In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import folium


df = pd.read_csv('case_time_series (2).csv')
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8000\2618737733.py in <module>
      4 import plotly.graph_objects as go
      5 import matplotlib.pyplot as plt
----> 6 import folium
      7
      8

ModuleNotFoundError: No module named 'folium'
```

In [ ]:

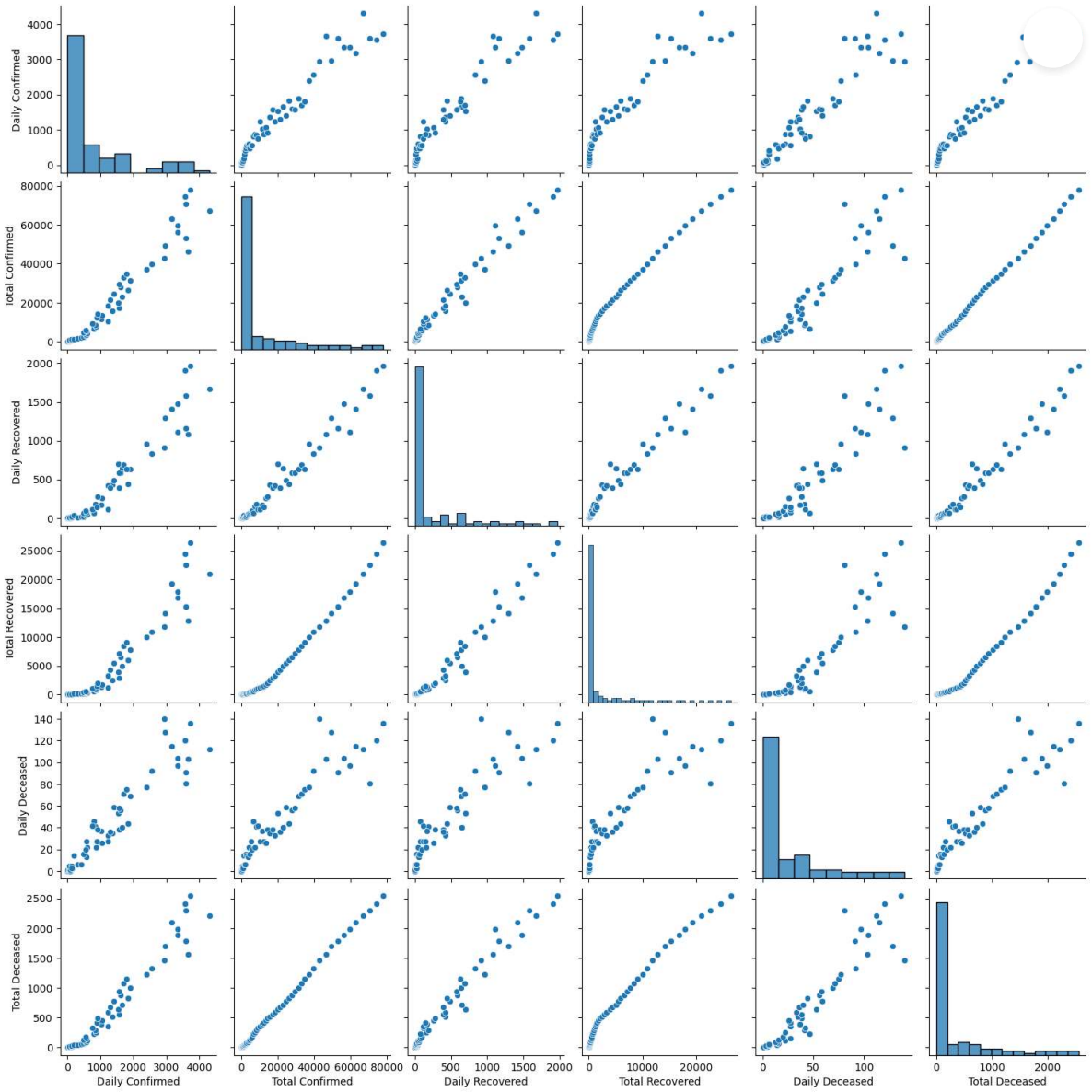
```
df
```

In [3]:

```
sns.pairplot(df)
```

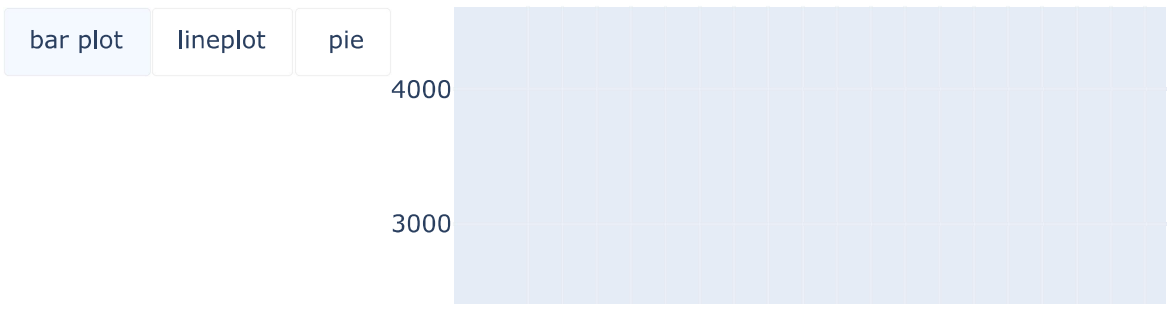
Out[3]:

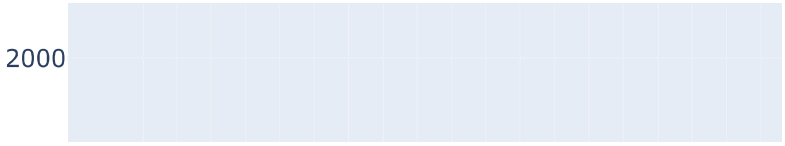
```
<seaborn.axisgrid.PairGrid at 0x196fd744ee0>
```



In [4]:

```
import plotly.graph_objects as go
import pandas as pd

# reading the database
data = pd.read_csv('case_time_series (2).csv')


plot = go.Figure(data=[go.Scatter(
    x=data['Date'],
    y=data['Daily Confirmed'],
    mode='markers',)
])

# Add dropdown
plot.update_layout(
    updatemenus=[
        dict(
            type="buttons",
            direction="left",
            buttons=list([
                dict(
                    args=["type", "bar"],
                    label="bar plot",
                    method="restyle"

                ),
                dict(
                    args=["type", "line"],
                    label="lineplot",
                    method="restyle"
                ),
                dict(
                    args=["type", "pie"],
                    label=" pie",
                    method="restyle"

                ),
            ]),
        ),
    ]
)

plot.show()
```




2000

In [5]:

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

data = pd.read_csv('case_time_series (2).csv')

Y = data.iloc[61:,1].values
R = data.iloc[61:,3].values
D = data.iloc[61:,5].values
X = data.iloc[61:,0]

plt.figure(figsize=(25,8))

ax = plt.axes()
ax.grid(linewidth=0.4, color='#8f8f8f')

ax.set_facecolor("black")
ax.set_xlabel('\nDate',size=25,color='#4bb4f2')
ax.set_ylabel('Number of Confirmed Cases\n',
              size=25,color='#4bb4f2')

plt.xticks(rotation='vertical',size='20',color='red')
plt.yticks(size=20,color='red')
plt.tick_params(size=20,color='red')

for i,j in zip(X,Y):
    ax.annotate(str(j),xy=(i,j+100),color='white',size='13')

ax.annotate('Second Lockdown 15th April',
            xy=(15.2, 860),
            xytext=(19.9,500),
            color='white',
            size='25',
            arrowprops=dict(color='red',
                            linewidth=0.025))

plt.title("COVID-19 IN : Daily Confirmed\n",
         size=50,color='#28a9ff')                    #28a9ff

ax.plot(X,Y,
        color='#1F77B4',
        marker='*',
         linewidth=4,
         markersize=15,
        markeredgecolor='#28a9ff')
```

Out[5]:

```
[<matplotlib.lines.Line2D at 0x1968742c130>]
```

In [6]:

```
data = pd.read_csv('district (1).csv')


data.head()
re=data.iloc[:15,5].values
de=data.iloc[:15,4].values
co=data.iloc[:15,3].values
x=list(data.iloc[:15,0])

plt.figure(figsize=(13,10))
ax=plt.axes()

ax.set_facecolor('black')
ax.grid(linewidth=0.4, color='#8f8f8f')


plt.xticks(rotation='vertical',size='20',color='black')#ticks of X

plt.yticks(size='20',color='black')


ax.set_xlabel('\nDistrict',size=25,color='#4bb4f2')    #17th may 2020

ax.invert_xaxis()

ax.set_ylabel('No. of cases\n',size=25,color='#4bb4f2')


plt.tick_params(size=20,color='white')


ax.set_title('andhrapradesh District wise breakdown\n',size=50,color='#28a9ff')

plt.bar(x,co,label='re')
plt.bar(x,re,label='re',color='green')
plt.bar(x,de,label='re',color='red')

for i,j in zip(x,co):
    ax.annotate(str(int(j)),xy=(i,j+3),color='white',size='15')

plt.legend(['Confirmed','Recovered','Deceased'],
fontsize=20)
```

Out[6]:

```
<matplotlib.legend.Legend at 0x1968748c070>
```

In [7]:

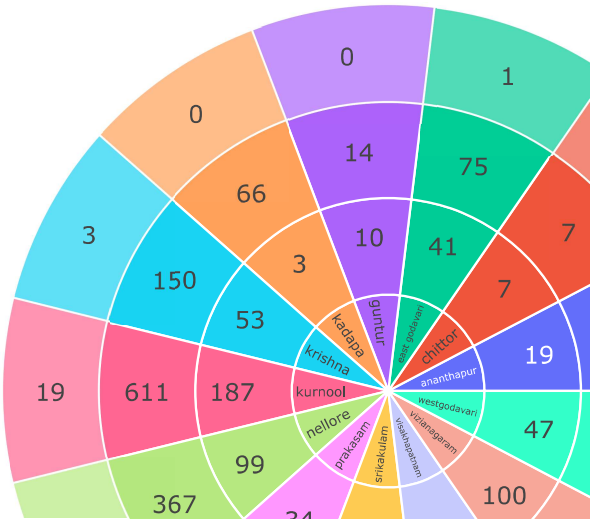
```
data
```

Out[7]:

	districtData/0/district	blank	districtData/0/active	districtData/0/confirmed	districtData/0/decea
0	anthapur	NaN	47	122	
1	chittor	NaN	100	177	
2	eastgodavari	NaN	13	52	
3	guntur	NaN	134	417	
4	kadapa	NaN	34	102	
5	krishna	NaN	99	367	
6	kurnool	NaN	187	611	
7	nellore	NaN	53	150	
8	prakasam	NaN	3	66	
9	srikakulam	NaN	10	14	
10	vishakapatnam	NaN	41	75	
11	vizayanagram	NaN	7	7	
12	weatgodavari	NaN	19	70	

In [8]:

```
import plotly.express as px
import pandas as pd


DISTRICTS = ["westgodavari", "vizianagaram", "visakhapatnam", "srikakulam", "prakasam", "ne
        "kurnool", "krishna", "kadapa", "guntur","east godavari","chittor","ananthapur"]

ACTIVE= ["47", "100", "13", "134", "34","99",
        "187", "53", "3", "10", "41","7","19"]
CONFIRMED = ["122", "177", "52", "417", "102","367" ,"611" ,
        "150", "66", "14", "75", "7","70"]

RECOVERED = ["4", "0", "0", "8", "0","15",
        "19", "3", "0", "0", "1","1","1"]


df = pd.DataFrame(dict(DISTRICTS=DISTRICTS, ACTIVE=ACTIVE, CONFIRMED=CONFIRMED, RECOVERED=R

fig = px.sunburst(df, path=['DISTRICTS','ACTIVE','CONFIRMED','RECOVERED'] )

fig.show()
```



In [9]:

```
import pandas as pd
df= pd.read_csv('Covid cases in India (3).csv')  #states of covid cases
```

In [10]:

```
df
```

Out[10]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
0	1	Andhra Pradesh	12	0	1	0
1	2	Chhattisgarh	6	0	0	0
2	3	Delhi	38	1	6	1
3	4	Gujarat	43	0	0	3
4	5	Haryana	16	14	11	0
5	6	Himachal Pradesh	4	0	0	1
6	7	Karnataka	20	0	3	2
7	8	Kerala	131	7	11	0
		Madhya				

In [11]:

```
df['Total Cases']=df['Total Confirmed cases (Indian National)']+df['Total Confirmed cases (
```

In [12]:

```
data
```

Out[12]:

	districtData/0/district	blank	districtData/0/active	districtData/0/confirmed	districtData/0/decea
0	anthapur	NaN	47	122	
1	chittor	NaN	100	177	
2	eastgodavari	NaN	13	52	
3	guntur	NaN	134	417	
4	kadapa	NaN	34	102	
5	krishna	NaN	99	367	
6	kurnool	NaN	187	611	
7	nellore	NaN	53	150	
8	prakasam	NaN	3	66	
9	srikakulam	NaN	10	14	
10	vishakapatnam	NaN	41	75	
11	vizayanagram	NaN	7	7	
12	weatgodavari	NaN	19	70	

In [13]:

```
total_cases_overall=df['Total Cases'].sum()
print('The total number of cases till now in India is ',total_cases_overall)
```

```
The total number of cases till now in India is  729
```

In [14]:

```
df['Active Cases']=df['Total Cases']-(df['Death']+df['Cured'])
```

In [15]:

```
df
```

Out[15]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total Cases	Active Cases
0	1	Andhra Pradesh	12	0	1	0	12	11
1	2	Chhattisgarh	6	0	0	0	6	6
2	3	Delhi	38	1	6	1	39	32
3	4	Gujarat	43	0	0	3	43	40
4	5	Haryana	16	14	11	0	30	19
5	6	Himachal Pradesh	4	0	0	1	4	3
6	7	Karnataka	20	0	3	2	20	15
7	8	Kerala	131	7	11	0	138	127
8	9	Madhya Pradesh	23	0	0	1	23	22
9	10	Maharashtra	144	3	15	4	147	128
10	11	Odisha	3	0	0	0	3	3
11	12	Puducherry	1	0	0	0	1	1
12	13	Punjab	29	0	0	1	29	28
13	14	Rajasthan	41	2	3	0	43	40
14	15	Tamil Nadu	32	3	1	1	35	33
15	16	Telengana	34	11	1	0	45	44
16	17	Chandigarh	7	0	0	0	7	7
17	18	Jammu and Kashmir	18	0	1	1	18	16
18	19	Ladakh	13	0	0	0	13	13
19	20	Uttar Pradesh	42	1	11	0	43	32
20	21	Uttarakhand	4	0	0	0	4	4
21	22	West Bengal	11	0	0	1	11	10
22	23	Bihar	7	0	0	1	7	6
23	24	Mizoram	1	0	0	0	1	1
24	25	Goa	6	0	0	0	6	6
25	26	Manipur	1	0	0	0	1	1

In [16]:

```
df.style.background_gradient(cmap='Reds')
```

Out[16]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total Cases	Active Cases
0	1	Andhra Pradesh	12	0	1	0	12	11
1	2	Chhattisgarh	6	0	0	0	6	6
2	3	Delhi	38	1	6	1	39	32
3	4	Gujarat	43	0	0	3	43	40
4	5	Haryana	16	14	11	0	30	19
5	6	Himachal Pradesh	4	0	0	1	4	3
6	7	Karnataka	20	0	3	2	20	15
7	8	Kerala	131	7	11	0	138	127
8	9	Madhya Pradesh	23	0	0	1	23	22
9	10	Maharashtra	144	3	15	4	147	128
10	11	Odisha	3	0	0	0	3	3
11	12	Puducherry	1	0	0	0	1	1
12	13	Punjab	29	0	0	1	29	28
13	14	Rajasthan	41	2	3	0	43	40
14	15	Tamil Nadu	32	3	1	1	35	33
15	16	Telengana	34	11	1	0	45	44
16	17	Chandigarh	7	0	0	0	7	7
17	18	Jammu and Kashmir	18	0	1	1	18	16
18	19	Ladakh	13	0	0	0	13	13
19	20	Uttar Pradesh	42	1	11	0	43	32
20	21	Uttarakhand	4	0	0	0	4	4
21	22	West Bengal	11	0	0	1	11	10
22	23	Bihar	7	0	0	1	7	6
23	24	Mizoram	1	0	0	0	1	1
24	25	Goa	6	0	0	0	6	6
25	26	Manipur	1	0	0	0	1	1

In [17]:

```
Total_Active_Cases=df.groupby('Name of State / UT')['Total Cases'].sum().sort_values(ascend
```

In [18]:

```
Total_Active_Cases
```

Out[18]:

Name of State / UT	Total Cases
Maharashtra	147
Kerala	138
Telengana	45
Uttar Pradesh	43
Gujarat	43
Rajasthan	43
Delhi	39
Tamil Nadu	35
Haryana	30

In [19]:

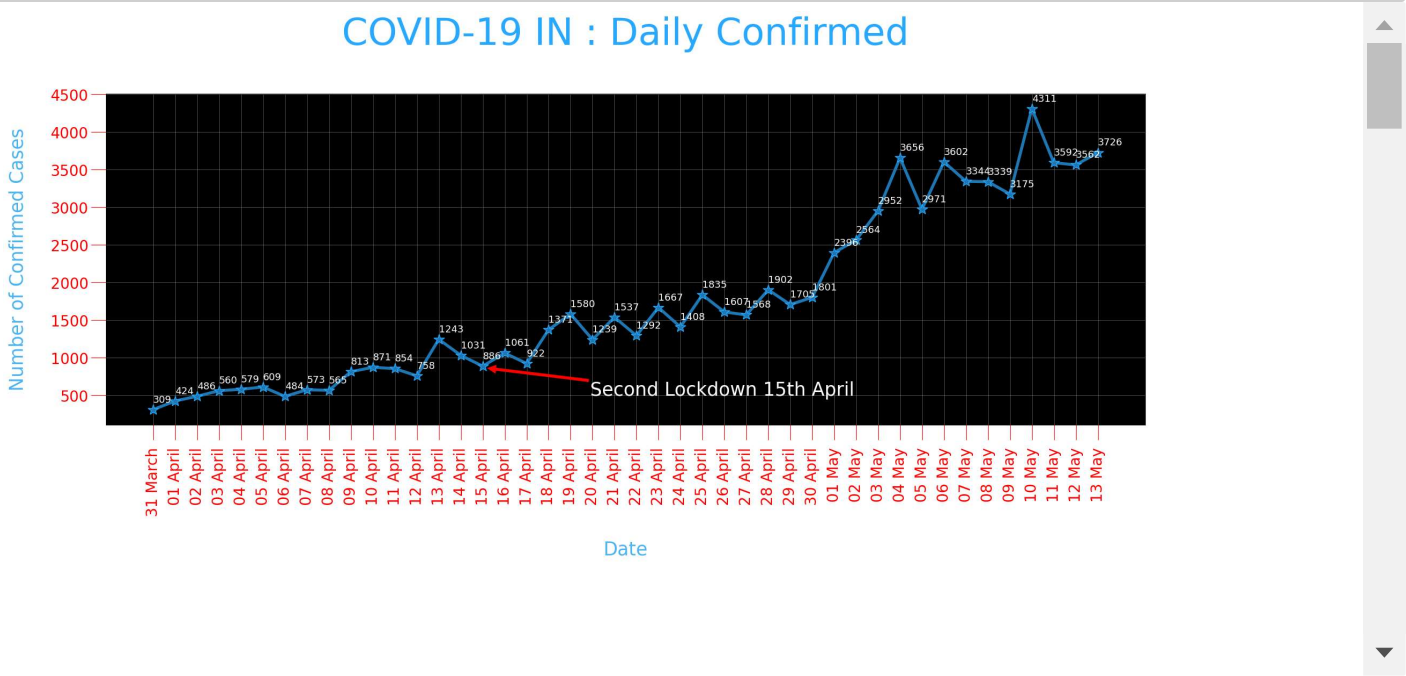
```
Total_Active_Cases.style.background_gradient(cmap='Reds')
```

Out[19]:

Name of State / UT	Total Cases
Maharashtra	147
Kerala	138
Telengana	45
Uttar Pradesh	43
Gujarat	43
Rajasthan	43
Delhi	39
Tamil Nadu	35
Haryana	30
Punjab	29
Madhya Pradesh	23
Karnataka	20
Jammu and Kashmir	18
Ladakh	13
Andhra Pradesh	12
West Bengal	11
Bihar	7
Chandigarh	7
Goa	6
Chhattisgarh	6
Himachal Pradesh	4
Uttarakhand	4
Odisha	3
Manipur	1
Mizoram	1
Puducherry	1

In [20]:

```
fig=plt.figure(figsize=(20,10),dpi=200)
axes=fig.add_axes([0,0,1,1])
axes.bar(df['Name of State / UT'],df['Total Cases'])
axes.set_title("Total Cases in India")
axes.set_xlabel("Name of State / UT")
axes.set_ylabel("Total Cases")

plt.show()

#plotly
fig=go.Figure()
fig.add_trace(go.Bar(x=df['Name of State / UT'],y=df['Total Cases']))
fig.update_layout(title='Total Cases in India',xaxis=dict(title='Name of State / UT'),yaxis
```



In [21]:

```
data= pd.read_csv('Covid cases in India (3).csv')  #st
```

In [22]:

```
data
```

Out[22]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
0	1	Andhra Pradesh	12	0	1	0
1	2	Chhattisgarh	6	0	0	0
2	3	Delhi	38	1	6	1
3	4	Gujarat	43	0	0	3
4	5	Haryana	16	14	11	0
5	6	Himachal Pradesh	4	0	0	1
6	7	Karnataka	20	0	3	2
7	8	Kerala	131	7	11	0
8	9	Madhya Pradesh	23	0	0	1
9	10	Maharashtra	144	3	15	4
10	11	Odisha	3	0	0	0
11	12	Puducherry	1	0	0	0
12	13	Punjab	29	0	0	1
13	14	Rajasthan	41	2	3	0
14	15	Tamil Nadu	32	3	1	1
15	16	Telengana	34	11	1	0
16	17	Chandigarh	7	0	0	0
17	18	Jammu and Kashmir	18	0	1	1
18	19	Ladakh	13	0	0	0
19	20	Uttar Pradesh	42	1	11	0
20	21	Uttarakhand	4	0	0	0
21	22	West Bengal	11	0	0	1
22	23	Bihar	7	0	0	1
23	24	Mizoram	1	0	0	0
24	25	Goa	6	0	0	0
25	26	Manipur	1	0	0	0

In [ ]:

```

```

In [23]:

```
Indian_Cord=pd.read_csv('Indian Coordinates.csv')
```

In [24]:

```
Indian_Cord
```

Out[24]:

	Name of State / UT	Latitude	Longitude	Unnamed: 3
0	Andaman And Nicobar	11.667026	92.735983	NaN
1	Andhra Pradesh	14.750429	78.570026	NaN
2	Arunachal Pradesh	27.100399	93.616601	NaN
3	Assam	26.749981	94.216667	NaN
4	Bihar	25.785414	87.479973	NaN
5	Chandigarh	30.719997	76.780006	NaN
6	Chhattisgarh	22.090420	82.159987	NaN
7	Dadra And Nagar Haveli	20.266578	73.016618	NaN
8	Delhi	28.669993	77.230004	NaN
9	Goa	15.491997	73.818001	NaN
10	Haryana	28.450006	77.019991	NaN
11	Himachal Pradesh	31.100025	77.166597	NaN
12	Union Territory of Jammu and Kashmir	33.450000	76.240000	NaN
13	Jharkhand	23.800393	86.419986	NaN
14	Karnataka	12.570381	76.919997	NaN
15	Kerala	8.900373	76.569993	NaN
16	Lakshadweep	10.562573	72.636867	NaN
17	Madhya Pradesh	21.300391	76.130019	NaN
18	Maharashtra	19.250232	73.160175	NaN
19	Manipur	24.799971	93.950017	NaN
20	Meghalaya	25.570492	91.880014	NaN
21	Mizoram	23.710399	92.720015	NaN
22	Nagaland	25.666998	94.116570	NaN
23	Orissa	19.820430	85.900017	NaN
24	Puducherry	11.934994	79.830000	NaN
25	Punjab	31.519974	75.980003	NaN
26	Rajasthan	26.449999	74.639981	NaN
27	Sikkim	27.333330	88.616647	NaN
28	Telengana	18.112400	79.019300	NaN
29	Tamil Nadu	12.920386	79.150042	NaN
30	Tripura	23.835404	91.279999	NaN
31	Uttar Pradesh	27.599981	78.050006	NaN
32	Uttarakhand	30.320409	78.050006	NaN
33	West Bengal	22.580390	88.329947	NaN

	Name of State / UT	Latitude	Longitude	Unnamed: 3
24	Union Territory of Ladakh	34.100000	77.340000	NaN

In [25]:

```
dd=pd.merge(Indian_Cord,df,on='Name of State / UT')
```

In [26]:

```
dd
```

Out[26]:

	Name of State / UT	Latitude	Longitude	Unnamed: 3	S. No.	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	C
0	Andhra Pradesh	14.750429	78.570026	NaN	1	12	0	1	0	
1	Delhi	28.669993	77.230004	NaN	3	38	1	6	1	
2	Haryana	28.450006	77.019991	NaN	5	16	14	11	0	
3	Karnataka	12.570381	76.919997	NaN	7	20	0	3	2	
4	Kerala	8.900373	76.569993	NaN	8	131	7	11	0	
5	Maharashtra	19.250232	73.160175	NaN	10	144	3	15	4	
6	Punjab	31.519974	75.980003	NaN	13	29	0	0	1	
7	Rajasthan	26.449999	74.639981	NaN	14	41	2	3	0	
8	Telengana	18.112400	79.019300	NaN	16	34	11	1	0	
9	Tamil Nadu	12.920386	79.150042	NaN	15	32	3	1	1	
10	Uttar Pradesh	27.599981	78.050006	NaN	20	42	1	11	0	
11	Uttarakhand	30.320409	78.050006	NaN	21	4	0	0	0	

In [27]:

```
map=folium.Map(location=[20,70],zoom_start=4,tiles='Stamenterrain')

for lat,long,value, name in zip(dd['Latitude'],dd['Longitude'],dd['Total Cases'],dd['Name o
    folium.CircleMarker([lat,long],radius=value*0.8,popup=('<strong>State</strong>: '+str(n
```

Type *Markdown* and LaTeX: α^2

In [28]:

```
map
```

Out[28]:

Type *Markdown* and LaTeX: α^2

In [ ]:

```
```