```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.express as px
import plotly.graph_objects as go
import plotly.offline as pyo
from plotly.offline import iplot
```

```
covid_df = pd.read_csv("D:/Data Analysis Projects/covid_19_india.csv")
```

```
covid_df.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Death	Cinfirmed
0	1	2020-01-30	6:00:00	Kerala	1	0	0	0	1
1	2	2020-02-02	6:00:00	Kerala	1	0	0	0	1
2	3	2020-02-03	6:00:00	Kerala	1	0	0	0	1
3	4	2020-02-04	6:00:00	Kerala	1	0	0	0	1
4	5	2020-02-05	6:00:00	Kerala	1	0	0	0	1

```
covid_df.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex:19830 entries,0 to 19829
Data columns (total 8 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Sno                     19830 non-null  int64
 1   Date                    19830 non-null  object
 2   Time                    19830 non-null  object
 3   State/UnionTerritory    19830 non-null  object
 4   ConfirmedIndianNational 19830 non-null  int64
 5   ConfirmedForeignNational 19830 non-null int64
 6   Cured                   19830 non-null  int64
 7   Death                   19830 non-null  int64
 8   Cinfirmed               19830 non-null  int64
dtypes: int64(6),object(3)
memory usage:1.3 MB
```

```
covid_df.describe()
```

	Sno	Cured	Death	Cinfirmed
count	19830.000000	19830.000000	19830.000000	19830.000000
mean	9915.500000	27875.174052	583.666818	28459.141502
std	5725.552076	72348.239029	3746.828851	73455.879715
min	1.000000	0.000000	0.000000	1.000000
25%	4958.250000	1070.250000	12.000000	1076.250000
50%	9915.500000	5260.500000	96.000000	5297.500000
75%	14872.750000	19400.750000	568.250000	19421.750000
max	19830.000000	510330.000000	52952.000000	511151.000000

```
covid_df.rename(columns={'Sno':'Sno','Date':'Date','Time':'Time','State/UnionTerritory':'State/UnionTerritory','ConfirmedIndianNational':'ConfirmedIndianNational','ConfirmedForeignNational':'ConfirmedForeignNational','Cured':'Cured','Death':'Death','Cinfirmed':'Confirmed'},inplace=True,errors='ignore')
```

```
covid_df.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Death	Confirmed
0	1	2020-01-30	6:00:00	Kerala	1	0	0	0	1
1	2	2020-02-02	6:00:00	Kerala	1	0	0	0	1
2	3	2020-02-03	6:00:00	Kerala	1	0	0	0	1
3	4	2020-02-04	6:00:00	Kerala	1	0	0	0	1
4	5	2020-02-05	6:00:00	Kerala	1	0	0	0	1

```
covid_df['Date'] = pd.to_datetime(covid_df['Date'])
```

```
covid_df.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Death	Confirmed
0	1	2020-01-30	6:00:00	Kerala	1	0	0	0	1
1	2	2020-02-02	6:00:00	Kerala	1	0	0	0	1
2	3	2020-02-03	6:00:00	Kerala	1	0	0	0	1
3	4	2020-02-04	6:00:00	Kerala	1	0	0	0	1
4	5	2020-02-05	6:00:00	Kerala	1	0	0	0	1

```
covid_df['Active'] = covid_df['Confirmed'] - covid_df['Death'] - covid_df['Cured']
covid_df.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Death	Confirmed	Active
0	1	2020-01-30	6:00:00	Kerala	1	0	0	0	1	1
1	2	2020-02-02	6:00:00	Kerala	1	0	0	0	1	1
2	3	2020-02-03	6:00:00	Kerala	1	0	0	0	1	1
3	4	2020-02-04	6:00:00	Kerala	1	0	0	0	1	1
4	5	2020-02-05	6:00:00	Kerala	1	0	0	0	1	1

```
covid_df.tail()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Death	Confirmed	Active
19825	19826	2021-06-27	Calculated	Tamil Nadu	0	0	0	0	0	0
19826	19827	2021-06-27	Calculated	Telangana	0	0	0	0	0	0
19827	19828	2021-06-27	Calculated	Tripura	0	0	0	0	0	0
19828	19829	2021-06-27	Calculated	Uttarakhand	0	0	0	0	0	0
19829	19830	2021-06-27	Calculated	Uttar Pradesh	0	0	0	0	0	0

```
statewise = pd.pivot_table(covid_df, values = ["Confirmed", "Death", "Cured"], index = "State/UnionTerritory",
                           aggfunc = sum)
```

```
statewise["Recovery Rate"] = statewise["Cured"]*100/statewise["Confirmed"]
```

```
statewise["Mortality Rate"] = statewise["Death"]*100/statewise["Confirmed"]
```

```
statewise = statewise.sort_values(by = "Confirmed", ascending = False)
```

```
statewise.style.background_gradient(cmap = "Pastel1")
```

	Confirmed	Cured	Death	Recovery Rate	Mortality Rate
State/UnionTerritory					
Maharashtra	1454642	1243451	43661	85.47	3.000
Kerala	1411151	1344000	5297	95.24	0.375
Karnataka	1341188	1207188	17485	90.000	1.304
Andhra Pradesh	1320168	1307168	7068	99.015	0.535
Tamil Nadu	1307286	1264286	17161	96.710	1.313
Uttar Pradesh	1143000	1143000	0	100.000	0.000
Delhi	1140846	1117846	18375	98.000	1.611
West Bengal	1140000	1140000	0	100.000	0.000
Odisha	1140000	1140000	0	100.000	0.000
Telangana	1140000	1140000	0	100.000	0.000
Bihar	1140000	1140000	0	100.000	0.000
Assam	1140000	1140000	0	100.000	0.000
Chhattisgarh	1140000	1140000	0	100.000	0.000
Gujarat	1140000	1140000	0	100.000	0.000
Haryana	1140000	1140000	0	100.000	0.000
Himachal Pradesh	1140000	1140000	0	100.000	0.000
Jharkhand	1140000	1140000	0	100.000	0.000
Madhya Pradesh	1140000	1140000	0	100.000	0.000
Manipur	1140000	1140000	0	100.000	0.000
Meghalaya	1140000	1140000	0	100.000	0.000
Mizoram	1140000	1140000	0	100.000	0.000
Nagaland	1140000	1140000	0	100.000	0.000
Punjab	1140000	1140000	0	100.000	0.000
Rajasthan	1140000	1140000	0	100.000	0.000
Sikkim	1140000	1140000	0	100.000	0.000
Tripura	1140000	1140000	0	100.000	0.000
Uttarakhand	1140000	1140000	0	100.000	0.000
Andaman and Nicobar Islands	1140000	1140000	0	100.000	0.000
Chandigarh	1140000	1140000	0	100.000	0.000
Dadra and Nagar Haveli	1140000	1140000	0	100.000	0.000
Daman and Diu	1140000	1140000	0	100.000	0.000
Goa	1140000	1140000	0	100.000	0.000
Jammu and Kashmir	1140000	1140000	0	100.000	0.000
Ladakh	1140000	1140000	0	100.000	0.000
Lakshadweep	1140000	1140000	0	100.000	0.000
Pondicherry	1140000	1140000	0	100.000	0.000
Telangana	1140000	1140000	0	100.000	0.000
Tripura	1140000	1140000	0	100.000	0.000
Uttarakhand	1140000	1140000	0	100.000	0.000
Uttar Pradesh	1140000	1140000	0	100.000	0.000

```
#Top 10 Active Cases States
```

```
top_10_active_cases = covid_df.groupby('S/UnionTerritory').max()[['Active Cases', 'Date']].sort_values(by = 'Active Cases',ascending = False).iloc[:10].style.background_gradient(cmap='Reds')
```

```
fig = px.figure(figsize=(8,6))
fig.suptitle("Top 10 Active with Most Active Cases in India", size = 20)
```

```
ax1.ticklabel_format(axis='x',style='plain')
for i in p.patches:
    ax1.text(i.get_width()+.3,i.get_y()+.38,str(round((i.get_width())),2),fontsize=10,color='dimgrey')
ax1.invert_yaxis()
ax1.set_ylabel('')
ax1.set_xlabel('')
ax1.set_title('Top 10 States with Most Active Cases in India',size=20)
sns.despine(left=True,bottom=True)
plt.show()
```

```
#Top 10 States with Highest Deaths
top_10_deaths = covid_df.groupby('S/UnionTerritory').max()[['Deaths', 'Date']].sort_values(by = 'Deaths',ascending = False).iloc[:10].style.background_gradient(cmap='Reds')
fig = px.figure(figsize=(8,6))
fig.suptitle("Top 10 States with Most Deaths in India", size = 20)
ax1.ticklabel_format(axis='x',style='plain')
for i in p.patches:
    ax1.text(i.get_width()+.3,i.get_y()+.38,str(round((i.get_width())),2),fontsize=10,color='dimgrey')
ax1.invert_yaxis()
ax1.set_ylabel('')
ax1.set_xlabel('')
ax1.set_title('Top 10 States with Most Deaths in India',size=20)
sns.despine(left=True,bottom=True)
plt.show()
```

```
#Active Trend
fig = px.figure(figsize = (12,6))
ax = px.lineplot(data=covid_df[covid_df['S/UnionTerritory'].isin(['Maharashtra', 'Karnataka', 'Kerala', 'Tamil Nadu',
                 'Andhra Pradesh', 'Uttar Pradesh', 'Delhi', 'West Bengal', 'Bihar', 'Assam', 'Odisha', 'Telangana'])],
                 x='Date', y='Active', hue='S/UnionTerritory', ci=False)
ax.set_title("Top 6 Affected States on Basis of Active", size=20)
ax.ticklabel_format(axis='y',style='plain')
fig.show()
```

```
covid_df.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Death	Confirmed	Active
0	1	2020-01-30	6:00:00	Kerala	1	0	0	0	1	1
1	2	2020-02-02	6:00:00	Kerala	1	0	0	0	1	1
2	3	2020-02-03	6:00:00	Kerala	1	0	0	0	1	1
3	4	2020-02-04	6:00:00	Kerala	1	0	0	0	1	1
4	5	2020-02-05	6:00:00	Kerala	1	0	0	0	1	1

```
covid_df.replace(inplace = True,
                 {'Dadra and Nagar Haveli':'Dadra and Nagar Haveli and Daman and Diu',
                 'Daman and Diu':'Dadra and Nagar Haveli and Daman and Diu',
                 'Telangana':'Telangana',
                 'Uttarakhand':'Uttarakhand',
                 'Uttar Pradesh':'Uttar Pradesh',
                 'Tripura':'Tripura',
                 'Pondicherry':'Puducherry',
                 'Ladakh':'Ladakh',
                 'Lakshadweep':'Lakshadweep',
                 'Jammu and Kashmir':'Jammu and Kashmir',
                 'Andaman and Nicobar Islands':'Andaman and Nicobar Islands',
                 'Chandigarh':'Chandigarh'})
```

```
covid_df.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex:19830 entries,0 to 19829
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Sno                     19830 non-null  int64
 1   Date                    19830 non-null  object
 2   Time                    19830 non-null  object
 3   State/UnionTerritory    19830 non-null  object
 4   ConfirmedIndianNational 19830 non-null  int64
 5   ConfirmedForeignNational 19830 non-null int64
 6   Cured                   19830 non-null  int64
 7   Death                   19830 non-null  int64
 8   Confirmed               19830 non-null  int64
 9   Active                  19830 non-null  int64
dtypes: int64(7),object(3)
memory usage:1.5 MB
```

```
covid_df.isnull().sum()
```

```
Sno                       0
Date                      0
Time                      0
State/UnionTerritory      0
ConfirmedIndianNational   0
ConfirmedForeignNational  0
Cured                     0
Death                     0
Confirmed                 0
Active                    0
dtype:object
```

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

```
array([Kerala' 'Delhi' 'Rajasthan' 'Confirmed' 'Active' 'Cured' 'Death'
       'Maharashtra' 'Karnataka' 'Andhra Pradesh' 'Tamil Nadu'
       'Uttar Pradesh' 'Haryana' 'Himachal Pradesh' 'Punjab' 'Uttarakhand'
       'Chandigarh' 'Jammu and Kashmir' 'Ladakh' 'Uttar Pradesh'
       'Uttarakhand' 'Odisha' 'Bihar' 'Gujarat' 'Madhya Pradesh'
       'Maharashtra' 'Manipur' 'Mizoram' 'Puducherry' 'Telangana'
       'Andaman and Nicobar Islands' 'Dadra and Nagar Haveli and Daman and Diu'
       'Goa' 'Lakshadweep' 'Tripura' 'Meghalaya' 'Assam' 'Nagaland'
       'Chhattisgarh' 'Sikkim' 'Jharkhand' 'West Bengal' 'Telangana'
       'Tripura' 'Uttarakhand' 'Uttar Pradesh' 'Ladakh' 'Lakshadweep'
       'Jammu and Kashmir' 'Andaman and Nicobar Islands' 'Chandigarh'
       'Dadra and Nagar Haveli and Daman and Diu' 'Goa' 'Puducherry'
       'Telangana' 'Tripura' 'Uttarakhand' 'Uttar Pradesh' 'Ladakh'
       'Lakshadweep' 'Jammu and Kashmir' 'Andaman and Nicobar Islands'
       'Chandigarh' 'Dadra and Nagar Haveli and Daman and Diu' 'Goa'
       'Puducherry' 'Telangana' 'Tripura' 'Uttarakhand' 'Uttar Pradesh'
       'Ladakh' 
```