In [11]:

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

In [3]:

```
df.isna().sum()#df.isnull.sum()
```

Out[3]:

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

In [4]:

```
df.drop(["ConfirmedIndianNational","ConfirmedForeignNational"],inplace=True,axis=1)
df
```

Out[4]:

	Sno	Date	Time	State/UnionTerritory	Cured	Deaths	Confirmed
0	1	2020-01-30	6:00 PM	Kerala	0	0	1
1	2	2020-01-31	6:00 PM	Kerala	0	0	1
2	3	2020-02-01	6:00 PM	Kerala	0	0	2
3	4	2020-02-02	6:00 PM	Kerala	0	0	3
4	5	2020-02-03	6:00 PM	Kerala	0	0	3
...	...	...	...	...	...	...	...
16845	16846	2021-07-07	8:00 AM	Telangana	613124	3703	628282
16846	16847	2021-07-07	8:00 AM	Tripura	63964	701	68612
16847	16848	2021-07-07	8:00 AM	Uttarakhand	332006	7338	340882
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	1682130	22656	1706818
16849	16850	2021-07-07	8:00 AM	West Bengal	1472132	17834	1507241

16850 rows × 7 columns

In [12]:

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

```
<class 'pandas.core.series.Series'>
RangeIndex: 16850 entries, 0 to 16849
Series name: Date
Non-Null Count  Dtype
--------------  -----
16850 non-null  datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 131.8 KB
```

In [ ]:

```
drop unsigned replace tel,diu daman,bihar
```

active cases

In [13]:

```
df["Active_cases"]=df["Confirmed"]-(df["Cured"]+df["Deaths"])
df.head()
```

Out[13]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	C
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 [16]:

```
statewise=df.groupby("State/UnionTerritory")[["Confirmed","Cured","Deaths","Active_cases"]]
```

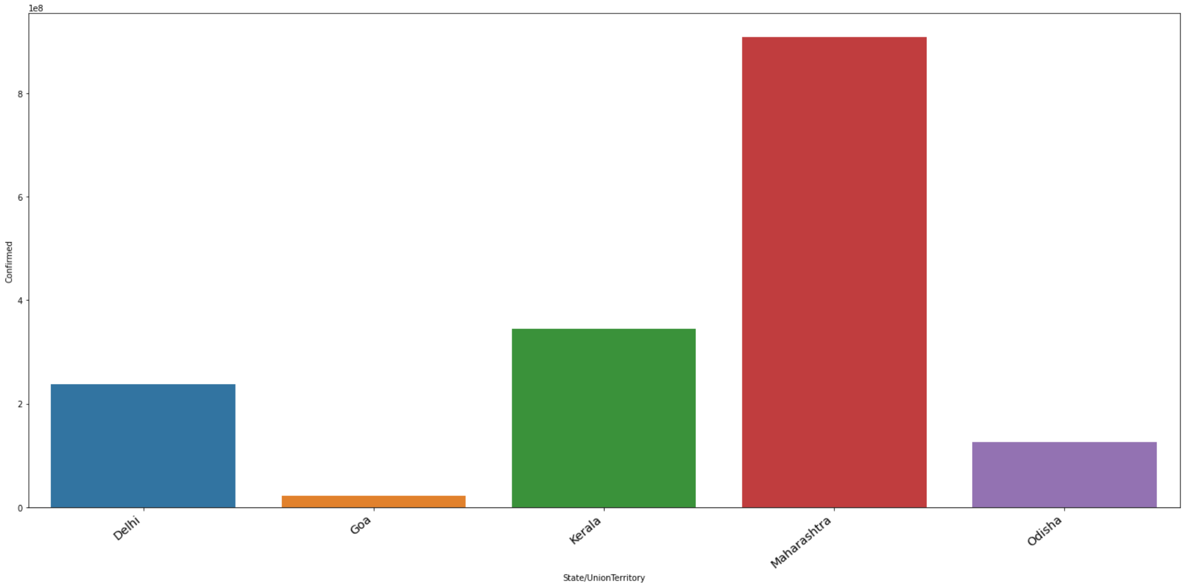
In [20]:

```
statewise_filter=statewise[statewise["State/UnionTerritory"].isin(["Maharashtra","Kerala","
```

state and confirmed

In [50]:

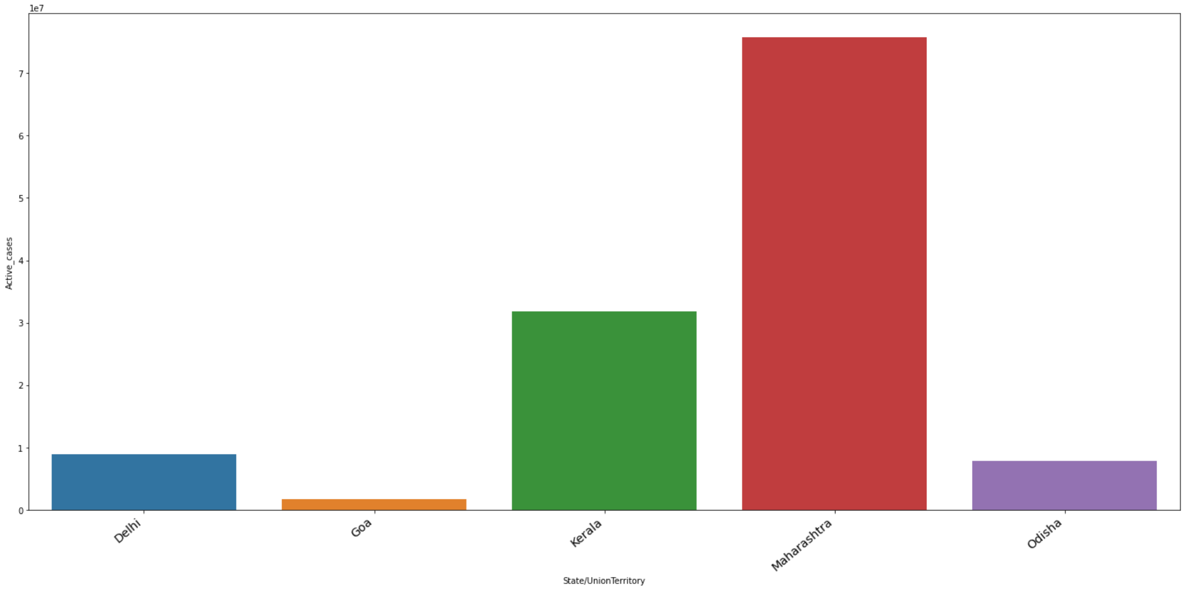
```
fig=plt.figure(figsize=(20,10))
sns.barplot(x="State/UnionTerritory",y="Confirmed",data=statewise_filter,palette="tab10")
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



state and active_cases

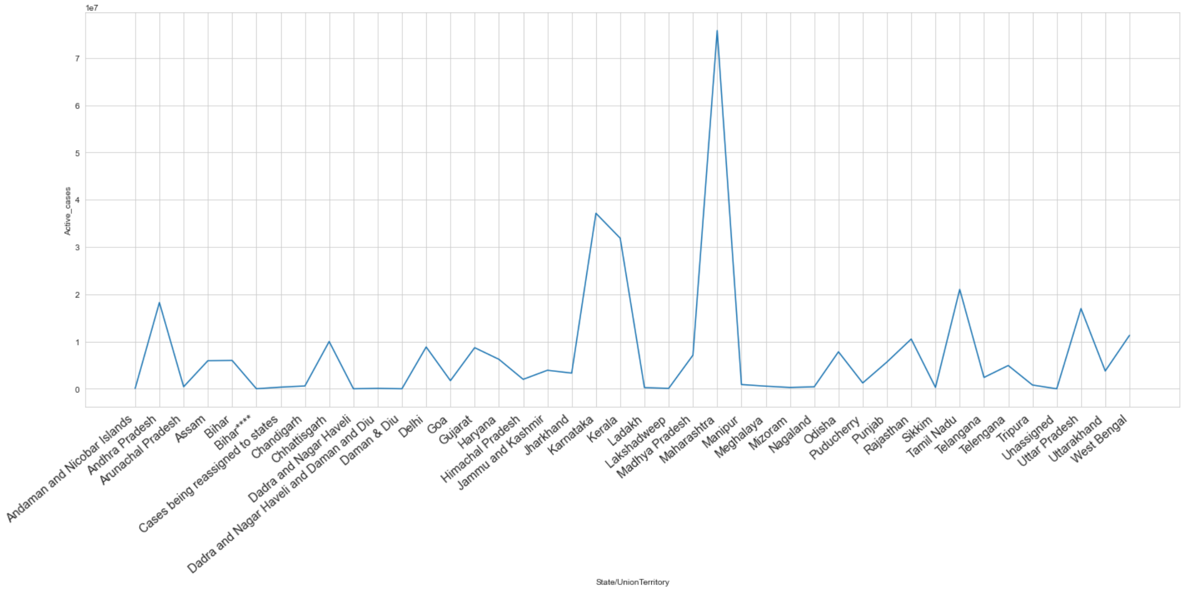
In [51]:

```
fig=plt.figure(figsize=(20,10))
sns.barplot(x="State/UnionTerritory",y="Active_cases",data=statewise_filter,palette="tab10"
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



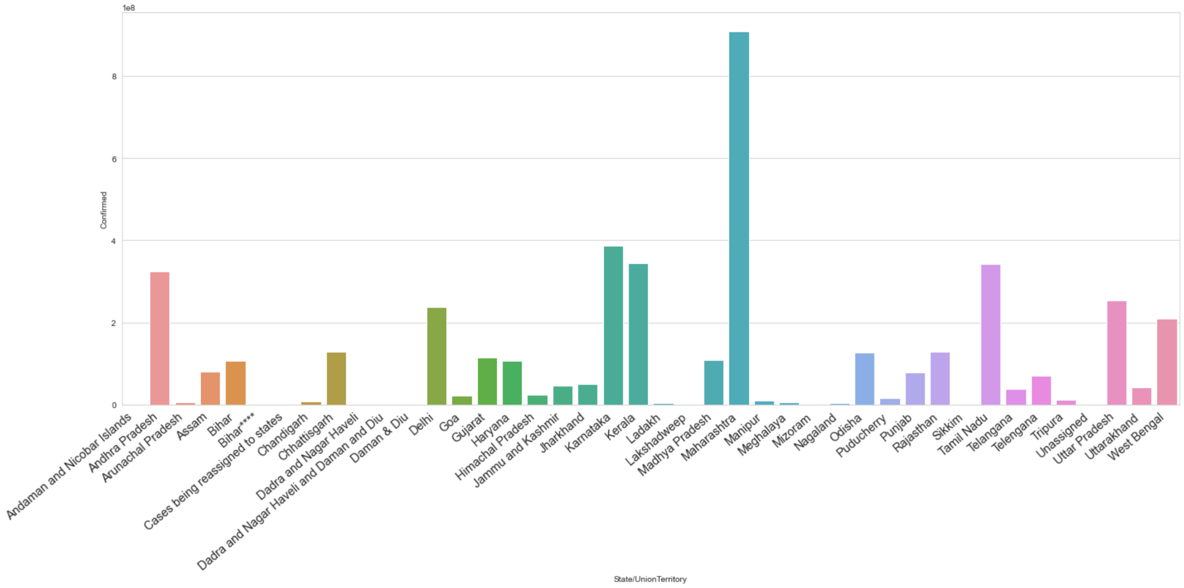
In [30]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.lineplot(x="State/UnionTerritory",y="Active_cases",data=statewise)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [33]:

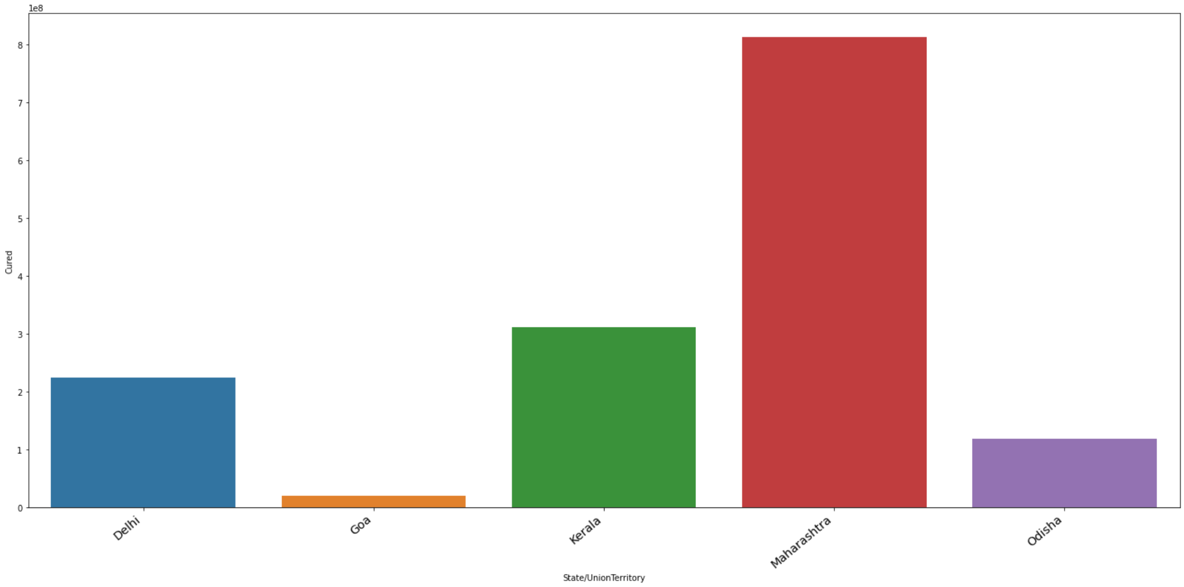
```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.barplot(x="State/UnionTerritory",y="Confirmed",data=statewise)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



state and cured/deaths

In [56]:

```
fig=plt.figure(figsize=(20,10))
sns.barplot(x="State/UnionTerritory",y="Cured",data=statewise_filter,palette="tab10")
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [34]:

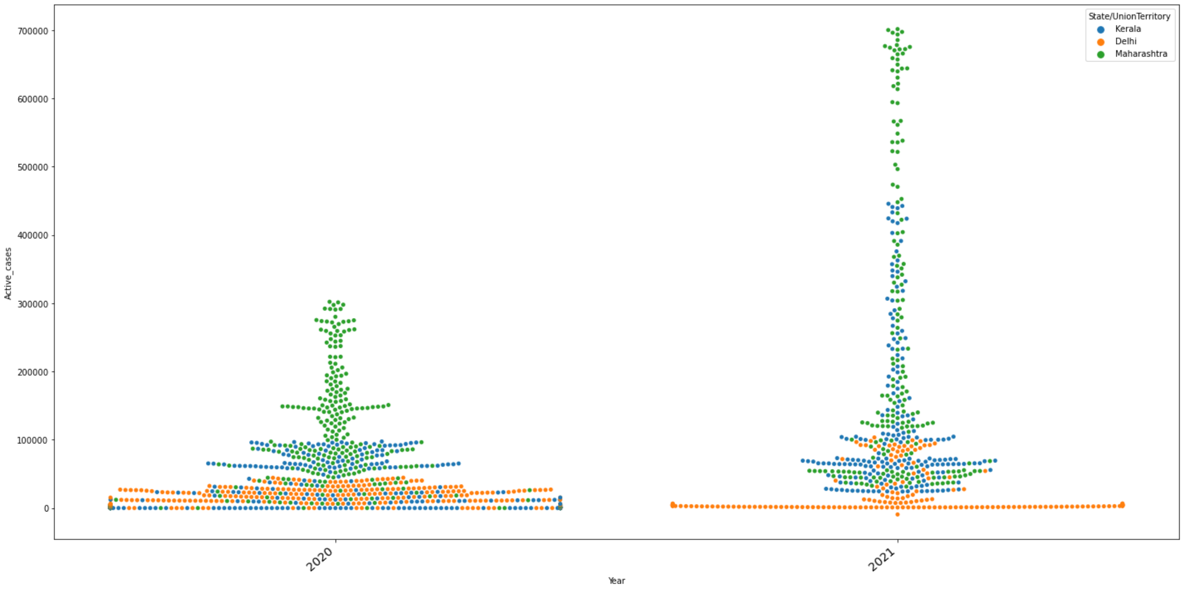
```
#df_filter=df[(df["State/UnionTerritory"].isin(["Maharashtra","Kerala","Delhi","Odisha","Go
```

data=filter state,x=state,y=active,hue=year

In [90]:

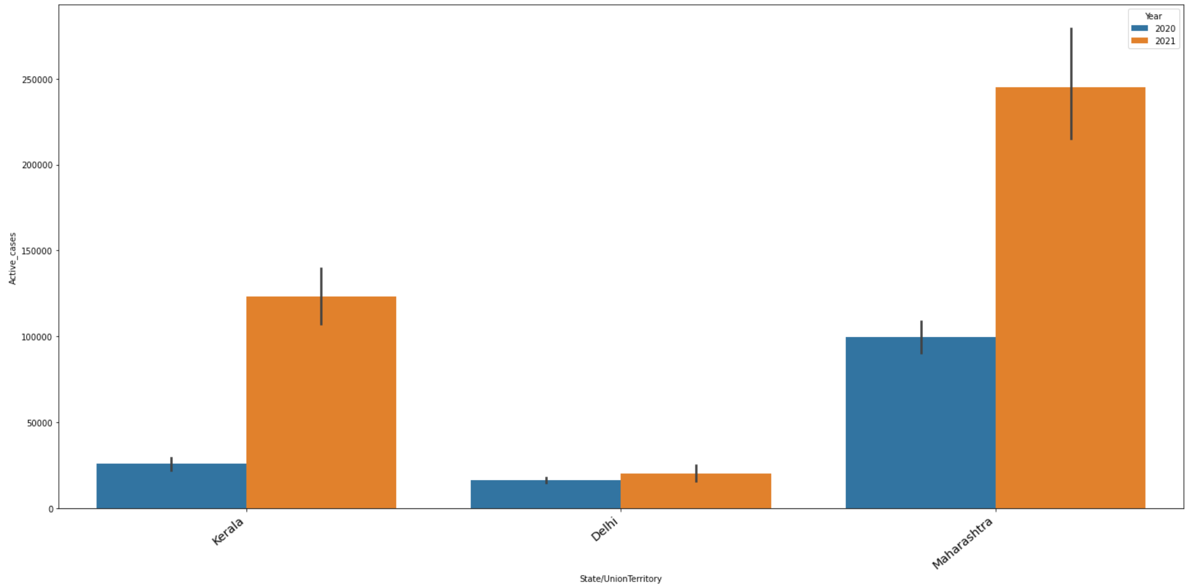
```
fig=plt.figure(figsize=(20,10))
sns.swarmplot(x="Year",y="Active_cases",data=df[df["State/UnionTerritory"].isin(["Maharasht
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

```
C:\Users\Shrutika\anaconda3\lib\site-packages\seaborn\categorical.py:1296: U
serWarning: 23.7% of the points cannot be placed; you may want to decrease t
he size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Shrutika\anaconda3\lib\site-packages\seaborn\categorical.py:1296: U
serWarning: 5.9% of the points cannot be placed; you may want to decrease th
e size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



In [44]:

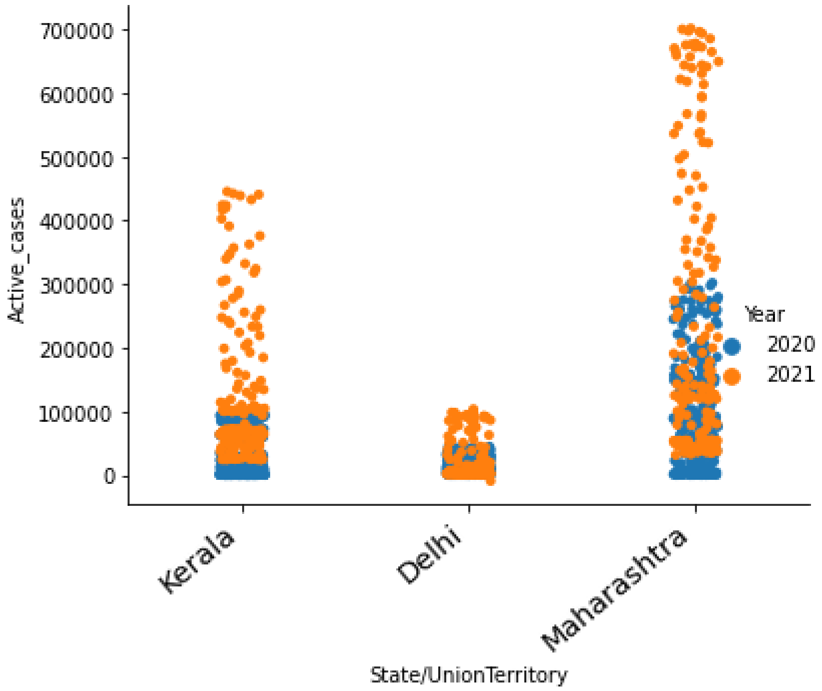
```
fig=plt.figure(figsize=(20,10))
sns.barplot(x="State/UnionTerritory",y="Active_cases",data=df[df["State/UnionTerritory"].is
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [45]:

```
fig=plt.figure(figsize=(20,10))
sns.catplot(x="State/UnionTerritory",y="Active_cases",data=df[df["State/UnionTerritory"].is
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

```
<Figure size 1440x720 with 0 Axes>
```



In [68]:

```
#date_filter=df.groupby("Date").sum().reset_index()
```

In [81]:

```
#date_filter1=date_filter[(date_filter["Date"]=="2020-02-01")|(date_filter["Date"]=="2020-0
(date_filter["Date"]=="2020-09-01")|(date_filter["Date"]=="2020-12-01")]
```

In [95]:

```
statewise2=df.groupby("Date")[["Confirmed","Cured","Deaths","Active_cases"]].sum().reset_in
```

In [96]:

```
statewise2
```

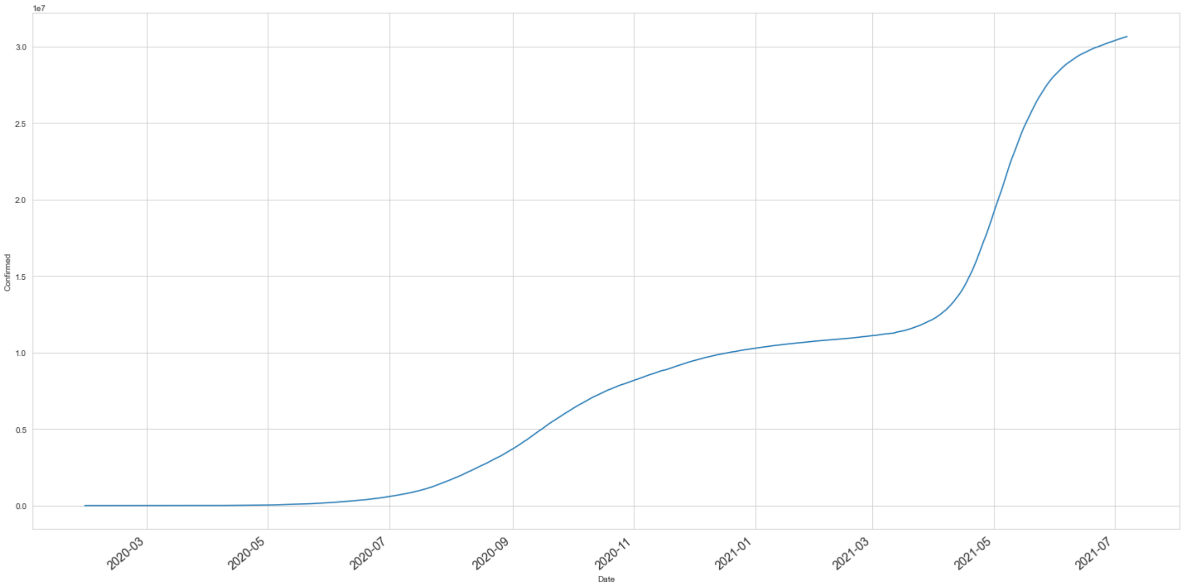
Out[96]:

	Date	Confirmed	Cured	Deaths	Active_cases
0	2020-01-30	1	0	0	1
1	2020-01-31	1	0	0	1
2	2020-02-01	2	0	0	2
3	2020-02-02	3	0	0	3
4	2020-02-03	3	0	0	3
...	...	...	...	...	...
520	2021-07-03	30502362	29605779	401050	495533
521	2021-07-04	30545433	29658078	402005	485350
522	2021-07-05	30585229	29700430	402728	482071
523	2021-07-06	30619932	29752294	403281	464357
524	2021-07-07	30663665	29799534	404211	459920

525 rows × 5 columns

In [97]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.lineplot(x="Date",y="Confirmed",data=statewise2)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



recovery and Fatality rate

In [98]:

```
statewise["Recovery_rate"]=statewise["Cured"]*100/statewise["Confirmed"]
statewise["Fatality_rate"]=statewise["Deaths"]*100/statewise["Confirmed"]
statewise
```

Out[98]:

	State/UnionTerritory	Confirmed	Cured	Deaths	Active_cases	Recovery_rate	Fatality
0	Andaman and Nicobar Islands	1675248	1589935	22624	62689	94.907441	1.35
1	Andhra Pradesh	324146783	303427899	2475816	18243068	93.608178	0.76
2	Arunachal Pradesh	5598324	5150519	19303	428502	92.001088	0.34
3	Assam	80418492	74011348	459575	5947569	92.032748	0.57
4	Bihar	106881540	100131380	756282	5993878	93.684447	0.70
5	Bihar****	1430909	1402468	18881	9560	98.012382	1.31
6	Cases being reassigned to states	345565	0	0	345565	0.000000	0.00
7	Chandigarh	8691806	7980284	119356	592166	91.813876	1.37
8	Chhattisgarh	128751782	117163544	1591126	9997112	90.999551	1.23
9	Dadra and Nagar Haveli	20722	20352	8	362	98.214458	0.03
10	Dadra and Nagar Haveli and Daman and Diu	1566846	1470986	874	94986	93.881977	0.05
11	Daman & Diu	2	0	0	2	0.000000	0.00
12	Delhi	236972842	224062704	4066907	8843231	94.552060	1.71
13	Goa	22280065	20224042	338359	1717664	90.771917	1.51
14	Gujarat	114557615	103995131	1866811	8695673	90.779763	1.62
15	Haryana	107408371	100010131	1166573	6231667	93.112045	1.08
16	Himachal Pradesh	23052151	20682770	371931	1997450	89.721649	1.61
17	Jammu and Kashmir	46899925	42295048	686680	3918197	90.181483	1.46
18	Jharkhand	49971564	46083978	569298	3318288	92.220404	1.13
19	Karnataka	387597335	345648926	4819018	37129391	89.177323	1.24
20	Kerala	344319045	311127643	1327754	31863648	90.360277	0.38
21	Ladakh	3344131	3059045	38578	246508	91.475035	1.15
22	Lakshadweep	561459	471712	2178	87569	84.015396	0.38
23	Madhya Pradesh	108712983	100169697	1427780	7115506	92.141430	1.31
24	Maharashtra	908892470	813788907	19314532	75789031	89.536324	2.12
25	Manipur	9440912	8420223	122089	898600	89.188661	1.29
26	Meghalaya	5221064	4606548	66293	548223	88.230062	1.26
27	Mizoram	1822190	1534630	5073	282487	84.218989	0.27
28	Nagaland	4089547	3628619	39420	421508	88.729118	0.96
29	Odisha	126408397	117984789	600149	7823459	93.336196	0.47
30	Puducherry	15858688	14376916	249683	1232089	90.656402	1.57

	State/UnionTerritory	Confirmed	Cured	Deaths	Active_cases	Recovery_rate	Fatality
31	Punjab	78999515	71108712	2216735	5674068	90.011580	2.80
32	Rajasthan	128998101	117312772	1159823	10525506	90.941472	0.89
33	Sikkim	2315519	1983899	41530	290090	85.678373	1.79
34	Tamil Nadu	342829697	317067499	4731627	21030571	92.485424	1.38
35	Telangana	38162058	35544978	217455	2399625	93.142194	0.56
36	Telengana	69990668	64666267	400427	4923974	92.392699	0.57
37	Tripura	11397656	10479169	124444	794043	91.941440	1.09
38	Unassigned	161	0	0	161	0.000000	0.00
39	Uttar Pradesh	252843682	232529439	3347656	16966587	91.965691	1.32
40	Uttarakhand	41179396	36684388	728512	3766496	89.084328	1.76
41	West Bengal	209822848	195296839	3214840	11311169	93.077013	1.53

In [97]:

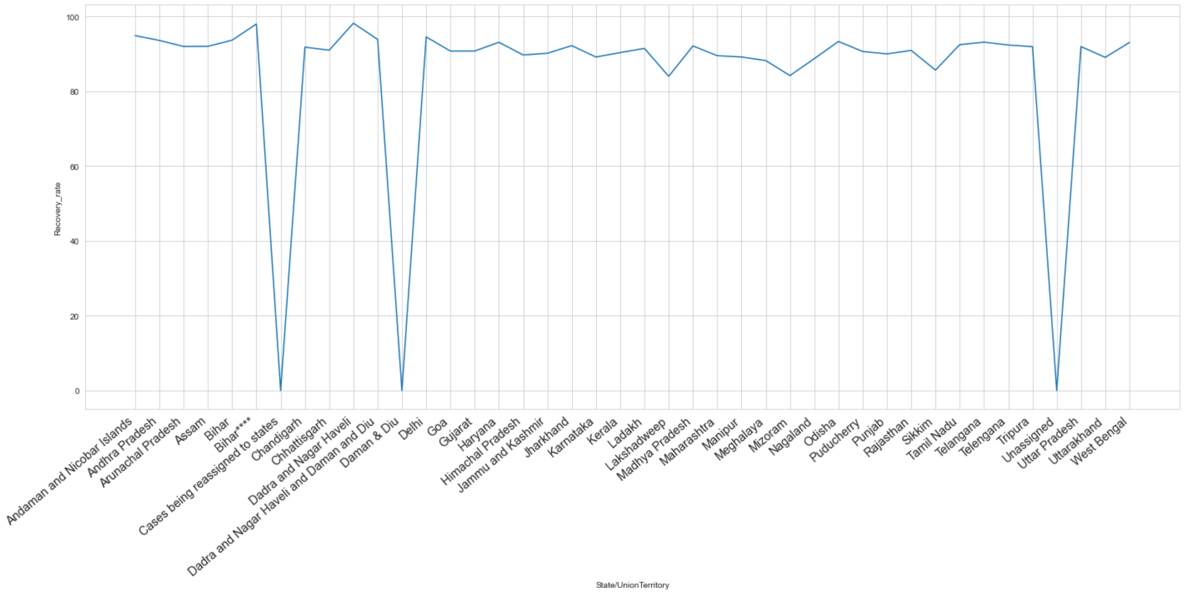
```
#statewise.style.background_gradient(cmap="cubehelix")#color map in matplotlib
```

In [85]:

```
#statewise.style.background_gradient(cmap="Reds")
```

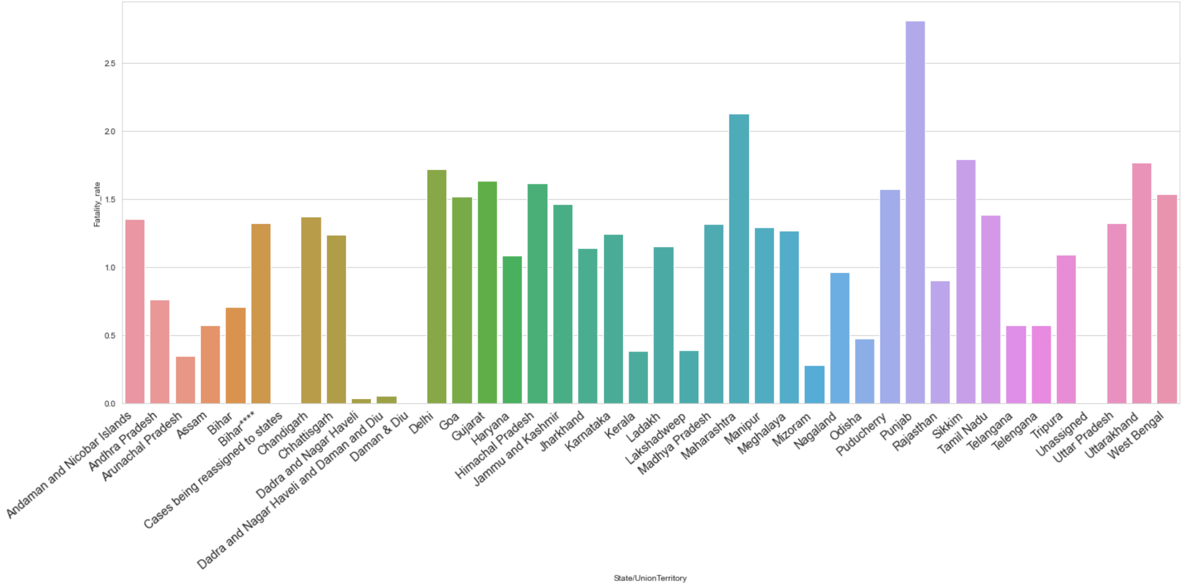
In [106]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.lineplot(x="State/UnionTerritory",y="Recovery_rate",data=statewise)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [107]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.barplot(x="State/UnionTerritory",y="Fatality_rate",data=statewise)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



percentage, of cured,deaths and confirmed

In [29]:

```
percentage=df.groupby("State/UnionTerritory")[["Confirmed","Cured","Deaths"]].sum()
percentage["Cured_percentage"]=(percentage["Cured"]/percentage["Cured"].sum())*100
percentage["Deaths_percentage"]=(percentage["Deaths"]/percentage["Deaths"].sum())*100
percentage["Confirmed_percentage"]=(percentage["Confirmed"]/percentage["Confirmed"].sum())*
percentage
```

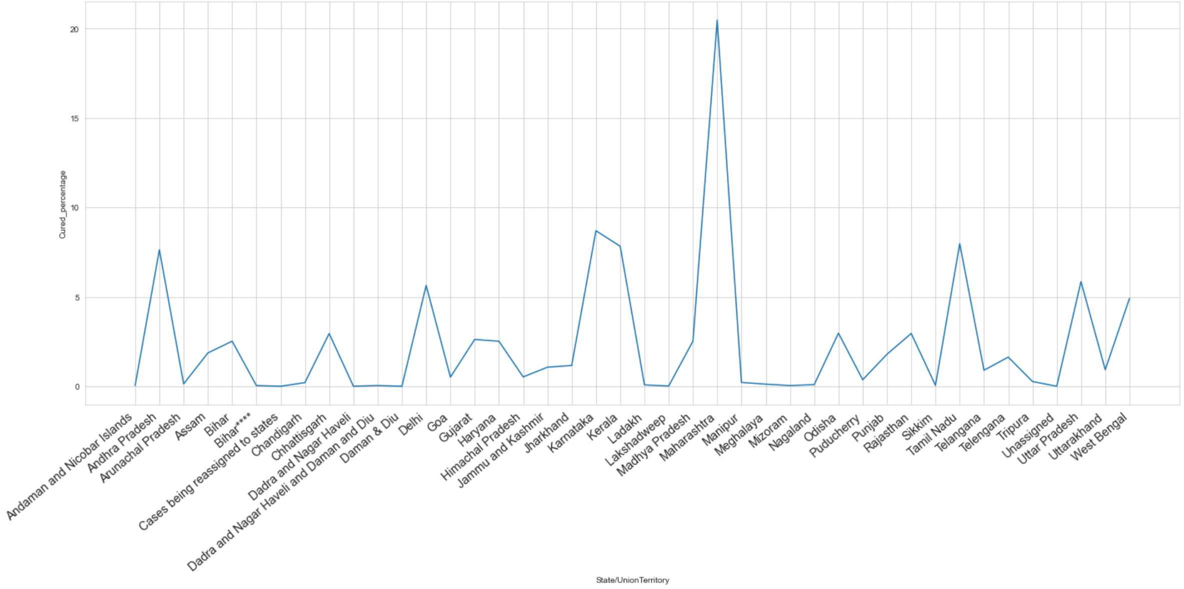
Out[29]:

	Confirmed	Cured	Deaths	Cured_percentage	Deaths_percentage
State/UnionTerritory					
Andaman and Nicobar Islands	1675248	1589935	22624	0.039976	0.038525
Andhra Pradesh	324146783	303427899	2475816	7.629195	4.215877
Arunachal Pradesh	5598324	5150519	19303	0.129501	0.032870
Assam	80418492	74011348	459575	1.860894	0.782575
Bihar	106881540	100131380	756282	2.517639	1.287815
Bihar****	1430909	1402468	18881	0.035263	0.032151
Cases being reassigned to states	345565	0	0	0.000000	0.000000
Chandigarh	8691806	7980284	119356	0.200651	0.203242
Chhattisgarh	128751782	117163544	1591126	2.945884	2.709406
Dadra and Nagar Haveli	20722	20352	8	0.000512	0.000014
Dadra and Nagar Haveli and Daman and Diu	1566846	1470986	874	0.036986	0.001488
Daman & Diu	2	0	0	0.000000	0.000000
Delhi	236972842	224062704	4066907	5.633688	6.925224
Goa	22280065	20224042	338359	0.508500	0.576166
Gujarat	114557615	103995131	1866811	2.614786	3.178849
Haryana	107408371	100010131	1166573	2.514590	1.986468
Himachal Pradesh	23052151	20682770	371931	0.520034	0.633333
Jammu and Kashmir	46899925	42295048	686680	1.063439	1.169295
Jharkhand	49971564	46083978	569298	1.158706	0.969414
Karnataka	387597335	345648926	4819018	8.690773	8.205936
Kerala	344319045	311127643	1327754	7.822792	2.260930
Ladakh	3344131	3059045	38578	0.076915	0.065692
Lakshadweep	561459	471712	2178	0.011860	0.003709
Madhya Pradesh	108712983	100169697	1427780	2.518602	2.431257
Maharashtra	908892470	813788907	19314532	20.461383	32.889235
Manipur	9440912	8420223	122089	0.211713	0.207896
Meghalaya	5221064	4606548	66293	0.115824	0.112885

	Confirmed	Cured	Deaths	Cured_percentage	Deaths_percentage
State/UnionTerritory					
Mizoram	1822190	1534630	5073	0.038586	0.008638
Nagaland	4089547	3628619	39420	0.091236	0.067125
Odisha	126408397	117984789	600149	2.966533	1.021948
Puducherry	15858688	14376916	249683	0.361484	0.425166
Punjab	78999515	71108712	2216735	1.787912	3.774708
Rajasthan	128998101	117312772	1159823	2.949637	1.974974
Sikkim	2315519	1983899	41530	0.049882	0.070718
Tamil Nadu	342829697	317067499	4731627	7.972140	8.057125
Telangana	38162058	35544978	217455	0.893720	0.370287
Telengana	69990668	64666267	400427	1.625927	0.681856
Tripura	11397656	10479169	124444	0.263481	0.211906
Unassigned	161	0	0	0.000000	0.000000
Uttar Pradesh	252843682	232529439	3347656	5.846570	5.700467
Uttarakhand	41179396	36684388	728512	0.922369	1.240527
West Bengal	209822848	195296839	3214840	4.910418	5.474304

In [175]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.lineplot(x="State/UnionTerritory",y="Cured_percentage",data=percentage)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



population

In [108]:

```
popln=pd.Series({"Population":[380581,49386799,1383727,31205576,104099452,0,0,1055450,25545
                      1458545,60439692,25351462,6864602,12541302,32988134,61095297,33406061,0,
                      1097206,1978502,41974218,1247953,27743338,68548437,610577,72147030,35193
                      91276115]})
```

In [118]:

```
population_wise=df.groupby("State/UnionTerritory")[["Cured","Deaths","Confirmed","Active_ca
```

In [119]:

```
population_wise
```

Out[119]:

	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases
0	Andaman and Nicobar Islands	1589935	22624	1675248	62689
1	Andhra Pradesh	303427899	2475816	324146783	18243068
2	Arunachal Pradesh	5150519	19303	5598324	428502
3	Assam	74011348	459575	80418492	5947569
4	Bihar	100131380	756282	106881540	5993878
5	Bihar****	1402468	18881	1430909	9560
6	Cases being reassigned to states	0	0	345565	345565
7	Chandigarh	7980284	119356	8691806	592166
8	Chhattisgarh	117163544	1591126	128751782	9997112
9	Dadra and Nagar Haveli	20352	8	20722	362
10	Dadra and Nagar Haveli and Daman and Diu	1470986	874	1566846	94986
11	Daman & Diu	0	0	2	2
12	Delhi	224062704	4066907	236972842	8843231
13	Goa	20224042	338359	22280065	1717664
14	Gujarat	103995131	1866811	114557615	8695673
15	Haryana	100010131	1166573	107408371	6231667
16	Himachal Pradesh	20682770	371931	23052151	1997450
17	Jammu and Kashmir	42295048	686680	46899925	3918197
18	Jharkhand	46083978	569298	49971564	3318288
19	Karnataka	345648926	4819018	387597335	37129391
20	Kerala	311127643	1327754	344319045	31863648
21	Ladakh	3059045	38578	3344131	246508
22	Lakshadweep	471712	2178	561459	87569
23	Madhya Pradesh	100169697	1427780	108712983	7115506
24	Maharashtra	813788907	19314532	908892470	75789031
25	Manipur	8420223	122089	9440912	898600
26	Meghalaya	4606548	66293	5221064	548223
27	Mizoram	1534630	5073	1822190	282487
28	Nagaland	3628619	39420	4089547	421508
29	Odisha	117984789	600149	126408397	7823459
30	Puducherry	14376916	249683	15858688	1232089
31	Punjab	71108712	2216735	78999515	5674068
32	Rajasthan	117312772	1159823	128998101	10525506
33	Sikkim	1983899	41530	2315519	290090
34	Tamil Nadu	317067499	4731627	342829697	21030571

	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases
35	Telangana	35544978	217455	38162058	2399625
36	Telengana	64666267	400427	69990668	4923974
37	Tripura	10479169	124444	11397656	794043
38	Unassigned	0	0	161	161
39	Uttar Pradesh	232529439	3347656	252843682	16966587
40	Uttarakhand	36684388	728512	41179396	3766496
41	West Bengal	195296839	3214840	209822848	11311169

In [120]:

```
population_wise["Population"]=popln["Population"]
```

In [121]:

```
population_wise
```

Out[121]:

	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases	Population
0	Andaman and Nicobar Islands	1589935	22624	1675248	62689	380581
1	Andhra Pradesh	303427899	2475816	324146783	18243068	49386799
2	Arunachal Pradesh	5150519	19303	5598324	428502	1383727
3	Assam	74011348	459575	80418492	5947569	31205576
4	Bihar	100131380	756282	106881540	5993878	104099452
5	Bihar****	1402468	18881	1430909	9560	0
6	Cases being reassigned to states	0	0	345565	345565	0
7	Chandigarh	7980284	119356	8691806	592166	1055450
8	Chhattisgarh	117163544	1591126	128751782	9997112	25545198
9	Dadra and Nagar Haveli	20352	8	20722	362	0
10	Dadra and Nagar Haveli and Daman and Diu	1470986	874	1566846	94986	586956
11	Daman & Diu	0	0	2	2	0
12	Delhi	224062704	4066907	236972842	8843231	16787941
13	Goa	20224042	338359	22280065	1717664	1458545
14	Gujarat	103995131	1866811	114557615	8695673	60439692
15	Haryana	100010131	1166573	107408371	6231667	25351462
16	Himachal Pradesh	20682770	371931	23052151	1997450	6864602
17	Jammu and Kashmir	42295048	686680	46899925	3918197	12541302
18	Jharkhand	46083978	569298	49971564	3318288	32988134
19	Karnataka	345648926	4819018	387597335	37129391	61095297
20	Kerala	311127643	1327754	344319045	31863648	33406061
21	Ladakh	3059045	38578	3344131	246508	0
22	Lakshadweep	471712	2178	561459	87569	64473
23	Madhya Pradesh	100169697	1427780	108712983	7115506	72626809
24	Maharashtra	813788907	19314532	908892470	75789031	112374333
25	Manipur	8420223	122089	9440912	898600	2855794
26	Meghalaya	4606548	66293	5221064	548223	2966889
27	Mizoram	1534630	5073	1822190	282487	1097206
28	Nagaland	3628619	39420	4089547	421508	1978502
29	Odisha	117984789	600149	126408397	7823459	41974218
30	Puducherry	14376916	249683	15858688	1232089	1247953
31	Punjab	71108712	2216735	78999515	5674068	27743338
32	Rajasthan	117312772	1159823	128998101	10525506	68548437
33	Sikkim	1983899	41530	2315519	290090	610577

	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases	Population
34	Tamil Nadu	317067499	4731627	342829697	21030571	72147030
35	Telangana	35544978	217455	38162058	2399625	35193978
36	Telengana	64666267	400427	69990668	4923974	0
37	Tripura	10479169	124444	11397656	794043	3673917
38	Unassigned	0	0	161	161	0
39	Uttar Pradesh	232529439	3347656	252843682	16966587	199812341
40	Uttarakhand	36684388	728512	41179396	3766496	10086292
41	West Bengal	195296839	3214840	209822848	11311169	91276115

In [108]:

```
statewise1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 42 entries, Andaman and Nicobar Islands to West Bengal
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Cured             42 non-null     int64
 1   Deaths            42 non-null     int64
 2   Popln_statewise1  42 non-null     int64
dtypes: int64(3)
memory usage: 1.3+ KB
```

In [82]:

```
#statewise2=df.groupby("State/UnionTerritory")[["Cured"]].sum()#remember this very imp
```

In [95]:

```
#statewise2
```

mortality rate, prevalance rate

In [122]:

```
population_wise["Mortality_rate"]=(population_wise["Deaths"]/population_wise["Population"])
population_wise["Prevalance_rate"]=population_wise["Active_cases"]/population_wise["Populat
```

In [123]:

```
population_wise
```

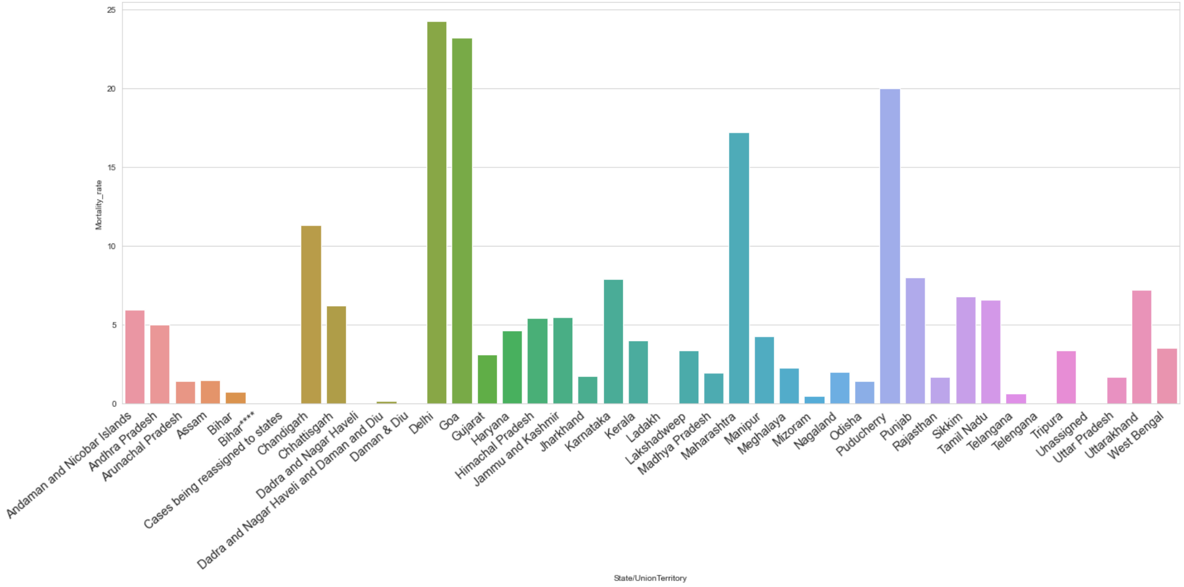
Out[123]:

	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases	Population	Mortality_r
0	Andaman and Nicobar Islands	1589935	22624	1675248	62689	380581	5.9445
1	Andhra Pradesh	303427899	2475816	324146783	18243068	49386799	5.0131
2	Arunachal Pradesh	5150519	19303	5598324	428502	1383727	1.3950
3	Assam	74011348	459575	80418492	5947569	31205576	1.4727
4	Bihar	100131380	756282	106881540	5993878	104099452	0.7265
5	Bihar****	1402468	18881	1430909	9560	0	
6	Cases being reassigned to states	0	0	345565	345565	0	N
7	Chandigarh	7980284	119356	8691806	592166	1055450	11.3085
8	Chhattisgarh	117163544	1591126	128751782	9997112	25545198	6.2286
9	Dadra and Nagar Haveli	20352	8	20722	362	0	
10	Dadra and Nagar Haveli and Daman and Diu	1470986	874	1566846	94986	586956	0.1489
11	Daman & Diu	0	0	2	2	0	N
12	Delhi	224062704	4066907	236972842	8843231	16787941	24.2251
13	Goa	20224042	338359	22280065	1717664	1458545	23.1983
14	Gujarat	103995131	1866811	114557615	8695673	60439692	3.0887
15	Haryana	100010131	1166573	107408371	6231667	25351462	4.6016
16	Himachal Pradesh	20682770	371931	23052151	1997450	6864602	5.4181
17	Jammu and Kashmir	42295048	686680	46899925	3918197	12541302	5.4753
18	Jharkhand	46083978	569298	49971564	3318288	32988134	1.7257
19	Karnataka	345648926	4819018	387597335	37129391	61095297	7.8877
20	Kerala	311127643	1327754	344319045	31863648	33406061	3.9745
21	Ladakh	3059045	38578	3344131	246508	0	
22	Lakshadweep	471712	2178	561459	87569	64473	3.3781
23	Madhya Pradesh	100169697	1427780	108712983	7115506	72626809	1.9659
24	Maharashtra	813788907	19314532	908892470	75789031	112374333	17.1876
25	Manipur	8420223	122089	9440912	898600	2855794	4.2751
26	Meghalaya	4606548	66293	5221064	548223	2966889	2.2344
27	Mizoram	1534630	5073	1822190	282487	1097206	0.4623
28	Nagaland	3628619	39420	4089547	421508	1978502	1.9924
29	Odisha	117984789	600149	126408397	7823459	41974218	1.4298
30	Puducherry	14376916	249683	15858688	1232089	1247953	20.0074
31	Punjab	71108712	2216735	78999515	5674068	27743338	7.9901

	State/UnionTerritory	Cured	Deaths	Confirmed	Active_cases	Population	Mortality_ra
32	Rajasthan	117312772	1159823	128998101	10525506	68548437	1.6919
33	Sikkim	1983899	41530	2315519	290090	610577	6.8017
34	Tamil Nadu	317067499	4731627	342829697	21030571	72147030	6.5583
35	Telangana	35544978	217455	38162058	2399625	35193978	0.6178
36	Telengana	64666267	400427	69990668	4923974	0	
37	Tripura	10479169	124444	11397656	794043	3673917	3.3872
38	Unassigned	0	0	161	161	0	N
39	Uttar Pradesh	232529439	3347656	252843682	16966587	199812341	1.6754
40	Uttarakhand	36684388	728512	41179396	3766496	10086292	7.2227
41	West Bengal	195296839	3214840	209822848	11311169	91276115	3.5221

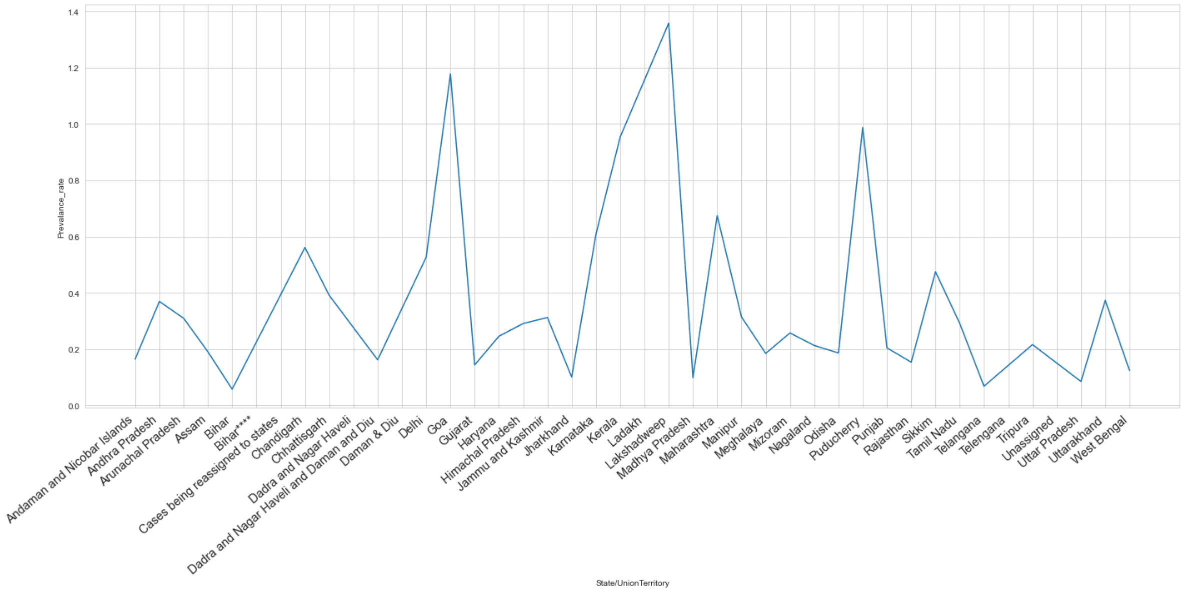
In [125]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.barplot(x="State/UnionTerritory",y="Mortality_rate",data=population_wise)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [127]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.lineplot(x="State/UnionTerritory",y="Prevalance_rate",data=population_wise)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



confirmed cases for per million ppl for year 2021

In [148]:

```
date_wise=df.groupby("Date")[["Cured","Deaths","Confirmed","Active_cases"]].sum()
```

In [169]:

```
confirm=df[["Date","Confirmed"]][(df["Date"]>="2021-01-01") & (df["Date"]<="2021-12-31")]
```

In [170]:

```
confirm["Confirmed_per_million"]=confirm["Confirmed"]/1000000
```

In [171]:

```
confirm
```

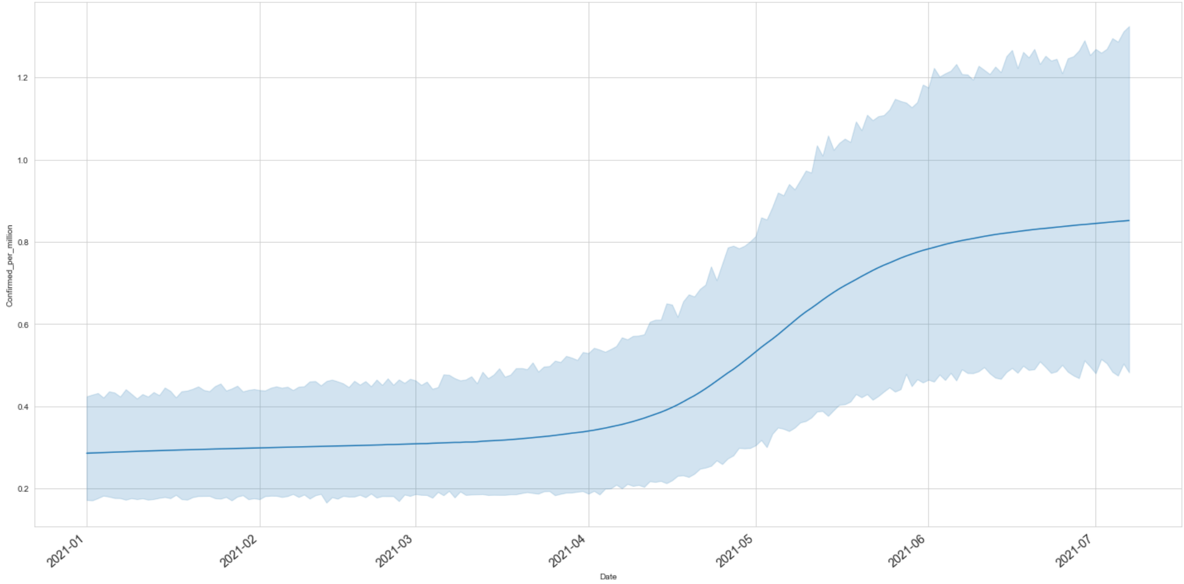
Out[171]:

	Date	Confirmed	Confirmed_per_million
10082	2021-01-01	882286	0.882286
10083	2021-01-01	4945	0.004945
10084	2021-01-01	16719	0.016719
10085	2021-01-01	216211	0.216211
10086	2021-01-01	251743	0.251743
...	...	...	...
16845	2021-07-07	628282	0.628282
16846	2021-07-07	68612	0.068612
16847	2021-07-07	340882	0.340882
16848	2021-07-07	1706818	1.706818
16849	2021-07-07	1507241	1.507241

6768 rows × 3 columns

In [174]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")#whitegrid
sns.lineplot(x="Date",y="Confirmed_per_million",data=confirm)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [ ]:

```
df1[["Cured","Deaths","Confirmed"]][(df1["Date"]>="2020-01-30") & (df1["Date"]<="2020-04-04
```

In [ ]:

```
df1[["State/UnionTerritory","Cured","Deaths","Confirmed"]][(df1["State/UnionTerritory"]=="K
                (df1["State/UnionTerritory"]=="West Bengal") | (df1["State/UnionTerritory"]=
```

In [ ]:

```
df1[(df1["Cured"]>df1["Deaths"])
```

In [ ]:

```
(df1["Deaths"]>df1["Cured"])]
```

In [ ]:

```
group_state4[["Cured","Deaths","Confirmed"]].agg([np.sum, np.mean, np.min,np.max])
```

highest confirmed states and lowest confirmed cases.

In [6]:

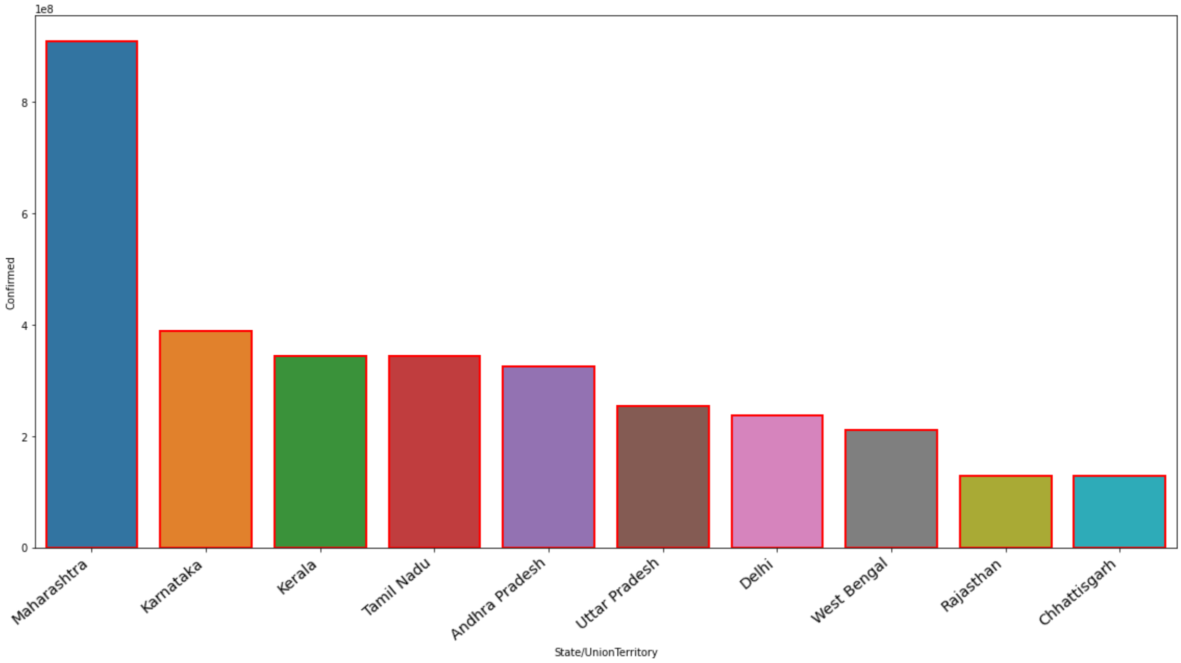
```
#top 5 states with high cured covid cases:
highest=df.groupby("State/UnionTerritory")[["Cured","Deaths","Confirmed"]].sum().sort_value
highest
```

Out[6]:

	State/UnionTerritory	Cured	Deaths	Confirmed
0	Maharashtra	813788907	19314532	908892470
1	Karnataka	345648926	4819018	387597335
2	Kerala	311127643	1327754	344319045
3	Tamil Nadu	317067499	4731627	342829697
4	Andhra Pradesh	303427899	2475816	324146783
5	Uttar Pradesh	232529439	3347656	252843682
6	Delhi	224062704	4066907	236972842
7	West Bengal	195296839	3214840	209822848
8	Rajasthan	117312772	1159823	128998101
9	Chhattisgarh	117163544	1591126	128751782

In [7]:

```
fig=plt.figure(figsize=(16,9))
sns.barplot(data=highest,x="State/UnionTerritory",y="Confirmed",linewidth=2,edgecolor="red"
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [10]:

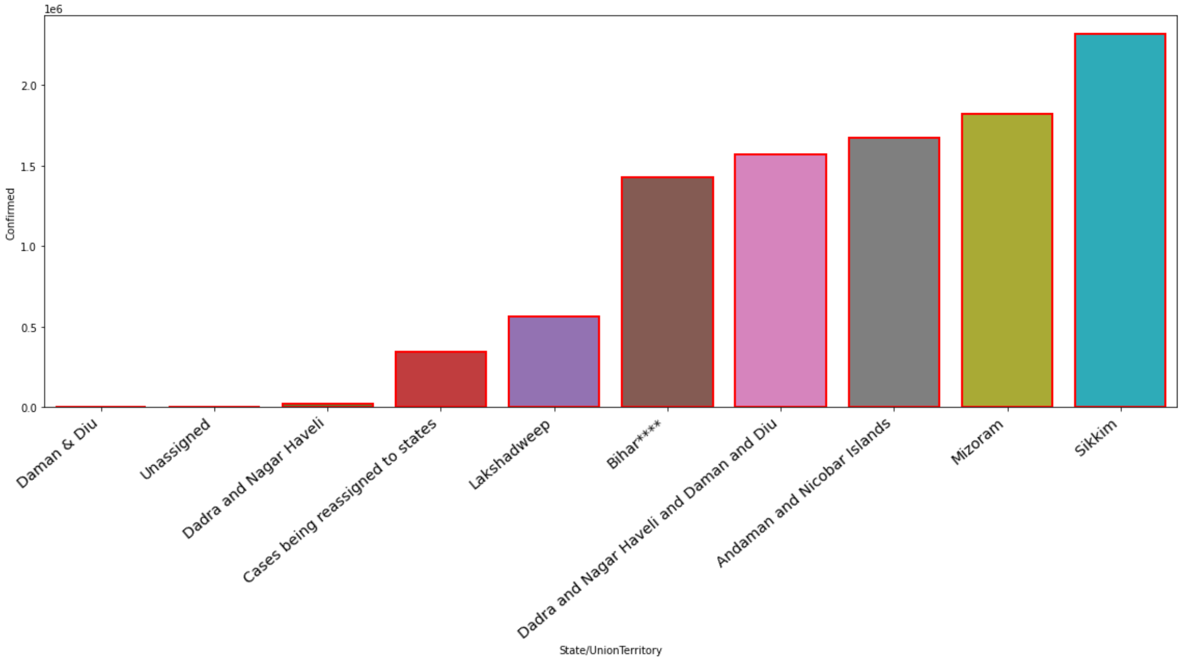
```
#top 5 states with low cured covid cases:
lowest=df.groupby("State/UnionTerritory")[["Cured","Deaths","Confirmed"]].sum().sort_values
lowest
```

Out[10]:

	State/UnionTerritory	Cured	Deaths	Confirmed
0	Daman & Diu	0	0	2
1	Unassigned	0	0	161
2	Dadra and Nagar Haveli	20352	8	20722
3	Cases being reassigned to states	0	0	345565
4	Lakshadweep	471712	2178	561459
5	Bihar****	1402468	18881	1430909
6	Dadra and Nagar Haveli and Daman and Diu	1470986	874	1566846
7	Andaman and Nicobar Islands	1589935	22624	1675248
8	Mizoram	1534630	5073	1822190
9	Sikkim	1983899	41530	2315519

In [11]:

```
fig=plt.figure(figsize=(16,9))
sns.barplot(data=lowest,x="State/UnionTerritory",y="Confirmed",linewidth=2,edgecolor="red")
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [181]:

```
highest_date=df.groupby("Date")[["Cured","Deaths","Confirmed"]].sum().sort_values(by="Confi
highest_date
```

Out[181]:

	Cured	Deaths	Confirmed
Date			
2021-07-07	29799534	404211	30663665
2021-07-06	29752294	403281	30619932
2021-07-05	29700430	402728	30585229
2021-07-04	29658078	402005	30545433
2021-07-03	29605779	401050	30502362
2021-07-02	29548302	400312	30458251
2021-07-01	29488918	399459	30411634
2021-06-30	29427330	398454	30362848
2021-06-29	29366601	397637	30316897
2021-06-28	29309607	396730	30279331

Growth rate

In [69]:

```
growth_rate=df[["Date","State/UnionTerritory","Active_cases"]][(df["State/UnionTerritory"]=
```

In [70]:

```
growth_rate
```

Out[70]:

	Date	State/UnionTerritory	Active_cases
0	2020-01-30	Kerala	1
1	2020-01-31	Kerala	1
2	2020-02-01	Kerala	2
3	2020-02-02	Kerala	3
4	2020-02-03	Kerala	3
...	...	...	...
16794	2021-07-06	Kerala	101097
16798	2021-07-06	Maharashtra	120061
16829	2021-07-07	Karnataka	40039
16830	2021-07-07	Kerala	104577
16834	2021-07-07	Maharashtra	117536

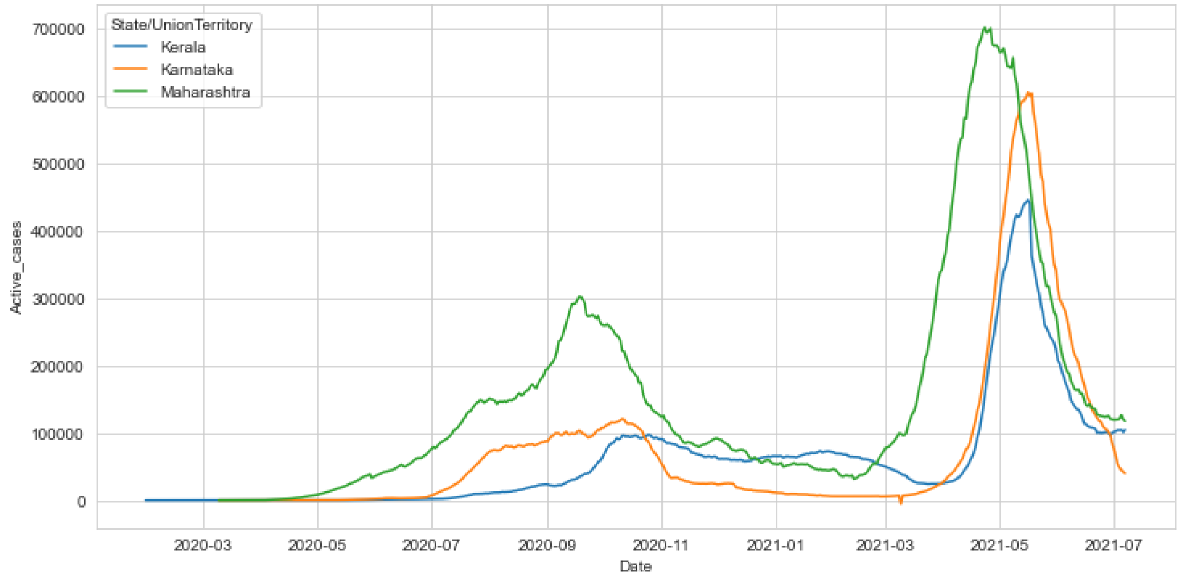
1497 rows × 3 columns

In [71]:

```
fig=plt.figure(figsize=(12,6))
sns.lineplot(data=growth_rate,x="Date",y="Active_cases",hue="State/UnionTerritory")
```

Out[71]:

```
<AxesSubplot:xlabel='Date', ylabel='Active_cases'>
```



In [22]:

```
df.groupby("Date")[["Confirmed","Cured","Deaths","State/UnionTerritory"]].max()
```

Out[22]:

```
            Confirmed     Cured   Deaths   State/UnionTerritory
      Date
2020-01-30          1         0        0                 Kerala
2020-01-31          1         0        0                 Kerala
2020-02-01          2         0        0                 Kerala
2020-02-02          3         0        0                 Kerala
2020-02-03          3         0        0                 Kerala
       ...        ...       ...      ...                    ...
2021-07-03    6079352   5836920   122353            West Bengal
2021-07-04    6088841   5845315   122724            West Bengal
2021-07-05    6098177   5848693   123030            West Bengal
2021-07-06    6104917   5861720   123136            West Bengal
2021-07-07    6113335   5872268   123531            West Bengal
```

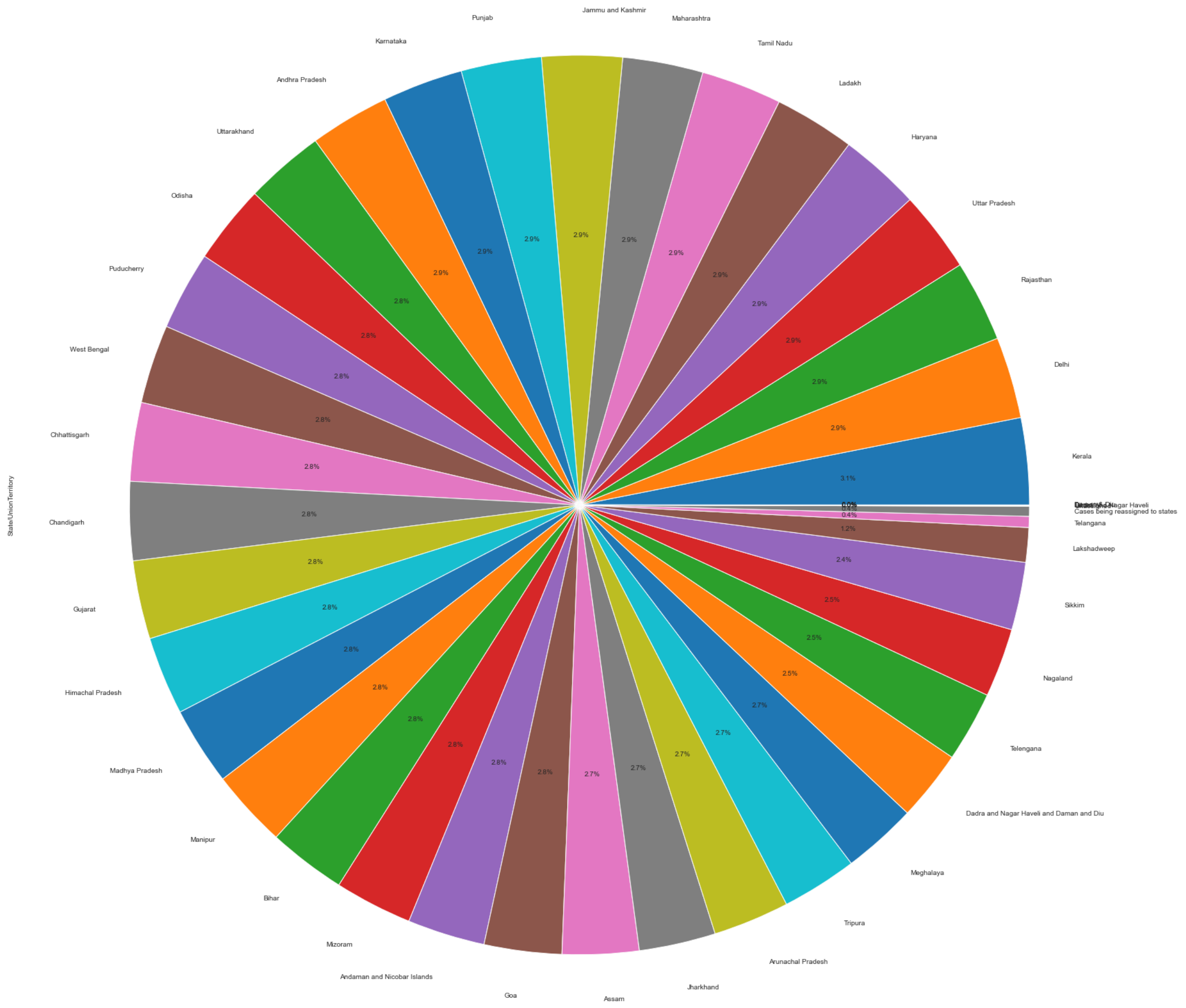
525 rows × 4 columns

In [128]:

```
plt.figure(figsize=(50,30))
df["State/UnionTerritory"].value_counts().plot.pie(autopct="%1.1f%%")
```

Out[128]:

```
<AxesSubplot:ylabel='State/UnionTerritory'>
```



year and month wise data

In [134]:

```
df["Year"]=pd.to_datetime(df["Date"],format="%Y-%m-%d").dt.year
df["Year"]
```

Out[134]:

```
0        2020
1        2020
2        2020
3        2020
4        2020
         ...
16845    2021
16846    2021
16847    2021
16848    2021
16849    2021
Name: Year, Length: 16850, dtype: int64
```

In [132]:

```
df["Month"]=pd.to_datetime(df["Date"],format="%Y-%m-%d").dt.month
```

time wise data

In [210]:

```
df["timing"]=pd.to_datetime(df["Time"]).dt.time
```

In [ ]:

```

```

In [8]:

```
one_year_df=df[(df["Date"]>="2021-01-01") & (df["Date"]<="2021-12-31")]
```

In [ ]:

```
month_year_df=in each year
```

In [ ]:

```
week_df=in each month
```

In [84]:

```
import matplotlib.dates as mdates
```

In [86]:

```
years=mdates.YearLocator()
```

In [87]:

```
years
```

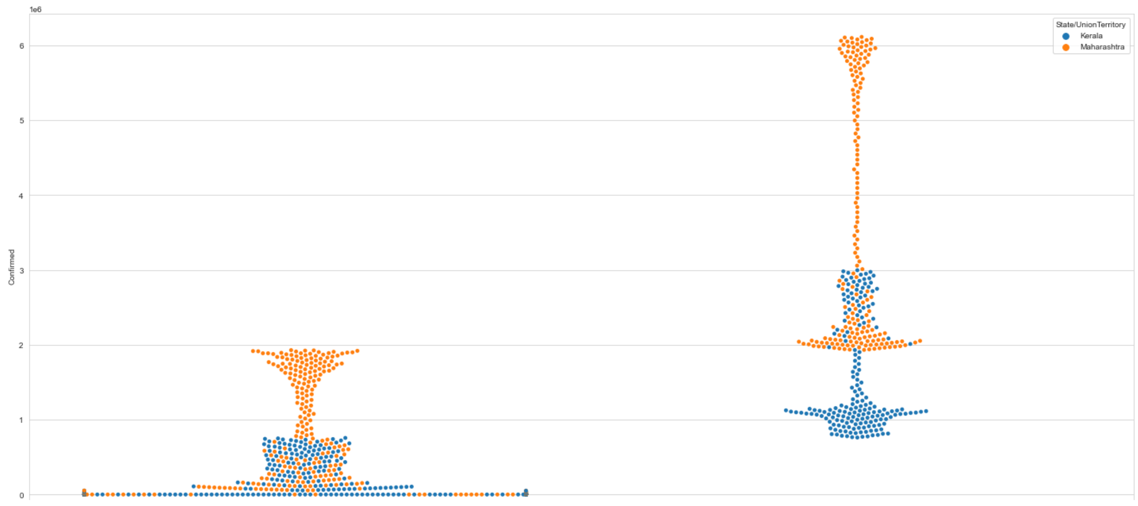
Out[87]:

```
<matplotlib.dates.YearLocator at 0x215ca6b0f10>
```

In [135]:

```
fig=plt.figure(figsize=(20,10))
sns.swarmplot(data=df[df["State/UnionTerritory"].isin(["Maharashtra","Kerala"])],x="Year",y
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

```
C:\Users\Shrutika\anaconda3\lib\site-packages\seaborn\categorical.py:129
6: UserWarning: 31.3% of the points cannot be placed; you may want to dec
rease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



In [ ]:

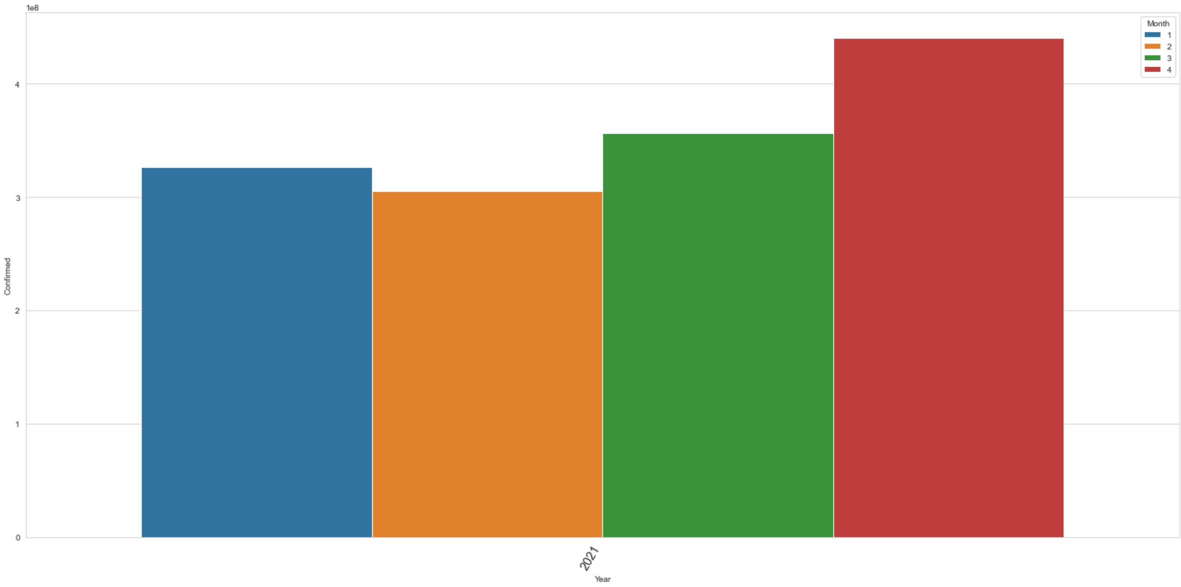
```

```

In [138]:

```
df_year_wise=df.groupby(["Year","Month"])[["Cured","Deaths","Confirmed"]].sum().reset_index
```

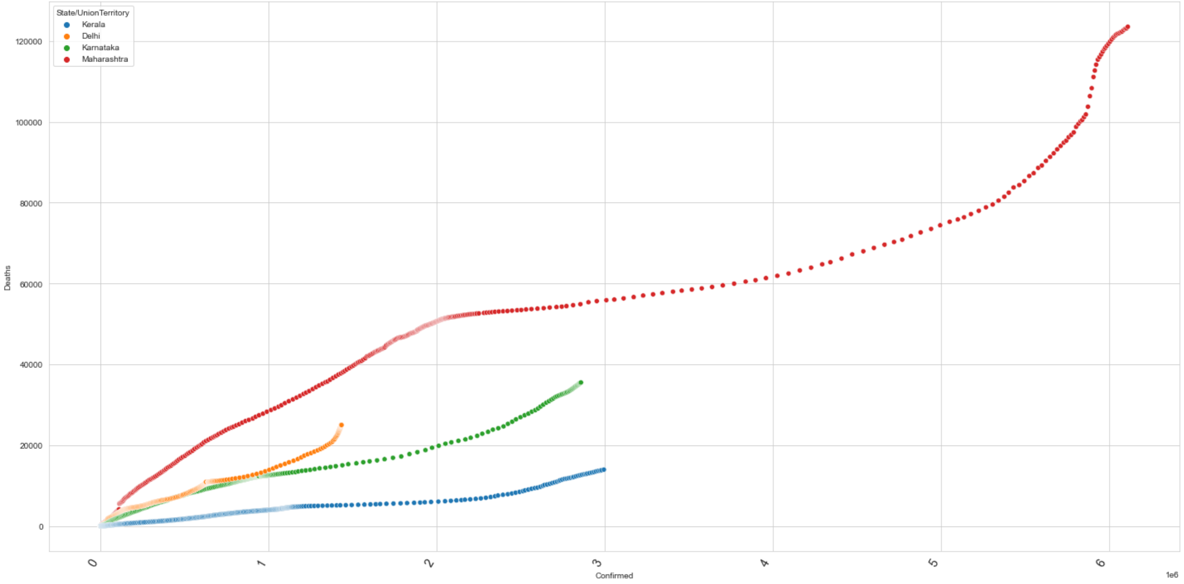
In [140]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.barplot(data=df_year_wise[(df_year_wise["Month"].isin([1,2,3,4])) & (df_year_wise["Year
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



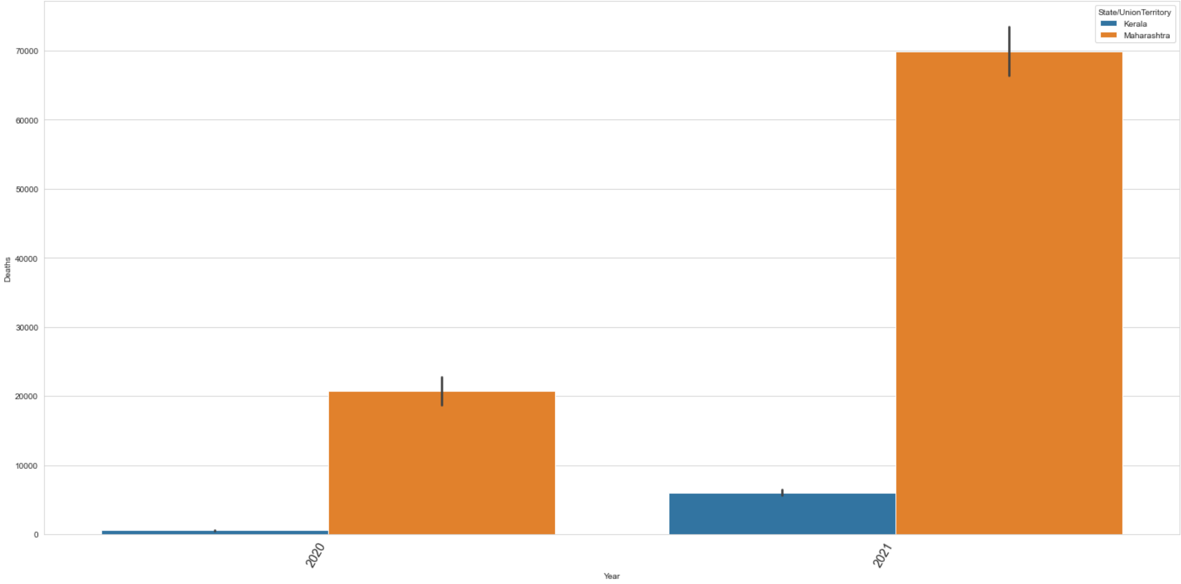
In [145]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.scatterplot(data=df[df["State/UnionTerritory"].isin(["Maharashtra","Kerala","Delhi","Ka
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [147]:

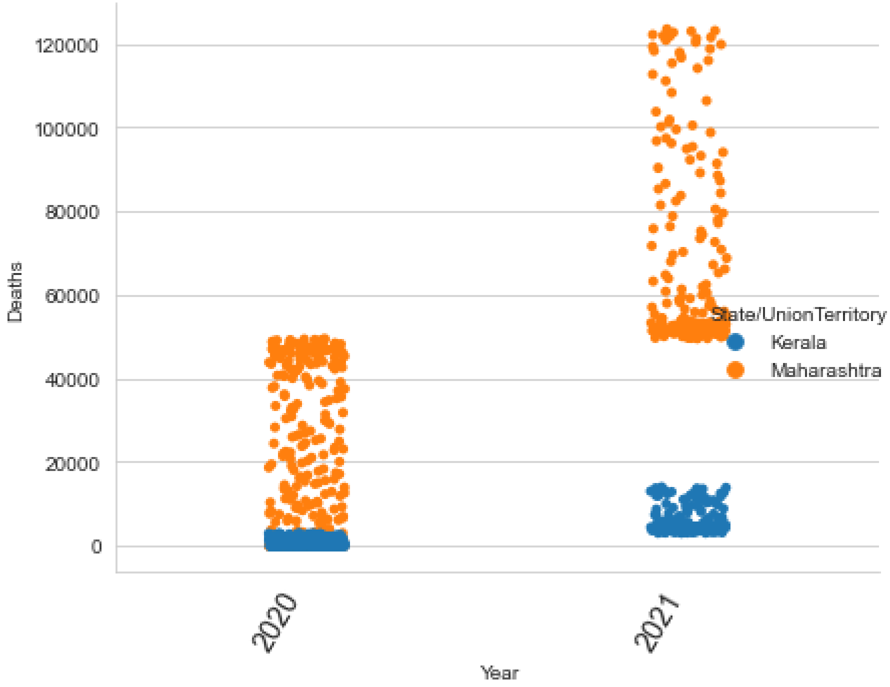
```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.barplot(data=df[df["State/UnionTerritory"].isin(["Maharashtra","Kerala"])],y="Deaths",x
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [148]:

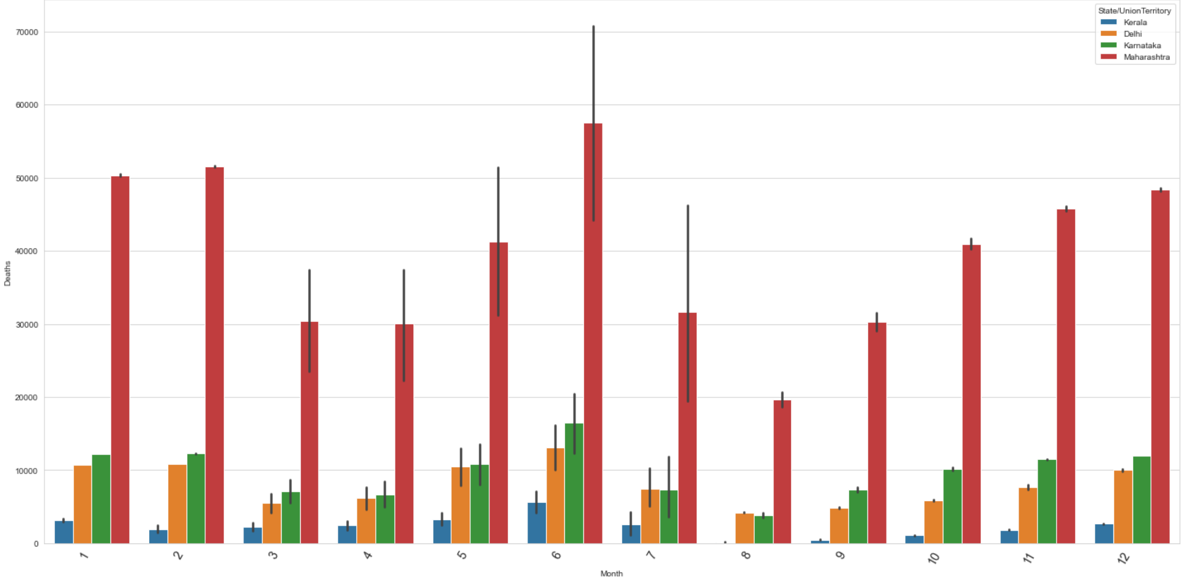
```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.catplot(data=df[df["State/UnionTerritory"].isin(["Maharashtra","Kerala"])],y="Deaths",x
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

```
<Figure size 1440x720 with 0 Axes>
```



In [149]:

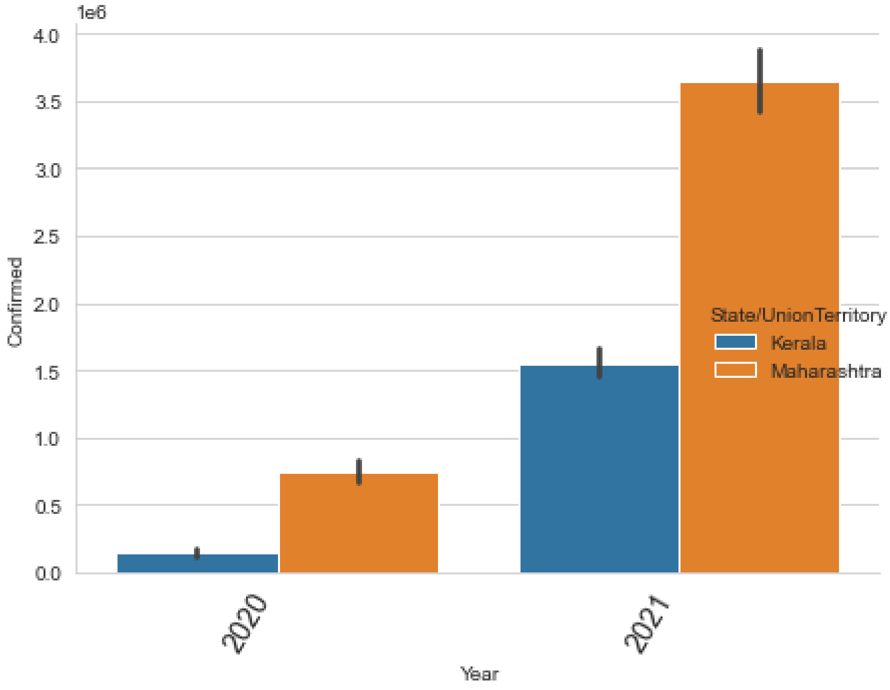
```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.barplot(data=df[df["State/UnionTerritory"].isin(["Maharashtra","Kerala","Delhi","Karnat
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [150]:

```
gsize=(20,10))
le="whitegrid")
df[df["State/UnionTerritory"].isin(["Maharashtra","Kerala"])],y="Confirmed",x="Year",hue="St
on=60,horizontalalignment="right",fontsize="x-large")
)
```

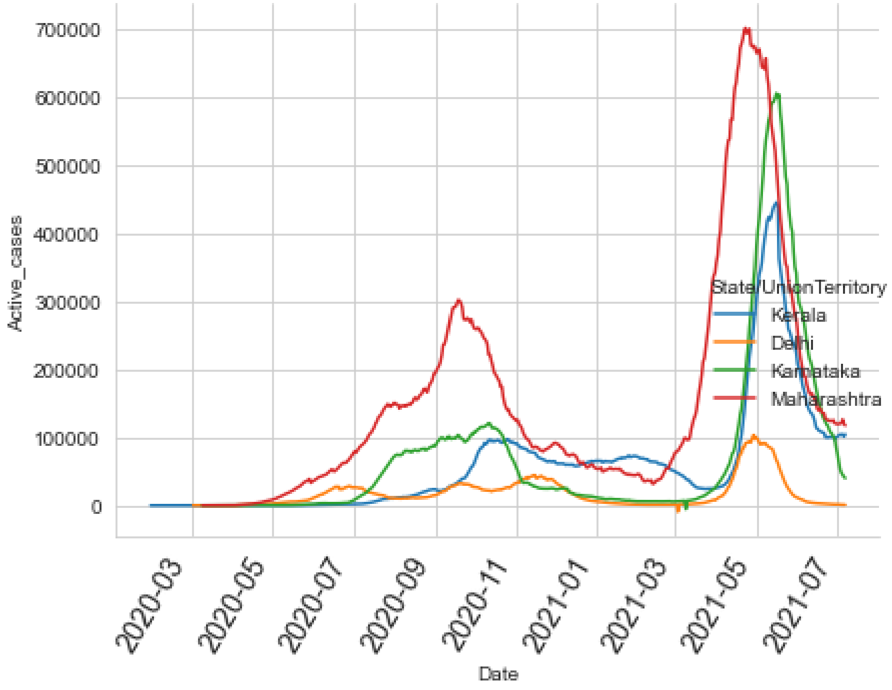
```
<Figure size 1440x720 with 0 Axes>
```



In [151]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.relplot(y="Active_cases",x="Date",hue="State/UnionTerritory",data=df[df["State/UnionTer
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

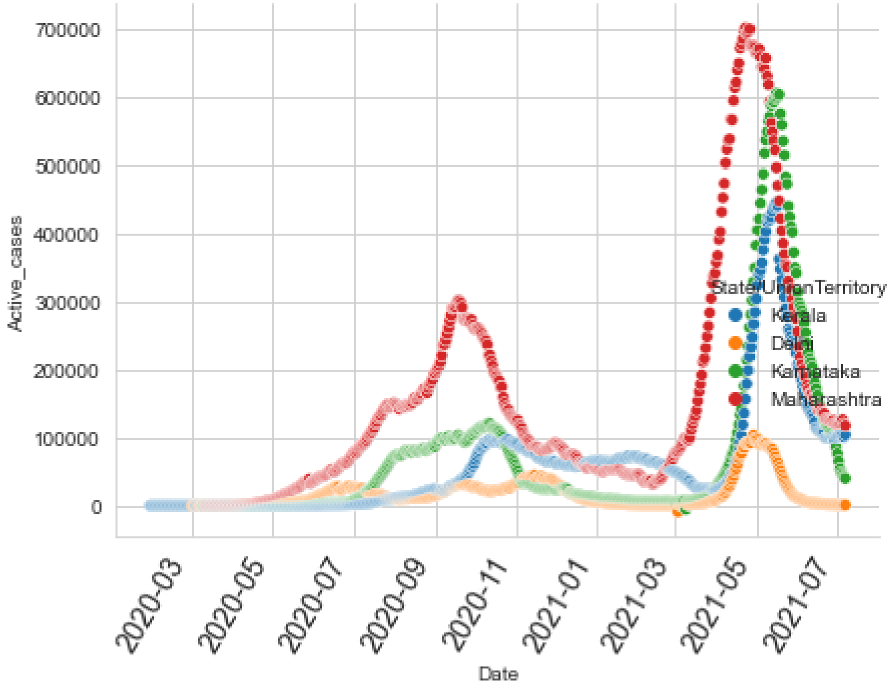
```
<Figure size 1440x720 with 0 Axes>
```



In [153]:

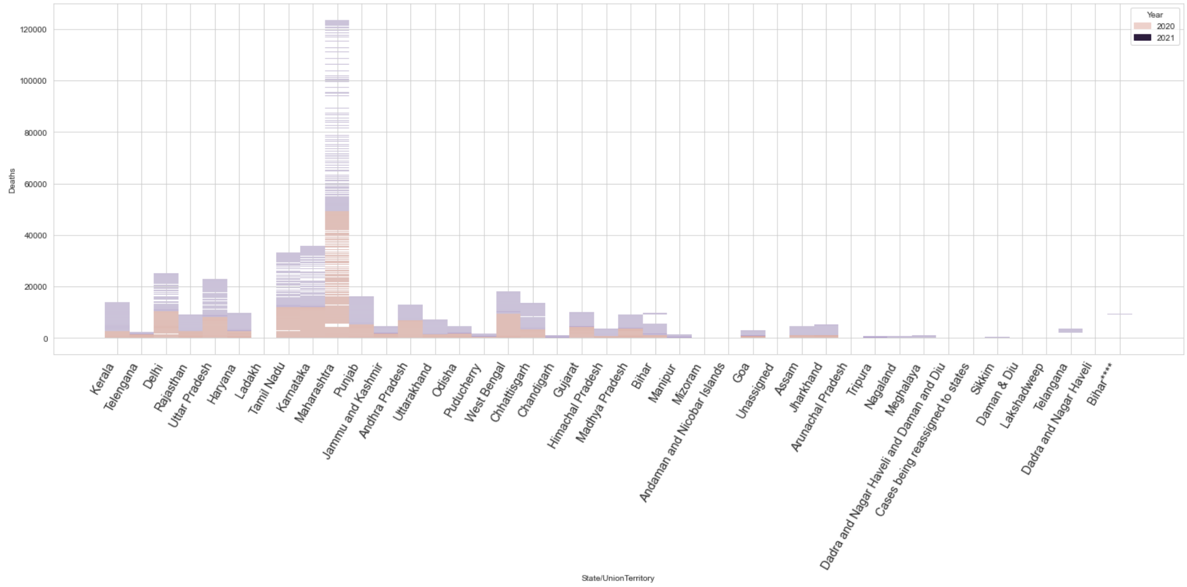
```
fig=plt.figure(figsize=(100,130))
sns.set_style(style="whitegrid")
sns.relplot(y="Active_cases",x="Date",hue="State/UnionTerritory",data=df[df["State/UnionTer
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

```
<Figure size 7200x9360 with 0 Axes>
```



In [154]:

```
fig=plt.figure(figsize=(20,10))
sns.histplot(df,x="State/UnionTerritory",y="Deaths",hue="Year")
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [ ]:

```
fig=plt.figure(figsize=(20,10))
sns.displot(dfs["State/UnionTerritory"])
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

In [ ]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.lineplot(data=dfs,x="Month",y="Cured",hue="State/UnionTerritory")
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```

In [ ]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.lineplot(data=dfs,x="Month",y="Cured",hue="State/UnionTerritory")
plt.xticks(rotation=60,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```