

```
In [1]: 1 import numpy as np  
2 import pandas as pd  
3 import matplotlib.pyplot as plt  
4 import seaborn as sns  
5 import plotly.express as px  
6
```

```
In [2]: 1 data = pd.read_csv(r'C:\Users\Tejas\Downloads\covid_19_india.csv')
```

```
In [3]: 1  
2 data1 = pd.DataFrame(data)
```

```
In [4]: 1 # creating active cases new column  
2 # creating recovery rate new column  
3 data1['Active'] = data1['Confirmed'] - (data1['Cured'] + data1['Deaths'])  
4 data1['Recovery Rate'] = data1['Cured']*100/data1['Confirmed']  
5 data1.replace('Bihar****', 'Bihar', inplace=True)  
6 data1.replace('Telengana', 'Telangana', inplace=True)  
7 data1.replace('-', '0', inplace=True)
```

In [5]: 1 | data1

Out[5]:

	Sno	Date	Time	State/Union Territory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Active	Recovery Rate
0	1	2020-01-30	6:00 PM	Kerala	1		0	0	0	1	1 0.000000
1	2	2020-01-31	6:00 PM	Kerala	1		0	0	0	1	1 0.000000
2	3	2020-02-01	6:00 PM	Kerala	2		0	0	0	2	2 0.000000
3	4	2020-02-02	6:00 PM	Kerala	3		0	0	0	3	3 0.000000
4	5	2020-02-03	6:00 PM	Kerala	3		0	0	0	3	3 0.000000
...	...	...	...	...	...		...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	0		0	613124	3703	628282	11455 97.587389
16846	16847	2021-07-07	8:00 AM	Tripura	0		0	63964	701	68612	3947 93.225675
16847	16848	2021-07-07	8:00 AM	Uttarakhand	0		0	332006	7338	340882	1538 97.396166
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	0		0	1682130	22656	1706818	2032 98.553566
16849	16850	2021-07-07	8:00 AM	West Bengal	0		0	1472132	17834	1507241	17275 97.670645

16850 rows × 11 columns

In [6]:

```
1 #top 10 Date by Deaths State
2 top_10_deaths_state = data1.groupby(['State/UnionTerritory'])['Deaths'].sum()
3 top_10 = top_10_deaths_state.sort_values(ascending=False).head(10)
4 pd.DataFrame(top_10)
```

Out[6]:

Deaths	
State/UnionTerritory	
<b>Maharashtra</b>	19314532
<b>Karnataka</b>	4819018
<b>Tamil Nadu</b>	4731627
<b>Delhi</b>	4066907
<b>Uttar Pradesh</b>	3347656
<b>West Bengal</b>	3214840
<b>Andhra Pradesh</b>	2475816
<b>Punjab</b>	2216735
<b>Gujarat</b>	1866811
<b>Chhattisgarh</b>	1591126

In [7]:

```
1 #top 10 Date by Active State
2 top_10_active_state = data1.groupby(['State/UnionTerritory'])['Active'].sum()
3 top_10 = top_10_active_state.sort_values(ascending=False).head(10)
4 pd.DataFrame(top_10)
```

Out[7]:

Active

State/UnionTerritory	Active
Maharashtra	75789031
Karnataka	37129391
Kerala	31863648
Tamil Nadu	21030571
Andhra Pradesh	18243068
Uttar Pradesh	16966587
West Bengal	11311169
Rajasthan	10525506
Chhattisgarh	9997112
Delhi	8843231

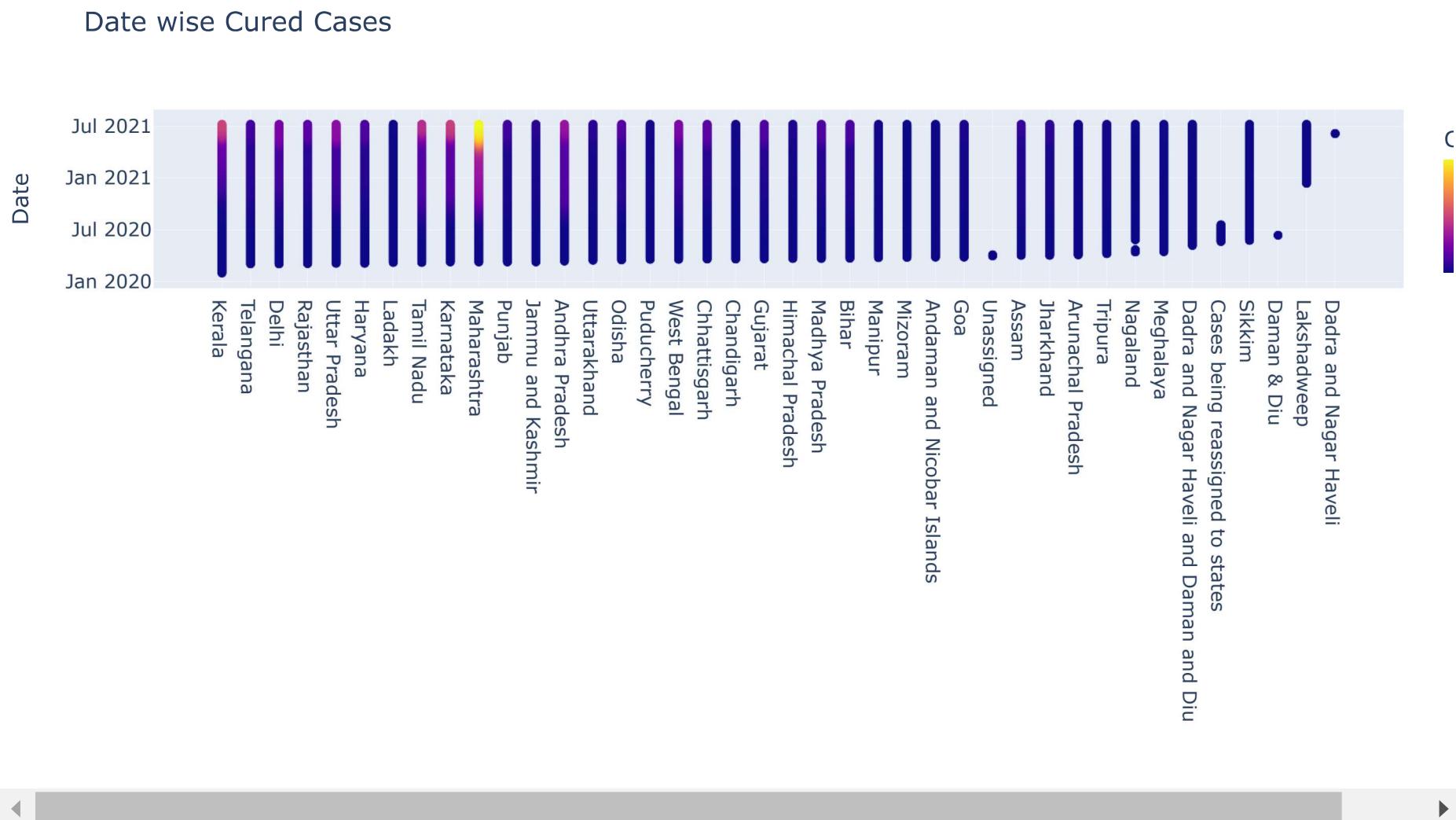
In [8]:

```
1 #top 10 Date by Cured State
2 top_10_cured_state = data1.groupby(['State/UnionTerritory'])['Cured'].sum()
3 top_10 = top_10_cured_state.sort_values(ascending=False).head(10)
4 pd.DataFrame(top_10)
```

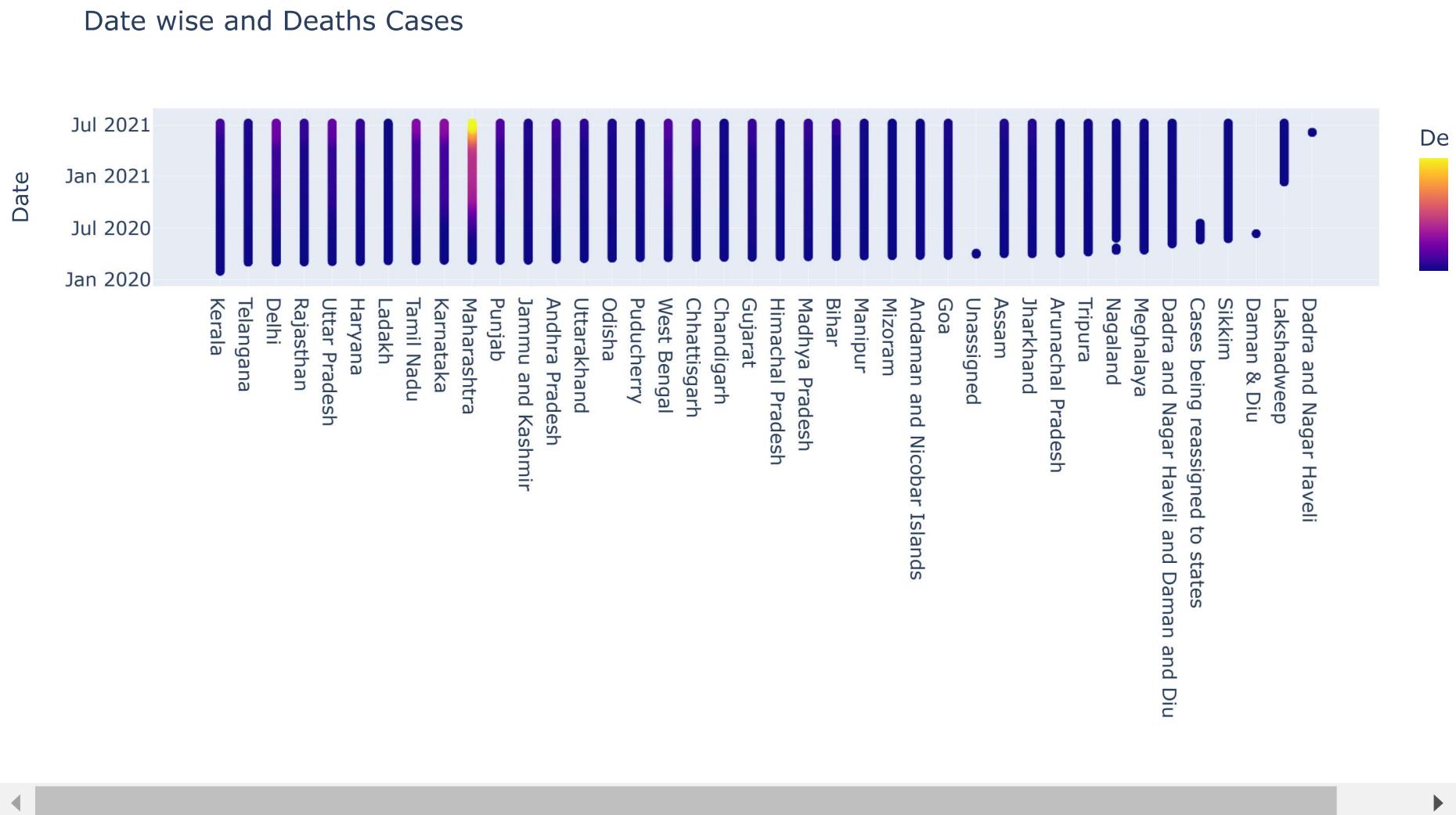
Out[8]:

Cured	
State/UnionTerritory	
<b>Maharashtra</b>	813788907
<b>Karnataka</b>	345648926
<b>Tamil Nadu</b>	317067499
<b>Kerala</b>	311127643
<b>Andhra Pradesh</b>	303427899
<b>Uttar Pradesh</b>	232529439
<b>Delhi</b>	224062704
<b>West Bengal</b>	195296839
<b>Odisha</b>	117984789
<b>Rajasthan</b>	117312772

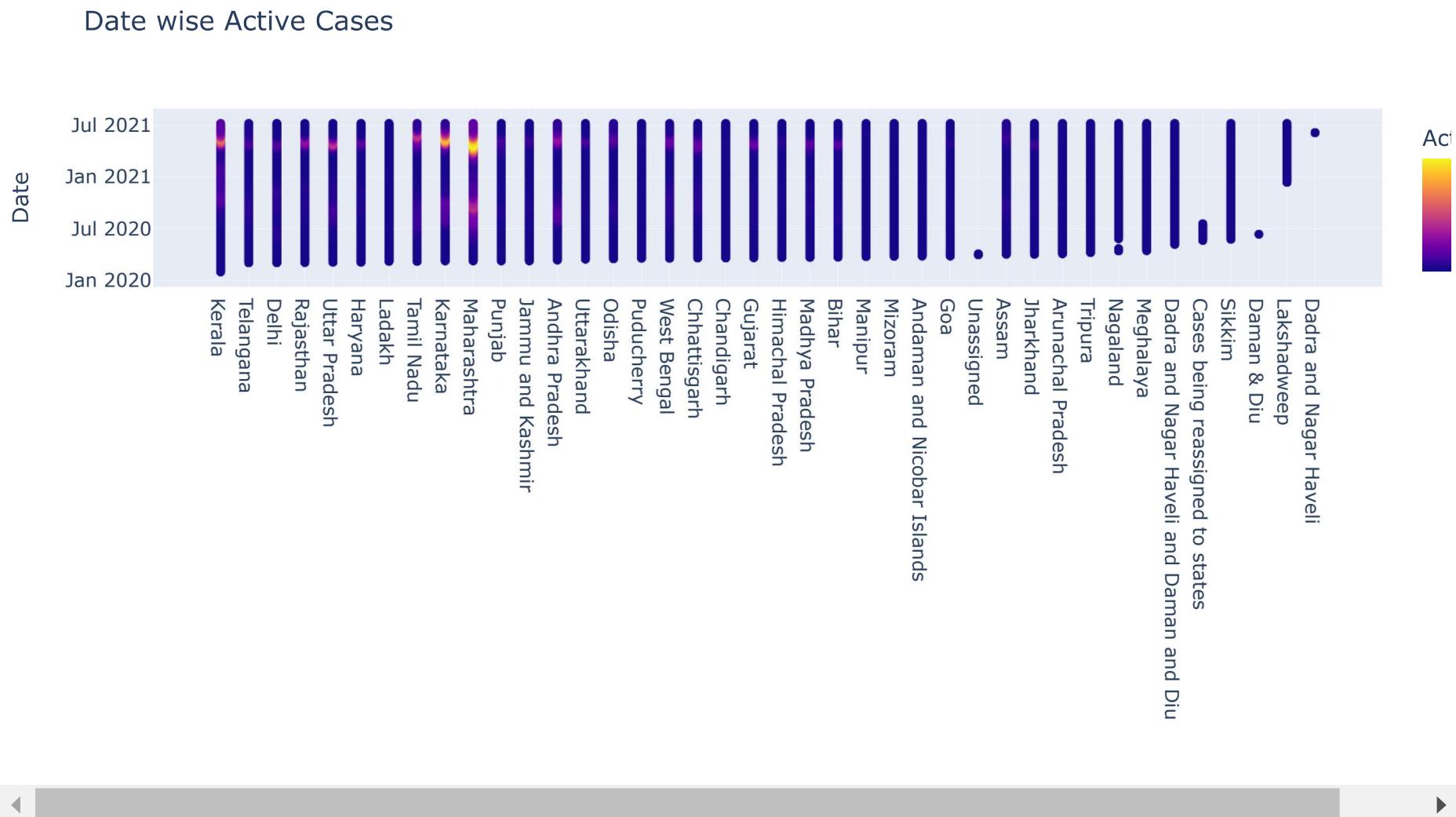
```
In [9]: 1 px.scatter(x='State/UnionTerritory',y='Date',data_frame=data1,color='Cured',title='Date wise Cured Cases')
```



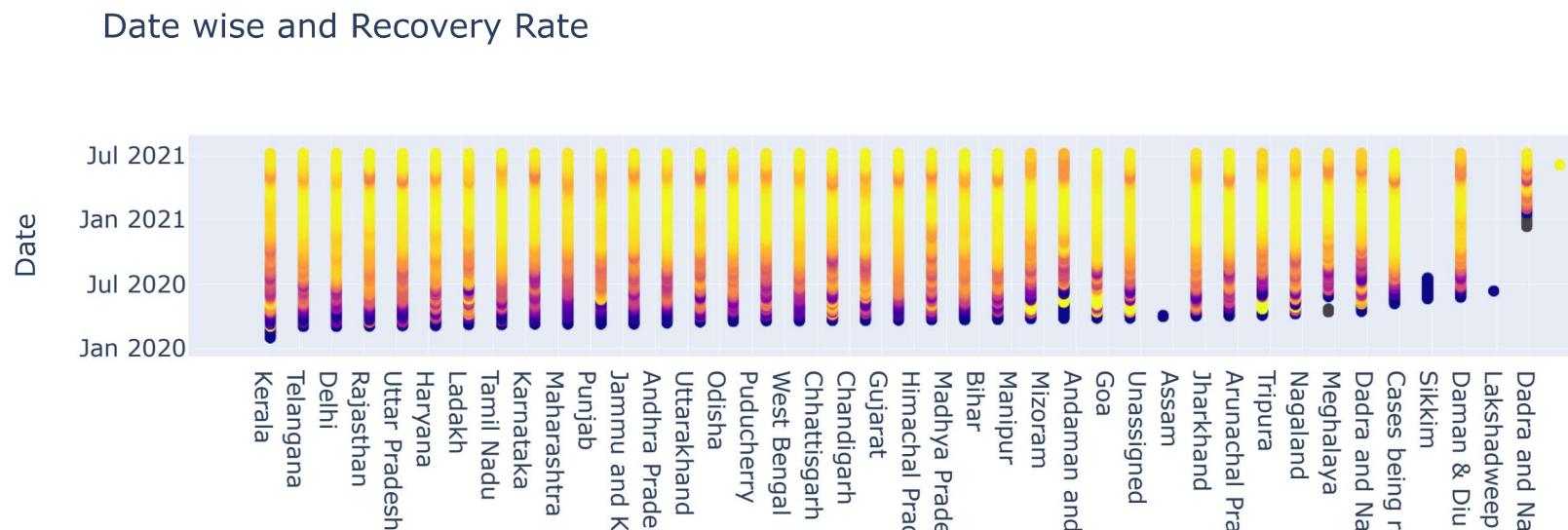
```
In [10]: 1 px.scatter(x='State/UnionTerritory',y='Date',data_frame=data1,color='Deaths',title='Date wise and Deaths Cases')
```



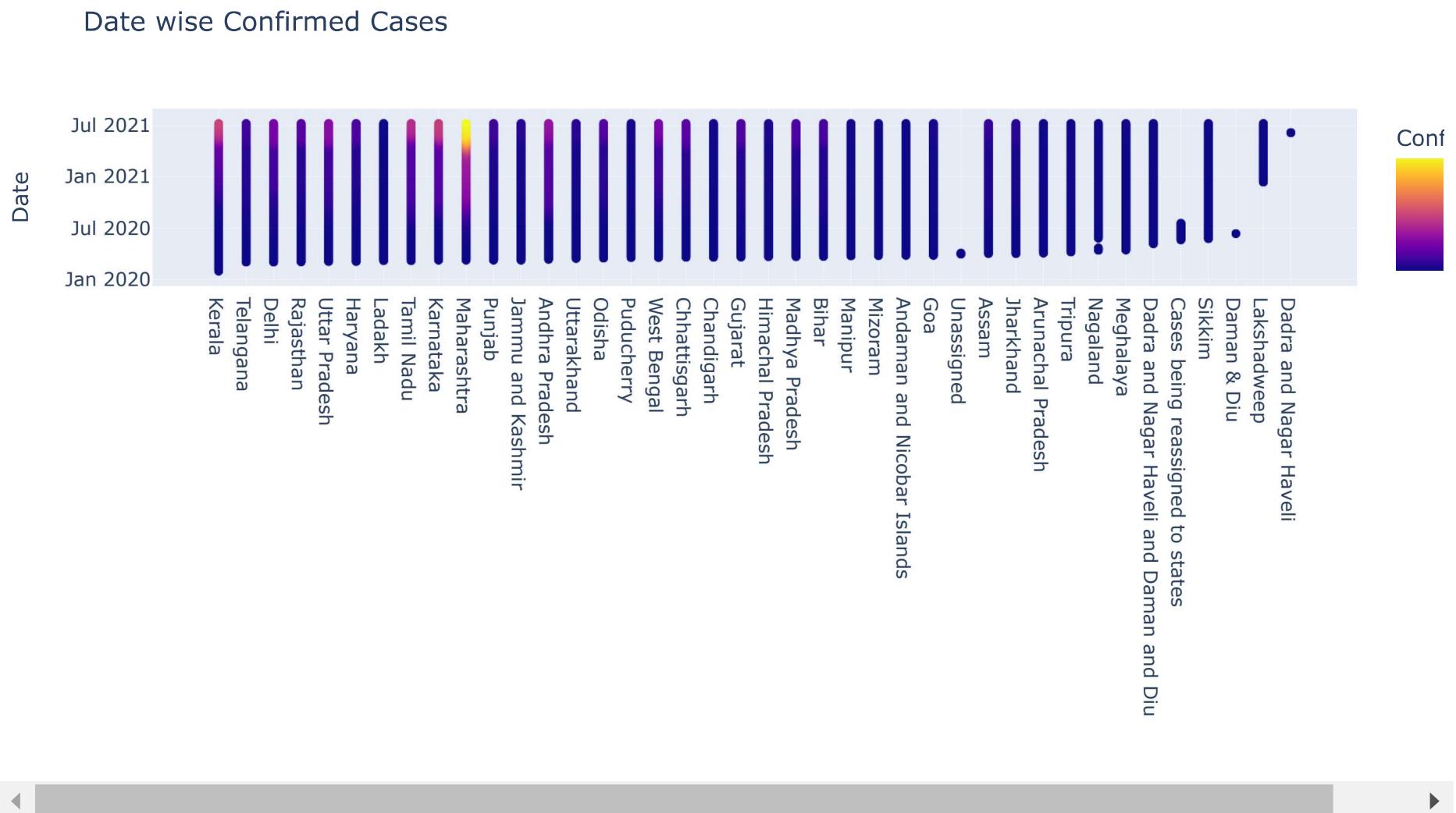
```
In [11]: 1 px.scatter(x='State/UnionTerritory',y='Date',data_frame=data1,color='Active',title='Date wise Active Cases')
```



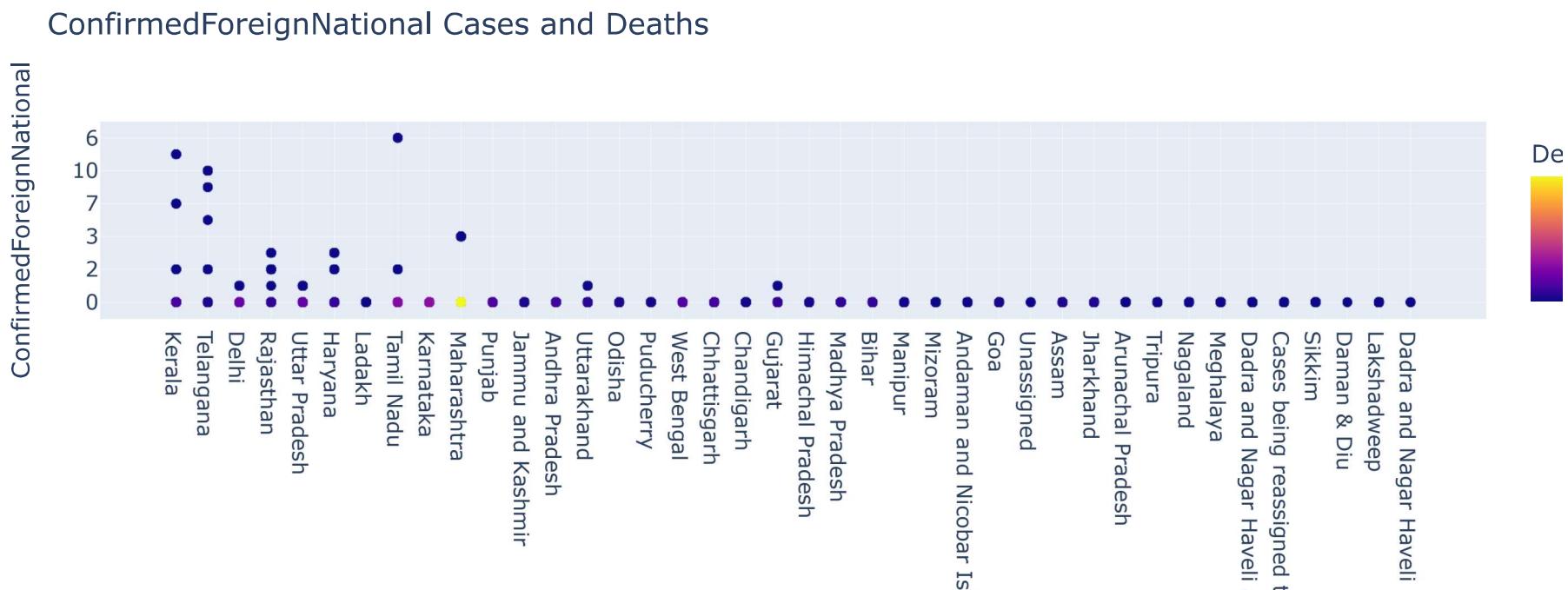
```
In [12]: 1 px.scatter(x='State/UnionTerritory',y='Date',color='Recovery Rate',data_frame=data1,title='Date wise and Recovery Ra
```



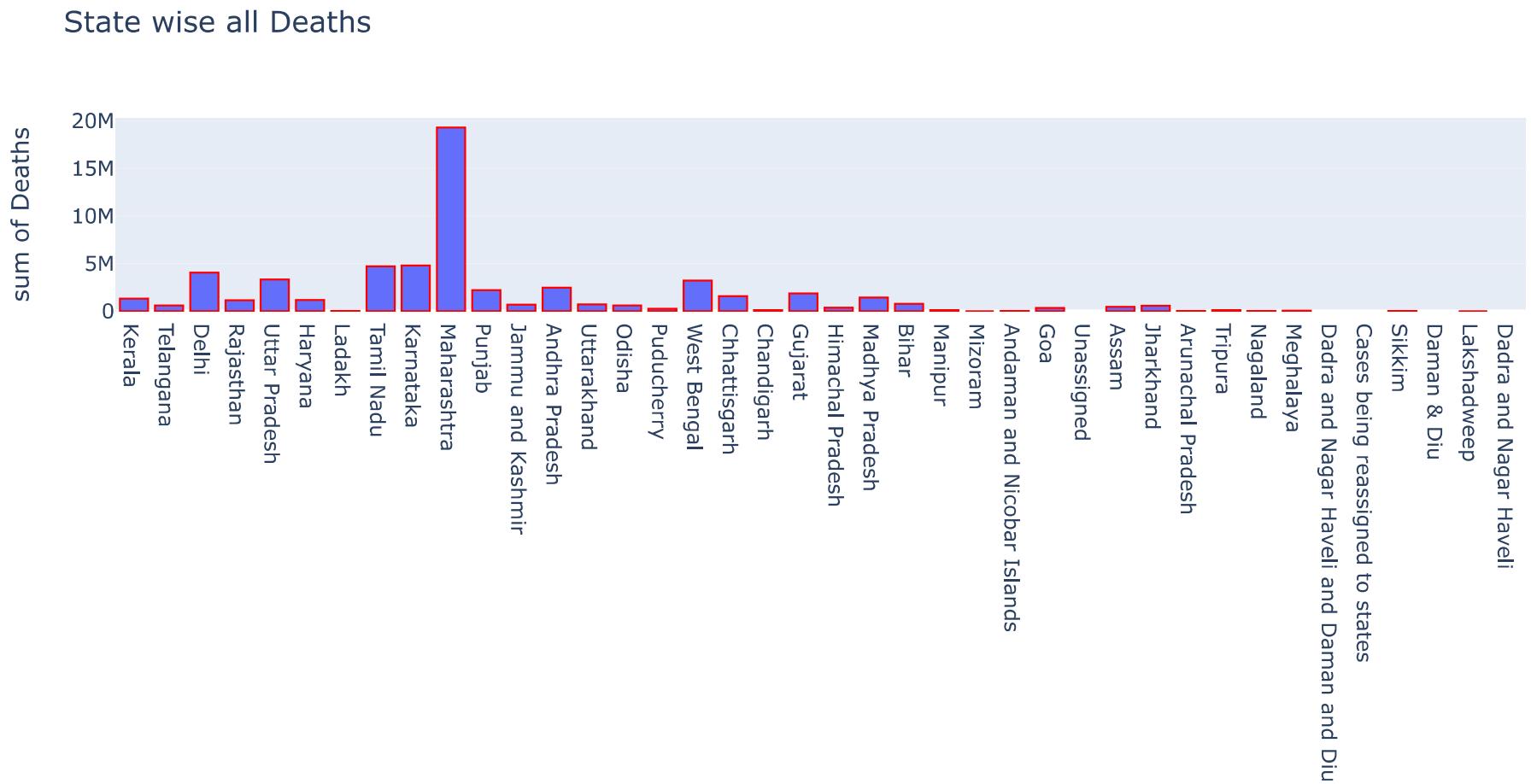
```
In [13]: 1 px.scatter(x='State/UnionTerritory',y='Date',color='Confirmed',data_frame=data1,title='Date wise Confirmed Cases')
```



```
In [14]: 1 px.scatter(x='State/UnionTerritory',y='ConfirmedForeignNational',data_frame=data1,color='Deaths',title='ConfirmedFor
```

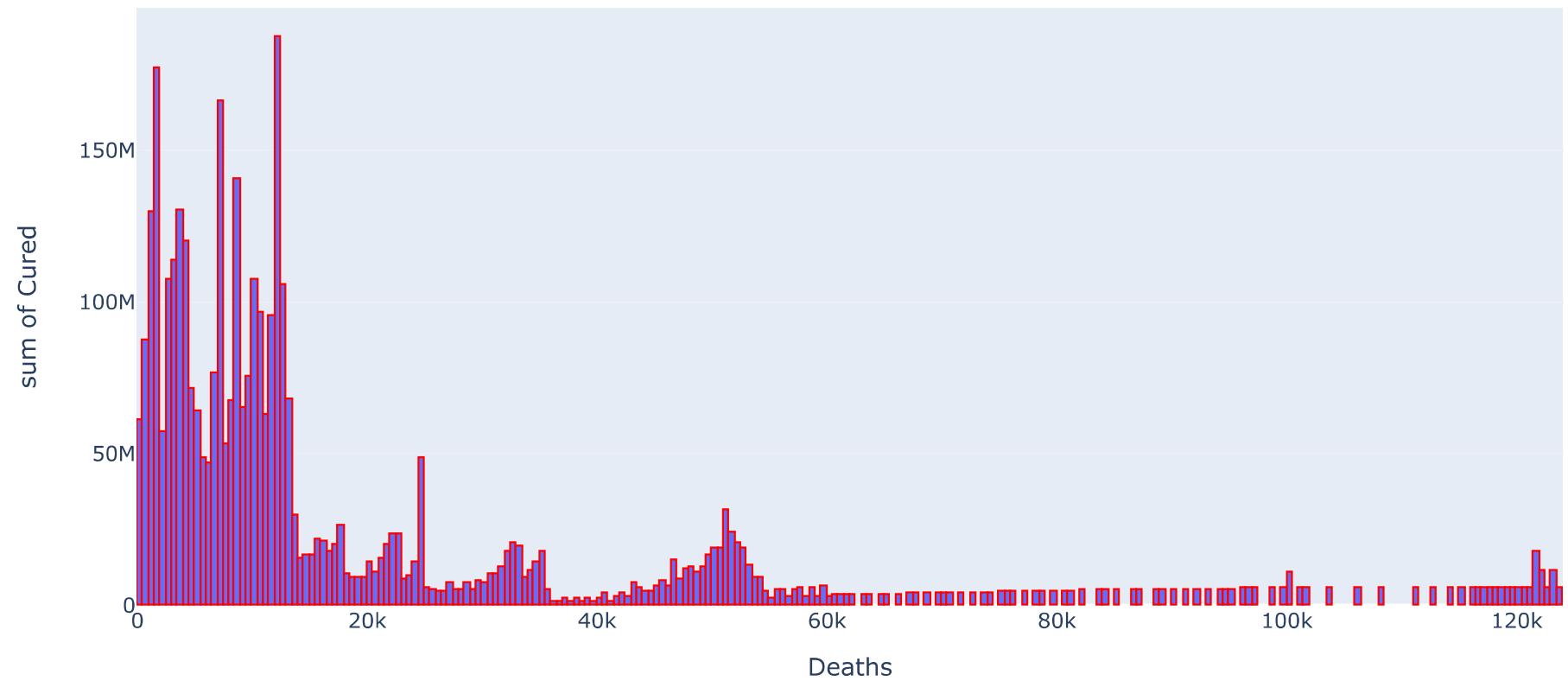


```
In [15]:  
1 fig = px.histogram(x='State/UnionTerritory',y='Deaths',data_frame=data1,title='State wise all Deaths')  
2 fig.update_traces(marker_line_width=1,marker_line_color='red')
```



```
In [16]: 1 fig = px.histogram(x='Deaths',y='Cured',data_frame=data1,title='Sum of Deaths Cases and Cured Cases')
2 fig.update_traces(marker_line_width=1,marker_line_color='red')
```

Sum of Deaths Cases and Cured Cases



In [17]: 1 | data1

Out[17]:

	Sno	Date	Time	State/Union Territory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Active	Recovery Rate
0	1	2020-01-30	6:00 PM	Kerala	1		0	0	0	1	1 0.000000
1	2	2020-01-31	6:00 PM	Kerala	1		0	0	0	1	1 0.000000
2	3	2020-02-01	6:00 PM	Kerala	2		0	0	0	2	2 0.000000
3	4	2020-02-02	6:00 PM	Kerala	3		0	0	0	3	3 0.000000
4	5	2020-02-03	6:00 PM	Kerala	3		0	0	0	3	3 0.000000
...	...	...	...	...	...		...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	0		0	613124	3703	628282	11455 97.587389
16846	16847	2021-07-07	8:00 AM	Tripura	0		0	63964	701	68612	3947 93.225675
16847	16848	2021-07-07	8:00 AM	Uttarakhand	0		0	332006	7338	340882	1538 97.396166
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	0		0	1682130	22656	1706818	2032 98.553566
16849	16850	2021-07-07	8:00 AM	West Bengal	0		0	1472132	17834	1507241	17275 97.670645

16850 rows × 11 columns

In [18]:

```
1 #Describe all Data  
2 data1.describe()
```

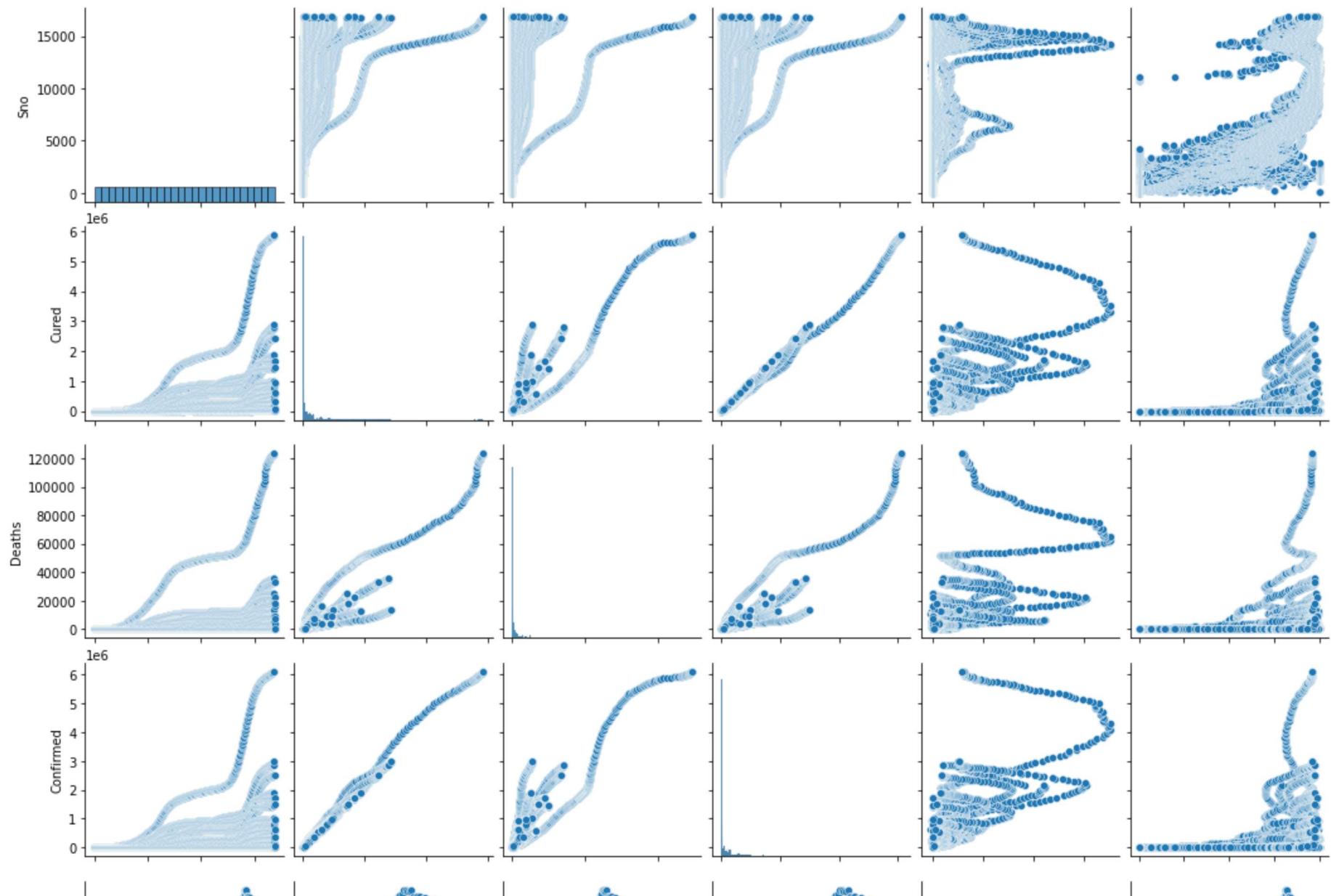
Out[18]:

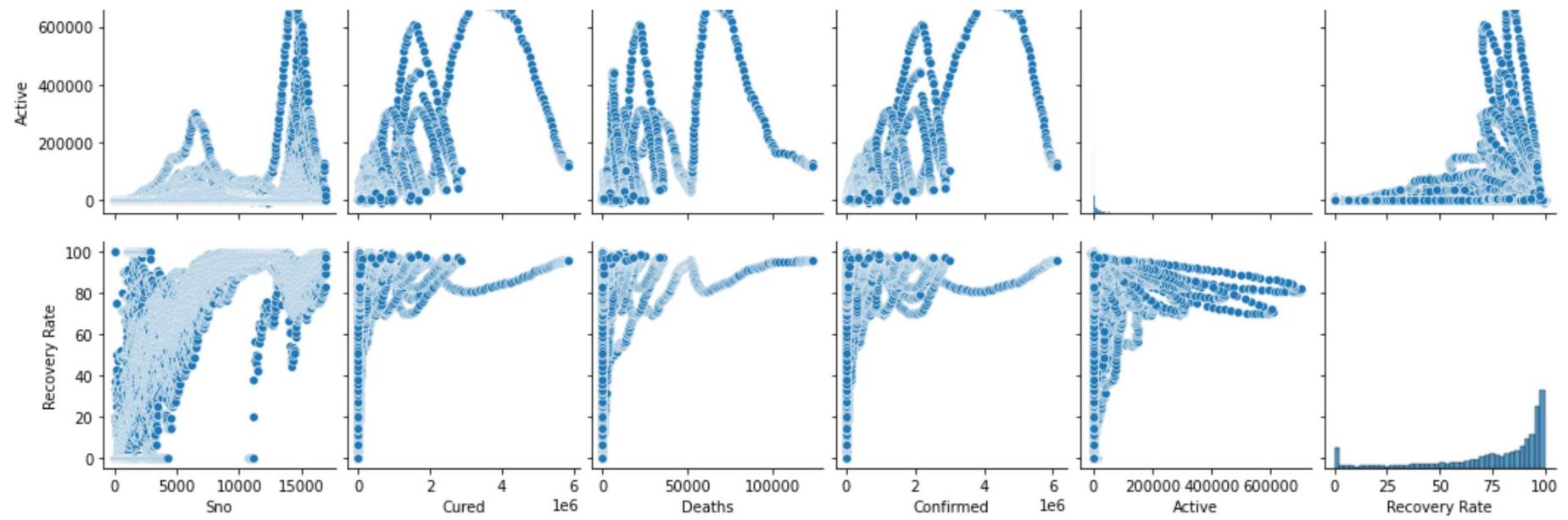
	Sno	Cured	Deaths	Confirmed	Active	Recovery Rate
<b>count</b>	16850.000000	1.685000e+04	16850.000000	1.685000e+04	16850.000000	16804.000000
<b>mean</b>	8425.500000	2.360353e+05	3485.222552	2.583667e+05	18846.168427	75.628702
<b>std</b>	4864.320353	5.225438e+05	9330.541749	5.672808e+05	54267.658198	27.777154
<b>min</b>	1.000000	0.000000e+00	0.000000	0.000000e+00	-9368.000000	0.000000
<b>25%</b>	4213.250000	2.658500e+03	22.000000	3.644750e+03	309.250000	65.855846
<b>50%</b>	8425.500000	2.889500e+04	453.000000	3.336150e+04	2394.500000	87.313332
<b>75%</b>	12637.750000	2.537510e+05	3071.250000	2.666530e+05	12838.500000	96.598930
<b>max</b>	16850.000000	5.872268e+06	123531.000000	6.113335e+06	701614.000000	100.000000

In [19]:

```
1 #Pairplot  
2 sns.pairplot(data1)
```

Out[19]: &lt;seaborn.axisgrid.PairGrid at 0x188ee3ddf70&gt;





In [20]: 1 all\_deaths = data1['Deaths'].sum()

In [21]: 1 #All Cured Cases  
2 data1['Cured'].sum()

Out[21]: 3977194136

In [22]: 1 all\_con = data1['Confirmed'].sum()

In [23]: 1 #All Deaths Cases  
2 all\_deaths

Out[23]: 58726000

In [24]: 1 #All Confirmed Cases  
2 all\_con

Out[24]: 4353478074

```
In [25]: 1 recover = all_con - all_deaths
```

```
In [26]: 1 #ALL Recover Cases  
2 recover
```

Out[26]: 4294752074

## Maharashtra State Data

```
In [27]: 1 data_maha = data1[data1['State/UnionTerritory']=='Maharashtra']  
2
```

```
In [28]: 1 data_maha.replace('-', '0', inplace=True)
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py:5238: SettingWithCopyWarning:

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

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

In [29]:

```
1 #Maharashtra Data  
2 data_maha.describe()
```

Out[29]:

	Sno	Cured	Deaths	Confirmed	Active	Recovery Rate
<b>count</b>	486.000000	4.860000e+02	486.000000	4.860000e+02	486.000000	486.000000
<b>mean</b>	8191.660494	1.674463e+06	39741.835391	1.870149e+06	155944.508230	70.262077
<b>std</b>	4955.978167	1.710989e+06	31861.231600	1.831266e+06	168833.300211	29.635395
<b>min</b>	77.000000	0.000000e+00	0.000000	2.000000e+00	2.000000	0.000000
<b>25%</b>	3884.000000	1.197165e+05	9299.500000	2.187718e+05	46101.000000	54.608309
<b>50%</b>	8138.500000	1.556812e+06	44884.500000	1.706879e+06	96492.500000	83.593260
<b>75%</b>	12470.000000	2.066541e+06	52468.500000	2.216942e+06	193650.750000	93.463383
<b>max</b>	16835.000000	5.872268e+06	123531.000000	6.113335e+06	701614.000000	96.056702

In [30]: 1 | data\_maha

Out[30]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Active	Recovery Rate
76	77	2020-03-09	6:00 PM	Maharashtra	2		0	0	0	2	2 0.000000
91	92	2020-03-10	6:00 PM	Maharashtra	5		0	0	0	5	5 0.000000
97	98	2020-03-11	6:00 PM	Maharashtra	2		0	0	0	2	2 0.000000
120	121	2020-03-12	6:00 PM	Maharashtra	11		0	0	0	11	11 0.000000
133	134	2020-03-13	6:00 PM	Maharashtra	14		0	0	0	14	14 0.000000
...	...	...	...	...	...		...	...	...	...	...
16690	16691	2021-07-03	8:00 AM	Maharashtra	0		0 5836920	122353	6079352	120079	96.012207
16726	16727	2021-07-04	8:00 AM	Maharashtra	0		0 5845315	122724	6088841	120802	96.000454
16762	16763	2021-07-05	8:00 AM	Maharashtra	0		0 5848693	123030	6098177	126454	95.908876
16798	16799	2021-07-06	8:00 AM	Maharashtra	0		0 5861720	123136	6104917	120061	96.016375
16834	16835	2021-07-07	8:00 AM	Maharashtra	0		0 5872268	123531	6113335	117536	96.056702

486 rows × 11 columns



```
In [31]: 1 data_maha.isnull().sum()
```

```
Out[31]: Sno          0  
Date         0  
Time         0  
State/UnionTerritory 0  
ConfirmedIndianNational 0  
ConfirmedForeignNational 0  
Cured         0  
Deaths        0  
Confirmed      0  
Active         0  
Recovery Rate   0  
dtype: int64
```

In [32]:

```
1 data_maha.fillna(value=0,inplace=True)
2 data_maha[ 'Active' ] = data_maha[ 'Confirmed' ] - (data_maha[ 'Cured' ] + data_maha[ 'Deaths' ])
3 data_maha[ 'Recovery Rate' ] = data_maha[ 'Cured'*100 / data_maha[ 'Confirmed' ] ]
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py:5176: SettingWithCopyWarning:

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

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

C:\Users\Tejas\AppData\Local\Temp\ipykernel\_7360/1440047907.py:2: SettingWithCopyWarning:

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

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

C:\Users\Tejas\AppData\Local\Temp\ipykernel\_7360/1440047907.py:3: SettingWithCopyWarning:

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

Try using .loc[row\_indexer,col\_indexer] = value instead

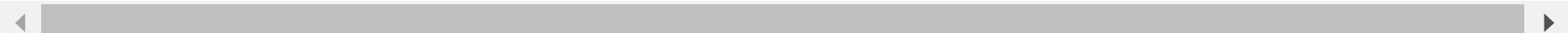
See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

In [33]:

```
1 #Maximum Deaths,Cured,Confirmed,Recovery Rate,Active
2 data_maha[data_maha['Deaths'] == data_maha['Deaths'].max()]
```

Out[33]:

Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Active	Recovery Rate
16834	16835	2021-07-07	8:00 AM	Maharashtra	0	0	5872268	123531	6113335	117536 96.056702



In [34]:

```
1 #Minimum Deaths
2 data_maha[data_maha['Deaths'] == data_maha['Deaths'].min()]
```

Out[34]:

Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Active	Recovery Rate
76	77	2020-03-09	6:00 PM	Maharashtra	2	0	0	0	2	2 0.0
91	92	2020-03-10	6:00 PM	Maharashtra	5	0	0	0	5	5 0.0
97	98	2020-03-11	6:00 PM	Maharashtra	2	0	0	0	2	2 0.0
120	121	2020-03-12	6:00 PM	Maharashtra	11	0	0	0	11	11 0.0
133	134	2020-03-13	6:00 PM	Maharashtra	14	0	0	0	14	14 0.0
146	147	2020-03-14	6:00 PM	Maharashtra	14	0	0	0	14	14 0.0
153	154	2020-03-15	6:00 PM	Maharashtra	32	0	0	0	32	32 0.0
167	168	2020-03-16	6:00 PM	Maharashtra	32	0	0	0	32	32 0.0

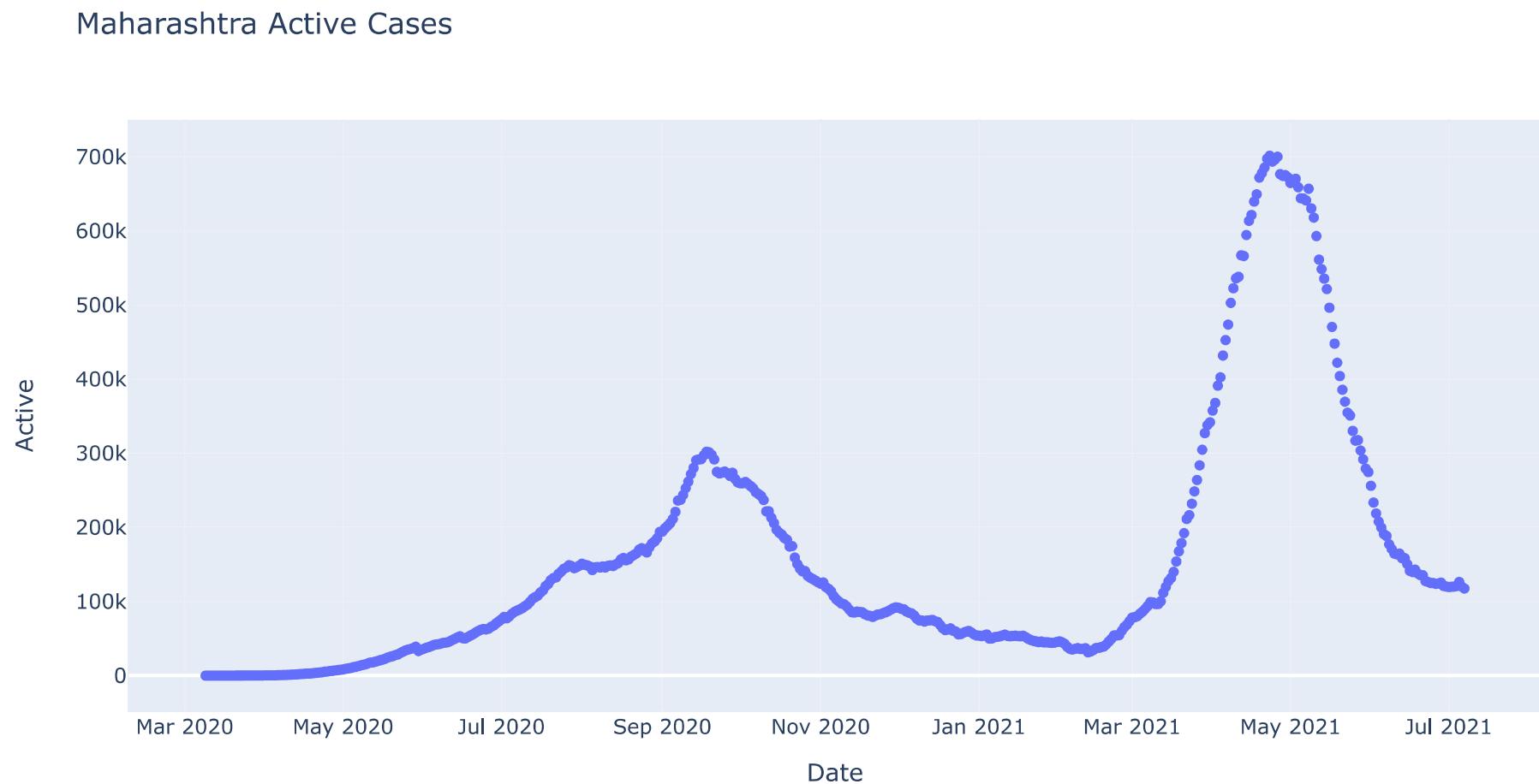
In [35]: 1 data\_maha.isnull().sum()

Out[35]:

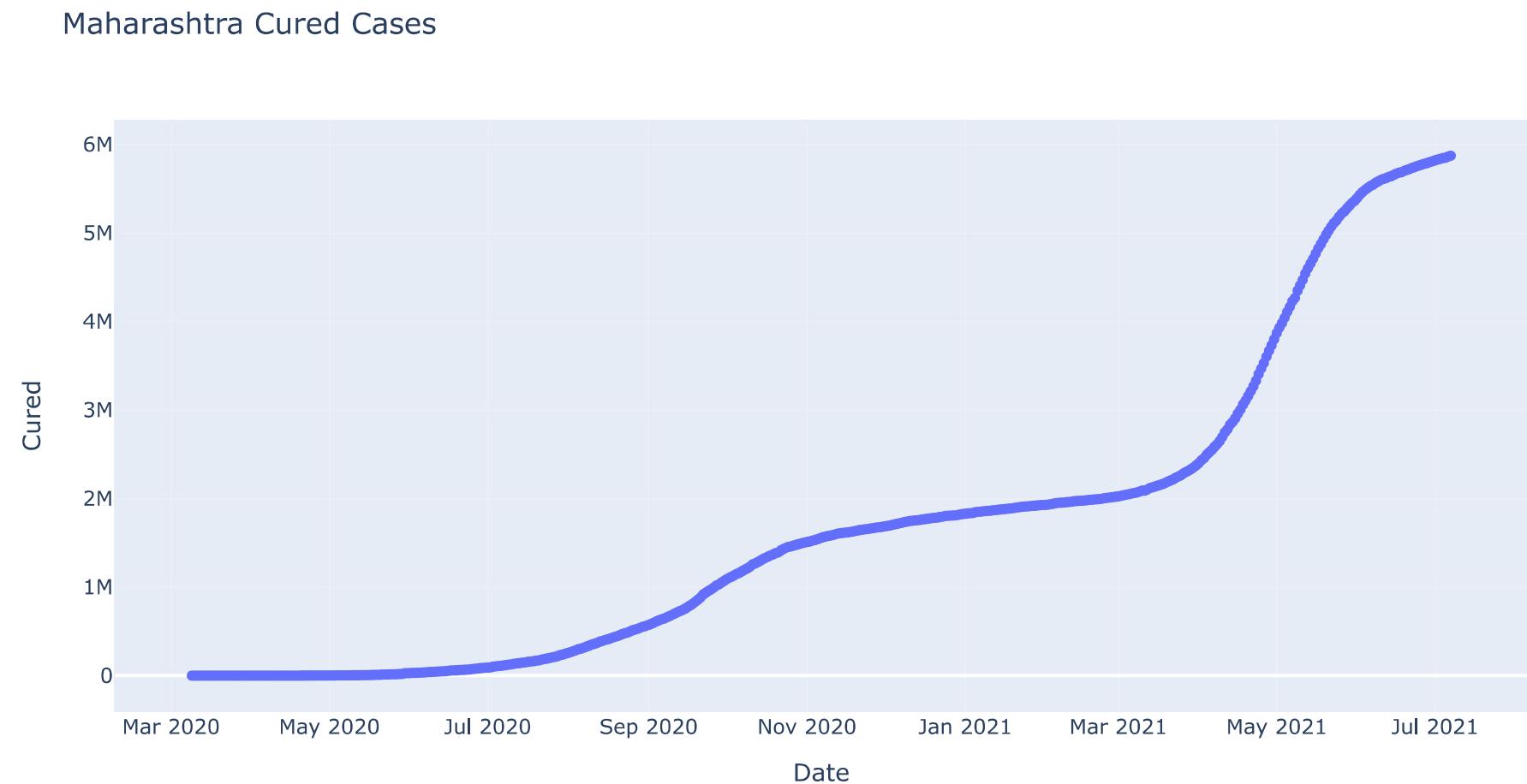
Sno	0
Date	0
Time	0
State/UnionTerritory	0
ConfirmedIndianNational	0
ConfirmedForeignNational	0
Cured	0
Deaths	0
Confirmed	0
Active	0
Recovery Rate	0

dtype: int64

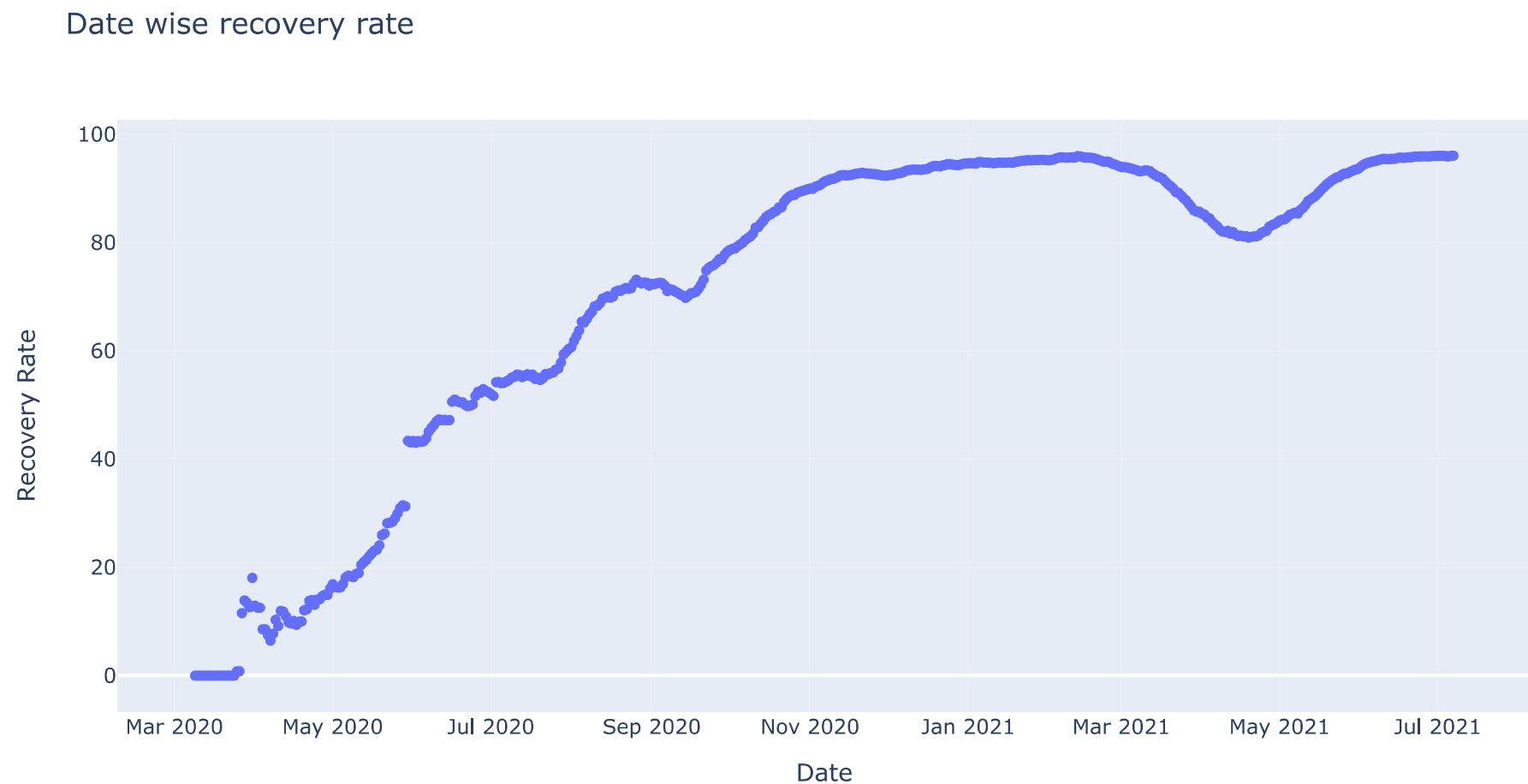
```
In [36]: 1 px.scatter(x='Date',y='Active',title='Maharashtra Active Cases ',data_frame=data_maha)
```



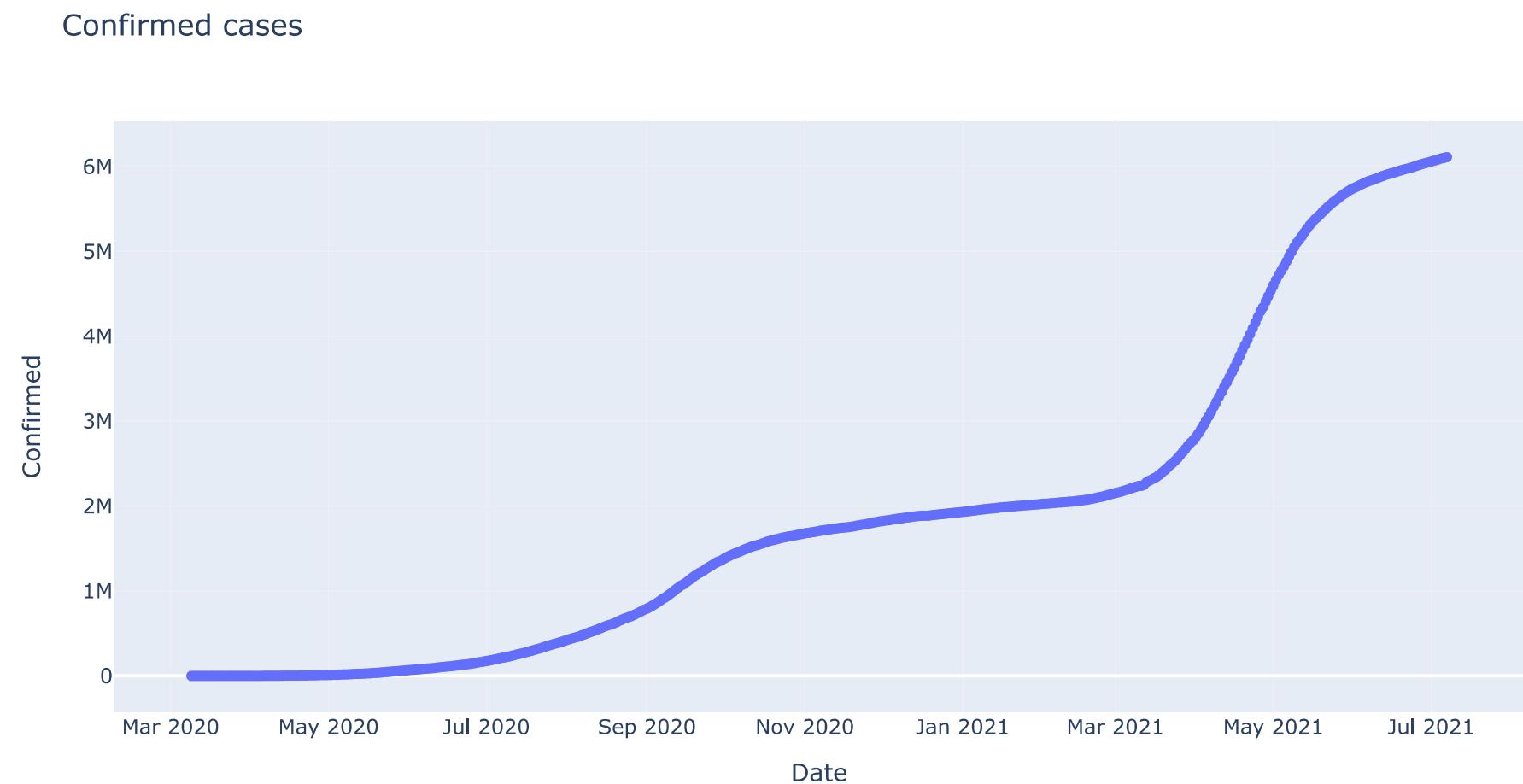
```
In [37]: 1 px.scatter(x='Date',y='Cured',title='Maharashtra Cured Cases',data_frame=data_maha)
```



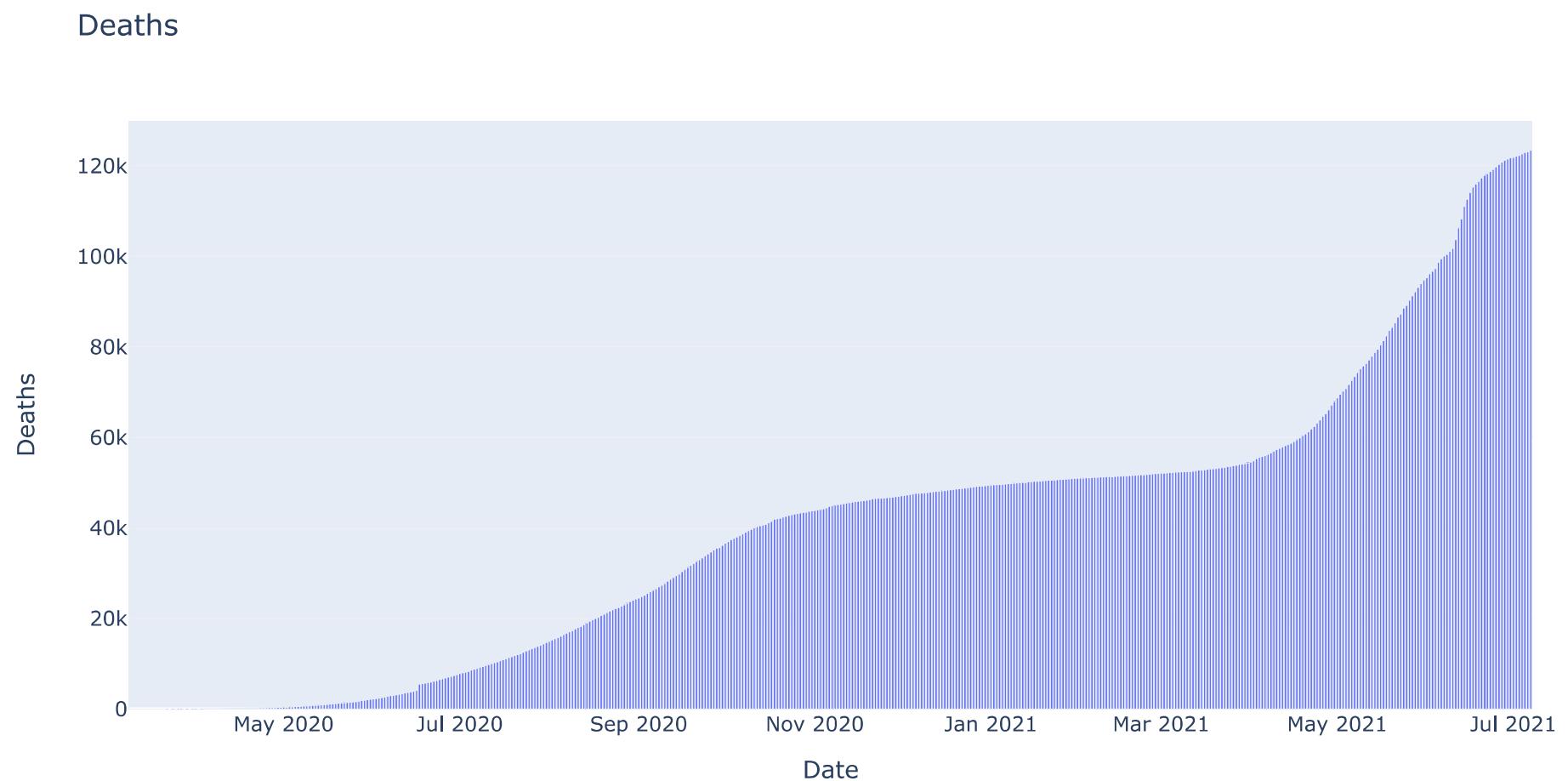
```
In [38]: 1 px.scatter(x='Date',y='Recovery Rate',title='Date wise recovery rate',data_frame=data_maha)
```



```
In [39]: 1 px.scatter(x='Date',y='Confirmed',title='Confirmed cases',data_frame=data_maha)
```

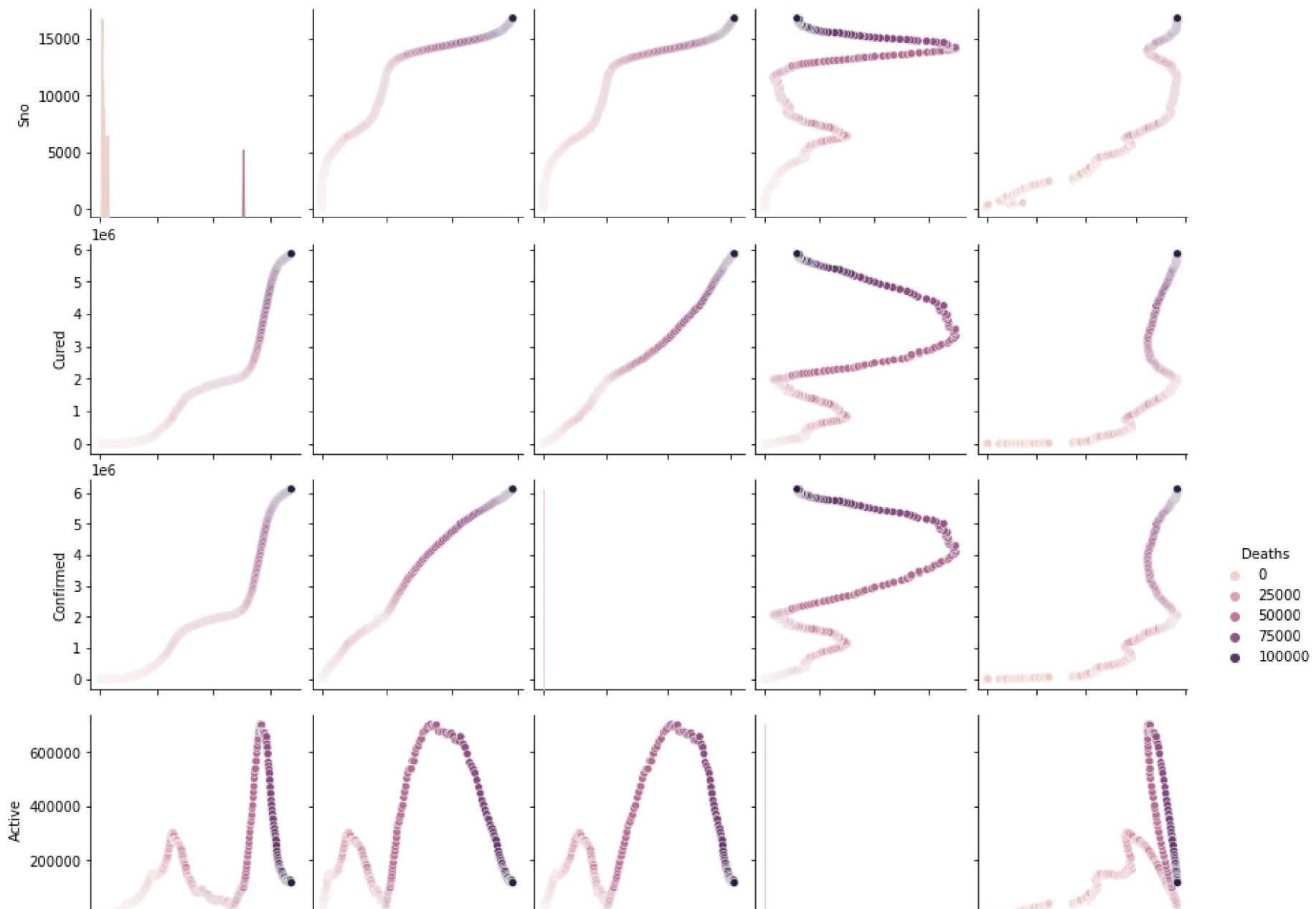


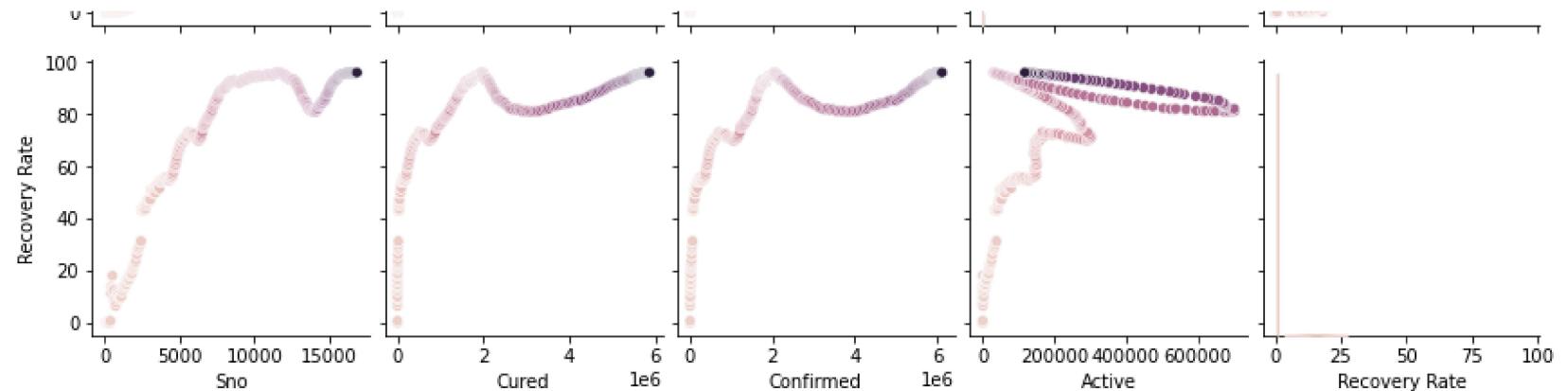
```
In [40]: 1 px.bar(x='Date',y='Deaths',title='Deaths',data_frame=data_maha)
```



In [41]: 1 sns.pairplot(data\_maha,hue='Deaths',kind='scatter')

Out[41]: <seaborn.axisgrid.PairGrid at 0x188f2e71520>





In [42]:

```
1 #Sum of Deaths  
2 data_maha['Deaths'].sum()
```

Out[42]: 19314532

In [43]:

```
1 #Sum of Confirmed  
2 data_maha['Confirmed'].sum()
```

Out[43]: 908892470

In [44]:

```
1 #sum of active cases  
2 data_maha['Active'].sum()
```

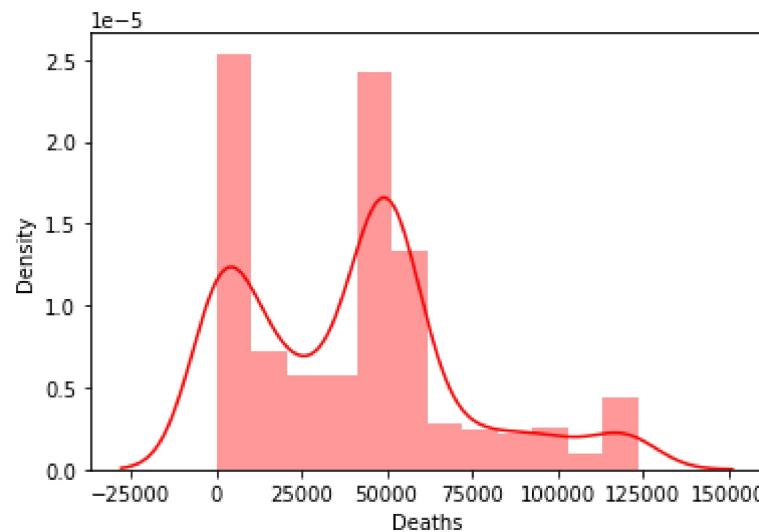
Out[44]: 75789031

```
In [45]: 1 sns.distplot(data_maha['Deaths'],color='red')
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

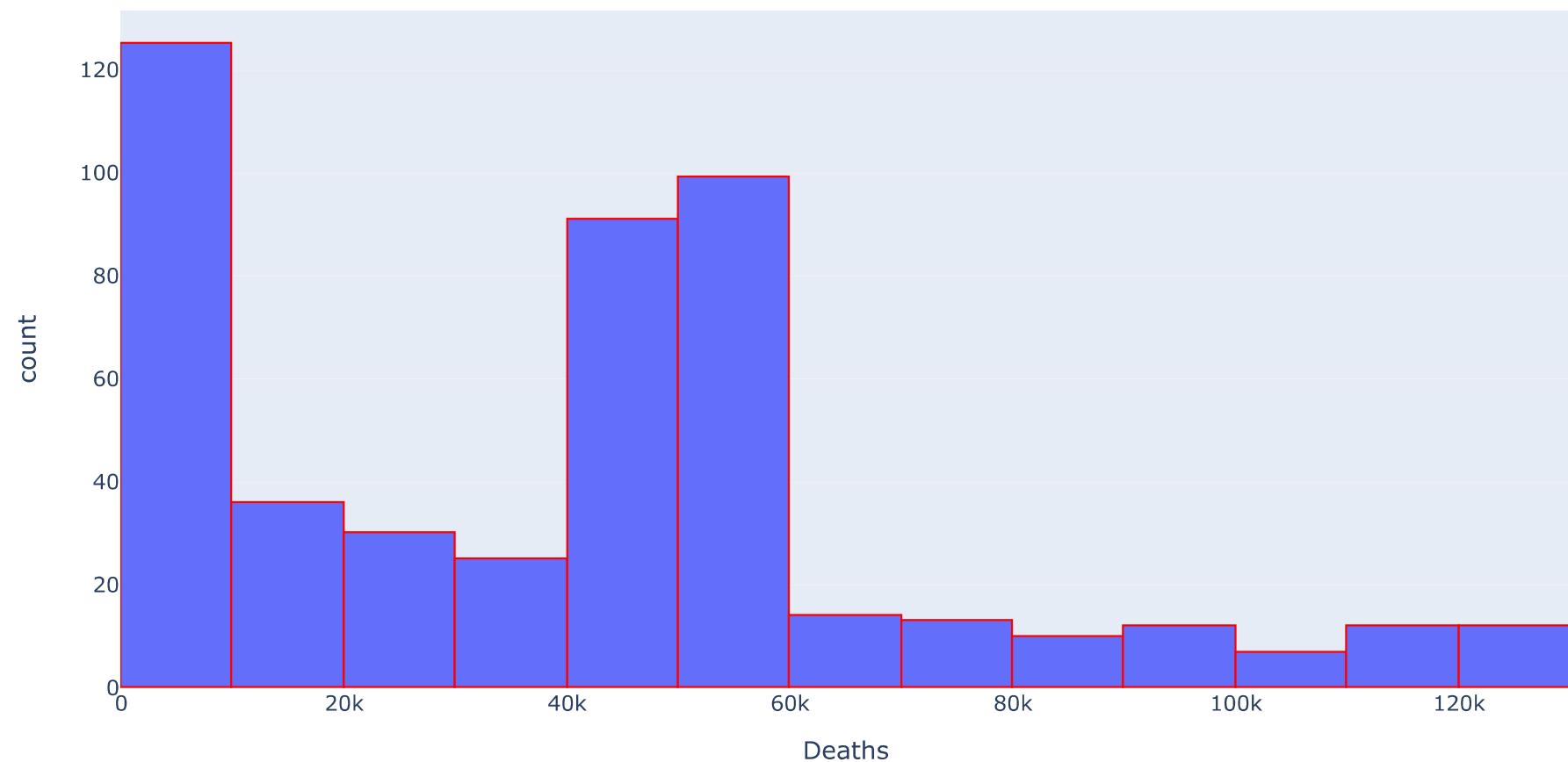
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
Out[45]: <AxesSubplot:xlabel='Deaths', ylabel='Density'>
```

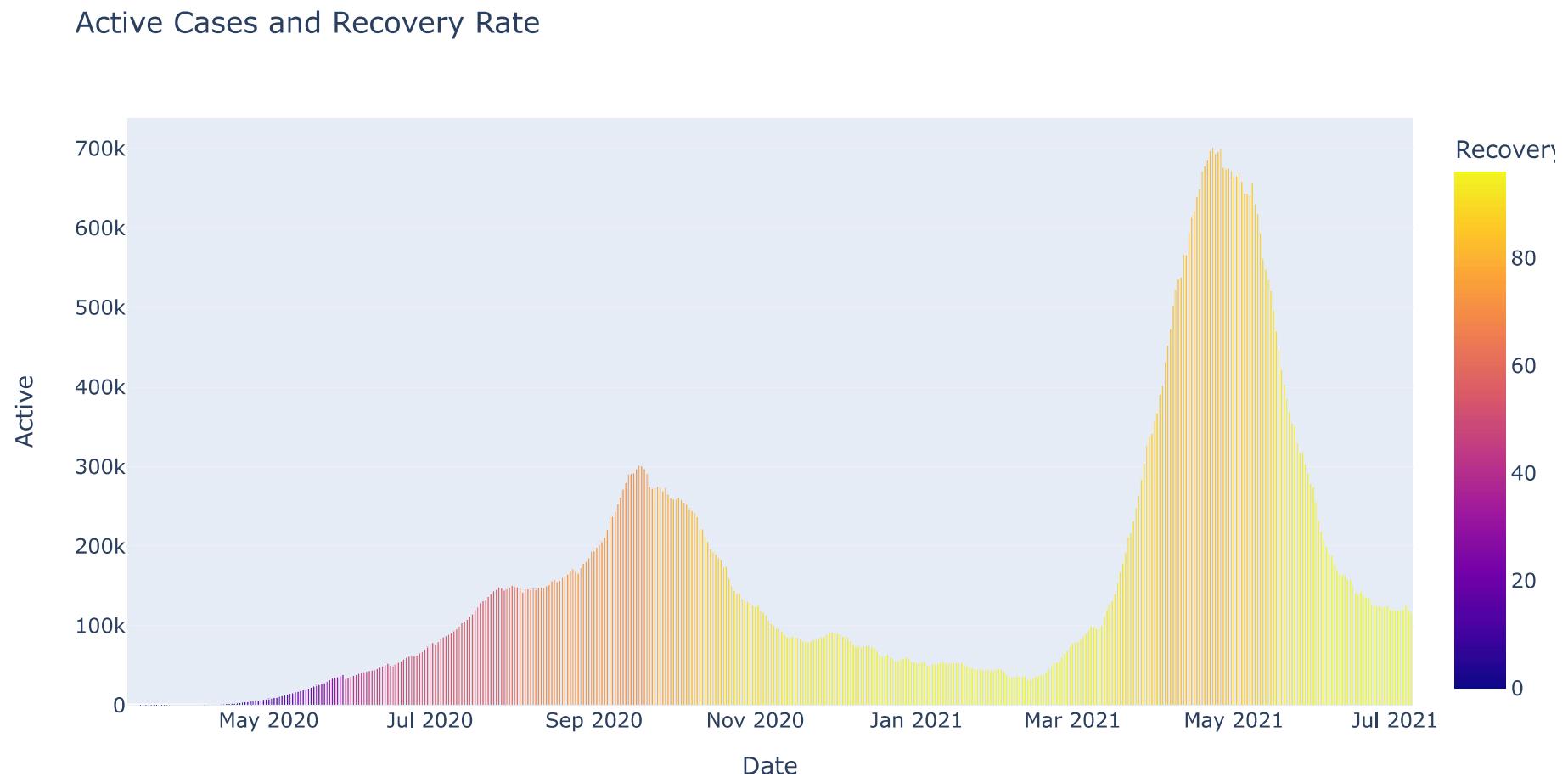


In [46]:

```
1 fig = px.histogram(x='Deaths',data_frame=data_maha)
2 fig.update_traces(marker_line_width=1,marker_line_color="red")
```

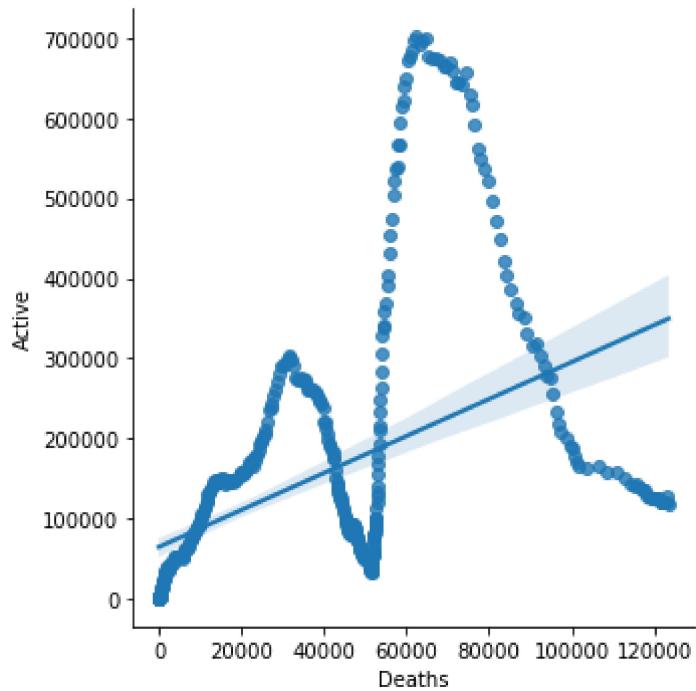


```
In [47]: 1 px.bar(x='Date',y='Active',data_frame=data_maha,color='Recovery Rate',title="Active Cases and Recovery Rate")
```



In [48]: 1 sns.lmplot(x='Deaths',y='Active',data=data\_maha)

Out[48]: <seaborn.axisgrid.FacetGrid at 0x188f4237ca0>



In [49]:

```
1 #top 10 Date by Active State
2 top_10_Active_cases = data_maha.groupby(['Date'])['Active'].sum()
3 top_10 = top_10_Active_cases.sort_values(ascending=False).head(10)
4 pd.DataFrame(top_10)
```

Out[49]:

Date	Active
2021-04-23	701614
2021-04-26	700207
2021-04-22	697467
2021-04-25	696298
2021-04-24	693632
2021-04-21	685552
2021-04-20	678198
2021-04-27	676647
2021-04-29	675451
2021-04-28	674358

In [50]:

```
1 #top 10 Date by Deaths State
2 top_10_Deaths_cases = data_maha.groupby(['Date'])['Deaths'].sum()
3 top_10 = top_10_Deaths_cases.sort_values(ascending=False).head(10)
4 pd.DataFrame(top_10)
```

Out[50]:

Date	Deaths
2021-07-07	123531
2021-07-06	123136
2021-07-05	123030
2021-07-04	122724
2021-07-03	122353
2021-07-02	122197
2021-07-01	121945
2021-06-30	121804
2021-06-29	121573
2021-06-28	121286

In [51]:

```
1 #top 10 Date by Cured State
2 top_10_Cured_cases = data_maha.groupby(['Date'])['Cured'].sum()
3 top_10 = top_10_Cured_cases.sort_values(ascending=False).head(10)
4 pd.DataFrame(top_10)
```

Out[51]:

Cured	Date
5872268	2021-07-07
5861720	2021-07-06
5848693	2021-07-05
5845315	2021-07-04
5836920	2021-07-03
5828535	2021-07-02
5819901	2021-07-01
5809548	2021-06-30
5800925	2021-06-29
5790113	2021-06-28

**Thank You.**

In [ ]:

1

