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

In [2]:
```
df = pd.read_csv('covid_19_india.csv')
df.head()
```

Out[2]:

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

In [3]:
```
#Checking the number of rows and column of the dataframe
df.shape
```

Out[3]: (16850, 9)

In [4]:
```
#Checking the datatypes of each column
df.dtypes
```

Out[4]:
```
Sno                          int64
Date                        object
Time                        object
State/UnionTerritory        object
ConfirmedIndianNational     object
ConfirmedForeignNational    object
Cured                        int64
Deaths                       int64
Confirmed                    int64
dtype: object
```

In [5]:
```
#Checking if there are any null values in the data
df.isnull().sum()
```

Out[5]:
```
Sno                         0
Date                        0
Time                        0
State/UnionTerritory        0
ConfirmedIndianNational     0
ConfirmedForeignNational    0
Cured                       0
Deaths                      0
Confirmed                   0
dtype: int64
```

In [6]:
```
#Checking the total number of values with '-' in ConfirmedIndianNational and ConfirmedForeign National
df[['ConfirmedIndianNational','ConfirmedForeignNational']].isin(['-']).sum()
```

Out[6]:
```
ConfirmedIndianNational     16404
ConfirmedForeignNational    16404
dtype: int64
```

In [7]:
```
#replacing the '-' with 0
df['ConfirmedIndianNational'].replace('-',0,inplace=True)
df['ConfirmedForeignNational'].replace('-',0,inplace=True)

df.tail()
```

Out[7]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
16845	16846	2021-07-07	8:00 AM	Telangana	0	0	613124	3703	628282
16846	16847	2021-07-07	8:00 AM	Tripura	0	0	63964	701	68612
16847	16848	2021-07-07	8:00 AM	Uttarakhand	0	0	332006	7338	340882
16848	16849	2021-07-07	8:00 AM	Uttar Pradesh	0	0	1682130	22656	1706818
16849	16850	2021-07-07	8:00 AM	West Bengal	0	0	1472132	17834	1507241

In [8]:
```
#Changing the dtypes of column
df[['ConfirmedIndianNational','ConfirmedForeignNational']] = df[['ConfirmedIndianNational','ConfirmedForeignNational
df.dtypes
```

Out[8]:
```
Sno                       int64
Date                     object
Time                     object
State/UnionTerritory     object
ConfirmedIndianNational   int64
ConfirmedForeignNational  int64
Cured                     int64
Deaths                    int64
Confirmed                 int64
dtype: object
```

In [9]:
```
#checking list of the mentioned states in states/unionterritory column
df['State/UnionTerritory'].unique()
```

Out[9]:
```
array(['Kerala', 'Telengana', 'Delhi', 'Rajasthan', 'Uttar Pradesh',
       'Haryana', 'Ladakh', 'Tamil Nadu', 'Karnataka', 'Maharashtra',
       'Punjab', 'Jammu and Kashmir', 'Andhra Pradesh', 'Uttarakhand',
       'Odisha', 'Puducherry', 'West Bengal', 'Chhattisgarh',
       'Chandigarh', 'Gujarat', 'Himachal Pradesh', 'Madhya Pradesh',
       'Bihar', 'Manipur', 'Mizoram', 'Andaman and Nicobar Islands',
       'Goa', 'Unassigned', 'Assam', 'Jharkhand', 'Arunachal Pradesh',
       'Tripura', 'Nagaland', 'Meghalaya',
       'Dadra and Nagar Haveli and Daman and Diu',
       'Cases being reassigned to states', 'Sikkim', 'Daman & Diu',
       'Lakshadweep', 'Telangana', 'Dadra and Nagar Haveli', 'Bihar****'],
      dtype=object)
```

In [10]:
```
#Renaming the wrongly typed states name
df['State/UnionTerritory'].replace(['Bihar****','Telengana','Dadra and Nagar Haveli','Daman & Diu'],['Bihar','Telang
```

In [11]:
```
Unassigned_index = df[df['State/UnionTerritory'] == 'Unassigned'].index
Reassigned_index = df[df['State/UnionTerritory'] == 'Cases being reassigned to states'].index
df.drop(Unassigned_index,inplace=True)
df.drop(Reassigned_index,inplace=True)
```

In [12]:
```
#Rechecking list of the mentioned states in states/unionterritory column
df['State/UnionTerritory'].value_counts()
```

Out[12]:
```
Kerala                                          525
Delhi                                           493
Telangana                                       493
Rajasthan                                       492
Uttar Pradesh                                   491
Haryana                                         491
Ladakh                                          488
Tamil Nadu                                      488
Punjab                                          486
Jammu and Kashmir                               486
Karnataka                                       486
Maharashtra                                     486
Andhra Pradesh                                  483
Uttarakhand                                     480
Odisha                                          479
Puducherry                                      477
West Bengal                                     477
Chandigarh                                      476
Chhattisgarh                                    476
Gujarat                                         475
Himachal Pradesh                                474
Madhya Pradesh                                  474
Bihar                                           473
Manipur                                         471
Mizoram                                         470
Andaman and Nicobar Islands                     469
Goa                                             469
Jharkhand                                       463
Assam                                           463
Arunachal Pradesh                               461
Tripura                                         457
Meghalaya                                       450
Dadra and Nagar Haveli and Daman and Diu        429
Nagaland                                        417
Sikkim                                          410
Lakshadweep                                     209
Name: State/UnionTerritory, dtype: int64
```

In [ ]:
```
1
```

In [13]:
```
1 df.describe()
```

Out[13]:

	Sno	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
count	16787.000000	16787.000000	16787.000000	1.678700e+04	16787.000000	1.678700e+04
mean	8445.623876	0.323822	0.039733	2.369211e+05	3498.302258	2.593157e+05
std	4862.049216	4.023742	0.630006	5.233229e+05	9345.586841	5.681323e+05
min	1.000000	0.000000	0.000000	0.000000e+00	0.000000	0.000000e+00
25%	4260.500000	0.000000	0.000000	2.802500e+03	24.000000	3.650500e+03
50%	8457.000000	0.000000	0.000000	2.900400e+04	463.000000	3.344100e+04
75%	12653.500000	0.000000	0.000000	2.547405e+05	3083.000000	2.674615e+05
max	16850.000000	177.000000	14.000000	5.872268e+06	123531.000000	6.113335e+06

Getting the total number of cases in each year and months

In [14]:
```
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
df['Day'] = df['Date'].dt.day
print(df['Year'].value_counts())
print(df['Month'].value_counts())
```

```
2020    10019
2021     6768
Name: Year, dtype: int64
5     2148
6     2131
4     2028
3     1612
7     1337
1     1118
12    1106
8     1085
10    1085
9     1050
11    1050
2     1037
Name: Month, dtype: int64
```

In [15]:
```
df.head(3)
```

Out[15]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Year	Month	Day
0	1	2020-01-30	6:00 PM	Kerala	1	0	0	0	1	2020	1	30
1	2	2020-01-31	6:00 PM	Kerala	1	0	0	0	1	2020	1	31
2	3	2020-02-01	6:00 PM	Kerala	2	0	0	0	2	2020	2	1

Getting the average of Cured,Deaths,Confirmed from every states

In [16]:
```
1 df.groupby('State/UnionTerritory')[['Cured','Deaths','Confirmed']].mean()
```

Out[16]:

State/UnionTerritory	Cured	Deaths	Confirmed
Andaman and Nicobar Islands	3.390053e+03	48.238806	3.571957e+03
Andhra Pradesh	6.282151e+05	5125.913043	6.711114e+05
Arunachal Pradesh	1.117249e+04	41.872017	1.214387e+04
Assam	1.598517e+05	992.602592	1.736900e+05
Bihar	2.146593e+05	1638.822410	2.289904e+05
Chandigarh	1.676530e+04	250.747899	1.826010e+04
Chhattisgarh	2.461419e+05	3342.701681	2.704869e+05
Dadra and Nagar Haveli and Daman and Diu	3.476312e+03	2.055944	3.700629e+03
Delhi	4.544882e+05	8249.304260	4.806751e+05
Goa	4.312162e+04	721.447761	4.750547e+04
Gujarat	2.189371e+05	3930.128421	2.411739e+05
Haryana	2.036866e+05	2375.912424	2.187543e+05
Himachal Pradesh	4.363454e+04	784.664557	4.863323e+04
Jammu and Kashmir	8.702685e+04	1412.921811	9.650190e+04
Jharkhand	9.953343e+04	1229.585313	1.079299e+05
Karnataka	7.112118e+05	9915.674897	7.975254e+05
Kerala	5.926241e+05	2529.055238	6.558458e+05
Ladakh	6.268535e+03	79.053279	6.852727e+03
Lakshadweep	2.256995e+03	10.421053	2.686407e+03
Madhya Pradesh	2.113285e+05	3012.194093	2.293523e+05
Maharashtra	1.674463e+06	39741.835391	1.870149e+06
Manipur	1.787733e+04	259.212314	2.004440e+04
Meghalaya	1.023677e+04	147.317778	1.160236e+04

State/UnionTerritory	Cured	Deaths	Confirmed
Mizoram	3.265170e+03	10.793617	3.877000e+03
Nagaland	8.701724e+03	94.532374	9.807067e+03
Odisha	2.463148e+05	1252.920668	2.639006e+05
Puducherry	3.014029e+04	523.444444	3.324673e+04
Punjab	1.463142e+05	4561.183128	1.625504e+05
Rajasthan	2.384406e+05	2357.363821	2.621913e+05
Sikkim	4.838778e+03	101.292683	5.647607e+03
Tamil Nadu	6.497285e+05	9695.956967	7.025199e+05
Telangana	2.032682e+05	1253.310345	2.193767e+05
Tripura	2.293035e+04	272.306346	2.494017e+04
Uttar Pradesh	4.735834e+05	6818.036660	5.149566e+05
Uttarakhand	7.642581e+04	1517.733333	8.579041e+04
West Bengal	4.094273e+05	6739.706499	4.398802e+05

In [17]:
```
# df.groupby('State/UnionTerritory').Deaths.mean()
# df.groupby('State/UnionTerritory').Confirmed.mean()
```

Getting the total number of Cured,Death,Confirmed cases from every states

In [18]:
```
1 df.groupby('State/UnionTerritory')[['Cured','Deaths','Confirmed']].sum()
```

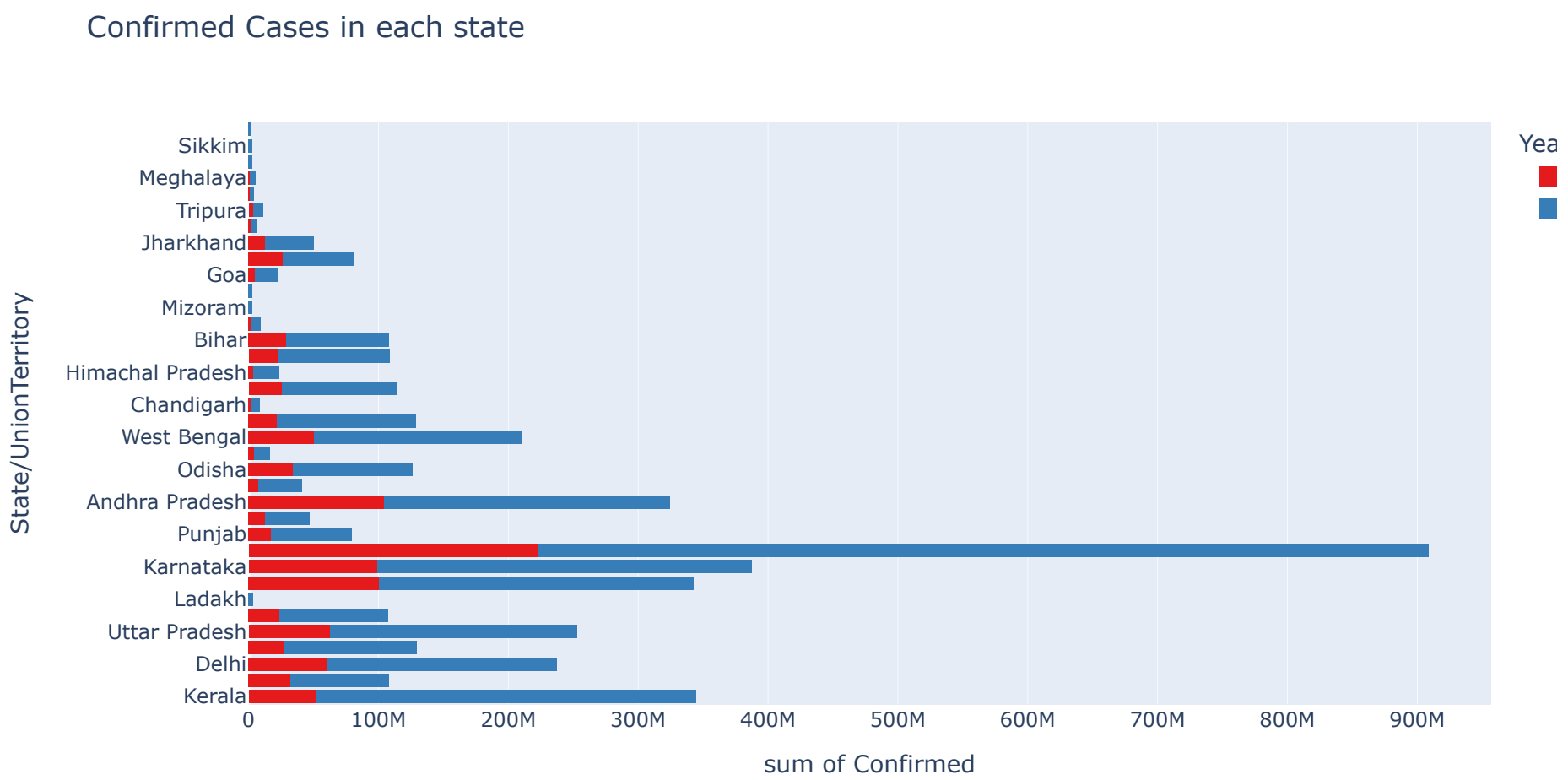
Out[18]:

State/UnionTerritory	Cured	Deaths	Confirmed
Andaman and Nicobar Islands	1589935	22624	1675248
Andhra Pradesh	303427899	2475816	324146783
Arunachal Pradesh	5150519	19303	5598324
Assam	74011348	459575	80418492
Bihar	101533848	775163	108312449
Chandigarh	7980284	119356	8691806
Chhattisgarh	117163544	1591126	128751782
Dadra and Nagar Haveli and Daman and Diu	1491338	882	1587570
Delhi	224062704	4066907	236972842
Goa	20224042	338359	22280065
Gujarat	103995131	1866811	114557615
Haryana	100010131	1166573	107408371
Himachal Pradesh	20682770	371931	23052151
Jammu and Kashmir	42295048	686680	46899925
Jharkhand	46083978	569298	49971564
Karnataka	345648926	4819018	387597335
Kerala	311127643	1327754	344319045
Ladakh	3059045	38578	3344131
Lakshadweep	471712	2178	561459
Madhya Pradesh	100169697	1427780	108712983
Maharashtra	813788907	19314532	908892470
Manipur	8420223	122089	9440912
Meghalaya	4606548	66293	5221064

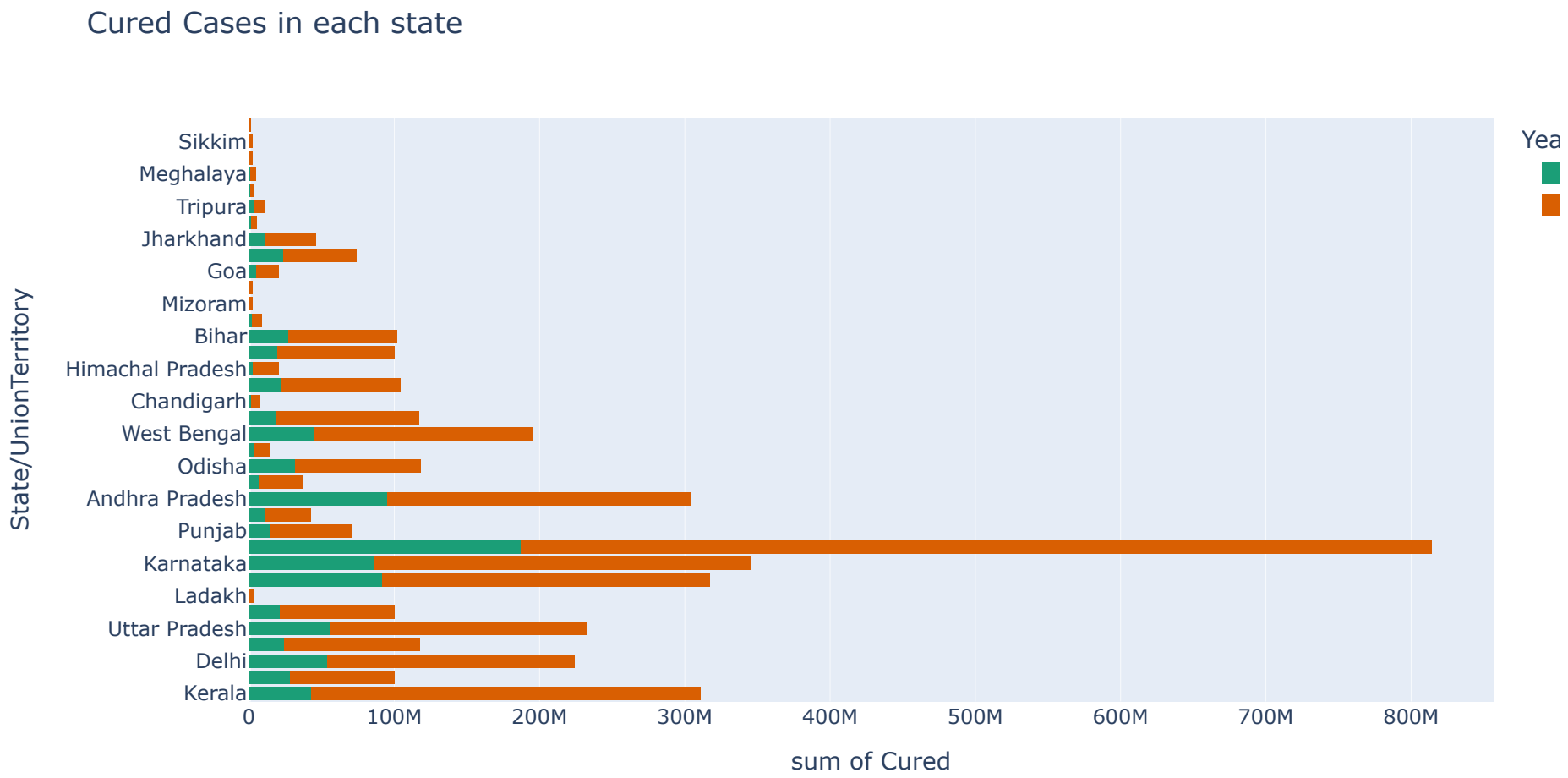
State/UnionTerritory	Cured	Deaths	Confirmed
Mizoram	1534630	5073	1822190
Nagaland	3628619	39420	4089547
Odisha	117984789	600149	126408397
Puducherry	14376916	249683	15858688
Punjab	71108712	2216735	78999515
Rajasthan	117312772	1159823	128998101
Sikkim	1983899	41530	2315519
Tamil Nadu	317067499	4731627	342829697
Telangana	100211245	617882	108152726
Tripura	10479169	124444	11397656
Uttar Pradesh	232529439	3347656	252843682
Uttarakhand	36684388	728512	41179396
West Bengal	195296839	3214840	209822848

In [19]:
```
# df.groupby('State/UnionTerritory').Deaths.sum()
# df.groupby('State/UnionTerritory').Confirmed.sum()
```

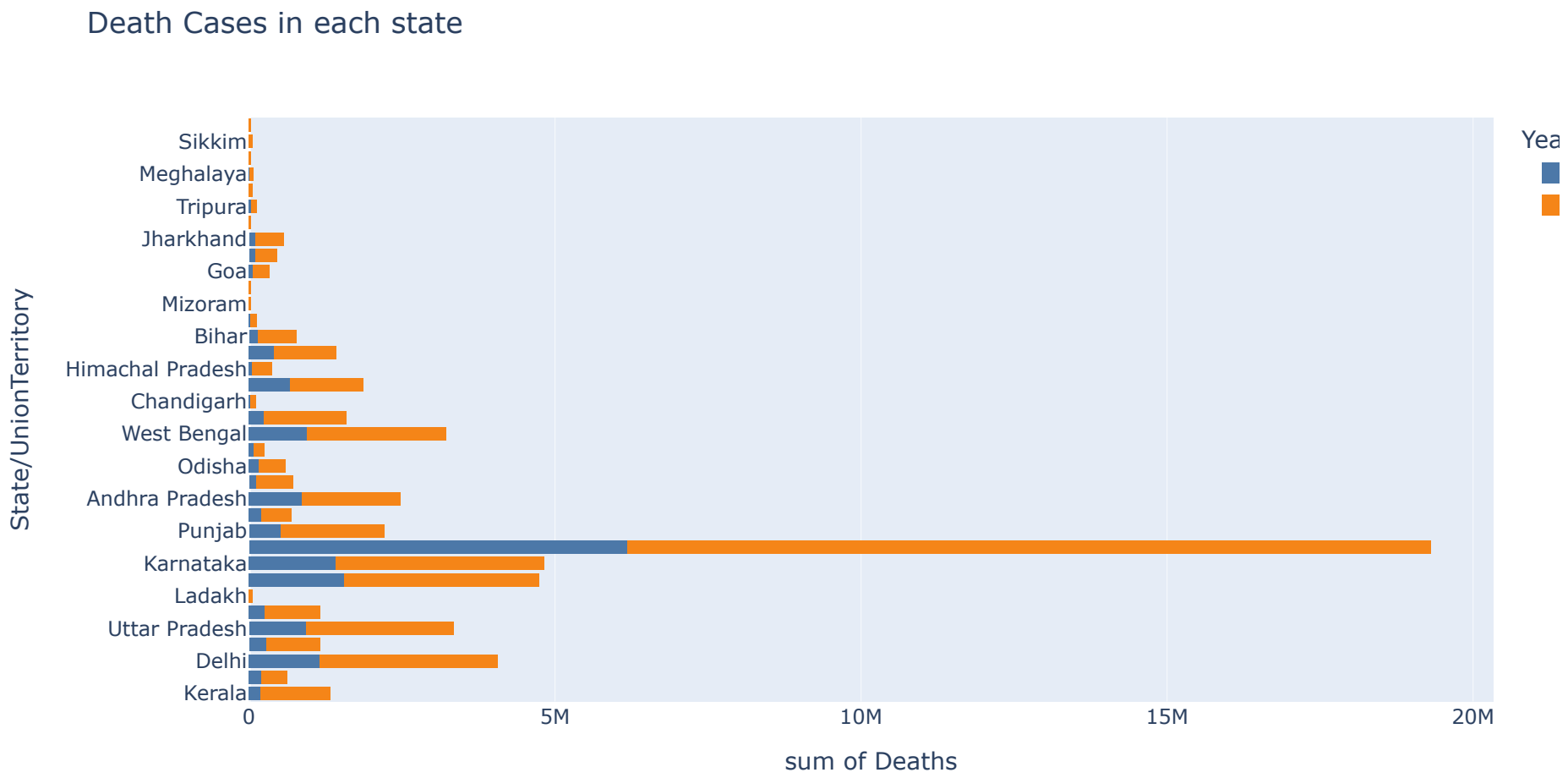
In [20]:
```
1 px.histogram(title='Confirmed Cases in each state',x='Confirmed',y='State/UnionTerritory',color='Year',data_frame=df
```



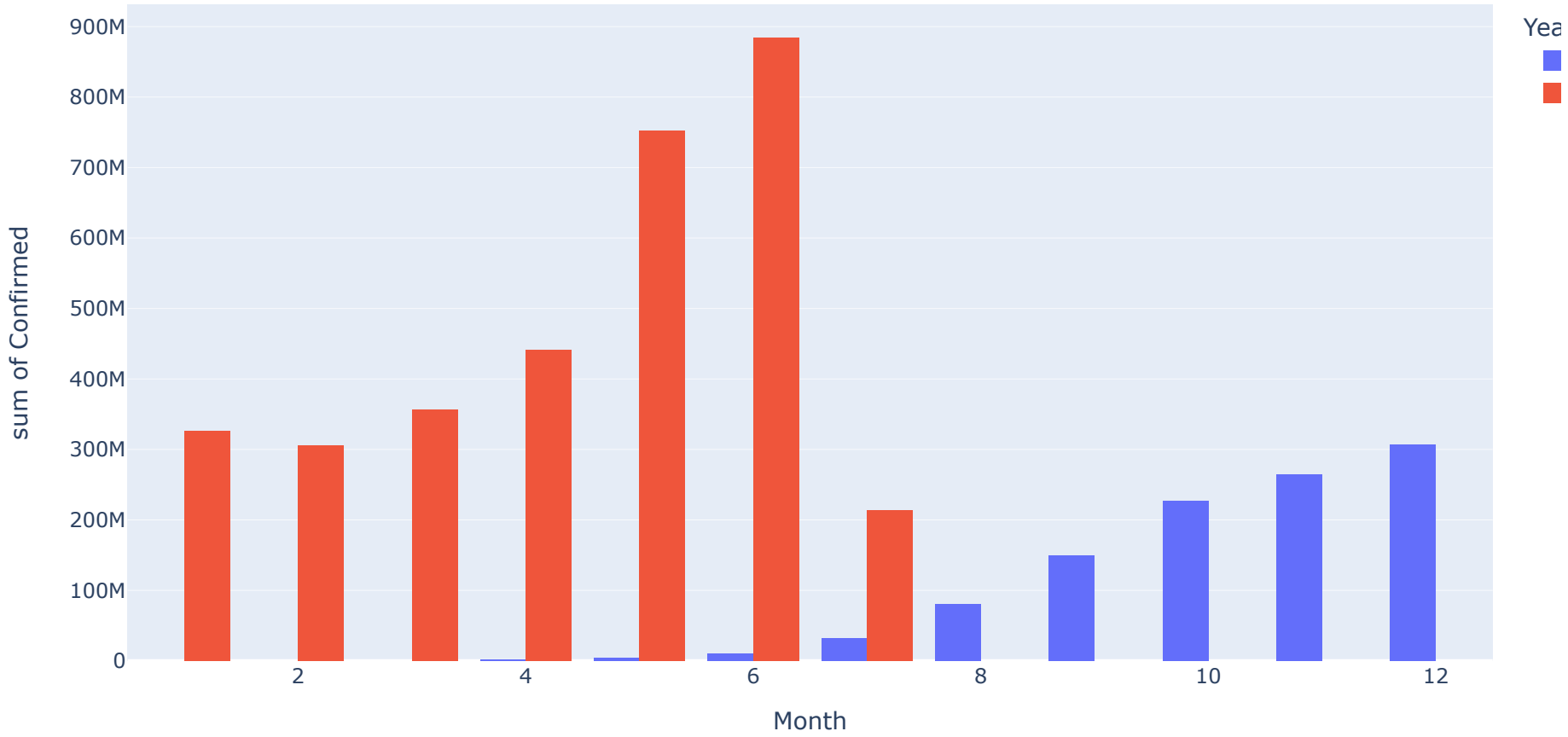
In [21]:
```
1 px.histogram(title='Cured Cases in each state',x='Cured',y='State/UnionTerritory',color='Year',data_frame=df,color_d
```



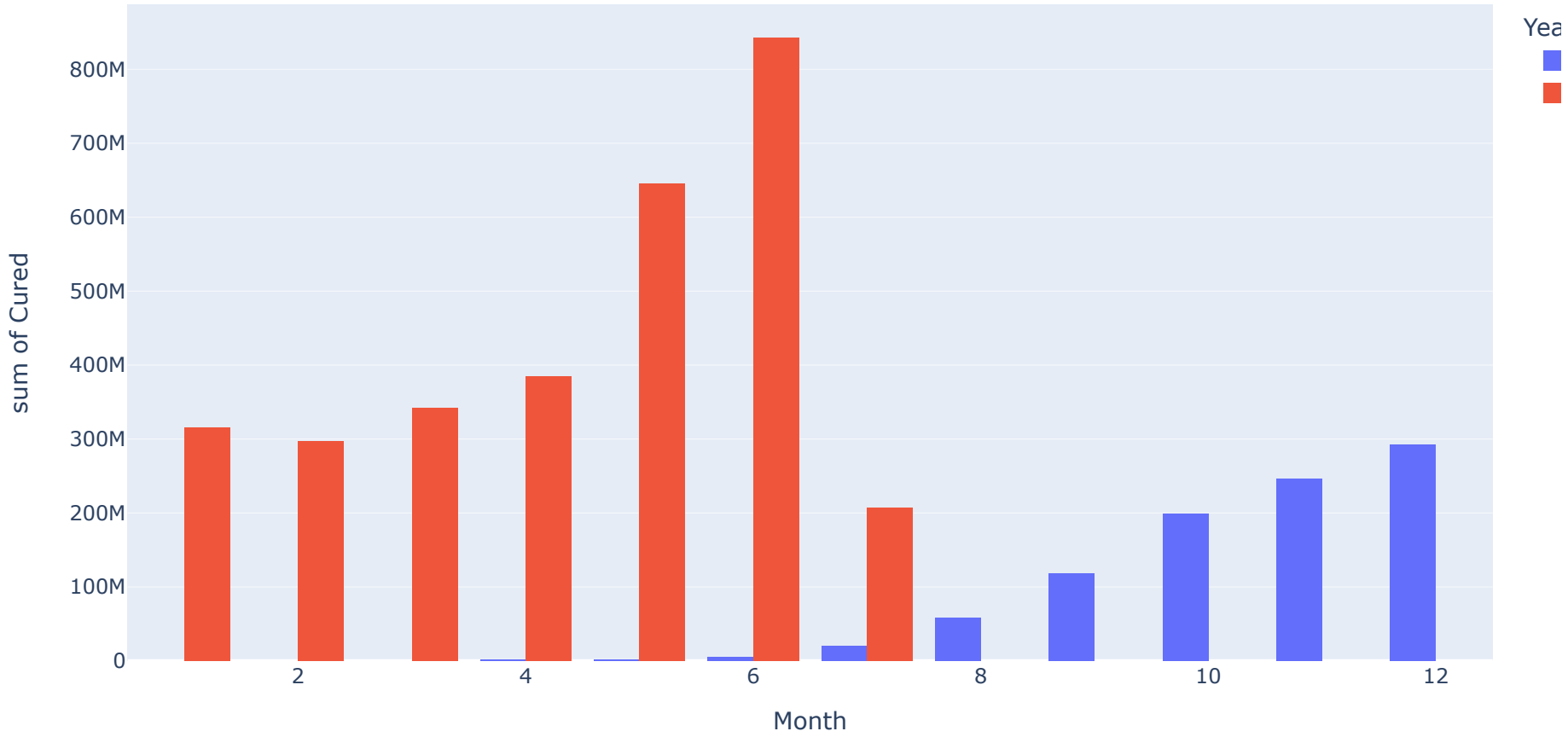
In [22]:
```
1 px.histogram(title='Death Cases in each state',x='Deaths',y='State/UnionTerritory',color='Year',data_frame=df,color_
```



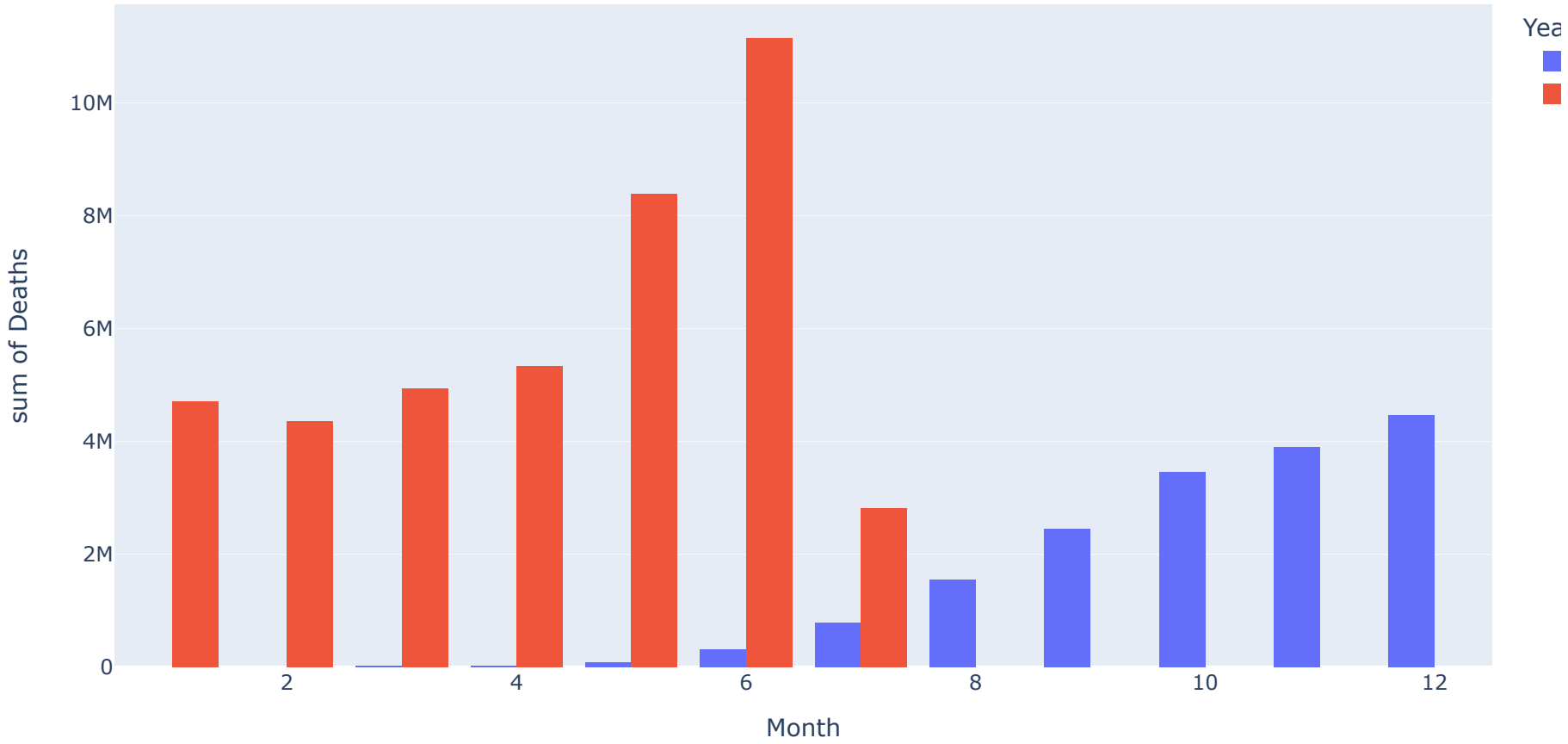
In [23]:
```
#Comparing confirmed case in each year
px.histogram(x='Month',y='Confirmed',barmode='group',color='Year',data_frame=df)
```



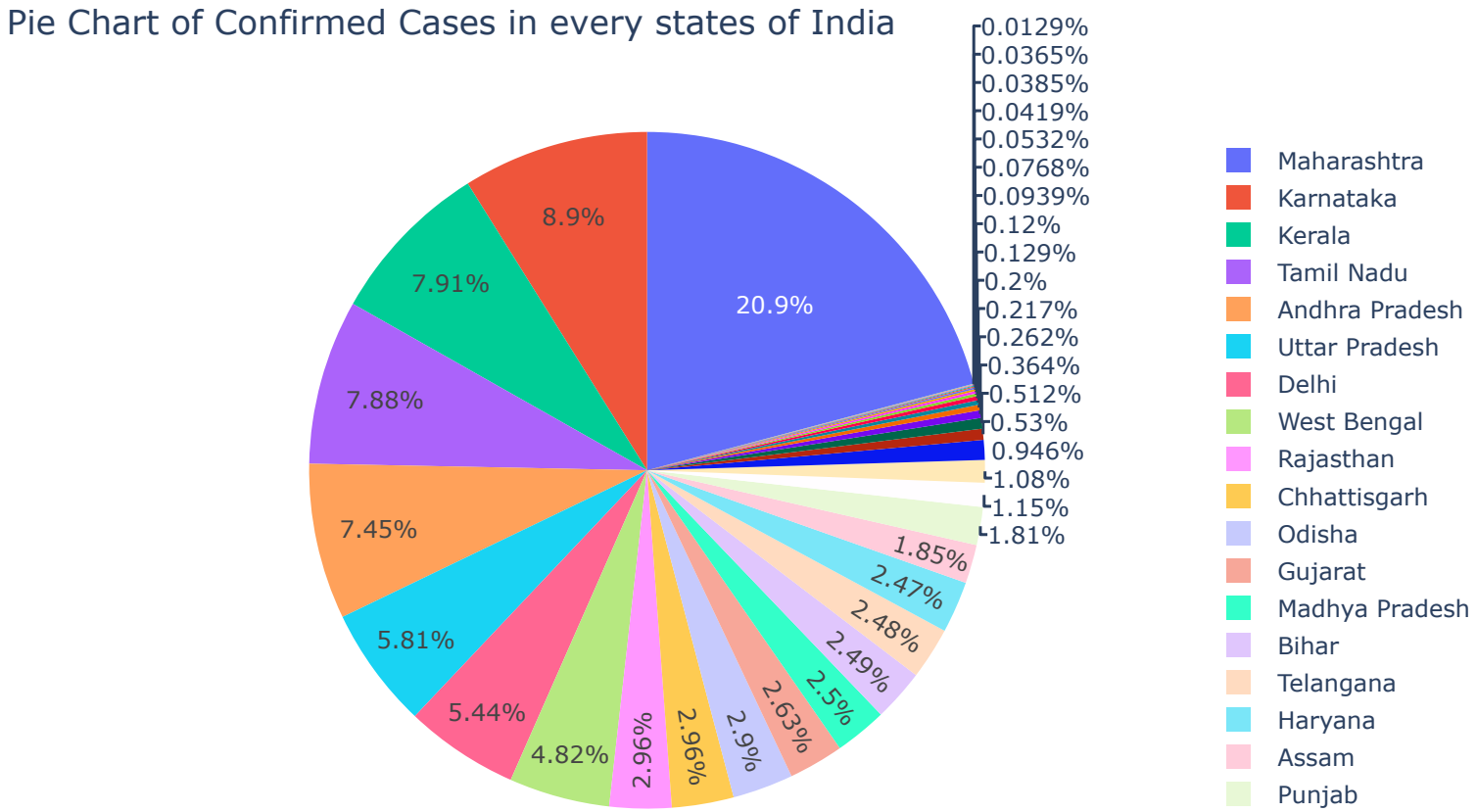
In [24]:
```
#Comparing Cured case in each year
px.histogram(x='Month',y='Cured',barmode='group',color='Year',data_frame=df)
```



In [25]:
```
#Comparing Death case in each year
px.histogram(x='Month',y='Deaths',barmode='group',color='Year',data_frame=df)
```



In [26]:
```
1 px.pie(data_frame=df,names='State/UnionTerritory',values='Confirmed',title='Pie Chart of Confirmed Cases in every st
```



In [27]:
```
1 # fig = px.choropleth(data_frame=df,geojson='https://gist.githubusercontent.com/jbrobst/56c13bbbf9d97d187fea01ca62ea
2 # fig.update_geos(fitbounds="locations", visible=False)
3 # fig.show()
```

In [ ]:
```
1
```

Analysis for Maharashtra State

In [28]:
```
1 df_maharashtra = df[df['State/UnionTerritory'] == 'Maharashtra']
2 df_maharashtra.head(3)
```

Out[28]:

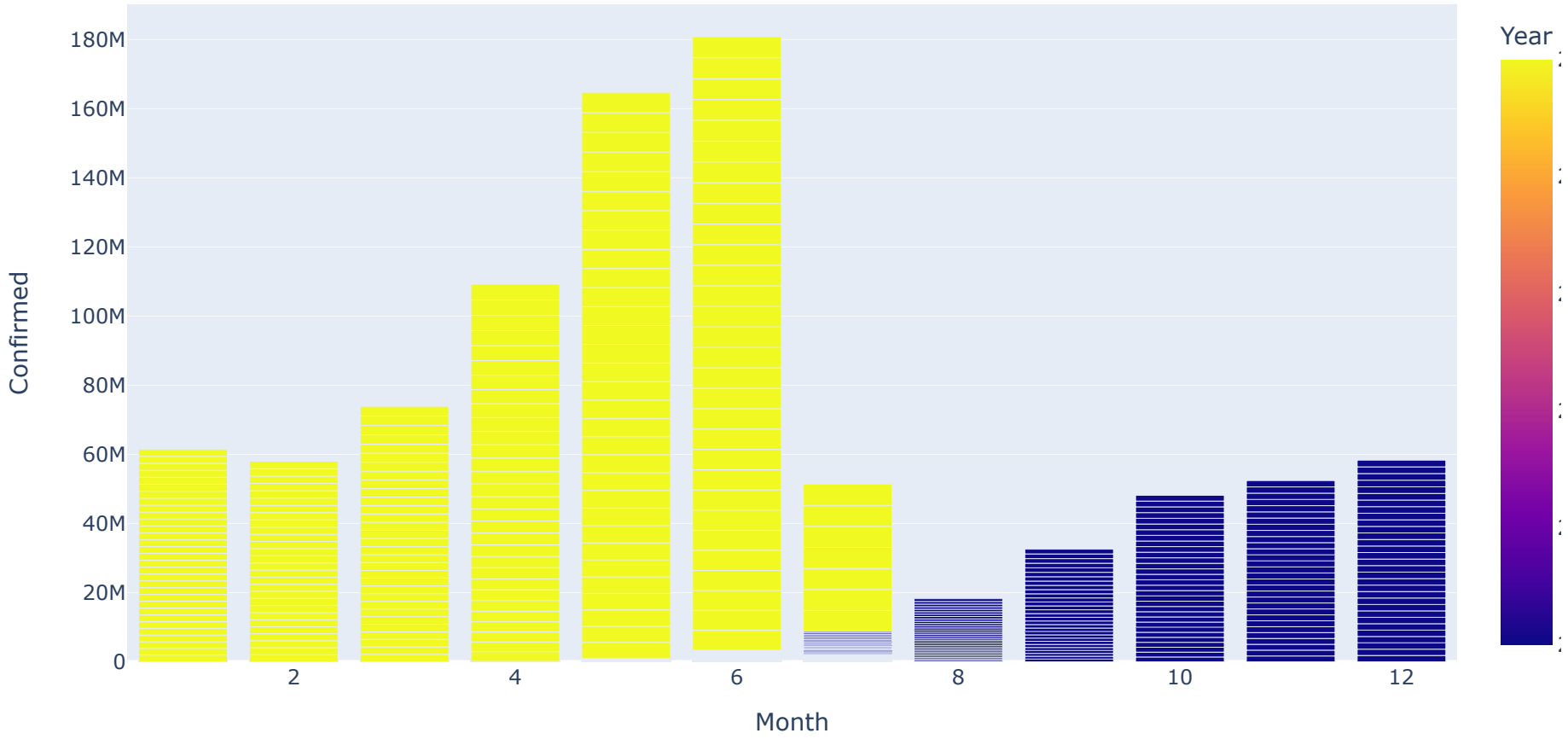
	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Year	Month	Day
76	77	2020-03-09	6:00 PM	Maharashtra	2	0	0	0	2	2020	3	9
91	92	2020-03-10	6:00 PM	Maharashtra	5	0	0	0	5	2020	3	10
97	98	2020-03-11	6:00 PM	Maharashtra	2	0	0	0	2	2020	3	11

In [29]:
```
1 df_maharashtra.describe()
```

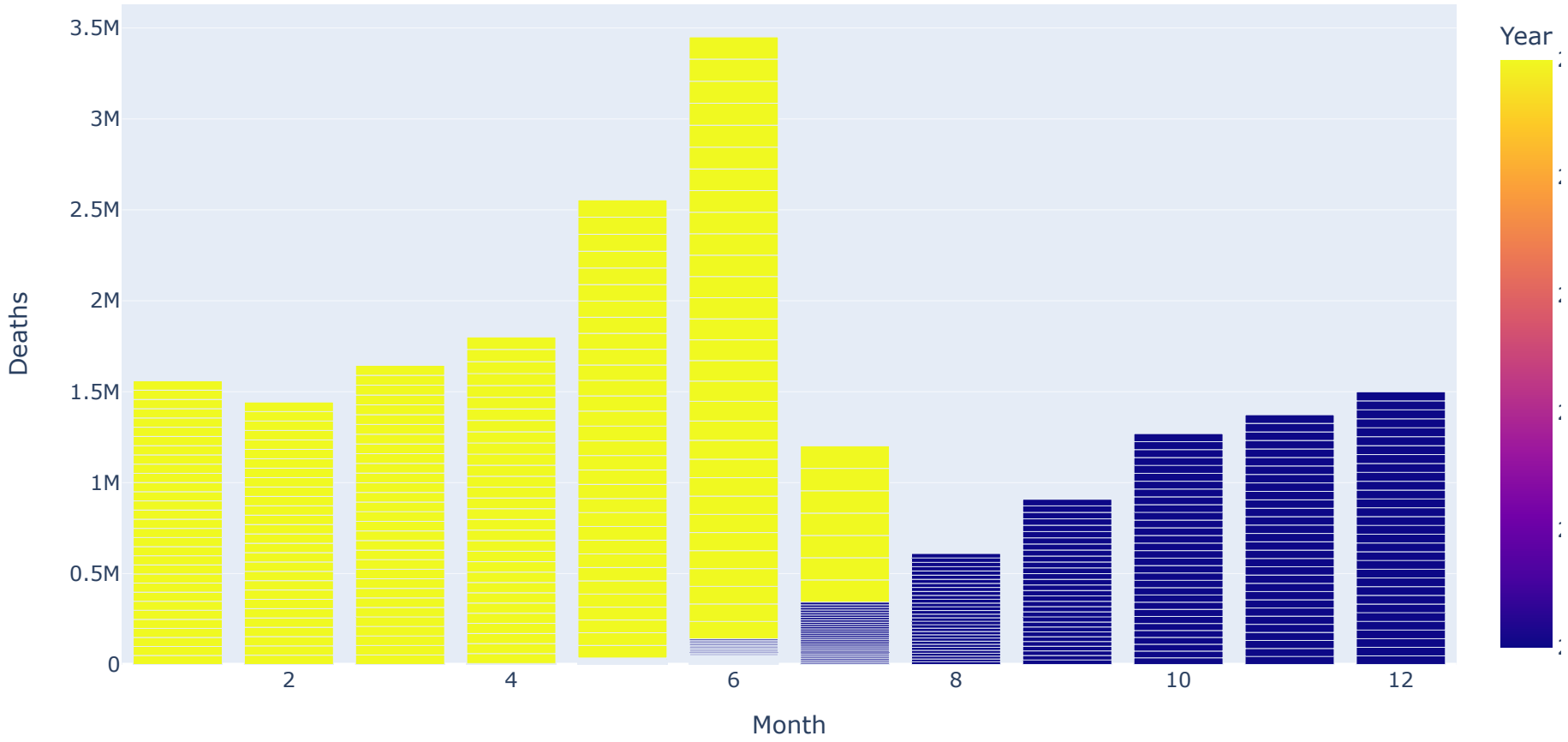
Out[29]:

	Sno	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Year	Month	
count	486.000000	486.000000	486.000000	4.860000e+02	486.000000	4.860000e+02	486.000000	486.000000	486.0
mean	8191.660494	2.286008	0.074074	1.674463e+06	39741.835391	1.870149e+06	2020.386831	6.080247	15.7
std	4955.978167	14.718335	0.466028	1.710989e+06	31861.231600	1.831266e+06	0.487526	3.146548	8.8
min	77.000000	0.000000	0.000000	0.000000e+00	0.000000	2.000000e+00	2020.000000	1.000000	1.0
25%	3884.000000	0.000000	0.000000	1.197165e+05	9299.500000	2.187718e+05	2020.000000	4.000000	8.0
50%	8138.500000	0.000000	0.000000	1.556812e+06	44884.500000	1.706879e+06	2020.000000	6.000000	16.0
75%	12470.000000	0.000000	0.000000	2.066541e+06	52468.500000	2.216942e+06	2021.000000	8.750000	23.0
max	16835.000000	177.000000	3.000000	5.872268e+06	123531.000000	6.113335e+06	2021.000000	12.000000	31.0

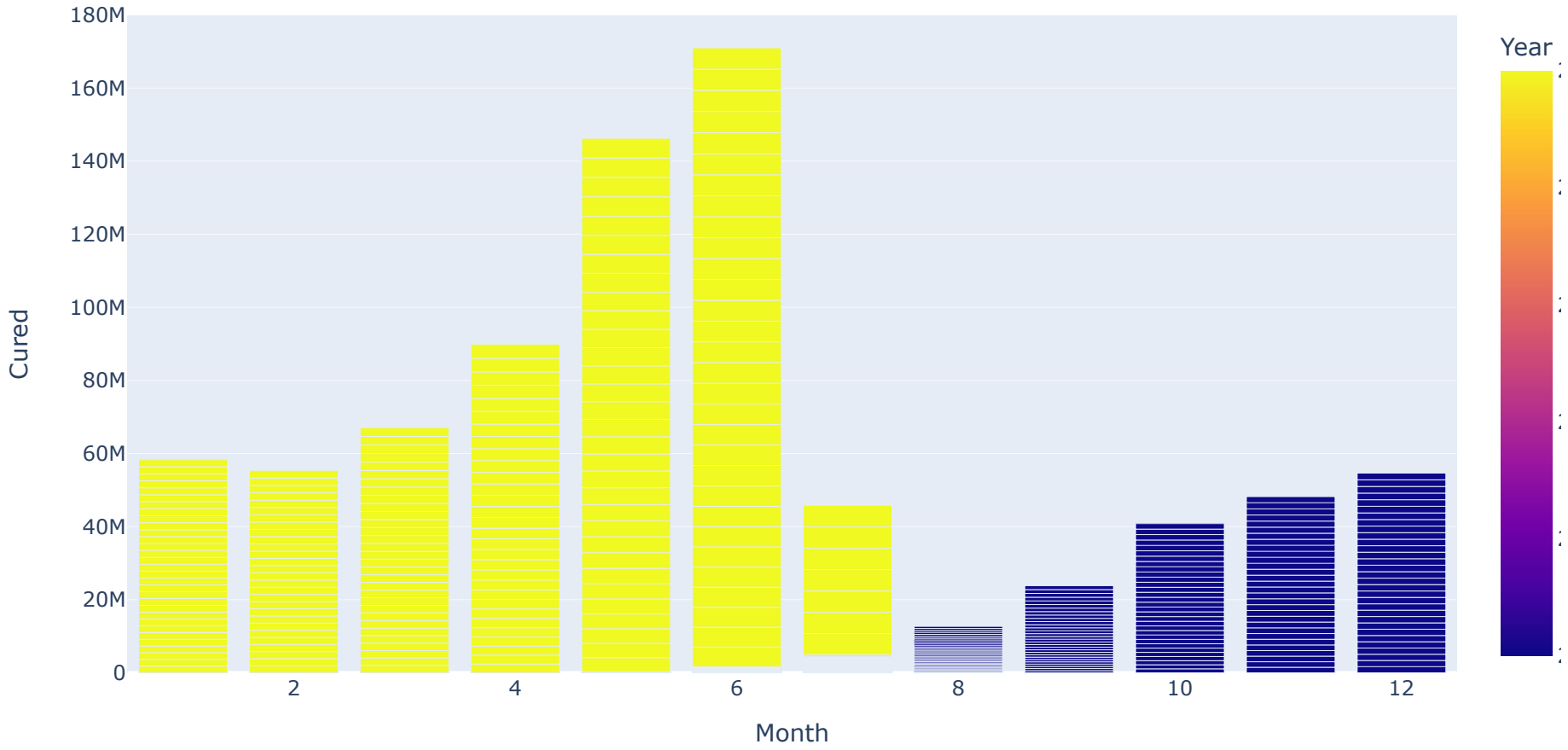
In [30]:
```
1 px.bar(x='Month',y='Confirmed',color='Year',data_frame=df_maharashtra)
```



In [31]:
```
1 px.bar(x='Month',y='Deaths',color='Year',data_frame=df_maharashtra)
```



In [32]:
```
1 px.bar(x='Month',y='Cured',color='Year',data_frame=df_maharashtra)
```



In [ ]:
```
1
```