```python
import os
from requests import request
import urllib.request
import json
from pandas.io.json import json_normalize

import numpy as np
import pandas as pd
import pandas_profiling
from pandas.plotting import register_matplotlib_converters
import seaborn as sns
import matplotlib.pyplot as plt
import folium
import plotly
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots

# color pallette
cnf, dth, rec, act = '#393e46', '#ff2e63', '#21bf73', '#fe9801'

# hide warnings
import warnings
warnings.filterwarnings('ignore')

from IPython.display import Markdown

%matplotlib inline
```

```python
register_matplotlib_converters()
```

```python
def bold(string):
    display(Markdown(string))
```

```python
def read_from_api(URL, x=None):
    """
    Read data from API and Return Normalized JSON

    Keyword arguments:
    URL -- String API URL
    x -- String name to normalize API request into JSON
    """
```

```
        response = request(url=URL, method='get')
        elevations = response.json()
        return json_normalize(elevations) if x==None else json_normalize(elevations[x])


    ''' Function to plot boxplot between two columns '''

    def boxplot(dfname,columnname1,columnname2,plotTitle):
      plt.figure(figsize=(12, 6), dpi = 100)
      sns.boxplot(x = columnname1, y = columnname2, data = dfname, palette = 'viridis')
      plt.title(plotTitle)
      plt.xlabel(columnname1)
      plt.xticks(rotation=90)
      plt.ylabel(columnname2)
      plt.tight_layout()
      plt.show()
      return


    ''' Function to plot countplot between to columns with bins valaues [0,20,30,40,50,60,70,8

    def countplot(columnname1,columnname2,plotTitle):
        bins = [0,20,30,40,50,60,70,80,90,100]
        plt.figure(figsize = (14,8))
        sns.countplot(x=pd.cut(columnname1,bins), hue = columnname2 , orient = 'h')
        plt.xlabel(columnname1.name)
        plt.yscale('log')
        plt.title(plotTitle)
        plt.grid(True)
        plt.show()
        return


    ''' Function to plot pieChart '''

    def pieChart(dfname,columnname, plotTitle):
        fig = px.pie(dfname, values=columnname, names=dfname.index
                 ,color_discrete_sequence=px.colors.sequential.Plasma_r,title=plotTitle)
        fig.update_traces(textposition='outside', textinfo='value+label')
        fig.show()
        return




    ''' Function to plot bar chart'''

    def barChart(dfname , columnname1 , columnname2, plotTitle ,barOrientation):
      fig = px.bar(dfname, x=columnname1, y=columnname2, orientation=barOrientation, text=colu
          color_discrete_sequence = ['#35495e'], title=plotTitle)
      fig.update_xaxes(title=columnname1)
      fig.update_yaxes(title=columnname2)
      fig.show()
      return
```

```
''' Function to plot Histogram Distribution'''

def histogramChart(dfname , columnname , plotTitle):
    fig = px.histogram(dfname, x=columnname, color_discrete_sequence = ['#35495e'], nbins=
    fig.show()
    return
```

```
''' Function to plot Tree Map'''

def treeMapCart(dfname , columnList , valueColumn , plotTitle):
  fig = px.treemap(dfname, path=columnList, values=valueColumn, height=700,
           title=plotTitle, color_discrete_sequence = px.colors.qualitative.Prism)
  fig.data[0].textinfo = 'label+text+value'
  fig.show()
  return
```

```
df_raw_data = read_from_api('https://api.covid19india.org/raw_data.json', 'raw_data')
df_death_and_recoveries = read_from_api('https://api.covid19india.org/deaths_recoveries.js
df_cases_time_series = read_from_api('https://api.covid19india.org/data.json','cases_time_
df_statewise = read_from_api('https://api.covid19india.org/data.json','statewise')
df_tested = read_from_api('https://api.covid19india.org/data.json','tested')
df_district_wise = read_from_api(URL='https://api.covid19india.org/v2/state_district_wise.
df_states_daily = read_from_api('https://api.covid19india.org/states_daily.json','states_d
df_resources = read_from_api('https://api.covid19india.org/resources/resources.json','reso
```

```
bold('**COVID19 - RAW DATA**')
df_raw_data.head()
```

```
<IPython.core.display.Markdown object>
```

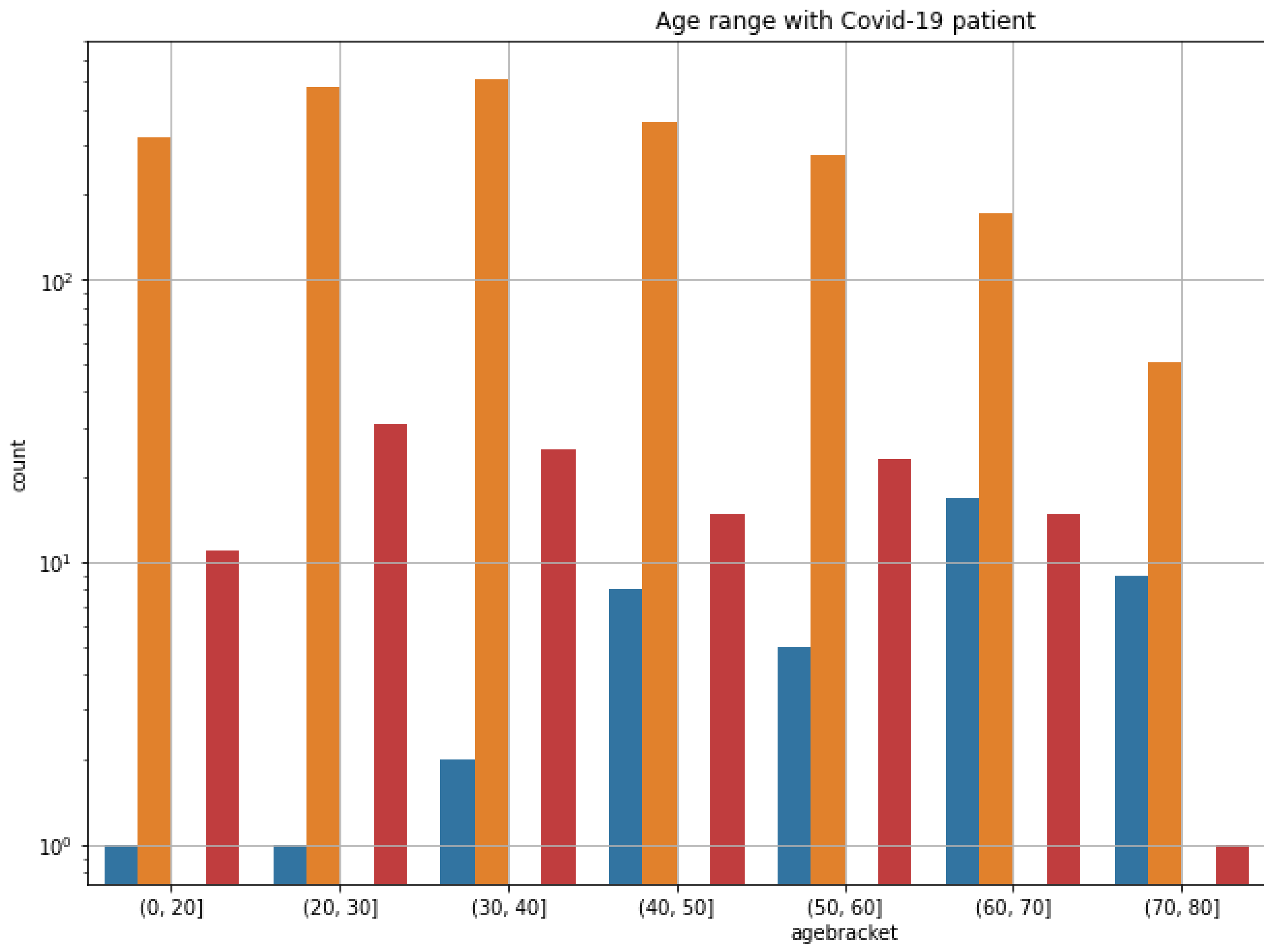
| | agebracket | backupnotes | contractedfromwhichpatientsuspected | currentstatus | datea |
|---|---|---|---|---|---|
| **0** | 20 | Student from Wuhan | | Recovered | 3 |
| **1** | | Student from Wuhan | | Recovered | 0 |
| **2** | | Student from Wuhan | | Recovered | 0 |
| **3** | 45 | Travel history to Italy and Austria | | Recovered | 0 |
| **4** | 24 | Travel history to Dubai, Singapore contact | | Recovered | 0 |

```
df_raw_data['agebracket'] = df_raw_data['agebracket'].replace('28-35', 35)
```

```
 #df_raw_data['agebracket'] = df_raw_data['agebracket'].astype(int)
df_raw_data['agebracket'] = pd.to_numeric(df_raw_data['agebracket'], errors='coerce')
df_raw_data['backupnotes'] = df_raw_data['backupnotes'].astype(str)
df_raw_data['contractedfromwhichpatientsuspected'] = df_raw_data['contractedfromwhichpatie
df_raw_data['currentstatus'] = df_raw_data['currentstatus'].astype('category')
df_raw_data['dateannounced'] = pd.to_datetime(df_raw_data['dateannounced'])
df_raw_data['detectedcity'] = df_raw_data['detectedcity'].astype(str)
df_raw_data['detecteddistrict'] = df_raw_data['detecteddistrict'].astype(str)
df_raw_data['detectedstate'] = df_raw_data['detectedstate'].astype(str)
df_raw_data['gender']= df_raw_data['gender'].astype('category')
df_raw_data['nationality']=df_raw_data['nationality'].astype(str)
df_raw_data['notes']= df_raw_data['notes'].astype('category')
df_raw_data['patientnumber'] = df_raw_data['patientnumber'].astype(int)
df_raw_data['source1']=df_raw_data['source1'].astype(str)
df_raw_data['source2']=df_raw_data['source2'].astype(str)
df_raw_data['source3']=df_raw_data['source3'].astype(str)
df_raw_data['statecode']=df_raw_data['statecode'].astype(str)
df_raw_data['statepatientnumber']=df_raw_data['statepatientnumber'].astype(str)
df_raw_data['statuschangedate']=pd.to_datetime(df_raw_data['statuschangedate'])
df_raw_data['typeoftransmission']=df_raw_data['typeoftransmission'].astype('category')
```

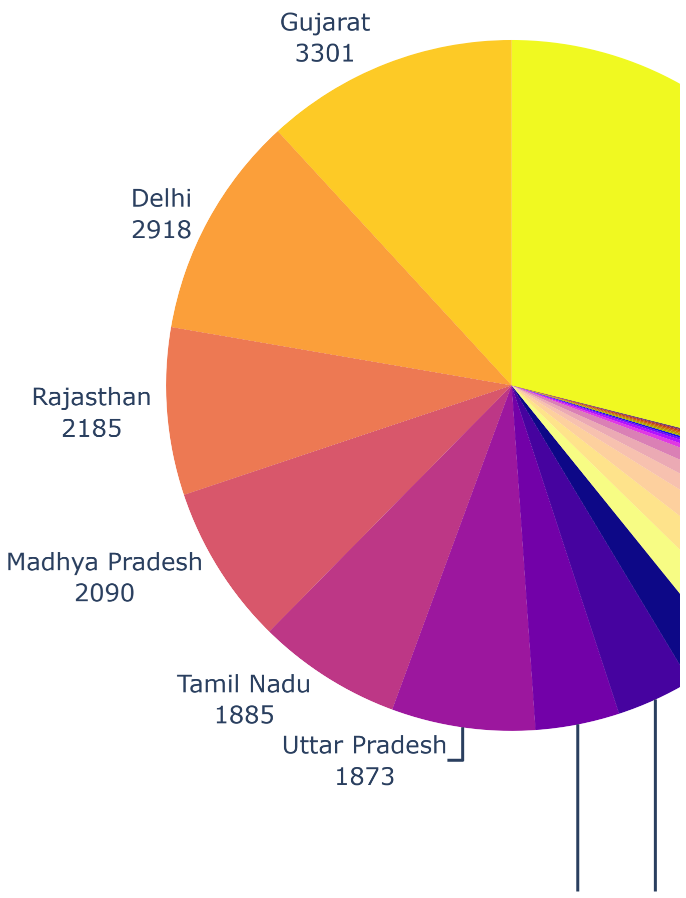
```
df_raw_data['agebracket'] = df_raw_data['agebracket'].replace('28-35', 35)
```

```
countplot(df_raw_data["agebracket"],df_raw_data["currentstatus"],"Age range with Covid-19
```

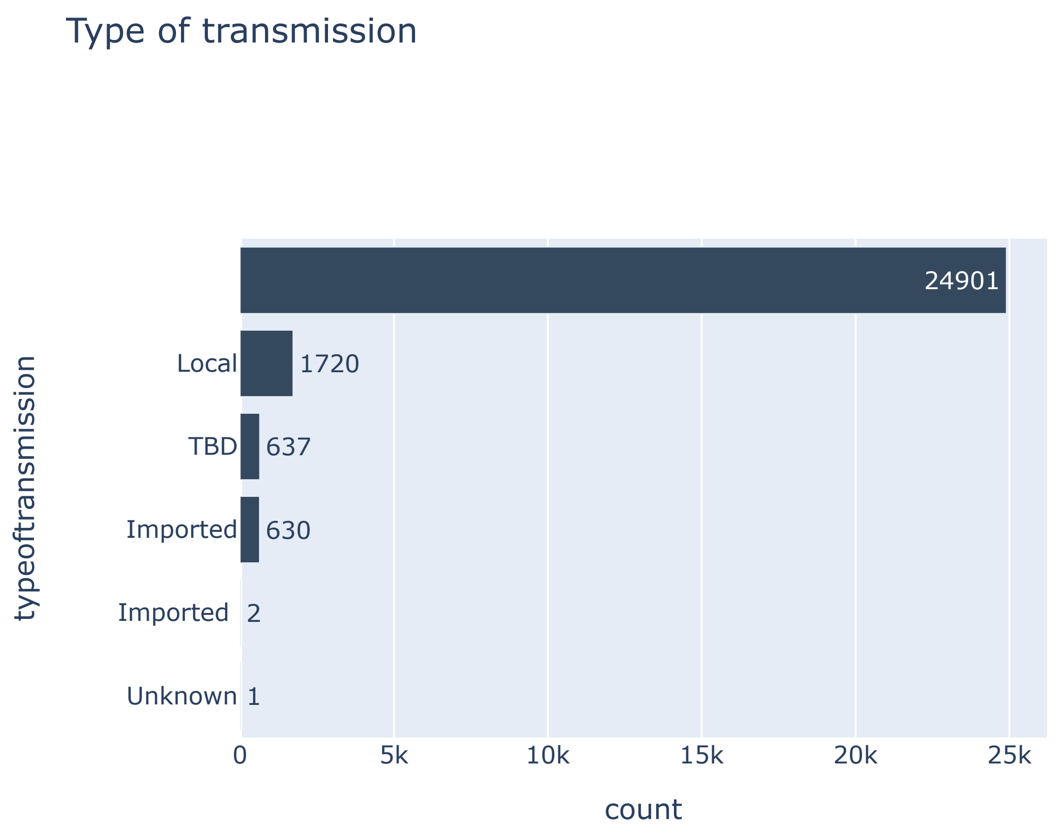


```
state = df_raw_data.groupby('detectedstate').count()
pieChart(state , 'currentstatus' ,'Covid19 cases based on State')
```

Covid19 cases based on State

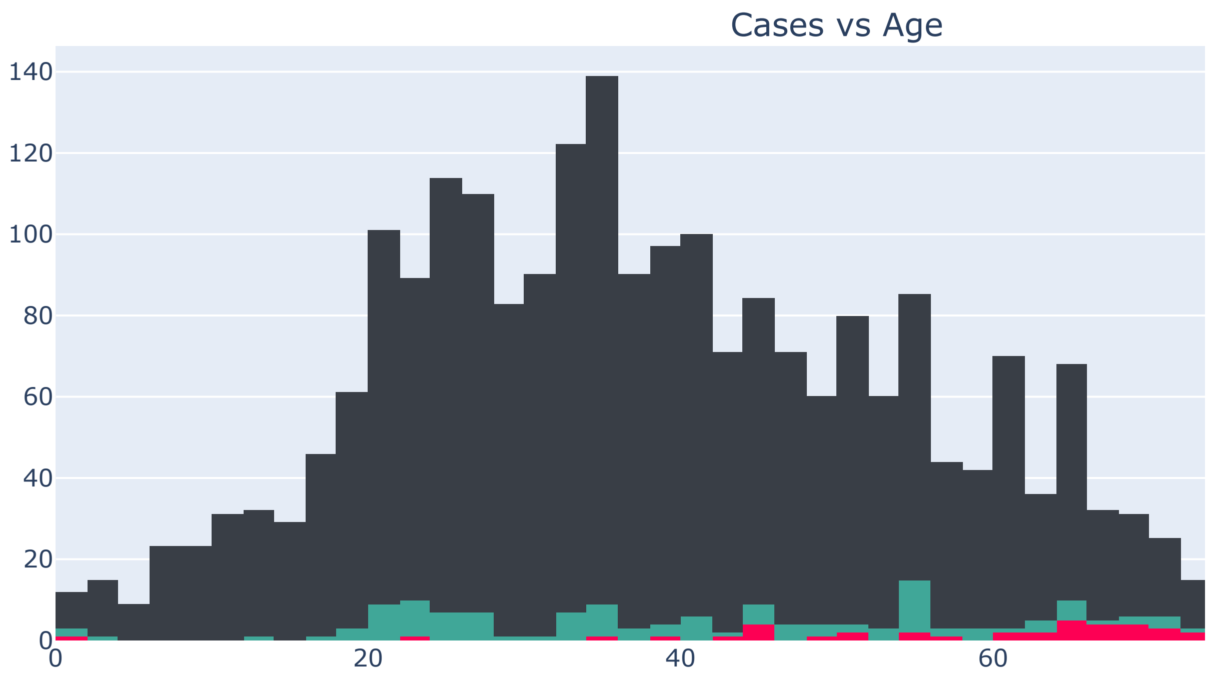


```
temp = pd.DataFrame(df_raw_data[['typeoftransmission']].groupby('typeoftransmission')['typ
temp = temp.dropna()
temp.columns = ['count']
temp = temp.reset_index().sort_values(by='count')

barChart(temp , 'count' , 'typeoftransmission' , 'Type of transmission','h' )
```



```
fig = plotly.subplots.make_subplots(
    rows=1, cols=2, column_widths=[0.8, 0.2],
    subplot_titles = ['Cases vs Age', ''],
    specs=[[{"type": "histogram"}, {"type": "pie"}]]
)

temp = df_raw_data[['agebracket', 'currentstatus']].dropna()
print('Total no. of values :', df_raw_data.shape[0], '\nNo. of missing values :', df_raw_d
gen_grp = temp.groupby('currentstatus').count()

fig.add_trace(go.Pie(values=gen_grp.values.reshape(-1).tolist(), labels=['Deceased', 'Hosp
                     marker_colors = ['#fd0054', '#393e46', '#40a798'], hole=.3),1, 2)

fig.add_trace(go.Histogram(x=temp[temp['currentstatus']=='Deceased']['agebracket'], nbinsx
fig.add_trace(go.Histogram(x=temp[temp['currentstatus']=='Recovered']['agebracket'], nbins
fig.add_trace(go.Histogram(x=temp[temp['currentstatus']=='Hospitalized']['agebracket'], nb

fig.update_layout(showlegend=False)
fig.update_layout(barmode='stack')
fig.data[0].textinfo = 'label+text+value+percent'

fig.show()
```

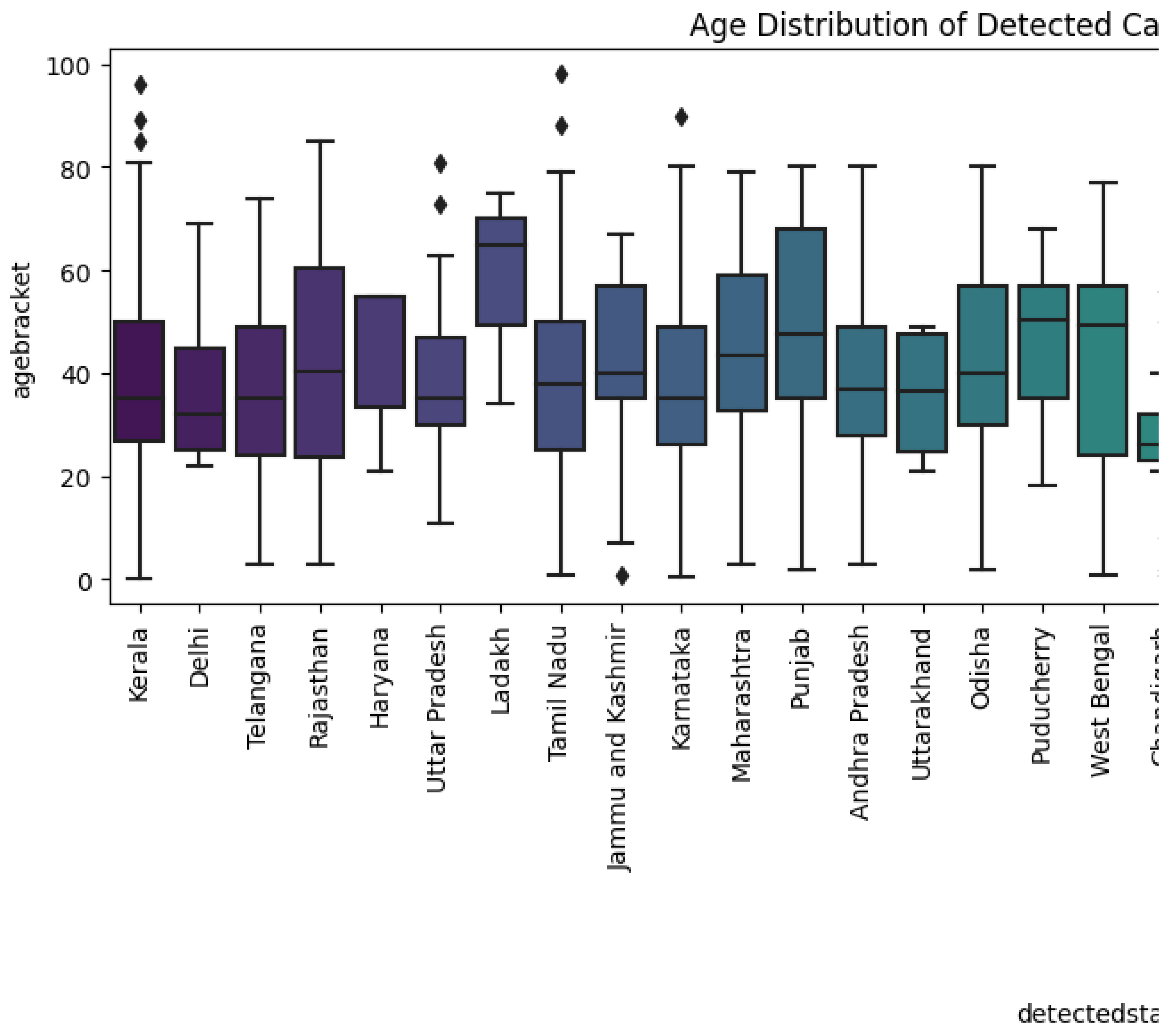
```
Total no. of values : 27891
No. of missing values : 25545
No. of available values : 2346
```



```
''' Function to plot Tree Map'''

def treeMapCart(dfname , columnList , valueColumn , plotTitle):
  fig = px.treemap(dfname, path=columnList, values=valueColumn, height=700,
           title=plotTitle, color_discrete_sequence = px.colors.qualitative.Prism)
  fig.data[0].textinfo = 'label+text+value'
  fig.show()
  return
```

```
statewise_cases = df_statewise[['state','active','confirmed','deaths','recovered']]
statewise_cases = statewise_cases[statewise_cases.state !='Total']
```

```
plt.figure(figsize=(12, 6), dpi = 100)
boxplot(df_raw_data,'detectedstate' , 'agebracket' ,'Age Distribution of Detected Cases ac
<Figure size 1200x600 with 0 Axes>
```



```
df_statewise.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38 entries, 0 to 37
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   active          38 non-null     object
 1   confirmed       38 non-null     object
 2   deaths          38 non-null     object
 3   deltaconfirmed  38 non-null     object
 4   deltadeaths     38 non-null     object
 5   deltarecovered  38 non-null     object
 6   lastupdatedtime 38 non-null     object
 7   recovered       38 non-null     object
 8   state           38 non-null     object
 9   statecode       38 non-null     object
 10  statenotes      38 non-null     object
dtypes: object(11)
memory usage: 3.4+ KB
```

```
df_statewise.head()
```

| | active | confirmed | deaths | deltaconfirmed | deltadeaths | deltarecovered | lastupda |
|---|---|---|---|---|---|---|---|
| **0** | 27749 | 40019 | 1325 | 190 | 2 | 89 | 03/05/2020 |
| **1** | 9775 | 12296 | 521 | 0 | 0 | 0 | 03/05/2020 |
| **2** | 3896 | 5054 | 262 | 0 | 0 | 0 | 02/05/2020 |
| **3** | 2802 | 4122 | 64 | 0 | 0 | 0 | 02/05/2020 |
| **4** | 2013 | 2788 | 151 | 0 | 0 | 0 | 03/05/2020 |

```
print("Data Shape : Rows = {} , Columns = {}".format(df_statewise.shape[0],df_statewise.sh
```

```
Data Shape : Rows = 38 , Columns = 11
```

```
print("Column Names are : \n", df_statewise.columns)
```

```
Column Names are :
 Index(['active', 'confirmed', 'deaths', 'deltaconfirmed', 'deltadeaths',
       'deltarecovered', 'lastupdatedtime', 'recovered', 'state', 'statecode',
       'statenotes'],
      dtype='object')
```

```
cols = ['active', 'confirmed', 'deaths', 'deltaconfirmed', 'deltadeaths',
       'deltarecovered', 'recovered']
```

```
df_statewise['lastupdatedtime'] = pd.to_datetime(df_statewise['lastupdatedtime'])
df_statewise[cols] = df_statewise[cols].astype(int)
df_statewise.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38 entries, 0 to 37
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   active           38 non-null     int64
 1   confirmed        38 non-null     int64
 2   deaths           38 non-null     int64
 3   deltaconfirmed   38 non-null     int64
 4   deltadeaths      38 non-null     int64
 5   deltarecovered   38 non-null     int64
 6   lastupdatedtime  38 non-null     datetime64[ns]
 7   recovered        38 non-null     int64
 8   state            38 non-null     object
 9   statecode        38 non-null     object
 10  statenotes       38 non-null     object
dtypes: datetime64[ns](1), int64(7), object(3)
memory usage: 3.4+ KB
```

```
statewise_cases = df_statewise[['state','active','confirmed','deaths','recovered']]
statewise_cases = statewise_cases[statewise_cases.state !='Total']
```

```
statewise_cases['death_rate (per 100)'] = np.round(100*statewise_cases['deaths']/statewise
statewise_cases.head()
```

| | state | active | confirmed | deaths | recovered | death_rate (per 100) |
|---|---|---|---|---|---|---|
| **1** | Maharashtra | 9775 | 12296 | 521 | 2000 | 4.24 |
| **2** | Gujarat | 3896 | 5054 | 262 | 896 | 5.18 |
| **3** | Delhi | 2802 | 4122 | 64 | 1256 | 1.55 |
| **4** | Madhya Pradesh | 2013 | 2788 | 151 | 624 | 5.42 |
| **5** | Rajasthan | 1489 | 2832 | 70 | 1273 | 2.47 |

```
statewise_cases.dropna(subset=['death_rate (per 100)'], how='all', inplace=True)
```

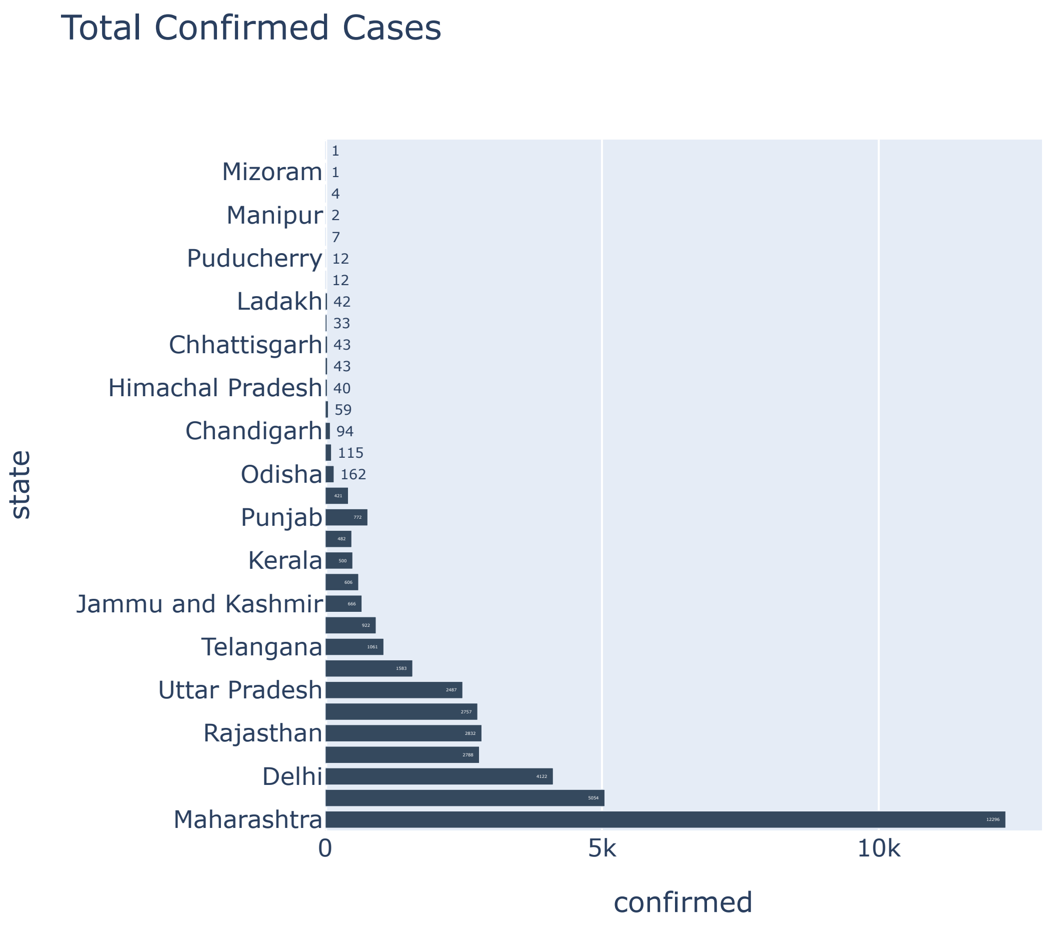
```
print('Total Confirmed Cases: ',statewise_cases['confirmed'].sum())
print('Total Deaths: ',statewise_cases['deaths'].sum())
print('Total Recovered Cases: ',statewise_cases['recovered'].count())
print('Death Rate (per 100): ',np.round(100*statewise_cases['deaths'].sum()/statewise_case
```

```
Total Confirmed Cases:  40019
Total Deaths:  1325
Total Recovered Cases:  32
Death Rate (per 100):  3.31
```

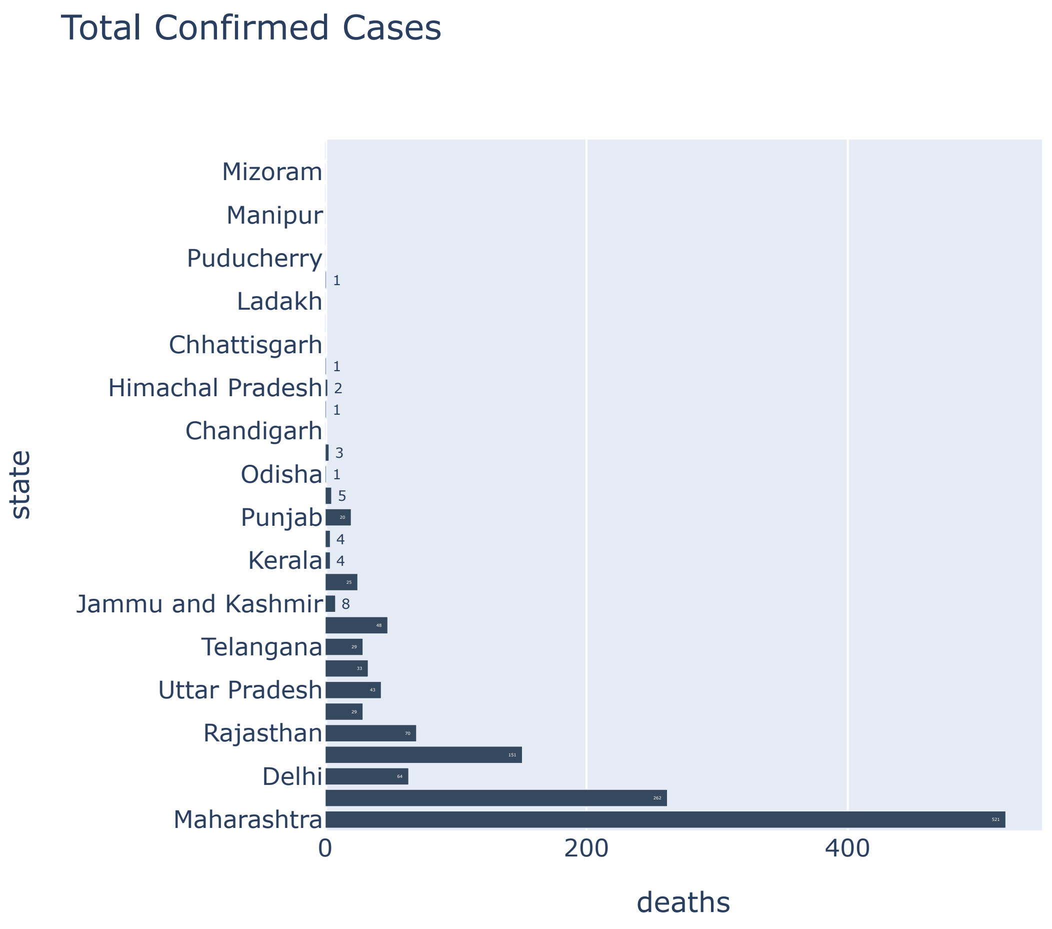
```
#statewise_cases = df_statewise[['state','confirmed','active','recovered','deaths','death_
bold("**STATE WISE CONFIRMED, DEATH AND RECOVERED CASES OF COVID-19**")
statewise_cases.sort_values('confirmed', ascending= False).style.background_gradient(cmap=
                        .background_gradient(cmap='Reds',subset=["deaths"])\
                        .background_gradient(cmap='Greens',subset=["recovered"])\
                        .background_gradient(cmap='Blues',subset=["active"])\
                        .background_gradient(cmap='Purples',subset=["death_rate (per 100)"
<IPython.core.display.Markdown object>
```

| | state | active | confirmed | deaths | recovered | death_rate (per 100) |
|---|---|---|---|---|---|---|
| **1** | Maharashtra | 9775 | 12296 | 521 | 2000 | 4.240000 |
| **2** | Gujarat | 3896 | 5054 | 262 | 896 | 5.180000 |
| **3** | Delhi | 2802 | 4122 | 64 | 1256 | 1.550000 |
| **5** | Rajasthan | 1489 | 2832 | 70 | 1273 | 2.470000 |
| **4** | Madhya Pradesh | 2013 | 2788 | 151 | 624 | 5.420000 |
| **6** | Tamil Nadu | 1387 | 2757 | 29 | 1341 | 1.050000 |
| **7** | Uttar Pradesh | 1746 | 2487 | 43 | 698 | 1.730000 |
| **8** | Andhra Pradesh | 1062 | 1583 | 33 | 488 | 2.080000 |
| **9** | Telangana | 533 | 1061 | 29 | 499 | 2.730000 |
| **10** | West Bengal | 723 | 922 | 48 | 151 | 5.210000 |
| **15** | Punjab | 640 | 772 | 20 | 112 | 2.590000 |
| **11** | Jammu and Kashmir | 404 | 666 | 8 | 254 | 1.200000 |
| **12** | Karnataka | 298 | 606 | 25 | 282 | 4.130000 |
| **13** | Kerala | 96 | 500 | 4 | 400 | 0.800000 |
| **14** | Bihar | 361 | 482 | 4 | 117 | 0.830000 |
| **16** | Haryana | 174 | 421 | 5 | 242 | 1.190000 |
| **17** | Odisha | 105 | 162 | 1 | 56 | 0.620000 |
| **18** | Jharkhand | 90 | 115 | 3 | 22 | 2.610000 |
| **19** | Chandigarh | 75 | 94 | 0 | 19 | 0.000000 |
| **20** | Uttarakhand | 19 | 59 | 1 | 39 | 1.690000 |
| **22** | Assam | 9 | 43 | 1 | 33 | 2.330000 |
| **23** | Chhattisgarh | 7 | 43 | 0 | 36 | 0.000000 |
| **25** | Ladakh | 25 | 42 | 0 | 17 | 0.000000 |
| **21** | Himachal Pradesh | 2 | 40 | 2 | 33 | 5.000000 |
| **24** | Andaman and Nicobar Islands | 7 | 33 | 0 | 26 | 0.000000 |
| **26** | Meghalaya | 1 | 12 | 1 | 10 | 8.330000 |
| **27** | Puducherry | 7 | 12 | 0 | 5 | 0.000000 |
| **28** | Goa | 0 | 7 | 0 | 7 | 0.000000 |
| **30** | Tripura | 2 | 4 | 0 | 2 | 0.000000 |
| **29** | Manipur | 0 | 2 | 0 | 2 | 0.000000 |
| **31** | Mizoram | 1 | 1 | 0 | 0 | 0.000000 |
| **32** | Arunachal Pradesh | 0 | 1 | 0 | 1 | 0.000000 |

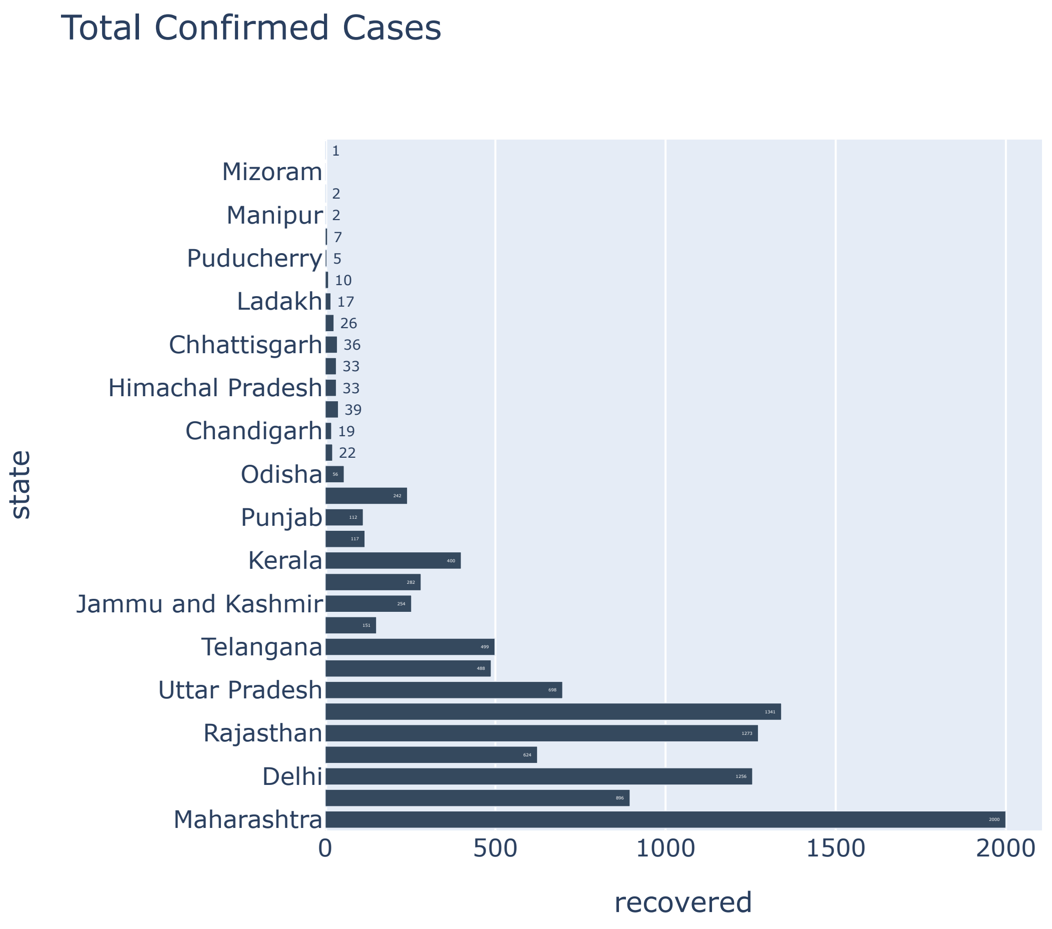
```
barChart(statewise_cases , 'confirmed' , 'state' , 'Total Confirmed Cases' ,'h' )
```



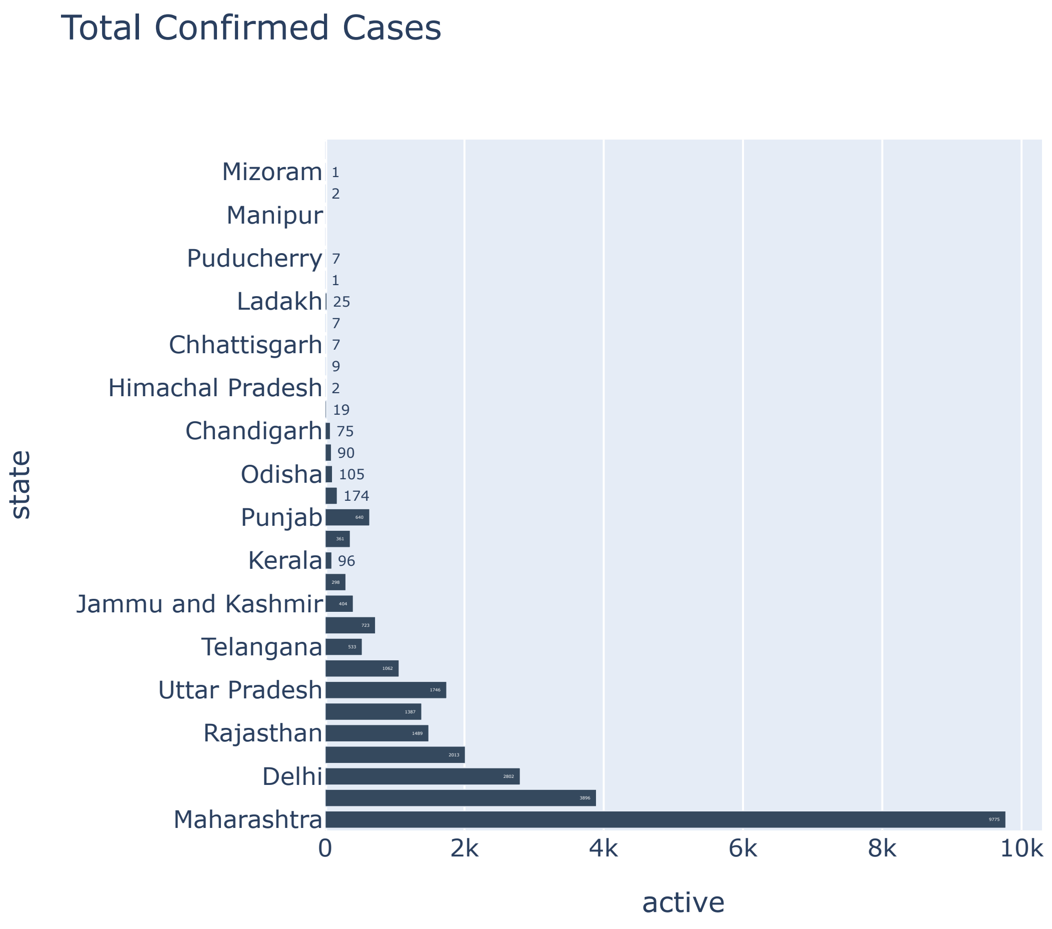
```
barChart(statewise_cases , 'deaths' , 'state' , 'Total Confirmed Cases' ,'h' )
```



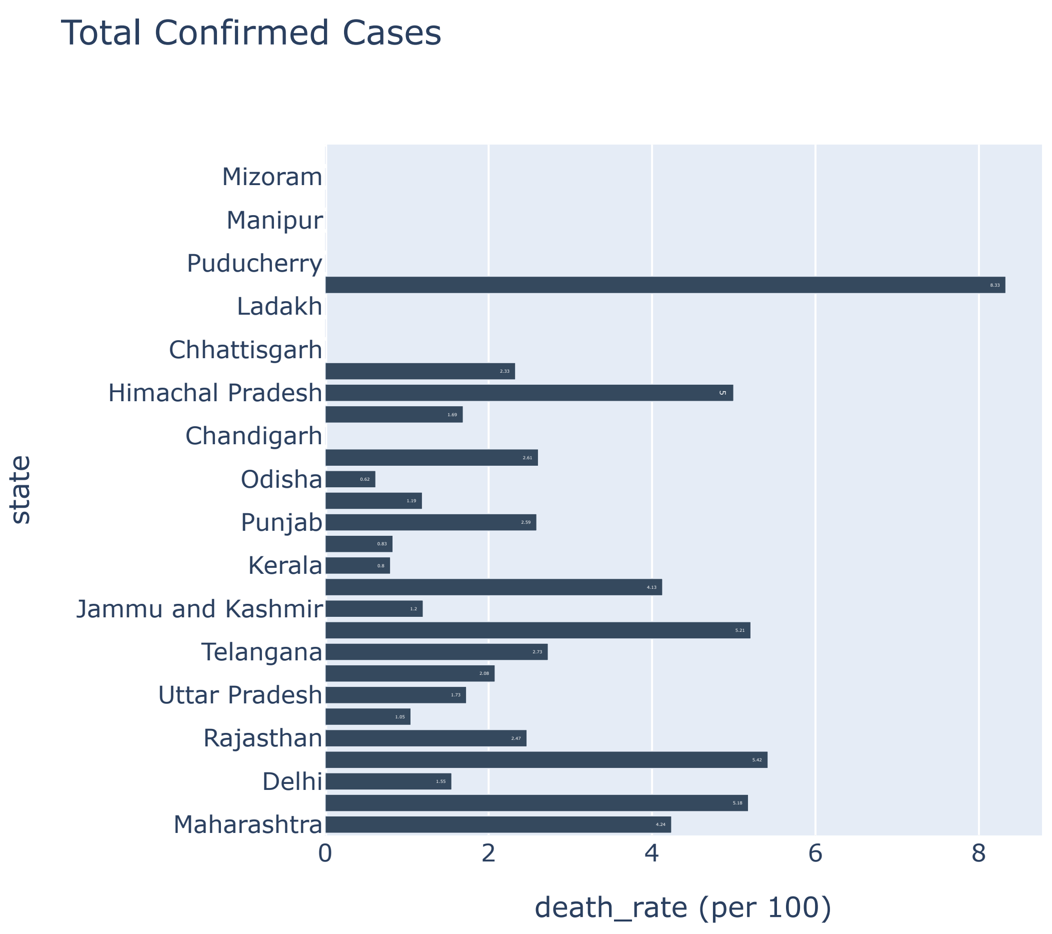
```
barChart(statewise_cases , 'recovered' , 'state' , 'Total Confirmed Cases' ,'h' )
```



```
barChart(statewise_cases , 'active' , 'state' , 'Total Confirmed Cases' ,'h' )
```



```
barChart(statewise_cases , 'death_rate (per 100)' , 'state' , 'Total Confirmed Cases' ,'h'
```



```
def statelat(state):
    lat = {
        "Maharashtra":19.7515,
        "Delhi":28.7041,
        "Tamil Nadu":11.1271,
        "Rajasthan":27.0238,
        "Madhya Pradesh":22.9734,
        "Telangana":18.1124,
        "Gujarat":22.2587,
        "Uttar Pradesh":26.8467,
        "Andhra Pradesh":15.9129,
        "Kerala":10.8505,
        "Jammu and Kashmir":33.7782,
        "Karnataka":15.3173,
        "Haryana":29.0588,
        "Punjab":31.1471,
        "West Bengal":22.9868,
        "Bihar":25.0961,
        "Odisha":20.9517,
        "Uttarakhand":30.0668,
        "Himachal Pradesh":31.1048,
        "Assam":26.2006,
        "Chhattisgarh":22.0797,
        "Chandigarh":30.7333,
        "Jharkhand":23.6102,
        "Ladakh":34.152588,
        "Andaman and Nicobar Islands":11.7401,
        "Goa":15.2993,
        "Puducherry":11.9416,
        "Manipur":24.6637,
        "Tripura":23.9408,
        "Mizoram":23.1645,
        "Arunachal Pradesh":28.2180,
        "Dadra and Nagar Haveli":20.1809,
        "Nagaland":26.1584,
        "Daman and Diu":20.4283,
        "Lakshadweep":8.295441,
        "Meghalaya":25.4670,
        "Sikkim":27.5330
    }
    return lat[state]


def statelong(state):
    long = {
        "Maharashtra":75.7139,
        "Delhi":77.1025,
        "Tamil Nadu":78.6569,
        "Rajasthan":74.2179,
        "Madhya Pradesh":78.6569,
        "Telangana":79.0193,
        "Gujarat":71.1924,
        "Uttar Pradesh":80.9462,
        "Andhra Pradesh":79.7400,
        "Kerala":76.2711,
        "Jammu and Kashmir":76.5762,
        "Karnataka":75.7139,
        "Haryana":76.0856,
        "Punjab":75.3412,
        "West Bengal":87.8550,
        "Bihar":85.3131,
        "Odisha":85.0985,
        "Uttarakhand":79.0193,
        "Himachal Pradesh":77.1734,
        "Assam":92.9376,
        "Chhattisgarh":82.1409,
        "Chandigarh":76.7794,
        "Jharkhand":85.2799,
        "Ladakh":77.577049,
        "Andaman and Nicobar Islands":92.6586,
        "Goa":74.1240,
        "Puducherry":79.8083,
        "Manipur":93.9063,
        "Tripura":91.9882,
        "Mizoram":92.9376,
        "Arunachal Pradesh":94.7278,
        "Dadra and Nagar Haveli":73.0169,
        "Nagaland":94.5624,
        "Daman and Diu":72.8397,
        "Lakshadweep":73.048973,
        "Meghalaya":91.3662,
        "Sikkim":88.5122
    }
    return long[state]
```

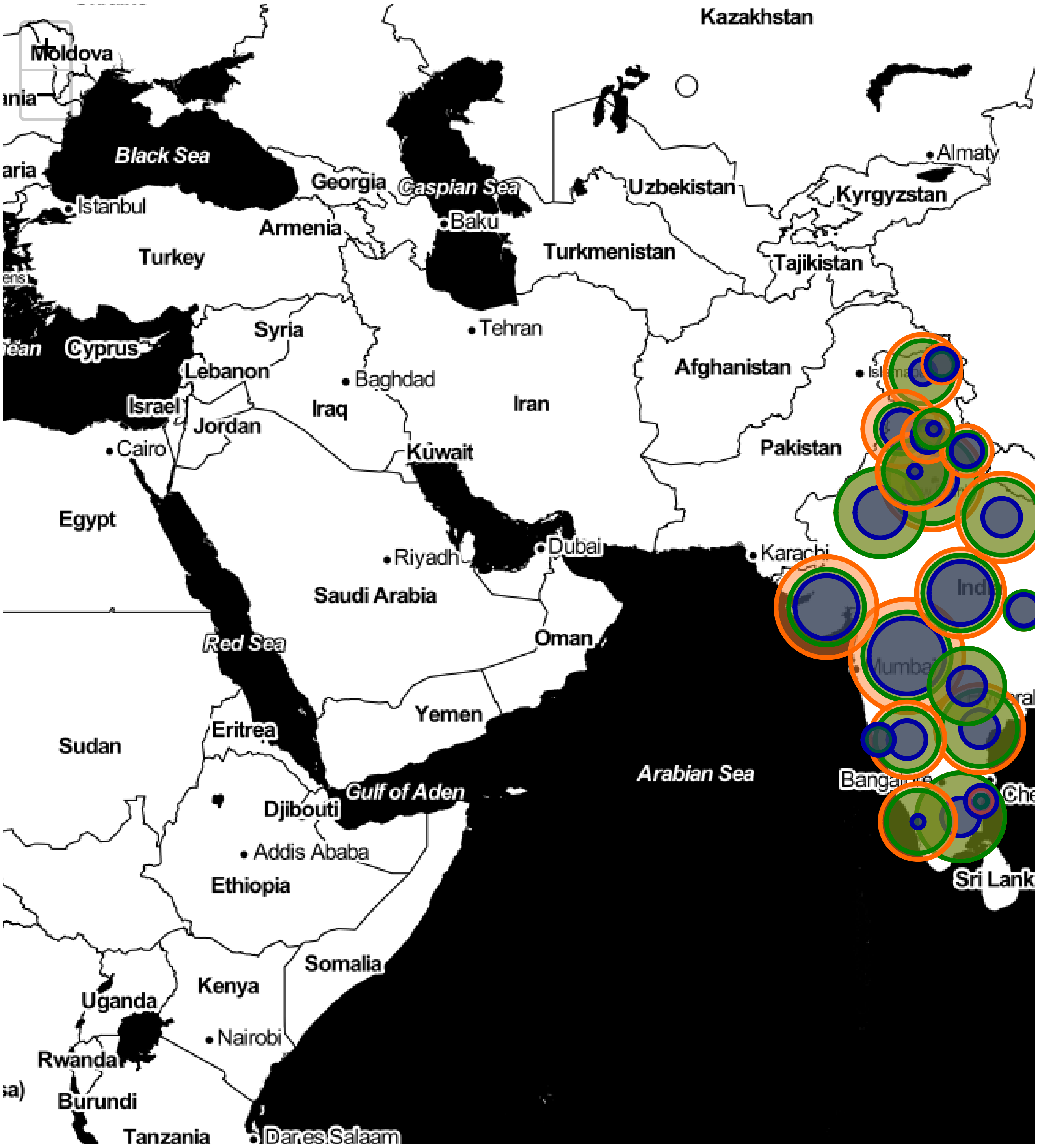
```
len(statewise_cases)
```

```
32
```

```
# states = []
# active = []
# confirmed = []
# deaths = []
# for index in range(len(statewise_cases)):
#     if index == 0:
#         continue
#     states.append(str(re.sub(',','',statewise_cases[index]['state'])))
#     active.append(int(re.sub(',','',statewise_cases[index]['active'])))
#     confirmed.append(int(re.sub(',','',statewise_cases[index]['confirmed'])))
#     deaths.append(int(re.sub(',','',statewise_cases[index]['deaths'])))

a = {'states':list(statewise_cases['state']),
    'lat':list(statewise_cases['state'].apply(lambda x : statelat(x))),
    'long':list(statewise_cases['state'].apply(lambda x : statelong(x))),
    'confirmed':list(statewise_cases['confirmed']),
    'recovered':list(statewise_cases['recovered']),
    'deaths':list(statewise_cases['deaths'])}

df = pd.DataFrame.from_dict(a, orient='index')
dx = df.transpose()
dx.sample(10)

# sates = statewise_cases
# india_map = pd.DataFrame()
# india_map.head()

# india_map['States'] = states
# india_map['lat'] = india_map['States'].apply(lambda x : statelat(x))
# india_map['long'] = india_map['States'].apply(lambda x : statelong(x))
# india_map['Confirmed'] = confirmed
# india_map['Recovered'] = list(np.array(confirmed) - np.array(active))
# india_map['Deaths'] = deaths
```

| | states | lat | long | confirmed | recovered | deaths |
|---|---|---|---|---|---|---|
| **21** | Assam | 26.2006 | 92.9376 | 43 | 33 | 1 |
| **27** | Goa | 15.2993 | 74.124 | 7 | 7 | 0 |
| **3** | Madhya Pradesh | 22.9734 | 78.6569 | 2788 | 624 | 151 |
| **10** | Jammu and Kashmir | 33.7782 | 76.5762 | 666 | 254 | 8 |
| **19** | Uttarakhand | 30.0668 | 79.0193 | 59 | 39 | 1 |
| **7** | Andhra Pradesh | 15.9129 | 79.74 | 1583 | 488 | 33 |
| **14** | Punjab | 31.1471 | 75.3412 | 772 | 112 | 20 |
| **24** | Ladakh | 34.1526 | 77.577 | 42 | 17 | 0 |
| **18** | Chandigarh | 30.7333 | 76.7794 | 94 | 19 | 0 |
| **31** | Arunachal Pradesh | 28.218 | 94.7278 | 1 | 1 | 0 |

```
indiaMap = folium.Map(location=[23,80], tiles="Stamen Toner", zoom_start=4)

for lat, lon, value1,value2,value3, name in zip(dx['lat'], dx['long'], dx['confirmed'],dx[
    folium.CircleMarker([lat, lon],
                        radius= (int((np.log(value1+1.00001))))*4,
                        popup = ('<strong>States</strong>: ' + str(name).capitalize() + '<
                                '<strong>Confirmed Cases</strong>: ' + str(value1) + '<br>
                        color='#ff6600',

                        fill_color='#ff8533',
                        fill_opacity=0.5 ).add_to(indiaMap)
    folium.CircleMarker([lat, lon],
                        radius= (int((np.log(value2+1.00001))))*4,
                        popup = ('<strong>States</strong>: ' + str(name).capitalize() + '<
                                '<strong>Confirmed Recovered</strong>: ' + str(value2) + '
                        color='#008000',

                        fill_color='#008000',
                        fill_opacity=0.4 ).add_to(indiaMap)
    folium.CircleMarker([lat, lon],
                        radius= (int((np.log(value3+1.00001))))*4,
                        popup = ('<strong>States</strong>: ' + str(name).capitalize() + '<
                                '<strong>Confirmed Deaths</strong>: ' + str(value3) + '<br
                        color='#0000A0',

                        fill_color='#0000A0',
                        fill_opacity=0.4 ).add_to(indiaMap)
indiaMap
```



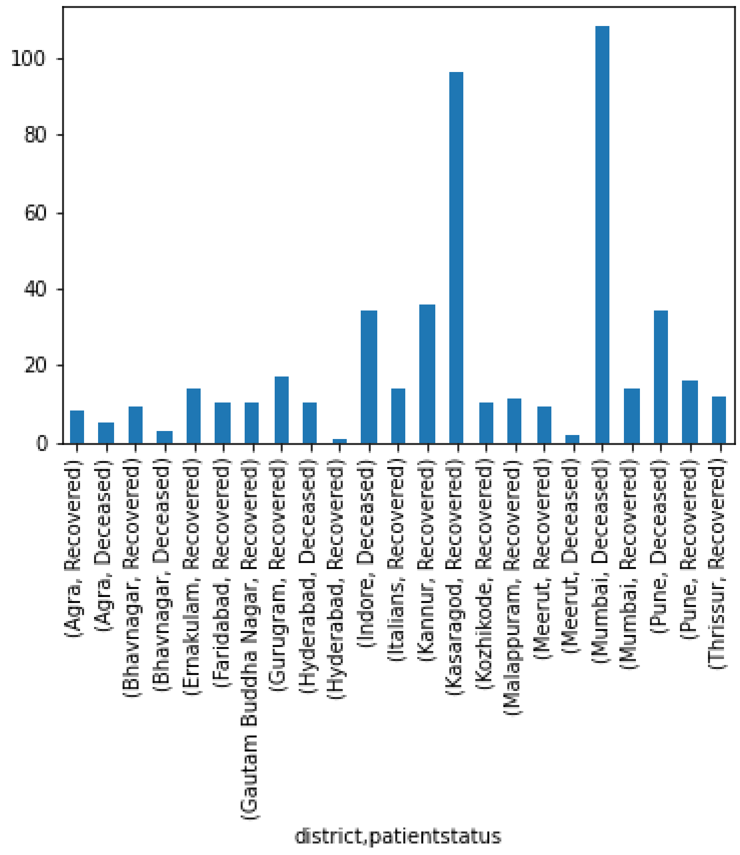
```
df_death_and_recoveries_data['district'].value_counts().sort_values(ascending = False).hea
```

```
                          1429
Mumbai                     122
Kasaragod                   96
Pune                        50
Kannur                      36
Ahmedabad                   35
Indore                      34
Gurugram                    17
Italians                    14
Ernakulam                   14
Agra                        13
Thrissur                    12
Surat                       12
Bhavnagar                   12
Hyderabad                   11
Malappuram                  11
Meerut                      11
Gautam Buddha Nagar         10
Faridabad                   10
Kozhikode                   10
Name: district, dtype: int64
```

```
to_20_district = ['Mumbai','Kasaragod','Pune','Kannur','Ahmadabad','Indore','Gurugram','It
```

```
df_death_and_recoveries_data[df_death_and_recoveries_data['district'].isin(to_20_district)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb095e467f0>
```



```
for i in df_death_and_recoveries_data['state'].unique():
    print ('-----------',i,'-----------')
    print (df_death_and_recoveries_data[df_death_and_recoveries_data['state']==i].groupby(
     ----------- Karnataka -----------
     district         patientstatus
                      Recovered        73
                      Deceased          3
     Bagalkote        Deceased          1
     Belagavi         Deceased          1
     Bengaluru Urban  Recovered         6
                      Deceased          3
     Chikkaballapura  Deceased          1
     Chitradurga      Recovered         1
     Davanagere       Recovered         1
     Gadag            Deceased          1
     Kalaburagi       Deceased          2
     Uttara Kannada   Recovered         1
     Vijayapura       Deceased          1
     Name: patientstatus, dtype: int64
     ----------- Maharashtra -----------
     district         patientstatus
                      Recovered        261
                      Deceased          28
     Ahmednagar       Deceased           1
                      Recovered          1
     Akola            Deceased           1
     Amravati         Deceased           1
     Aurangabad       Deceased           2
                      Recovered          1
     Buldhana         Deceased           1
     Dhule            Deceased           1
     Mumbai           Deceased         108
                      Recovered         14
     Mumbai Suburban  Deceased           1
     Nagpur           Recovered          4
                      Deceased           1
     Nashik           Deceased           1
     Palghar          Deceased           3
     Pune             Deceased          34
                      Recovered         16
     Satara           Deceased           1
     Solapur          Deceased           1
     Thane            Deceased           9
     Yavatmal         Recovered          3
     Name: patientstatus, dtype: int64
     ----------- Kerala -----------
     district            patientstatus
                         Recovered        39
                         Deceased          2
     Alappuzha           Recovered         1
     Ernakulam           Recovered        14
     Idukki              Recovered         7
     Kannur              Recovered        36
     Kasaragod           Recovered        96
     Kollam              Recovered         3
     Kottayam            Recovered         1
     Kozhikode           Recovered        10
     Malappuram          Recovered        11
     Palakkad            Recovered         2
     Pathanamthitta      Recovered         6
     Thiruvananthapuram  Recovered         5
                         Deceased          1
     Thrissur            Recovered        12
     Wayanad             Recovered         2
```

```
Name: patientstatus, dtype: int64
----------- Delhi -----------
district  patientstatus
          Recovered       52
          Deceased        39
Name: patientstatus, dtype: int64
----------- Telangana -----------
district   patientstatus
           Recovered       185
           Deceased          8
Hyderabad  Deceased         10
           Recovered         1
Name: patientstatus, dtype: int64
----------- Gujarat -----------
district     patientstatus
             Recovered       21
             Deceased         2
Ahmedabad    Deceased        20
             Recovered       15
Bhavnagar    Recovered        9
             Deceased         3
Botad        Deceased         1
Gandhinagar  Recovered        4
Gir Somnath  Recovered        1
Jamnagar     Deceased         1
Kutch        Deceased         1
Panchmahal   Deceased         1
Patan        Recovered        4
             Deceased         1
Porbandar    Recovered        2
Rajkot       Recovered        6
Surat        Recovered        7
             Deceased         5
Vadodara     Deceased         6
             Recovered        4
Name: patientstatus, dtype: int64
----------- Tamil Nadu -----------
district      patientstatus
              Recovered        175
              Deceased           3
Chennai       Deceased           7
Coimbatore    Recovered          5
Erode         Deceased           1
Theni         Deceased           1
Thoothukkudi  Deceased           1
Vellore       Deceased           1
Viluppuram    Deceased           1
Name: patientstatus, dtype: int64
----------- Madhya Pradesh -----------
district    patientstatus
            Recovered       66
            Deceased        22
Bhopal      Deceased         4
Chhindwara  Deceased         1
Dewas       Deceased         5
Indore      Deceased        34
Jabalpur    Recovered        4
Khargone    Deceased         1
Ujjain      Deceased         2
Name: patientstatus, dtype: int64
----------- Jammu and Kashmir -----------
district   patientstatus
```

```
           Recovered       38
           Deceased         2
Bandipore  Deceased         1
Baramula   Deceased         1
Udhampur   Deceased         1
Name: patientstatus, dtype: int64
----------- Punjab -----------
district      patientstatus
              Recovered       27
              Deceased         3
Amritsar      Deceased         2
Hoshiarpur    Deceased         1
Jalandhar     Deceased         2
Ludhiana      Deceased         3
Patiala       Recovered        1
Rupnagar      Deceased         1
S.A.S. Nagar  Deceased         2
              Recovered        1
Name: patientstatus, dtype: int64
----------- West Bengal -----------
district           patientstatus
                   Recovered                42
                   Deceased                  7
Howrah             Deceased                  1
                   NotCountedbyAnyState#     1
Kalimpong          Deceased                  1
Kolkata            NotCountedbyAnyState#     1
North 24 Parganas  Deceased                  1
                   NotCountedbyAnyState#     1
Name: patientstatus, dtype: int64
----------- Bihar -----------
district    patientstatus
            Recovered       8
Begusarai   Recovered       1
Bhagalpur   Recovered       1
Gaya        Recovered       3
Gopalganj   Recovered       2
Lakhisarai  Recovered       1
Munger      Recovered       6
            Deceased        1
Nalanda     Recovered       2
Nawada      Recovered       1
Patna       Recovered       5
Saran       Recovered       1
Siwan       Recovered       6
Name: patientstatus, dtype: int64
----------- Himachal Pradesh -----------
district  patientstatus
          Recovered       13
          Deceased         1
Chamba    Recovered        3
Solan     Deceased         1
Name: patientstatus, dtype: int64
----------- Uttar Pradesh -----------
district            patientstatus
                    Recovered       34
Agra                Recovered        8
                    Deceased         5
Basti               Deceased         1
Bulandshahr         Deceased         1
Gautam Buddha Nagar  Recovered      10
```

```
Ghaziabad           Recovered        3
Kanpur Nagar        Deceased         1
Lakhimpur Kheri     Recovered        1
Lucknow             Deceased         1
                    Recovered        1
Meerut              Recovered        9
                    Deceased         2
Moradabad           Deceased         2
Pilibhit            Recovered        1
Shamli              Recovered        1
Varanasi            Deceased         1
Name: patientstatus, dtype: int64
----------- Rajasthan -----------
district  patientstatus
          Recovered       164
          Deceased         13
Bikaner   Deceased          1
Jaipur    Deceased          1
Kota      Deceased          1
Tonk      Deceased          1
Name: patientstatus, dtype: int64
----------- Haryana -----------
district   patientstatus
           Recovered        4
           Deceased         2
Ambala     Recovered        2
Bhiwani    Recovered        2
Faridabad  Recovered       10
Fatehabad  Recovered        1
Gurugram   Recovered       17
Italians   Recovered       14
Karnal     Recovered        3
           Deceased         1
Nuh        Recovered        1
Palwal     Recovered        4
Panchkula  Recovered        2
Panipat    Recovered        3
Sirsa      Recovered        2
Name: patientstatus, dtype: int64
----------- Andhra Pradesh -----------
district        patientstatus
                Recovered       13
Anantapur       Deceased         2
Guntur          Deceased         4
Krishna         Deceased         4
Kurnool         Deceased         1
S.P.S. Nellore  Deceased         2
                Recovered        1
Visakhapatnam   Recovered        6
Y.S.R Kadapa    Deceased         1
Name: patientstatus, dtype: int64
----------- Ladakh -----------
district  patientstatus
          Recovered       12
Kargil    Recovered        1
Leh       Recovered        1
Name: patientstatus, dtype: int64
----------- Uttarakhand -----------
district  patientstatus
          Recovered       9
Name: patientstatus, dtype: int64
----------- Chhattisgarh -----------
```