

Q.1 Check unique state/UT names, fix state names appearing twice or more due to spelling mistakes

In [2]:

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

In [6]:

```
df = pd.read_csv("covid_19_india.csv")
```

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9291 entries, 0 to 9290
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Sno                                    9291 non-null   int64
1   Date                                  9291 non-null   object
2   Time                                  9291 non-null   object
3   State/UnionTerritory                 9291 non-null   object
4   ConfirmedIndianNational              9291 non-null   object
5   ConfirmedForeignNational             9291 non-null   object
6   Cured                                9291 non-null   int64
7   Deaths                              9291 non-null   int64
8   Confirmed                            9291 non-null   int64
dtypes: int64(4), object(5)
memory usage: 653.4+ KB
```

In [8]:

```
df.describe()
```

Out[8]:

	Sno	Cured	Deaths	Confirmed
count	9291.000000	9.291000e+03	9291.000000	9.291000e+03
mean	4646.000000	7.863266e+04	1487.620385	9.183978e+04
std	2682.225009	1.931102e+05	4713.813690	2.166014e+05
min	1.000000	0.000000e+00	0.000000	0.000000e+00
25%	2323.500000	1.520000e+02	2.000000	5.385000e+02
50%	4646.000000	4.308000e+03	66.000000	6.832000e+03
75%	6968.500000	5.772650e+04	926.500000	7.885600e+04
max	9291.000000	1.737080e+06	47827.000000	1.859367e+06

In [9]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[9]:

<AxesSubplot:>



In [10]:

```
df.head()
```

Out[10]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational
0	1	30/01/20	6:00 PM	Kerala	1	0
1	2	31/01/20	6:00 PM	Kerala	1	0
2	3	01/02/20	6:00 PM	Kerala	2	0
3	4	02/02/20	6:00 PM	Kerala	3	0
4	5	03/02/20	6:00 PM	Kerala	3	0

In [11]:

```
df.tail()
```

Out[11]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNatic
9286	9287	09/12/20	8:00 AM	Telengana	-	
9287	9288	09/12/20	8:00 AM	Tripura	-	
9288	9289	09/12/20	8:00 AM	Uttarakhand	-	
9289	9290	09/12/20	8:00 AM	Uttar Pradesh	-	
9290	9291	09/12/20	8:00 AM	West Bengal	-	

In [12]:

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

Out[12]:

```
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', 'Dadar Nagar Haveli',
      'Cases being reassigned to states', 'Sikkim', 'Daman & Diu',
      'Dadra and Nagar Haveli and Daman and Diu', 'Telangana',
      'Telangana***', 'Telengana***', 'Maharashtra***', 'Chandigarh***',
      'Punjab***'], dtype=object)
```

In [14]:

```
df["State/UnionTerritory"].replace("Maharashtra***", "Maharashtra" )
df["State/UnionTerritory"].replace("Telengana***", "Telangana" )
df["State/UnionTerritory"].replace("Chandigarh***", "Chandigarh" )
df["State/UnionTerritory"].replace("Punjab***", "Punjab" )
df["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', 'Dadar Nagar Haveli',
      'Cases being reassigned to states', 'Sikkim', 'Daman & Diu',
      'Dadra and Nagar Haveli and Daman and Diu', 'Telangana',
      'Telangana***', 'Telengana***', 'Maharashtra***', 'Chandigarh***',
      'Punjab***'], dtype=object)
```

Q.2 Calculate per day average confirmed cases for all states/UT.

In [17]:

```
df.groupby(["Date", "State/UnionTerritory"])[ "Confirmed" ].mean()
```

Out[17]:

Date	State/UnionTerritory	
01/02/20	Kerala	2
01/03/20	Kerala	3
01/04/20	Andaman and Nicobar Islands	10
	Andhra Pradesh	83
	Assam	1
	Bihar	23
	Chandigarh	16
	Chhattisgarh	9
	Delhi	152
	Goa	5
	Gujarat	82
	Haryana	43
	Himachal Pradesh	3
	Jammu and Kashmir	62
	Jharkhand	1
	Karnataka	101
	Kerala	241
	Ladakh	13
	Madhya Pradesh	66
	Maharashtra	302
	Manipur	1
	Mizoram	1
	Odisha	4
	Puducherry	3
	Punjab	42
	Rajasthan	93
	Tamil Nadu	234
	Telengana	96
	Uttar Pradesh	103
	Uttarakhand	7
	...	
31/10/20	Chandigarh	14351
	Chhattisgarh	185306
	Dadra and Nagar Haveli and Daman and Diu	3248
	Delhi	381644
	Goa	43416
	Gujarat	171847
	Haryana	165467
	Himachal Pradesh	21798
	Jammu and Kashmir	94330
	Jharkhand	101287
	Karnataka	820398
	Kerala	425122
	Ladakh	6194
	Madhya Pradesh	170690
	Maharashtra	1672858
	Manipur	18272
	Meghalaya	9382
	Mizoram	2722
	Nagaland	8945
	Odisha	288646
	Puducherry	34908
	Punjab	133158
	Rajasthan	195213

Sikkim	3913
Tamil Nadu	722011
Telengana	238632
Tripura	30717
Uttar Pradesh	480082
Uttarakhand	61915
West Bengal	369671

Name: Confirmed, Length: 9291, dtype: int64

Q.3 Plot a linegraph that shows distribution of per day confirmed cases in Maharashtra in 2020.

In [17]:

```
mh = df[df["State/UnionTerritory"]=="Maharashtra"]
```

In [18]:

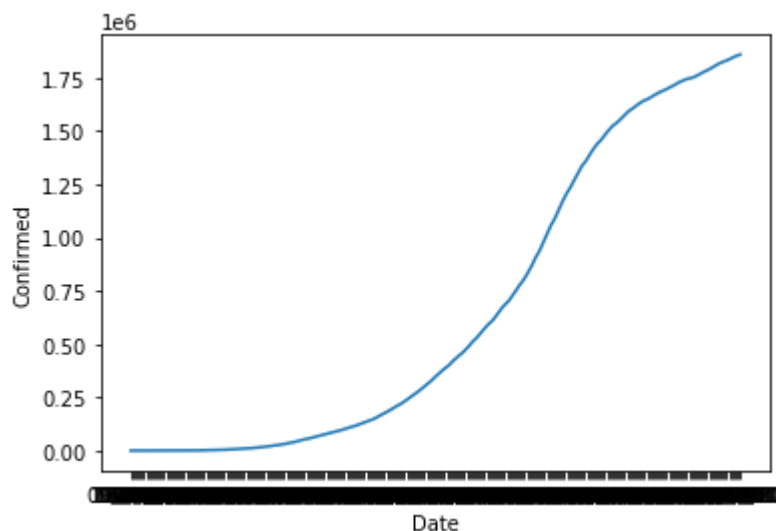
```
x=mh["Date"]
y=mh["Confirmed"]
```

In [21]:

```
sns.lineplot(x,y)
plt.show()
```

C:\Users\Vaishnavi\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

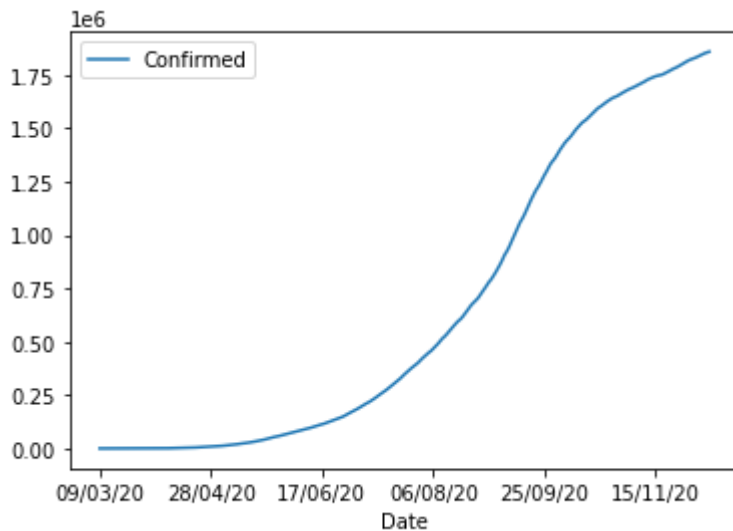


In [22]:

```
mh.plot(x="Date",y="Confirmed")
```

Out[22]:

<AxesSubplot:xlabel='Date'>



Q.4 Plot a pie-chart displaying percentage of total cured and total death cases in Maharashtra.

In [23]:

```
mh["Cured"].sum()
```

Out[23]:

146212339

In [24]:

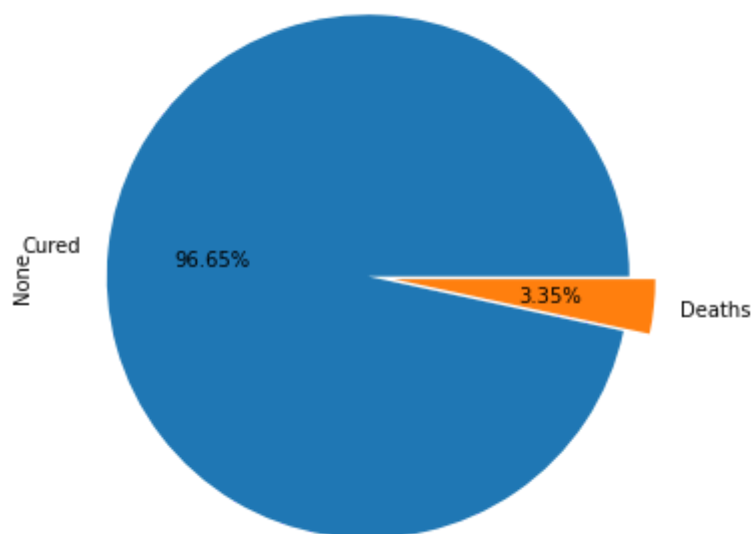
```
mh["Deaths"].sum()
```

Out[24]:

5068405

In [26]:

```
plt.figure(figsize=(6,6))
mh[["Cured","Deaths"]].sum().plot(kind = "pie",autopct = "%1.2f%%",explode=(0,0.1))
plt.show()
```



Q5. Plot a barplot showing top-5 states with maximum number of total confirmed cases.

In [31]:

```
total = df.groupby(["State/UnionTerritory"])["Confirmed"].sum()
```

In [32]:

```
maximum = total.sort_values(ascending=False).head(5)
```

In [33]:

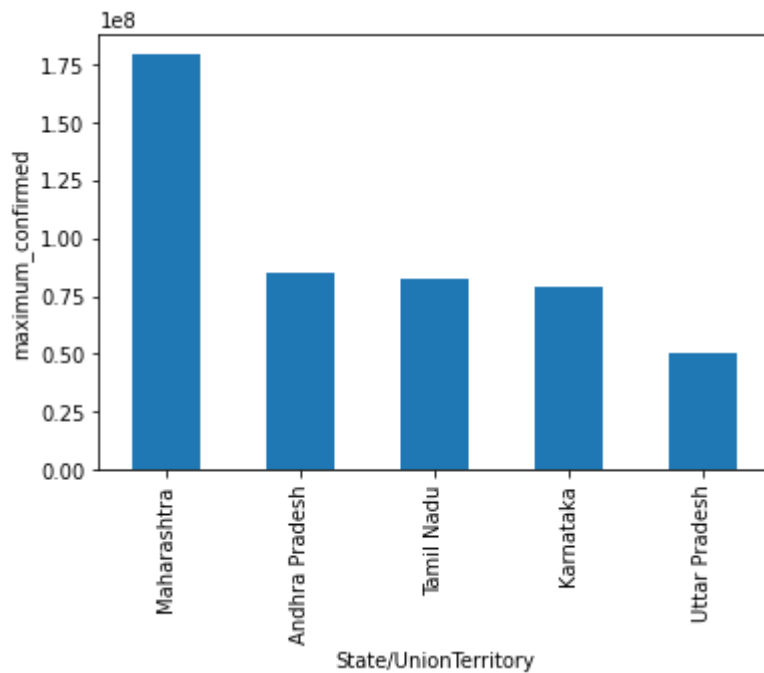
```
maximum
```

Out[33]:

```
State/UnionTerritory
Maharashtra      179450627
Andhra Pradesh   84815303
Tamil Nadu       82789540
Karnataka        79354749
Uttar Pradesh    50283822
Name: Confirmed, dtype: int64
```


In [34]:

```
plt.figure()
maximum.plot(kind="bar")
plt.ylabel("maximum_confirmed")
plt.show()
```



Q.6 Which 3 states have lowest total death cases ?

In [35]:

```
states = df.groupby("State/UnionTerritory")["Deaths"].sum()
```

In [36]:

```
states.sort_values()
```

Out[36]:

State/UnionTerritory	
Unassigned	0
Cases being reassigned to states	0
Dadar Nagar Haveli	0
Daman & Diu	0
Mizoram	159
Chandigarh***	246
Dadra and Nagar Haveli and Daman and Diu	296
Telangana***	455
Telangana***	480
Arunachal Pradesh	3268
Nagaland	3528
Punjab***	4428
Sikkim	5998
Andaman and Nicobar Islands	6420
Meghalaya	7403
Ladakh	7750
Telangana	13342
Manipur	14435
Chandigarh	20455
Himachal Pradesh	32216
Tripura	32265
Maharashtra***	45325
Goa	55207
Puducherry	57565
Assam	85655
Jharkhand	87762
Uttarakhand	90610
Bihar	121993
Odisha	127833
Kerala	132008
Telangana	149324
Jammu and Kashmir	158081
Chhattisgarh	170085
Haryana	194907
Rajasthan	230703
Madhya Pradesh	338597
Punjab	403512
Gujarat	574906
Andhra Pradesh	715536
West Bengal	736550
Uttar Pradesh	756398
Delhi	931396
Karnataka	1144931
Tamil Nadu	1291048
Maharashtra	5068405
Name: Deaths, dtype: int64	

In [38]:

```
states.sort_values().head(3)
```

Out[38]:

```
State/UnionTerritory
Unassigned           0
Cases being reassigned to states  0
Dadar Nagar Haveli   0
Name: Deaths, dtype: int64
```

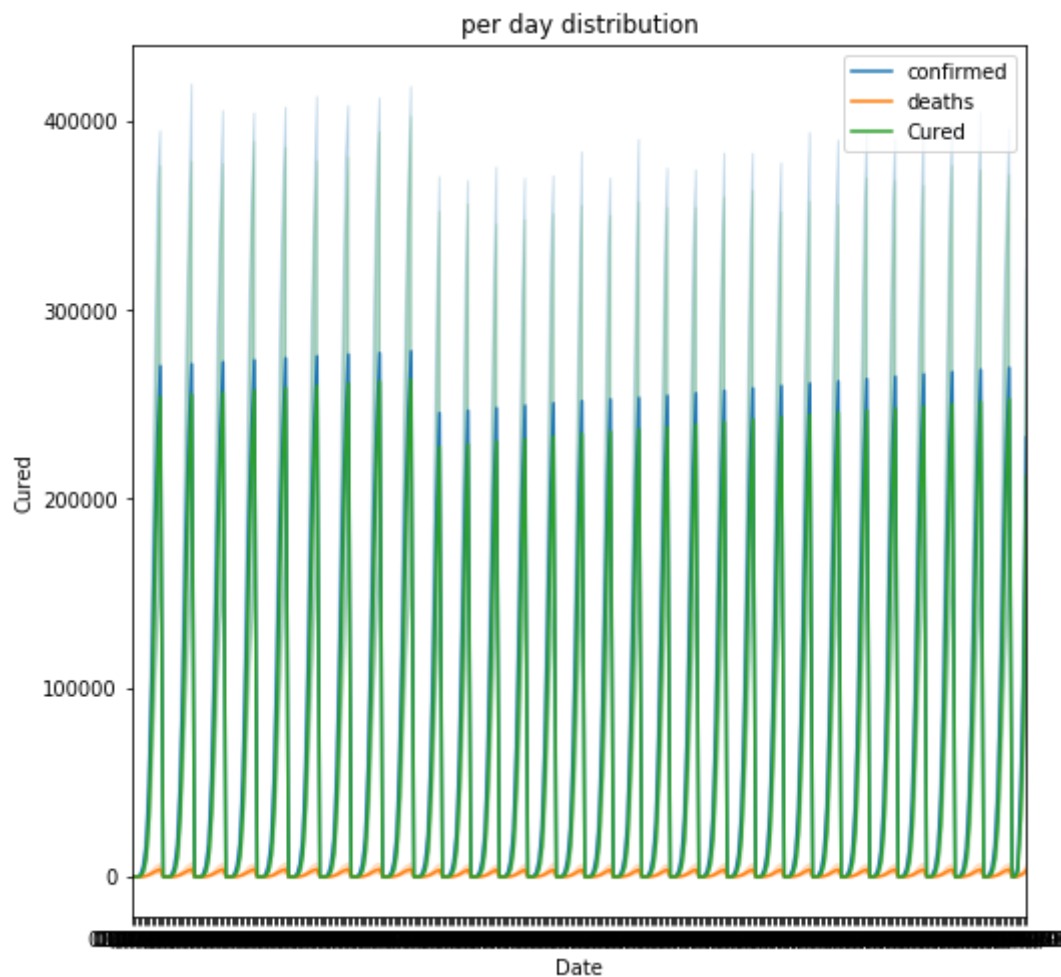
Q7. Plot multi linegraph that shows distribution of per day confirmed cases, death cases and cured cases in India in 2020.

In [32]:

```
x=df["Date"]
y= df["Confirmed"]
z = df["Deaths"]
v = df["Cured"]
```

In [33]:

```
plt.figure(figsize=(8,8))
sns.lineplot(x,y,label="confirmed")
sns.lineplot(x,z,label="deaths")
sns.lineplot(x,v,label="Cured")
plt.xlim((df["Date"].min(),df["Date"].max()))
plt.title("per day distribution")
plt.show()
```



In []: