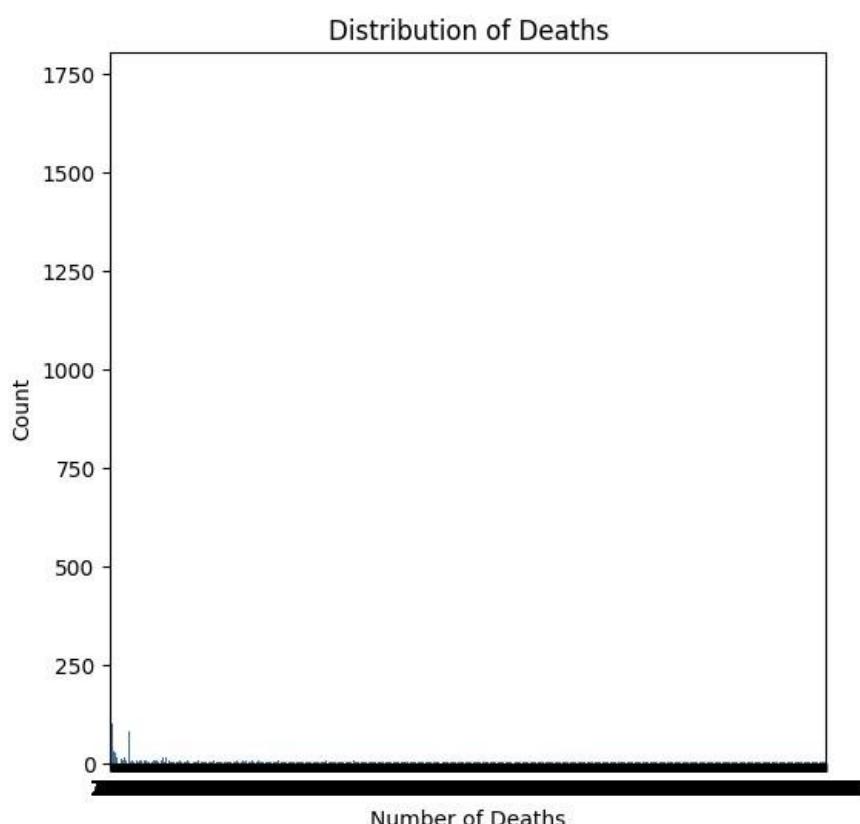




```
In [22]: import seaborn as sns
import matplotlib.pyplot as plt

# Assuming you have a DataFrame named 'covid_df' with
plt.figure(figsize=(6,6))
sns.countplot(data=covid_df, x='Deaths')

# Set plot title and
plt.title('Distribution of Deaths')
plt.xlabel('Number of Deaths')
plt.ylabel('Count')
# Show the
plt.show()
```



In [2]: import seaborn as sns

```

import matplotlib.pyplot as plt
import pandas as pd

# Load the CSV data into
covid_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_19_india (1).csv")

# Include the following line to display plots inline in Jupyter
%matplotlib inline

# Check the column names in your
print(covid_df.columns)

# Create a categorical scatter plot
plt.figure(figsize=(12, 8))
sns.scatterplot(x='variable', y='value', hue='Deaths', data=pd.melt(covid_df[['Confirmed', 'Cured', 'Deaths']]),

# Set plot title and
plt.title('Pairplot of COVID-19 Data with Deaths as')
plt.xlabel('Variable')
plt.ylabel('Value')

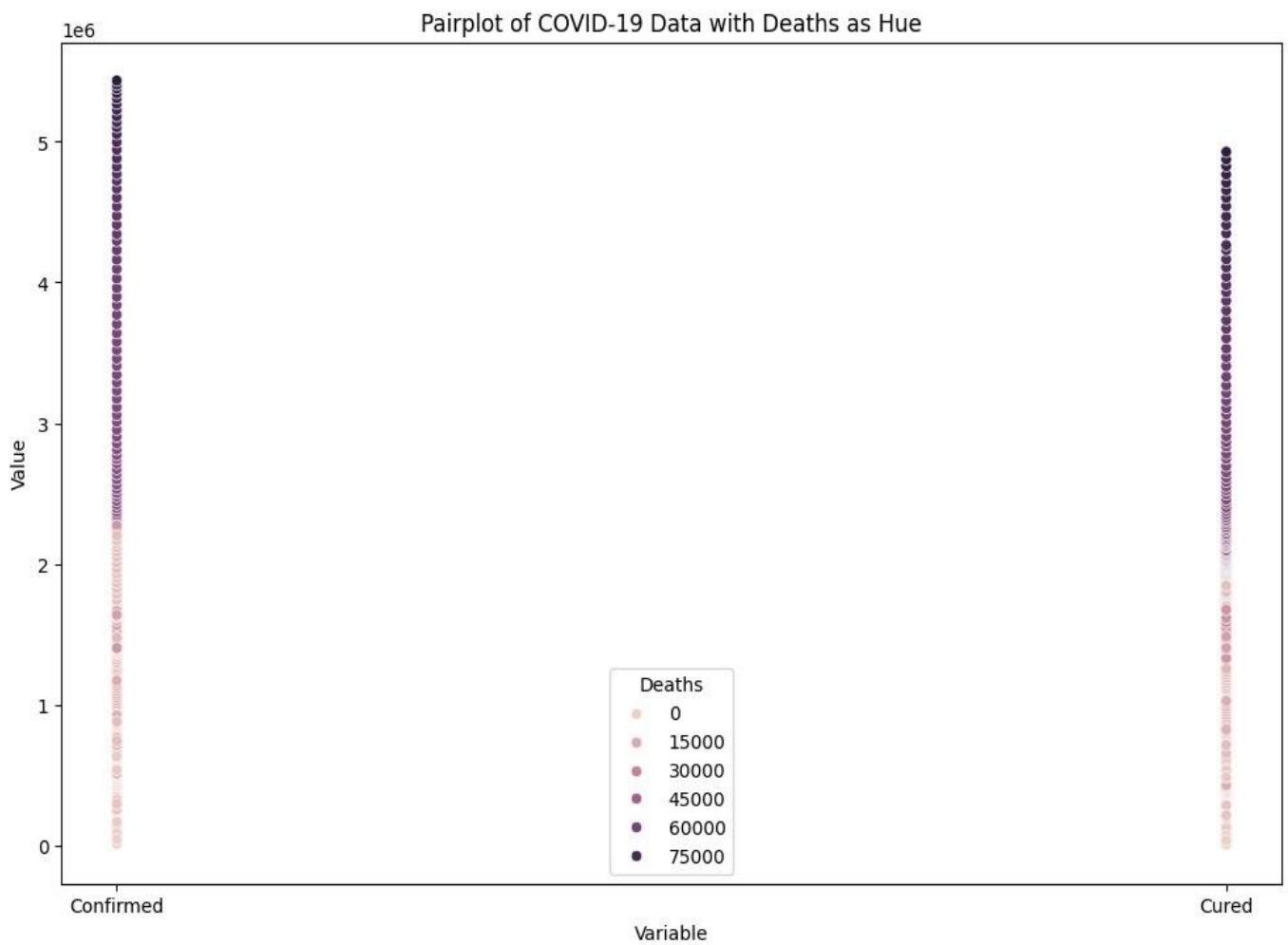
# Show the
plt.show()

```

```

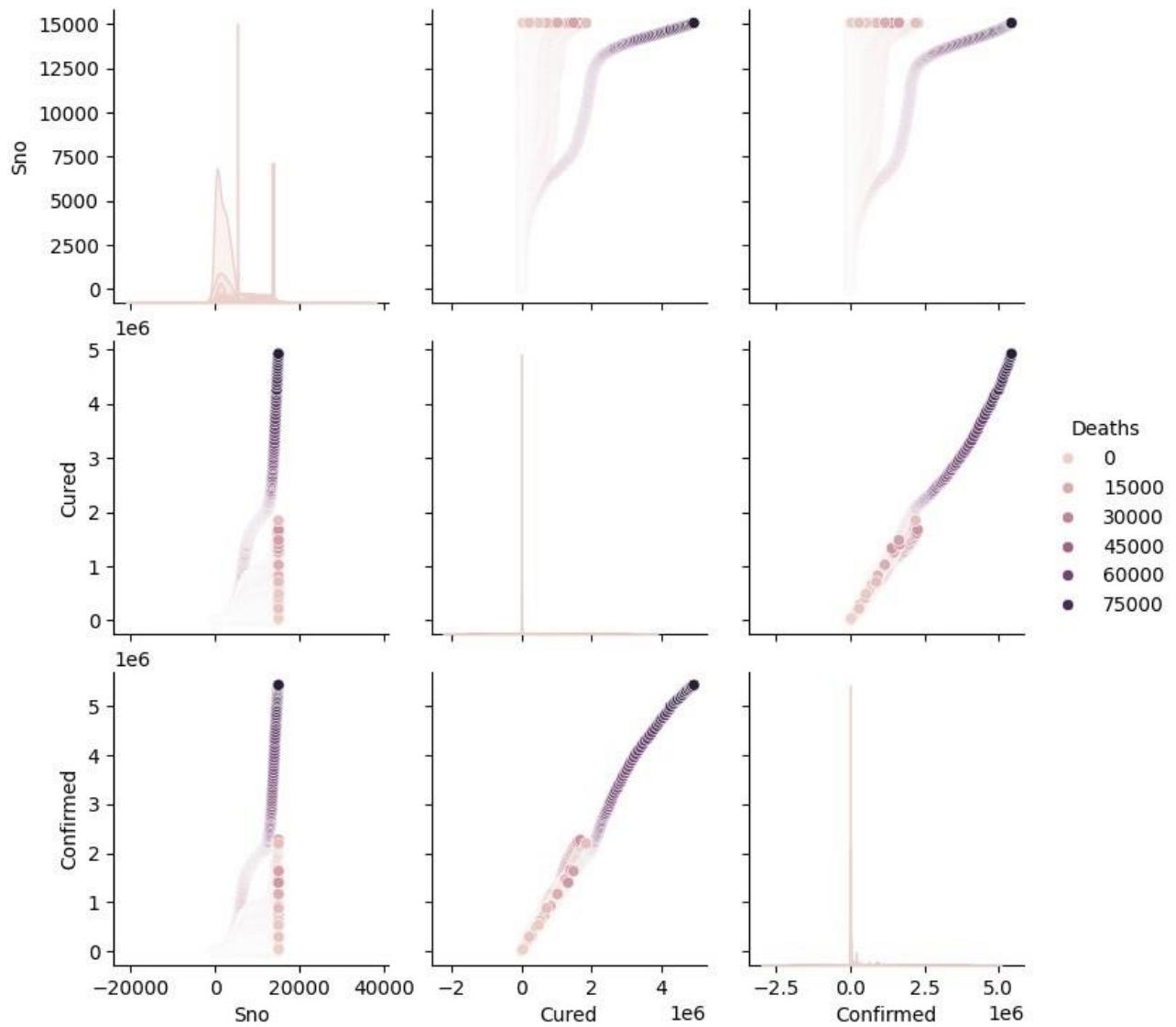
Index(['Sno', 'Date', 'Time', 'State/UnionTerritory',
       'ConfirmedIndianNational',

```



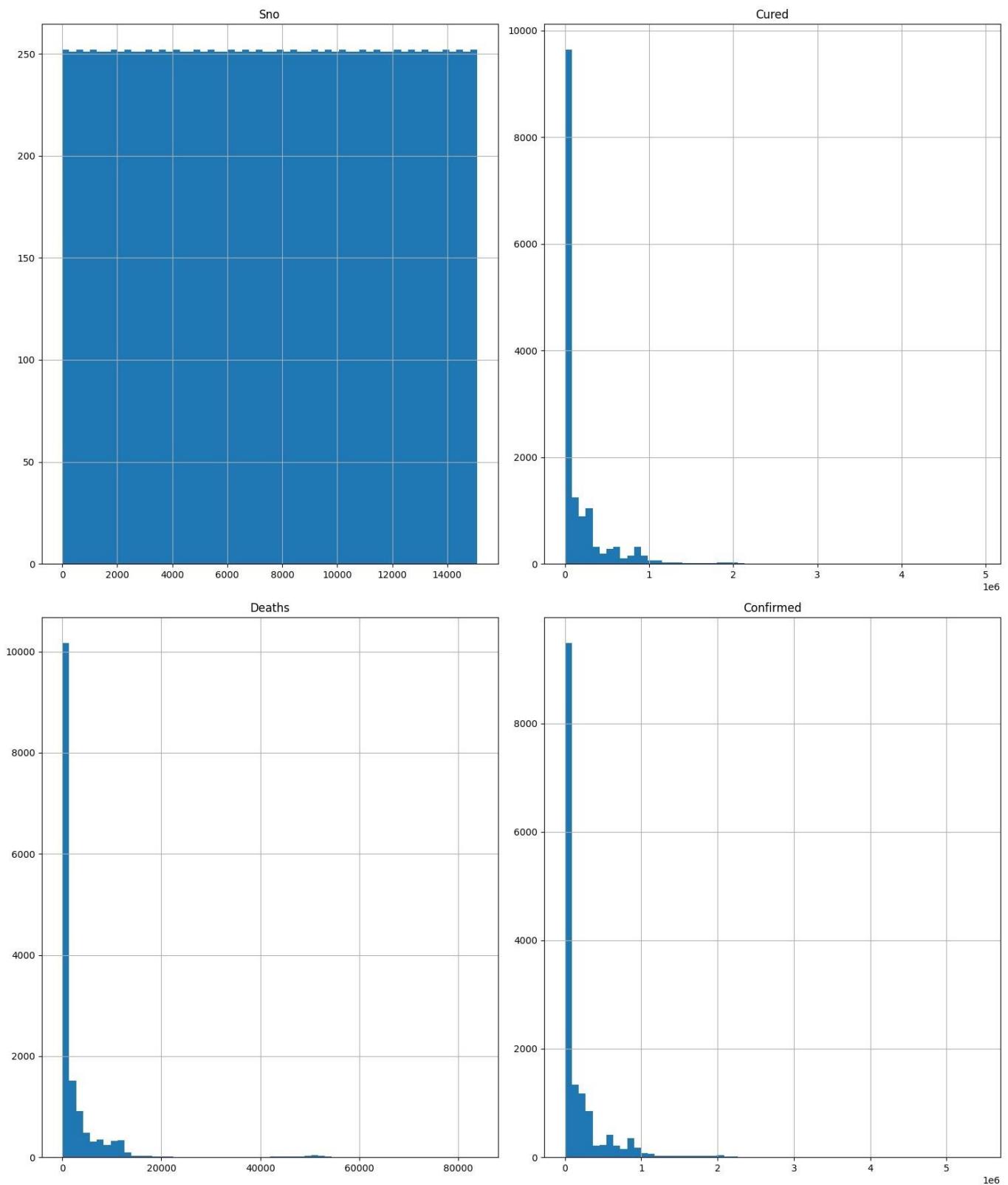
```
In [3]: covid_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_19_india (1).csv")
sns.pairplot(covid_df,hue='Deaths')
```

```
Out[3]: <seaborn.axisgrid.PairGrid at 0x273e450>
```



```
In [3]: covid_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_19_india (1).csv")
covid_df.hist(figsize=(15,18),bins=60)
plt.suptitle('Features Distribution',y=1.02,ha='center',fontsize='large')
plt.tight_layout()
```

Features Distribution



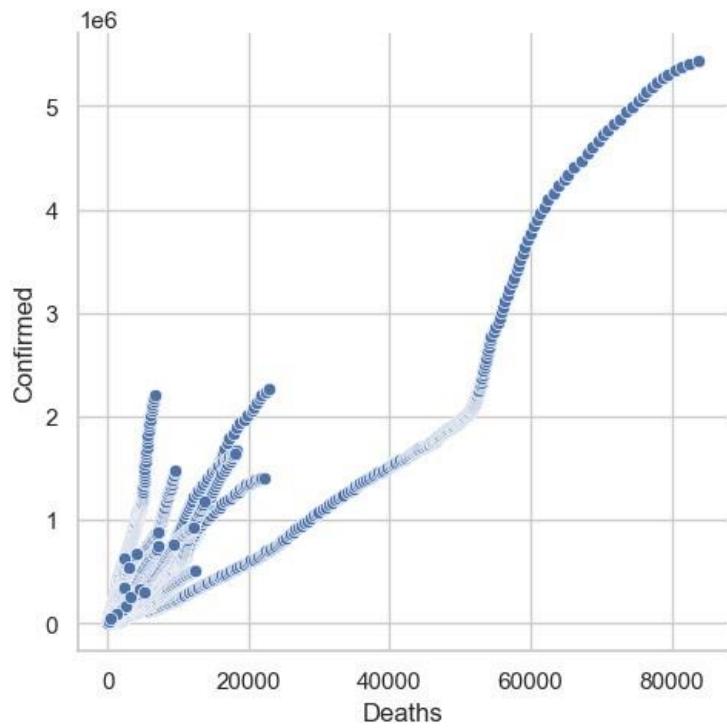
In [5]:

```

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
covid_df =pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_19_india(1).csv")
sns.relplot(x="Deaths", y="Confirmed", data=covid_df)

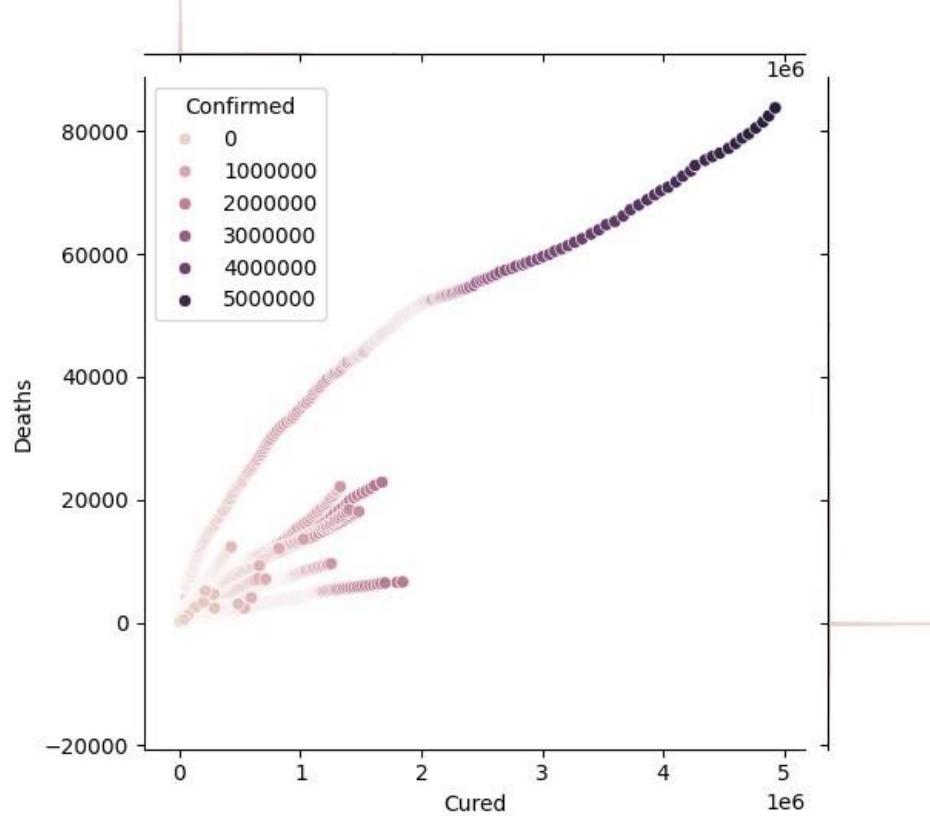
```

Out[5]: <seaborn.axisgrid.FacetGrid at 0x0ba791107d>



```
In [2]: plt.figure(figsize=(10,8))
sns.jointplot(data=covid_df,x='Cured',y='Deaths',hue='Confirmed')
```

```
Out[2]: <seaborn.axisgrid.JointGrid at 0x50211267b>
```



```
In [4]: import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Assuming you have a DataFrame
plt.figure(1, figsize=(15, 3))
```

```

n = 0
covid_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_19_india (1).csv")
# Iterate over the columns 'Deaths', 'Cured', and 'Confirmed'
for x in ['Deaths', 'Cured', 'Confirmed']:
    n += 1
    plt.subplot(1, 3, n)
    plt.subplots_adjust(hspace=0.5, wspace=0.5)
    sns.distplot(covid_df[x], bins=20)
    plt.title('Distplot of {}'.format(x))

plt.show()

```

C:\Users\cmgou\AppData\Local\Temp\ipykernel_26724\3519111254.py:14: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```

sns.distplot(covid_df[x], bins=20)
C:\Users\cmgou\AppData\Local\Temp\ipykernel_26724\3519111254.py:14: UserWarning:

```

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```

sns.distplot(covid_df[x], bins=20)
C:\Users\cmgou\AppData\Local\Temp\ipykernel_26724\3519111254.py:14: UserWarning:

```

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

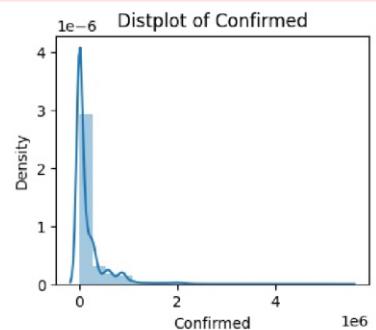
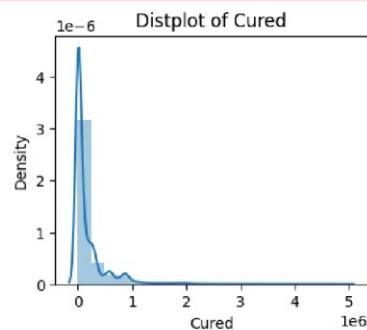
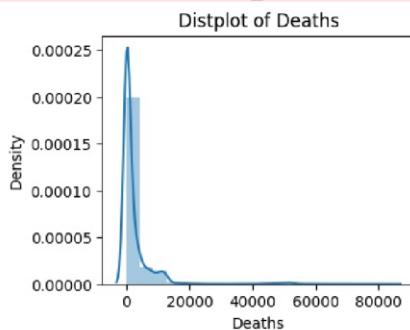
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```

sns.distplot(covid_df[x], bins=20)

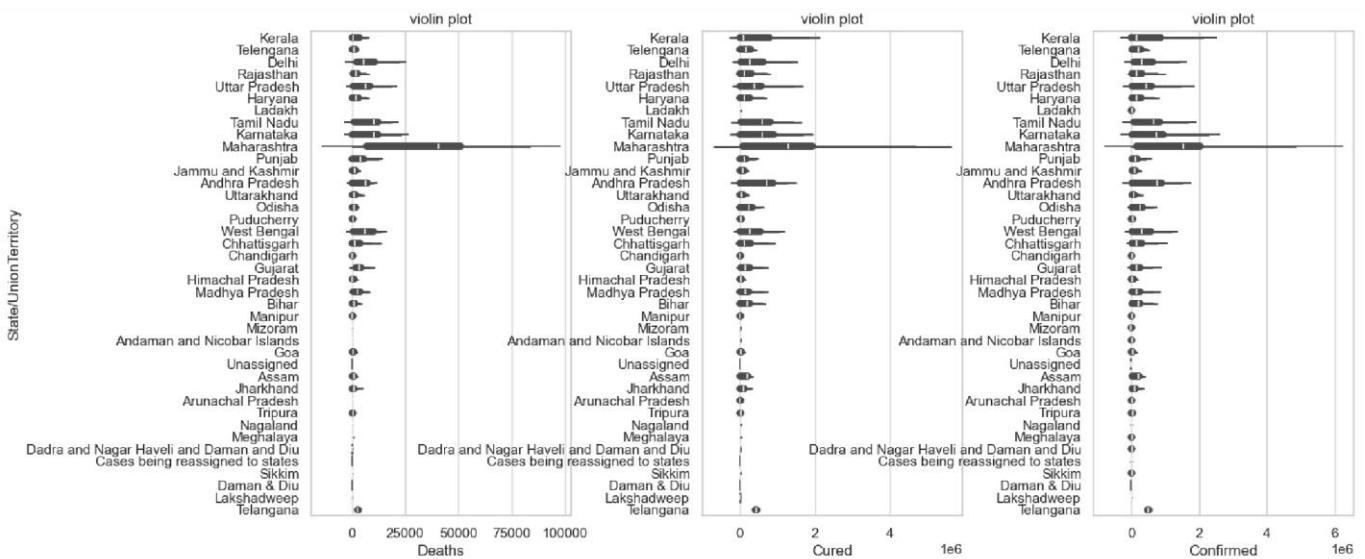
```



```

In [7]: plt.figure(1, figsize=(15, 7))
n = 0
for cols in ['Deaths', 'Cured', 'Confirmed']:
    n += 1
    plt.subplot(1, 3, n)
    plt.subplots_adjust(hspace=0.5, wspace=0.5)
    sns.set(style="whitegrid")
    sns.violinplot(x = cols, y = 'State/UnionTerritory', data = covid_df)
    plt.ylabel('State/UnionTerritory' if n == 1 else ' ')
    plt.title('violin plot')
    plt.show()

```



```
In [8]: import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Assuming you have a DataFrame named 'covid_df'
plt.figure(1, figsize=(15, 7))
n = 0

# Specify the two states you want to include
selected_states = ['Kerala', 'Telangana']

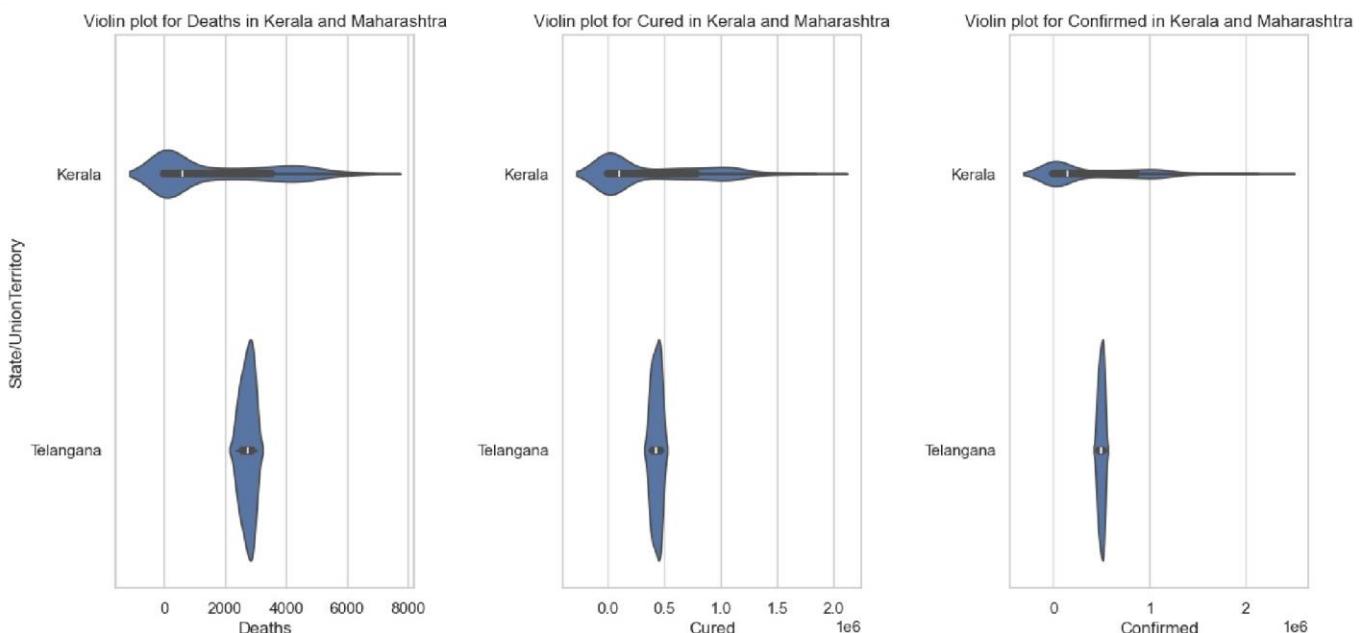
for cols in ['Deaths', 'Cured', 'Confirmed']:
    n += 1
    plt.subplot(1, 3, n)
    sns.set(style="whitegrid")
    plt.subplots_adjust(hspace=0.5, wspace=0.5)

    # Filter the data for the selected states
    filtered_data = covid_df[covid_df['State/UnionTerritory'].isin(selected_states)]

    sns.violinplot(x=cols, y='State/UnionTerritory', data=filtered_data)

    plt.ylabel('State/UnionTerritory' if n == 1 else ' ')
    plt.title(f'Violin plot for {cols} in Kerala and Maharashtra')

plt.show()
```



```
In [4]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(1, figsize=(15, 7))
n = 0
```

```

# Read the CSV file
vaccine_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_vaccine_statewise.csv")

# Filter the DataFrame for the specific state (India)
selected_state = 'India'
filtered_data = vaccine_df[vaccine_df['State'] == selected_state]

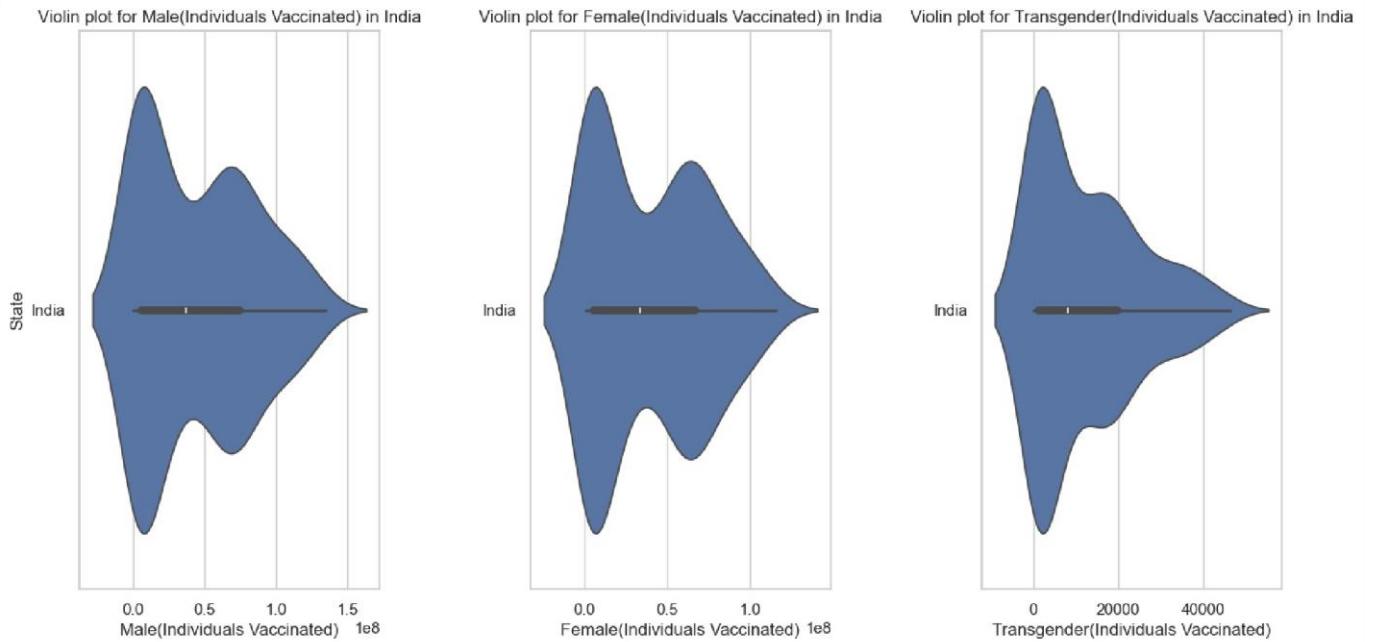
for cols in ['Male(Individuals Vaccinated)', 'Female(Individuals Vaccinated)', 'Transgender(Individuals Vaccinated)']:
    n += 1
    plt.subplot(1, 3, n)
    sns.set(style="whitegrid")
    plt.subplots_adjust(hspace=0.5, wspace=0.5)

    sns.violinplot(x=cols, y='State', data=filtered_data)

    plt.ylabel('State' if n == 1 else ' ')
    plt.title(f'Violin plot for {cols} in {selected_state}')

plt.show()

```



```

In [1]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming you have a DataFrame named 'covid_df'
covid_df = pd.read_csv("C:/Users/cmgou/Desktop/5th/DAV/project/covid_19_india (1).csv")

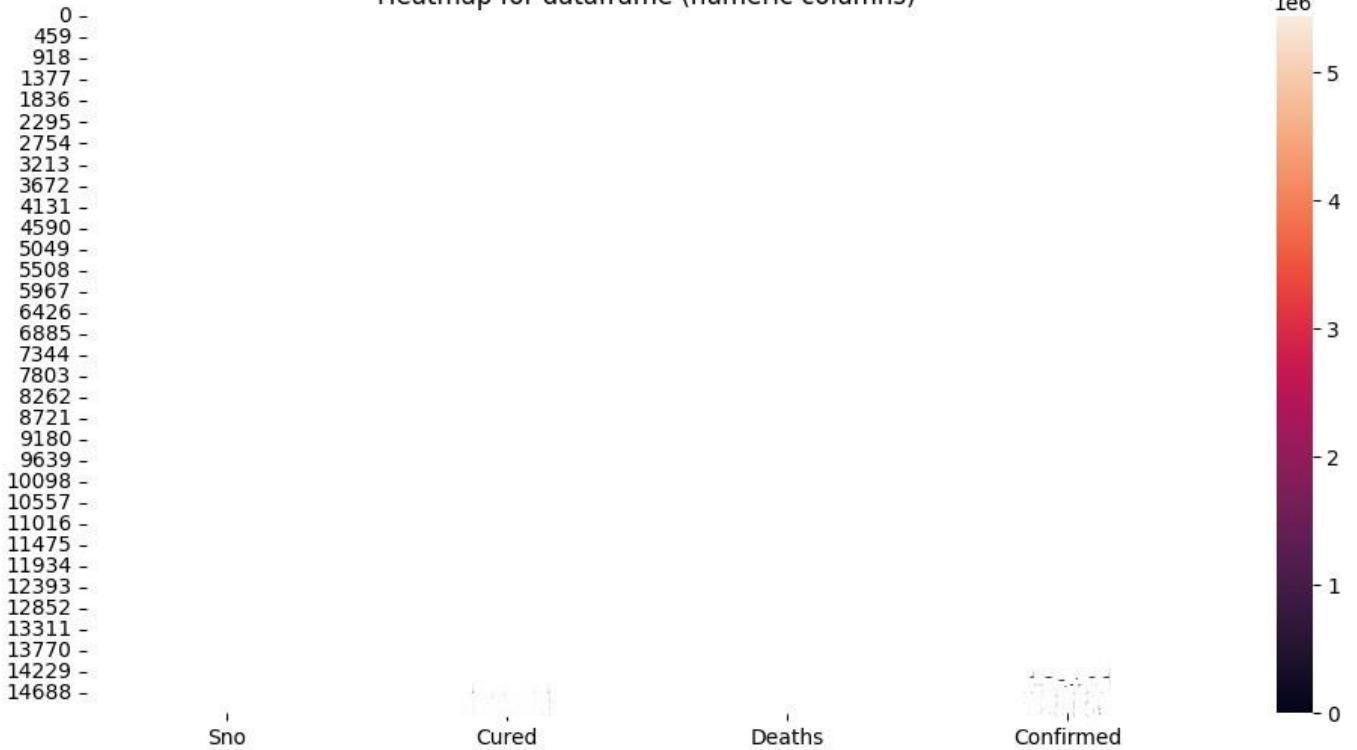
# Select numeric columns only
numeric_columns = covid_df.select_dtypes(include=['number'])

plt.figure(figsize=(12, 6))
plt.title('Heatmap for dataframe (numeric columns)')
sns.heatmap(numeric_columns, annot=True, linecolor='white', linewidths=0.2)

plt.show()

```

Heatmap for dataframe (numeric columns)



In []:

```
[50]: cured = covid_df["Cured"].sum()
deaths = covid_df["Deaths"].sum()
confirmed = covid_df["Confirmed"].sum()
active_cases = covid_df["Active_cases"].sum()

fig = px.pie(names=["CURED", "DEATHS", "CONFIRMED", "ACTIVE_CASES"],
              values=[cured, deaths, confirmed, active_cases],
              title="classification of cases")

fig.show()
```

classification of cases



```
[52]: cured = covid_df["Cured"].sum()
deaths = covid_df["Deaths"].sum()
fig = px.pie(names=["CURED", "DEATHS"],
              values=[cured, deaths],
              title="number of deaths and cured cases")

fig.show()
```

number of deaths and cured cases

