Data Visualization Project On Covid-19 Data

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

In [2]:
```
dataset = pd.read_csv(r"C://Users//NIKHIL//Downloads/covid_19_india.csv")
```

In [3]:
```
dataset
```

Out[3]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational
0	1	2020-01-30	6:00 PM	Kerala	1	0
1	2	2020-01-31	6:00 PM	Kerala	1	0
2	3	2020-02-01	6:00 PM	Kerala	2	0
3	4	2020-02-02	6:00 PM	Kerala	3	0
4	5	2020-02-03	6:00 PM	Kerala	3	0
...	...	...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	-	-
16846	16847	2021-07-07	8:00 AM	Tripura	-	-
16847	16848	2021-07-07	8:00 AM	Uttarakhand	-	-
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	-	-
16849	16850	2021-07-07	8:00 AM	West Bengal	-	-

16850 rows × 9 columns

In [4]:
```
import plotly.express as px
import re
```

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

Out[5]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cure
0	1	2020-01-30	6:00 PM	Kerala	1	0	
1	2	2020-01-31	6:00 PM	Kerala	1	0	
2	3	2020-02-01	6:00 PM	Kerala	2	0	
3	4	2020-02-02	6:00 PM	Kerala	3	0	
4	5	2020-02-03	6:00 PM	Kerala	3	0	

In [6]:
```
1 dataset.tail()
```

Out[6]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational
16845	16846	2021-07-07	8:00 AM	Telangana	-	-
16846	16847	2021-07-07	8:00 AM	Tripura	-	-
16847	16848	2021-07-07	8:00 AM	Uttarakhand	-	-
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	-	-
16849	16850	2021-07-07	8:00 AM	West Bengal	-	-

In [7]:
```
1 dataset.shape
```

Out[7]: (16850, 9)

In [8]:
```
1 dataset.describe()
```

Out[8]:

	Sno	Cured	Deaths	Confirmed
count	16850.000000	1.685000e+04	16850.000000	1.685000e+04
mean	8425.500000	2.360353e+05	3485.222552	2.583667e+05
std	4864.320353	5.225438e+05	9330.541749	5.672808e+05
min	1.000000	0.000000e+00	0.000000	0.000000e+00
25%	4213.250000	2.658500e+03	22.000000	3.644750e+03
50%	8425.500000	2.889500e+04	453.000000	3.336150e+04
75%	12637.750000	2.537510e+05	3071.250000	2.666530e+05
max	16850.000000	5.872268e+06	123531.000000	6.113335e+06

In [9]:
```
1 dataset.values
```

Out[9]:
```
array([[1, '2020-01-30', '6:00 PM', ..., 0, 0, 1],
       [2, '2020-01-31', '6:00 PM', ..., 0, 0, 1],
       [3, '2020-02-01', '6:00 PM', ..., 0, 0, 2],
       ...,
       [16848, '2021-07-07', '8:00 AM', ..., 332006, 7338, 340882],
       [16849, '2021-07-07', '8:00 AM', ..., 1682130, 22656, 1706818],
       [16850, '2021-07-07', '8:00 AM', ..., 1472132, 17834, 1507241]],
      dtype=object)
```

In [10]:
```
1 dataset.columns
```

Out[10]:
```
Index(['Sno', 'Date', 'Time', 'State/UnionTerritory',
       'ConfirmedIndianNational', 'ConfirmedForeignNational', 'Cured',
       'Deaths', 'Confirmed'],
      dtype='object')
```

In [11]:
```
1 dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16850 entries, 0 to 16849
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Sno                       16850 non-null  int64 
 1   Date                      16850 non-null  object
 2   Time                      16850 non-null  object
 3   State/UnionTerritory      16850 non-null  object
 4   ConfirmedIndianNational   16850 non-null  object
 5   ConfirmedForeignNational  16850 non-null  object
 6   Cured                     16850 non-null  int64 
 7   Deaths                    16850 non-null  int64 
 8   Confirmed                 16850 non-null  int64 
dtypes: int64(4), object(5)
memory usage: 1.2+ MB
```

In [12]:
```
1 dataset.isna().sum()
```

Out[12]:
```
Sno                         0
Date                        0
Time                        0
State/UnionTerritory        0
ConfirmedIndianNational     0
ConfirmedForeignNational    0
Cured                       0
Deaths                      0
Confirmed                   0
dtype: int64
```

In [13]:
```
1 dataset.dtypes
```

Out[13]:
```
Sno                          int64
Date                        object
Time                        object
State/UnionTerritory        object
ConfirmedIndianNational     object
ConfirmedForeignNational    object
Cured                        int64
Deaths                       int64
Confirmed                    int64
dtype: object
```

In [14]:
```
1 dataset["State/UnionTerritory"].value_counts()
```

Out[14]:
```
Kerala                                          525
Delhi                                           493
Rajasthan                                       492
Uttar Pradesh                                   491
Haryana                                         491
Ladakh                                          488
Tamil Nadu                                      488
Maharashtra                                     486
Jammu and Kashmir                               486
Punjab                                          486
Karnataka                                       486
Andhra Pradesh                                  483
Uttarakhand                                     480
Odisha                                          479
Puducherry                                      477
West Bengal                                     477
Chhattisgarh                                    476
Chandigarh                                      476
Gujarat                                         475
Himachal Pradesh                                474
Madhya Pradesh                                  474
Manipur                                         471
Bihar                                           471
Mizoram                                         470
Andaman and Nicobar Islands                     469
Goa                                             469
Assam                                           463
Jharkhand                                       463
Arunachal Pradesh                               461
Tripura                                         457
Meghalaya                                       450
Dadra and Nagar Haveli and Daman and Diu        426
Telengana                                       426
Nagaland                                        417
Sikkim                                          410
Lakshadweep                                     209
Telangana                                        67
Cases being reassigned to states                 60
Unassigned                                        3
Dadra and Nagar Haveli                            2
Bihar****                                         2
Daman & Diu                                       1
Name: State/UnionTerritory, dtype: int64
```

In [15]:
```
1 data1 = dataset.replace(r'^-*$',np.nan, regex = True)
```

In [16]:
```
1 data1
```

Out[16]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational
0	1	2020-01-30	6:00 PM	Kerala	1	0
1	2	2020-01-31	6:00 PM	Kerala	1	0
2	3	2020-02-01	6:00 PM	Kerala	2	0
3	4	2020-02-02	6:00 PM	Kerala	3	0
4	5	2020-02-03	6:00 PM	Kerala	3	0
...	...	...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	NaN	NaN
16846	16847	2021-07-07	8:00 AM	Tripura	NaN	NaN
16847	16848	2021-07-07	8:00 AM	Uttarakhand	NaN	NaN
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	NaN	NaN
16849	16850	2021-07-07	8:00 AM	West Bengal	NaN	NaN

16850 rows × 9 columns

In [17]:
```
1 data2 = data1.fillna(0)
```

In [18]:
```
1 data2
```

Out[18]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational
0	1	2020-01-30	6:00 PM	Kerala	1	0
1	2	2020-01-31	6:00 PM	Kerala	1	0
2	3	2020-02-01	6:00 PM	Kerala	2	0
3	4	2020-02-02	6:00 PM	Kerala	3	0
4	5	2020-02-03	6:00 PM	Kerala	3	0
...	...	...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	0	0
16846	16847	2021-07-07	8:00 AM	Tripura	0	0
16847	16848	2021-07-07	8:00 AM	Uttarakhand	0	0
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	0	0
16849	16850	2021-07-07	8:00 AM	West Bengal	0	0

16850 rows × 9 columns

In [19]:
```
1 Ds1 = ['Cases being reassigned to states']
2 Ds2 = data2.loc[data2['State/UnionTerritory'].isin(Ds1)]
```

In [20]:
```
1 Ds2
```

Out[20]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	C
2133	2134	2020-05-20	8:00 AM	Cases being reassigned to states	0	0	
2167	2168	2020-05-21	8:00 AM	Cases being reassigned to states	0	0	
2201	2202	2020-05-22	8:00 AM	Cases being reassigned to states	0	0	
2235	2236	2020-05-23	8:00 AM	Cases being reassigned to states	0	0	
2270	2271	2020-05-24	8:00 AM	Cases being reassigned to states	0	0	
2305	2306	2020-05-25	8:00 AM	Cases being reassigned to states	0	0	
2341	2342	2020-05-26	8:00 AM	Cases being reassigned to states	0	0	
2377	2378	2020-05-27	8:00 AM	Cases being reassigned to states	0	0	
2413	2414	2020-05-28	8:00 AM	Cases being reassigned to states	0	0	
2449	2450	2020-05-29	8:00 AM	Cases being reassigned to states	0	0	
2485	2486	2020-05-30	8:00 AM	Cases being reassigned to states	0	0	
2521	2522	2020-05-31	8:00 AM	Cases being reassigned to states	0	0	
2557	2558	2020-06-01	8:00 AM	Cases being reassigned to states	0	0	
2593	2594	2020-06-02	8:00 AM	Cases being reassigned to states	0	0	
2629	2630	2020-06-03	8:00 AM	Cases being reassigned to states	0	0	
2665	2666	2020-06-04	8:00 AM	Cases being reassigned to states	0	0	
2701	2702	2020-06-05	8:00 AM	Cases being reassigned to states	0	0	
2737	2738	2020-06-06	8:00 AM	Cases being reassigned to states	0	0	
2773	2774	2020-06-07	8:00 AM	Cases being reassigned to states	0	0	
2809	2810	2020-06-08	8:00 AM	Cases being reassigned to states	0	0	
2845	2846	2020-06-09	8:00 AM	Cases being reassigned to states	0	0	
2881	2882	2020-06-10	8:00 AM	Cases being reassigned to states	0	0	

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	C
2918	2919	2020-06-11	8:00 AM	Cases being reassigned to states	0	0	
2954	2955	2020-06-12	8:00 AM	Cases being reassigned to states	0	0	
2990	2991	2020-06-13	8:00 AM	Cases being reassigned to states	0	0	
3026	3027	2020-06-14	8:00 AM	Cases being reassigned to states	0	0	
3062	3063	2020-06-15	8:00 AM	Cases being reassigned to states	0	0	
3098	3099	2020-06-16	8:00 AM	Cases being reassigned to states	0	0	
3134	3135	2020-06-17	8:00 AM	Cases being reassigned to states	0	0	
3170	3171	2020-06-18	8:00 AM	Cases being reassigned to states	0	0	
3206	3207	2020-06-19	8:00 AM	Cases being reassigned to states	0	0	
3242	3243	2020-06-20	8:00 AM	Cases being reassigned to states	0	0	
3278	3279	2020-06-21	8:00 AM	Cases being reassigned to states	0	0	
3314	3315	2020-06-22	8:00 AM	Cases being reassigned to states	0	0	
3350	3351	2020-06-23	8:00 AM	Cases being reassigned to states	0	0	
3386	3387	2020-06-24	8:00 AM	Cases being reassigned to states	0	0	
3422	3423	2020-06-25	8:00 AM	Cases being reassigned to states	0	0	
3458	3459	2020-06-26	8:00 AM	Cases being reassigned to states	0	0	
3494	3495	2020-06-27	8:00 AM	Cases being reassigned to states	0	0	
3530	3531	2020-06-28	8:00 AM	Cases being reassigned to states	0	0	
3566	3567	2020-06-29	8:00 AM	Cases being reassigned to states	0	0	
3602	3603	2020-06-30	8:00 AM	Cases being reassigned to states	0	0	
3638	3639	2020-07-01	8:00 AM	Cases being reassigned to states	0	0	
3674	3675	2020-07-02	8:00 AM	Cases being reassigned to states	0	0	
3710	3711	2020-07-03	8:00 AM	Cases being reassigned to states	0	0	

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	C
3746	3747	2020-07-04	8:00 AM	Cases being reassigned to states	0	0	
3782	3783	2020-07-05	8:00 AM	Cases being reassigned to states	0	0	
3818	3819	2020-07-06	8:00 AM	Cases being reassigned to states	0	0	
3854	3855	2020-07-07	8:00 AM	Cases being reassigned to states	0	0	
3890	3891	2020-07-08	8:00 AM	Cases being reassigned to states	0	0	
3926	3927	2020-07-09	8:00 AM	Cases being reassigned to states	0	0	
3962	3963	2020-07-10	8:00 AM	Cases being reassigned to states	0	0	
3998	3999	2020-07-11	8:00 AM	Cases being reassigned to states	0	0	
4034	4035	2020-07-12	8:00 AM	Cases being reassigned to states	0	0	
4070	4071	2020-07-13	8:00 AM	Cases being reassigned to states	0	0	
4106	4107	2020-07-14	8:00 AM	Cases being reassigned to states	0	0	
4142	4143	2020-07-15	8:00 AM	Cases being reassigned to states	0	0	
4178	4179	2020-07-16	8:00 AM	Cases being reassigned to states	0	0	
4214	4215	2020-07-17	8:00 AM	Cases being reassigned to states	0	0	
4250	4251	2020-07-18	8:00 AM	Cases being reassigned to states	0	0	

In [21]:
```
ua1 = ['Unassigned']
ua2 = data1.loc[data1['State/UnionTerritory'].isin(ua1)]
```

In [22]:
```
1 ua2
```

Out[22]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cu
500	501	2020-03-30	9:30 PM	Unassigned	NaN	NaN	
528	529	2020-03-31	8:30 PM	Unassigned	NaN	NaN	
617	618	2020-04-03	6:00 PM	Unassigned	NaN	NaN	

In [23]:
```
1 data3 = data2['State/UnionTerritory'] = data2['State/UnionTerritory'].replac
```

In [24]:
```
1 data2['State/UnionTerritory'].unique()
```

Out[24]:
```
array(['Kerala', 'Telangana', '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'],
      dtype=object)
```

In [25]:
```
1 data3 = data2.copy()
```

In [26]:
```
1 data3.drop(data3[data3['State/UnionTerritory']=='Cases being reassigned to s
2 data3.drop(data3[data3['State/UnionTerritory']=='Unassigned'].index, inplace
```

In [27]:
```
1 data3['State/UnionTerritory'].unique()
```

Out[27]:
```
array(['Kerala', 'Telangana', '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', 'Assam', 'Jharkhand', 'Arunachal Pradesh', 'Tripura',
       'Nagaland', 'Meghalaya',
       'Dadra and Nagar Haveli and Daman and Diu', 'Sikkim',
       'Lakshadweep'], dtype=object)
```

In [28]:
```
data4=data3.groupby("State/UnionTerritory")['Cured','Deaths','Confirmed'].su
data4
```

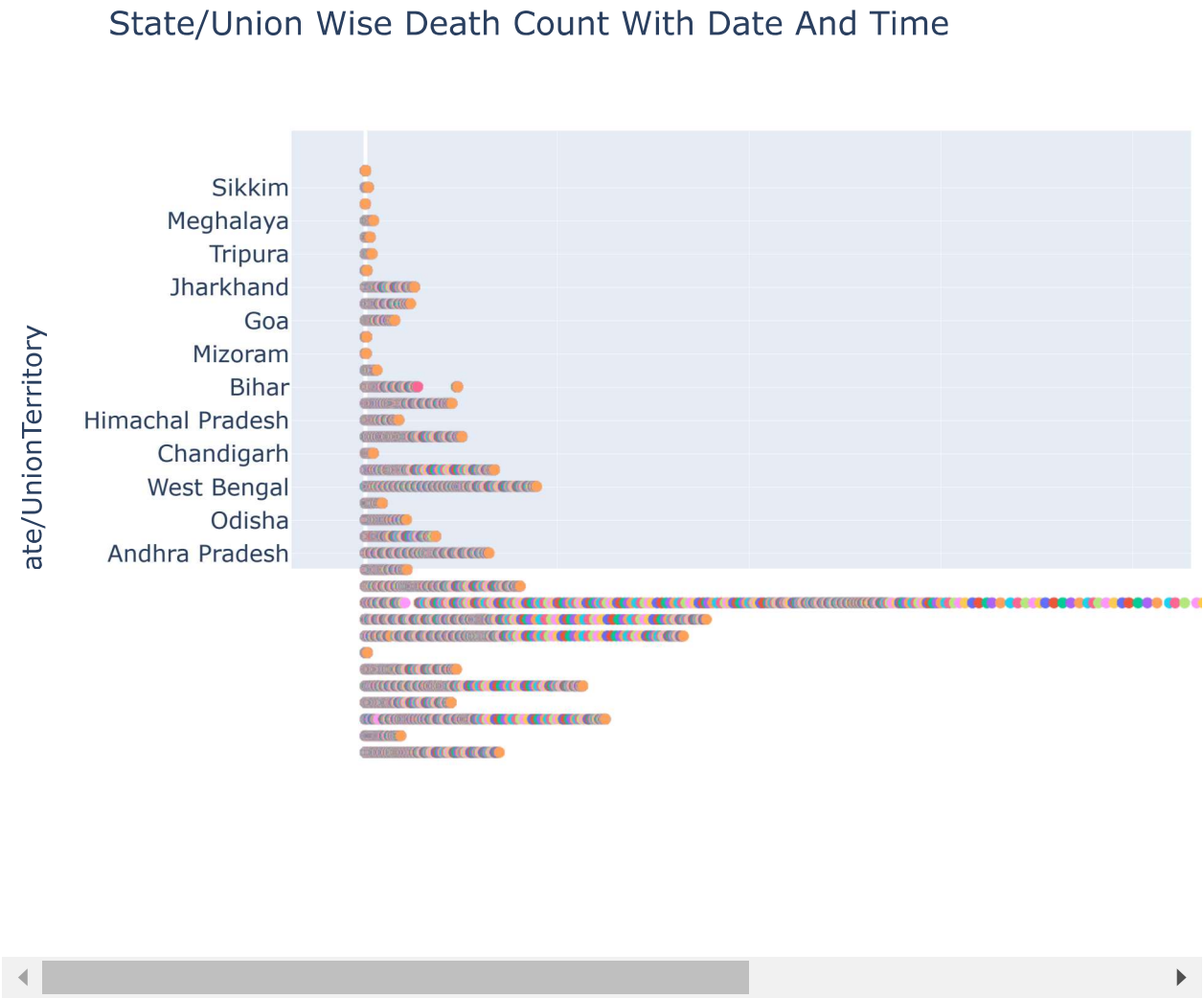
```
C:\Users\NIKHIL\AppData\Local\Temp/ipykernel_11904/2709595874.py:1: FutureWarni
ng: Indexing with multiple keys (implicitly converted to a tuple of keys) will
be deprecated, use a list instead.
  data4=data3.groupby("State/UnionTerritory")['Cured','Deaths','Confirmed'].sum
().reset_index()
```

Out[28]:

	State/UnionTerritory	Cured	Deaths	Confirmed
0	Andaman and Nicobar Islands	1589935	22624	1675248
1	Andhra Pradesh	303427899	2475816	324146783
2	Arunachal Pradesh	5150519	19303	5598324
3	Assam	74011348	459575	80418492
4	Bihar	101533848	775163	108312449
5	Chandigarh	7980284	119356	8691806
6	Chhattisgarh	117163544	1591126	128751782
7	Dadra and Nagar Haveli and Daman and Diu	1491338	882	1587570
8	Delhi	224062704	4066907	236972842
9	Goa	20224042	338359	22280065
10	Gujarat	103995131	1866811	114557615
11	Haryana	100010131	1166573	107408371
12	Himachal Pradesh	20682770	371931	23052151
13	Jammu and Kashmir	42295048	686680	46899925
14	Jharkhand	46083978	569298	49971564
15	Karnataka	345648926	4819018	387597335
16	Kerala	311127643	1327754	344319045
17	Ladakh	3059045	38578	3344131
18	Lakshadweep	471712	2178	561459
19	Madhya Pradesh	100169697	1427780	108712983
20	Maharashtra	813788907	19314532	908892470
21	Manipur	8420223	122089	9440912
22	Meghalaya	4606548	66293	5221064
23	Mizoram	1534630	5073	1822190
24	Nagaland	3628619	39420	4089547
25	Odisha	117984789	600149	126408397
26	Puducherry	14376916	249683	15858688
27	Punjab	71108712	2216735	78999515
28	Rajasthan	117312772	1159823	128998101

	State/UnionTerritory	Cured	Deaths	Confirmed
29	Sikkim	1983899	41530	2315519
30	Tamil Nadu	317067499	4731627	342829697
31	Telangana	100211245	617882	108152726
32	Tripura	10479169	124444	11397656
33	Uttar Pradesh	232529439	3347656	252843682
34	Uttarakhand	36684388	728512	41179396
35	West Bengal	195296839	3214840	209822848

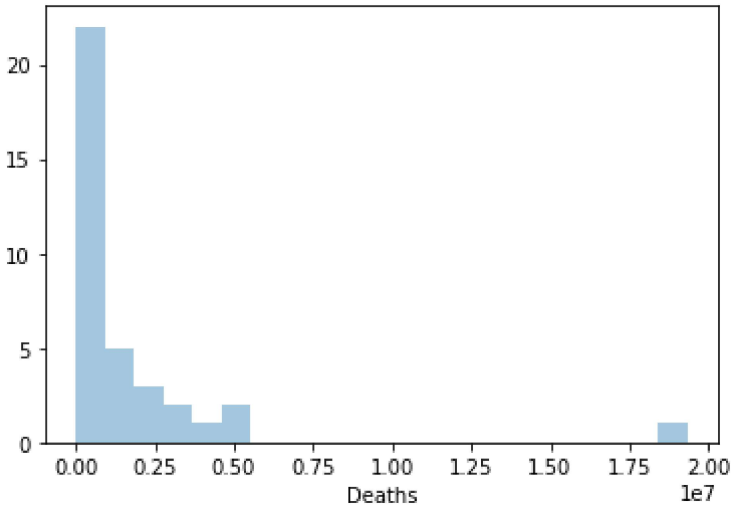
In [29]:
```
1 import plotly.express as px
2 px.scatter(x = 'Deaths' , y = 'State/UnionTerritory' , data_frame = data3, c
```



In [30]:
```
1 sns.distplot(data4['Deaths'], kde = False)
```

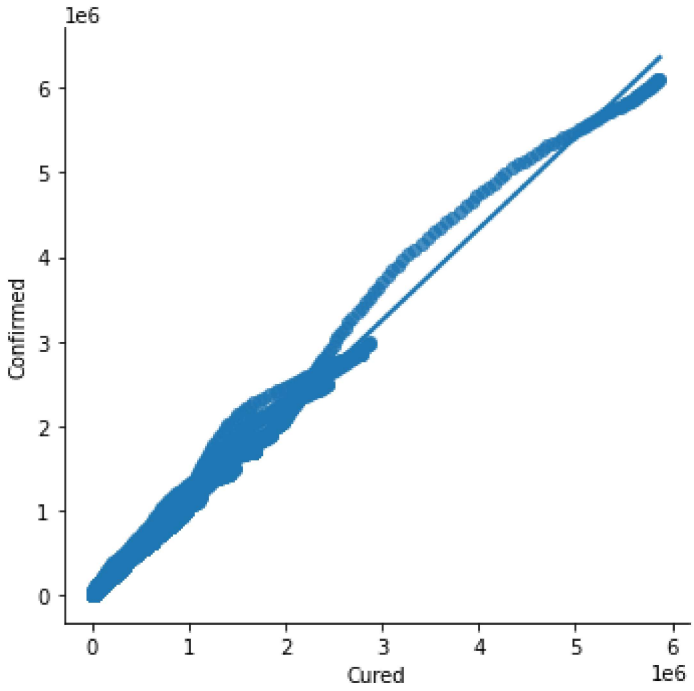
```
C:\Users\NIKHIL\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futu
reWarning:

`distplot` is a deprecated function and will be removed in a future version. Pl
ease adapt your code to use either `displot` (a figure-level function with simi
lar flexibility) or `histplot` (an axes-level function for histograms).
```

Out[30]: <AxesSubplot:xlabel='Deaths'>



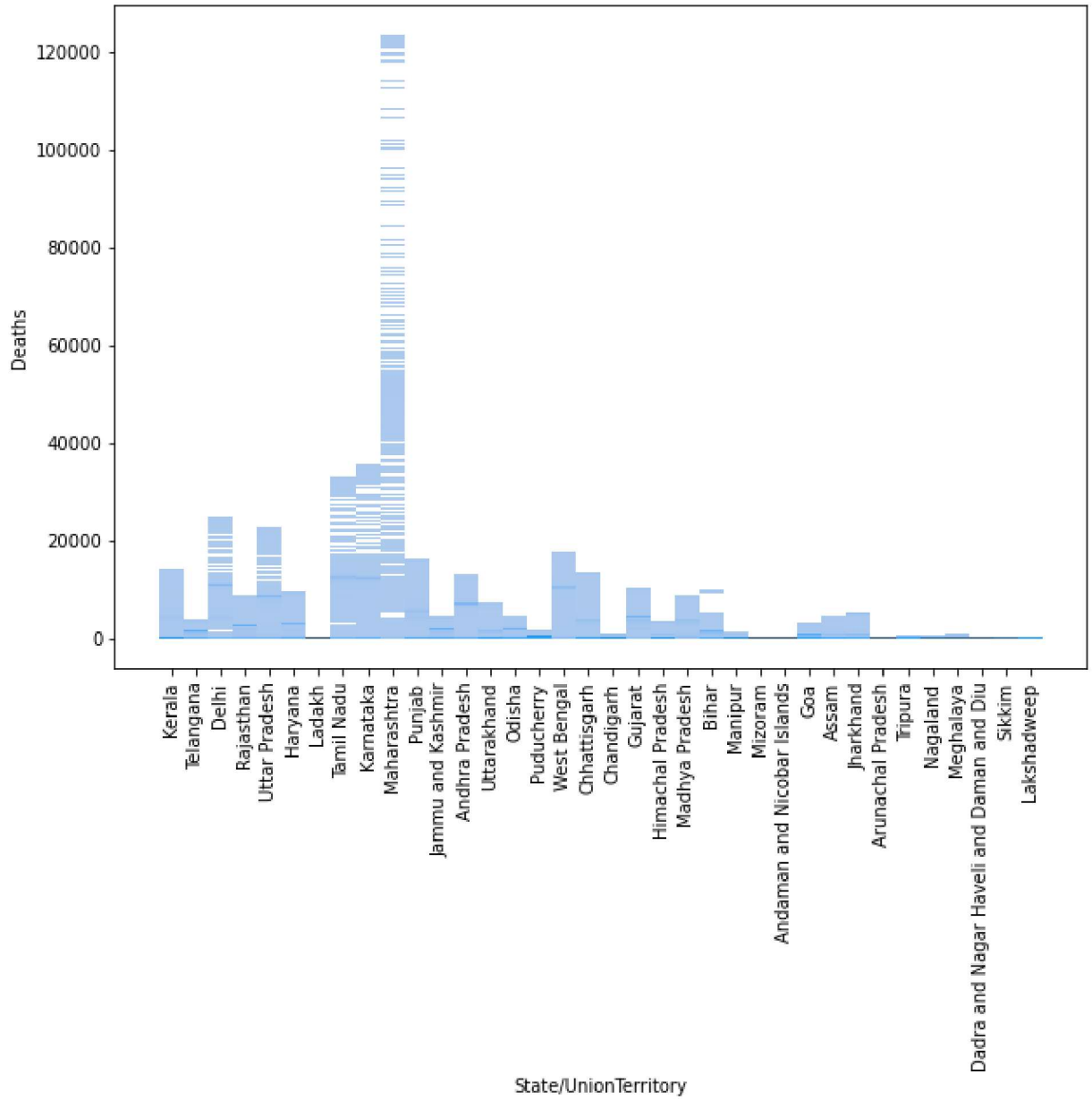
In [31]:
```
1 sns.lmplot( x = 'Cured', y = 'Confirmed', data = data3)
```

Out[31]: <seaborn.axisgrid.FacetGrid at 0x20cb5b168b0>



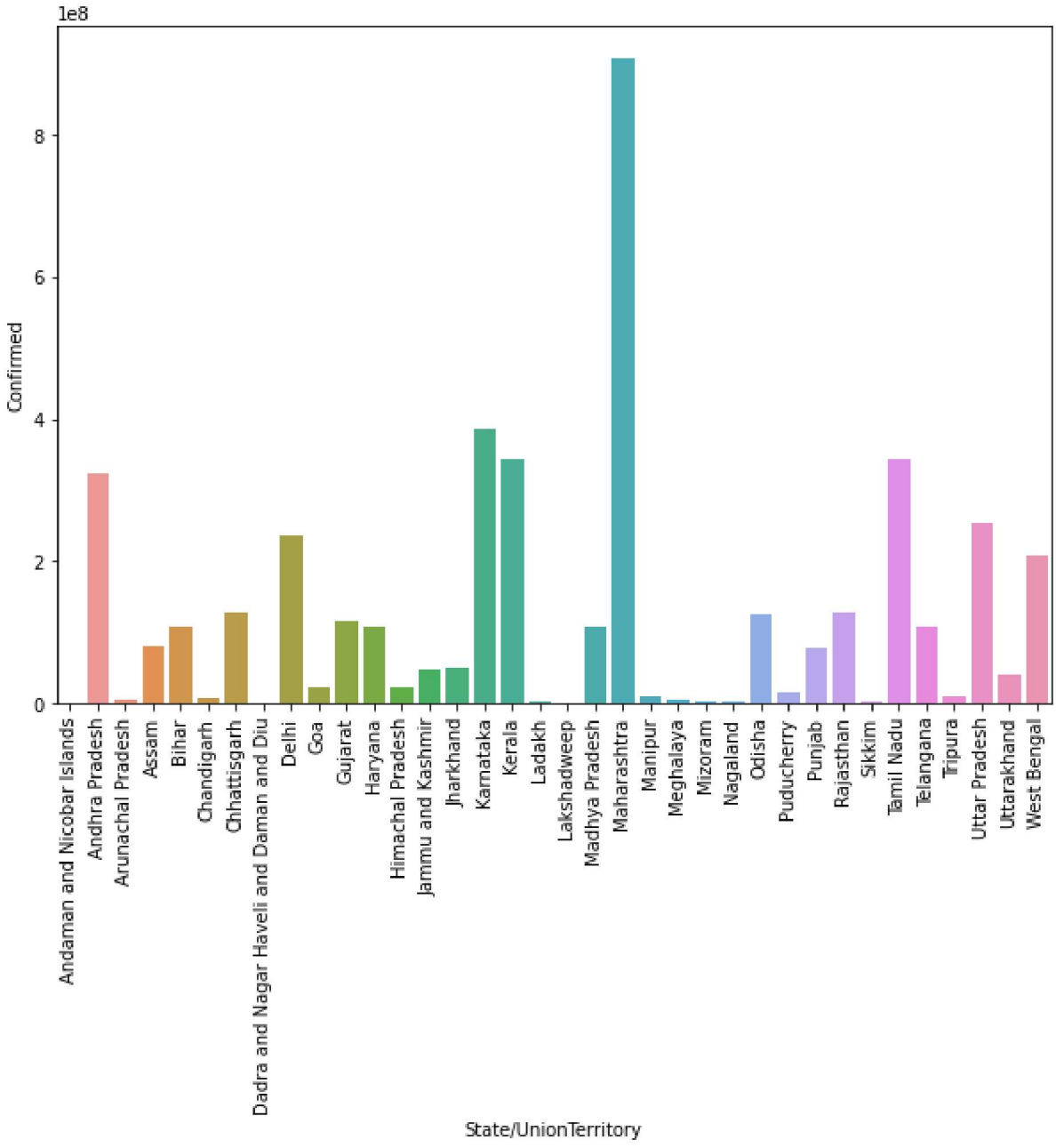
In [32]:
```
fig = plt.figure()
fig = plt.figure(figsize=(10,7))
wer = sns.histplot(data = data3 , x = 'State/UnionTerritory', y = 'Deaths',
plt.setp(wer.get_xticklabels(), rotation=90)
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



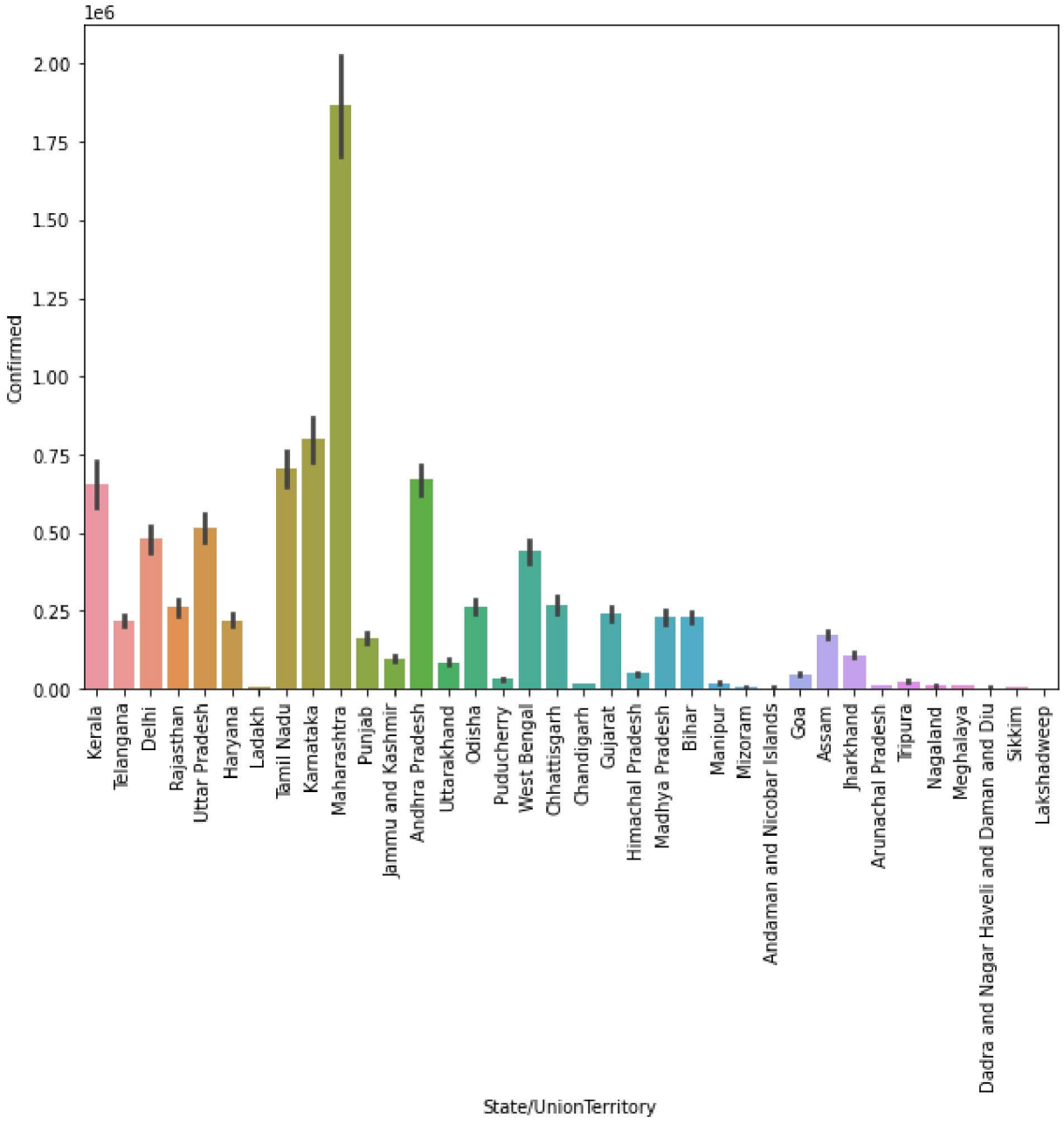
In [33]:
```
fig = plt.figure()
fig = plt.figure(figsize=(10,7))
wer = sns.barplot(x = 'State/UnionTerritory', y = 'Confirmed', data = data4)
plt.setp(wer.get_xticklabels(), rotation = 90)
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



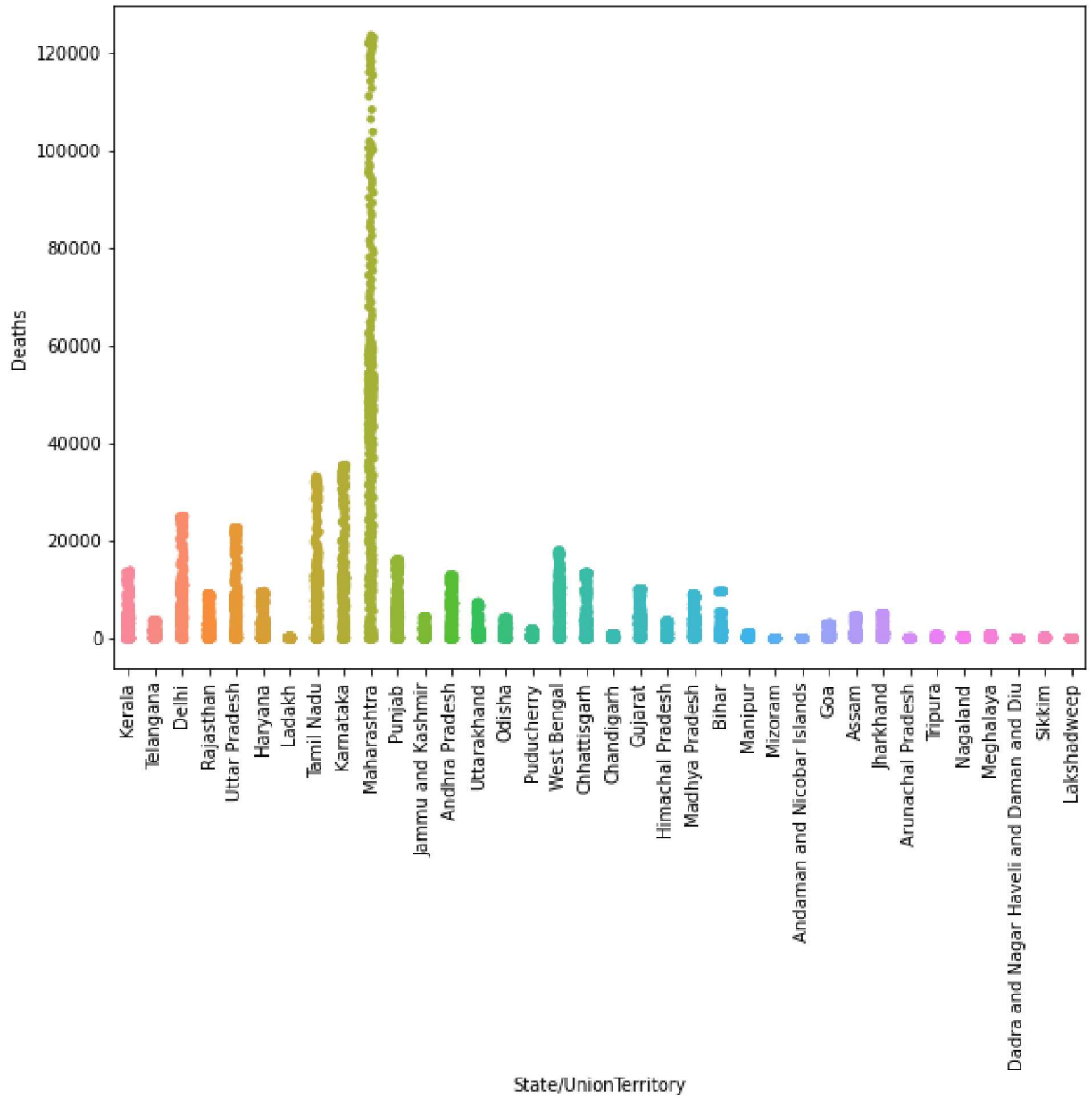
In [34]:
```
fig = plt.figure()
fig = plt.figure(figsize=(10,7))
wer = sns.barplot(x = 'State/UnionTerritory', y = 'Confirmed', data = data3)
plt.setp(wer.get_xticklabels(), rotation = 90)
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

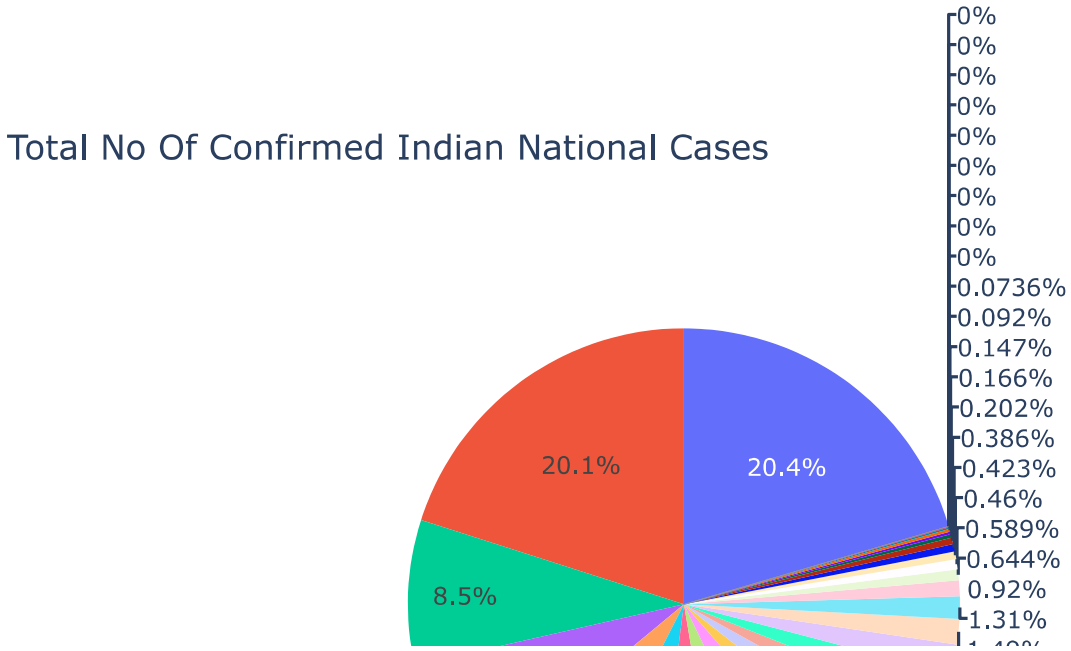


In [35]:
```
fig = plt.figure()
fig = plt.figure(figsize=(10,7))
sns.stripplot(x = 'State/UnionTerritory' , y='Deaths', data=data3)
plt.xticks(rotation=90)

plt.show()
```

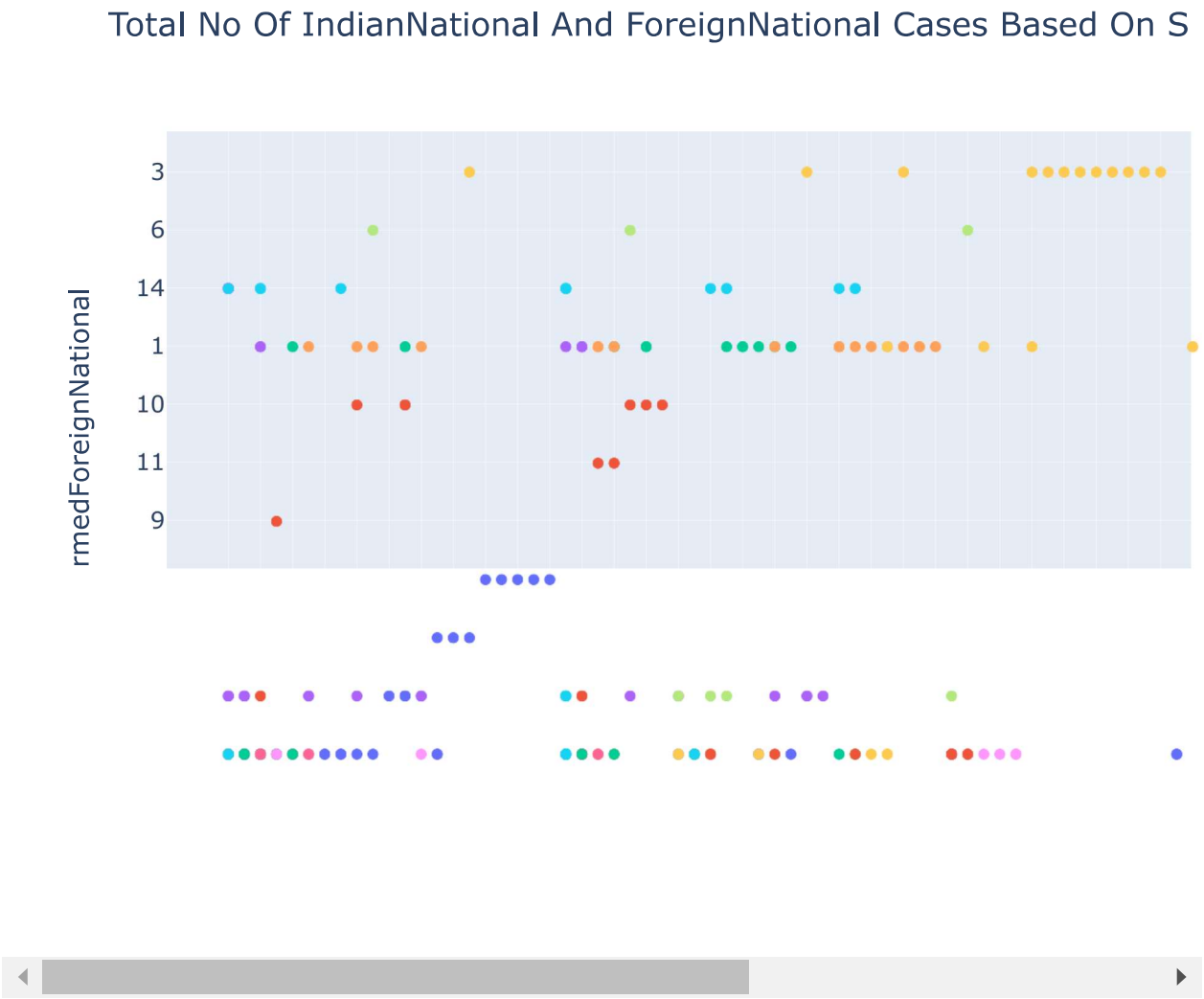
```
<Figure size 432x288 with 0 Axes>
```



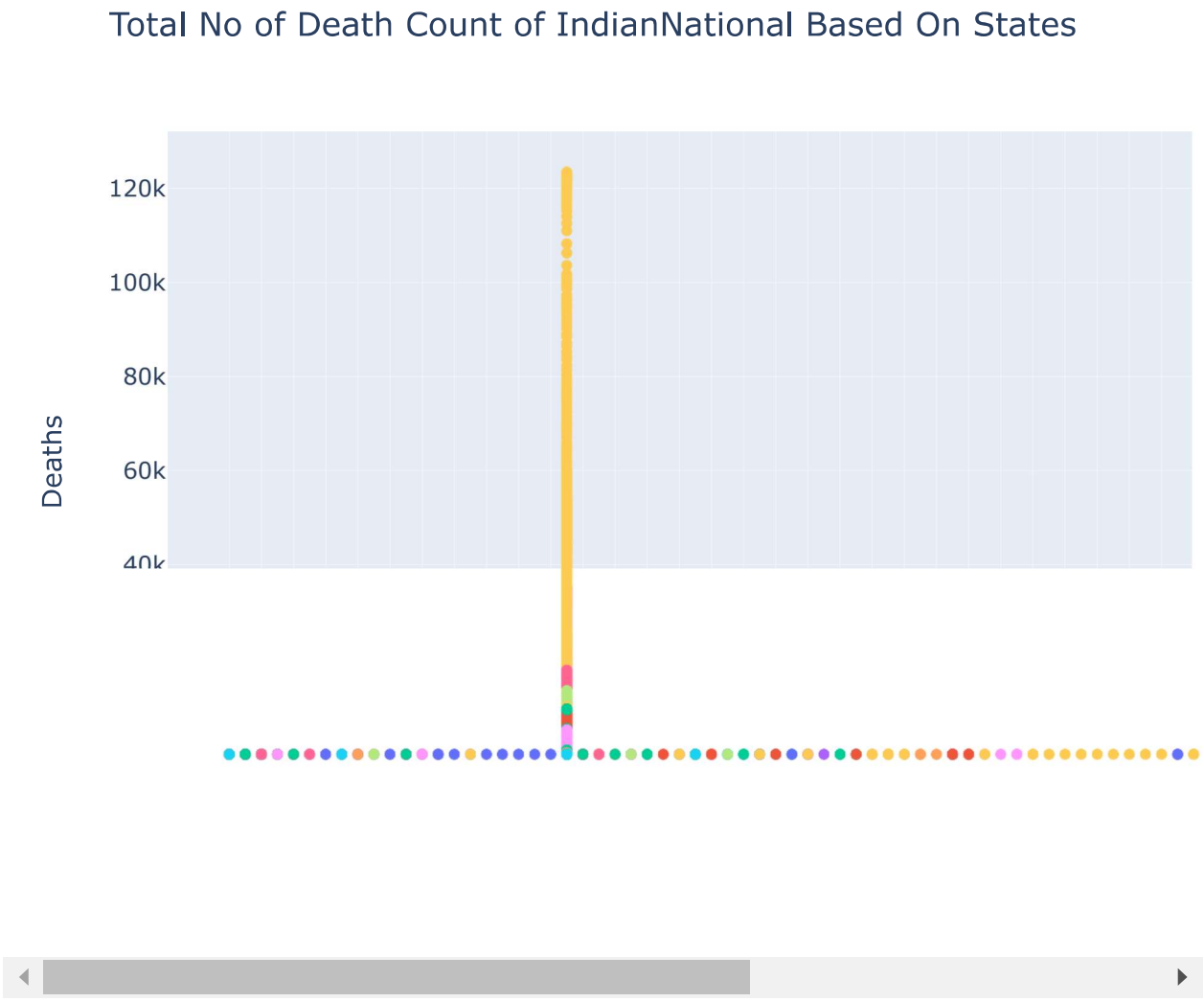
In [36]:
```
1 px.pie( data3 , names = 'State/UnionTerritory', values = 'ConfirmedIndianNat
```



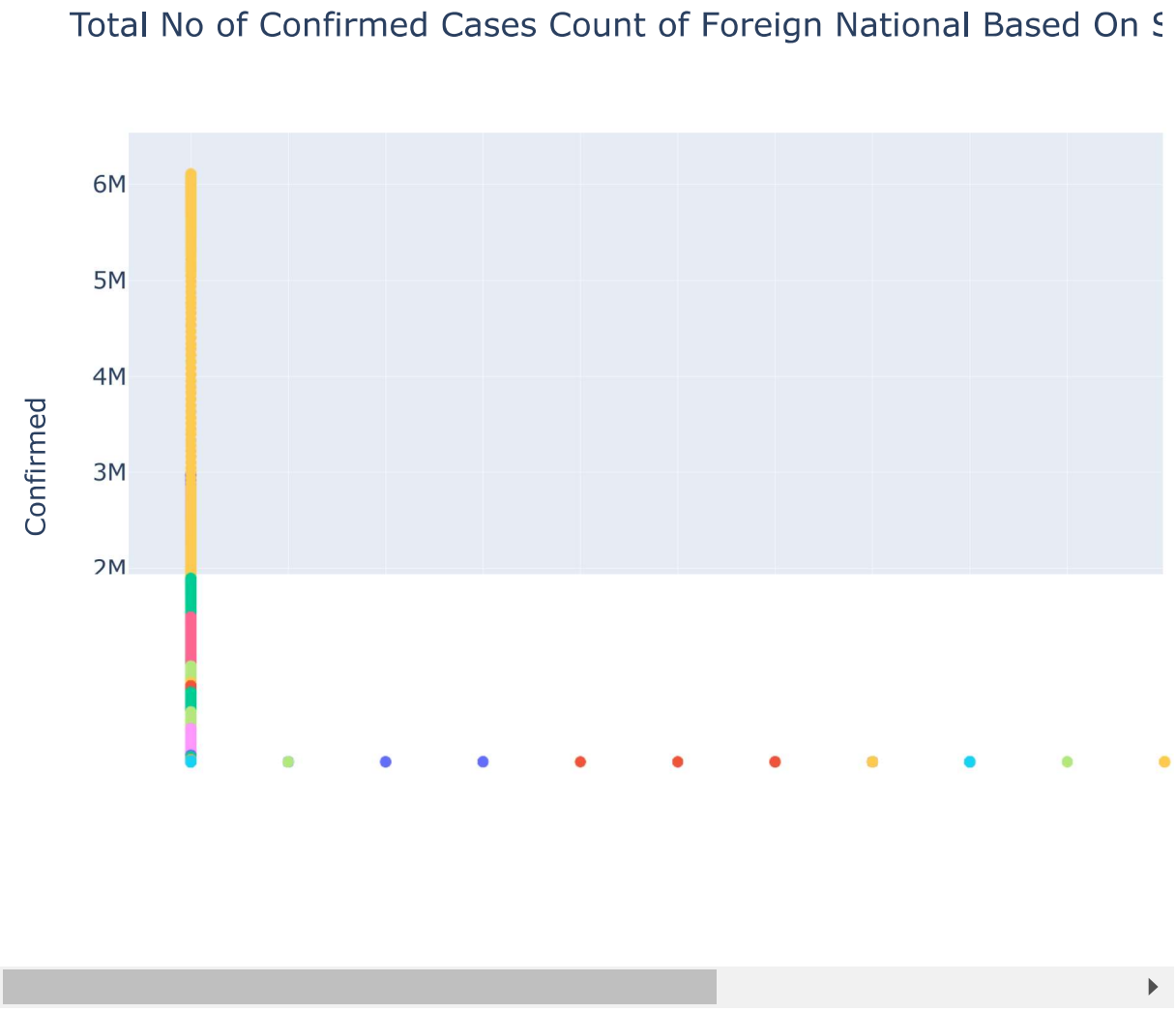
In [37]:
```
1 px.scatter(x = 'ConfirmedIndianNational', y = 'ConfirmedForeignNational', da
```



In [38]:
```
1 px.scatter(x = 'ConfirmedIndianNational', y = 'Deaths', data_frame = data3,
```



In [39]:
```
1 px.scatter(x = 'ConfirmedForeignNational', y = 'Confirmed', data_frame = dat
```



In [40]:
```
1 statewise = data4[data4['Deaths']>100000]
```

In [41]:
```
1 statewise
```

Out[41]:

	State/UnionTerritory	Cured	Deaths	Confirmed
1	Andhra Pradesh	303427899	2475816	324146783
3	Assam	74011348	459575	80418492
4	Bihar	101533848	775163	108312449
5	Chandigarh	7980284	119356	8691806
6	Chhattisgarh	117163544	1591126	128751782
8	Delhi	224062704	4066907	236972842
9	Goa	20224042	338359	22280065
10	Gujarat	103995131	1866811	114557615
11	Haryana	100010131	1166573	107408371
12	Himachal Pradesh	20682770	371931	23052151
13	Jammu and Kashmir	42295048	686680	46899925
14	Jharkhand	46083978	569298	49971564
15	Karnataka	345648926	4819018	387597335
16	Kerala	311127643	1327754	344319045
19	Madhya Pradesh	100169697	1427780	108712983
20	Maharashtra	813788907	19314532	908892470
21	Manipur	8420223	122089	9440912
25	Odisha	117984789	600149	126408397
26	Puducherry	14376916	249683	15858688
27	Punjab	71108712	2216735	78999515
28	Rajasthan	117312772	1159823	128998101
30	Tamil Nadu	317067499	4731627	342829697
31	Telangana	100211245	617882	108152726
32	Tripura	10479169	124444	11397656
33	Uttar Pradesh	232529439	3347656	252843682
34	Uttarakhand	36684388	728512	41179396
35	West Bengal	195296839	3214840	209822848

In [42]:
```
top_ten = statewise[['State/UnionTerritory','Deaths']]
top_ten
```

Out[42]:

	State/UnionTerritory	Deaths
1	Andhra Pradesh	2475816
3	Assam	459575
4	Bihar	775163
5	Chandigarh	119356
6	Chhattisgarh	1591126
8	Delhi	4066907
9	Goa	338359
10	Gujarat	1866811
11	Haryana	1166573
12	Himachal Pradesh	371931
13	Jammu and Kashmir	686680
14	Jharkhand	569298
15	Karnataka	4819018
16	Kerala	1327754
19	Madhya Pradesh	1427780
20	Maharashtra	19314532
21	Manipur	122089
25	Odisha	600149
26	Puducherry	249683
27	Punjab	2216735
28	Rajasthan	1159823
30	Tamil Nadu	4731627
31	Telangana	617882
32	Tripura	124444
33	Uttar Pradesh	3347656
34	Uttarakhand	728512
35	West Bengal	3214840

In [43]:
```
top_ten1 = top_ten.sort_values(by='Deaths', ascending=False)
top_ten1 = top_ten1[0:11]
top_ten1
```

Out[43]:

	State/UnionTerritory	Deaths
20	Maharashtra	19314532
15	Karnataka	4819018
30	Tamil Nadu	4731627
8	Delhi	4066907
33	Uttar Pradesh	3347656
35	West Bengal	3214840
1	Andhra Pradesh	2475816
27	Punjab	2216735
10	Gujarat	1866811
6	Chhattisgarh	1591126
19	Madhya Pradesh	1427780

In [44]:
```
statewise1 = data4[data4["Deaths"] < 100000]
```

In [45]:
```
statewise1
```

Out[45]:

	State/UnionTerritory	Cured	Deaths	Confirmed
0	Andaman and Nicobar Islands	1589935	22624	1675248
2	Arunachal Pradesh	5150519	19303	5598324
7	Dadra and Nagar Haveli and Daman and Diu	1491338	882	1587570
17	Ladakh	3059045	38578	3344131
18	Lakshadweep	471712	2178	561459
22	Meghalaya	4606548	66293	5221064
23	Mizoram	1534630	5073	1822190
24	Nagaland	3628619	39420	4089547
29	Sikkim	1983899	41530	2315519

In [46]:
```
least_ten = statewise1[['State/UnionTerritory','Deaths']]
least_ten
```

Out[46]:

	State/UnionTerritory	Deaths
0	Andaman and Nicobar Islands	22624
2	Arunachal Pradesh	19303
7	Dadra and Nagar Haveli and Daman and Diu	882
17	Ladakh	38578
18	Lakshadweep	2178
22	Meghalaya	66293
23	Mizoram	5073
24	Nagaland	39420
29	Sikkim	41530

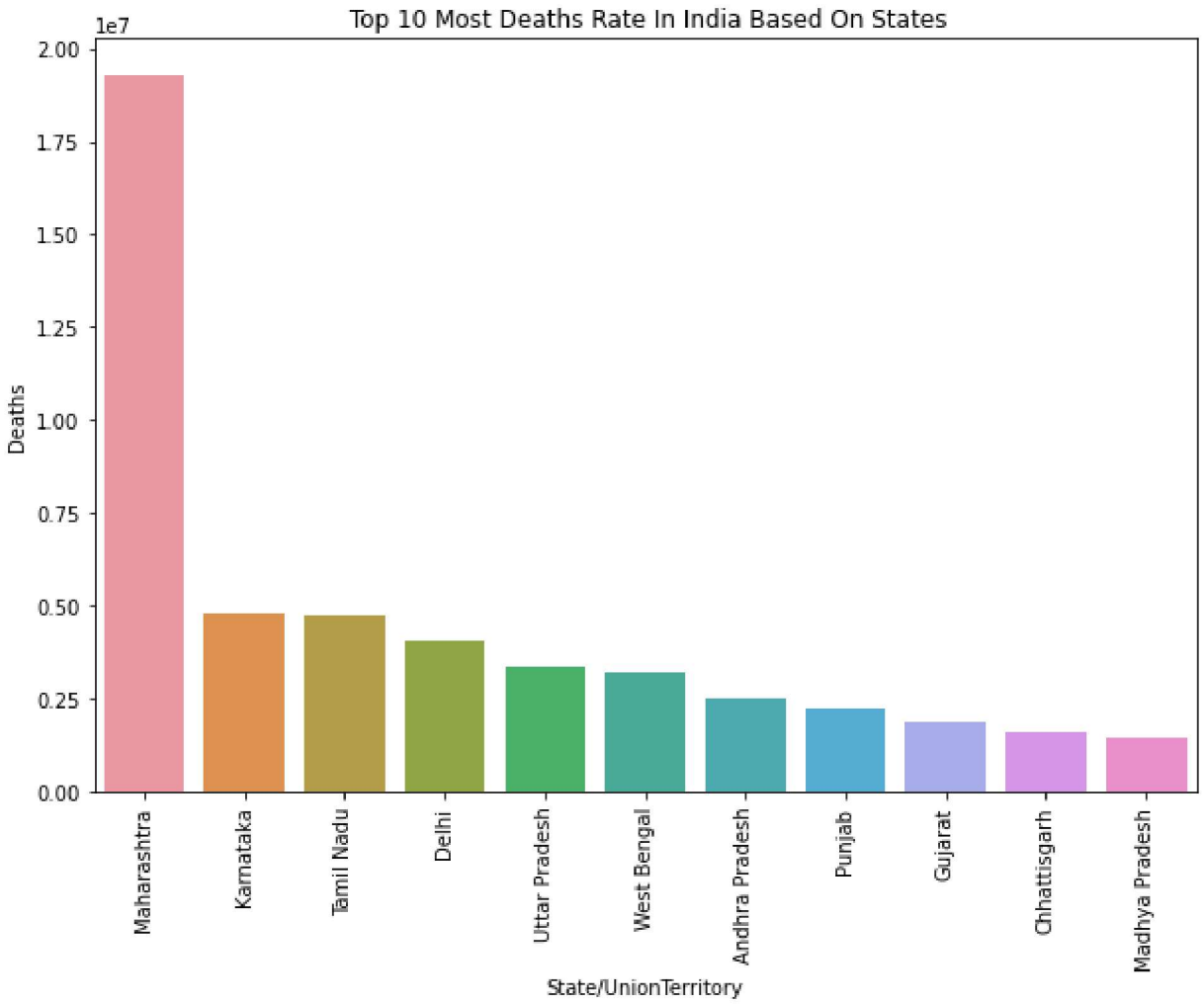
In [47]:
```
least_ten1 = least_ten.sort_values(by='Deaths', ascending=True)
least_ten1 = least_ten1[0:11]
least_ten1
```

Out[47]:

	State/UnionTerritory	Deaths
7	Dadra and Nagar Haveli and Daman and Diu	882
18	Lakshadweep	2178
23	Mizoram	5073
2	Arunachal Pradesh	19303
0	Andaman and Nicobar Islands	22624
17	Ladakh	38578
24	Nagaland	39420
29	Sikkim	41530
22	Meghalaya	66293

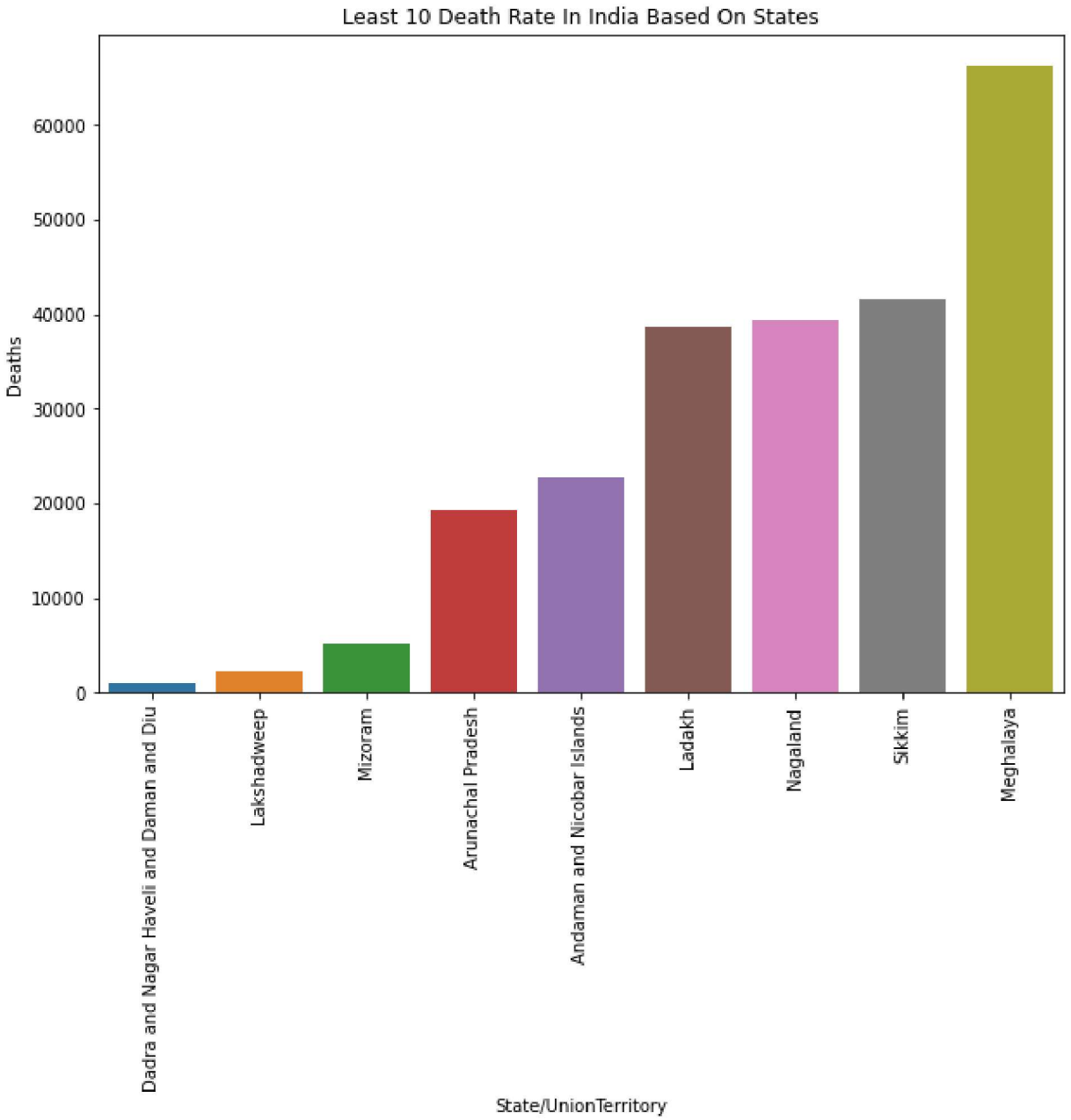
In [48]:
```
fig = plt.figure()
fig = plt.figure(figsize=(10,7))
sns.barplot(x = 'State/UnionTerritory', y ='Deaths', data = top_ten1).set(ti
plt.xticks(rotation = 90)
plt.show()

```

```
<Figure size 432x288 with 0 Axes>
```



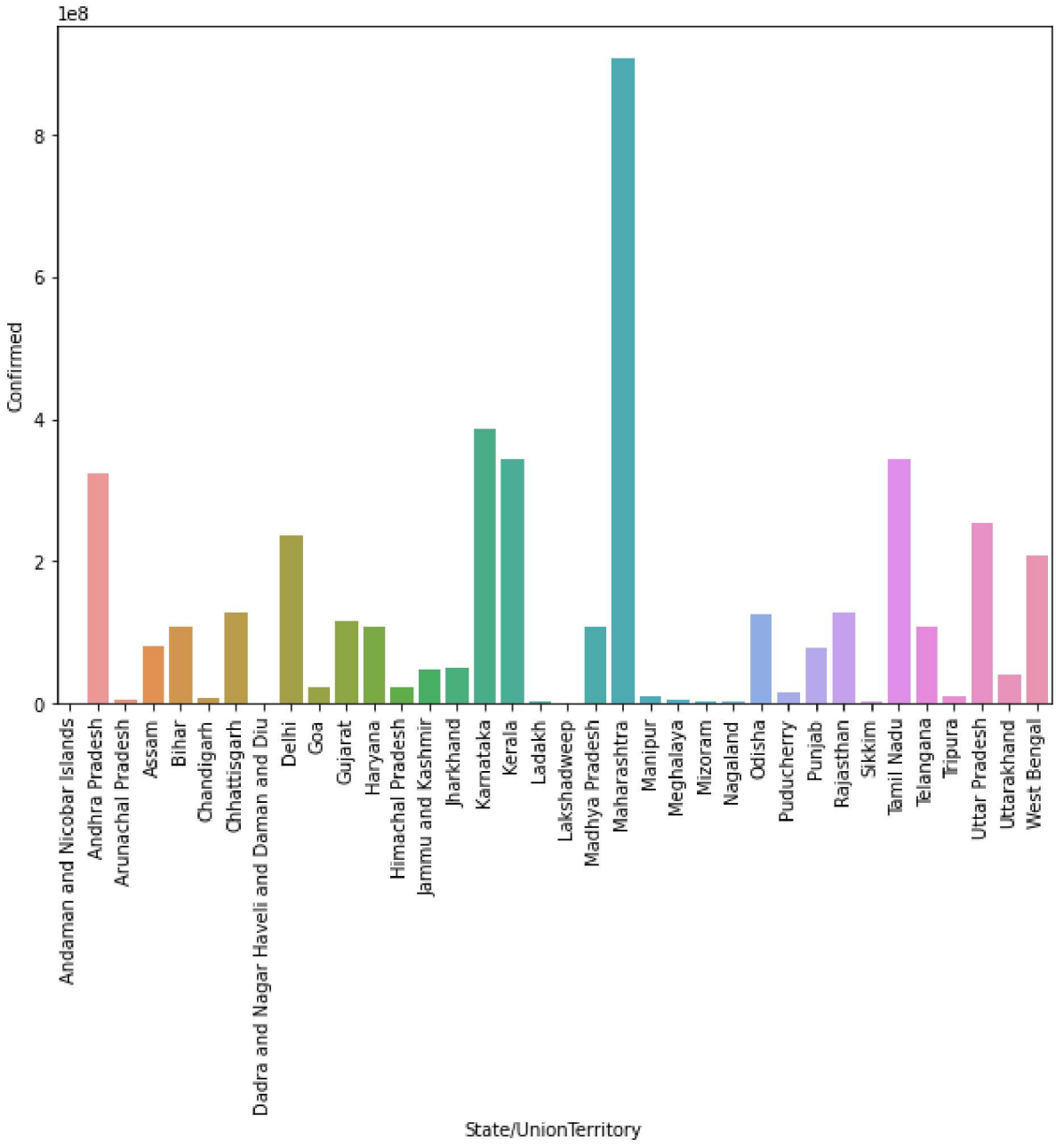
In [49]:
```
fig = plt.figure()
fig = plt.figure(figsize=(10,7))
sns.barplot(x = 'State/UnionTerritory', y = 'Deaths', data = least_ten1).set
plt.xticks(rotation = 90)
plt.show()

```

```
<Figure size 432x288 with 0 Axes>
```



In [50]:
```
fig = plt.figure()
fig = plt.figure(figsize=(10,7))
wer = sns.barplot(x = 'State/UnionTerritory', y = 'Confirmed', data = data4)
plt.setp(wer.get_xticklabels(), rotation = 90)
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Dealing With Dates

In [51]:
```
1 data3['Date'] = pd.to_datetime(data3['Date'])
```

In [52]:
```
1 data3['Date']
```

Out[52]:
```
0       2020-01-30
1       2020-01-31
2       2020-02-01
3       2020-02-02
4       2020-02-03
           ...
16845   2021-07-07
16846   2021-07-07
16847   2021-07-07
16848   2021-07-07
16849   2021-07-07
Name: Date, Length: 16787, dtype: datetime64[ns]
```

In [53]:
```
1 data3['Year'] = data3['Date'].dt.year
```

In [54]:
```
1 data3['Month'] = data3['Date'].dt.month
```

In [55]:
```
1 data3['Day'] = data3['Date'].dt.day
```

In [56]:
```
1 data3['Weekday'] = data3['Date'].dt.weekday
```

In [57]:
```
1 data3
```

Out[57]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational
0	1	2020-01-30	6:00 PM	Kerala	1	0
1	2	2020-01-31	6:00 PM	Kerala	1	0
2	3	2020-02-01	6:00 PM	Kerala	2	0
3	4	2020-02-02	6:00 PM	Kerala	3	0
4	5	2020-02-03	6:00 PM	Kerala	3	0
...	...	...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	0	0
16846	16847	2021-07-07	8:00 AM	Tripura	0	0
16847	16848	2021-07-07	8:00 AM	Uttarakhand	0	0
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	0	0
16849	16850	2021-07-07	8:00 AM	West Bengal	0	0

16787 rows × 13 columns

In [58]:
```
1 data3
```

Out[58]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational
0	1	2020-01-30	6:00 PM	Kerala	1	0
1	2	2020-01-31	6:00 PM	Kerala	1	0
2	3	2020-02-01	6:00 PM	Kerala	2	0
3	4	2020-02-02	6:00 PM	Kerala	3	0
4	5	2020-02-03	6:00 PM	Kerala	3	0
...	...	...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	0	0
16846	16847	2021-07-07	8:00 AM	Tripura	0	0
16847	16848	2021-07-07	8:00 AM	Uttarakhand	0	0
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	0	0
16849	16850	2021-07-07	8:00 AM	West Bengal	0	0

16787 rows × 13 columns

State/Union Territory Wise Total Confirmed, Cured, Deaths Cases

In [59]:
```
1 data3.groupby('State/UnionTerritory')[['Cured', 'Deaths', 'Confirmed']].max(
```

Out[59]:

State/UnionTerritory	Cured	Deaths	Confirmed
Maharashtra	5872268	123531	6113335
Kerala	2877557	13960	2996094
Karnataka	2784030	35526	2859595
Tamil Nadu	2435872	33132	2503481
Andhra Pradesh	1861937	12898	1908065
Uttar Pradesh	1682130	22656	1706818
West Bengal	1472132	17834	1507241
Delhi	1408853	25001	1434687
Chhattisgarh	977893	13462	996359
Rajasthan	942882	8942	952836
Odisha	897362	4299	927186
Gujarat	811699	10072	823964
Madhya Pradesh	780578	9017	790042
Haryana	758442	9506	769030
Bihar	711913	9612	722746
Telangana	613124	3703	628282
Punjab	578590	16131	596736
Assam	493306	4717	522267
Jharkhand	340365	5118	346038
Uttarakhand	332006	7338	340882
Jammu and Kashmir	309554	4345	317481
Himachal Pradesh	198134	3485	202945
Goa	162787	3079	167823
Puducherry	114673	1763	118227
Manipur	66132	1218	73581
Tripura	63964	701	68612
Chandigarh	60837	809	61752
Meghalaya	47173	880	52358
Arunachal Pradesh	34525	181	37879
Nagaland	23982	503	25619
Mizoram	18383	98	22155
Sikkim	19200	309	21403
Ladakh	19733	204	20137

State/UnionTerritory	Cured	Deaths	Confirmed
Dadra and Nagar Haveli and Daman and Diu	10532	4	10575
Lakshadweep	9643	49	9947
Andaman and Nicobar Islands	7343	128	7487

Total Covid-19 Cases In India

In [60]:
```
total_cases = data3.groupby('State/UnionTerritory')['Confirmed'].max().sum()
```

In [61]:
```
total_cured = data3.groupby('State/UnionTerritory')['Cured'].max().sum()
```

In [62]:
```
total_deaths = data3.groupby('State/UnionTerritory')['Deaths'].max().sum()

```

In [63]:
```
active_cases = total_cases - (total_cured + total_deaths)
```

In [64]:
```
print(f"Total Covid-19 Confirmed Cases in India: {total_cases}")
print(f"Total Covid-19 Cured Cases in India: {total_cured}")
print(f"Total Covid-19 Deaths in India: {total_deaths}")
print(f"Total Covid-19 Active Cases in India: {active_cases}")
```

```
Total Covid-19 Confirmed Cases in India: 30663665
Total Covid-19 Cured Cases in India: 29799534
Total Covid-19 Deaths in India: 404211
Total Covid-19 Active Cases in India: 459920
```

In [65]:
```
recovery_rate = (total_cured / total_cases) * 100
```

In [66]:
```
death_rate = (total_deaths / total_cases) * 100
```

In [67]:
```
active_rate = (active_cases / total_cases) * 100
```

In [68]:
```
print('Recovery rate is {:.2f} %'.format(recovery_rate))
print('Death rate is {:.2f} %'.format(death_rate))
print('Active rate is {:.2f} %'.format(active_rate))
```

```
Recovery rate is 97.18 %
Death rate is 1.32 %
Active rate is 1.50 %
```

Total Cases In May 2021

In [69]:
```
april_2021 = data3[(data3['Year'] == 2021) & (data3['Month'] == 4)].groupby(
```

In [70]:
```
may_2021 = data3[(data3['Year'] == 2021) & (data3['Month'] == 5)].groupby('S
```

In [71]:
```
confirmed = (may_2021['Confirmed'] - april_2021['Confirmed']).sum()
```

In [72]:
```
cured = (may_2021['Cured'] - april_2021['Cured']).sum()
```

In [73]:
```
deaths = (may_2021['Deaths'] - april_2021['Deaths']).sum()
```

In [74]:
```
print(f"Total Covid-19 Confirmed Cases in May 2021: {confirmed}")
print(f"Total Covid-19 Cured Cases in May 2021: {cured}")
print(f"Total Covid-19 Deaths in May 2021: {deaths}")
```

```
Total Covid-19 Confirmed Cases in May 2021: 9282634
Total Covid-19 Cured Cases in May 2021: 10307924
Total Covid-19 Deaths in May 2021: 120770
```

Cases In Maharashtra

In [75]:
```
maharashtra_df = data3[data3['State/UnionTerritory'] == 'Maharashtra'].group
```

```
C:\Users\NIKHIL\AppData\Local\Temp/ipykernel_11904/1650171465.py:1: FutureWarni
ng:

Dropping invalid columns in DataFrameGroupBy.max is deprecated. In a future ver
sion, a TypeError will be raised. Before calling .max, select only columns whic
h should be valid for the function.
```

In [76]:
```
maharashtra_df
```

Out[76]:

	Year	Month	Sno	Date	Time	State/UnionTerritory	Cured	Deaths	Confirmed	Day	Wee
0	2020	3	517	2020-03-31	9:30 PM	Maharashtra	39	9	216	31	
1	2020	4	1465	2020-04-30	7:30 PM	Maharashtra	1593	432	9915	30	
2	2020	5	2506	2020-05-31	8:00 AM	Maharashtra	28081	2197	65168	31	
3	2020	6	3587	2020-06-30	8:00 AM	Maharashtra	88960	7610	169883	30	
4	2020	7	4691	2020-07-31	8:00 AM	Maharashtra	248615	14729	411798	31	
5	2020	8	5776	2020-08-31	8:00 AM	Maharashtra	562401	24399	780689	31	
6	2020	9	6826	2020-09-30	8:00 AM	Maharashtra	1069159	36181	1366129	30	
7	2020	10	7911	2020-10-31	8:00 AM	Maharashtra	1503050	43837	1672858	31	
8	2020	11	8961	2020-11-30	8:00 AM	Maharashtra	1680926	47071	1820059	30	
9	2020	12	10067	2020-12-31	8:00 AM	Maharashtra	1824934	49463	1928603	31	
10	2021	1	11183	2021-01-31	8:00 AM	Maharashtra	1927335	51042	2023814	31	
11	2021	2	12191	2021-02-28	8:00 AM	Maharashtra	2020951	52092	2146777	28	
12	2021	3	13307	2021-03-31	8:00 AM	Maharashtra	2377127	54422	2773436	31	
13	2021	4	14387	2021-04-30	8:00 AM	Maharashtra	3799266	67985	4539553	30	
14	2021	5	15503	2021-05-31	8:00 AM	Maharashtra	5362370	94844	5731815	31	
15	2021	6	16583	2021-06-30	8:00 AM	Maharashtra	5809548	121804	6051633	30	
16	2021	7	16835	2021-07-07	8:00 AM	Maharashtra	5872268	123531	6113335	7	

In [77]:
```
def monthly(series):
    i = 0
    res = []
    for data in series:
        res.append(data - i)
        i = data
    return pd.Series(res)
```

In [78]:
```
maharashtra_df['Confirmed'] = monthly(maharashtra_df['Confirmed'])
```

In [79]:
```
maharashtra_df['Cured'] = monthly(maharashtra_df['Cured'])
```

In [80]:
```
maharashtra_df['Deaths'] = monthly(maharashtra_df['Deaths'])
```

In [81]:
```
maharashtra_df
```

Out[81]:

	Year	Month	Sno	Date	Time	State/UnionTerritory	Cured	Deaths	Confirmed	Day	Wee
0	2020	3	517	2020-03-31	9:30 PM	Maharashtra	39	9	216	31	
1	2020	4	1465	2020-04-30	7:30 PM	Maharashtra	1554	423	9699	30	
2	2020	5	2506	2020-05-31	8:00 AM	Maharashtra	26488	1765	55253	31	
3	2020	6	3587	2020-06-30	8:00 AM	Maharashtra	60879	5413	104715	30	
4	2020	7	4691	2020-07-31	8:00 AM	Maharashtra	159655	7119	241915	31	
5	2020	8	5776	2020-08-31	8:00 AM	Maharashtra	313786	9670	368891	31	
6	2020	9	6826	2020-09-30	8:00 AM	Maharashtra	506758	11782	585440	30	
7	2020	10	7911	2020-10-31	8:00 AM	Maharashtra	433891	7656	306729	31	
8	2020	11	8961	2020-11-30	8:00 AM	Maharashtra	177876	3234	147201	30	
9	2020	12	10067	2020-12-31	8:00 AM	Maharashtra	144008	2392	108544	31	
10	2021	1	11183	2021-01-31	8:00 AM	Maharashtra	102401	1579	95211	31	
11	2021	2	12191	2021-02-28	8:00 AM	Maharashtra	93616	1050	122963	28	
12	2021	3	13307	2021-03-31	8:00 AM	Maharashtra	356176	2330	626659	31	
13	2021	4	14387	2021-04-30	8:00 AM	Maharashtra	1422139	13563	1766117	30	
14	2021	5	15503	2021-05-31	8:00 AM	Maharashtra	1563104	26859	1192262	31	
15	2021	6	16583	2021-06-30	8:00 AM	Maharashtra	447178	26960	319818	30	
16	2021	7	16835	2021-07-07	8:00 AM	Maharashtra	62720	1727	61702	7	

Total Covid-19 Cases In Maharashtra

In [82]:
```
mh_total_confirmed = maharashtra_df['Confirmed'].sum()
```

In [83]:
```
mh_total_cured = maharashtra_df['Cured'].sum()
```

In [84]:
```
mh_total_deaths = maharashtra_df['Deaths'].sum()
```

In [85]:
```
mh_active_cases = mh_total_confirmed - (mh_total_cured + mh_total_deaths)
```

In [86]:
```
print(f"Total Covid-19 Confirmed Cases in Maharashtra: {mh_total_confirmed}"
print(f"Total Covid-19 Cured Cases in Maharashtra: {mh_total_cured}")
print(f"Total Covid-19 Deaths in Maharashtra: {mh_total_deaths}")
print(f"Total Covid-19 Active Cases in Maharashtra: {mh_active_cases}")
```

```
Total Covid-19 Confirmed Cases in Maharashtra: 6113335
Total Covid-19 Cured Cases in Maharashtra: 5872268
Total Covid-19 Deaths in Maharashtra: 123531
Total Covid-19 Active Cases in Maharashtra: 117536
```

In [87]:
```
mh_recovery_rate = (mh_total_cured / mh_total_confirmed) * 100
```

In [88]:
```
mh_death_rate = (mh_total_deaths / mh_total_confirmed) * 100
```

In [89]:
```
mh_active_rate = (mh_active_cases / mh_total_confirmed) * 100
```

In [90]:
```
print('Recovery rate is {:.2f} %'.format(mh_recovery_rate))
print('Death rate is {:.2f} %'.format(mh_death_rate))
print('Active rate is {:.2f} %'.format(mh_active_rate))
```

```
Recovery rate is 96.06 %
Death rate is 2.02 %
Active rate is 1.92 %
```

Total Cases In May 2021 In Maharashtra

In [91]:
```
april = april_2021[april_2021['State/UnionTerritory'] == 'Maharashtra']
```

In [92]:
```
may = may_2021[april_2021['State/UnionTerritory'] == 'Maharashtra']
```

In [93]:
```
may_confirmed = may['Confirmed'] - april['Confirmed']
```

In [94]:
```
may_cured = may['Cured'] - april['Cured']
```

In [95]:
```
may_deaths = may['Deaths'] - april['Deaths']
```

In [96]:
```
print(f"Confirmed Cases in May 2021: {may_confirmed.loc[20]}")
print(f"Cured Cases in May 2021: {may_cured.loc[20]}")
print(f"Deaths in May 2021: {may_deaths.loc[20]}")
```

```
Confirmed Cases in May 2021: 1192262
Cured Cases in May 2021: 1563104
Deaths in May 2021: 26859
```

Data Visualization

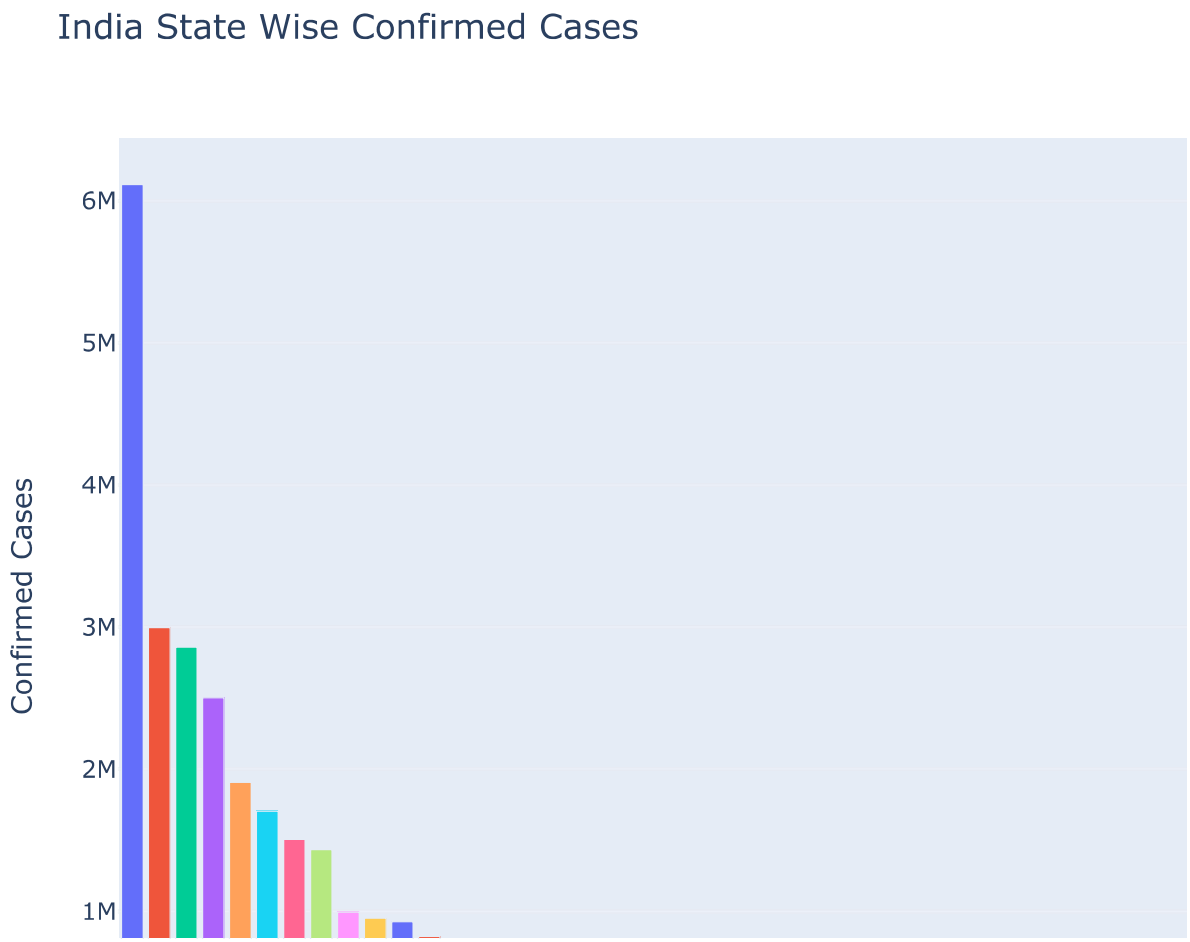
In [ ]:
```

```

India Statewise Cases

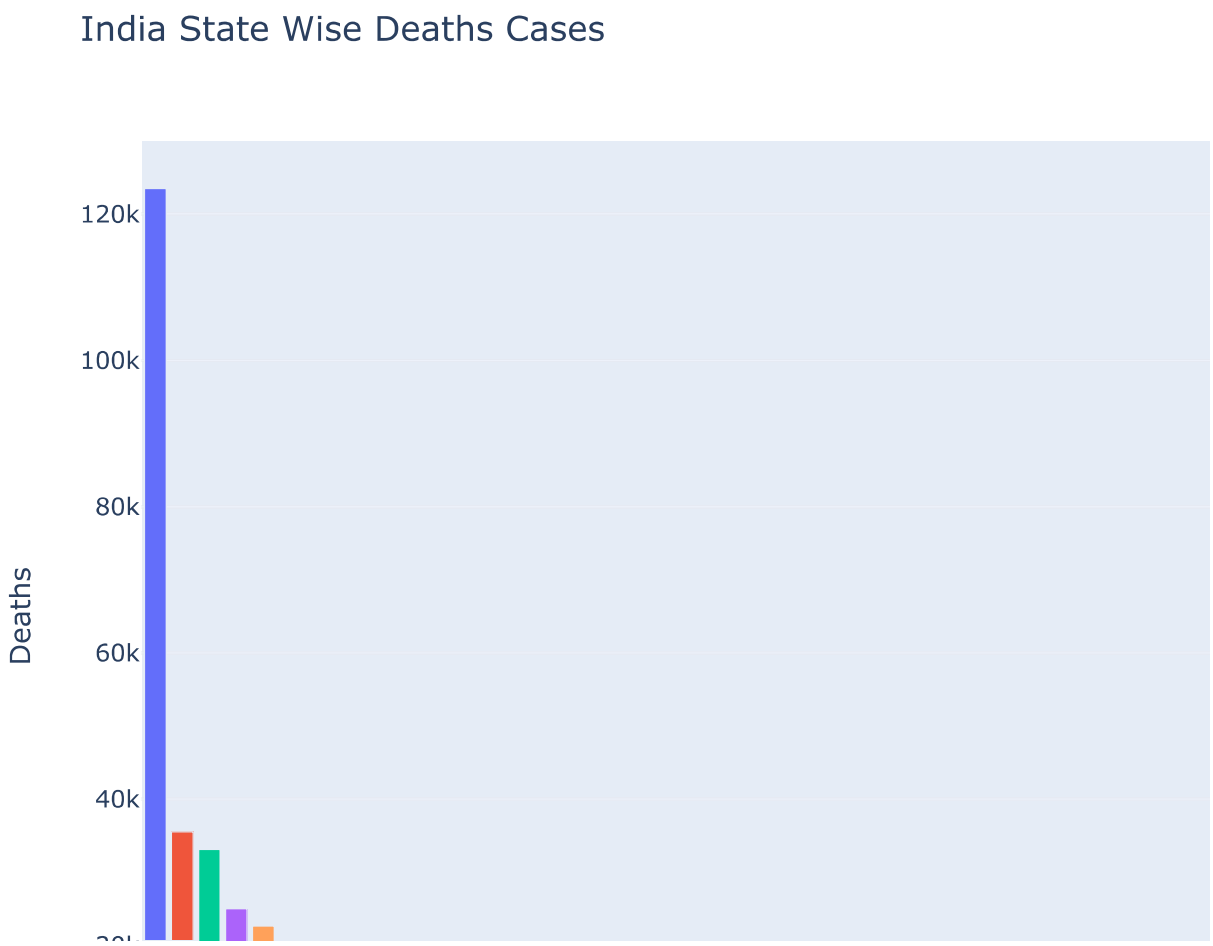
In [97]:
```
statewise = dataset.groupby('State/UnionTerritory')[['Cured', 'Deaths', 'Con
```

In [98]:
```
statewise = statewise.reset_index()
```

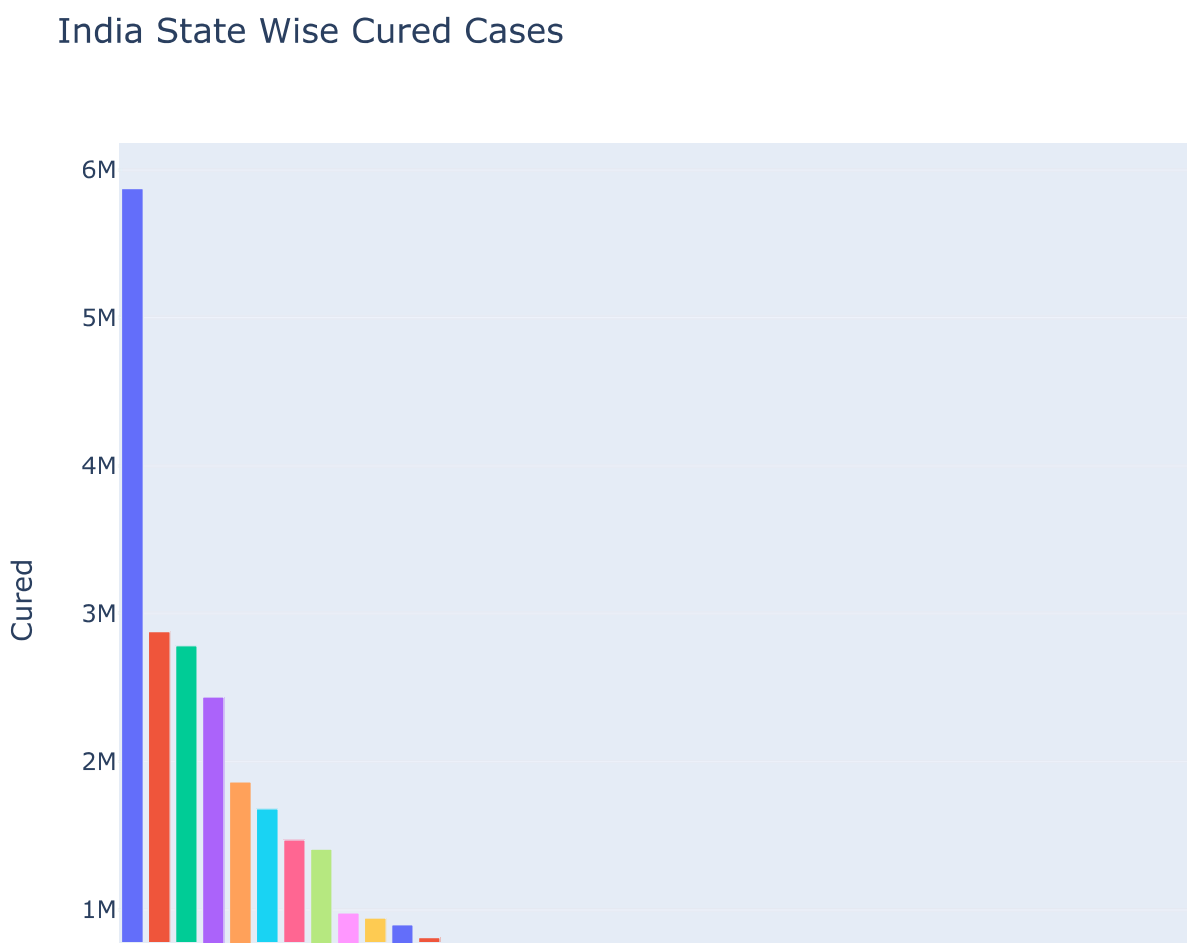
In [99]:
```
px.bar(statewise,x='State/UnionTerritory', y='Confirmed',
       title="India State Wise Confirmed Cases",
       labels={'Confirmed':'Confirmed Cases'},
      color='State/UnionTerritory',
      height=800)
```



In [100]:
```
px.bar(statewise.sort_values(by='Deaths',ascending=False), x='State/UnionTer
       title="India State Wise Deaths Cases",
      color='State/UnionTerritory',
      height=800)
```

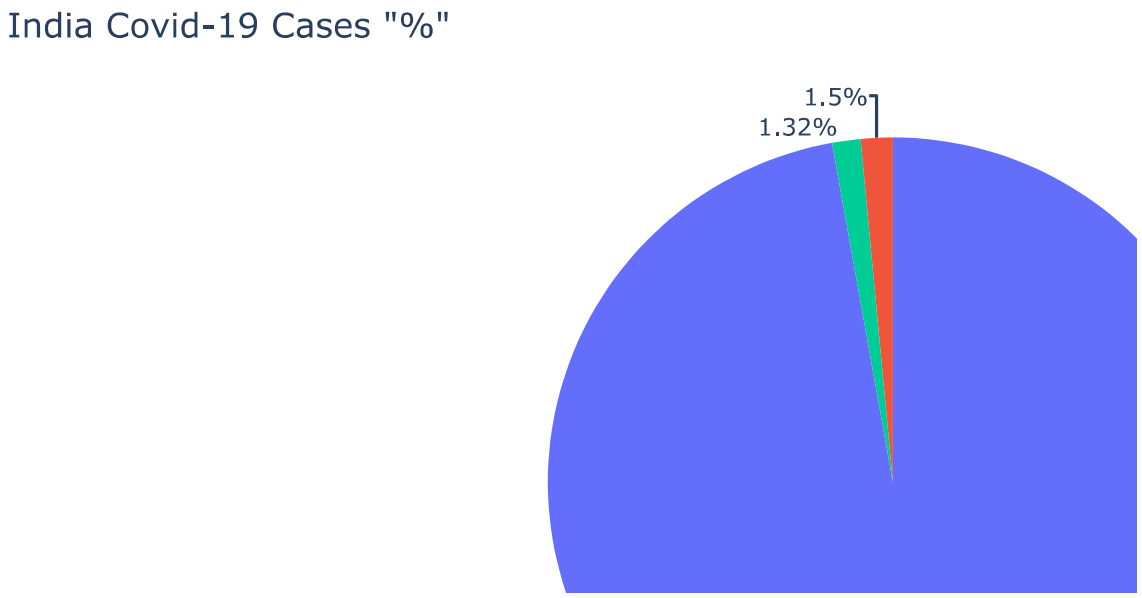


In [101]:
```
px.bar(statewise, x='State/UnionTerritory', y='Cured',
       title="India State Wise Cured Cases",
      color='State/UnionTerritory',
      height=800)
```

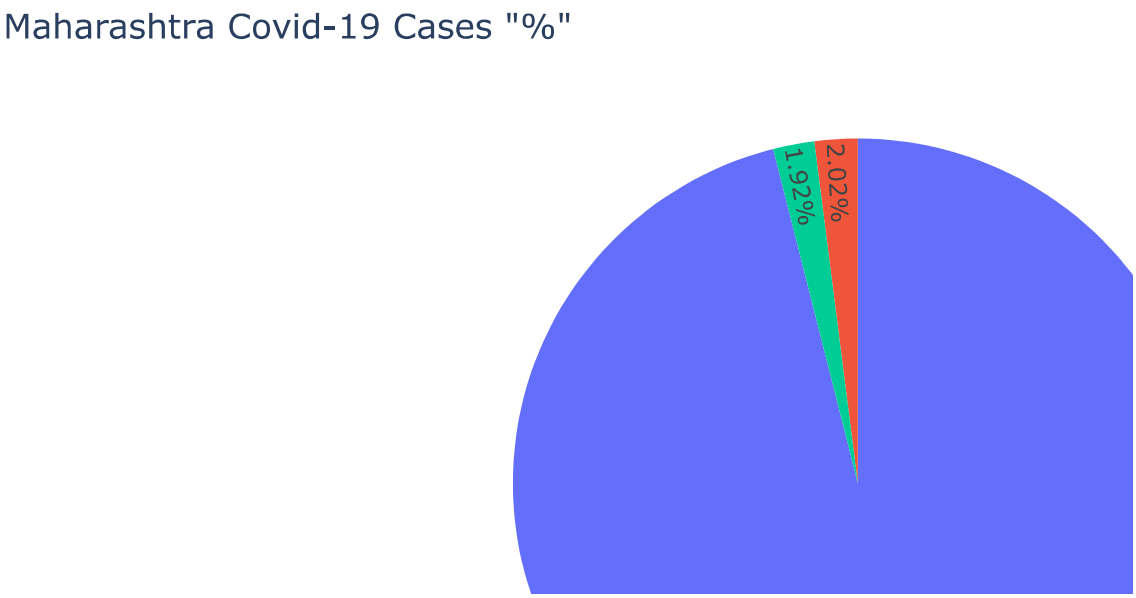


Cured/Death/Active

In [102]:
```
px.pie(names=['Cured','Death','Active'],
       values=[recovery_rate, death_rate, active_rate],
       title='India Covid-19 Cases "%"')
```



In [103]:
```
px.pie(names=['Cured','Death','Active'],
       values=[mh_recovery_rate, mh_death_rate, mh_active_rate],
       title='Maharashtra Covid-19 Cases "%"')
```

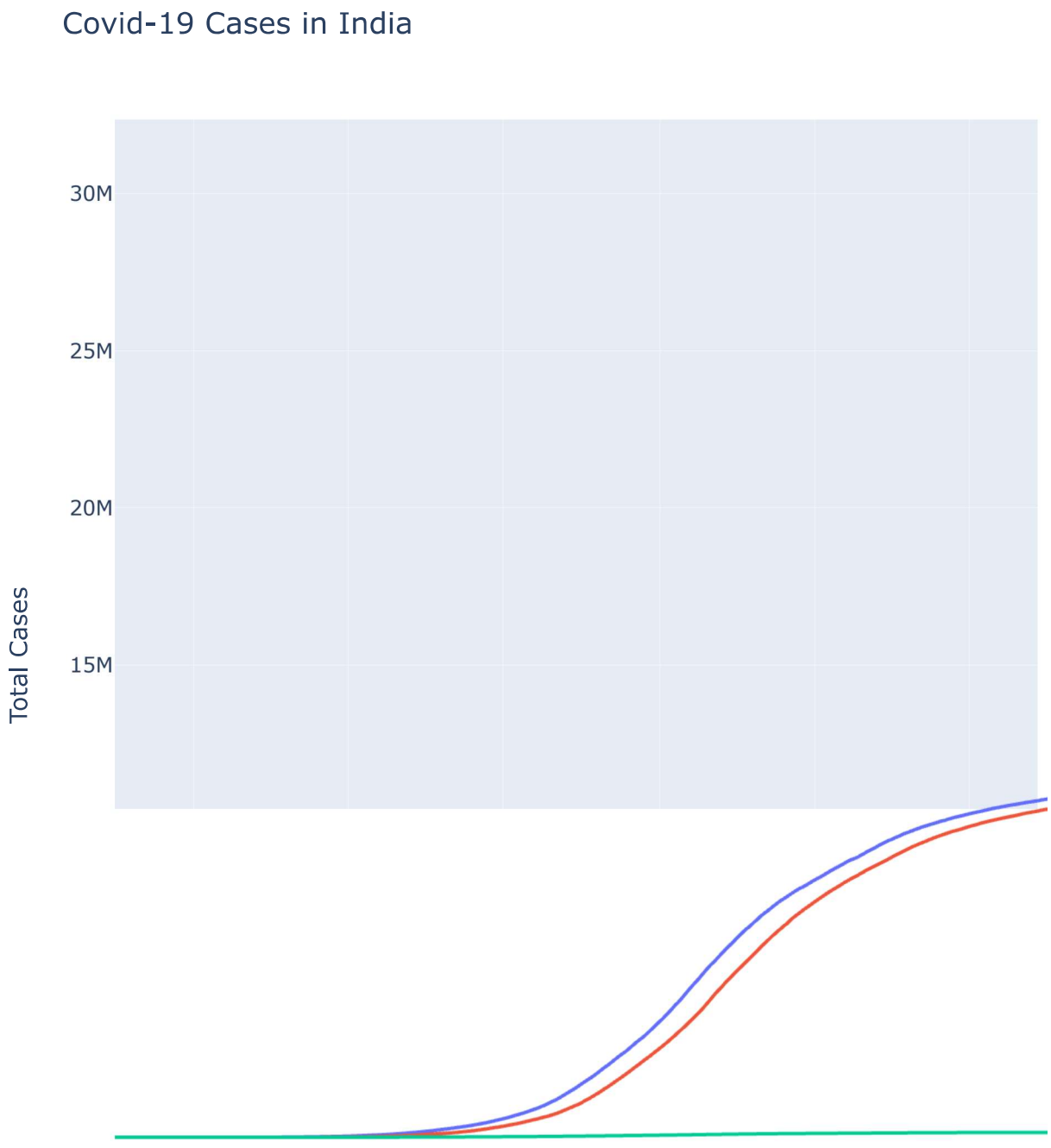


Time Series Analysis

Cases In India

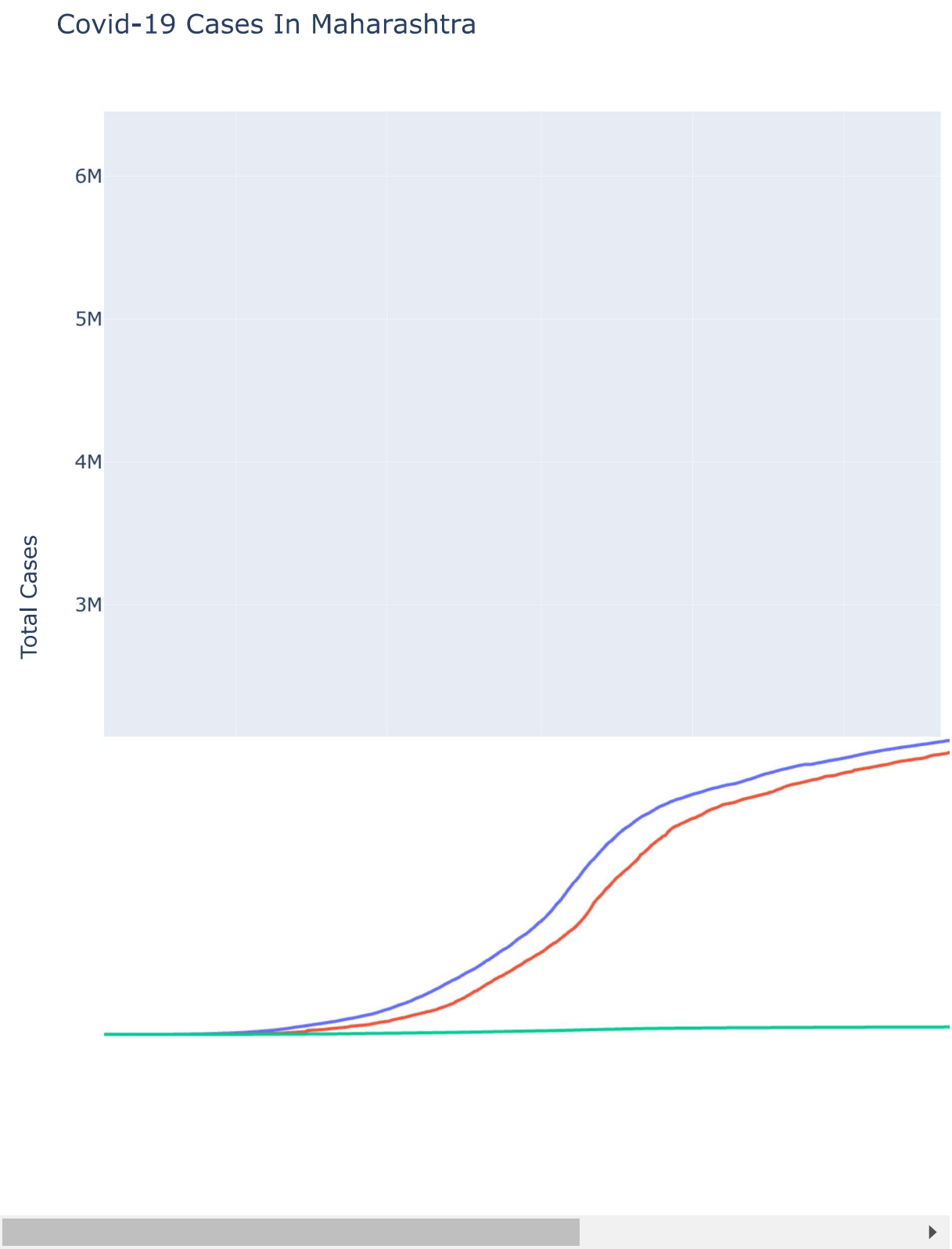
In [104]:
```
px.line(dataset.groupby('Date')['Confirmed','Cured','Deaths'].sum().reset_in
        x='Date', y=['Confirmed','Cured','Deaths'],
        labels={'value':'Total Cases'},
        title='Covid-19 Cases in India', height=800)

```

```
C:\Users\NIKHIL\AppData\Local\Temp/ipykernel_11904/1476617511.py:1: FutureWarni
ng:

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



Cases In Maharashtra

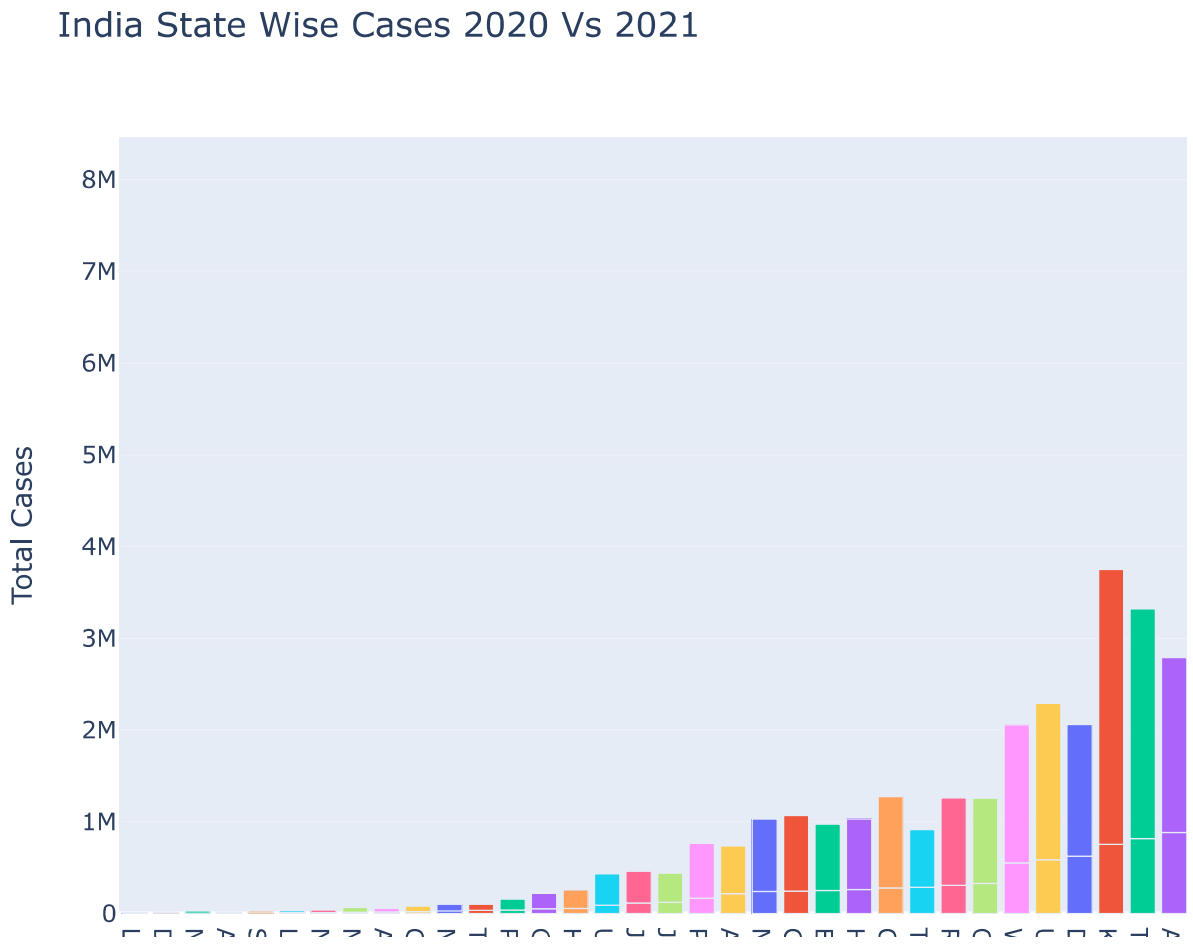
In [105]:
```
px.line(data_frame=dataset[dataset['State/UnionTerritory'] == 'Maharashtra']
        x='Date', y=['Confirmed','Cured','Deaths'],
        labels={'value':'Total Cases'},
       height=800,title='Covid-19 Cases In Maharashtra')
```



Covid Cases In 2020 Vs 2021

In [106]:
```
1 yearly_cases = data3.groupby(['Year','State/UnionTerritory'])[['Deaths', 'Co
```

In [107]:
```
px.bar(yearly_cases, x='State/UnionTerritory', y='Confirmed',
      color='State/UnionTerritory',
      hover_name='Year', height=800,
      labels={'Confirmed':'Total Cases'},
      title="India State Wise Cases 2020 Vs 2021"
      )
```



In [108]:
```
px.bar(yearly_cases, x='State/UnionTerritory', y='Deaths',
       color='State/UnionTerritory',
       hover_name='Year', height=800,
       labels={'Deaths':'Total Deaths'},
       title="India State Wise Deaths 2020 Vs 2021"
       )
```