TamilNadu’s Health Centers Report

(For Covid-19)

Done by: Shreebuvan V

Introduction:

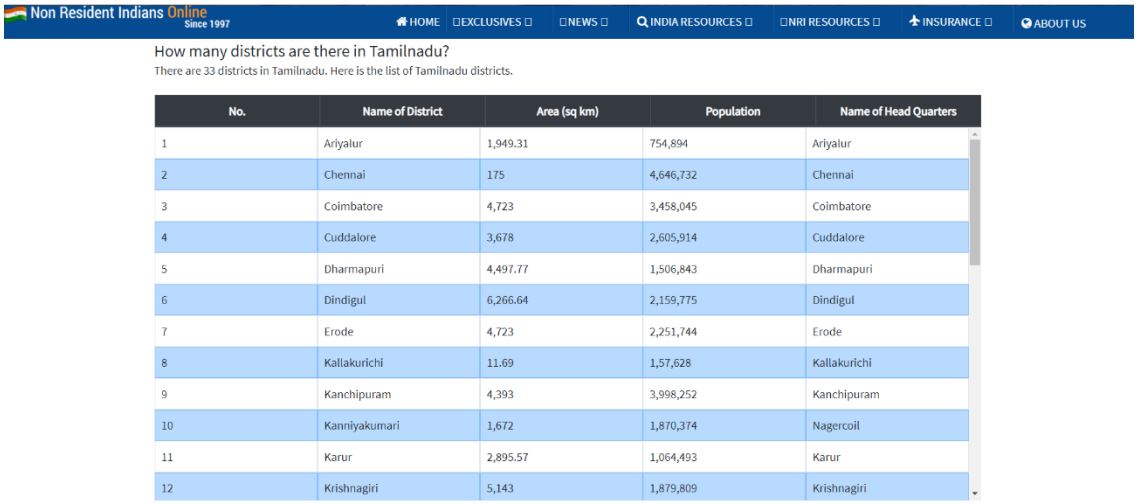
The present scenario of the outbreak of the COVID-19 has led to a pandemic which none of us have never even imagined off. This situation has made it really hard for even developed countries like US. It has also made a wild swipe in India as well with almost 10 lakhs+ cases at present and still counting. Tamilnadu has become one of the most affected states in India with almost 150k+ cases. Although Tamilnadu govt. have taken many steps to bring this pandemic to a control, the population density has always been an alarming problem. Therefore, there is always a huge problem in bringing forth the isolation centers in the state and making this accessible to the masses is also a rising problem. The issues to be kept in mind for this is as follows:

- Centralized centers alone cannot do any good to the people as people from the other parts of the state will not be able to get in access to the facilities
- When the infected are to be taken to the hospitals there are chances that the attendees may get affected and the social distancing fact is taken for granted in a state like Tamilnadu.

So decentralized testing and isolation wards can be very useful for the govt. to both keep the pandemic under control as well as treat the infected with the best facilities available and this work would be of great help.

Data:

I have surfed through the Internet and found out the NRIOL website where anyone can get the split-up of the population and the area (in sq. km) of a given state, district-wise. The most important feature is that we get all the details in tabular form with the .asp file extension which is easier to play around with. The information in the website is as shown in the image below:



No.	Name of District	Area (sq km)	Population	Name of Head Quarters
1	Ariyalur	1,949.31	754,894	Ariyalur
2	Chennai	175	4,646,732	Chennai
3	Coimbatore	4,723	3,458,045	Coimbatore
4	Cuddalore	3,678	2,605,914	Cuddalore
5	Dharmapuri	4,497.77	1,506,843	Dharmapuri
6	Dindigul	6,266.64	2,159,775	Dindigul
7	Erode	4,723	2,251,744	Erode
8	Kallakurichi	11.69	1,57,628	Kallakurichi
9	Kanchipuram	4,393	3,998,252	Kanchipuram
10	Kanniyakumari	1,672	1,870,374	Nagercoil
11	Karur	2,895.57	1,064,493	Karur
12	Krishnagiri	5,143	1,879,809	Krishnagiri

I have done the web-scrapping using the BeautifulSoup available as a part of the bs4 library in python. This information is then compiled and made into a dataframe and the latitude and longitude values where extracted using the Nominatim function in Geopy Library with the names of the districts which where web-scrapped from the NRIOL website. Now this information together makes the dataframe ready to play with and get the needed insights from it. The final dataframe with all the data values looks like the image below.

[54]

	Name of District	Area(sq km)	Population	Latitude	Longitude
0	Ariyalur	1949.31	754894.0	11.076036	79.117455
1	Chennai	175.00	4646732.0	13.080172	80.283833
2	Coimbatore	4723.00	3458045.0	11.001812	76.962842
3	Cuddalore	3678.00	2605914.0	11.742694	79.750306
4	Dharmapuri	4497.77	1506843.0	12.096805	78.193043
5	Dindigul	6266.64	2159775.0	10.330330	78.067398
6	Erode	4723.00	2251744.0	11.369204	77.676627
7	Kallakurichi	11.69	157628.0	11.794685	79.038821
8	Kanchipuram	4393.00	3998252.0	12.964716	79.983969
9	Kanniyakumari	1672.00	1870374.0	8.087964	77.546741
10	Karur	2895.57	1064493.0	10.930152	78.084855
11	Krishnagiri	5143.00	1879809.0	12.518883	78.220654
12	Madurai	3741.73	3038252.0	9.926115	78.114098
13	Nagapattinam	2715.83	1616450.0	10.805628	79.824660
14	Namakkal	3363.00	1726601.0	11.219132	78.237398
15	Nilgiris	2452.50	735394.0	11.400000	76.700000
16	Perambalur	1757.00	565223.0	11.228772	78.818256
17	Pudukkottai	4663.00	1618345.0	10.500000	78.833333
18	Ramanathapuram	4089.57	1353445.0	9.389552	78.859071
19	Salem	5205.00	3482056.0	44.939157	-123.033121

Exploratory Data Analysis:

The data from the NROIL website is scrapped and stored in three lists namely District, Area and Population. Which is shown below.

```
[ ] District = []
    Area = []
    Population=[]


    for i in range(0, len(fields),5):
        District.append(fields[i+1].text.strip())
        Area.append(fields[i+2].text.strip())
        Population.append(fields[i+3].text.strip())
```

```
[ ] District
```

```
['Ariyalur',
 'Chennai',
 'Coimbatore',
 'Cuddalore',
 'Dharmapuri',
 'Dindigul',
 'Erode',
 'Kallakurichi',
 'Kanchipuram',
 'Kanniyakumari',
 'Karur',
 'Krishnagiri',
 'Madurai',
 'Nagapattinam',
 'Namakkal',
 'Nilgiris',
 'Perambalur',
 'Pudukkottai',
```

Next these three lists together form the intermediate dataframe as shown below

Dataframe formed from the data webscrapped from the official NRIO website

```
[ ] df
```

	0	1	2
0	Ariyalur	1,949.31	754,894
1	Chennai	175	4,646,732
2	Coimbatore	4,723	3,458,045
3	Cuddalore	3,678	2,605,914
4	Dharmapuri	4,497.77	1,506,843
5	Dindigul	6,266.64	2,159,775
6	Erode	4,723	2,251,744
7	Kallakurichi	11.69	1,57,628
8	Kanchipuram	4,393	3,998,252
9	Kanniyakumari	1,672	1,870,374
10	Karur	2,895.57	1,064,493
11	Krishnagiri	5,143	1,879,809
12	Madurai	3,741.73	3,038,252
13	Nagapattinam	2,715.83	1,616,450
14	Namakkal	3,363	1,726,601
15	Nilgiris	2,452.5	735,394

The area and population columns datatype are changed to float using the astype() method and the comma’s in the numerical values are also removed using the replace() method.

```
[ ] df.dtypes
```

```
Name of District    object
Area(sq km)         object
Population          object
dtype: object
```

```
[48] df['Area(sq km)']=df['Area(sq km)'].str.replace(",","")
     df['Population']=df['Population'].str.replace(",","")
```

```
[50] df[['Area(sq km)','Population']]=df[['Area(sq km)','Population']].astype('float64')
```

```
[52] df.head()
```

	Name of District	Area(sq km)	Population
0	Ariyalur	1949.31	754894.0
1	Chennai	175.00	4646732.0
2	Coimbatore	4723.00	3458045.0
3	Cuddalore	3678.00	2605914.0
4	Dharmapuri	4497.77	1506843.0

Distribution chart of the population denstity spread accross the various districts in Tamilnadu

```
[73] %matplotlib inline
     import matplotlib.pyplot as plt
     df_density.plot(kind='pie',figsize=(10,10))
     plt.xlabel('Population per km^2')
     plt.ylabel('')
```

```
[73] Text(0, 0.5, '')
```

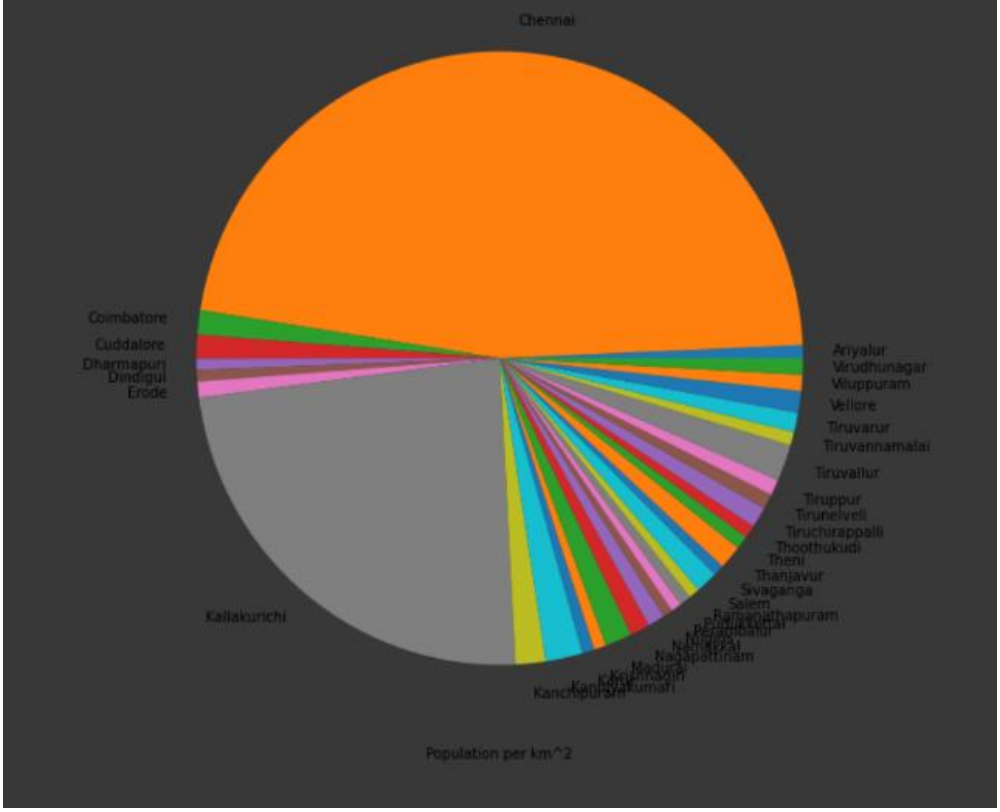


Figure: The population distribution in Tamilnadu (population/km^2)

Dataframe formed from the data webscrapped from the official NRIO website with the latitude and longitude details from the Nominatin function

```
Latitude=[]
Longitude=[]
for i in range(0,df.shape[0]):
    geolocator=Nominatim(user_agent="Shreebuvan")
    location = geolocator.geocode(df["Name of District"].iloc[i])
    Latitude.append(location.latitude)
    Longitude.append(location.longitude)
Latitude=pd.DataFrame(Latitude)
Longitude=pd.DataFrame(Longitude)
df[["Latitude"]]=Latitude
df[["Longitude"]]=Longitude
df
```

	Name of District	Area(sq km)	Population	Latitude	Longitude
0	Ariyalur	1949.31	754894.0	11.076036	79.117455
1	Chennai	175.00	4646732.0	13.080172	80.283833
2	Coimbatore	4723.00	3458045.0	11.001812	76.962842
3	Cuddalore	3678.00	2605914.0	11.742694	79.750306
4	Dharmapuri	4497.77	1506843.0	12.096805	78.193043
5	Dindigul	6266.64	2159775.0	10.330330	78.067398
6	Erode	4723.00	2251744.0	11.369204	77.676627
7	Kallakurichi	11.69	157628.0	11.794685	79.038821
8	Kanchipuram	4393.00	3998252.0	12.964716	79.983969

Figure: Shows the final dataframe with the latitude and longitute of the different districts

Methodology and Results:

From all the analysis made, I have decided to get the individual details of all the nearby hospitals, medical supply centers, medical labs and emergency centers in Tamilnadu to a radius of 5000 set to the foursquare api call, then at last merge all the details and cluster them together to give an insight of the facilities that are available for the affected and most importantly, facilities closest to them.

Renaming the columns in the dataframe

```
[131] nearby_hospitals.columns = [col.split(".")[-1] for col in nearby_hospitals.columns]

    nearby_hospitals.drop_duplicates(inplace=True)
    nearby_hospitals=nearby_hospitals.set_index("name")
    nearby_hospitals=nearby_hospitals.reset_index()
    nearby_hospitals
```

	name	categories	lat	lng
0	Rajiv Gandhi Government General Hospital	Hospital	13.081822	80.276719
1	Apollo Hospitals	Hospital	13.062761	80.251790
2	Dr. U Mohan Rau Memorial Hospital	Hospital	13.079339	80.257258
3	Apollo First Med Hospitals	Hospital	13.077976	80.246140
4	Apollo hospitals	Hospital	13.062767	80.251862
5	Apollo Hospital	Hospital	13.066777	80.252211
6	Kumaran Hospital	Hospital	13.078651	80.249352
7	Apollo Heart Centre	Hospital	13.060618	80.254335
8	Sankara Nethralaya Eye Hospital	Hospital	13.066993	80.250306
9	Shankara Netralaya	Hospital	13.065819	80.250147
10	Apollo Childrens Hospital	Hospital	13.060041	80.252538
11	Seethapathy Clinic and Hospital	Hospital	13.048841	80.264537
12	Apollo Cradle Royale	Hospital	13.059802	80.252626

Figure: The details of the Hospitals retrieved from the foursquare api.

Renaming the columns in the dataframe

```
[139] nearby_medicalsupply.columns = [col.split(".")[-1] for col in nearby_medicalsupply.columns]

    nearby_medicalsupply.drop_duplicates(inplace=True)
    nearby_medicalsupply=nearby_medicalsupply.set_index("name")
    nearby_medicalsupply=nearby_medicalsupply.reset_index()
    nearby_medicalsupply
```

	name	categories	lat	lng
0	Draeger India Private Limited	Medical Supply Store	13.068533	80.241600
1	Pacific Medical Supply	Medical Supply Store	44.950934	-123.035338
2	Providence Home Medical Equipment - Salem	Medical Supply Store	44.916395	-123.027078
3	Beltone Hearing Care Center	Medical Supply Store	44.951094	-122.995100
4	Apria Healthcare	Medical Supply Store	44.910185	-123.020241
5	Scrubs & Beyond	Medical Supply Store	44.943658	-122.982511
6	Fred Meyer Pharmacy	Pharmacy	44.968285	-123.030550
7	Boydon Medical	Medical Center	44.939856	-123.035873
8	Fred Meyer Pharmacy	Pharmacy	44.905755	-123.040987
9	Fred Meyer Pharmacy	Pharmacy	44.948714	-122.986186
10	MedPlus	Medical Supply Store	12.234352	79.070809

Figure: The details of the Medical Supply Centers from the foursquare api.

Renaming the columns in the dataframe

```
[129] nearby_medical_labs.columns = [col.split(".")[-1] for col in nearby_medical_labs.columns]

    nearby_medical_labs.drop_duplicates(inplace=True)
    nearby_medical_labs=nearby_medical_labs.set_index("name")
    nearby_medical_labs=nearby_medical_labs.reset_index()
    nearby_medical_labs
```

	name	categories	lat	lng
0	kiomedworld	Medical Lab	13.089009	80.276582
1	Premier Health Center	Medical Lab	13.079259	80.257644
2	Anderson Diagnostics Lab	Medical Lab	13.079122	80.254898
3	Apollo Heart Centre Immigration Checks	Medical Lab	13.060726	80.254341
4	Ehrlich Laboratory	Medical Lab	13.048591	80.266273
5	R&D Histopathology Lab	Medical Lab	13.042812	80.261078
6	Accura Diagnostics	Medical Lab	11.013256	76.966895
7	Saberwal Surgicals	Medical Lab	11.010440	76.951442
8	ThyroCare	Medical Lab	11.008856	76.949062
9	Immuno Clinical Services	Medical Lab	11.009766	76.947863
10	Hearing Aid Center	Medical Lab	11.016632	76.974030
11	Bioline Lab	Medical Lab	11.012847	76.947621
12	Flash Dental x-ray	Medical Lab	9.942406	78.129320

Figure: The details of the Medical Lab from the foursquare api.

Renaming the columns in the dataframe

```
[128] nearby_emergency_centres.columns = [col.split(".")[-1] for col in nearby_emergency_centres.columns]

    nearby_emergency_centres.drop_duplicates(inplace=True)
    nearby_emergency_centres=nearby_emergency_centres.set_index("name")
    nearby_emergency_centres=nearby_emergency_centres.reset_index()
    nearby_emergency_centres
```

	name	categories	lat	lng
0	EMRI	Emergency Room	13.061679	80.277275
1	Salem Hospital ER	Emergency Room	44.932509	-123.033359
2	Hearing Rehab Center	Emergency Room	44.925669	-123.044020
3	Providence Express Care At Home - House Call D...	Emergency Room	44.917240	-123.028780

Figure: The details of the Emergency Centers from the foursquare api.

Merging data for the available Hospitals, Medical Supply, Medical Labs, Emergency centers in the given radius

```
[140] df_Tamilnadu_Health_Centers = pd.concat([nearby_hospitals, nearby_medicalsupply, nearby_medical_labs,nearby_emergency_centres], ignore_index=True)
    df_Tamilnadu_Health_Centers
```

102	Durai Gastro Care Hospital	Hospital	11.933467	79.484753
103	E.S. Hospital	Hospital	11.937008	79.485914
104	Draeger India Private Limited	Medical Supply Store	13.068533	80.241600
105	Pacific Medical Supply	Medical Supply Store	44.950934	-123.035338
106	Providence Home Medical Equipment - Salem	Medical Supply Store	44.916395	-123.027078
107	Beltone Hearing Care Center	Medical Supply Store	44.951094	-122.995100
108	Apria Healthcare	Medical Supply Store	44.910185	-123.020241
109	Scrubs & Beyond	Medical Supply Store	44.943658	-122.982511
110	Fred Meyer Pharmacy	Pharmacy	44.968285	-123.030550
111	Boydon Medical	Medical Center	44.939856	-123.035873
112	Fred Meyer Pharmacy	Pharmacy	44.905755	-123.040987
113	Fred Meyer Pharmacy	Pharmacy	44.948714	-122.986186
114	MedPlus	Medical Supply Store	12.234352	79.070809
115	kiomedworld	Medical Lab	13.089009	80.276582
116	Premier Health Center	Medical Lab	13.079259	80.257644
117	Anderson Diagnostics Lab	Medical Lab	13.079122	80.254898

Figure: The details of the Hospitals, Medical Supply Centers, Medical Labs and Emergency Centers are all merged together.

Now that I am ready with the datails of all the facilities avalilable as a dataframe the next is to visualize all the facilities individually and collectively on the Tamilnadu map. I have done this using the folium.Map() function in the folium library in available in python.

Map for the available Hospitals in the given radius

```
[147] Tamilnadu_map = folium.Map(location=[latitude, longitude], zoom_start=7)

    for lat, lng, name, categories in zip(nearby_hospitals['lat'], nearby_hospitals['lng'],nearby_hospitals['name'], nearby_hospitals['categories']):
        label = '{}'.format(name)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='blue',
            fill=True,
            fill_color='blue',
            fill_opacity=0.7,
            parse_html=False).add_to(Tamilnadu_map)

    Tamilnadu_map
```

Figure: Passing the latitude and longitude values of Hospitals that should we marked.

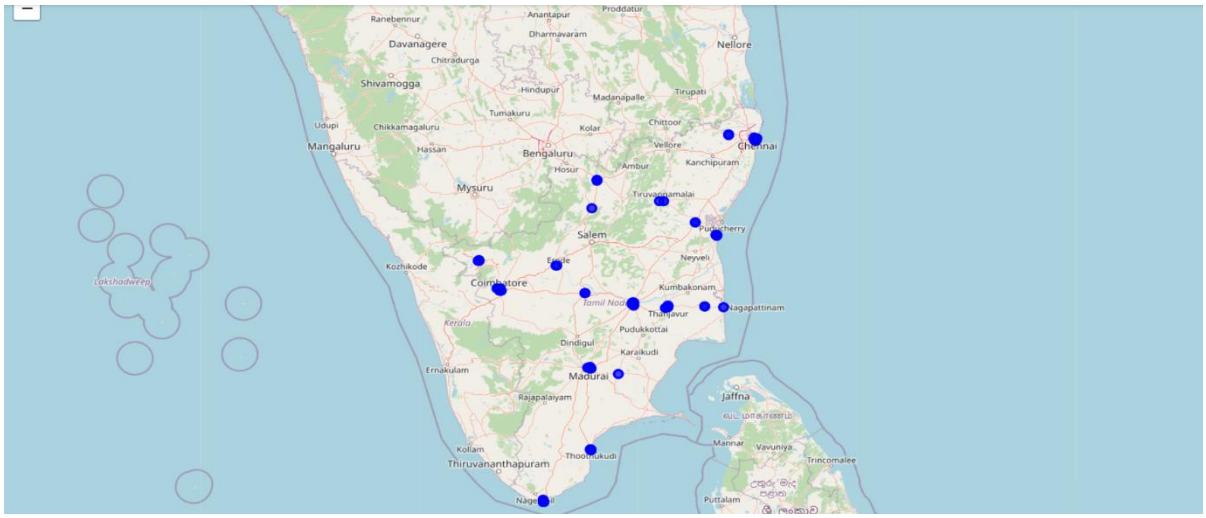


Figure: Map with the available hospitals marked on it.

Map for the available Medical Supply in the given radius

```
[156] Tamilnadu_map = folium.Map(location=[latitude, longitude], zoom_start=7)

    for lat, lng, name, categories in zip(nearby_medicalsupply['lat'], nearby_medicalsupply['lng'],nearby_medicalsupply['name'], nearby_medicalsupply['categories']):
        label = '{}'.format(name)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='red',
            fill=True,
            fill_color='red',
            fill_opacity=0.7,
            parse_html=False).add_to(Tamilnadu_map)

    Tamilnadu_map
```

Figure: Passing the latitude and longitude values of Medical Supply that should we marked.

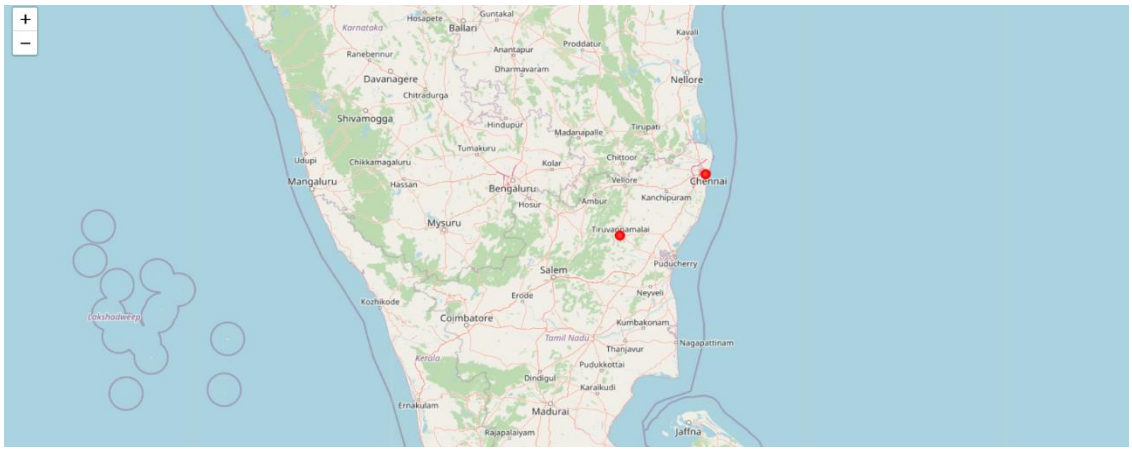


Figure: Map with the available Medical Supply Centers marked on it.

Map for the available Medical Labs in the given radius

```
[157] Tamilnadu_map = folium.Map(location=[latitude, longitude], zoom_start=7)

    for lat, lng, name, categories in zip(nearby_medical_labs['lat'], nearby_medical_labs['lng'],nearby_medical_labs['name'], nearby_medical_labs['categories']):
        label = '{}'.format(name)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='blue',
            fill=True,
            fill_color='blue',
            fill_opacity=0.7,
            parse_html=False).add_to(Tamilnadu_map)

    Tamilnadu_map
```

Figure: Passing the latitude and longitude values of Medical Labs that should we marked.

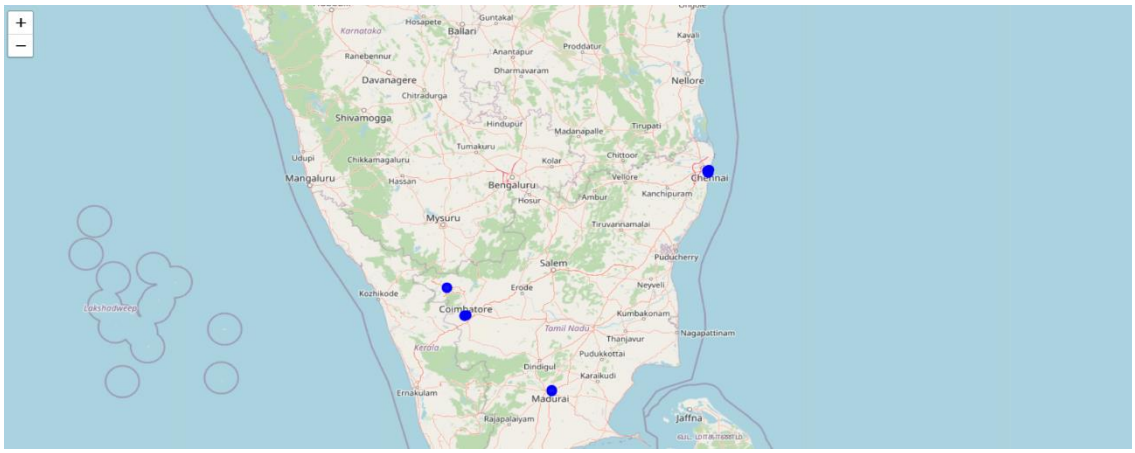


Figure: Map with the available Medical Labs marked on it.

Map for the available Emergency centers in the given radius

```
[150] Tamilnadu_map = folium.Map(location=[latitude, longitude], zoom_start=7)

    for lat, lng, name, categories in zip(nearby_emergency_centres['lat'], nearby_emergency_centres['lng'],nearby_emergency_centres['name'], nearby_emergency_centres['categories']):
        label = '{}'.format(name)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='red',
            fill=True,
            fill_color='red',
            fill_opacity=0.7,
            parse_html=False).add_to(Tamilnadu_map)

    Tamilnadu_map
```

Figure: Passing the latitude, longitude values of Emergency centers that should we marked.

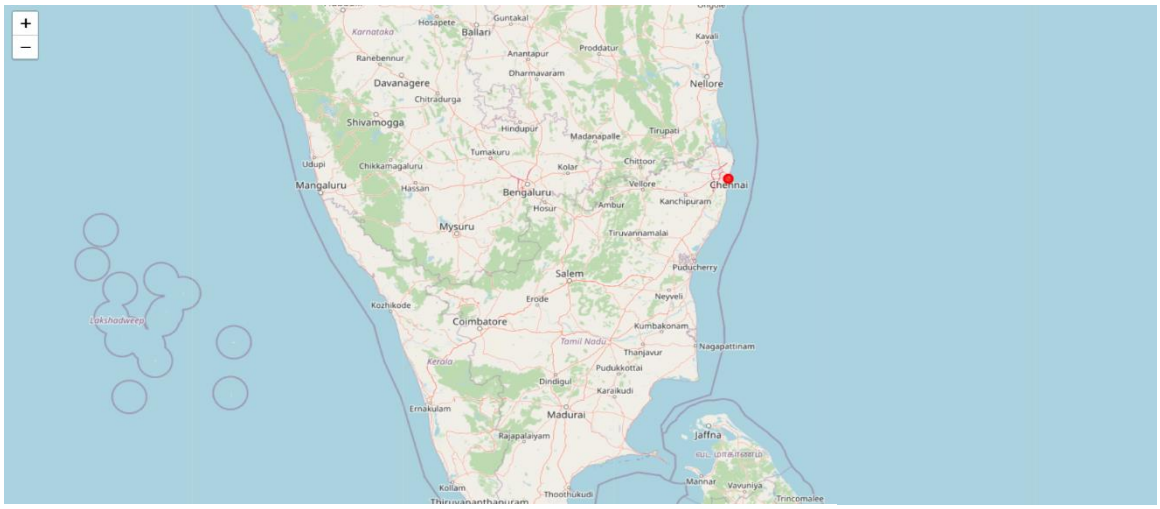


Figure: Map with the available Emergency Centers marked on it.

Map for the available Hospitals, Medical Labs, Medical Supply, Emergency centers in the given radius

```
[146] Tamilnadu_map = folium.Map(location=[latitude,longitude], zoom_start=7)

    for lat, lng, name, categories in zip(df_Tamilnadu_Health_Centers['lat'], df_Tamilnadu_Health_Centers['lng'],df_Tamilnadu_Health_Centers['name'], df_Tamilnadu_Health_Centers['categori
        label = '{}'.format(name)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='red',
            fill=True,
            fill_color='red',
            fill_opacity=0.7,
            parse_html=False).add_to(Tamilnadu_map)

    Tamilnadu_map
```

Figure: Passing the latitude, longitude values of all the four places that should we marked.

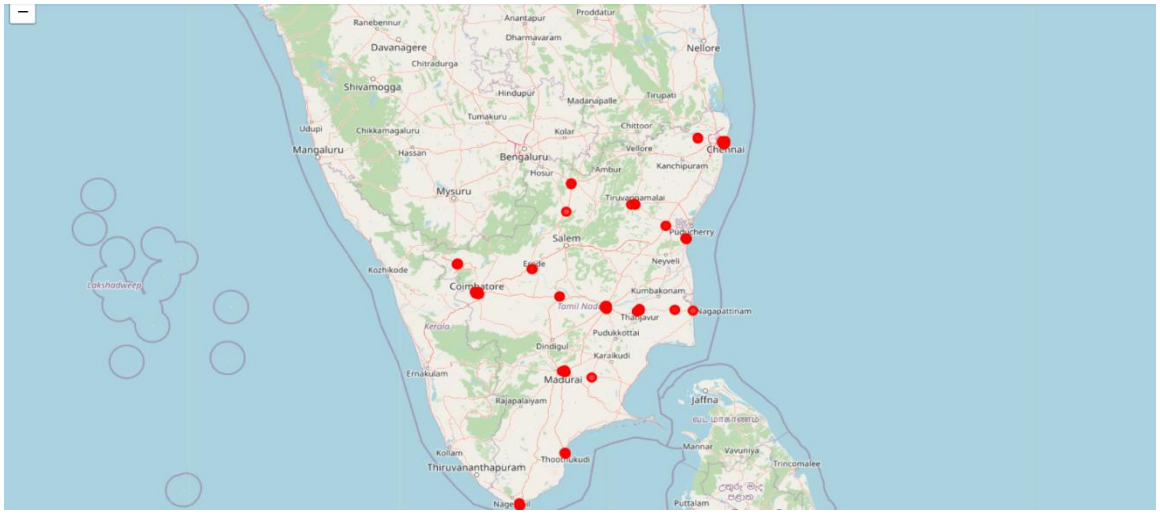


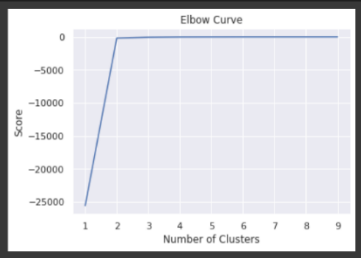
Figure: Map with all the available four places marked on it.

Now as we have seen from the trends above the facilities in not spread evenly across the whole state of Tamilnadu therefore, clustering this data will be a more feasibly option and this can we done using k-means algorithm and the value for k can be found out using the eblow curve. The elbow curve for this case looks like the one given below.

K-Means

```
[152] df_kmeans=df_Tamilnadu_Health_Centers

[153] K_clusters = range(1,10)
      kmeans = [KMeans(n_clusters=i) for i in K_clusters]
      Y_axis = df_kmeans[['lat']]
      X_axis = df_kmeans[['lng']]
      score = [kmeans[i].fit(Y_axis).score(Y_axis) for i in range(len(kmeans))]
      # Visualize
      plt.plot(K_clusters, score)
      plt.xlabel('Number of Clusters')
      plt.ylabel('Score')
      plt.title('Elbow Curve')
      plt.show()
```



From the above graph for elbow method 2 clusters is ideal

The two images below show the final dataframe with the cluster lable for each of the four available facilities in Tamilnadu and as to which will we the most feasible and the nearest option for the affected patients.

```
kmeans = KMeans(n_clusters = 2, init ='k-means++')
kmeans.fit(df_kmeans[df_kmeans.columns[3:5]]) # Compute k-means clustering.
df_kmeans['cluster_label'] = kmeans.fit_predict(df_kmeans[df_kmeans.columns[2:4]])
centers = kmeans.cluster_centers_ # Coordinates of cluster centers.
labels = kmeans.predict(df_kmeans[df_kmeans.columns[2:4]]) # Labels of each point
df_kmeans
```

	name	categories	lat	lng	cluster_label
0	Rajiv Gandhi Government General Hospital	Hospital	13.081822	80.276719	0
1	Apollo Hospitals	Hospital	13.062761	80.251790	0
2	Dr. U Mohan Rau Memorial Hospital	Hospital	13.079339	80.257258	0
3	Apollo First Med Hospitals	Hospital	13.077976	80.246140	0
4	Apollo hospitals	Hospital	13.062767	80.251862	0
5	Apollo Hospital	Hospital	13.066777	80.252211	0
6	Kumaran Hospital	Hospital	13.078651	80.249352	0
7	Apollo Heart Centre	Hospital	13.060618	80.254335	0
8	Sankara Nethralaya Eye Hospital	Hospital	13.066993	80.250306	0
9	Shankara Netralaya	Hospital	13.065819	80.250147	0
10	Apollo Childrens Hospital	Hospital	13.060041	80.252538	0
11	Seethapathy Clinic and Hospital	Hospital	13.048841	80.264537	0
12	Apollo Cradle Royale	Hospital	13.059902	80.252626	0
13	Apollo Hospital in Chennai	Hospital	13.056570	80.254907	0
14	KMC	Hospital	13.077061	80.242165	0
15	GG Hospital	Hospital	13.055364	80.248089	0

52	Web Legend	Office	9.936641	78.096485	0
53	Nagapattinam Govt hospital	Hospital	10.772499	79.843770	0
54	Ooty Government Hospital	Hospital	11.407698	76.700503	0
55	Vijaya Hospital	Hospital	11.407398	76.708143	0
56	S.M.Hospital	Hospital	11.411125	76.708649	0
57	S.M. hospital	Hospital	11.409166	76.711990	0
58	Salem Hospital - Family Birth Center	Hospital	44.932941	-123.032889	1
59	Drug & Alcohol Rehab Salem	Hospital	44.942883	-123.032394	1
60	Waddell & Reed Inc	Hospital	44.940114	-123.040481	1
61	Salem Hospital Surgery Center	Hospital	44.932972	-123.033646	1
62	Salem Hospital	Hospital	44.932932	-123.034122	1
63	Salem Hospital Building B	Hospital	44.932955	-123.034816	1
64	Building C - Salem Hospital	Hospital	44.932842	-123.034036	1
65	Salem Hospital OR	Hospital	44.932664	-123.033299	1

Discussion:

- As we have seen so far, I have taken the problem the medical amenities not being made available to everyone in this pandemic situation.
- I have made a study of all the facilities that are available in the state as a whole and made a visualization of it which will give an insight on which are the parts of the state where these facilities are available/not available.
- From this the government can take in steps to improve the facilities in the regions of the state where there are limited facilities and improve the present status of the medical support in those places which are deprived of it.

Conclusion:

- To improve this situation of the decentralized people getting deprived of the facilities can we minimized by seeing this study and bringing in temporary medical camps in the places where people are not able to gain access to the medical facilities.
- Now it is up to the government to step up and take a move in bringing a change to this current situation that is prevailing.
- I really hope that this study can help in making the basic medical facilities made available to all the people during this difficult phase of pandemic that we are going through right now.