In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import folium
```

1. Import pandas and read in the india_statewise.json file into a dataframe called covid.

In [97]:
```
df=pd.read_json('india_statewise.json')
```

In [98]:
```
df
```

Out[98]:

	success	data	lastRefreshed	lastOriginUpdate
lastRefreshed	True	2020-07-12T03:54:15.207Z	2020-07-12T03:54:15.207Z	2020-07-12T03:54:15.207Z
source	True	covid19india.org	2020-07-12T03:54:15.207Z	2020-07-12T03:54:15.207Z
statewise	True	[{'state': 'Maharashtra', 'confirmed': 246600,...	2020-07-12T03:54:15.207Z	2020-07-12T03:54:15.207Z
total	True	{'confirmed': 850358, 'recovered': 536231, 'de...	2020-07-12T03:54:15.207Z	2020-07-12T03:54:15.207Z

2. Normalize the dataframe with ‘data’ as column and ‘statewise’ as row and rename it as df_india

In [99]:
```
df_india = pd.io.json.json_normalize(df['data']['statewise'])
df_india
```

```
C:\Users\Toshiba\anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWar
ning: pandas.io.json.json_normalize is deprecated, use pandas.json_normalize in
stead
  """Entry point for launching an IPython kernel.
```

Out[99]:

	state	confirmed	recovered	deaths	active
0	Maharashtra	246600	136985	10116	99202
1	Tamil Nadu	134226	85915	1898	46413
2	Delhi	110921	87692	3334	19895
3	Gujarat	41027	28685	2033	10309
4	Uttar Pradesh	35092	22689	913	11490
5	Karnataka	36216	14718	615	20879
6	Telangana	33402	20919	348	12135
7	West Bengal	28453	17959	906	9588
8	Andhra Pradesh	27235	14393	309	12533
9	Rajasthan	23748	17869	503	5376
10	Haryana	20582	15394	297	4891
11	Madhya Pradesh	17201	12679	644	3878
12	Assam	16072	10427	37	5605
13	Bihar	15039	10991	118	3930
14	Odisha	12526	8360	79	4087
15	Jammu and Kashmir	10156	5895	169	4092
16	Punjab	7587	5040	195	2352
17	Kerala	7439	3963	30	3442
18	State Unassigned	3416	0	0	3416
19	Chhattisgarh	3897	3070	17	810
20	Uttarakhand	3417	2718	46	623
21	Jharkhand	3663	2256	24	1383
22	Goa	2368	1428	12	928
23	Tripura	1962	1375	2	571
24	Manipur	1593	843	0	750
25	Puducherry	1336	690	18	628
26	Himachal Pradesh	1182	895	10	264
27	Ladakh	1077	928	1	148
28	Nagaland	748	313	0	435
29	Chandigarh	549	413	8	128

	state	confirmed	recovered	deaths	active
30	Dadra and Nagar Haveli and Daman and Diu	482	237	1	240
31	Arunachal Pradesh	341	125	2	214
32	Mizoram	227	150	0	77
33	Andaman and Nicobar Islands	156	92	0	64
34	Sikkim	160	80	0	67
35	Meghalaya	262	45	2	215
36	Lakshadweep	0	0	0	0

3. Now make ‘state’ column as index

In [100]:
```
df_india = df_india.set_index('state')
df_india.head()
```

Out[100]:

state	confirmed	recovered	deaths	active
Maharashtra	246600	136985	10116	99202
Tamil Nadu	134226	85915	1898	46413
Delhi	110921	87692	3334	19895
Gujarat	41027	28685	2033	10309
Uttar Pradesh	35092	22689	913	11490

4. Now Identify a row which is not a valid state and drop it

In [101]:
```
df_india = df_india.drop([ 'State Unassigned'], axis=0)
df_india
```

Out[101]:

state	confirmed	recovered	deaths	active
Maharashtra	246600	136985	10116	99202
Tamil Nadu	134226	85915	1898	46413
Delhi	110921	87692	3334	19895
Gujarat	41027	28685	2033	10309
Uttar Pradesh	35092	22689	913	11490
Karnataka	36216	14718	615	20879
Telangana	33402	20919	348	12135
West Bengal	28453	17959	906	9588
Andhra Pradesh	27235	14393	309	12533
Rajasthan	23748	17869	503	5376
Haryana	20582	15394	297	4891
Madhya Pradesh	17201	12679	644	3878
Assam	16072	10427	37	5605
Bihar	15039	10991	118	3930
Odisha	12526	8360	79	4087
Jammu and Kashmir	10156	5895	169	4092
Punjab	7587	5040	195	2352
Kerala	7439	3963	30	3442
Chhattisgarh	3897	3070	17	810
Uttarakhand	3417	2718	46	623
Jharkhand	3663	2256	24	1383
Goa	2368	1428	12	928
Tripura	1962	1375	2	571
Manipur	1593	843	0	750
Puducherry	1336	690	18	628
Himachal Pradesh	1182	895	10	264
Ladakh	1077	928	1	148
Nagaland	748	313	0	435
Chandigarh	549	413	8	128
Dadra and Nagar Haveli and Daman and Diu	482	237	1	240
Arunachal Pradesh	341	125	2	214
Mizoram	227	150	0	77

state	confirmed	recovered	deaths	active
Andaman and Nicobar Islands	156	92	0	64
Sikkim	160	80	0	67
Meghalaya	262	45	2	215
Lakshadweep	0	0	0	0

5. Calculate the Mortality rate (deaths/confirmed) for each state and identify top 5 sates with highest mortality rate.

In [102]:
```
df_india['Mortality Rate'] =(df_india['deaths']/df_india['confirmed'])*100
df_india
```

Out[102]:

state	confirmed	recovered	deaths	active	Mortality Rate
Maharashtra	246600	136985	10116	99202	4.102190
Tamil Nadu	134226	85915	1898	46413	1.414033
Delhi	110921	87692	3334	19895	3.005743
Gujarat	41027	28685	2033	10309	4.955273
Uttar Pradesh	35092	22689	913	11490	2.601733
Karnataka	36216	14718	615	20879	1.698144
Telangana	33402	20919	348	12135	1.041854
West Bengal	28453	17959	906	9588	3.184199
Andhra Pradesh	27235	14393	309	12533	1.134569
Rajasthan	23748	17869	503	5376	2.118073
Haryana	20582	15394	297	4891	1.443008
Madhya Pradesh	17201	12679	644	3878	3.743968
Assam	16072	10427	37	5605	0.230214
Bihar	15039	10991	118	3930	0.784627
Odisha	12526	8360	79	4087	0.630688
Jammu and Kashmir	10156	5895	169	4092	1.664041
Punjab	7587	5040	195	2352	2.570186
Kerala	7439	3963	30	3442	0.403280
Chhattisgarh	3897	3070	17	810	0.436233
Uttarakhand	3417	2718	46	623	1.346210
Jharkhand	3663	2256	24	1383	0.655201
Goa	2368	1428	12	928	0.506757
Tripura	1962	1375	2	571	0.101937
Manipur	1593	843	0	750	0.000000
Puducherry	1336	690	18	628	1.347305
Himachal Pradesh	1182	895	10	264	0.846024
Ladakh	1077	928	1	148	0.092851
Nagaland	748	313	0	435	0.000000
Chandigarh	549	413	8	128	1.457195
Dadra and Nagar Haveli and Daman and Diu	482	237	1	240	0.207469
Arunachal Pradesh	341	125	2	214	0.586510
Mizoram	227	150	0	77	0.000000

state	confirmed	recovered	deaths	active	Mortality Rate
Andaman and Nicobar Islands	156	92	0	64	0.000000
Sikkim	160	80	0	67	0.000000
Meghalaya	262	45	2	215	0.763359
Lakshadweep	0	0	0	0	NaN

In [103]:
```
df_india.nlargest(5, ['Mortality Rate']) #identify top 5 sates with highest morta
```

Out[103]:

state	confirmed	recovered	deaths	active	Mortality Rate
Gujarat	41027	28685	2033	10309	4.955273
Maharashtra	246600	136985	10116	99202	4.102190
Madhya Pradesh	17201	12679	644	3878	3.743968
West Bengal	28453	17959	906	9588	3.184199
Delhi	110921	87692	3334	19895	3.005743

Now add the latitude and longitude information of each state in the df_india dataframe using the dictionary dataset (named locations) given below:

In [104]:
```
locations = {
 "Kerala" : [10.8505,76.2711],
 "Maharashtra" : [19.7515,75.7139],
 "Karnataka": [15.3173,75.7139],
 "Telangana": [18.1124,79.0193],
 "Uttar Pradesh": [26.8467,80.9462],
 "Rajasthan": [27.0238,74.2179],
 "Gujarat":[22.2587,71.1924],
 "Delhi" : [28.7041,77.1025],
 "Punjab":[31.1471,75.3412],
 "Tamil Nadu": [11.1271,78.6569],
 "Haryana": [29.0588,76.0856],
 "Madhya Pradesh":[22.9734,78.6569],
 "Jammu and Kashmir":[33.7782,76.5762],
 "Ladakh": [34.1526,77.5770],
 "Andhra Pradesh":[15.9129,79.7400],
 "West Bengal": [22.9868,87.8550],
 "Bihar": [25.0961,85.3131],
 "Chhattisgarh":[21.2787,81.8661],
 "Chandigarh":[30.7333,76.7794],
 "Uttarakhand":[30.0668,79.0193],
 "Himachal Pradesh":[31.1048,77.1734],
 "Goa": [15.2993,74.1240],
 "Odisha":[20.9517,85.0985],
 "Andaman and Nicobar Islands": [11.7401,92.6586],
 "Puducherry":[11.9416,79.8083],
 "Manipur":[24.6637,93.9063],
 "Mizoram":[23.1645,92.9376],
 "Assam":[26.2006,92.9376],
 "Meghalaya":[25.4670,91.3662],
 "Tripura":[23.9408,91.9882],
 "Arunachal Pradesh":[28.2180,94.7278],
 "Jharkhand" : [23.6102,85.2799],
 "Nagaland": [26.1584,94.5624],
 "Sikkim": [27.5330,88.5122],
 "Dadra and Nagar Haveli and Daman and Diu":[20.1809,73.0169],
 "Lakshadweep":[10.5667,72.6417],
}
```

In [105]:
```
df_india['Lat'] = ""
df_india['Long'] = ""
for index in df_india.index:
    df_india.loc[df_india.index==index,'Lat']=locations[index][0]# index value ir
                                                                 # in new column
    df_india.loc[df_india.index==index,'Long']=locations[index][1]
df_india
```

Out[105]:

state	confirmed	recovered	deaths	active	Mortality Rate	Lat	Long
Maharashtra	246600	136985	10116	99202	4.102190	19.7515	75.7139
Tamil Nadu	134226	85915	1898	46413	1.414033	11.1271	78.6569
Delhi	110921	87692	3334	19895	3.005743	28.7041	77.1025
Gujarat	41027	28685	2033	10309	4.955273	22.2587	71.1924
Uttar Pradesh	35092	22689	913	11490	2.601733	26.8467	80.9462
Karnataka	36216	14718	615	20879	1.698144	15.3173	75.7139
Telangana	33402	20919	348	12135	1.041854	18.1124	79.0193
West Bengal	28453	17959	906	9588	3.184199	22.9868	87.855
Andhra Pradesh	27235	14393	309	12533	1.134569	15.9129	79.74
Rajasthan	23748	17869	503	5376	2.118073	27.0238	74.2179
Haryana	20582	15394	297	4891	1.443008	29.0588	76.0856
Madhya Pradesh	17201	12679	644	3878	3.743968	22.9734	78.6569
Assam	16072	10427	37	5605	0.230214	26.2006	92.9376
Bihar	15039	10991	118	3930	0.784627	25.0961	85.3131
Odisha	12526	8360	79	4087	0.630688	20.9517	85.0985
Jammu and Kashmir	10156	5895	169	4092	1.664041	33.7782	76.5762
Punjab	7587	5040	195	2352	2.570186	31.1471	75.3412
Kerala	7439	3963	30	3442	0.403280	10.8505	76.2711
Chhattisgarh	3897	3070	17	810	0.436233	21.2787	81.8661
Uttarakhand	3417	2718	46	623	1.346210	30.0668	79.0193
Jharkhand	3663	2256	24	1383	0.655201	23.6102	85.2799
Goa	2368	1428	12	928	0.506757	15.2993	74.124
Tripura	1962	1375	2	571	0.101937	23.9408	91.9882
Manipur	1593	843	0	750	0.000000	24.6637	93.9063
Puducherry	1336	690	18	628	1.347305	11.9416	79.8083
Himachal Pradesh	1182	895	10	264	0.846024	31.1048	77.1734
Ladakh	1077	928	1	148	0.092851	34.1526	77.577
Nagaland	748	313	0	435	0.000000	26.1584	94.5624

state	confirmed	recovered	deaths	active	Mortality Rate	Lat	Long
Chandigarh	549	413	8	128	1.457195	30.7333	76.7794
Dadra and Nagar Haveli and Daman and Diu	482	237	1	240	0.207469	20.1809	73.0169
Arunachal Pradesh	341	125	2	214	0.586510	28.218	94.7278
Mizoram	227	150	0	77	0.000000	23.1645	92.9376
Andaman and Nicobar Islands	156	92	0	64	0.000000	11.7401	92.6586
Sikkim	160	80	0	67	0.000000	27.533	88.5122
Meghalaya	262	45	2	215	0.763359	25.467	91.3662
Lakshadweep	0	0	0	0	NaN	10.5667	72.6417

Now plot this data using folium map. The expected output should be is shown below:

In [106]:
```
india = folium.Map(location=[23,80],tiles = 'cartodbpositron',zoom_start=4,max_zo

for i in range(0,len(df_india[df_india['confirmed']>0].index)):# range from 0 to
                                                               #confirmed cases
    folium.Circle(
        location=[df_india.iloc[i]['Lat'],df_india.iloc[i]['Long']],#Location of
        tooltip = "<h5 style='text-align:Centre;font-weight:bold'>"+df_india.iloc
        "<li>confirmed:"+str(df_india.iloc[i]['confirmed'])+"</li>"+# This we war
        "<li>recovered:"+str(df_india.iloc[i]['recovered'])+"</li>"+# This we war
        "<li>active:"+str(df_india.iloc[i]['active'])+"</li>"+# This we want to p
        "<li>deaths:"+str(df_india.iloc[i]['deaths'])+"</li>"+# This we want to p
        "<li>Mortality Rate:"+str(df_india.iloc[i]['Mortality Rate'])+"</li>",# T
        radius = (int(df_india.iloc[i]['confirmed'])), # radiu equivalent to conf
        color='#FF4500',#html colour codes oranged
        fill_color="#FF4500",
        fill=True).add_to(india)
```

In [107]:
```
india
```

Out[107]:

In [ ]:

In [ ]: