Week 2 - Battle of the Neighbourhoods – Viewing Toronto

Introduction -The Problem

Toronto is a very varied city that has some great places to live in. The video linked below is looking at 140 neighbourhoods from a magazine 'Toronto Life' and compares various areas to live.

https://youtu.be/7RF1yl29Jms

https://torontolife.com/neighbourhood-rankings/#

The problem now at the start of June 2020 is we have a Corona 19 Virus which is a Pandemic – world wide. So the problem we need to solve is: Where are the rate of infections the lowest? Also as in many places – (if anything like the UK) - are closed so I want to find areas where there are parks to walk and exercise. The problem therefore is not looking at the Toronto Life magazine neighbourhood rankings but looking at low risk areas which have nearby parks.

How to tackle the problem and what data to use:

The first thing to do is to get a list of the Neighbourhoods. I produced a table (data frame) using https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

However only 10 Neighbourhood name match the about list as taken from the Open Data Toronto dataset. https://open.toronto.ca/dataset/neighbourhoods/

So this was used instead. It is difficult to clean so it can be used, Together with

https://drive.google.com/file/d/1euhrML0rkV_hHF1thiA0G5vSSeZCqxHY/view

to get the Neighbour hood rate of infections. I then produce a bubble plot of the infected areas. The next step was to find two houses, for sale, to choose in different neighbourhoods. Both being in low infected areas These where:

88 Boulton Ave, Toronto

and

489 Merton St, Toronto

The geocoder recommended does no work all the time so I used https://www.whatsmygps.com as we only have two locations to find to get the GPS coordinates.

Next I used the Fouresquare API https://foursquare.com/city-guide to find the parks around both locations.

K-means clustering may then be further used to investigate various groups.

Methology

We started with the Wikipedia list of neighbourhoods and post codes and then added the GPS coordinates :

```
In [14]: df = pd.merge(df, df2, on='Post Code', how='outer')
         df.head()
```

Out[14]:

	Post Code	Borough	Neighbourhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494

Now we look at the c COVID19 data:

```
In [15]: df3 = pd.read_csv('/kaggle/input/toronto2/COVID-19Neighbourh.csv')
         df3.head()
```

Out[15]:

	Neighbourhood Name	Rate per 100,000 people
0	Yorkdale-Glen Park	763.307214
1	York University Heights	1021.998333
2	Yonge-St.Clair	151.660281
3	Yonge-Eglinton	101.548616
4	Wychwood	459.962367

and change the header names to match the neighbourhood data

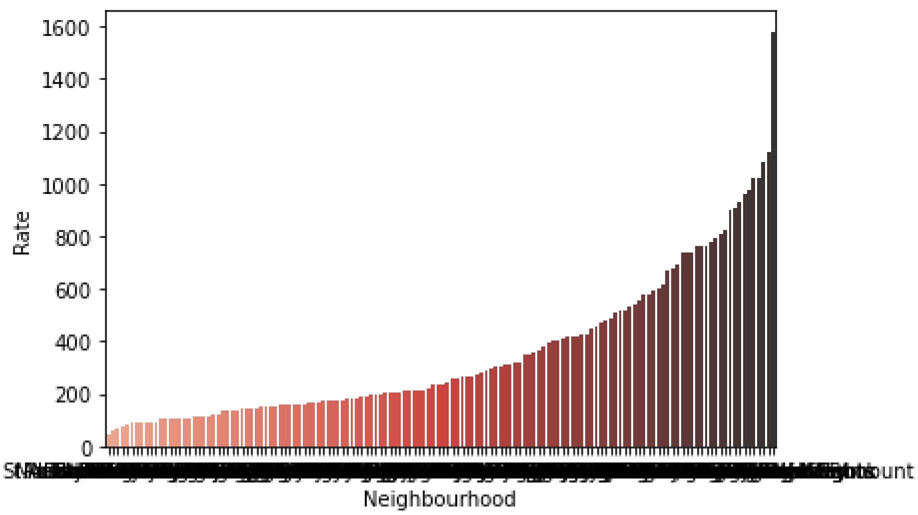
In [16]:
```
df3 = df3.rename(columns={"Neighbourhood Name": "Neighbourhood"}) #to match df name
```

In [17]:
```
df3 = df3.rename(columns={"Rate per 100,000 people": "Rate"}) #to match df name
```

In [18]:
```
df3.head()
```

Out[18]:

	Neighbourhood	Rate
0	Yorkdale-Glen Park	763.307214
1	York University Heights	1021.998333
2	Yonge-St.Clair	151.660281
3	Yonge-Eglinton	101.548616
4	Wychwood	459.962367



This produces the Bar Plot above

Which looking at the head and tail of the data we have:

Out[20]:

	Neighbourhood	Rate
92	Guildwood	1018.453161
1	York University Heights	1021.998333
94	Glenfield-Jane Heights	1085.566233
13	Weston	1122.721209
86	Humber Heights-Westmount	1580.197296

In [21]:

```
result.head()
```

Out[21]:

	Neighbourhood	Rate
28	St.Andrew-Windfields	44.913541
23	The Beaches	60.277275
57	Mount Pleasant East	65.573770
111	Danforth	72.418787
35	Rosedale-Moore Park	81.250299

So the rate per
100,000 of infection varies between 81.25 and 1018.45.

When we plot the COVID19 data we have the wide ranging bar plot. The problem now is that only 10 of the names of the Covid data match the original neighbourhood data so we then load the https://open.toronto.ca/dataset/neighbourhoods/ and clean the data and check for missing data a row at a time to produce a folium map.

Changing the header names to match both data sets we can then merge the two set together as shown below:

32]:

```
df4 = pd.merge(df4, df3, on='Neighbourhood', how='outer')
# df4 = df4.apply (pd.to_numeric, errors='coerce')

#df4 = df4.dropna()
df4
```

32]:

	AREA_ID	AREA_ATTR_ID	PARENT_AREA_ID	AREA_SHORT_CODE	AREA_LONG_CODE	Neighbourhood	AREA_
0	25886861.0	25926662.0	49885.0	94.0	94.0	Wychwood (	Wychw (94)

This gives 173 x 14 grid size.

Use a coordinates of Toronto `T_COORDINATES = (43.651070, -79.347015)`

we then can produce a folium map

```
# I can add marker one by one on the map
import math
for i in range(0,len(mapdata)):
        location=[mapdata.iloc[i]['Latitude'], mapdata.iloc[i]['Longitude']]
        # print (location[0])
        if (location[0] > 20):
            rate = mapdata.iloc[i]['Rate']
            if (math.isnan(rate)):
                #print(rate)
                rate = 1
            #print (location)
            # print(rate)
            folium.Circle(
              location=[mapdata.iloc[i]['Latitude'], mapdata.iloc[i]['Longitude']]
              popup=mapdata.iloc[i]['Neighbourhood'],
              # radius=mapdata.iloc[i]['Rate']/1,
              radius = rate,
              color='crimson',
              fill=True,
              fill_color='crimson'
       ).add_to(venues_map)
#print(df4.shape)
```

Note we cannot get rid of all the NaN’s in one hit so we have to check thaere is data in each row using the math.isnan Python function.

The next step is to use the FourSquare API to look for park areas around the first address: `address = '88 Boulton Ave, Toronto'`

This was simply found randomly looking for properties for sale in a low risk area.

This gives 30 locations of parks within waking distance from the property

]:

	name	categories	address	lat	lng	labeledLatLngs	distance	postalCode	c
0	Riverdale Park East	Park	550 Broadview Ave	43.669951	-79.355493	[{'label': 'display', 'lat': 43.66995145540962...	991	M4K 2P1	(
1	Jimmie Simpson Park	Park	870 Queen St E	43.659230	-79.345063	[{'label': 'display', 'lat': 43.65922968838468...	468	M4M 3G9	(
2	Riverdale Park West	Park	500 Gerrard St.	43.666048	-79.360941	[{'label': 'display', 'lat': 43.66604827676354...	1089	M5A 2H3	(
3	John Chang Neighbourhood Park	Park	NaN	43.662245	-79.342200	[{'label': 'display', 'lat': 43.662245, 'lng':...	485	NaN	(
4	Withrow Park Off Leash Dog Park	Dog Run	Logan Ave	43.674019	-79.346764	[{'label': 'display', 'lat': 43.67401899043838...	1255	NaN	(
5	Dog Park	Dog Run	Audley Avenue	43.664785	-79.337999	[{'label': 'display', 'lat': 43.66478470338474...	849	M4m1v9	(
6	DeGrassi Street Park	Park	NaN	43.662681	-79.347560	[{'label': 'display', 'lat': 43.66268101995241...	51	NaN	(

This was repeated for the next address which was:

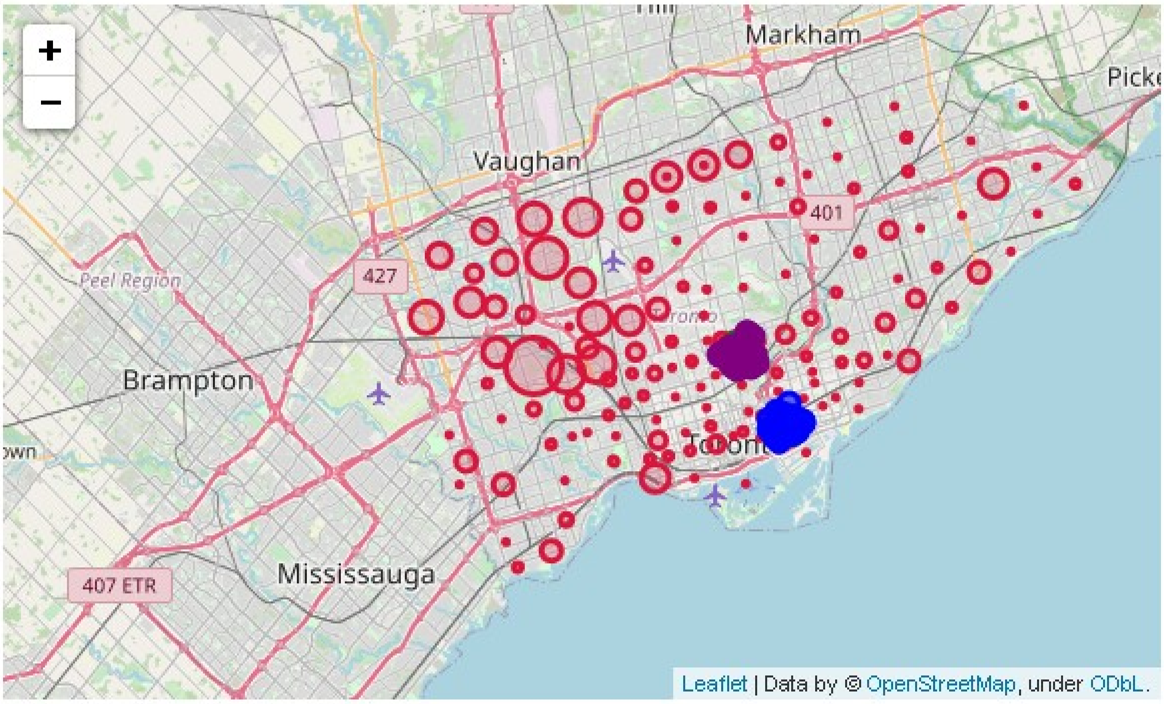
```
address = '489 Merton St, Toronto'
```

This gave 23 places.

53]:

	name	categories	address	crossStreet	lat	lng	labeledLatLngs	distance	cc	city
0	June Rowlands Park	Park	220 Davisville Ave	btwn Mt Pleasant Rd & Acacia Rd	43.700517	-79.389189	[{'label': 'display', 'lat': 43.70051698238234...	623	CA	Toror
1	Sandy Bruce Dog Park	Dog Run	Bayview and Moore	NaN	43.697571	-79.371043	[{'label': 'display', 'lat': 43.69757053841549...	874	CA	Toror
2	Mount Pleasant Parkette	Park	441 Mount Pleasant Rd	at Davisville An	43.700001	-79.386830	[{'label': 'display', 'lat': 43.70000099187981...	425	CA	Toror
3	Moorevale Park	Playground	Moore Ave.	at Welland Ave.	43.693610	-79.383465	[{'label': 'display', 'lat': 43.69360987933692...	612	CA	Toror
4	Moore Park Ravine	Park	205 Moore Ave	NaN	43.691879	-79.373667	[{'label': 'display', 'lat': 43.69187857154135...	1022	CA	Toror
5	Moore Park	Neighborhood	NaN	NaN	43.693001	-79.387743	[{'label': 'display', 'lat': 43.69300097466770...	821	CA	Toror
6	Park Between 99 Davisville & 118 Balliol	Park	NaN	NaN	43.698589	-79.392853	[{'label': 'display', 'lat': 43.698589, 'lng':...	895	CA	Toror

Now we can combine the three maps to give:



The red bubble plot is the extent of the Covid-19 virus – the largest blobs are where there are most cases. The purple circles are the sparks around second property *489 Merton St, Toronto and the blue dots the parks around the first property 489 Merton St, Toronto*

Results

Clearly from he Covid-19 infection data it can be seen that the majory of infections are clustered towards the North West of the city. Also both addresses have parks surrounding them.

Discussion

As most eating and drinking places could be closed it makes sense to me to look for Park Areas so as to avoid crowds. This also give a pleasant place to live. We guessed at two addresses. One has 23 places and the other 30. Also riverside is nearer the sea. We could look for other types of places around each area before commuting.

1.

Conclusion

I think the Foursqaure API is useful to get various places close to an address. Also Folium is excellent for map drawing. The bubble plot clearly shows the different sizes of infection. What surprised me was the huge difference in Infection rates between neighbourhoods. Also the number of parks in a City is larger than I expected.