

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from datetime import datetime
```

In [51]:

```
import matplotlib.pyplot as plt
import plotly.express as px
```

In [2]:

```
covid_df=pd.read_csv(r'C:\Users\NIKUNJ\Downloads\covid_19_india.csv')
```

In [3]:

```
df=covid_df.copy()
```

In [4]:

df

Out[4]:

| | Sno | Date | Time | State/UnionTerritory | ConfirmedIndianNational | ConfirmedForeignNatio |
|-------|-------|------------|---------|----------------------|-------------------------|-----------------------|
| 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 | |
| 3 | 4 | 2020-02-02 | 6:00 PM | Kerala | 3 | |
| 4 | 5 | 2020-02-03 | 6:00 PM | Kerala | 3 | |
| ... | ... | ... | ... | ... | ... | |
| 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 [5]:

df.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 [7]:

```
df.isnull().sum()
```

Out[7]:

| | |
|--------------------------|---|
| Sno | 0 |
| Date | 0 |
| Time | 0 |
| State/UnionTerritory | 0 |
| ConfirmedIndianNational | 0 |
| ConfirmedForeignNational | 0 |
| Cured | 0 |
| Deaths | 0 |
| Confirmed | 0 |
| dtype: int64 | |

In [8]:

```
df.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]:

```
df['ConfirmedIndianNational'].replace('-', '0', inplace=True)
```

In [10]:

```
df['ConfirmedForeignNational'].replace('-', '0', inplace=True)
```

In [11]:

```
df.drop(df[df['State/UnionTerritory']=='Unassigned'].index, inplace=True)
```

In [12]:

```
df.drop(df[df['State/UnionTerritory']=='Cases being reassigned to states'].index, inplace=True)
```

In [13]:

```
df.loc[df["State/UnionTerritory"]=="Bihar***", "State/UnionTerritory"]="Bihar"
```

In [14]:

```
df.loc[df["State/UnionTerritory"]=="Telengana", "State/UnionTerritory"]="Telangana"
```

In [15]:

```
df.loc[df["State/UnionTerritory"]=="Daman & Diu", "State/UnionTerritory"]="Dadra and Nagar
```

In [16]:

```
df.loc[df["State/UnionTerritory"]=="Dadra and Nagar Haveli", "State/UnionTerritory"]="Dadra
```

In [17]:

```
df
```

Out[17]:

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

16787 rows × 9 columns



In [22]:

df.columns

Out[22]:

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

In [23]:

df.drop(['Sno', 'Time', 'ConfirmedIndianNational', 'ConfirmedForeignNational'], inplace=True, axis=1)

In [24]:

df.columns

Out[24]:

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

In [25]:

df

Out[25]:

| | 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 |
| ... | ... | ... | ... | ... | ... |
| 16845 | 2021-07-07 | Telangana | 613124 | 3703 | 628282 |
| 16846 | 2021-07-07 | Tripura | 63964 | 701 | 68612 |
| 16847 | 2021-07-07 | Uttarakhand | 332006 | 7338 | 340882 |
| 16848 | 2021-07-07 | Uttar Pradesh | 1682130 | 22656 | 1706818 |
| 16849 | 2021-07-07 | West Bengal | 1472132 | 17834 | 1507241 |

16787 rows × 5 columns

In [35]:

```
df['State/UnionTerritory'].unique()
```

Out[35]:

```
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 [32]:

```
df['State/UnionTerritory'].nunique()
```

Out[32]:

```
36
```

In [36]:

```
df.max()
```

Out[36]:

```
Date           2021-07-07
State/UnionTerritory    West Bengal
Cured          5872268
Deaths         123531
Confirmed      6113335
dtype: object
```

In [38]:

```
df['Active_case']=df['Confirmed']-(df['Cured']+df['Deaths'])
```

In [39]:

df

Out[39]:

| | Date | State/UnionTerritory | Cured | Deaths | Confirmed | Active_case |
|-------|------------|----------------------|---------|--------|-----------|-------------|
| 0 | 2020-01-30 | Kerala | 0 | 0 | 1 | 1 |
| 1 | 2020-01-31 | Kerala | 0 | 0 | 1 | 1 |
| 2 | 2020-02-01 | Kerala | 0 | 0 | 2 | 2 |
| 3 | 2020-02-02 | Kerala | 0 | 0 | 3 | 3 |
| 4 | 2020-02-03 | Kerala | 0 | 0 | 3 | 3 |
| ... | ... | ... | ... | ... | ... | ... |
| 16845 | 2021-07-07 | Telangana | 613124 | 3703 | 628282 | 11455 |
| 16846 | 2021-07-07 | Tripura | 63964 | 701 | 68612 | 3947 |
| 16847 | 2021-07-07 | Uttarakhand | 332006 | 7338 | 340882 | 1538 |
| 16848 | 2021-07-07 | Uttar Pradesh | 1682130 | 22656 | 1706818 | 2032 |
| 16849 | 2021-07-07 | West Bengal | 1472132 | 17834 | 1507241 | 17275 |

16787 rows × 6 columns

In [40]:

df['Recovery_rate']=(df['Cured']/df['Confirmed'])*100

In [41]:

df

Out[41]:

| | Date | State/UnionTerritory | Cured | Deaths | Confirmed | Active_case | Recovery_rate |
|-------|------------|----------------------|---------|--------|-----------|-------------|---------------|
| 0 | 2020-01-30 | Kerala | 0 | 0 | 1 | 1 | 0.000000 |
| 1 | 2020-01-31 | Kerala | 0 | 0 | 1 | 1 | 0.000000 |
| 2 | 2020-02-01 | Kerala | 0 | 0 | 2 | 2 | 0.000000 |
| 3 | 2020-02-02 | Kerala | 0 | 0 | 3 | 3 | 0.000000 |
| 4 | 2020-02-03 | Kerala | 0 | 0 | 3 | 3 | 0.000000 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 16845 | 2021-07-07 | Telangana | 613124 | 3703 | 628282 | 11455 | 97.587389 |
| 16846 | 2021-07-07 | Tripura | 63964 | 701 | 68612 | 3947 | 93.225675 |
| 16847 | 2021-07-07 | Uttarakhand | 332006 | 7338 | 340882 | 1538 | 97.396166 |
| 16848 | 2021-07-07 | Uttar Pradesh | 1682130 | 22656 | 1706818 | 2032 | 98.553566 |
| 16849 | 2021-07-07 | West Bengal | 1472132 | 17834 | 1507241 | 17275 | 97.670645 |

16787 rows × 7 columns

In [46]:

df.isnull().sum()

Out[46]:

| | |
|----------------------|----|
| Date | 0 |
| State/UnionTerritory | 0 |
| Cured | 0 |
| Deaths | 0 |
| Confirmed | 0 |
| Active_case | 0 |
| Recovery_rate | 46 |
| dtype: int64 | |

In [47]:

```
df.groupby(['Date'])[['Confirmed', 'Cured', 'Deaths', 'State/UnionTerritory', 'Active_case', 'Re
```

Out[47]:

| Date | Confirmed | Cured | Deaths | State/UnionTerritory | Active_case | Recovery_rate |
|------------|-----------|---------|--------|----------------------|-------------|---------------|
| 2020-01-30 | 1 | 0 | 0 | Kerala | 1 | 0.000000 |
| 2020-01-31 | 1 | 0 | 0 | Kerala | 1 | 0.000000 |
| 2020-02-01 | 2 | 0 | 0 | Kerala | 2 | 0.000000 |
| 2020-02-02 | 3 | 0 | 0 | Kerala | 3 | 0.000000 |
| 2020-02-03 | 3 | 0 | 0 | Kerala | 3 | 0.000000 |
| ... | ... | ... | ... | ... | ... | ... |
| 2021-07-03 | 6079352 | 5836920 | 122353 | West Bengal | 120079 | 99.555135 |
| 2021-07-04 | 6088841 | 5845315 | 122724 | West Bengal | 120802 | 99.574146 |
| 2021-07-05 | 6098177 | 5848693 | 123030 | West Bengal | 126454 | 99.602611 |
| 2021-07-06 | 6104917 | 5861720 | 123136 | West Bengal | 120061 | 99.630996 |
| 2021-07-07 | 6113335 | 5872268 | 123531 | West Bengal | 117536 | 99.593381 |

525 rows × 6 columns

In [48]:

```
df.groupby(['State/UnionTerritory'])[['Confirmed', 'Cured', 'Deaths', 'Active_case', 'Recovery_
```

Out[48]:

| State/UnionTerritory | Confirmed | Cured | Deaths | Active_case | Recovery_rate |
|---|-----------|-------|--------|-------------|---------------|
| Andaman and Nicobar Islands | 1 | 0 | 0 | 0 | 0.0 |
| Andhra Pradesh | 1 | 0 | 0 | 1 | 0.0 |
| Arunachal Pradesh | 1 | 0 | 0 | 0 | 0.0 |
| Assam | 1 | 0 | 0 | 1 | 0.0 |
| Bihar | 2 | 0 | 1 | -1257 | 0.0 |
| Chandigarh | 1 | 0 | 0 | 1 | 0.0 |
| Chhattisgarh | 1 | 0 | 0 | 1 | 0.0 |
| Dadra and Nagar Haveli and Daman and Diu | 1 | 0 | 0 | 0 | 0.0 |
| Delhi | 1 | 0 | 0 | -9368 | 0.0 |
| Goa | 3 | 0 | 0 | 0 | 0.0 |
| Gujarat | 5 | 0 | 0 | 5 | 0.0 |
| Haryana | 2 | 0 | 0 | 2 | 0.0 |
| Himachal Pradesh | 2 | 0 | 0 | -589 | 0.0 |
| Jammu and Kashmir | 1 | 0 | 0 | 1 | 0.0 |
| Jharkhand | 1 | 0 | 0 | 1 | 0.0 |
| Karnataka | 1 | 0 | 0 | -5533 | 0.0 |
| Kerala | 1 | 0 | 0 | 0 | 0.0 |
| Ladakh | 2 | 0 | 0 | 0 | 0.0 |
| Lakshadweep | 0 | 0 | 0 | 0 | 0.0 |
| Madhya Pradesh | 4 | 0 | 0 | -1876 | 0.0 |
| Maharashtra | 2 | 0 | 0 | 2 | 0.0 |
| Manipur | 1 | 0 | 0 | 0 | 0.0 |
| Meghalaya | 1 | 0 | 0 | 0 | 0.0 |
| Mizoram | 1 | 0 | 0 | 0 | 0.0 |
| Nagaland | 0 | 0 | 0 | 0 | 0.0 |
| Odisha | 1 | 0 | 0 | 1 | 0.0 |
| Puducherry | 1 | 0 | 0 | 1 | 0.0 |
| Punjab | 1 | 0 | 0 | -3299 | 0.0 |
| Rajasthan | 1 | 0 | 0 | 1 | 0.0 |
| Sikkim | 1 | 0 | 0 | 1 | 0.0 |
| Tamil Nadu | 1 | 0 | 0 | 1 | 0.0 |
| Telangana | 1 | 0 | 0 | 1 | 0.0 |
| Tripura | 1 | 0 | 0 | 0 | 0.0 |

| State/UnionTerritory | Confirmed | Cured | Deaths | Active_case | Recovery_rate |
|----------------------|-----------|-------|--------|-------------|---------------|
| Uttar Pradesh | 6 | 0 | 0 | 6 | 0.0 |
| Uttarakhand | 1 | 0 | 0 | 1 | 0.0 |
| West Bengal | 1 | 0 | 0 | 1 | 0.0 |

In [49]:

```
df.groupby(['State/UnionTerritory'])[['Confirmed', 'Cured', 'Deaths', 'Active_case', 'Recovery_
```

Out[49]:

| State/UnionTerritory | Confirmed | Cured | Deaths | Active_case | Recovery_rate |
|---|-----------|---------|--------|-------------|---------------|
| Andaman and Nicobar Islands | 7487 | 7343 | 128 | 1154 | 100.000000 |
| Andhra Pradesh | 1908065 | 1861937 | 12898 | 211554 | 99.129422 |
| Arunachal Pradesh | 37879 | 34525 | 181 | 3918 | 100.000000 |
| Assam | 522267 | 493306 | 4717 | 56295 | 99.253852 |
| Bihar | 722746 | 711913 | 9612 | 115152 | 99.888988 |
| Chandigarh | 61752 | 60837 | 809 | 8653 | 98.518267 |
| Chhattisgarh | 996359 | 977893 | 13462 | 131245 | 98.146652 |
| Dadra and Nagar Haveli and Daman and Diu | 10575 | 10532 | 4 | 2081 | 99.941142 |
| Delhi | 1434687 | 1408853 | 25001 | 103424 | 99.754600 |
| Goa | 167823 | 162787 | 3079 | 32953 | 100.000000 |
| Gujarat | 823964 | 811699 | 10072 | 148297 | 98.511464 |
| Haryana | 769030 | 758442 | 9506 | 116867 | 98.623201 |
| Himachal Pradesh | 202945 | 198134 | 3485 | 40008 | 99.310430 |
| Jammu and Kashmir | 317481 | 309554 | 4345 | 52848 | 97.970514 |
| Jharkhand | 346038 | 340365 | 5118 | 61195 | 98.735071 |
| Karnataka | 2859595 | 2784030 | 35526 | 605515 | 99.275356 |
| Kerala | 2996094 | 2877557 | 13960 | 445692 | 100.000000 |
| Ladakh | 20137 | 19733 | 204 | 2041 | 100.000000 |
| Lakshadweep | 9947 | 9643 | 49 | 2320 | 96.943802 |
| Madhya Pradesh | 790042 | 780578 | 9017 | 111366 | 99.229073 |
| Maharashtra | 6113335 | 5872268 | 123531 | 701614 | 96.056702 |
| Manipur | 73581 | 66132 | 1218 | 9613 | 100.000000 |
| Meghalaya | 52358 | 47173 | 880 | 8255 | 98.854935 |
| Mizoram | 22155 | 18383 | 98 | 4471 | 100.000000 |
| Nagaland | 25619 | 23982 | 503 | 5049 | 99.239326 |
| Odisha | 927186 | 897362 | 4299 | 106493 | 99.257203 |
| Puducherry | 118227 | 114673 | 1763 | 18277 | 98.308500 |
| Punjab | 596736 | 578590 | 16131 | 79963 | 98.644027 |
| Rajasthan | 952836 | 942882 | 8942 | 212753 | 98.955329 |
| Sikkim | 21403 | 19200 | 309 | 4306 | 97.333333 |
| Tamil Nadu | 2503481 | 2435872 | 33132 | 313048 | 98.072796 |
| Telangana | 628282 | 613124 | 3703 | 80695 | 98.902172 |
| Tripura | 68612 | 63964 | 701 | 8302 | 100.000000 |

| State/UnionTerritory | Confirmed | Cured | Deaths | Active_case | Recovery_rate |
|----------------------|-----------|---------|--------|-------------|---------------|
| Uttar Pradesh | 1706818 | 1682130 | 22656 | 310783 | 98.553566 |
| Uttarakhand | 340882 | 332006 | 7338 | 80000 | 97.792747 |
| West Bengal | 1507241 | 1472132 | 17834 | 132181 | 97.834341 |

In [53]:

```
df2=df.groupby(['State/UnionTerritory'])['Deaths'].sum()
```

In [55]:

```
df2.sort_values(ascending=False).head(5) # top 5 heighest deth cases in india
```

Out[55]:

```
State/UnionTerritory
Maharashtra      19314532
Karnataka        4819018
Tamil Nadu       4731627
Delhi             4066907
Uttar Pradesh     3347656
Name: Deaths, dtype: int64
```

In [56]:

```
df2.sort_values(ascending=True).head(5) # top 5 Lowest death cases in india
```

Out[56]:

```
State/UnionTerritory
Dadra and Nagar Haveli and Daman and Diu      882
Lakshadweep                               2178
Mizoram                                    5073
Arunachal Pradesh                         19303
Andaman and Nicobar Islands                22624
Name: Deaths, dtype: int64
```

In [57]:

```
df3=df.groupby(['State/UnionTerritory'])['Cured'].sum()
```

In [58]:

```
df3.sort_values(ascending=False).head(5) # top 5 highest cured cases in india
```

Out[58]:

| State/UnionTerritory | |
|----------------------|-----------|
| Maharashtra | 813788907 |
| Karnataka | 345648926 |
| Tamil Nadu | 317067499 |
| Kerala | 311127643 |
| Andhra Pradesh | 303427899 |

Name: Cured, dtype: int64

In [59]:

```
df3.sort_values(ascending=True).head(5) # top 5 Lowest cured cases in india
```

Out[59]:

| State/UnionTerritory | |
|--|---------|
| Lakshadweep | 471712 |
| Dadra and Nagar Haveli and Daman and Diu | 1491338 |
| Mizoram | 1534630 |
| Andaman and Nicobar Islands | 1589935 |
| Sikkim | 1983899 |

Name: Cured, dtype: int64

In [60]:

```
df4=df.groupby(['State/UnionTerritory'])['Confirmed'].sum()
```

In [61]:

```
df4.sort_values(ascending=False).head(5) # top 5 highest confirmed cases in india
```

Out[61]:

| State/UnionTerritory | |
|----------------------|-----------|
| Maharashtra | 908892470 |
| Karnataka | 387597335 |
| Kerala | 344319045 |
| Tamil Nadu | 342829697 |
| Andhra Pradesh | 324146783 |

Name: Confirmed, dtype: int64

In [62]:

```
df4.sort_values(ascending=True).head(5) # top 5 Lowest confirmed cases in india
```

Out[62]:

| State/UnionTerritory | |
|--|---------|
| Lakshadweep | 561459 |
| Dadra and Nagar Haveli and Daman and Diu | 1587570 |
| Andaman and Nicobar Islands | 1675248 |
| Mizoram | 1822190 |
| Sikkim | 2315519 |

Name: Confirmed, dtype: int64

In [63]:

```
df5=df.groupby(['State/UnionTerritory'])['Active_case'].sum()
```

In [64]:

```
df5.sort_values(ascending=False).head(5) # top 5 heighest active cases in india
```

Out[64]:

```
State/UnionTerritory
Maharashtra      75789031
Karnataka        37129391
Kerala           31863648
Tamil Nadu       21030571
Andhra Pradesh   18243068
Name: Active_case, dtype: int64
```

In [65]:

```
df5.sort_values(ascending=True).head(5) # top 5 Lowest active cases in india
```

Out[65]:

```
State/UnionTerritory
Andaman and Nicobar Islands          62689
Lakshadweep                          87569
Dadra and Nagar Haveli and Daman and Diu 95350
Ladakh                                246508
Mizoram                               282487
Name: Active_case, dtype: int64
```

In [66]:

```
df7=df.groupby(['State/UnionTerritory'])['Recovery_rate'].sum()%100
```

In [67]:

```
df7.sort_values(ascending=False).head(5) # top 5 heighest recovery rate in india
```

Out[67]:

```
State/UnionTerritory
Karnataka     97.723075
Ladakh        94.709098
Meghalaya    92.429170
Gujarat       90.851964
Bihar          78.230679
Name: Recovery_rate, dtype: float64
```

In [68]:

```
df7.sort_values(ascending=True).head(5) # top 5 Lowest rate in india
```

Out[68]:

```
State/UnionTerritory
Punjab           1.222771
Sikkim            4.681875
Nagaland          12.298198
Arunachal Pradesh 13.762354
Uttarakhand       15.673192
Name: Recovery_rate, dtype: float64
```

In [69]:

```
px.line(df, x="Date", y="Confirmed", title='Date confirmed Cases in India')
```

Date confirmed Cases in India



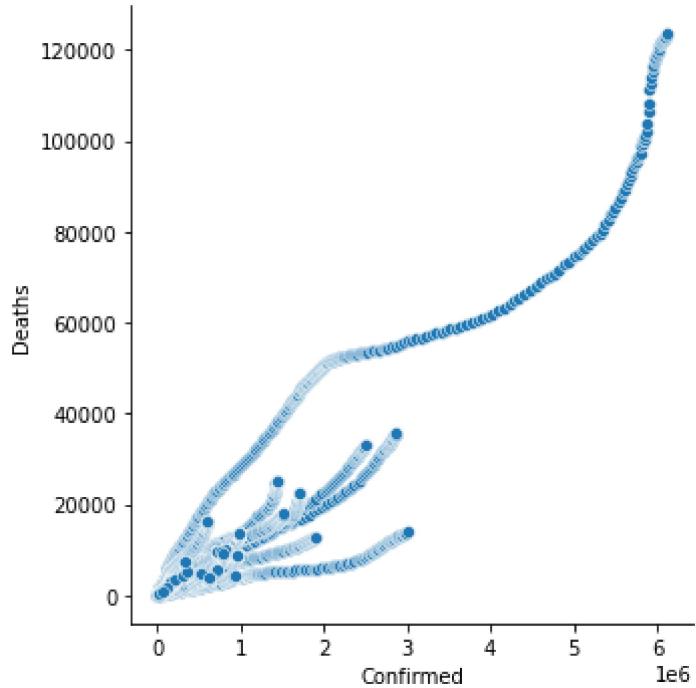
In [70]:

```
fig=plt.figure(figsize=(10,10))
sns.relplot(x='Confirmed',y='Deaths',data=df)
plt.show
```

Out[70]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

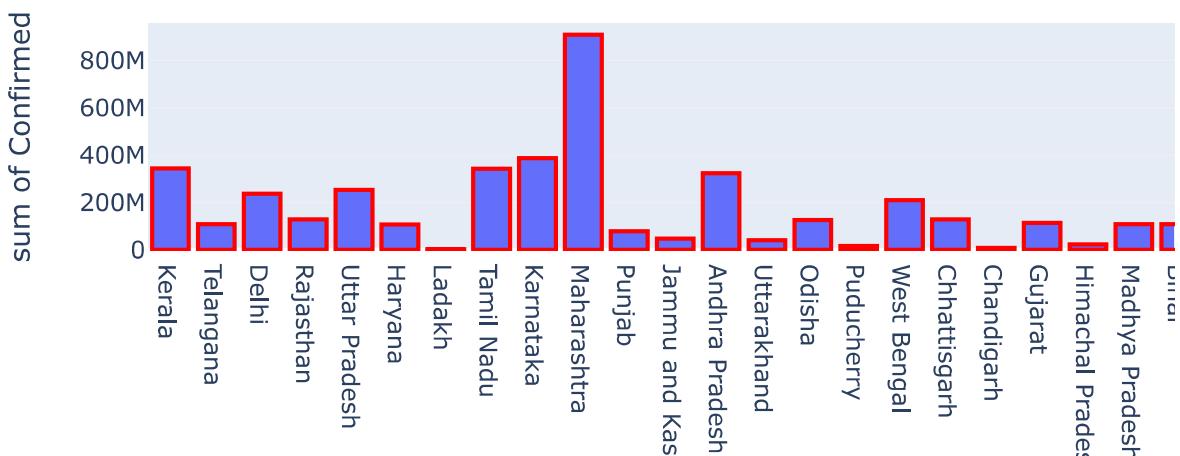
```
<Figure size 720x720 with 0 Axes>
```



In [71]:

```
fig = px.histogram(x='State/UnionTerritory',y='Confirmed',title='Statewise Confirmed cases  
fig.update_traces(marker_line_width=2,marker_line_color="red")
```

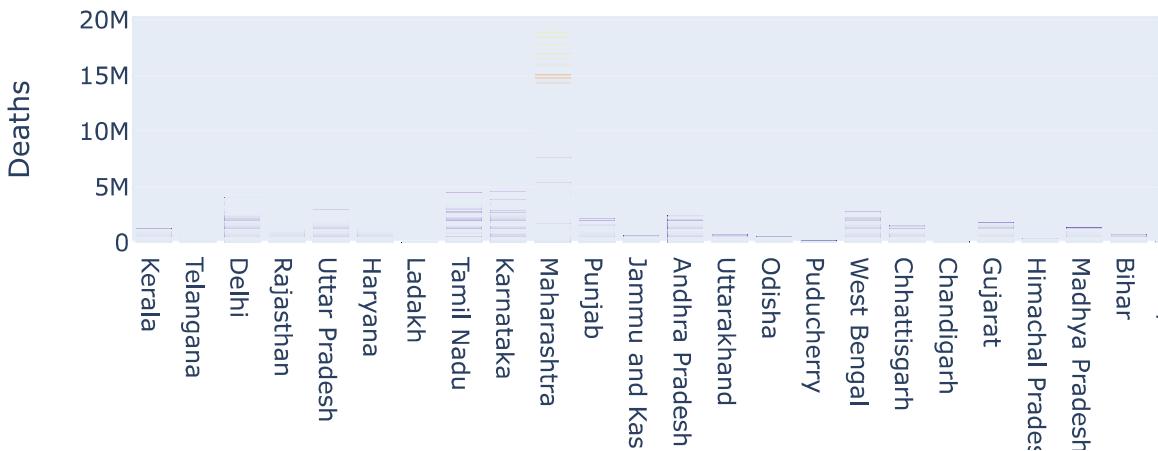
Statewise Confirmed cases in india



In [72]:

```
px.bar(x='State/UnionTerritory',y='Deaths',data_frame=df,color='Deaths',title="Statewise de
```

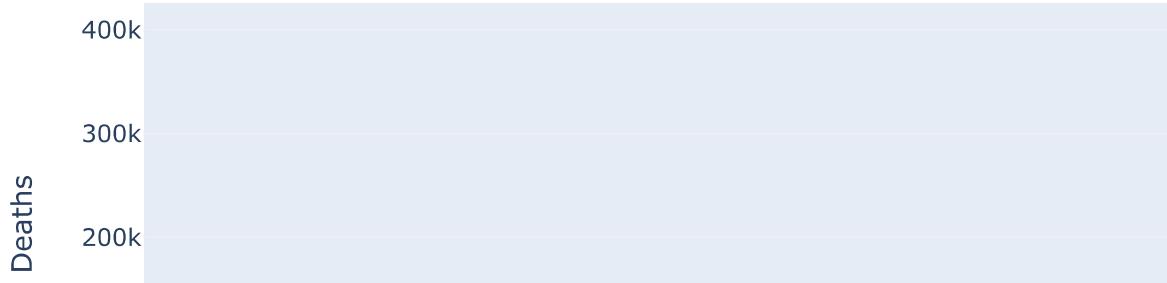
Statewise death Cases in india



In [73]:

```
px.bar(df, x="Date", y="Deaths", color="Deaths", title=' Datewise death cases in india' ,hei
```

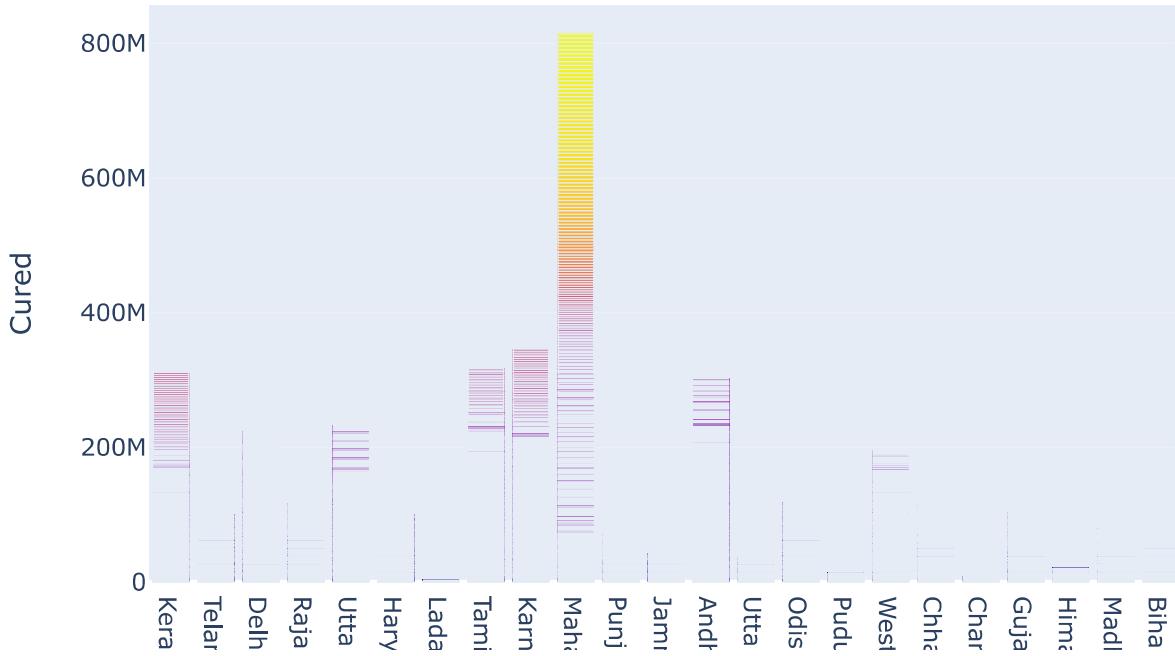
Datewise death cases in india



In [74]:

```
px.bar(df, x='State/UnionTerritory', y="Cured", color="Cured",title='Statwise cured cases i
```

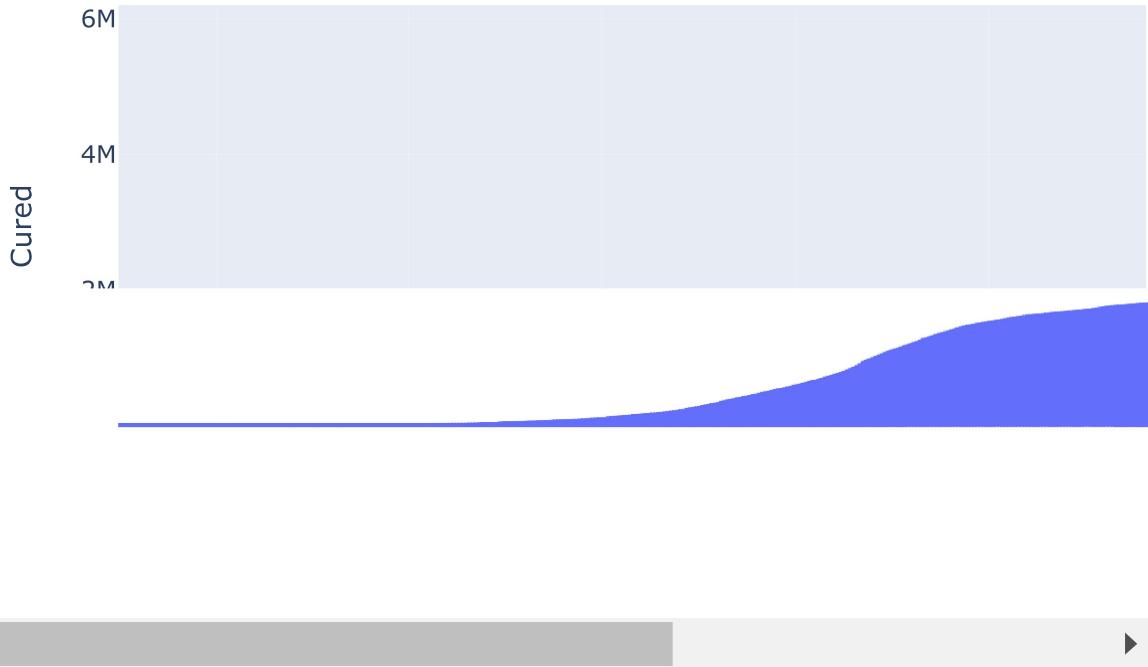
Statwise cured cases in india



In [76]:

```
px.line(df,x="Date", y="Cured",title='Monthwise cured case in india',height=400)
```

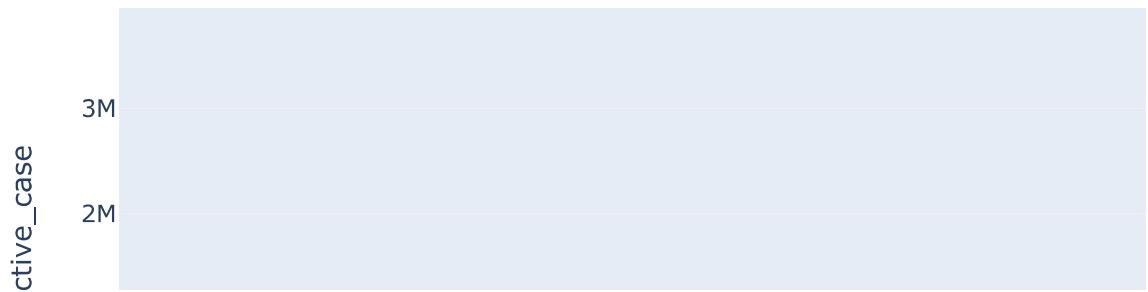
Monthwise cured case in india



In [77]:

```
px.bar(df,x="Date", y="Active_case",title='Monthwise active cases in india',height=400)
```

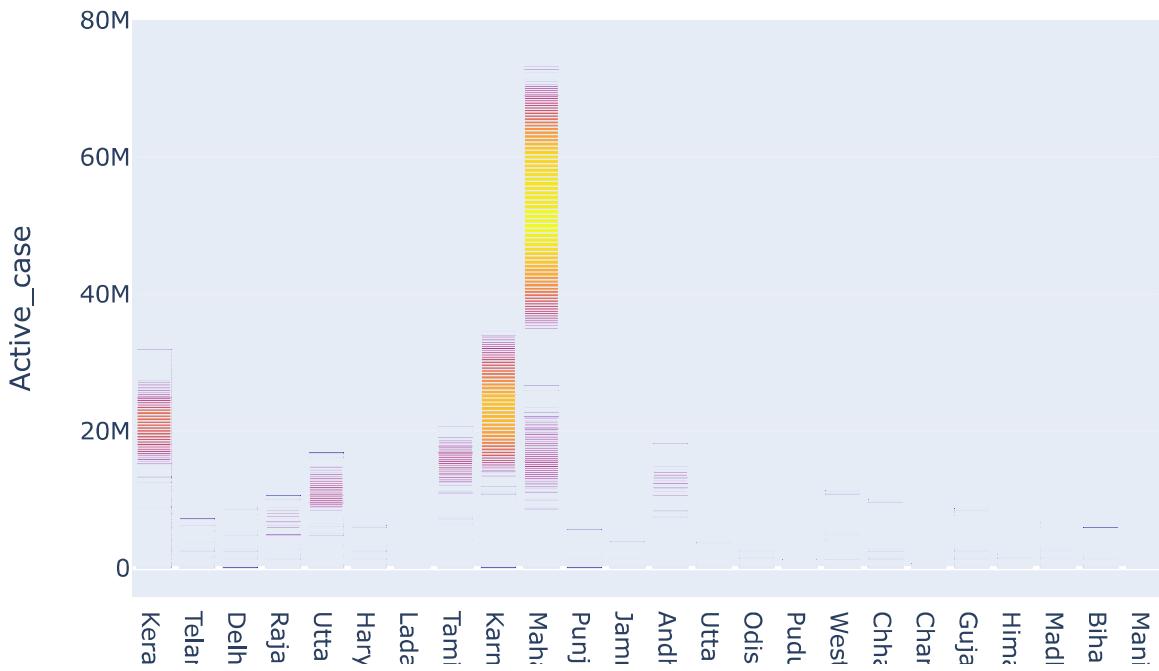
Monthwise active cases in india



In [156]:

```
px.bar(df, x='State/UnionTerritory', y="Active_case", color="Active_case",title='Statewise
```

Statewise cured cases in india



In []:

In []:

In []:

In [82]:

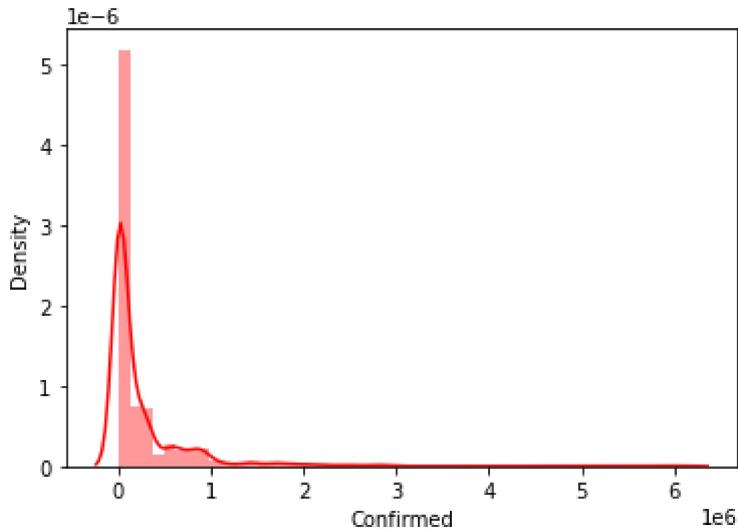
```
sns.distplot(df['Confirmed'], kde=True, color='red')
```

C:\Users\NIKUNJ\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

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

Out[82]:

```
<AxesSubplot:xlabel='Confirmed', ylabel='Density'>
```

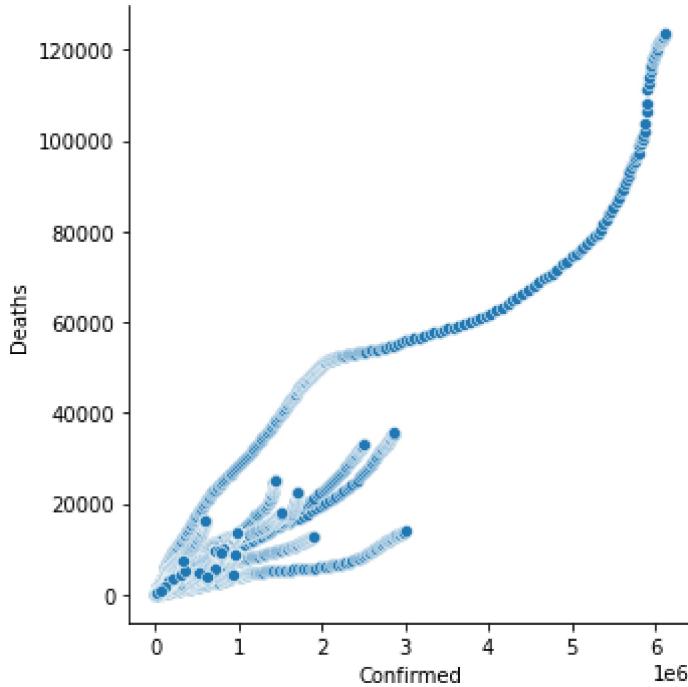


In [83]:

```
sns.relplot(x='Confirmed',y='Deaths',data=df)
```

Out[83]:

```
<seaborn.axisgrid.FacetGrid at 0x11bdbf68e20>
```

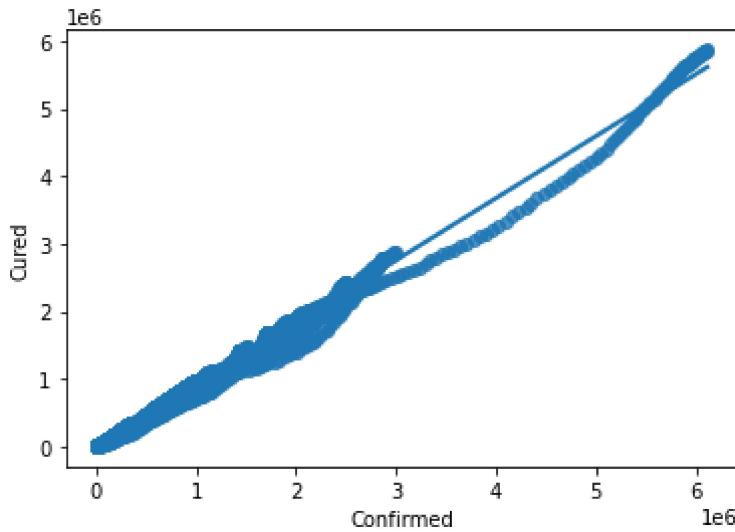


In [84]:

```
sns.regplot(x='Confirmed',y='Cured',data=df)
```

Out[84]:

```
<AxesSubplot:xlabel='Confirmed', ylabel='Cured'>
```



In [85]:

```
df['Date']= pd.to_datetime(df['Date'], format = "%Y-%m-%d")
df['Year']= pd.DatetimeIndex(df['Date']).year
df['Months']= pd.DatetimeIndex(df['Date']).month
df['Day']= pd.DatetimeIndex(df['Date']).day
```

In [86]:

```
df['Date'].describe()
```

C:\Users\NIKUNJ\AppData\Local\Temp\ipykernel_8292/3134759576.py:1: FutureWarning:

Treating datetime data as categorical rather than numeric in `describe` is deprecated and will be removed in a future version of pandas. Specify `datetime_is_numeric=True` to silence this warning and adopt the future behavior now.

Out[86]:

```
count          16787
unique         525
top    2021-07-07 00:00:00
freq            36
first   2020-01-30 00:00:00
last    2021-07-07 00:00:00
Name: Date, dtype: object
```

In [87]:

```
df['Year'].describe()
```

Out[87]:

```
count    16787.000000
mean      2020.403169
std        0.490549
min      2020.000000
25%      2020.000000
50%      2020.000000
75%      2021.000000
max      2021.000000
Name: Year, dtype: float64
```

In [88]:

```
df['Year'].nunique()
```

Out[88]:

2

In [89]:

```
df['Year'].unique()
```

Out[89]:

```
array([2020, 2021], dtype=int64)
```

In [90]:

```
df.groupby('Year')[['Confirmed', 'Cured', 'Deaths', 'Active_case']].sum()
```

Out[90]:

| | Confirmed | Cured | Deaths | Active_case |
|-------------|------------|------------|----------|-------------|
| Year | | | | |
| 2020 | 1073677055 | 941314195 | 17022508 | 115340352 |
| 2021 | 3279455293 | 3035879941 | 41703492 | 201871860 |

In [91]:

```
df_month=df.groupby('Months')[['Confirmed', 'Cured', 'Deaths', 'Active_case']].sum()
```

In [92]:

```
df_month
```

Out[92]:

| | Confirmed | Cured | Deaths | Active_case |
|-----------|-----------|-----------|----------|-------------|
| Months | | | | |
| 1 | 326469749 | 315332019 | 4709167 | 6428563 |
| 2 | 305631889 | 297133802 | 4359434 | 4138653 |
| 3 | 356315219 | 342611205 | 4935455 | 8768559 |
| 4 | 441083036 | 385065633 | 5353568 | 50663835 |
| 5 | 754828200 | 646240106 | 8480751 | 100107343 |
| 6 | 894991190 | 848822379 | 11475067 | 34693744 |
| 7 | 245445610 | 227533465 | 3606557 | 14305588 |
| 8 | 80749620 | 58580895 | 1553468 | 20615257 |
| 9 | 149113758 | 118592934 | 2443374 | 28077450 |
| 10 | 226770312 | 198824412 | 3457615 | 24488285 |
| 11 | 264556412 | 246213201 | 3894165 | 14449046 |
| 12 | 307177353 | 292244085 | 4457379 | 10475889 |

In [93]:

```
df.groupby(['Year', 'State/UnionTerritory'])[['Confirmed', 'Cured', 'Deaths', 'Active_case']].s
```

Out[93]:

| | | Confirmed | Cured | Deaths | Active_case |
|------|------------------------------------|-----------|-----------|---------|-------------|
| Year | State/UnionTerritory | | | | |
| 2020 | Andaman and Nicobar Islands | 590838 | 534731 | 7772 | 48335 |
| | Andhra Pradesh | 104134066 | 95094768 | 871178 | 8168120 |
| | Arunachal Pradesh | 1679508 | 1442769 | 4488 | 232251 |
| | Assam | 26440101 | 23558817 | 108050 | 2773234 |
| | Bihar | 29298924 | 26953520 | 151678 | 2193726 |
| ... | ... | ... | ... | ... | ... |
| 2021 | Telangana | 75887876 | 71676818 | 420988 | 3790070 |
| | Tripura | 7583078 | 7158306 | 83818 | 340954 |
| | Uttar Pradesh | 189954972 | 177050153 | 2411300 | 10493519 |
| | Uttarakhand | 33219139 | 29819818 | 606811 | 2792510 |
| | West Bengal | 159727639 | 150788352 | 2273362 | 6665925 |

72 rows × 4 columns

In [94]:

```
df.groupby(['Months','State/UnionTerritory'])[['Confirmed','Cured','Deaths','Active_case']]
```

Out[94]:

| Months | State/UnionTerritory | Confirmed | Cured | Deaths | Active_case |
|--------|------------------------------------|-----------|----------|--------|-------------|
| 1 | Andaman and Nicobar Islands | 154187 | 151473 | 1922 | 792 |
| | Andhra Pradesh | 27448884 | 27160550 | 221186 | 67148 |
| | Arunachal Pradesh | 520415 | 516950 | 1736 | 1729 |
| | Assam | 6718515 | 6600118 | 33089 | 85308 |
| | Bihar | 7952001 | 7800177 | 44922 | 106902 |
| ... | ... | ... | ... | ... | ... |
| 12 | Telangana | 8646516 | 8372522 | 46521 | 227473 |
| | Tripura | 1024869 | 1003331 | 11710 | 9828 |
| | Uttar Pradesh | 17564562 | 16732432 | 250784 | 581346 |
| | Uttarakhand | 2579458 | 2364361 | 42515 | 172582 |
| | West Bengal | 16211044 | 15314155 | 282550 | 614339 |

428 rows × 4 columns

Maharashtra covid data

In []:

In [95]:

```
dfm=df[df['State/UnionTerritory']=='Maharashtra']
```

In [96]:

dfm

Out[96]:

| | Date | State/Union Territory | Cured | Deaths | Confirmed | Active_case | Recovery_rate | Ye |
|-------|------------|-----------------------|---------|--------|-----------|-------------|---------------|-----|
| 76 | 2020-03-09 | Maharashtra | 0 | 0 | 2 | 2 | 0.000000 | 20 |
| 91 | 2020-03-10 | Maharashtra | 0 | 0 | 5 | 5 | 0.000000 | 20 |
| 97 | 2020-03-11 | Maharashtra | 0 | 0 | 2 | 2 | 0.000000 | 20 |
| 120 | 2020-03-12 | Maharashtra | 0 | 0 | 11 | 11 | 0.000000 | 20 |
| 133 | 2020-03-13 | Maharashtra | 0 | 0 | 14 | 14 | 0.000000 | 20 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 16690 | 2021-07-03 | Maharashtra | 5836920 | 122353 | 6079352 | 120079 | 96.012207 | 20 |
| 16726 | 2021-07-04 | Maharashtra | 5845315 | 122724 | 6088841 | 120802 | 96.000454 | 20 |
| 16762 | 2021-07-05 | Maharashtra | 5848693 | 123030 | 6098177 | 126454 | 95.908876 | 20 |
| 16798 | 2021-07-06 | Maharashtra | 5861720 | 123136 | 6104917 | 120061 | 96.016375 | 20 |
| 16834 | 2021-07-07 | Maharashtra | 5872268 | 123531 | 6113335 | 117536 | 96.056702 | 20 |

486 rows × 10 columns



In [97]:

dfm.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 486 entries, 76 to 16834
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Date              486 non-null    datetime64[ns]
 1   State/UnionTerritory 486 non-null    object  
 2   Cured             486 non-null    int64   
 3   Deaths            486 non-null    int64   
 4   Confirmed         486 non-null    int64   
 5   Active_case       486 non-null    int64   
 6   Recovery_rate     486 non-null    float64 
 7   Year              486 non-null    int64   
 8   Months            486 non-null    int64   
 9   Day               486 non-null    int64  
dtypes: datetime64[ns](1), float64(1), int64(7), object(1)
memory usage: 41.8+ KB
```

In [98]:

df.describe()

Out[98]:

| | Cured | Deaths | Confirmed | Active_case | Recovery_rate | Year |
|-------|--------------|---------------|--------------|---------------|---------------|--------------|
| count | 1.678700e+04 | 16787.000000 | 1.678700e+04 | 16787.000000 | 16741.000000 | 16787.000000 |
| mean | 2.369211e+05 | 3498.302258 | 2.593157e+05 | 18896.301424 | 75.913310 | 2020.403169 |
| std | 5.233229e+05 | 9345.586841 | 5.681323e+05 | 54362.930997 | 27.438427 | 0.490549 |
| min | 0.000000e+00 | 0.000000 | 0.000000e+00 | -9368.000000 | 0.000000 | 2020.000000 |
| 25% | 2.802500e+03 | 24.000000 | 3.650500e+03 | 307.000000 | 66.186923 | 2020.000000 |
| 50% | 2.900400e+04 | 463.000000 | 3.344100e+04 | 2371.000000 | 87.445291 | 2020.000000 |
| 75% | 2.547405e+05 | 3083.000000 | 2.674615e+05 | 12929.000000 | 96.617651 | 2021.000000 |
| max | 5.872268e+06 | 123531.000000 | 6.113335e+06 | 701614.000000 | 100.000000 | 2021.000000 |



In [99]:

```
dfm.count()
```

Out[99]:

| | |
|----------------------|-----|
| Date | 486 |
| State/UnionTerritory | 486 |
| Cured | 486 |
| Deaths | 486 |
| Confirmed | 486 |
| Active_case | 486 |
| Recovery_rate | 486 |
| Year | 486 |
| Months | 486 |
| Day | 486 |
| dtype: int64 | |

In [101]:

```
dfm.groupby(['Date'])[['Confirmed', 'Cured', 'Deaths', 'Active_case', 'Recovery_rate']].max()
```

Out[101]:

| | Confirmed | Cured | Deaths | Active_case | Recovery_rate |
|------------|-----------|---------|--------|-------------|---------------|
| Date | | | | | |
| 2020-03-09 | 2 | 0 | 0 | 2 | 0.000000 |
| 2020-03-10 | 5 | 0 | 0 | 5 | 0.000000 |
| 2020-03-11 | 2 | 0 | 0 | 2 | 0.000000 |
| 2020-03-12 | 11 | 0 | 0 | 11 | 0.000000 |
| 2020-03-13 | 14 | 0 | 0 | 14 | 0.000000 |
| ... | ... | ... | ... | ... | ... |
| 2021-07-03 | 6079352 | 5836920 | 122353 | 120079 | 96.012207 |
| 2021-07-04 | 6088841 | 5845315 | 122724 | 120802 | 96.000454 |
| 2021-07-05 | 6098177 | 5848693 | 123030 | 126454 | 95.908876 |
| 2021-07-06 | 6104917 | 5861720 | 123136 | 120061 | 96.016375 |
| 2021-07-07 | 6113335 | 5872268 | 123531 | 117536 | 96.056702 |

486 rows × 5 columns

In [102]:

```
dfm1=dfm.groupby(['Date'])['Deaths'].sum()
```

In [104]:

```
dfm1.sort_values(ascending=False).head(5) # top 5 heighest Date's of death cases in maharas
```

Out[104]:

```
Date
2021-07-07    123531
2021-07-06    123136
2021-07-05    123030
2021-07-04    122724
2021-07-03    122353
Name: Deaths, dtype: int64
```

In [105]:

```
dfm1.sort_values(ascending=True).head(5) # Lowest death cases in maharashtra
```

Out[105]:

```
Date
2020-03-09    0
2020-03-10    0
2020-03-11    0
2020-03-12    0
2020-03-13    0
Name: Deaths, dtype: int64
```

In [106]:

```
dfm3=dfm.groupby(['Date'])['Confirmed'].sum()
```

In [108]:

```
dfm3.sort_values(ascending=False).head(5) # top 5 heighest Date's of confirmed cases in maha
```

Out[108]:

```
Date
2021-07-07    6113335
2021-07-06    6104917
2021-07-05    6098177
2021-07-04    6088841
2021-07-03    6079352
Name: Confirmed, dtype: int64
```

In [109]:

```
dfm3.sort_values(ascending=True).head(5) #Lowest confirmed cases in maharashtra
```

Out[109]:

```
Date
2020-03-09      2
2020-03-11      2
2020-03-10      5
2020-03-12     11
2020-03-13     14
Name: Confirmed, dtype: int64
```

In [110]:

```
dfm4=dfm.groupby(['Date'])['Cured'].sum()
```

In [111]:

```
dfm4.sort_values(ascending=False).head(5) # top 5 heighest Date's of Cured cases in maharashtra
```

Out[111]:

```
Date
2021-07-07    5872268
2021-07-06    5861720
2021-07-05    5848693
2021-07-04    5845315
2021-07-03    5836920
Name: Cured, dtype: int64
```

In [112]:

```
dfm4.sort_values(ascending=True).head(5) #Lowest confirmed cases in maharashtra
```

Out[112]:

```
Date
2020-03-09      0
2020-03-24      0
2020-03-23      0
2020-03-21      0
2020-03-20      0
Name: Cured, dtype: int64
```

In [113]:

```
dfm5=dfm.groupby(['Date'])['Active_case'].sum()
```

In [114]:

```
dfm5.sort_values(ascending=False).head(5) # top 5 heighest Date's of Active cases in mahara
```

Out[114]:

```
Date
2021-04-23    701614
2021-04-26    700207
2021-04-22    697467
2021-04-25    696298
2021-04-24    693632
Name: Active_case, dtype: int64
```

In [115]:

```
dfm5.sort_values(ascending=True).head(5) #Lowest active cases cases in maharashtra
```

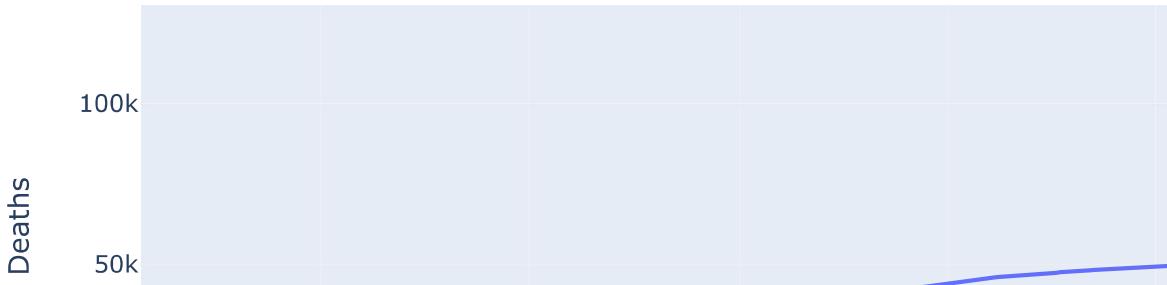
Out[115]:

```
Date
2020-03-09    2
2020-03-11    2
2020-03-10    5
2020-03-12    11
2020-03-13    14
Name: Active_case, dtype: int64
```

In [157]:

```
px.line(dfm,x="Date", y="Deaths",height=400,title='Death cases in Maharashtra')
```

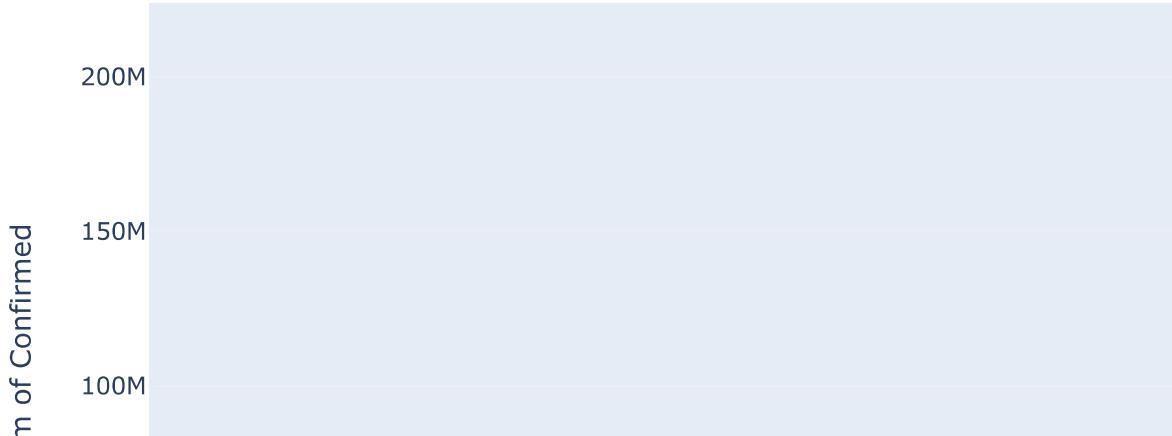
Death cases in Maharashtra



In [159]:

```
fig = px.histogram(x='Date',y='Confirmed',title='Confirmed cases in Maharashtra',data_frame=fig.update_traces(marker_line_width=2,marker_line_color="red")
```

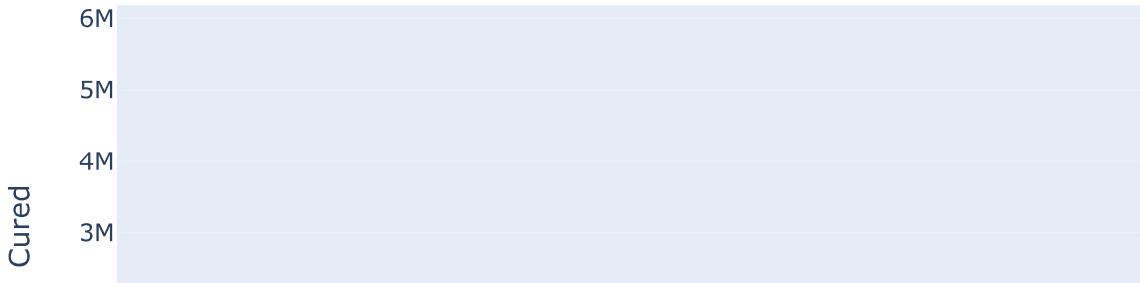
Confirmed cases in Maharashtra



In [121]:

```
px.bar(dfm, x='Date', y="Cured", color="Cured",title='cured cases in Maharashtra',height=40)
```

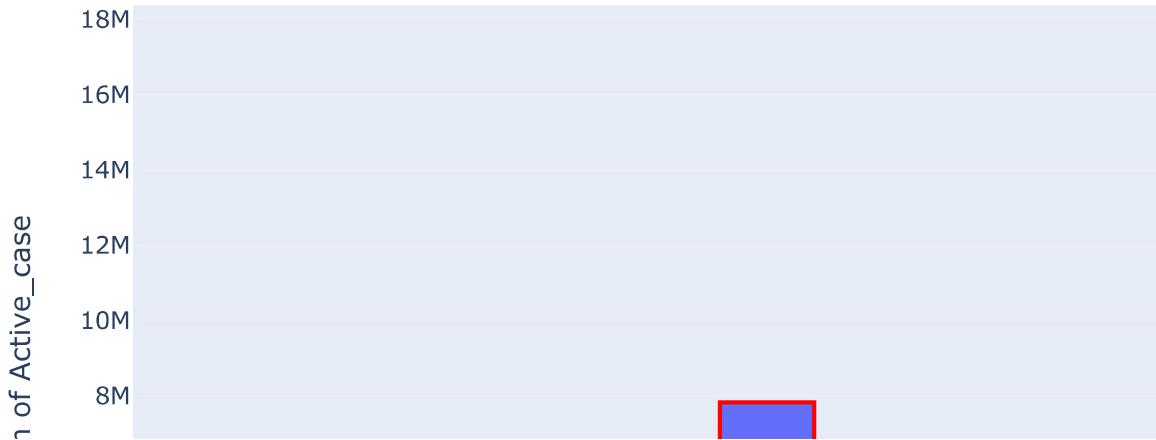
cured cases in Maharashtra



In [128]:

```
fig = px.histogram(x='Date',y='Active_case',title='Active cases in Maharashtra',data_frame=fig.update_traces(marker_line_width=2,marker_line_color="red")
```

Active cases in Maharashtra



In [123]:

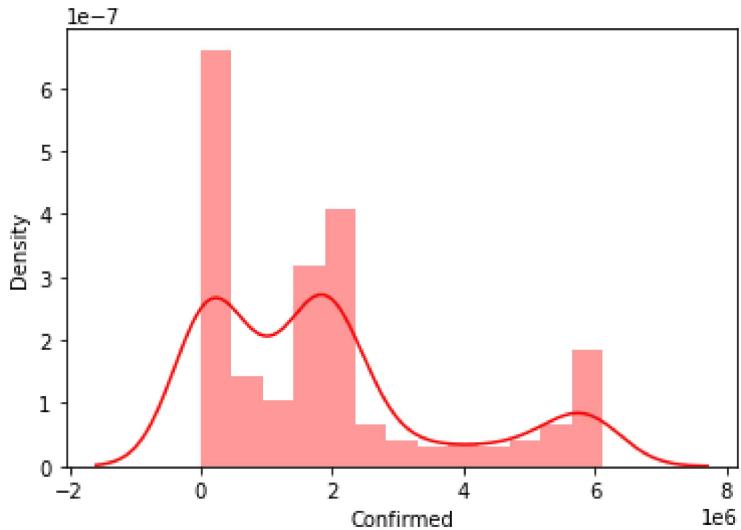
```
sns.distplot(dfm['Confirmed'], kde=True, color='red')
```

C:\Users\NIKUNJ\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

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

Out[123]:

```
<AxesSubplot:xlabel='Confirmed', ylabel='Density'>
```

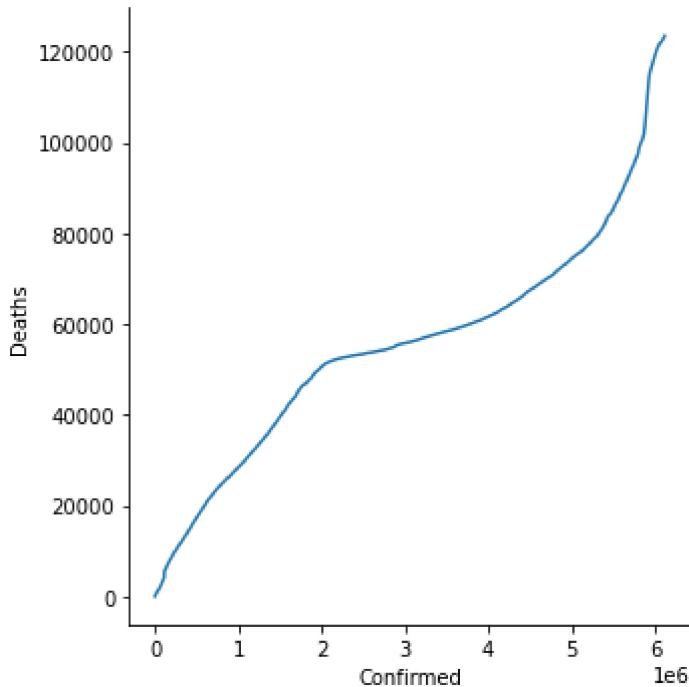


In [124]:

```
sns.relplot(x='Confirmed',y='Deaths',data=dfm,kind='line')
```

Out[124]:

```
<seaborn.axisgrid.FacetGrid at 0x11bde947a60>
```

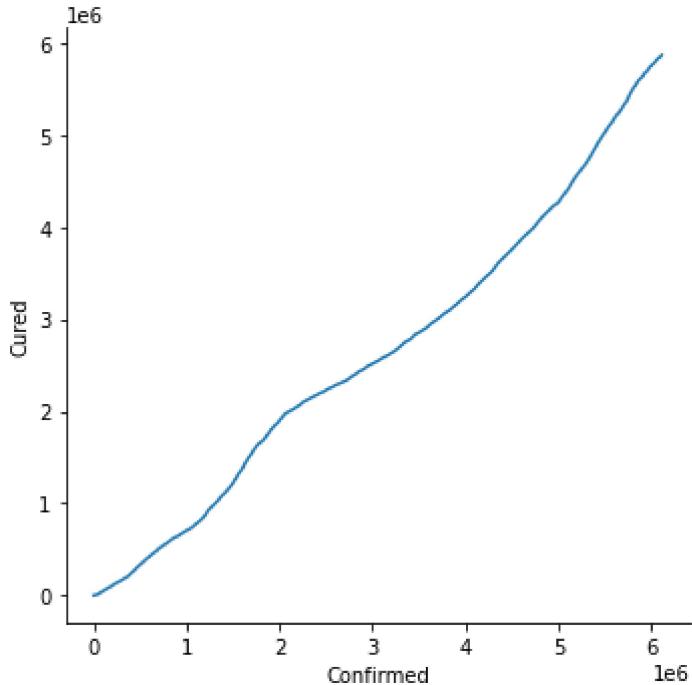


In [125]:

```
sns.relplot(x='Confirmed',y='Cured',data=dfm,kind='line')
```

Out[125]:

```
<seaborn.axisgrid.FacetGrid at 0x11bdea30e80>
```

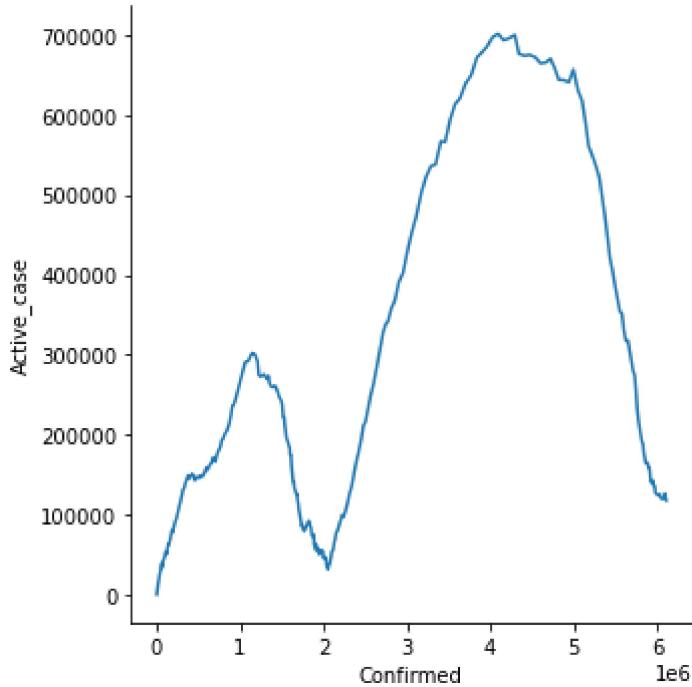


In [130]:

```
sns.relplot(x='Confirmed',y='Active_case',data=dfm,kind='line')
```

Out[130]:

```
<seaborn.axisgrid.FacetGrid at 0x11bde829b50>
```



In [131]:

```
dfm['Year'].describe()
```

Out[131]:

```
count      486.000000
mean      2020.386831
std       0.487526
min      2020.000000
25%      2020.000000
50%      2020.000000
75%      2021.000000
max      2021.000000
Name: Year, dtype: float64
```

In [132]:

```
dfm['Months'].describe()
```

Out[132]:

```
count      486.000000
mean       6.080247
std        3.146548
min       1.000000
25%       4.000000
50%       6.000000
75%       8.750000
max      12.000000
Name: Months, dtype: float64
```

In [134]:

```
dfm['Year'].unique()
```

Out[134]:

```
array([2020, 2021], dtype=int64)
```

In [135]:

```
dfm['Year'].nunique()
```

Out[135]:

2

In [136]:

```
dfm.groupby('Year')[['Confirmed', 'Cured', 'Deaths', 'Active_case']].sum()
```

Out[136]:

| | Confirmed | Cured | Deaths | Active_case |
|-------------|-----------|-----------|----------|-------------|
| Year | | | | |
| 2020 | 222900632 | 187034270 | 6184938 | 29681424 |
| 2021 | 685991838 | 626754637 | 13129594 | 46107607 |

In [137]:

```
df.groupby(['Months', 'State/UnionTerritory'])[['Confirmed', 'Cured', 'Deaths', 'Active_case']]
```

Out[137]:

| | | Confirmed | Cured | Deaths | Active_case |
|---------------|------------------------------------|-----------|----------|--------|-------------|
| Months | State/UnionTerritory | | | | |
| 1 | Andaman and Nicobar Islands | 154187 | 151473 | 1922 | 792 |
| | Andhra Pradesh | 27448884 | 27160550 | 221186 | 67148 |
| | Arunachal Pradesh | 520415 | 516950 | 1736 | 1729 |
| | Assam | 6718515 | 6600118 | 33089 | 85308 |
| | Bihar | 7952001 | 7800177 | 44922 | 106902 |
| ... | ... | ... | ... | ... | ... |
| 12 | Telangana | 8646516 | 8372522 | 46521 | 227473 |
| | Tripura | 1024869 | 1003331 | 11710 | 9828 |
| | Uttar Pradesh | 17564562 | 16732432 | 250784 | 581346 |
| | Uttarakhand | 2579458 | 2364361 | 42515 | 172582 |
| | West Bengal | 16211044 | 15314155 | 282550 | 614339 |

428 rows × 4 columns

In []:

In []:

In []:

