TITLE : GROUP A : BDA LAB ASSIGNMENT 1

AIM

Working with Dataset-Retrieving and Visualization the required data.

OBJECTIVE

1. Getting data to work with: Download dataset from Kaggle
2. Setting up the working directory.
3. Unpacking the data. Decompress the file locally.
4. Looking at the data. Display the top(10) and bottom(10) of the file.
5. Measuring the length of the data set. Count the number of lines in the file.
6. Encode the categorical data
7. Plot a graph stating the state-wise Covid cases(active/ deceased/ recovered)

The below commands are shown on Covid Dataset from kaggle (do not use the same dataset for the implementation. You may lose points if the same dataset is used .)

```
from google.colab import files
uploaded = files.upload()
```

Choose Files corona.csv
- **corona.csv**(application/vnd.ms-excel) - 658552 bytes, last modified: 8/21/2020 - 100% done

```
Saving corona.csv to corona (2).csv
```

```
import pandas as pd
```

```
df = pd.read_csv('corona.csv', parse_dates=True)
```

```
df.head(10)
```

	Date Announced	Age Bracket	Gender	Detected City	Detected District	Detected State	Current Status	Notes	Contracted from which Patient (Suspected)	Nationality	Type of transmission	Backup Notes
0	30/01/2020	20	F	Thrissur	Thrissur	Kerala	Recovered	Travelled from Wuhan	NaN	India	Imported	NaN
1	02/02/2020	NaN	NaN	Alappuzha	Alappuzha	Kerala	Recovered	Travelled from Wuhan	NaN	India	Imported	Student from Wuhan
2	03/02/2020	NaN	NaN	Kasaragod	Kasaragod	Kerala	Recovered	Travelled from Wuhan	NaN	India	Imported	Student from Wuhan
3	02/03/2020	45	M	East Delhi (Mayur Vihar)	East Delhi	Delhi	Recovered	Travelled from Austria, Italy	NaN	India	Imported	Travel history to Italy and Austria
4	02/03/2020	24	M	Hyderabad	Hyderabad	Telangana	Recovered	Travelled from Dubai to Bangalore on 20th Feb,...	NaN	India	Imported	Travel history to Dubai, Singapore contact
5	03/03/2020	69	M	Jaipur	Italians*	Rajasthan	Recovered	Travelled from Italy	NaN	Italy	Imported	Italian tourist
6	04/03/2020	55	NaN	Gurugram	Italians*	Haryana	Recovered	Travelled from Italy	P6	Italy	Imported	Italian tourist
7	04/03/2020	55	NaN	Gurugram	Italians*	Haryana	Recovered	Travelled from Italy	P6	Italy	Imported	Italian tourist

```
df.tail(10)
```

	Date Announced	Age Bracket	Gender	Detected City	Detected District	Detected State	Current Status	Notes	Contracted from which Patient (Suspected)	Nationality	Type of transmission	Backup Notes
10090	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10091	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10092	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10093	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10094	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10095	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10096	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10097	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10098	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10099	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
df.isna().count()
```

```
Date Announced                               10100
Age Bracket                                  10100
Gender                                       10100
Detected City                                10100
Detected District                            10100
Detected State                               10100
Current Status                               10100
Notes                                        10100
Contracted from which Patient (Suspected)    10100
Nationality                                  10100
Type of transmission                         10100
Backup Notes                                 10100
dtype: int64
```

```
df.columns
```

```
Index(['Date Announced', 'Age Bracket', 'Gender', 'Detected City',
       'Detected District', 'Detected State', 'Current Status', 'Notes',
       'Contracted from which Patient (Suspected)', 'Nationality',
       'Type of transmission', 'Backup Notes'],
      dtype='object')
```

```
df.dropna(axis=0, inplace=True)
```

```
df.tail()
```

	Date Announced	Age Bracket	Gender	Detected City	Detected District	Detected State	Current Status	Notes	Contracted from which Patient (Suspected)	Nationality	Type of transmission	Backup Notes
305	21/03/2020	45	M	Nawanshahr	Shahid Bhagat Singh Nagar	Punjab	Hospitalized	Son of P182	P182	India	Local	Son of P182
306	21/03/2020	40	F	Nawanshahr	Shahid Bhagat Singh Nagar	Punjab	Hospitalized	in Law of P182	P182	India	Local	Daughter in Law of P182
307	21/03/2020	17	M	Nawanshahr	Shahid Bhagat Singh Nagar	Punjab	Hospitalized	Grand daughter of P182	P182	India	Local	Grand daughter of P182
308	21/03/2020	36	F	Nawanshahr	Shahid Bhagat Singh Nagar	Punjab	Hospitalized	Daughter of P182	P182	India	Local	Daughter of P182
310	21/03/2020	60	M	Garhshankar	Hoshiarpur	Punjab	Hospitalized	In contact with	P182	India	Local	In contact with

```
df.shape
```

```
(48, 12)
```

```
df.dtypes
```

```
Date Announced                              object
Age Bracket                                 object
Gender                                      object
Detected City                               object
Detected District                           object
Detected State                              object
Current Status                              object
Notes                                       object
Contracted from which Patient (Suspected)   object
Nationality                                 object
Type of transmission                        object
Backup Notes                                object
dtype: object
```

```
df.drop(['Detected City','Detected District','Notes','Contracted from which Patient (Suspected)','Backup Notes','Nationality','Type of transmission'], axis=1, inplace=True)
```

```
df.set_index('Date  Announced',inplace=True)
df.head(5)
```

	Age Bracket	Gender	Detected State	Current Status
Date Announced				
04/03/2020	70	F	Rajasthan	Recovered
04/03/2020	45	F	Uttar Pradesh	Recovered
04/03/2020	16	M	Uttar Pradesh	Recovered
08/03/2020	54	M	Kerala	Recovered
08/03/2020	53	F	Kerala	Recovered

```
#df = pd.get_dummies(df, columns=['Gender','Detected State','Current Status','Nationality','Type of transmission'], prefix='',prefix_sep='')
df = pd.get_dummies(df, columns=['Current Status'], prefix='',prefix_sep='')
```

```
df.head(3)
```

	Age Bracket	Gender	Detected State	Deceased	Hospitalized	Recovered
Date Announced						
04/03/2020	70	F	Rajasthan	0	0	1
04/03/2020	45	F	Uttar Pradesh	0	0	1
04/03/2020	16	M	Uttar Pradesh	0	0	1

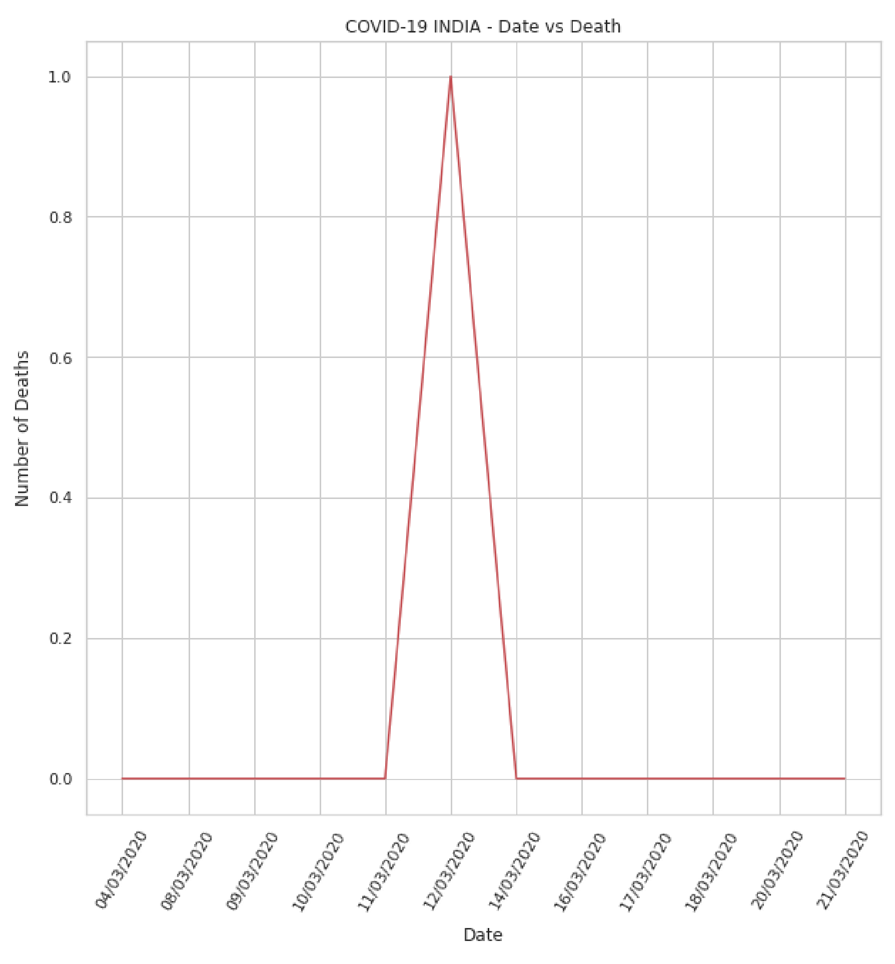
```
dateData = df.groupby(['Date Announced'])['Deceased','Hospitalized','Recovered'].sum().reset_index()
dateData.head()
```

```
stateData  =  df.groupby(['Detected  State'])['Deceased','Hospitalized','Recovered'].sum().reset_index()
stateData.head()
```

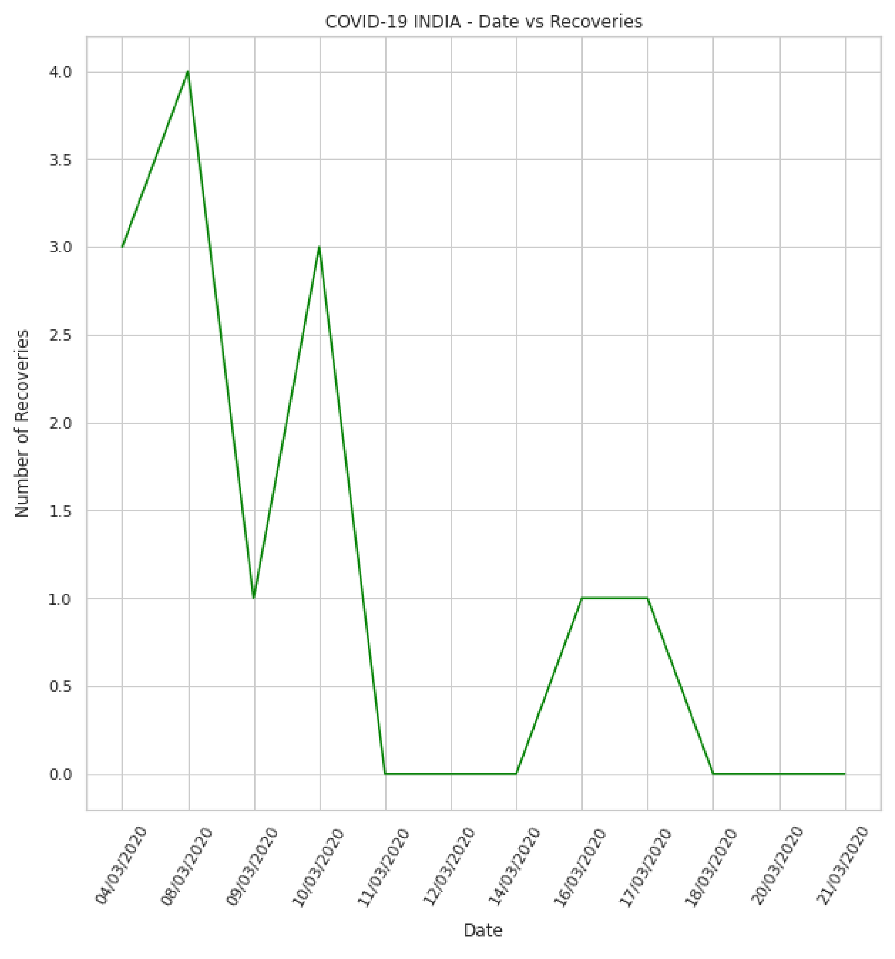
```
ageData = df.groupby(['Age Bracket'])['Deceased','Hospitalized','Recovered'].sum().reset_index()
ageData.head()
```

```
import matplotlib.pyplot as plt
```

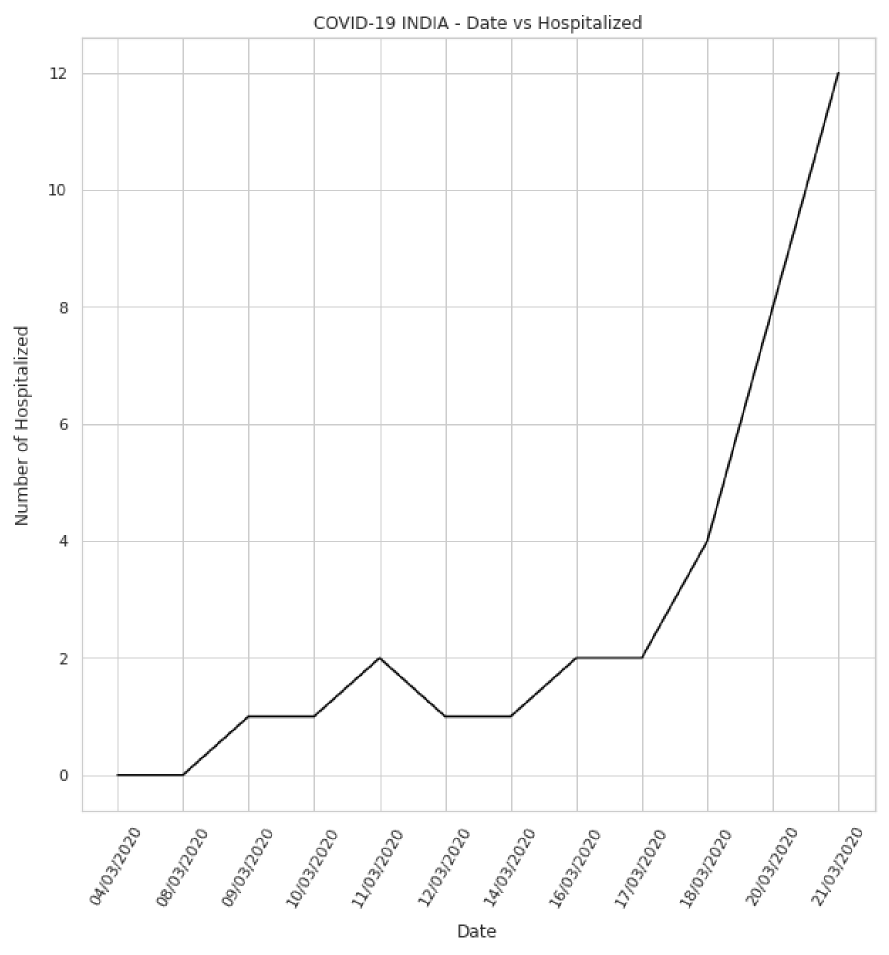
```
plt.figure(figsize=(10,10))
plt.plot(dateData['Date   Announced'],dateData['Deceased'],color='r') plt.title('COVID-
19 INDIA - Date vs Death')
plt.xticks(rotation=60)
plt.xlabel('Date',labelpad=10)
plt.ylabel('Number of Deaths',labelpad=10)
plt.show()
```



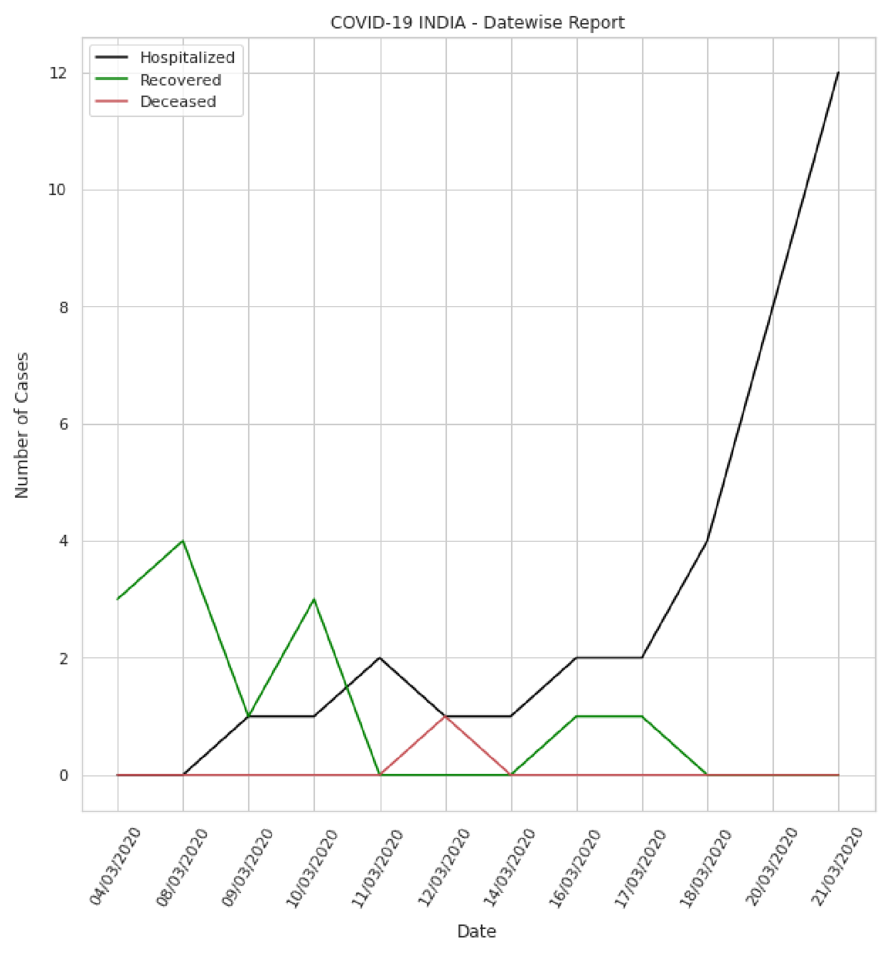
```
plt.figure(figsize=(10,10))
plt.plot(dateData['Date   Announced'],dateData['Recovered'],color='green') plt.title('COVID-
19 INDIA - Date vs Recoveries')
plt.xticks(rotation=60)
plt.xlabel('Date',labelpad=10)
plt.ylabel('Number of Recoveries',labelpad=10)
plt.show()
```



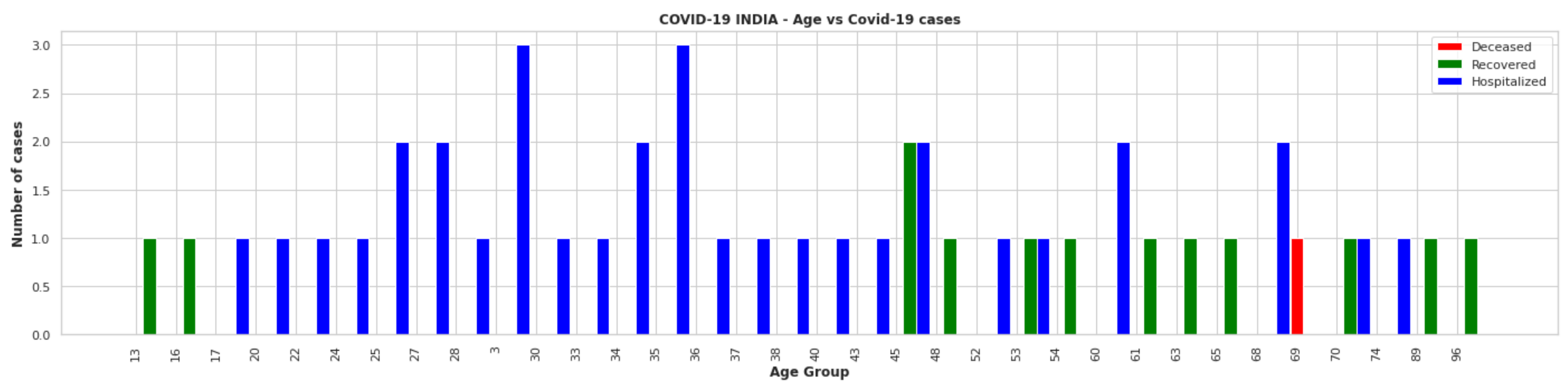
```
plt.figure(figsize=(10,10))
plt.plot(dateData['Date   Announced'],dateData['Hospitalized'],color='black')
plt.title('COVID-19 INDIA - Date vs Hospitalized')
plt.xticks(rotation=60)
plt.xlabel('Date',labelpad=10)
plt.ylabel('Number of Hospitalized',labelpad=10)
plt.show()
```



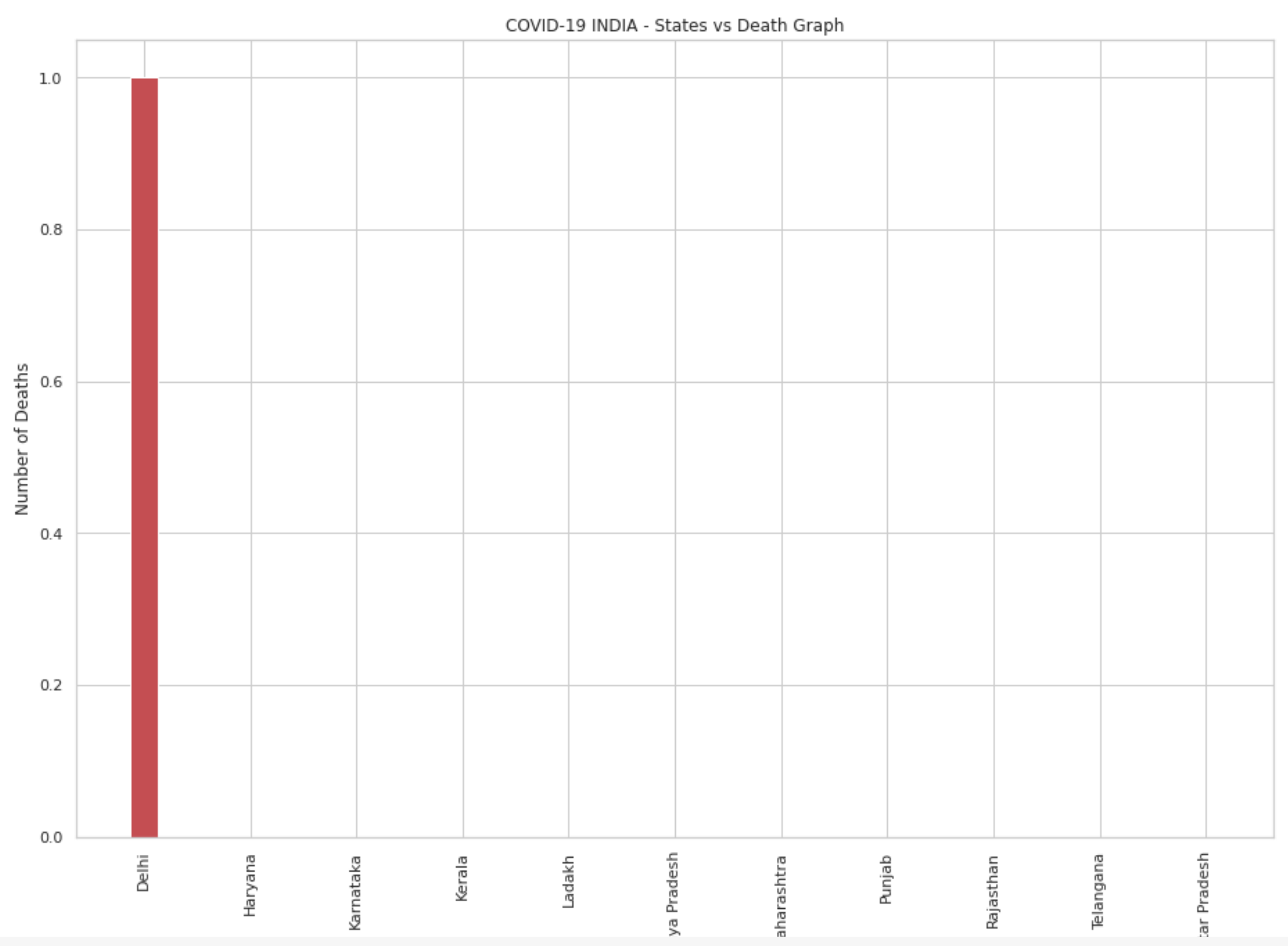
```
plt.figure(figsize=(10,10))
ax  =  plt.plot(dateData['Date  Announced'],dateData['Hospitalized'],color='black',label='Hospitalized')
ax1 = plt.plot(dateData['Date Announced'],dateData['Recovered'],color='green',label='Recovered')
ax2  =plt.plot(dateData['Date  Announced'],dateData['Deceased'],color='r',label='Deceased')
plt.legend()
plt.title('COVID-19 INDIA - Datewise Report')
plt.xticks(rotation=60)
plt.xlabel('Date',labelpad=10)
plt.ylabel('Number of Cases',labelpad=10)
plt.show()
```



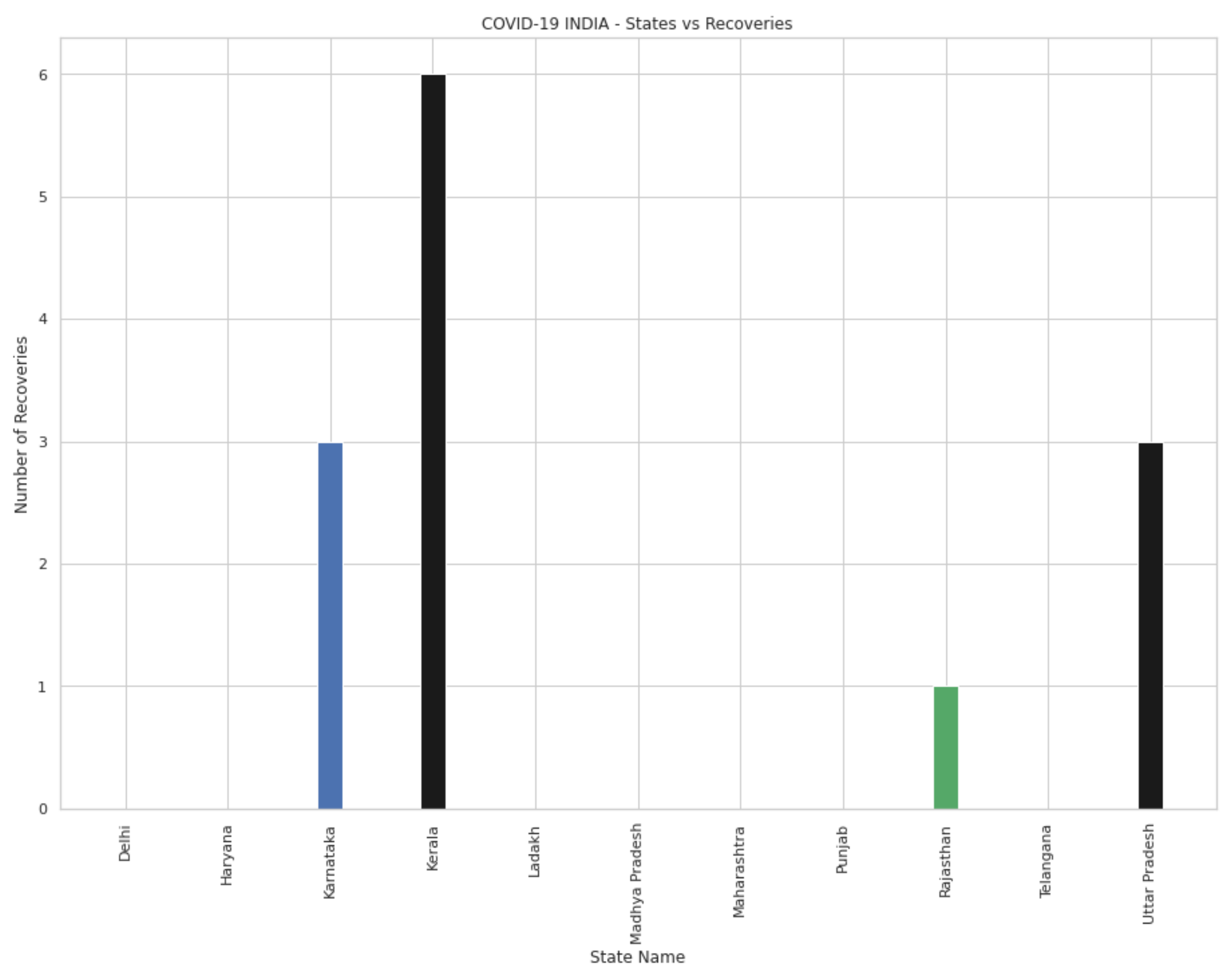
```
barWidth = 0.33

# set height of bar
bars1 = ageData['Deceased']
bars2 = ageData['Recovered']
bars3 = ageData['Hospitalized']

# Set position of bar on X axis
r1 = np.arange(len(bars1))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]

plt.figure(figsize=(24,5))

ax = plt.bar(ageData['Age Bracket'], bars1, color='red', width=barWidth, edgecolor='white', label='Deceased',align='center')
ax1 = plt.bar(r2,bars2, color='green', width=barWidth, edgecolor='white', label='Recovered',align='center')
ax2 = plt.bar(r3,bars3, color='blue', width=barWidth, edgecolor='white', label='Hospitalized',align='center') plt.title('COVID-

19 INDIA - Age vs Covid-19 cases', fontweight='bold')

plt.xlabel('Age Group', fontweight='bold')
plt.ylabel('Number of cases', fontweight='bold')

plt.xticks(rotation = 90)


plt.legend()
plt.show()
```



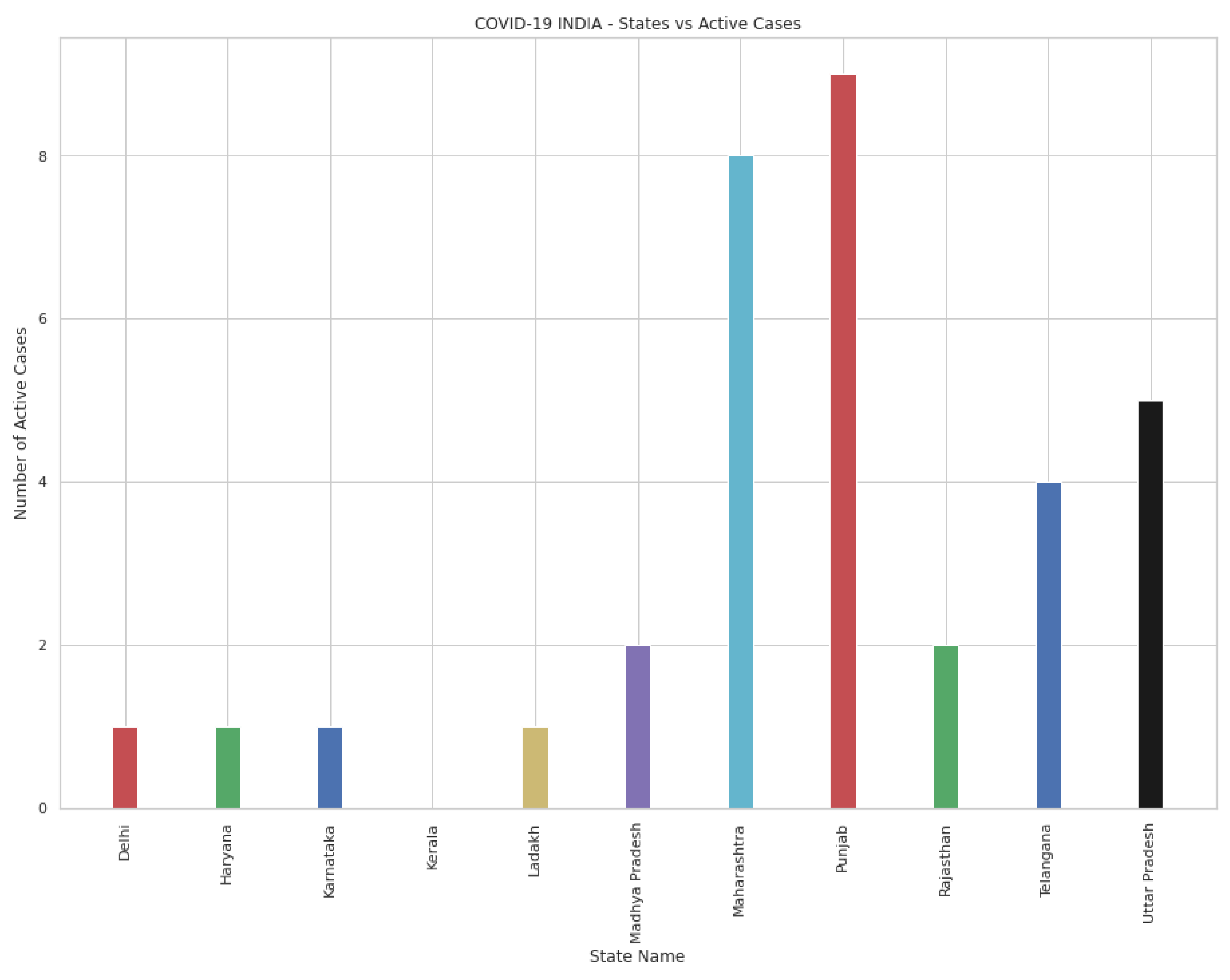
```
fig = plt.figure()
ax = fig.add_axes([0,0,2,2])
ax.bar(stateData['Detected State'],stateData['Deceased'], color = list('rgbkymc'), width = 0.25) plt.title('COVID-
19 INDIA - States vs Death Graph')
plt.xticks(rotation=90)
plt.xlabel('State Name')
plt.ylabel('Number of Deaths')
plt.show()
```



```
fig = plt.figure()
ax = fig.add_axes([0,0,2,2])
ax.bar(stateData['Detected State'],stateData['Recovered'], color = list('rgbkymc'), width = 0.25)
plt.title('COVID-19 INDIA - States vs Recoveries')
plt.xticks(rotation=90)
plt.xlabel('State Name')
plt.ylabel('Number of Recoveries')
plt.show()
```

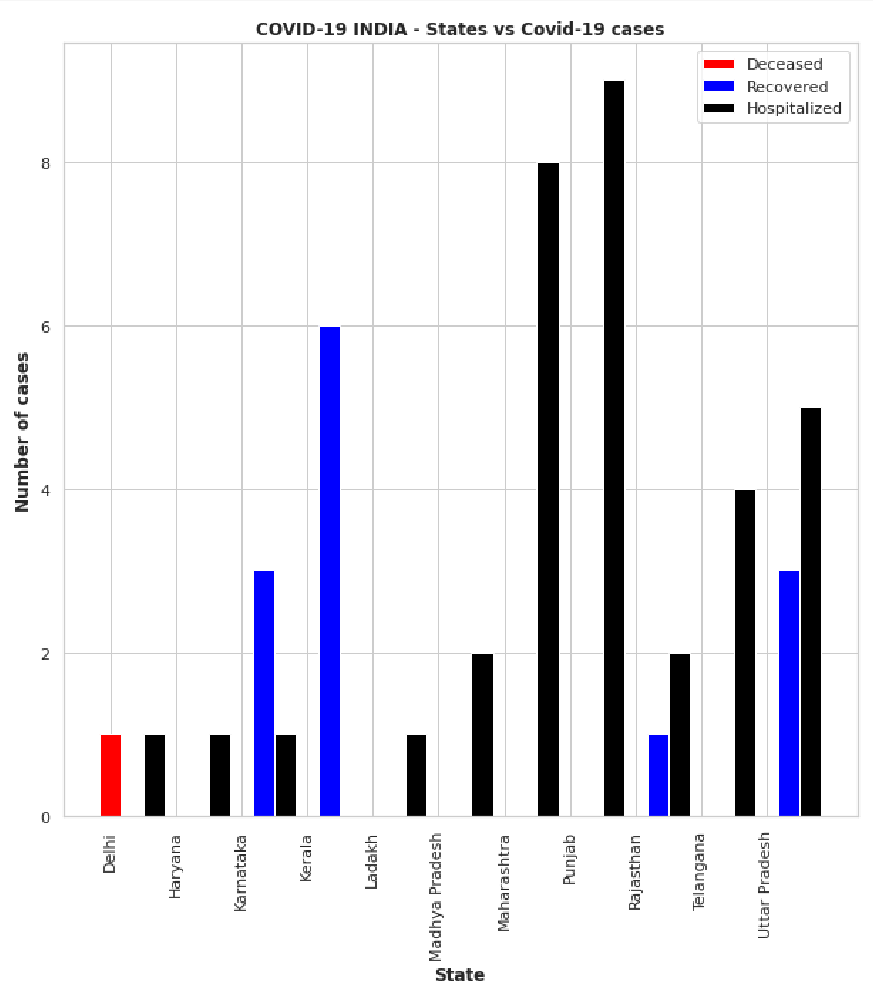


```
fig = plt.figure()
ax = fig.add_axes([0,0,2,2])
ax.bar(stateData['Detected State'],stateData['Hospitalized'], color = list('rgbkymc'), width = 0.25) plt.title('COVID-
19 INDIA - States vs Active Cases')
plt.xticks(rotation=90)
plt.xlabel('State Name')
plt.ylabel('Number of Active Cases')
plt.show()
```

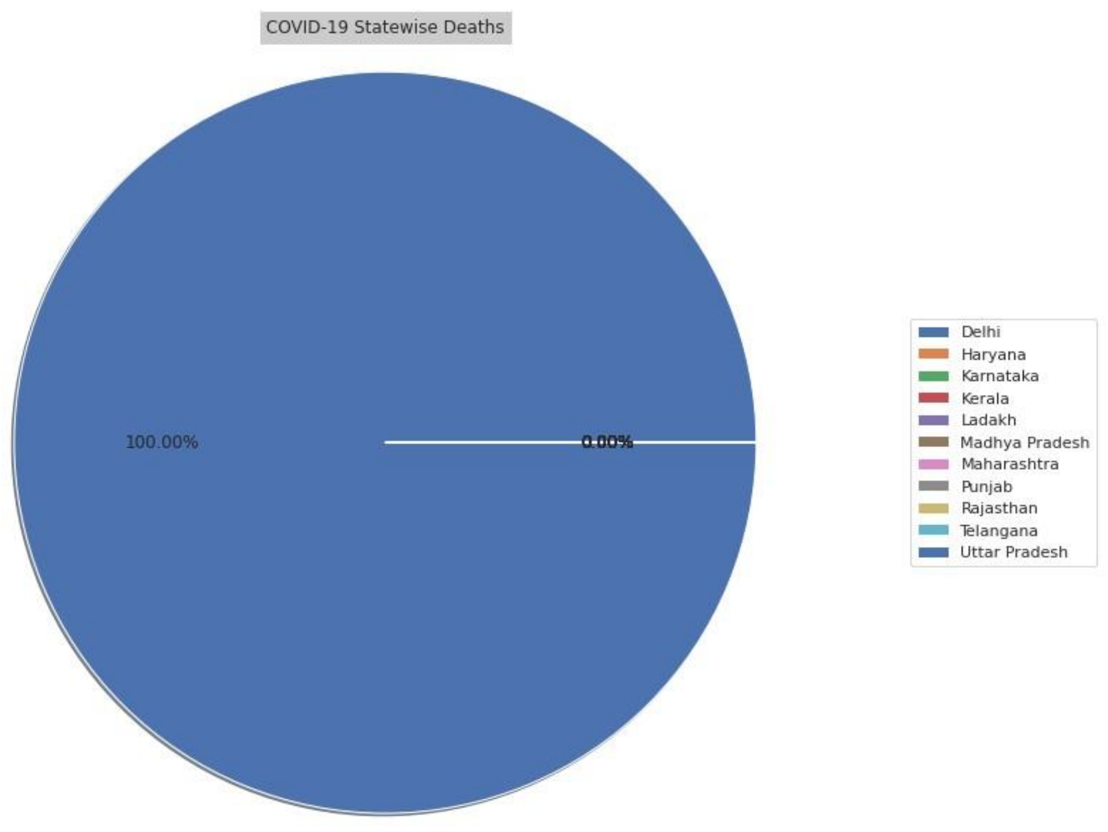


```
barWidth = 0.33

# set height of bar
bars1 = stateData['Deceased']
bars2 = stateData['Recovered']
bars3 = stateData['Hospitalized']

# Set position of bar on X axis
r1 = np.arange(len(bars1))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]

plt.figure(figsize=(10,10))

ax = plt.bar(stateData['Detected State'], bars1, color='red', width=barWidth, edgecolor='white', label='Deceased',align='center')
ax1 = plt.bar(r2,bars2, color='blue', width=barWidth, edgecolor='white', label='Recovered',align='center')
ax2 = plt.bar(r3,bars3, color='black', width=barWidth, edgecolor='white', label='Hospitalized',align='center')
```

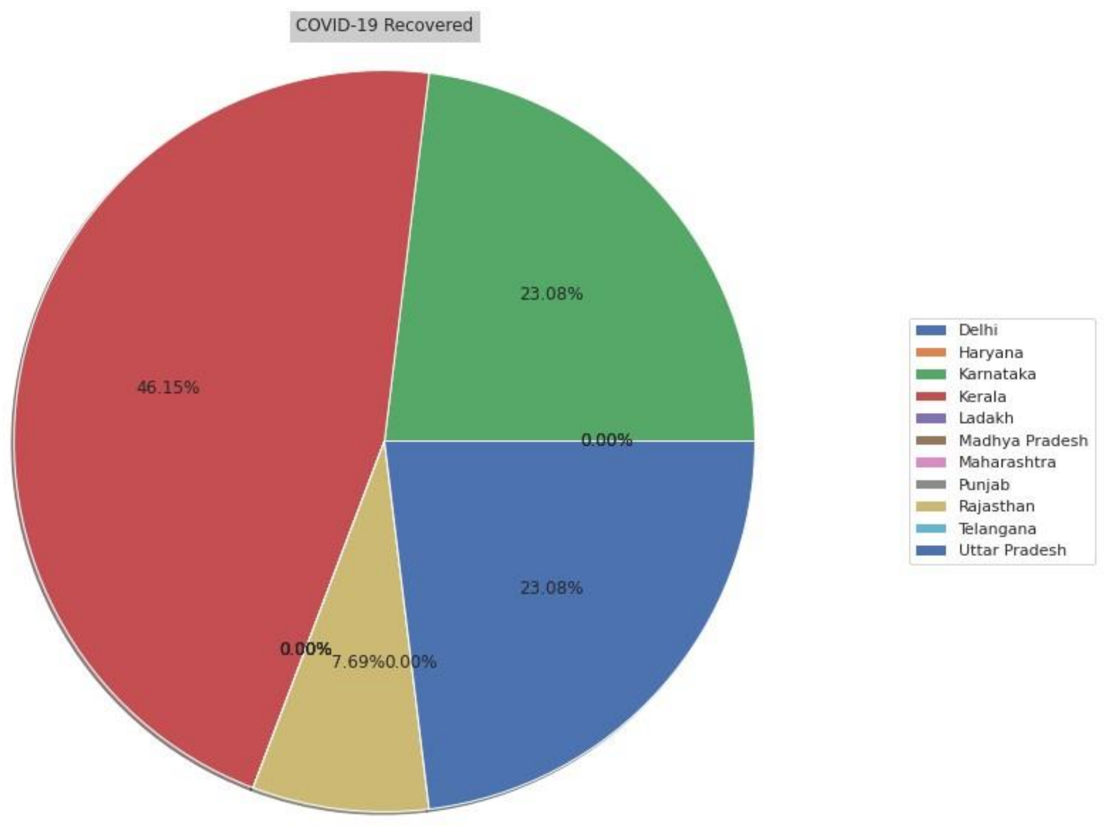
```
plt.title( COVID-19 INDIA - States vs Covid-19 cases , fontweight= bold )

plt.xlabel('State',  fontweight='bold')
plt.ylabel('Number of cases', fontweight='bold')

plt.xticks(rotation = 90)


plt.legend()
plt.show()
```



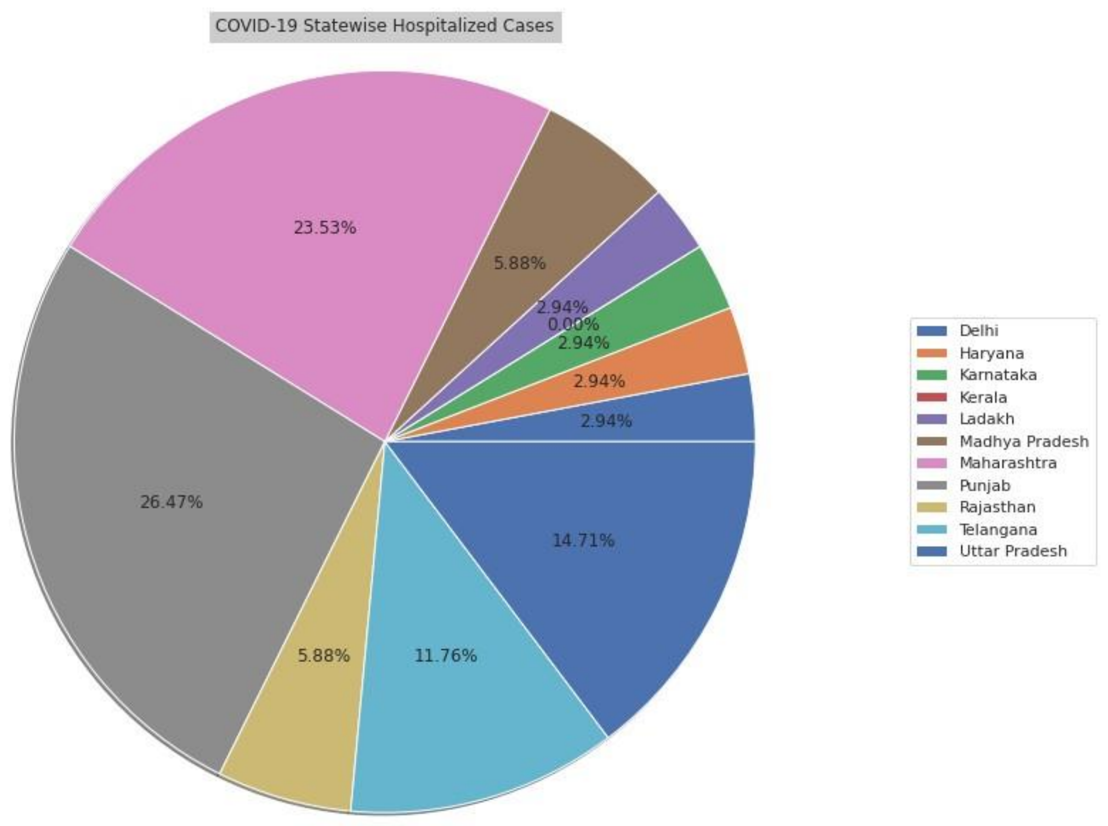
```
state = stateData['Detected State']
cases = stateData['Deceased']
plt.title('COVID-19 Statewise Deaths', bbox={'facecolor':'0.8', 'pad':5}).set_position([.5,1.8])
plt.pie(cases,autopct='%1.2f%%',shadow=True,  radius=3)
plt.legend(state, loc='center',bbox_to_anchor=(2.5, 0.5))
plt.show()
```



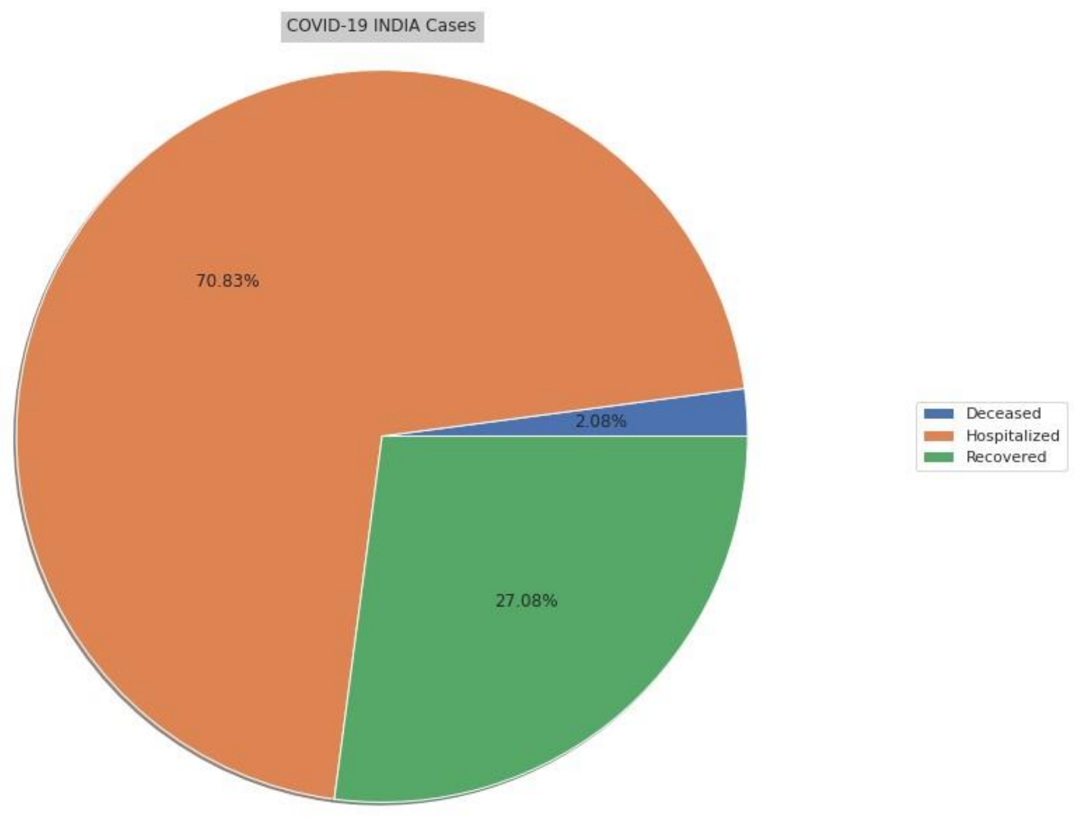
```
state = stateData['Detected State']
cases = stateData['Recovered']
plt.title('COVID-19 Recovered', bbox={'facecolor':'0.8', 'pad':5}).set_position([.5,1.8])
plt.pie(cases,autopct='%1.2f%%',shadow=True,  radius=3)
plt.legend(state, loc='center',bbox_to_anchor=(2.5, 0.5))
plt.show()
```



```
state = stateData['Detected State']
cases = stateData['Hospitalized']
plt.title('COVID-19 Statewise Hospitalized Cases', bbox={'facecolor':'0.8', 'pad':5}).set_position([.5,1.8])
plt.pie(cases,autopct='%1.2f%%',shadow=True,  radius=3)
plt.legend(state, loc='center',bbox_to_anchor=(2.5, 0.5))
plt.show()
```



```
labels = 'Deceased' , 'Hospitalized' , 'Recovered'
sizes = [ ]
for i in range(len(labels)):
  sizes.append(dateData[labels[i]].sum())

plt.title('COVID-19 INDIA Cases', bbox={'facecolor':'0.8', 'pad':5}).set_position([.5,1.8])
plt.pie(sizes,autopct='%1.2f%%',shadow=True,   radius=3,)
plt.legend(labels, loc='center',bbox_to_anchor=(2.5, 0.5))
plt.show()
```


COVID-19 INDIA Cases
Deceased
Hospitalized
Recovered