Segmenting and Clustering Neighborhoods in Fredericton, NB

Applied Data Science Capstone Week 5 Peer-Graded Project Report

Introduction to the opportunity Fredericton is the Capital City of the only Canadian fully-bilingual Province of New Brunswick and is beautifully located on the banks of the Saint John River. While one of the least populated provincial capital cities with a population base of less than 60 thousand residents, it offers a wide spectrum of venues and is a governement, university and cultural hub. As the city grows and develops, it becomes increasingly important to examine and understand it quantitiatively. The City of Fredericton provides open data for everyone and encourages entrepreneurial use to develop services for the benefit of its ciitzens. Developers, investors, policy makers and/or city planners have an interest in answering the following questions as the need for additional services and citizen protection: 1. What neighbourhoods have the highest crime? 2. Is population density correlated to crime level? 3. Using Foursquare data, what venues are most common in different locations within the city? 4. 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.

In [73]:
```
from IPython.display import Image
from IPython.core.display import HTML
Image(url= "http://www.tourismfredericton.ca/sites/default/files/field/image/fredericton.jpg")
```

Out[73]:

Data

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

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

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:

1. Loading each data set
2. Examine the crime frequency by neighbourhood
3. Study the crime types and then pivot analysis of crime type frequency by neighbourhood
4. Understand correlation between crimes and population density
5. Perform k-means statisical analysis on venues by locations of interest based on findings from crimes and neighbourhood
6. Determine which venues are most common statistically in the region of greatest crime count then in all other locations of interest.
7. 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:

1. Neighbourhoods in Fredericton
2. Crime freqency by neighbourhood
3. Crime type frequency and statistics. The mean crime count in the City of Fredericton is 22.
4. 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.

1. 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.
2. 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.
3. 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.

1. We were able to determine the top 10 most common venues by location of interest.
2. 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

In [74]:
```
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 com
pleted the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longit
ude values

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

# 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.')
```

```
Solving environment: done

# All requested packages already installed.

Solving environment: done

# All requested packages already installed.

Libraries imported.
```

In [75]:
```
r = requests.get('https://opendata.arcgis.com/datasets/823d86e17a6d47808c6e4f1c2dd9
7928_0.geojson')
fredericton_geo = r.json()
```

In [76]:
```
neighborhoods_data = fredericton_geo['features']
```

In [77]:
```
neighborhoods_data[0]
```

Out[77]:

```
{'type': 'Feature',
 'properties': {'FID': 1,
  'OBJECTID': 1,
  'Neighbourh': 'Fredericton South',
  'Shape_Leng': 40412.2767429,
  'Shape_Area': 32431889.0002},
 'geometry': {'type': 'Polygon',
  'coordinates': [[[-66.6193489311946, 45.8688925859664],
    [-66.5986068312843, 45.8934317575498],
    [-66.5998465063764, 45.8962889533894],
    [-66.6005561754508, 45.8987959122414],
    [-66.6007627879662, 45.9004150599189],
    [-66.6005112596866, 45.9020341603803],
    [-66.5993703992758, 45.9049409211054],
    [-66.5983912356161, 45.9066536507875],
    [-66.5950405196063, 45.9110977503182],
    [-66.5924713378938, 45.9137165396725],
    [-66.5975198697905, 45.9151915074375],
    [-66.6016161874861, 45.9165914405789],
    [-66.6063862416448, 45.9184662957134],
    [-66.6102310310608, 45.9201848572716],
    [-66.6193938469588, 45.9264149777787],
    [-66.6194297795702, 45.9243466803461],
    [-66.6206694546623, 45.9221345790227],
    [-66.6241459348118, 45.9181100781124],
    [-66.6249634017204, 45.9177976046497],
    [-66.6258796833102, 45.917910095299],
    [-66.6292124330143, 45.9200348758374],
    [-66.632733828928, 45.9225720071846],
    [-66.6356353872957, 45.924409167803],
    [-66.6362731911474, 45.9249840491044],
    [-66.6381955858555, 45.9258900999313],
    [-66.6400281490351, 45.9272147820915],
    [-66.6469721261813, 45.9309512150791],
    [-66.6492628301558, 45.9324257247173],
    [-66.6501521622871, 45.9331254782868],
    [-66.6504306400252, 45.9337564984884],
    [-66.6505653873178, 45.9347436246005],
    [-66.6503587748024, 45.9357182382069],
    [-66.6520745569951, 45.9352246860213],
    [-66.6532513500173, 45.9350872403269],
    [-66.6541855979128, 45.9351122304785],
    [-66.6557756159657, 45.9353808738969],
    [-66.6597461695215, 45.9365616400027],
    [-66.6692323789218, 45.9408659130747],
    [-66.6702205257343, 45.9411720097543],
    [-66.6705888350008, 45.9415718069541],
    [-66.6717027459531, 45.9418654061867],
    [-66.6805601346545, 45.9456570693391],
    [-66.6808206460869, 45.945613344883],
    [-66.690998558256, 45.9498794400526],
    [-66.6932353633134, 45.9503791076107],
    [-66.6956697977334, 45.9504478115476],
    [-66.6955530167465, 45.9498607024316],
    [-66.695014027576, 45.9498607024316],
    [-66.6956248819692, 45.948261735435],
    [-66.699766115429, 45.9452510552052],
    [-66.6993978061625, 45.9450511702315],
    [-66.6996762839006, 45.9448512845371],
    [-66.6992271262585, 45.9446139193389],
    [-66.7022364824603, 45.9407722096716],
    [-66.7041049782513, 45.9393666396225],
    [-66.7046080348104, 45.9387919073835],
    [-66.7061441539463, 45.9390980155132],
```

In [78]:
```
g = requests.get('https://opendata.arcgis.com/datasets/6179d35eacb144a5b5fdcc869f86
dfb5_0.geojson')
demog_geo = g.json()
```

In [79]:
```
demog_data = demog_geo['features']
demog_data[0]
```

Out[79]:

```
{'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': 2.81310897700361e-06,
  'Shape__Length': 0.00746165464503067},
 'geometry': {'type': 'Polygon',
  'coordinates': [[[-66.634784212921, 45.9519239912381],
    [-66.6351046935752, 45.9507605156138],
    [-66.6378263667982, 45.9510868696778],
    [-66.636944377136, 45.9521037018384],
    [-66.634784212921, 45.9519239912381]]]}}
```


In [ ]:

In [80]:
```
import os
os.listdir('.')
```

Out[80]:
```
['Capstone Project Course.ipynb',
 'Fredericton_Census_Tract_Demographics.csv',
 '.DS_Store',
 'Fredericton_Census_Tract_Demographics.xlsx',
 'Crime_by_neighbourhood_2017.xlsx',
 'Capstone Fredericton Crime and Police Station Location.ipynb',
 'Boston_Neighborhoods (1).geojson',
 'Fredericton Locations.xlsx',
 'Week 3 Capstone - Segmenting and Clustering Neighbourhoods in Toronto_Part 2.i
pynb',
 'Fredericton.jpg',
 'Week 3 Capstone - Segmenting and Clustering Neighbourhoods in Toronto_Part 2.p
df',
 'Boston_Neighborhoods.geojson',
 '.ipynb_checkpoints',
 '.git',
 'Week 3 Capstone - Segmenting and Clustering Neighbourhoods in Toronto.ipynb',
 'Week 4 Capstone - Segmenting and Clustering Neighbourhoods in Boston.ipynb',
 'Week 3 Capstone - Segmenting and Clustering Neighbourhoods in Toronto_Part 2.h
tm',
 'Week 4 Capstone - Segmenting and Clustering Neighbourhoods in Fredericton.ipyn
b',
 'Week 4 Capstone - Segmenting and Clustering Neighbourhoods in Fredericton - Gi
thub submit.ipynb',
 'Week 3 Capstone - Segmenting and Clustering Neighbourhoods in Toronto_Part 2_f
iles']
```

In [81]:
```
opencrime = 'Crime_by_neighbourhood_2017.xlsx'
```

In [82]:
```
workbook = pd.ExcelFile(opencrime)
print(workbook.sheet_names)
```

```
['Crime_by_neighbourhood_2017']
```

In [83]:
```
crime_df = workbook.parse('Crime_by_neighbourhood_2017')
crime_df.head()
```

Out[83]:

	Neighbourhood	From_Date	To_Date	Crime_Code	Crime_Type	Ward	(
0	Fredericton South	2017-01-05T00:00:00.000Z	2017-01-26T00:00:00.000Z	2120	B&E NON-RESIDNCE	7	Frederi
1	Fredericton South	2017-03-04T00:00:00.000Z	2017-03-06T00:00:00.000Z	2120	B&E NON-RESIDNCE	7	Frederi
2	Fredericton South	2017-05-07T00:00:00.000Z	NaN	2120	B&E NON-RESIDNCE	12	Frederi
3	Fredericton South	2017-06-20T00:00:00.000Z	2017-06-21T00:00:00.000Z	2120	B&E NON-RESIDNCE	12	Frederi
4	Fredericton South	2017-07-09T00:00:00.000Z	2017-07-10T00:00:00.000Z	2120	B&E NON-RESIDNCE	7	Frederi

In [84]:
```
crime_df.drop(['From_Date', 'To_Date'], axis=1,inplace=True)
```

What is the crime count by neighbourhood?

In [128]:
```
crime_data = crime_df.groupby(['Neighbourhood']).size().to_frame(name='Count').res
et_index()
crime_data
```

Out[128]:

	Neighbourhood	Count
0	Barkers Point	47
1	Brookside	54
2	Brookside Estates	9
3	Brookside Mini Home Park	5
4	College Hill	41
5	Colonial heights	9
6	Cotton Mill Creek	4
7	Diamond Street	1
8	Doak Road	1
9	Douglas	3
10	Downtown	127
11	Dun's Crossing	18
12	Forest Hill	12
13	Fredericton South	85
14	Fulton Heights	36
15	Garden Creek	13
16	Garden Place	4
17	Gilridge Estates	3
18	Golf Club	7
19	Grasse Circle	1
20	Greenwood Minihome Park	2
21	Hanwell North	8
22	Heron Springs	3
23	Highpoint Ridge	5
24	Kelly's Court Minihome Park	1
25	Knob Hill	4
26	Knowledge Park	1
27	Lian / Valcore	7
28	Lincoln	13
29	Lincoln Heights	14
30	Main Street	78
31	Marysville	39
32	McKnight	4
33	McLeod Hill	3
34	Monteith / Talisman	12
35	Montogomery / Prospect East	16
36	Nashwaaksis	25
37	Nethervue Minihome Park	1
38	North Devon	113
39	Northbrook Heights	10

In [153]:
```
crime_data.describe()
```

Out[153]:

	Count
count	66.000000
mean	22.121212
std	34.879359
min	1.000000
25%	3.000000
50%	9.000000
75%	23.250000
max	198.000000

In [86]:
```
crime_data.rename(index=str, columns={'Neighbourhood':'Neighbourh','Count':'Crime_C
ount'}, inplace=True)
crime_data
```

Out[86]:

	Neighbourh	Crime_Count
0	Barkers Point	47
1	Brookside	54
2	Brookside Estates	9
3	Brookside Mini Home Park	5
4	College Hill	41
5	Colonial heights	9
6	Cotton Mill Creek	4
7	Diamond Street	1
8	Doak Road	1
9	Douglas	3
10	Downtown	127
11	Dun's Crossing	18
12	Forest Hill	12
13	Fredericton South	85
14	Fulton Heights	36
15	Garden Creek	13
16	Garden Place	4
17	Gilridge Estates	3
18	Golf Club	7
19	Grasse Circle	1
20	Greenwood Minihome Park	2
21	Hanwell North	8
22	Heron Springs	3
23	Highpoint Ridge	5
24	Kelly's Court Minihome Park	1
25	Knob Hill	4
26	Knowledge Park	1
27	Lian / Valcore	7
28	Lincoln	13
29	Lincoln Heights	14
30	Main Street	78
31	Marysville	39
32	McKnight	4
33	McLeod Hill	3
34	Monteith / Talisman	12
35	Montogomery / Prospect East	16
36	Nashwaaksis	25
37	Nethervue Minihome Park	1
38	North Devon	113
39	Northbrook Heights	10

In [87]:
```
crime_data.rename({'Platt': 'Plat'},inplace=True)
crime_data.rename(index=str, columns={'Neighbourhood':'Neighbourh','Count':'Crime_C
ount'}, inplace=True)
crime_data
```

Out[87]:

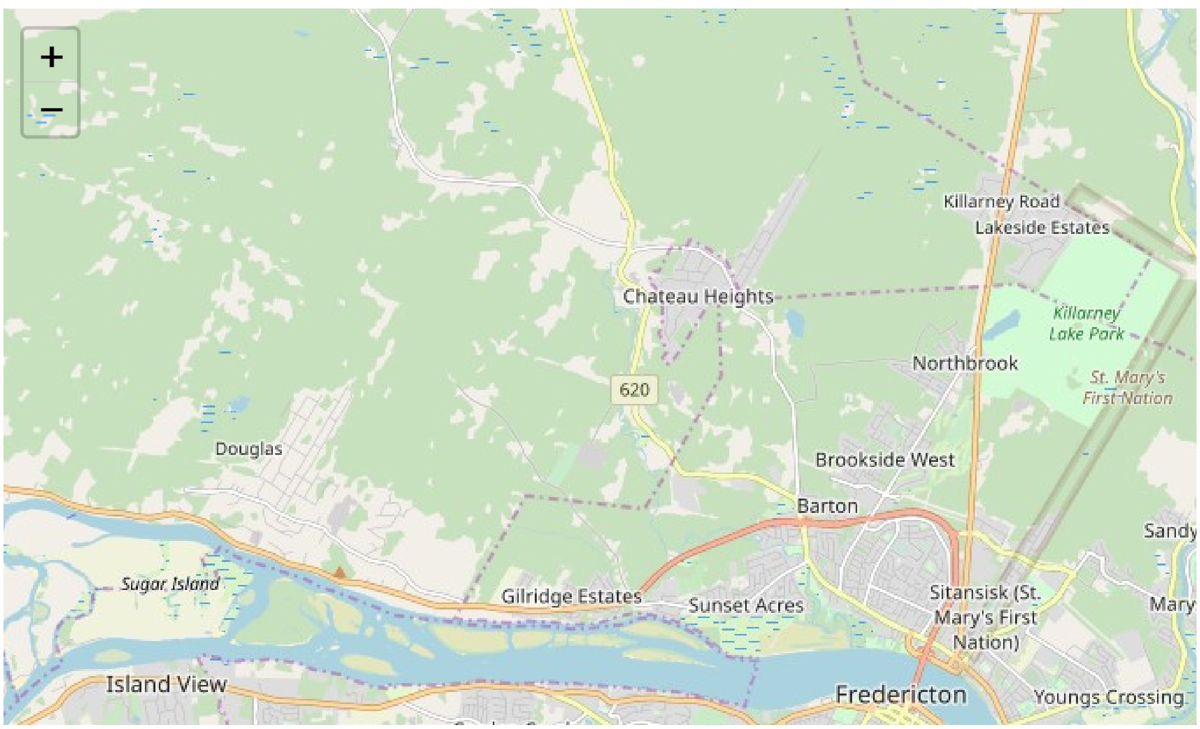
	Neighbourh	Crime_Count
0	Barkers Point	47
1	Brookside	54
2	Brookside Estates	9
3	Brookside Mini Home Park	5
4	College Hill	41
5	Colonial heights	9
6	Cotton Mill Creek	4
7	Diamond Street	1
8	Doak Road	1
9	Douglas	3
10	Downtown	127
11	Dun's Crossing	18
12	Forest Hill	12
13	Fredericton South	85
14	Fulton Heights	36
15	Garden Creek	13
16	Garden Place	4
17	Gilridge Estates	3
18	Golf Club	7
19	Grasse Circle	1
20	Greenwood Minihome Park	2
21	Hanwell North	8
22	Heron Springs	3
23	Highpoint Ridge	5
24	Kelly's Court Minihome Park	1
25	Knob Hill	4
26	Knowledge Park	1
27	Lian / Valcore	7
28	Lincoln	13
29	Lincoln Heights	14
30	Main Street	78
31	Marysville	39
32	McKnight	4
33	McLeod Hill	3
34	Monteith / Talisman	12
35	Montogomery / Prospect East	16
36	Nashwaaksis	25
37	Nethervue Minihome Park	1
38	North Devon	113
39	Northbrook Heights	10

In [88]:
```
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))
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3: DeprecationWarni
ng: Using Nominatim with the default "geopy/1.18.1" `user_agent` is strongly dis
couraged, as it violates Nominatim's ToS https://operations.osmfoundation.org/po
licies/nominatim/ and may possibly cause 403 and 429 HTTP errors. Please specify
a custom `user_agent` with `Nominatim(user_agent="my-application")` or by overri
ding 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 unti
l

The geograpical coordinate of Fredericton, New Brunswick is 45.966425, -66.64581
3.
```

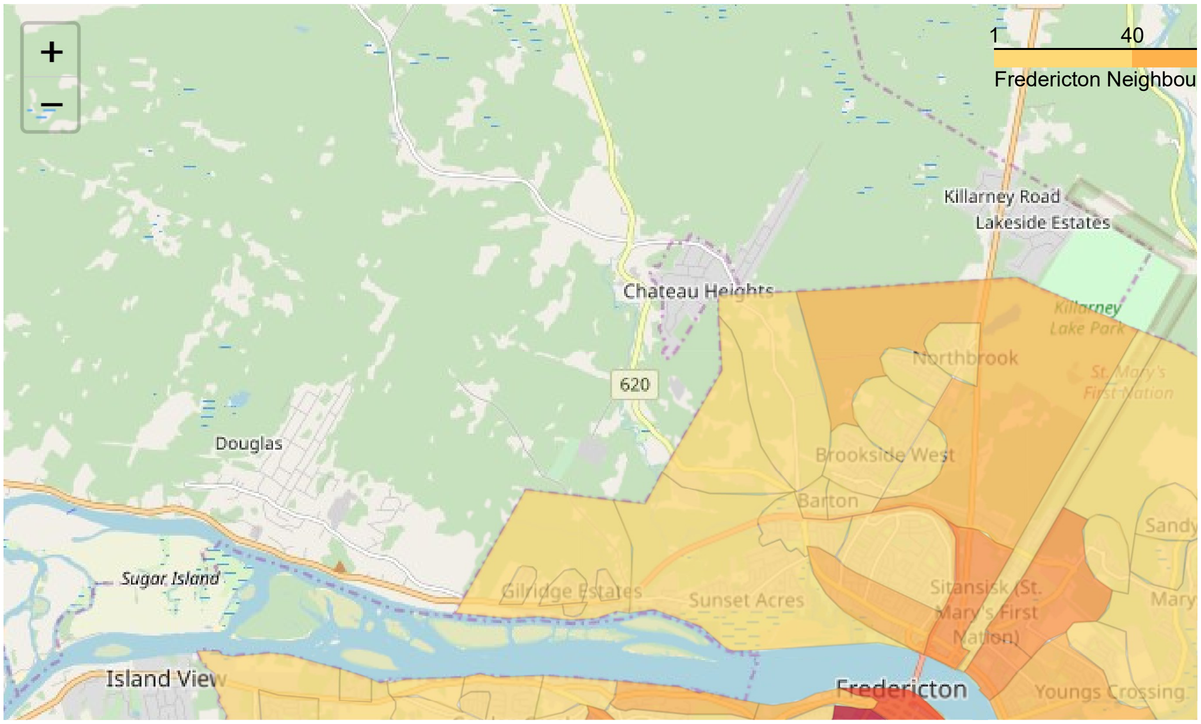
In [89]:
```
world_geo = r'world_countries.json' # geojson file

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

fredericton_1_map
```

Out[89]:



In [90]:
```
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[90]:



Examine Crime Types

In [131]:
```
crimetype_data = crime_df.groupby(['Crime_Type']).size().to_frame(name='Count').re
set_index()
crimetype_data
```

Out[131]:

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

In [154]:
```
crimetype_data.describe()
```

Out[154]:

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

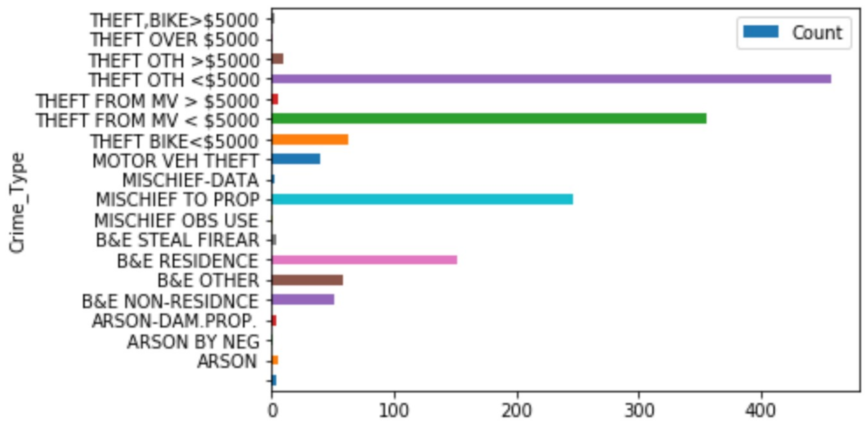
In [140]:
```
crimepivot = crime_df.pivot_table(index='Neighbourhood', columns='Crime_Type', agg
func=pd.Series.count, fill_value=0)
crimepivot
```

Out[140]:

Crime_Type	City	ARSON	ARSON BY NEG	ARSON-DAM.PROP.	B&E NON-RESIDNCE	B&E OTHER	B&E RESIDENCE	B&E STEAL FIREAR	MISCHIEF OBS USE	MISCH TO PR
Neighbourhood										
Barkers Point	0	0	0	0	2	7	7	1	0	
Brookside	0	0	0	0	2	0	0	0	0	
Brookside Estates	0	0	0	0	1	1	0	0	0	
Brookside Mini Home Park	0	0	0	0	0	0	0	1	0	
College Hill	0	2	0	0	0	2	13	0	0	
Colonial heights	0	0	0	0	0	0	3	0	0	
Cotton Mill Creek	0	0	0	0	0	0	0	0	0	
Diamond Street	0	0	0	0	0	0	0	0	0	
Doak Road	0	0	0	0	0	0	0	0	0	
Douglas	0	0	0	0	0	0	0	0	0	
Downtown	0	1	0	1	7	0	3	0	0	
Dun's Crossing	0	0	0	0	0	0	1	0	0	
Forest Hill	0	0	0	0	1	0	0	0	0	
Fredericton South	1	0	0	0	6	1	1	0	0	
Fulton Heights	0	0	0	0	1	0	6	0	0	
Garden Creek	0	0	0	0	2	1	1	0	0	
Garden Place	0	0	0	0	0	0	0	0	0	
Gilridge Estates	0	0	0	0	0	0	0	0	0	
Golf Club	0	0	0	0	0	0	1	0	0	
Grasse Circle	1	0	0	0	0	0	0	0	0	
Greenwood Minihome Park	0	0	0	0	0	1	0	0	0	
Hanwell North	0	0	0	0	0	1	2	0	0	
Heron Springs	0	0	0	0	0	0	1	0	0	
Highpoint Ridge	0	0	0	0	0	0	0	0	0	
Kelly's Court Minihome Park	0	0	0	0	0	0	0	0	0	
Knob Hill	0	0	0	0	0	0	1	0	0	
Knowledge Park	1	0	0	0	0	0	0	0	0	
Lian / Valcore	0	0	0	0	0	0	0	0	0	
Lincoln	0	0	0	0	2	2	2	0	0	
Lincoln Heights	0	0	0	0	0	1	1	0	0	
Main Street	0	0	0	1	2	4	8	0	1	

In [92]:
```
crimetype_data.plot(x='Crime_Type', y='Count', kind='barh')
```

Out[92]: <matplotlib.axes._subplots.AxesSubplot at 0x11682a860>



In [ ]:

Let's examine theft from vehicles

In [93]:
```
mvcrime_df = crime_df.loc[crime_df['Crime_Type'] == 'THEFT FROM MV < $5000']
mvcrime_df
```

Out[93]:

	Neighbourhood	Crime_Code	Crime_Type	Ward	City	FID
18	Fredericton South	2142	THEFT FROM MV < $5000	7	Fredericton	19
19	Fredericton South	2142	THEFT FROM MV < $5000	7	Fredericton	20
20	Fredericton South	2142	THEFT FROM MV < $5000	7	Fredericton	21
21	Fredericton South	2142	THEFT FROM MV < $5000	12	Fredericton	22
22	Fredericton South	2142	THEFT FROM MV < $5000	12	Fredericton	23
23	Fredericton South	2142	THEFT FROM MV < $5000	7	Fredericton	24
24	Fredericton South	2142	THEFT FROM MV < $5000	7	Fredericton	25
25	Fredericton South	2142	THEFT FROM MV < $5000	7	Fredericton	26
26	Fredericton South	2142	THEFT FROM MV < $5000	11	Fredericton	27
27	Fredericton South	2142	THEFT FROM MV < $5000	11	Fredericton	28
28	Fredericton South	2142	THEFT FROM MV < $5000	12	Fredericton	29
29	Fredericton South	2142	THEFT FROM MV < $5000	12	Fredericton	30
30	Fredericton South	2142	THEFT FROM MV < $5000	7	Fredericton	31
51	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	52
52	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	53
53	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	54
54	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	55
55	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	56
56	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	57
57	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	58
58	Barkers Point	2142	THEFT FROM MV < $5000	6	Fredericton	59
100	Sandyville	2142	THEFT FROM MV < $5000	5	Fredericton	101
107	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	108
108	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	109
109	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	110
110	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	111
111	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	112
112	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	113
113	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	114
114	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	115
115	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	116
116	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	117
117	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	118
118	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	119
119	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	120
120	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	121
121	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	122
122	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	123
123	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	124
124	South Devon	2142	THEFT FROM MV < $5000	4	Fredericton	125

In [94]:
```
mvcrime_data = mvcrime_df.groupby(['Neighbourhood']).size().to_frame(name='Count').
reset_index()
mvcrime_data
```

Out[94]:

	Neighbourhood	Count
0	Barkers Point	8
1	Brookside Estates	3
2	College Hill	10
3	Colonial heights	6
4	Diamond Street	1
5	Douglas	1
6	Downtown	21
7	Dun's Crossing	9
8	Forest Hill	8
9	Fredericton South	20
10	Fulton Heights	12
11	Garden Creek	1
12	Garden Place	3
13	Gilridge Estates	1
14	Golf Club	5
15	Hanwell North	3
16	Heron Springs	2
17	Highpoint Ridge	4
18	Knob Hill	1
19	Lian / Valcore	1
20	Lincoln	1
21	Lincoln Heights	11
22	Main Street	10
23	Marysville	10
24	McKnight	1
25	McLeod Hill	2
26	Monteith / Talisman	3
27	Montogomery / Prospect East	3
28	Nashwaaksis	9
29	Nethervue Minihome Park	1
30	North Devon	17
31	Northbrook Heights	5
32	Plat	40
33	Poet's Hill	2
34	Prospect	11
35	Rail Side	2
36	Saint Mary's First Nation	1
37	Saint Thomas University	1
38	Sandyville	3
39	Shadowood Estates	2

In [155]:
```
mvcrime_data.describe()
```

Out[155]:

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

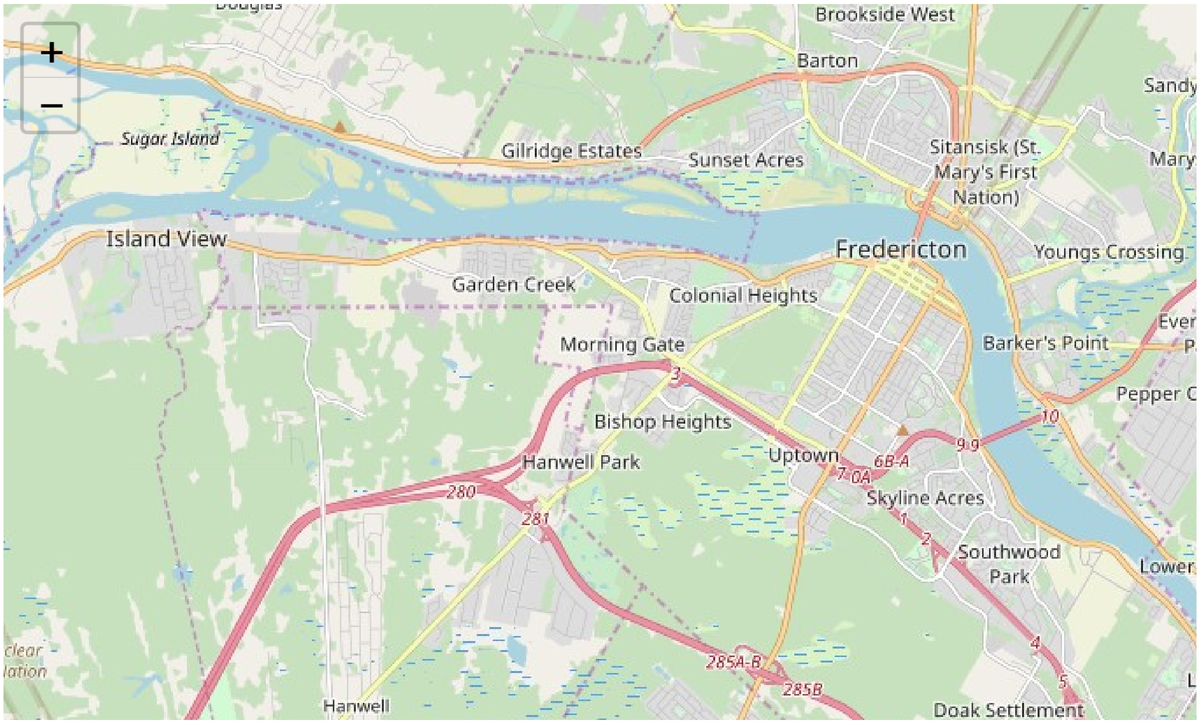
In [95]:
```
mvcrime_data.rename({'Platt': 'Plat'},inplace=True)
mvcrime_data.rename(index=str, columns={'Neighbourhood':'Neighbourh','Count':'MVCri
me_Count'}, inplace=True)
mvcrime_data
```

Out[95]:

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

In [96]:
```
world_geo = r'world_countries.json' # geojson file

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

fredericton_c_map
```

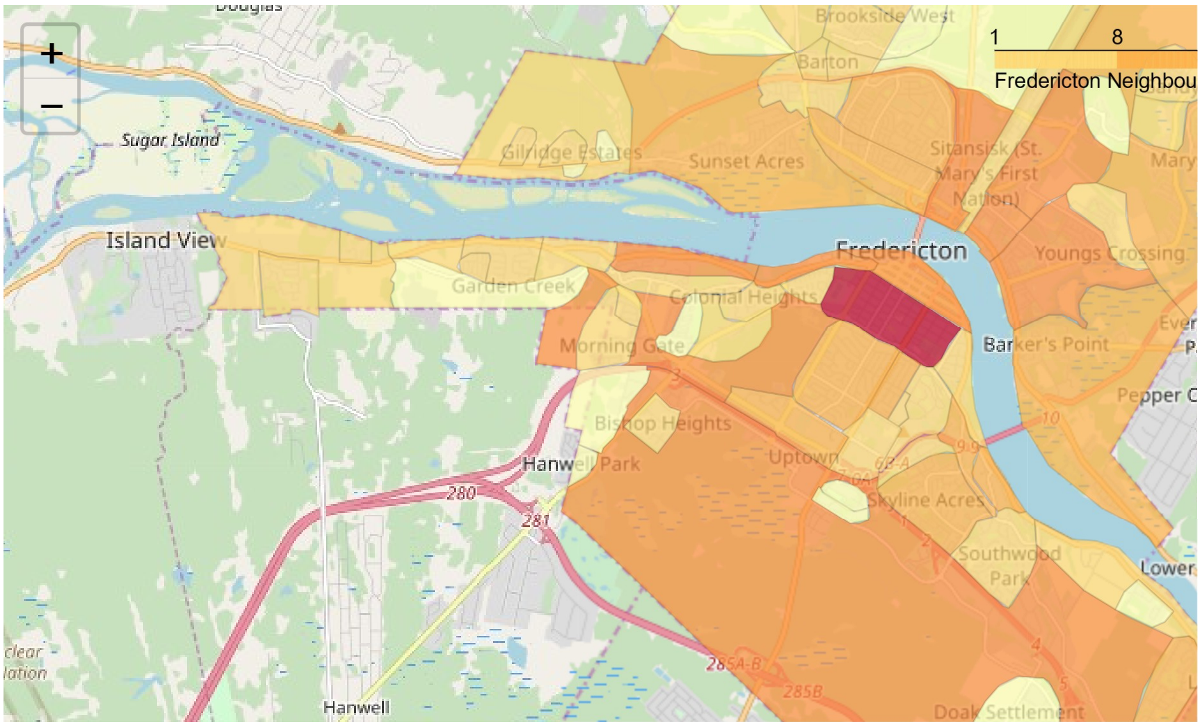
Out[96]:



In [97]:

```
## Motor Vehicle Crime <$5000 Count
fredericton_geo = r.json()
threshold_scale = np.linspace(mvcrime_data['MVCrime_Count'].min(), mvcrime_data['MV
Crime_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=['N
eighbourh', 'MVCrime_Count'],key_on='feature.properties.Neighbourh',
    threshold_scale=threshold_scale, fill_color='YlOrRd',fill_opacity=0.7,line_opac
ity=0.1,legend_name='Fredericton Neighbourhoods')
fredericton_c_map
```

Out[97]:



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

In [98]:

```
opendemog = 'Fredericton_Census_Tract_Demographics.xlsx'

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

```
['Fredericton_Census_Tract_Demogr']
```

In [99]:
```python
demog_df = workbook.parse('Fredericton_Census_Tract_Demogr')
demog_df.head()
```

Out[99]:

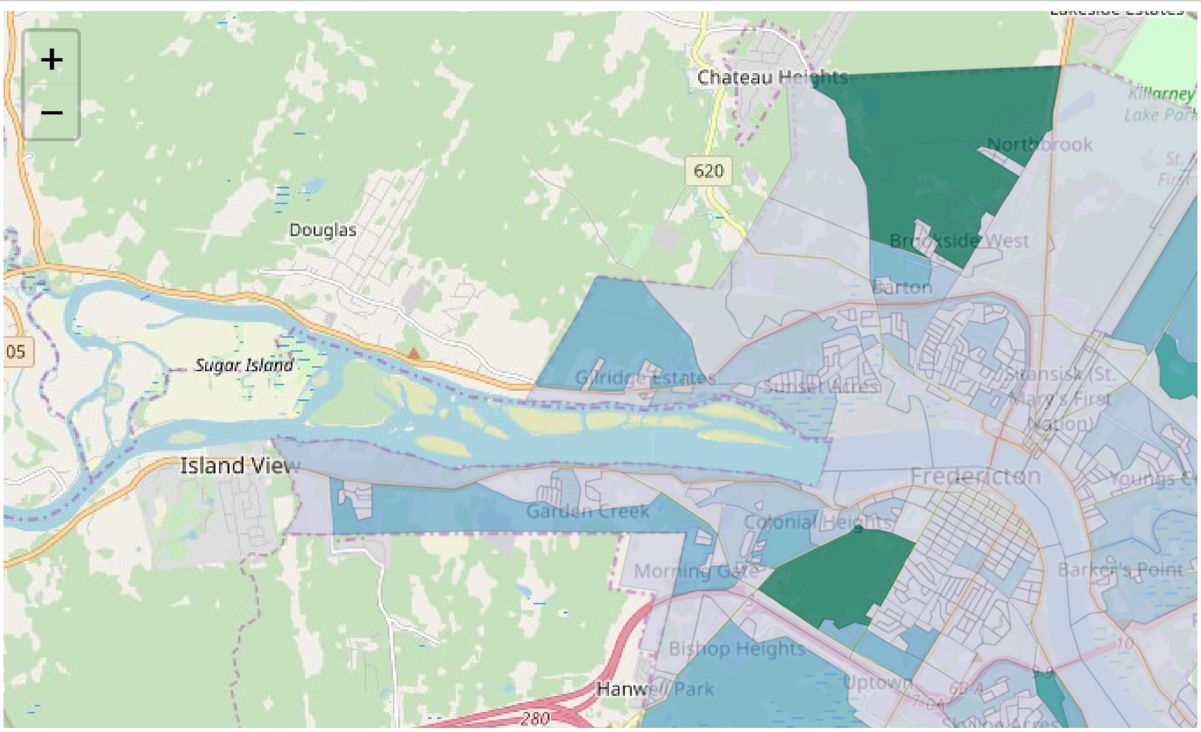
	FID	OBJECTID	DBUID	DAUID	CDUID	CTUID	CTNAME	DBuid_1	DBpop2011	DBtdwell20	D
0	1	501	1310024304	13100243	1310	3200002	2	1310024304	60	25	
1	2	502	1310032004	13100320	1310	3200010	10	1310032004	15	3	
2	3	503	1310017103	13100171	1310	3200014	14	1310017103	0	0	
3	4	504	1310018301	13100183	1310	3200012	12	1310018301	108	60	
4	5	505	1310022905	13100229	1310	3200007	7	1310022905	129	47	

In [ ]:

In [ ]:

In [100]:
```python
# Population Density
world_geo = r'world_countries.json' # geojson file
fredericton_d_map = folium.Map(location=[45.94, -66.63], width=1200, height=750,zo
om_start=12)
fredericton_d_map

threshold_scale = np.linspace(demog_df['DBpop2011'].min(),demog_df['DBpop2011'].ma
x(),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.properties.OBJECTID',
    threshold_scale=threshold_scale,fill_color='PuBuGn',fill_opacity=0.7, line_opa
city=0.1,legend_name='Fredericton Population Density')
fredericton_d_map
```

Out[100]:



Let's look at specific locations in Fredericton

In [101]:
```
pointbook = 'Fredericton Locations.xlsx'

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

```
['Sheet1']
```

In [102]:
```
location_df = workbook_2.parse('Sheet1')
location_df
```

Out[102]:

	Location	Neighbourh	Latitude	Longitude
0	Knowledge Park	NaN	45.931143	-66.652700
1	Fredericton Hill	NaN	45.948512	-66.656045
2	Nashwaaksis	NaN	45.983382	-66.644856
3	University of New Brunswick	NaN	45.948121	-66.641406
4	Devon	NaN	45.968802	-66.622738
5	New Maryland	NaN	45.892795	-66.683673
6	Marysville	NaN	45.978913	-66.589491
7	Skyline Acres	NaN	45.931827	-66.640339
8	Hanwell	NaN	45.902315	-66.755113
9	Downtown	NaN	45.958327	-66.647211

In [103]:
```
location_df.drop(['Neighbourh'], axis=1,inplace=True)
location_df
```

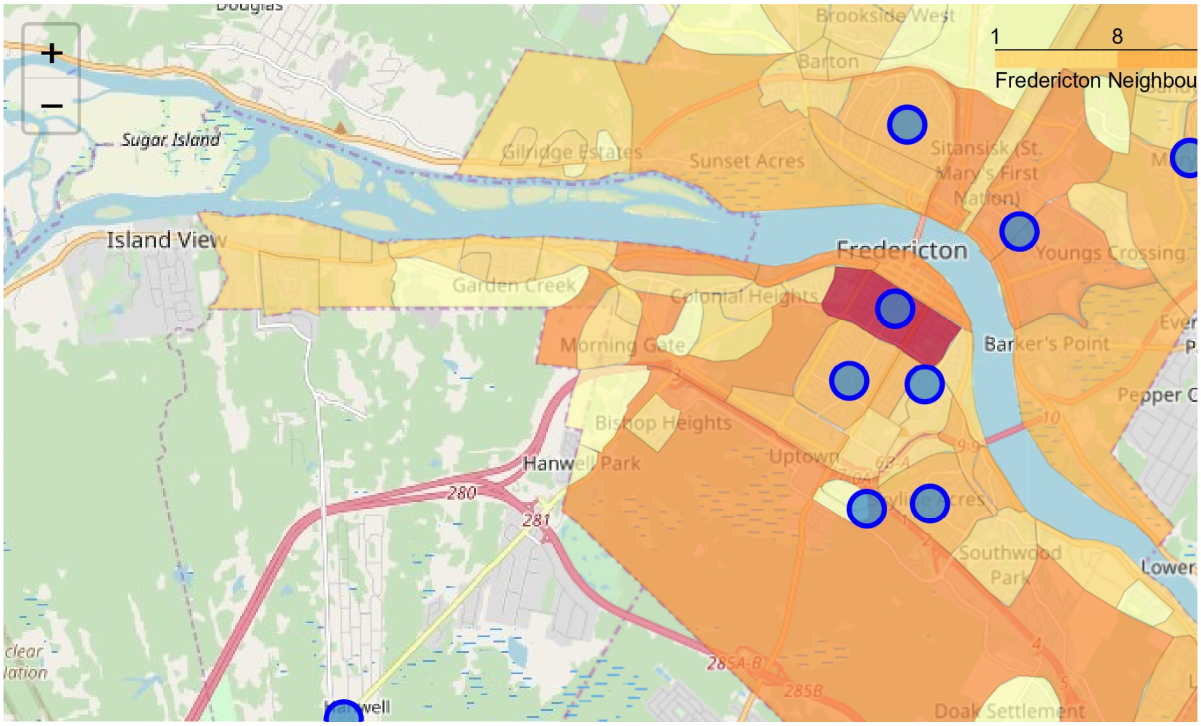
Out[103]:

	Location	Latitude	Longitude
0	Knowledge Park	45.931143	-66.652700
1	Fredericton Hill	45.948512	-66.656045
2	Nashwaaksis	45.983382	-66.644856
3	University of New Brunswick	45.948121	-66.641406
4	Devon	45.968802	-66.622738
5	New Maryland	45.892795	-66.683673
6	Marysville	45.978913	-66.589491
7	Skyline Acres	45.931827	-66.640339
8	Hanwell	45.902315	-66.755113
9	Downtown	45.958327	-66.647211

Add location markers to map

In [104]:
```
for lat, lng, point in zip(location_df['Latitude'], location_df['Longitude'], loca
tion_df['Location']):
    label = '{}'.format(point)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker([lat, lng],radius=1,popup=label,color='blue',fill=True,fil
l_color='#3186cc',fill_opacity=0.7,
        parse_html=False).add_to(fredericton_c_map)
fredericton_c_map
```

Out[104]:



In [ ]:

Explore Fredericton Neighbourhoods

Define Foursquare Credentials and Version

In [2]:
```
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

In [106]:

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

    venues_list=[]
    for 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_s
ecret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            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])

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

    return(nearby_venues)
```

In [107]:

```
fredericton_data_venues = getNearbyVenues(names=location_df['Location'],
                                   latitudes=location_df['Latitude'],
                                   longitudes=location_df['Longitude']
                                  )
```

```
Knowledge Park
Fredericton Hill
Nashwaaksis
University of New Brunswick
Devon
New Maryland
Marysville
Skyline Acres
Hanwell
Downtown
```

In [108]:
```
print(fredericton_data_venues.shape)
fredericton_data_venues
```

```
(166, 8)
```

Out[108]:

	Location	Location Latitude	Location Longitude	Venue	Venue id	Venue Latitude	Venue Longitude	Ca
0	Knowledge Park	45.931143	-66.652700	Costco Wholesale	4e18ab92183880768f43bff6	45.927034	-66.663447	War
1	Knowledge Park	45.931143	-66.652700	PetSmart	4bbca501a0a0c9b6078f1a0f	45.929768	-66.659939	P
2	Knowledge Park	45.931143	-66.652700	Montana's	4e50406e62844166699b0780	45.931511	-66.662507	Res
3	Knowledge Park	45.931143	-66.652700	Boston Pizza	4b64944af964a52041bf2ae3	45.938123	-66.660037	Sp
4	Knowledge Park	45.931143	-66.652700	Michaels	4c489858417b20a13b82e1a9	45.929965	-66.659548	Arts &
5	Knowledge Park	45.931143	-66.652700	Alcool NB Liquor	4b77335df964a5202c872ee3	45.930680	-66.664180	Liqu
6	Knowledge Park	45.931143	-66.652700	Best Buy	5520124a498e0467bb6e81c8	45.937673	-66.660380	Ele
7	Knowledge Park	45.931143	-66.652700	Wal-Mart	4bad313ff964a5208c373be3	45.934081	-66.663539	
8	Knowledge Park	45.931143	-66.652700	Booster Juice	4c42414e520fa59334f9caac	45.935198	-66.663602	S
9	Knowledge Park	45.931143	-66.652700	Dairy Queen	4b86f05bf964a52009a731e3	45.938004	-66.659442	Fa Res
10	Knowledge Park	45.931143	-66.652700	H&M	509c3265498efdffc5739a0f	45.935196	-66.663290	(
11	Knowledge Park	45.931143	-66.652700	Dairy Queen (Treat)	4cc6123cbde8f04d9ce0b44b	45.934520	-66.663988	Fa Res
12	Knowledge Park	45.931143	-66.652700	Winners	4caa46a744a8224b96e42640	45.930427	-66.659758	(
13	Knowledge Park	45.931143	-66.652700	East Side Mario's	4b55d89bf964a520a2f227e3	45.931376	-66.663417	Res
14	Knowledge Park	45.931143	-66.652700	McDonald's	4c6e9ef665eda09377e951d0	45.934575	-66.663319	Fa Res
15	Knowledge Park	45.931143	-66.652700	Home Sense	54024f60498ee424eedb7bf9	45.930528	-66.660103	Dep
16	Knowledge Park	45.931143	-66.652700	The Shoe company	4bd76dfa5cf276b0fb469b00	45.929636	-66.660449	Sho
17	Knowledge Park	45.931143	-66.652700	Avalon Spa Uptown	4cd99e0f51fc8cfa4369f05d	45.930774	-66.660927	
18	Knowledge Park	45.931143	-66.652700	Wicker Emporium	4e6baff588772457c4fd1968	45.930897	-66.661338	Fu Hom
19	Knowledge Park	45.931143	-66.652700	Dollarama	4ba3dd18f964a520d86738e3	45.930897	-66.661714	D
20	Knowledge Park	45.931143	-66.652700	Bed Bath & Beyond	5083f283e4b0bf87c15e9ea1	45.930097	-66.662166	Fu Hom
21	Knowledge Park	45.931143	-66.652700	GAP Factory Store	50a8f005e4b0e4f42e033a2a	45.930211	-66.662416	(
22	Knowledge Park	45.931143	-66.652700	carter's | OshKosh B'gosh	50a51363e4b0a3e2f7db76bf	45.929978	-66.662966	Kic
23	Knowledge Park	45.931143	-66.652700	Deluxe Fish & Chips	4e5d0b99fa76a4cf148d9a15	45.931722	-66.663131	S Res
24	Knowledge Park	45.931143	-66.652700	Hallmark	4cd96cf651fc8cfa522eef5d	45.930646	-66.663745	G
25	Knowledge	45.931143	-66.652700	NB Liquor	5985f08b6cf01a7e38b85fba	45.930228	-66.664395	Liqu

In [109]:
```
print('There are {} unique venue categories.'.format(len(fredericton_data_venues['
Venue Category'].unique())))
```

```
There are 73 unique venue categories.
```

In [110]:
```
print('There are {} unique venues.'.format(len(fredericton_data_venues['Venue id
'].unique())))
```

```
There are 153 unique venues.
```

In [111]:
```
univen = fredericton_data_venues.groupby('Location').nunique('Venue Category')
univen
```

Out[111]:

Location	Location	Location Latitude	Location Longitude	Venue	Venue id	Venue Latitude	Venue Longitude	Venue Category
Devon	1	1	1	12	12	12	12	11
Downtown	1	1	1	61	62	62	62	44
Fredericton Hill	1	1	1	17	17	17	17	13
Hanwell	1	1	1	2	2	2	2	2
Knowledge Park	1	1	1	31	31	31	31	23
Marysville	1	1	1	5	5	5	5	5
Nashwaaksis	1	1	1	17	19	19	19	15
New Maryland	1	1	1	4	4	4	4	4
Skyline Acres	1	1	1	4	4	4	4	3
University of New Brunswick	1	1	1	9	10	10	10	8

In [112]:
```
fredericton_data_venues.groupby('Venue Category').nunique()
```

Out[112]:

Venue Category	Location	Location Latitude	Location Longitude	Venue	Venue id	Venue Latitude	Venue Longitude	Venue Category
Art Gallery	2	2	2	1	1	1	1	1
Art Museum	1	1	1	1	1	1	1	1
Arts & Crafts Store	2	2	2	2	2	2	2	1
Auto Dealership	1	1	1	1	1	1	1	1
Bakery	3	3	3	5	5	5	5	1
Bank	1	1	1	1	1	1	1	1
Bar	3	3	3	4	4	4	4	1
Baseball Field	3	3	3	3	3	3	3	1
Baseball Stadium	1	1	1	1	1	1	1	1
Basketball Court	1	1	1	1	1	1	1	1
Beer Store	1	1	1	1	1	1	1	1
Big Box Store	1	1	1	1	1	1	1	1
Bookstore	1	1	1	1	1	1	1	1
Breakfast Spot	1	1	1	1	1	1	1	1
Brewery	1	1	1	1	1	1	1	1
Burger Joint	2	2	2	1	1	1	1	1
Café	1	1	1	3	3	3	3	1
Chinese Restaurant	2	2	2	3	3	3	3	1
Clothing Store	1	1	1	3	3	3	3	1
Coffee Shop	7	7	7	6	13	13	13	1
Dance Studio	1	1	1	1	1	1	1	1
Department Store	2	2	2	2	2	2	2	1
Discount Store	1	1	1	1	1	1	1	1
Electronics Store	2	2	2	2	2	2	2	1
Farmers Market	2	2	2	3	3	3	3	1
Fast Food Restaurant	5	5	5	9	10	10	10	1
Furniture / Home Store	1	1	1	2	2	2	2	1
Gas Station	2	2	2	1	2	2	2	1
Gastropub	1	1	1	1	1	1	1	1
Gift Shop	1	1	1	1	1	1	1	1
Greek Restaurant	1	1	1	1	1	1	1	1
Grocery Store	4	4	4	4	4	4	4	1
Gym	4	4	4	2	2	2	2	1
Gym / Fitness Center	1	1	1	1	1	1	1	1

In [ ]:

Analyze each Location

In [113]:

```
# one hot encoding
freddy_onehot = pd.get_dummies(fredericton_data_venues[['Venue Category']], prefix
="", prefix_sep="")

# add neighbourhood column back to dataframe
freddy_onehot['Location'] = fredericton_data_venues['Location']

# move neighbourhood column to the first column
fixed_columns = [freddy_onehot.columns[-1]] + list(freddy_onehot.columns[:-1])
freddy_onehot = freddy_onehot[fixed_columns]

freddy_onehot.head()
```

Out[113]:

	Location	Art Gallery	Art Museum	Arts & Crafts Store	Auto Dealership	Bakery	Bank	Bar	Baseball Field	Baseball Stadium	Basketball Court	Beer Store
0	Knowledge Park	0	0	0	0	0	0	0	0	0	0	0
1	Knowledge Park	0	0	0	0	0	0	0	0	0	0	0
2	Knowledge Park	0	0	0	0	0	0	0	0	0	0	0
3	Knowledge Park	0	0	0	0	0	0	0	0	0	0	0
4	Knowledge Park	0	0	1	0	0	0	0	0	0	0	0

In [114]:

```
freddy_onehot.shape
```

Out[114]: (166, 74)

Group rows by location and by the mean of the frequency of occurrence of each category

In [115]:
```
freddy_grouped = freddy_onehot.groupby('Location').mean().reset_index()
freddy_grouped
```

Out[115]:

	Location	Art Gallery	Art Museum	Arts & Crafts Store	Auto Dealership	Bakery	Bank	Bar	Baseball Field	Baseball Stadium	Ba
0	Devon	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.083333	0.0	
1	Downtown	0.016129	0.016129	0.000000	0.000000	0.016129	0.016129	0.048387	0.000000	0.0	
2	Fredericton Hill	0.000000	0.000000	0.000000	0.000000	0.176471	0.000000	0.058824	0.000000	0.0	
3	Hanwell	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	
4	Knowledge Park	0.000000	0.000000	0.032258	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	
5	Marysville	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.2	
6	Nashwaaksis	0.000000	0.000000	0.052632	0.052632	0.052632	0.000000	0.000000	0.000000	0.0	
7	New Maryland	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.250000	0.0	
8	Skyline Acres	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.250000	0.0	
9	University of New Brunswick	0.100000	0.000000	0.000000	0.000000	0.000000	0.000000	0.200000	0.000000	0.0	

In [116]:
```
freddy_grouped.shape
```

Out[116]: (10, 74)

Print each Location with the top 5 most common venues

In [117]:
```
num_top_venues = 5

for hood in freddy_grouped['Location']:
    print("----"+hood+"----")
    temp = freddy_grouped[freddy_grouped['Location'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(nu
m_top_venues))
    print('\n')
```

```
----Devon----
                  venue  freq
0  Fast Food Restaurant  0.17
1           Coffee Shop  0.08
2         Grocery Store  0.08
3    Seafood Restaurant  0.08
4          Skating Rink  0.08


----Downtown----
         venue  freq
0  Coffee Shop  0.10
1          Pub  0.08
2         Café  0.05
3   Restaurant  0.05
4          Bar  0.05


----Fredericton Hill----
            venue  freq
0          Bakery  0.18
1     Pizza Place  0.18
2    Hockey Arena  0.06
3      Smoke Shop  0.06
4  Ice Cream Shop  0.06


----Hanwell----
                 venue  freq
0          Coffee Shop   0.5
1       Rental Service   0.5
2          Art Gallery   0.0
3  Rental Car Location   0.0
4            Racetrack   0.0


----Knowledge Park----
                    venue  freq
0    Fast Food Restaurant  0.13
1          Clothing Store  0.10
2            Liquor Store  0.06
3              Restaurant  0.06
4  Furniture / Home Store  0.06


----Marysville----
              venue  freq
0       Coffee Shop   0.2
1          Pharmacy   0.2
2              Park   0.2
3  Baseball Stadium   0.2
4       Gas Station   0.2


----Nashwaaksis----
                  venue  freq
0        Farmers Market  0.11
1        Sandwich Place  0.11
2           Coffee Shop  0.11
3  Fast Food Restaurant  0.11
4            Beer Store  0.05


----New Maryland----
```

Now into a pandas dataframe

In [118]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

In [119]:

```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Location']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
location_venues_sorted = pd.DataFrame(columns=columns)
location_venues_sorted['Location'] = freddy_grouped['Location']

for ind in np.arange(freddy_grouped.shape[0]):
    location_venues_sorted.iloc[ind, 1:] = return_most_common_venues(freddy_groupe
d.iloc[ind, :], num_top_venues)

location_venues_sorted
```

Out[119]:

	Location	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
0	Devon	Fast Food Restaurant	Grocery Store	Smoke Shop	Pharmacy	Coffee Shop	Seafood Restaurant	Park	Department Store
1	Downtown	Coffee Shop	Pub	Bar	Café	Restaurant	Park	Pizza Place	Grocery Store
2	Fredericton Hill	Bakery	Pizza Place	Hockey Arena	Smoke Shop	Hardware Store	Video Store	Ice Cream Shop	Park
3	Hanwell	Rental Service	Coffee Shop	Warehouse Store	Dance Studio	Department Store	Discount Store	Electronics Store	Farmers Market
4	Knowledge Park	Fast Food Restaurant	Clothing Store	Furniture / Home Store	Liquor Store	Restaurant	Warehouse Store	Shoe Store	Pet Store
5	Marysville	Baseball Stadium	Gas Station	Pharmacy	Park	Coffee Shop	Gift Shop	Gastropub	Greek Restaurant
6	Nashwaaksis	Coffee Shop	Sandwich Place	Farmers Market	Fast Food Restaurant	Gym	Spa	Electronics Store	Beer Store
7	New Maryland	Gas Station	Dance Studio	Fast Food Restaurant	Baseball Field	Furniture / Home Store	Department Store	Discount Store	Electronics Store
8	Skyline Acres	Chinese Restaurant	Baseball Field	Hockey Arena	Arts & Crafts Store	Coffee Shop	Gym / Fitness Center	Gym	Grocery Store
9	University of New Brunswick	Bar	Coffee Shop	Art Gallery	Pub	Burger Joint	Basketball Court	Grocery Store	Gym

Cluster Fredericton Locations

Run k-means to cluster Locations into 5 clusters

In [120]:
```
# set number of clusters
kclusters = 5

freddy_grouped_clustering = freddy_grouped.drop('Location', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(freddy_grouped_clusterin
g)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

Out[120]:
```
array([1, 1, 1, 0, 1, 4, 1, 3, 2, 1], dtype=int32)
```

Now creating a new dataframe including the cluster as well as the top 10 venues for each Location

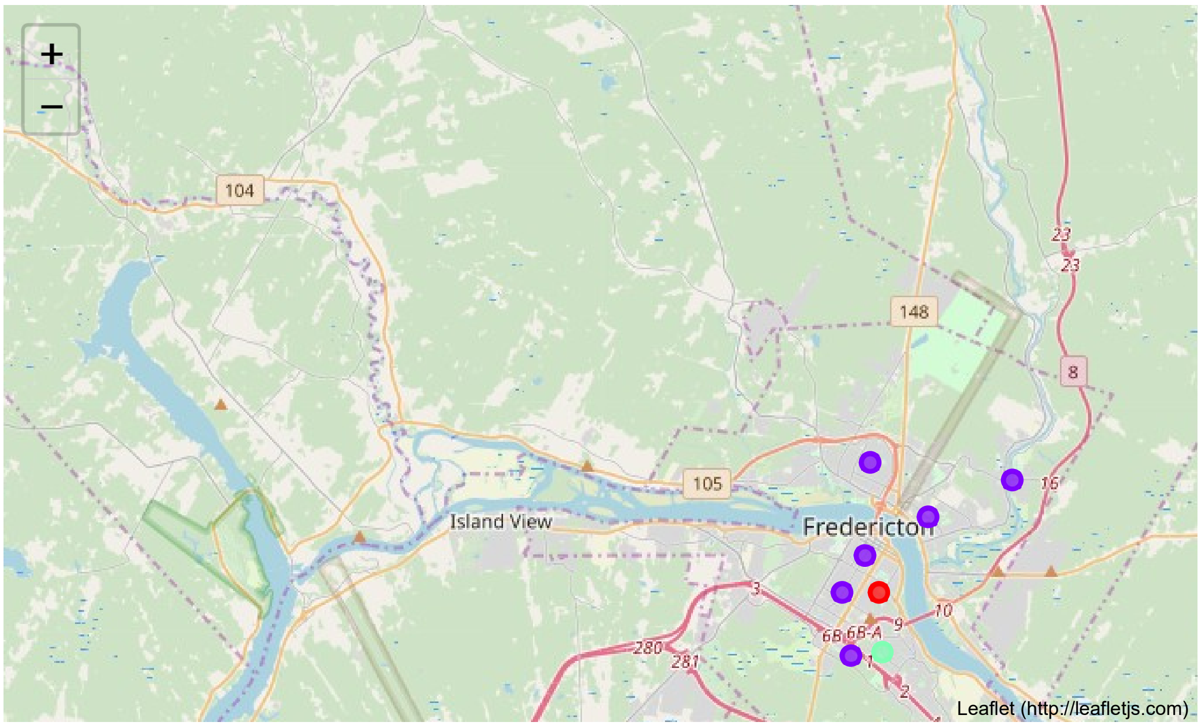
In [121]:
```
freddy_merged = location_df

# add clustering labels
freddy_merged['Cluster Labels'] = kmeans.labels_

# merge fredericton_grouped with location df to add latitude/longitude for each lo
cation
freddy_merged = freddy_merged.join(location_venues_sorted.set_index('Location'), o
n='Location')

freddy_merged# check the last columns!
```

Out[121]:

	Location	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Cor \
0	Knowledge Park	45.931143	-66.652700	1	Fast Food Restaurant	Clothing Store	Furniture / Home Store	Liquor Store	Restaurant	Warel
1	Fredericton Hill	45.948512	-66.656045	1	Bakery	Pizza Place	Hockey Arena	Smoke Shop	Hardware Store	Video
2	Nashwaaksis	45.983382	-66.644856	1	Coffee Shop	Sandwich Place	Farmers Market	Fast Food Restaurant	Gym	
3	University of New Brunswick	45.948121	-66.641406	0	Bar	Coffee Shop	Art Gallery	Pub	Burger Joint	Bask
4	Devon	45.968802	-66.622738	1	Fast Food Restaurant	Grocery Store	Smoke Shop	Pharmacy	Coffee Shop	Se Resta
5	New Maryland	45.892795	-66.683673	4	Gas Station	Dance Studio	Fast Food Restaurant	Baseball Field	Furniture / Home Store	Depar
6	Marysville	45.978913	-66.589491	1	Baseball Stadium	Gas Station	Pharmacy	Park	Coffee Shop	Gift
7	Skyline Acres	45.931827	-66.640339	3	Chinese Restaurant	Baseball Field	Hockey Arena	Arts & Crafts Store	Coffee Shop	F (
8	Hanwell	45.902315	-66.755113	2	Rental Service	Coffee Shop	Warehouse Store	Dance Studio	Department Store	Dis
9	Downtown	45.958327	-66.647211	1	Coffee Shop	Pub	Bar	Café	Restaurant	

In [122]:
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i+x+(i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(freddy_merged['Latitude'], freddy_merged['Longit
ude'], freddy_merged['Location'], freddy_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker([lat, lon], radius=5,popup=label,color=rainbow[cluster-1],
fill=True,fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)
map_clusters
```

Out[122]:



In [ ]: