Applied Data Science Capstone Week 4 Peer-Graded Assignment

The Battle of Neighborhoods

Introduction to the problem

New Brunswick's capital city is rich in culture, history and riverside beauty. Stand in awe at the newly expanded Beaverbrook Art Gallery; discover Fredericton's past in the Historic Garrison District; tap your toes at the award-winning Harvest Jazz & Blues Festival; walk or bike over 115 km of riverside trails; explore the Saint John River by kayak or canoe. Craft beer enthusiast? Fredericton boasts the highest concentration of craft breweries and tasting .experiences in the Maritimes

:We have an interest in answering the following questions regarding the city of Fredericton

- ?What neighbourhoods have the highest crime
- ?Is population density correlated to crime level
- ?Using Foursquare data, what venues are most common in different locations within the city
- ?Does the Knowledge Park really need a coffee shop

Does the Open Data project have specific enough or thick enough data to empower decisions to be made or is it too aggregate to provide value in its current .detail? Let's find out

:[5] In

```
mport requests
rom PIL import Image
rl = 'https://www.tourismnewbrunswick.ca/-/media/Images/Website/Products/R/RegentStreetWharf/City_of_Fredericton2013.asl
m = Image.open(requests.get(url, stream=True).raw)
m
```

Out[5]:

Data

¶:To understand and explore we will need the following City of Fredericton Open Data

- /Open Data Site: http://data-fredericton.opendata.arcgis.com
- Fredericton Neighbourhoods: http://data-fredericton.opendata.arcgis.com/datasets/neighbourhoods--quartiers
- Fredericton Crime by Neighbourhood: http://data-fredericton.opendata.arcgis.com/datasets/crime-by-neighbourhood-2017--crime-par-quartier-2017 Fredericton Census Tract Demographics: http://data-fredericton.opendata.arcgis.com/datasets/census-tract-demographics--donn%C3%A9es-d%C3%A9mographiques-du-secteur-de-recensement Fredericton locations of interest: https://github.com/JasonLUrquhart/Applied-Data-Science-Capstone/blob/master/Fredericton%20Locations.xlsx

Using this data will allow exploration and examination to answer the questions. The neighbourhood data will enable us to properly group crime by neighbourhood. The Census data will enable us to then compare the population density to examine if areas of highest crime are also most densely populated. .Fredericton locations of interest will then allow us to cluster and quantitatively understand the venues most common to that location

Methodology

.All steps are referenced beleow in the Appendix: Analysis section

:The methodology will include

- Examine the crime frequency by neighbourhood
- Study the crime types and then pivot analysis of crime type frequency by neighbourhood
- nderstand correlation between crimes and population density
- Perform k-means statisical analysis on venues by locations of interest based on findings from crimes and neighbourhood
- .Determine which venues are most common statistically in the region of greatest crime count then in all other locations of interest
- .Determine if an area, such as the Knowledge Park needs a coffee shop

Loading the data

After loading the applicable libraries, the referenced geojson neighbourhood data was loaded from the City of Fredericton Open Data site. This dataset uses block polygon shape coordinates which are better for visualization and comparison. The City also uses Ward data but the Neighbourhood location data is .more accurate and includes more details. The same type of dataset was then loaded for the population density from the Stats Canada Census tracts

The third dataset, an excel file, "Crime by Neighbourhood 2017" downloaded from the City of Fredericton Open Data site is found under the Public Safety domain. This dataset was then uploaded for the analysis. It's interesting to note the details of this dataset are aggregated by neighbourhood. It is not an .exhaustive set by not including all crimes (violent offenses) nor specific location data of the crime but is referenced by neighbourhood

This means we can gain an understanding of the crime volume by type by area but not specific enough to understand the distribution properties. Valuable questions such as, "are these crimes occuring more often in a specific area and at a certain time by a specific demographic of people?" cannot be answered .nor explored due to what is reasonably assumed to be personal and private information with associated legal risks

There is value to the city to explore the detailed crime data using data science to predict frequency, location, timing and conditions to best allocated resources for the benefit of its citizens and it's police force. However, human behaviour is complex requiring thick profile data by individual and the conditions .surrounding the event(s). To be sufficient for reliable future prediction it would need to demonstrate validity, currency, reliability and sufficiency

Exploring the data

Exploring the count of crimes by neighbourhood gives us the first glimpse into the distribution.

One note is the possibility neighbourhoods names could change at different times. The crime dataset did not mention which specific neighbourhood naming dataset it was using but we assumed the neighbourhood data provided aligned with the neighbourhoods used in the crime data. It may be beneficial for the City to note and timestamp neighbourhood naming in the future or simply reference with neighbourhood naming file it used for the crime dataset.

An example of data errors: There was an error found in the naming of the neighbourhood "Platt". The neighbourhood data stated "Plat" while the crime data stated "Platt". Given the crime dataset was most simple to manipulate it was modified to "Plat". The true name of the neighbourhood is "Platt".

First Visualization of Crime

Once the data was prepared, a choropleth map was created to view the crime count by neighbourhood. As expected the region of greatest crime count was found in the downtown and Platt neighbourhoods.

Examining the crime types enables us to learn the most frequent occuring crimes which we then plot as a bar chart to see most frequenty type.

Theft from motor vehicles is most prevalent in the same area as the most frequent crimes. It's interesting to note this area is mostly residential and most do not have garages. It would be interesting to further examine if surveillance is a deterant for motor vehicle crimes in the downtown core compared to low surveillance in the Platt neighbourhood.

Examining 2nd most common crime given it is specific: theft from vehicles

After exploring the pivot table showing Crime_Type by Neighbourhood, we drill into a specific type of crime, theft from vehicles and plot the choropleth map to see which area has the greatest frequency.

Again, the Platt neighbourhood appears as the most frequent.

Is this due to population density?

Introducing the Census data to explore the correlation between crime frequency and population density.

Visualising the population density enables us to determine that the Platt neighbourhood has lower correlation to crime frequency than I would have expected.

It would be interesting to further study the Census data and if this captures the population that is renting or more temporary/transient poplution, given the City is a University hub.

Look at specific locations to understand the connection to venues using Foursquare data

Loading the "Fredericton Locations" data enables us to perform a statistical analysis on the most common venues by location.

We might wonder if the prevalence of bars and clubs in the downtown region has something to do with the higher crime rate in the near Platt region.

Plotting the latitude and longitude coordinates of the locations of interest onto the crime choropleth map enables us to now study the most common venues by using the Foursquare data.

Analysing each Location

Grouping rows by location and the mean of the frequency of occurance of each category we venue categories we study the top five most common venues.

Putting this data into a pandas dataframe we can then determine the most common venues by location and plot onto a map.

Results

The analysis enabled us to discover and describe visually and quantitatively:

- Neighbourhoods in Fredericton

- Crime freqency by neighbourhood
- Crime type frequency and statistics. The mean crime count in the City of Fredericton is 22.
- Crime type count by neighbourhood. Theft from motor vehicles is most prevalent in the same area as the most frequent crimes. It's interesting to note this area is mostly residential and most do not have garages. It would be interesting to further examine if surveillance is a deterant for motor vehicle crimes in the downtown core compared to low surveillance in the Platt neighbourhood.
- Motor Vehicle crimes less than $5000 analysis by neighbourhood and resulting statistics. The most common crime is Other Theft less than 5k followed by Motor Vehicle Theft less than 5k. There is a mean of 6 motor vehicle thefts less than 5k by neighbourhood in the City.
- That population density and resulting visual correlation is not strongly correlated to crime frequency. Causation for crime is not able to be determined given lack of open data specificity by individual and environment.
- Using k-menas, we were able to determine the top 10 most common venues within a 1 km radius of the centroid of the highest crime neighbourhood. The most common venues in the highest crime neighbourhood are coffee shops followed by Pubs and Bars.

While, it is not valid, consistent, reliable or sufficient to assume a higher concentration of the combination of coffee shops, bars and clubs predicts the amount of crime occurance in the City of Fredericton, this may be a part of the model needed to be able to in the future.

- We were able to determine the top 10 most common venues by location of interest.
- Statisically, we determined there are no coffee shops within the Knowledge Park clusters.

Discussion and Recommendations

The City of Fredericton Open Data enables us to gain an understanding of the crime volume by type by area but not specific enough to understand the distribution properties. Valuable questions such as, "are these crimes occuring more often in a specific area and at a certain time by a specific demographic of people?" cannot be answered nor explored due to what is reasonably assumed to be personal and private information with associated legal risks.

There is value to the city to explore the detailed crime data using data science to predict frequency, location, timing and conditions to best allocated resources for the benefit of its citizens and it's police force. However, human behaviour is complex requiring thick profile data by individual and the conditions surrounding the event(s). To be sufficient for reliable future prediction it would need to demonstrate validity, currency, reliability and sufficiency.

A note of caution is the possibility neighbourhoods names could change. The crime dataset did not mention which specific neighbourhood naming dataset it was using but we assumed the neighbourhood data provided aligned with the neighbourhoods used in the crime data. It may be beneficial for the City to note and timestamp neighbourhood naming in the future or simply reference with neighbourhood naming file it used for the crime dataset.

Errors exist in the current open data. An error was found in the naming of the neighbourhood "Platt". The neighbourhood data stated "Plat" while the crime data stated "Platt". Given the crime dataset was most simple to manipulate it was modified to "Plat". The true name of the neighbourhood is "Platt".

Theft from motor vehicles is most prevalent in the same area as the most frequent crimes. It is interesting to note this area is mostly residential and most do not have garages. It would be interesting to further examine if surveillance is a deterant for motor vehicle crimes in the downtown core compared to low surveillance in the Platt neighbourhood.

It would be interesting to further study the Census data and if this captures the population that is renting or more temporary/transient poplution, given the City is a University hub.

Given the findings of the top 10 most frequent venues by locations of interest, the Knowledge Park does not have Coffee Shops in the top 10 most common venues as determined from the Foursquare dataset. Given this area has the greatest concentration of stores and shops as venues, it would be safe to assume .a coffee shop would be beneficial to the business community and the citizens of Fredericton

Conclusion

Using a combination of datasets from the City of Fredericton Open Data project and Foursquare venue data we were able to analyse, discover and describe .neighbhourhoods, crime, population density and statistically describe quantitatively venues by locations of interest

While overall, the City of Fredericton Open Data is interesting, it misses the details required for true valued quantitiatve analysis and predictive analytics which .would be most valued by investors and developers to make appropriate investments and to minimize risk

The Open Data project is a great start and empowers the need for a "Citizens Like Me" model to be developed where citizens of digital Fredericton are able to .share their data as they wish for detailed analysis that enables the creation of valued services

APPENDIX: Analysis

Load Libraries

:[1] In

```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

# for webscraping import Beautiful Soup
from bs4 import BeautifulSoup

import xml

!conda install -c conda-forge folium=0.5.0 --yes
import folium # map rendering library

print('Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /opt/anaconda3

:added / updated specs
geopy -


:The following packages will be downloaded

package                    |            build
---------------------------|-----------------
geographiclib-1.50         |             py_0          34 KB  conda-forge
geopy-1.21.0               |             py_0          58 KB  conda-forge
------------------------------------------------------------
                                       Total:          92 KB

:The following NEW packages will be INSTALLED

geographiclib      conda-forge/noarch::geographiclib-1.50-py_0
geopy              conda-forge/noarch::geopy-1.21.0-py_0



Downloading and Extracting Packages
 geopy-1.21.0         | 58 KB     | #################################### | 100%
 geographiclib-1.50   | 34 KB     | #################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Collecting package metadata (current_repodata.json): done
.Solving environment: failed with initial frozen solve. Retrying with flexible solve
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /opt/anaconda3

:added / updated specs
folium=0.5.0 -


:The following packages will be downloaded

package                    |            build
---------------------------|-----------------
altair-4.1.0               |             py_1         614 KB  conda-forge
certifi-2019.11.28         |           py37_0         148 KB  conda-forge
folium-0.5.0               |             py_0          45 KB  conda-forge
------------------------------------------------------------
                                       Total:         808 KB

:The following NEW packages will be INSTALLED

altair             conda-forge/noarch::altair-4.1.0-py_1

:The following packages will be SUPERSEDED by a higher-priority channel

certifi                                         pkgs/main --> conda-forge

:The following packages will be DOWNGRADED

folium                                      0.10.1-py_0 --> 0.5.0-py_0



Downloading and Extracting Packages
 altair-4.1.0         | 614 KB    | #################################### | 100%
```

```
 certifi-2019.11.28   | 148 KB    | ##################################### | 100%
 folium-0.5.0         | 45 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
.Libraries imported
```

:[3] In
```
r = requests.get('https://opendata.arcgis.com/datasets/823d86e17a6d47808c6e4f1c2dd97928_0.geojson')
fredericton_geo = r.json()
```

:[4] In
```
neighborhoods_data = fredericton_geo['features']
```

:[5] In
```
neighborhoods_data[0]
```


```
,[45.9350997354041 ,66.6591442982811-]
,[45.934818595486 ,66.6601863440107-]
,[45.9340751297235 ,66.6616146653125-]
,[45.9341813397283 ,66.6619290756619-]
,[45.9327880982037 ,66.6631238349898-]
,[45.932475707408 ,66.6631867170597-]
,[45.9322757763752 ,66.6629172224744-]
,[45.9309387190672 ,66.6599797314954-]
,[45.9292704762017 ,66.6557756159657-]
,[[45.9291455121693 ,66.6550120479742-]
,[45.938648223393 ,66.6934150263703-]]
,[45.9418404190785 ,66.6899565125264-]
,[45.9443827996167 ,66.6889683657139-]
,[45.9444702504357 ,66.6890312477838-]
,[45.9445014828376 ,66.6899475293736-]
,[45.9449262417569 ,66.6912141539242-]
,[45.9467626619838 ,66.6939180829294-]
,[45.9422339647247 ,66.7001973067654-]
{{[[[45.938648223393 ,66.6934150263703-]
```


:[6] In
```
g = requests.get('https://opendata.arcgis.com/datasets/6179d35eacb144a5b5fdcc869f86dfb5_0.geojson')
demog_geo = g.json()
```

:[7] In
```
demog_data = demog_geo['features']
demog_data[0]
```

Out[7]:


```
,'type': 'Feature'}
,properties': {'FID': 1'
,OBJECTID': 501'
,'DBUID': '1310024304'
,'DAUID': '13100243'
,'CDUID': '1310'
,'CTUID': '3200002.00'
,'CTNAME': '0002.00'
,'DBuid_1': '1310024304'
,DBpop2011': 60'
,DBtdwell20': 25'
,DBurdwell2': 22'
,Shape_Leng': 0.00746165241824'
,Shape_Area': 2.81310751889e-06'
,CTIDLINK': 3200002'
,Shape__Area': 50139.15234375'
,{Shape__Length': 929.421025287179'
,'geometry': {'type': 'Polygon'
,coordinates': [[[-66.634784212921, 45.9519239912381]'
,[45.9507605156138 ,66.6351046935752-]
,[45.9510868696778 ,66.6378263667982-]
,[45.9521037018384 ,66.636944377136-]
{{[[[45.9519239912381 ,66.634784212921-]
```


:[16] In
```
import os
```

:[17] In
```
opencrime = 'Crime_by_neighbourhood_2017.xlsx'
```

:[18] In
```
workbook = pd.ExcelFile(opencrime)
print(workbook.sheet_names)
```

```
['Worksheet']
```

:[19] In
```
crime_df = workbook.parse('Worksheet')
crime_df.head()
```

Out[19]:

FID	City	Ward	Crime_Type	Crime_Code	To_Date	From_Date	Neighbourhood	
1	Fredericton	7	B&E NON-RESIDNCE	2120	1.485389e+12	1.483574e+12	Fredericton South	**0**
2	Fredericton	7	B&E NON-RESIDNCE	2120	1.488758e+12	1.488586e+12	Fredericton South	**1**
3	Fredericton	12	B&E NON-RESIDNCE	2120	NaN	1.494115e+12	Fredericton South	**2**
4	Fredericton	12	B&E NON-RESIDNCE	2120	1.498003e+12	1.497917e+12	Fredericton South	**3**
5	Fredericton	7	B&E NON-RESIDNCE	2120	1.499645e+12	1.499558e+12	Fredericton South	**4**

:[20] In
```
crime_df.drop(['From_Date', 'To_Date'], axis=1,inplace=True)
```

?What is the crime count by neighbourhood

:[21] In
```
crime_data = crime_df.groupby(['Neighbourhood']).size().to_frame(name='Count').reset_index()
crime_data
```

Out[21]:

Count	Neighbourhood	
47	Barkers Point	**0**
54	Brookside	**1**
9	Brookside Estates	**2**
5	Brookside Mini Home Park	**3**
41	College Hill	**4**
9	Colonial heights	**5**
4	Cotton Mill Creek	**6**
1	Diamond Street	**7**
1	Doak Road	**8**
3	Douglas	**9**
127	Downtown	**10**

:[22] In
```
crime_data.describe()
```

Out[22]:

Count	
66.000000	**count**
22.121212	**mean**
34.879359	**std**

1.000000	min
3.000000	25%
9.000000	50%
23.250000	75%
198.000000	max

:[23] In

```
crime_data.rename(index=str, columns={'Neighbourhood':'Neighbourh','Count':'Crime_Count'}, inplace=True)
crime_data
```

Out[23]:

Crime_Count	Neighbourh	
47	Barkers Point	0
54	Brookside	1
9	Brookside Estates	2
5	Brookside Mini Home Park	3
41	College Hill	4
9	Colonial heights	5
4	Cotton Mill Creek	6
1	Diamond Street	7
1	Doak Road	8
3	Douglas	9
127	Downtown	10

:[24] In

```
crime_data.rename({'Platt': 'Plat'},inplace=True)
crime_data.rename(index=str, columns={'Neighbourhood':'Neighbourh','Count':'Crime_Count'}, inplace=True)
crime_data
```

Out[24]:

Crime_Count	Neighbourh	
47	Barkers Point	0
54	Brookside	1
9	Brookside Estates	2
5	Brookside Mini Home Park	3
41	College Hill	4
9	Colonial heights	5
4	Cotton Mill Creek	6
1	Diamond Street	7
1	Doak Road	8
3	Douglas	9
127	Downtown	10

:[25] In

```
address = 'Fredericton, Canada'

geolocator = Nominatim()
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Fredericton, New Brunswick is {}, {}.'.format(latitude, longitude))
```

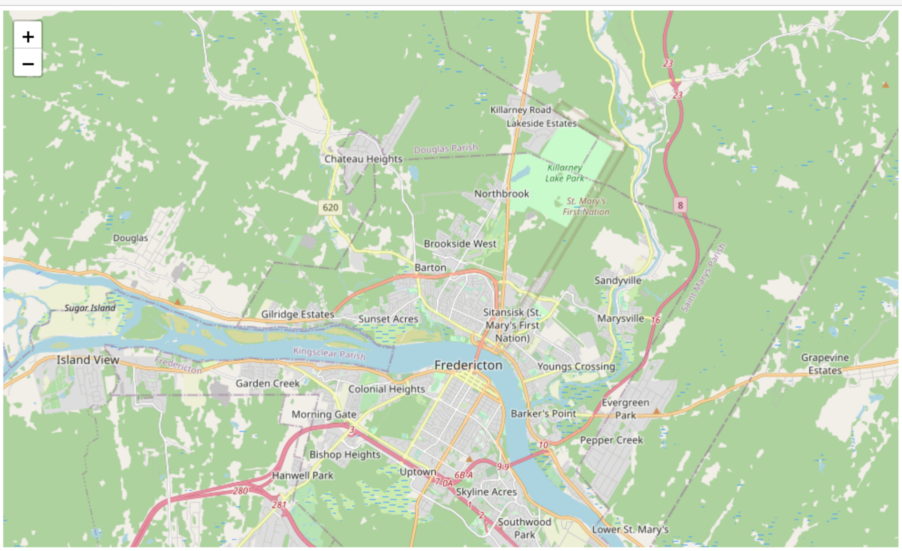
```
opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3: DeprecationWarning: Using Nominatim with the defa/
ult "geopy/1.21.0" `user_agent` is strongly discouraged, as it violates Nominatim's ToS https://operations.osmfoundat
.on.org/policies/nominatim/ and may possibly cause 403 and 429 HTTP errors. Please specify a custom `user_agent` with
`Nominatim(user_agent="my-application")` or by overriding the default `user_agent`: `geopy.geocoders.options.default_
.user_agent = "my-application"`. In geopy 2.0 this will become an exception
This is separate from the ipykernel package so we can avoid doing imports until
```

```
.The geograpical coordinate of Fredericton, New Brunswick is 45.966425, -66.645813
```

:[26] In

```
world_geo = r'world_countries.json' # geojson file

fredericton_1_map = folium.Map(location=[45.97, -66.65], width=1000, height=750,zoom_start=12)

fredericton_1_map
```

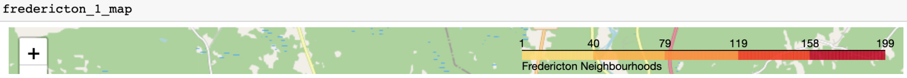
Out[26]:

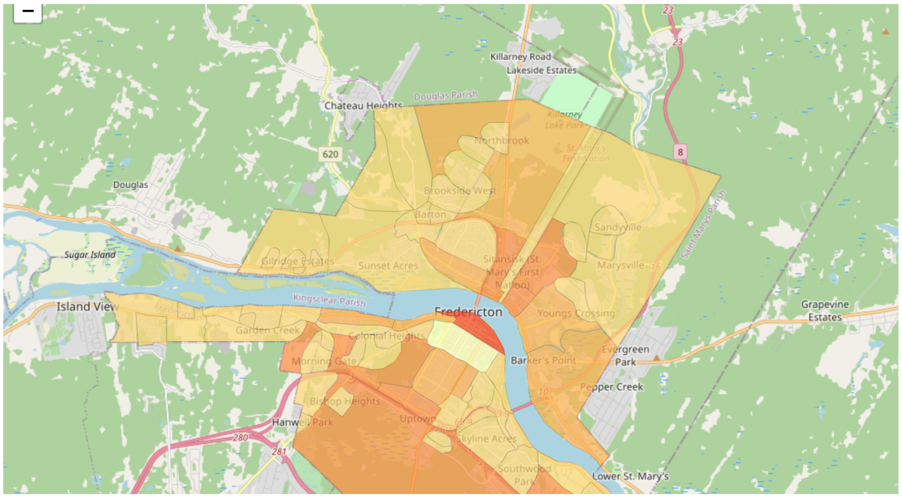


:[27] In

```
fredericton_geo = r.json()

threshold_scale = np.linspace(crime_data['Crime_Count'].min(),crime_data['Crime_Count'].max(), 6,dtype=int)
threshold_scale = threshold_scale.tolist()
threshold_scale[-1] = threshold_scale[-1]+1

fredericton_1_map.choropleth(geo_data=fredericton_geo, data=crime_data,columns=['Neighbourh', 'Crime_Count'],
    key_on='feature.properties.Neighbourh', threshold_scale=threshold_scale,fill_color='YlOrRd', fill_opacity=0.7,
    line_opacity=0.1, legend_name='Fredericton Neighbourhoods')

fredericton_1_map
```

Out[27]:





Examine Crime Types

```
crimetype_data = crime_df.groupby(['Crime_Type']).size().to_frame(name='Count').reset_index()
crimetype_data
```

:[28] In

Out[28]:

Count	Crime_Type	
4		0
5	ARSON	1
1	ARSON BY NEG	2
4	.ARSON-DAM.PROP	3
51	B&E NON-RESIDNCE	4
58	B&E OTHER	5
151	B&E RESIDENCE	6
3	B&E STEAL FIREAR	7
1	MISCHIEF OBS USE	8
246	MISCHIEF TO PROP	9
2	MISCHIEF-DATA	10
40	MOTOR VEH THEFT	11
63	THEFT BIKE<$5000	12
356	THEFT FROM MV < $5000	13
5	THEFT FROM MV > $5000	14
458	THEFT OTH <$5000	15
9	THEFT OTH >$5000	16
1	THEFT OVER $5000	17
2	THEFT,BIKE>$5000	18

```
crimetype_data.describe()
```

:[29] In

Out[29]:

Count	
19.000000	count
76.842105	mean
133.196706	std
1.000000	min
2.500000	25%
5.000000	50%
60.500000	75%
458.000000	max

```
crimepivot = crime_df.pivot_table(index='Neighbourhood', columns='Crime_Type', aggfunc=pd.Series.count, fill_value=0)
crimepivot
```

:[30] In

Out[30]:

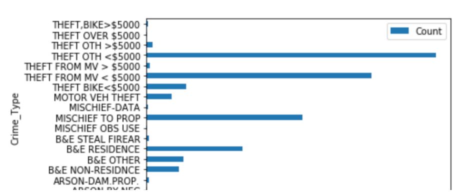
T I	THEFT FROM MV < $5000	THEFT BIKE<$5000	MOTOR VEH THEFT	MISCHIEF-DATA	MISCHIEF TO PROP	MISCHIEF OBS USE	B&E STEAL FIREAR	B&E RESIDENCE	B&E OTHER	B&E NON-RESIDNCE	ARSON-.DAM.PROP	ARSON BY NEG	ARSON		Crime_Type
															Neighbourhood
0	8	2	2	0	7	0	1	7	7	2	0	0	0	0	Barkers Point
0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	Brookside
0	3	0	1	0	1	0	0	0	1	1	0	0	0	0	Brookside Estates
0	0	0	1	0	3	0	1	0	0	0	0	0	0	0	Brookside Mini Home Park
0	10	2	0	0	4	0	0	13	2	0	0	0	2	0	College Hill
0	6	0	0	0	0	0	0	3	0	0	0	0	0	0	Colonial

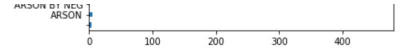
```
crimetype_data.plot(x='Crime_Type', y='Count', kind='barh')
```

:[31] In

Out[31]:

<matplotlib.axes._subplots.AxesSubplot at 0x1a1ba4c6d0>





Let's examine theft from vehicles

```
mvcrime_df = crime_df.loc[crime_df['Crime_Type'] == 'THEFT FROM MV < $5000']
mvcrime_df
```

:[32] In

Out[32]:

FID	City	Ward	Crime_Type	Crime_Code	Neighbourhood	
19	Fredericton	7	THEFT FROM MV < $5000	2142	Fredericton South	**18**
20	Fredericton	7	THEFT FROM MV < $5000	2142	Fredericton South	**19**
21	Fredericton	7	THEFT FROM MV < $5000	2142	Fredericton South	**20**
22	Fredericton	12	THEFT FROM MV < $5000	2142	Fredericton South	**21**
23	Fredericton	12	THEFT FROM MV < $5000	2142	Fredericton South	**22**
24	Fredericton	7	THEFT FROM MV < $5000	2142	Fredericton South	**23**
25	Fredericton	7	THEFT FROM MV < $5000	2142	Fredericton South	**24**
26	Fredericton	7	THEFT FROM MV < $5000	2142	Fredericton South	**25**
27	Fredericton	11	THEFT FROM MV < $5000	2142	Fredericton South	**26**
28	Fredericton	11	THEFT FROM MV < $5000	2142	Fredericton South	**27**
29	Fredericton	12	THEFT FROM MV < $5000	2142	Fredericton South	**28**

```
mvcrime_data = mvcrime_df.groupby(['Neighbourhood']).size().to_frame(name='Count').reset_index()
mvcrime_data
```

:[33] In

Out[33]:

Count	Neighbourhood	
8	Barkers Point	**0**
3	Brookside Estates	**1**
10	College Hill	**2**
6	Colonial heights	**3**
1	Diamond Street	**4**
1	Douglas	**5**
21	Downtown	**6**
9	Dun's Crossing	**7**
8	Forest Hill	**8**
20	Fredericton South	**9**
12	Fulton Heights	**10**
1	Garden Creek	**11**
3	Garden Place	**12**
1	Gilridge Estates	**13**
5	Golf Club	**14**
3	Hanwell North	**15**
2	Heron Springs	**16**
4	Highpoint Ridge	**17**
1	Knob Hill	**18**
1	Lian / Valcore	**19**

1	Lincoln	**20**
11	Lincoln Heights	**21**
10	Main Street	**22**
10	Marysville	**23**
1	McKnight	**24**
2	McLeod Hill	**25**
3	Monteith / Talisman	**26**
3	Montogomery / Prospect East	**27**
9	Nashwaaksis	**28**
1	Nethervue Minihome Park	**29**
17	North Devon	**30**
5	Northbrook Heights	**31**
40	Platt	**32**
2	Poet's Hill	**33**
11	Prospect	**34**
2	Rail Side	**35**
1	Saint Mary's First Nation	**36**
1	Saint Thomas University	**37**
3	Sandyville	**38**
2	Shadowood Estates	**39**
2	Silverwood	**40**
13	Skyline Acrea	**41**
22	South Devon	**42**
7	Southwood Park	**43**
7	Sunshine Gardens	**44**
11	The Hill	**45**
4	University Of New Brunswick	**46**
3	Waterloo Row	**47**
6	Williams / Hawkins Area	**48**
20	Woodstock Road	**49**
6	Youngs Crossing	**50**

:[34] In

```
mvcrime_data.describe()
```

Out[34]:

Count	
51.000000	**count**
6.980392	**mean**
7.457855	**std**
1.000000	**min**
2.000000	**25%**
4.000000	**50%**
10.000000	**75%**
40.000000	**max**

:[35] In

```
mvcrime_data.rename({'Platt': 'Plat'},inplace=True)
mvcrime_data.rename(index=str, columns={'Neighbourhood':'Neighbourh','Count':'MVCrime_Count'}, inplace=True)
mvcrime_data
```

Out[35]:

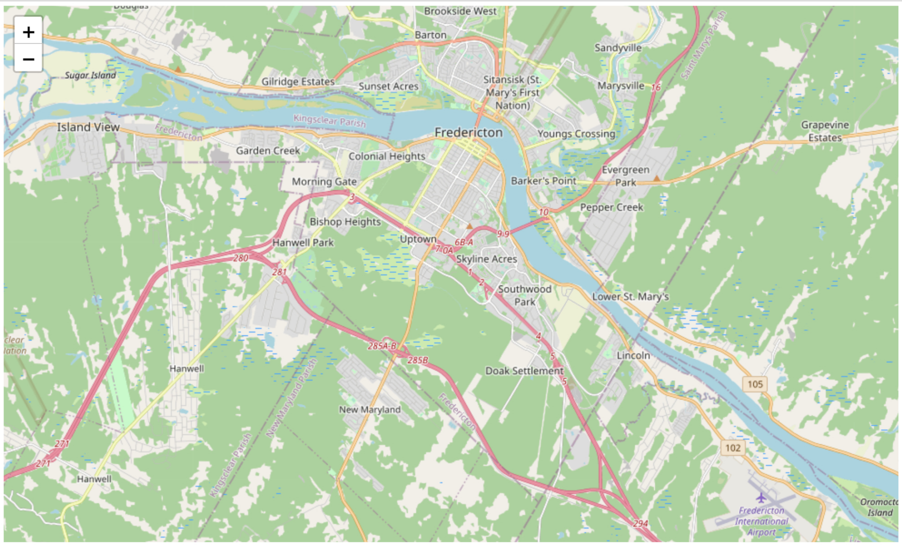
MVCrime_Count	Neighbourh	
8	Barkers Point	**0**
3	Brookside Estates	**1**
10	College Hill	**2**
6	Colonial heights	**3**
1	Diamond Street	**4**
1	Douglas	**5**
21	Downtown	**6**
9	Dun's Crossing	**7**
8	Forest Hill	**8**
20	Fredericton South	**9**
12	Fulton Heights	**10**
1	Garden Creek	**11**
3	Garden Place	**12**
1	Gilridge Estates	**13**
5	Golf Club	**14**
3	Hanwell North	**15**
2	Heron Springs	**16**
4	Highpoint Ridge	**17**
1	Knob Hill	**18**
1	Lian / Valcore	**19**
1	Lincoln	**20**
11	Lincoln Heights	**21**
10	Main Street	**22**
10	Marysville	**23**
1	McKnight	**24**
2	McLeod Hill	**25**
3	Monteith / Talisman	**26**
3	Montogomery / Prospect East	**27**
9	Nashwaaksis	**28**
1	Nethervue Minihome Park	**29**
17	North Devon	**30**
5	Northbrook Heights	**31**
40	Platt	**32**
2	Poet's Hill	**33**

11	Prospect	34
2	Rail Side	35
1	Saint Mary's First Nation	36
1	Saint Thomas University	37
3	Sandyville	38
2	Shadowood Estates	39
2	Silverwood	40
13	Skyline Acrea	41
22	South Devon	42
7	Southwood Park	43
7	Sunshine Gardens	44
11	The Hill	45
4	University Of New Brunswick	46
3	Waterloo Row	47
6	Williams / Hawkins Area	48
20	Woodstock Road	49
6	Youngs Crossing	50

:[36] In

```
world_geo = r'world_countries.json' # geojson file

fredericton_c_map = folium.Map(location=[45.91, -66.65], width=1000, height=750,zoom_start=12)

fredericton_c_map
```

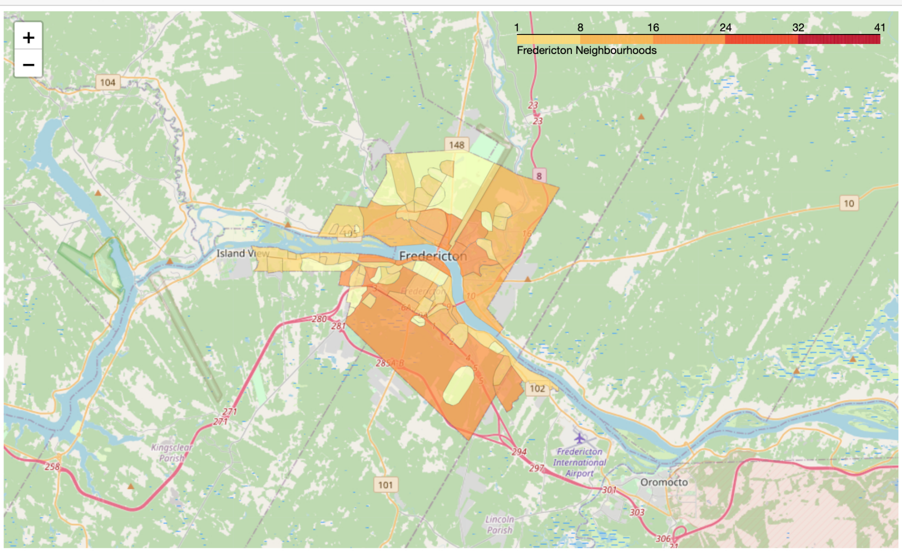
Out[36]:



:[37] In

```
fredericton_geo = r.json()
threshold_scale = np.linspace(mvcrime_data['MVCrime_Count'].min(), mvcrime_data['MVCrime_Count'].max(),6,dtype=int)
threshold_scale = threshold_scale.tolist()
threshold_scale[-1] = threshold_scale[-1]+1

fredericton_c_map.choropleth(geo_data=fredericton_geo,data=mvcrime_data,columns=['Neighbourh', 'MVCrime_Count'],key_on=
    threshold_scale=threshold_scale, fill_color='YlOrRd',fill_opacity=0.7,line_opacity=0.1,legend_name='Fredericton Nei
fredericton_c_map
```

Out[37]:



?Is it possible the higher rate of crime in the downtown area is due to population density

:[38] In

```
opendemog = 'Fredericton_Census_Tract_Demographics.xlsx'

workbook = pd.ExcelFile(opendemog)
print(workbook.sheet_names)
```

['Worksheet']

:[39] In

```
demog_df = workbook.parse('Worksheet')
demog_df.head()
```

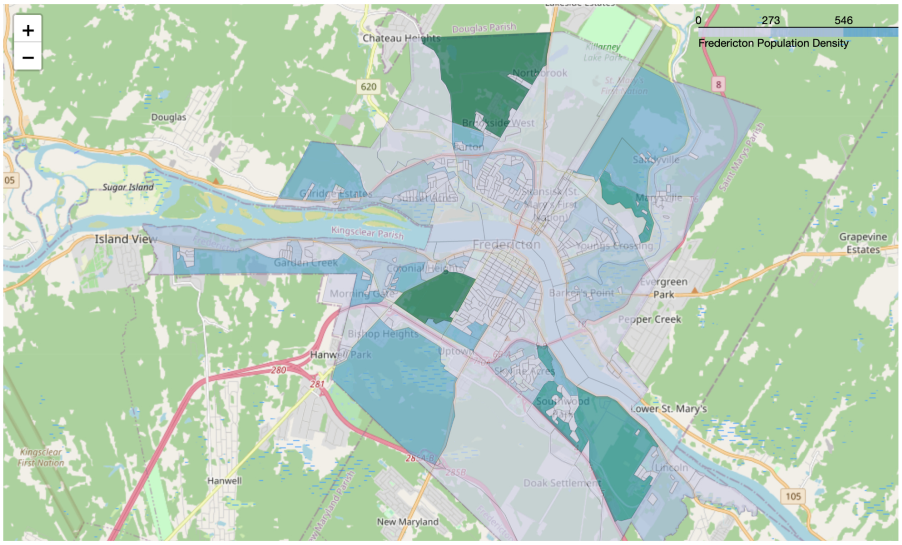
Out[39]:

CTIDLINK	Shape_Area	Shape_Leng	DBurdwell2	DBtdwell20	DBpop2011	DBuid_1	CTNAME	CTUID	CDUID	DAUID	DBUID	OBJECTID	FID	
3200002	0.000003	0.007462	22	25	60	1310024304	2.0	3200002	1310	13100243	1310024304	501	1	**0**
3200010	0.000003	0.009008	3	3	15	1310032004	10.0	3200010	1310	13100320	1310032004	502	2	**1**
3200014	0.000007	0.010602	0	0	0	1310017103	14.0	3200014	1310	13100171	1310017103	503	3	**2**
3200012	0.000068	0.039599	50	60	108	1310018301	12.0	3200012	1310	13100183	1310018301	504	4	**3**
3200007	0.000005	0.011833	44	47	129	1310022905	7.0	3200007	1310	13100229	1310022905	505	5	**4**

:[41] In

```
world_geo = r'world_countries.json' # geojson file
fredericton_d_map = folium.Map(location=[45.94, -66.63], width=1200, height=750,zoom_start=12)
fredericton_d_map

threshold_scale = np.linspace(demog_df['DBpop2011'].min(),demog_df['DBpop2011'].max(),6,dtype=int)
threshold_scale = threshold_scale.tolist()
threshold_scale[-1] = threshold_scale[-1]+1

fredericton_d_map.choropleth(geo_data=demog_geo,data=demog_df,columns=['OBJECTID','DBpop2011'],key_on='feature.properti
    threshold_scale=threshold_scale,fill_color='PuBuGn',fill_opacity=0.7, line_opacity=0.1,legend_name='Fredericton Pop
fredericton_d_map
```

Out[41]:



Let's look at specific locations in Fredericton

:[42] In

```
pointbook = 'Fredericton Locations.xlsx'

workbook_2 = pd.ExcelFile(pointbook)
print(workbook_2.sheet_names)
```

```
['Sheet1']
```

:[43] In

```
location_df = workbook_2.parse('Sheet1')
location_df
```

Out[43]:

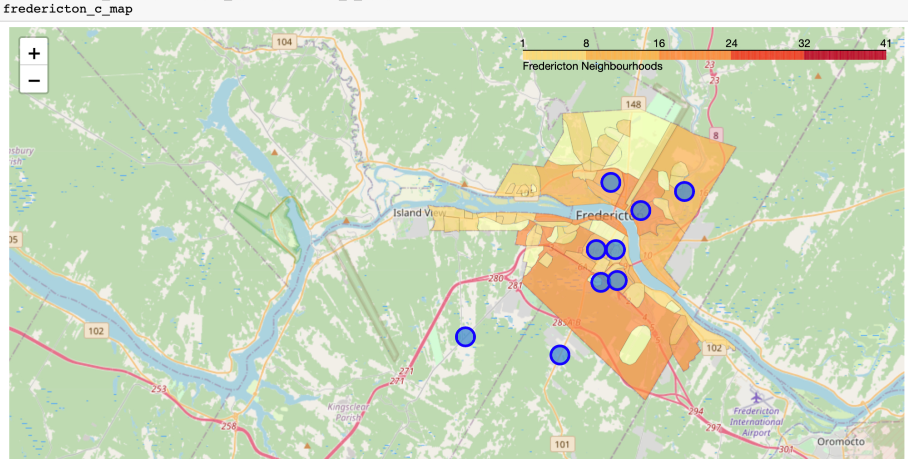
Longitude	Latitude	Location	Unnamed: 0	
66.652700-	45.931143	Knowledge Park	0	**0**
66.383550-	45.963026	Fredericton Downtown	1	**1**
66.656045-	45.948512	Fredericton Hill	2	**2**
66.644856-	45.983382	Nashwaaksis	3	**3**
66.641406-	45.948121	University of New Brunswick	4	**4**
66.622738-	45.968802	Devon	5	**5**
66.683673-	45.892795	New Maryland	6	**6**
66.589491-	45.978913	Marysville	7	**7**
66.640339-	45.931827	Skyline Acres	8	**8**
66.755113-	45.902315	Hanwell	9	**9**

Add location markers to map

:[46] In

```
for lat, lng, point in zip(location_df['Latitude'], location_df['Longitude'], location_df['Location']):
    label = '{}'.format(point)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker([lat, lng],radius=1,popup=label,color='blue',fill=True,fill_color='#3186cc',fill_opacity=0.7,
        parse_html=False).add_to(fredericton_c_map)
fredericton_c_map
```

Out[46]:



Explore Fredericton Neighbourhoods

:[47] In

```
CLIENT_ID = 'Nope' # your Foursquare ID
CLIENT_SECRET = 'Secret' # your Foursquare Secret
VERSION = '20181201' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
:Your credentails
CLIENT_ID: Nope
CLIENT_SECRET:Secret
```

Let's take a look at nearby venues

:[48] In

```
tNearbyVenues(names, latitudes, longitudes, radius=1000, LIMIT=100):

nues_list=[]
r name, lat, lng in zip(names, latitudes, longitudes):
  print(name)

  # create the API request URL
  url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.
      CLIENT_ID,
      CLIENT_SECRET,
      VERSION,
      lat,
      lng,
      radius,
      LIMIT)

  # make the GET request
  results = requests.get(url).json()["response"]['groups'][0]['items']

  # return only relevant information for each nearby venue
  venues_list.append([(
      name,
      lat,
      lng,
      v['venue']['name'],
      v['venue']['id'],
      v['venue']['location']['lat'],
      v['venue']['location']['lng'],
      v['venue']['categories'][0]['name']) for v in results])

arby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
arby_venues.columns = ['Location',
             'Location Latitude',
             'Location Longitude',
             'Venue',
             'Venue id',
             'Venue Latitude',
             'Venue Longitude',
             'Venue Category'
              ]

turn(nearby_venues)
```

:[ ] In

:[ ] In