In [1]:

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

In [2]:

```
#load dataset
covid=pd.read_csv(r"D:\Seema\important-20230710T100403Z-001\important\covid_19_
covid
```

Out[2]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNationa
0	1	30-01-2020	6:00 PM	Kerala	1	
1	2	31-01-2020	6:00 PM	Kerala	1	
2	3	01-02-2020	6:00 PM	Kerala	2	
3	4	02-02-2020	6:00 PM	Kerala	3	
4	5	03-02-2020	6:00 PM	Kerala	3	
...	...	...	...	...	...	
18105	18106	11-08-2021	8:00 AM	Telangana	-	
18106	18107	11-08-2021	8:00 AM	Tripura	-	
18107	18108	11-08-2021	8:00 AM	Uttarakhand	-	
18108	18109	11-08-2021	8:00 AM	Uttar Pradesh	-	
18109	18110	11-08-2021	8:00 AM	West Bengal	-	

18110 rows × 9 columns

In [3]:

```
df=covid[['Date','State/UnionTerritory','Cured','Deaths','Confirmed']]
df
```

Out[3]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed
0	30-01-2020	Kerala	0	0	1
1	31-01-2020	Kerala	0	0	1
2	01-02-2020	Kerala	0	0	2
3	02-02-2020	Kerala	0	0	3
4	03-02-2020	Kerala	0	0	3
...	...	...	...	...	...
18105	11-08-2021	Telangana	638410	3831	650353
18106	11-08-2021	Tripura	77811	773	80660
18107	11-08-2021	Uttarakhand	334650	7368	342462
18108	11-08-2021	Uttar Pradesh	1685492	22775	1708812
18109	11-08-2021	West Bengal	1506532	18252	1534999

18110 rows × 5 columns

In [8]:

```
#fetching confirmed cases on particular date
today=df[df['Date']=='11-08-2021']
all_sorted_confirmed=today.sort_values(by='Confirmed',ascending=False)
top_five_state_confirmed=all_sorted_confirmed[0:5]
top_five_state_confirmed
```

Out[8]:

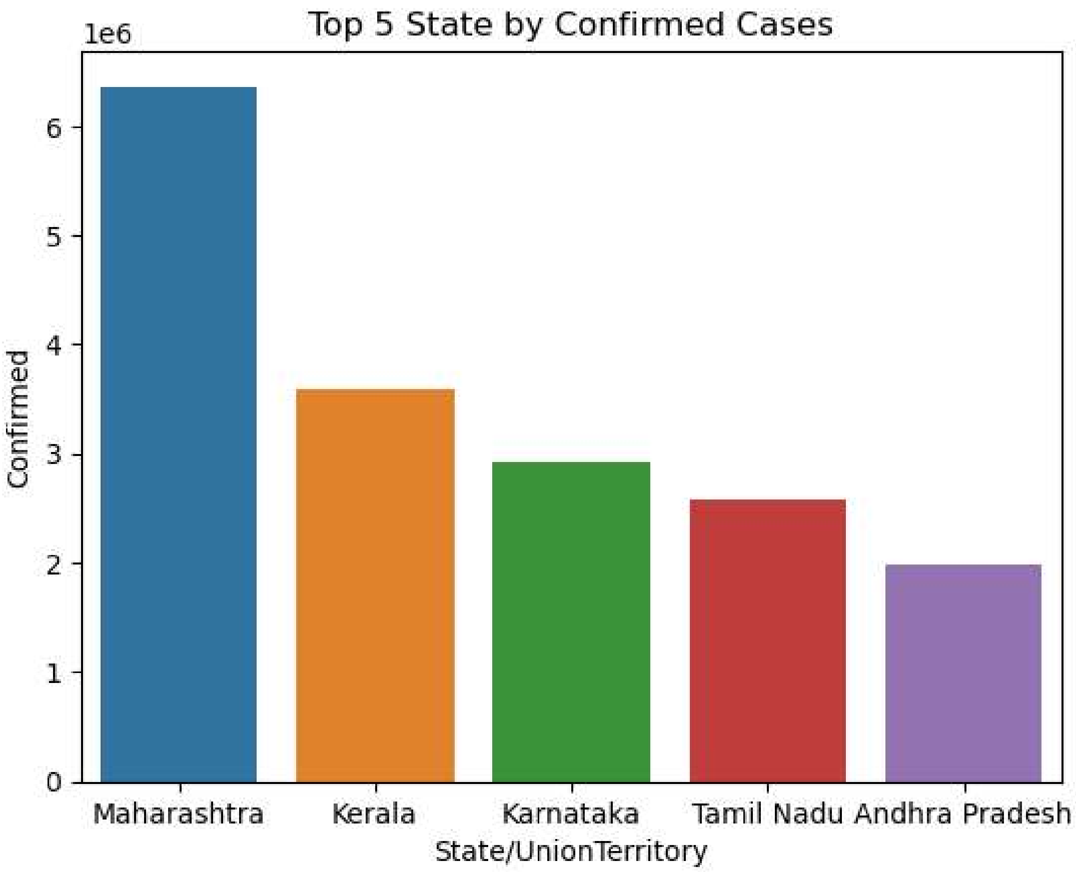
	Date	State/UnionTerritory	Cured	Deaths	Confirmed
18094	11-08-2021	Maharashtra	6159676	134201	6363442
18090	11-08-2021	Kerala	3396184	18004	3586693
18089	11-08-2021	Karnataka	2861499	36848	2921049
18104	11-08-2021	Tamil Nadu	2524400	34367	2579130
18075	11-08-2021	Andhra Pradesh	1952736	13564	1985182

In [10]:

```
#Graphical Representation
sns.barplot(x='State/UnionTerritory',y='Confirmed',data=top_five_state_confirm
plt.title("Top 5 State by Confirmed Cases")
```

Out[10]:

```
Text(0.5, 1.0, 'Top 5 State by Confirmed Cases')
```



In [11]:

```
#State wise deaths
statedeaths=df[['State/UnionTerritory','Deaths']]
statedeaths
```

Out[11]:

	State/UnionTerritory	Deaths
0	Kerala	0
1	Kerala	0
2	Kerala	0
3	Kerala	0
4	Kerala	0
...	...	...
18105	Telangana	3831
18106	Tripura	773
18107	Uttarakhand	7368
18108	Uttar Pradesh	22775
18109	West Bengal	18252

18110 rows × 2 columns

In [14]:

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

Out[14]:

```
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****',
       'Madhya Pradesh***', 'Himanchal Pradesh', 'Karanataka',
       'Maharashtra***'], dtype=object)
```

In [15]:

```
#replace names
statedeaths['State/UnionTerritory'].replace('Bihar****','Bihar',inplace=True)
statedeaths['State/UnionTerritory'].replace('Maharashtra***','Maharashtra',inp
statedeaths['State/UnionTerritory'].replace('Madhya Pradesh***','Madhya Prades
```

```
C:\Users\PC\AppData\Local\Temp\ipykernel_24840\3184741632.py:2: SettingWithCo
pyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  statedeaths['State/UnionTerritory'].replace('Bihar****','Bihar',inplace=Tru
e)
C:\Users\PC\AppData\Local\Temp\ipykernel_24840\3184741632.py:3: SettingWithCo
pyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  statedeaths['State/UnionTerritory'].replace('Maharashtra***','Maharashtra',
inplace=True)
C:\Users\PC\AppData\Local\Temp\ipykernel_24840\3184741632.py:4: SettingWithCo
pyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  statedeaths['State/UnionTerritory'].replace('Madhya Pradesh***','Madhya Pra
desh',inplace=True)
```

In [16]:

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

Out[16]:

```
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',
       'Himanchal Pradesh', 'Karanataka'], dtype=object)
```

In [22]:

```
top_five_state_deaths=statedeaths.groupby('State/UnionTerritory').sum().sort_v
top_five_state_deaths[0:5]
```

Out[22]:

	Deaths
State/UnionTerritory	
Maharashtra	23868185
Karnataka	6053762
Tamil Nadu	5916658
Delhi	4943294
Uttar Pradesh	4143450

In [26]:

```
#visualization
a=top_five_state_deaths[0:5]
```

In [27]:

```
a
```

Out[27]:

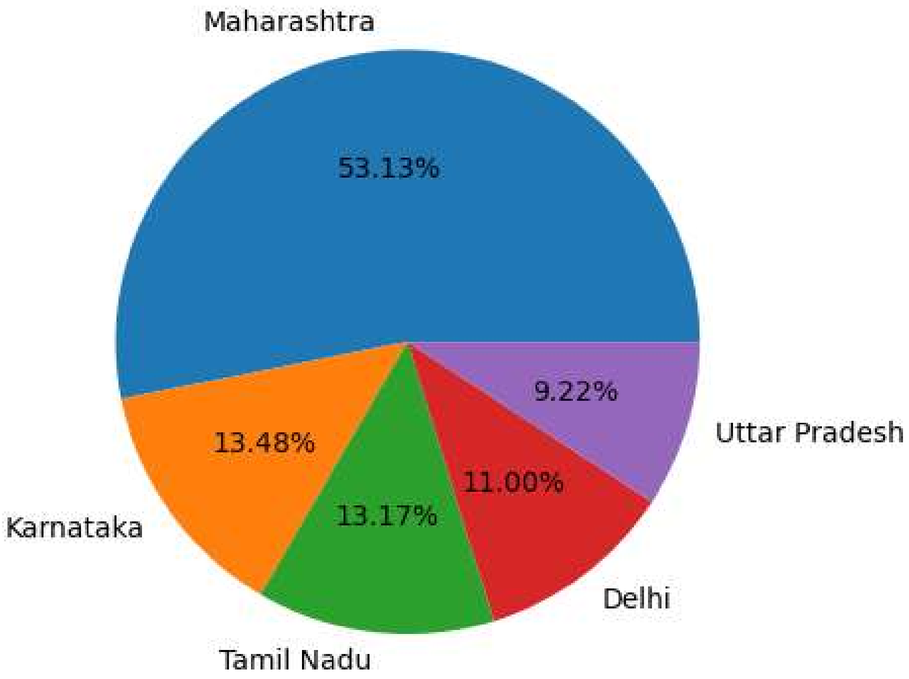
	Deaths
State/UnionTerritory	
Maharashtra	23868185
Karnataka	6053762
Tamil Nadu	5916658
Delhi	4943294
Uttar Pradesh	4143450

In [30]:

```
values=a['Deaths']
states=a.index
plt.pie(values,labels=states,autopct="%0.2f%%")
```

Out[30]:

```
([<matplotlib.patches.Wedge at 0x2684b4bdc90>,
  <matplotlib.patches.Wedge at 0x2684b4bdba0>,
  <matplotlib.patches.Wedge at 0x2684b4be8c0>,
  <matplotlib.patches.Wedge at 0x2684b4bef50>,
  <matplotlib.patches.Wedge at 0x2684b4db5b0>],
 [Text(-0.10794102595533751, 1.0946911596042554, 'Maharashtra'),
  Text(-0.895325487137805, -0.6390557660184691, 'Karnataka'),
  Text(-0.12491786386532254, -1.0928840410982883, 'Tamil Nadu'),
  Text(0.6618634841666275, -0.8785992990702943, 'Delhi'),
  Text(1.0541477049606487, -0.3142811100371727, 'Uttar Pradesh')],
 [Text(-0.05887692324836591, 0.5971042688750483, '53.13%'),
  Text(-0.48835935662062085, -0.3485758723737104, '13.48%'),
  Text(-0.06813701665381229, -0.5961185678717935, '13.17%'),
  Text(0.3610164459090695, -0.4792359813110696, '11.00%'),
  Text(0.5749896572512628, -0.171426060020276, '9.22%')])
```



In [31]:

```
datewise=df[['Date','Deaths']]
datewise['Date'].unique()
```

```
          '22-04-2021', '23-04-2021', '24-04-2021', '25-04-2021',
          '26-04-2021', '27-04-2021', '28-04-2021', '29-04-2021',
          '30-04-2021', '01-05-2021', '02-05-2021', '03-05-2021',
          '04-05-2021', '05-05-2021', '06-05-2021', '07-05-2021',
          '08-05-2021', '09-05-2021', '10-05-2021', '11-05-2021',
          '12-05-2021', '13-05-2021', '14-05-2021', '15-05-2021',
          '16-05-2021', '17-05-2021', '18-05-2021', '19-05-2021',
          '20-05-2021', '21-05-2021', '22-05-2021', '23-05-2021',
          '24-05-2021', '25-05-2021', '26-05-2021', '27-05-2021',
          '28-05-2021', '29-05-2021', '30-05-2021', '31-05-2021',
          '01-06-2021', '02-06-2021', '03-06-2021', '04-06-2021',
          '05-06-2021', '06-06-2021', '07-06-2021', '08-06-2021',
          '09-06-2021', '10-06-2021', '11-06-2021', '12-06-2021',
          '13-06-2021', '14-06-2021', '15-06-2021', '16-06-2021',
          '17-06-2021', '18-06-2021', '19-06-2021', '20-06-2021',
          '21-06-2021', '22-06-2021', '23-06-2021', '24-06-2021',
          '25-06-2021', '26-06-2021', '27-06-2021', '28-06-2021',
          '29-06-2021', '30-06-2021', '01-07-2021', '02-07-2021',
          '03-07-2021', '04-07-2021', '05-07-2021', '06-07-2021',
          '07-07-2021', '08-07-2021', '09-07-2021', '10-07-2021',
```

In [32]:

```
datewise.head(3)
```

Out[32]:

	Date	Deaths
0	30-01-2020	0
1	31-01-2020	0
2	01-02-2020	0

In [34]:

```
lastweek=datewise.query('Date>="05-08-2021" and Date<="11-08-2021"')
lwcases=lastweek.groupby('Date').sum()
lwcases
```

Out[34]:

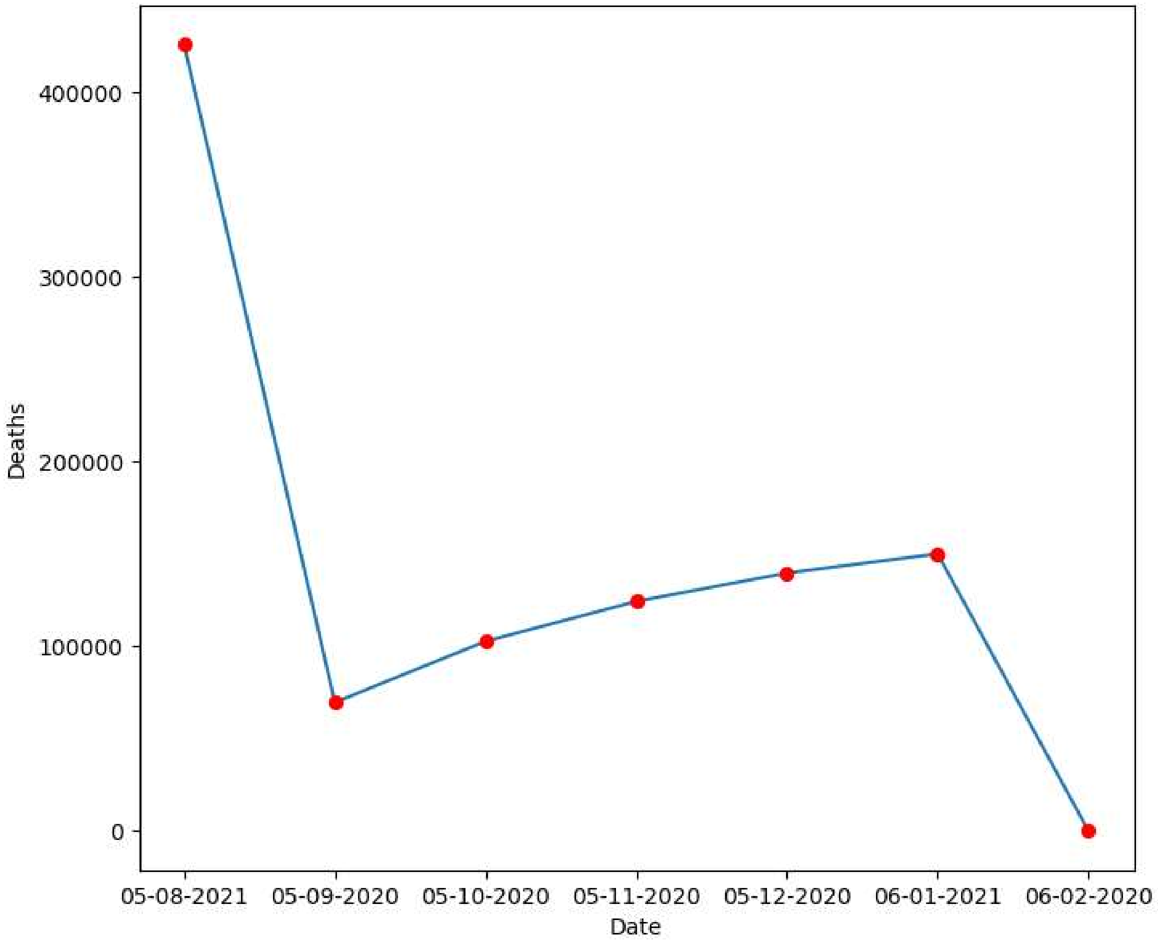
```
             Deaths
      Date
05-08-2021   426290
05-09-2020    69561
05-10-2020   102685
05-11-2020   124315
05-12-2020   139700
       ...      ...
11-06-2021   363079
11-07-2020    22123
11-07-2021   408040
11-08-2020    45257
11-08-2021   429179

115 rows × 1 columns
```

In [40]:

```
#graphical form
plt.figure(figsize=(8,7))
b=lwcases[0:7]
sns.lineplot(x=b.index,y='Deaths',data=b,marker="o",mfc="r",mec="r")
```

Out[40]:

```
<Axes: xlabel='Date', ylabel='Deaths'>
```



In [ ]:

```

```