Selecting the Best Forward Sortation Area (FSA) for a New Restaurant in Toronto

By: Nassim Al-Abed

July 13, 2019

Introduction

- Normally site selection is a task that is carried out by Geographic Information Systems (GIS) software after specifying several criteria. GIS is very efficient in carrying out, such as task if the needed spatial data and metadata are available. For this project, the use of specialized GIS package is not advisable because we need to apply the knowledge gained in the courses of data science taken during the previous months in solving a site selection problem.

Introduction (Continue)

- Normally starting a new business and opening a new restaurant involves investing time in studying the market and calculating the risk associated with the restaurant site selection. Utilizing the free tools in data science in analysing the freely available data either from the internet or from governmental sources is advantageous.

Identifying the problem

- Selecting the best Forward Sortation Area (FSA) for a new Chinese restaurant in Toronto was the objective of this project.
- A forward sortation area (FSA) is a way to designate a geographical unit based on the first three characters in a Canadian postal code.

Identifying the problem (Continue)

- I chosed two criterions for selecting the best FSA, number one the suitable FSA should have the highest population density in Toronto. The second was having not less than 10% of the FSA's residents of Chines origin.
- Percentage of FSA residents of Chines origin was determined from Toronto's demographic data by linking it to the second spoken language in the FSAs.

Identifying the problem (Continue)

- Not all of the needed data for this project was readily available.
- Raw data from different sources were obtained, processed and analysed.
- The data was cleaned, prepared and converted to the format appropriate for analysis to be processed by Folium to create the choropleth map that can help in finding a suitable FSA for the new restaurant.

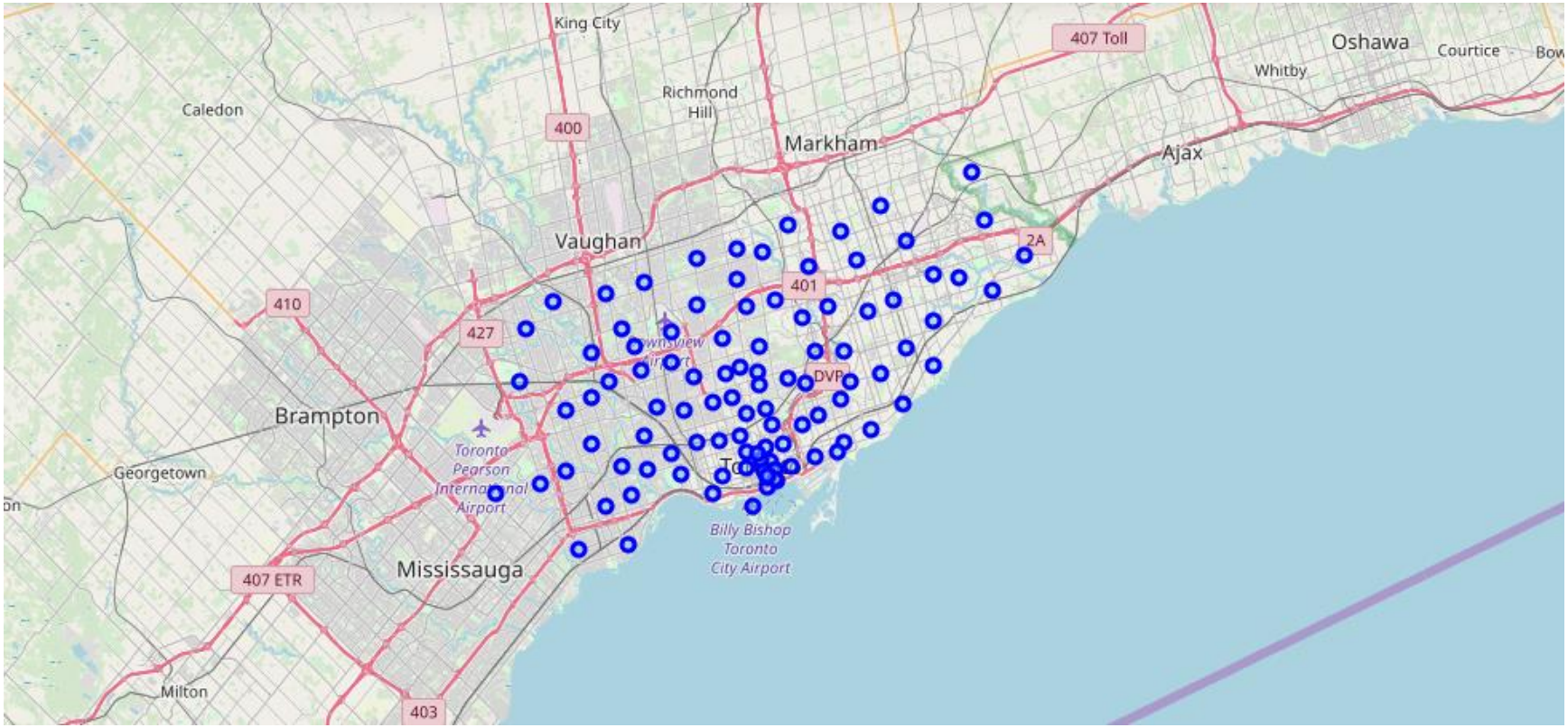
Data acquisition and cleaning

- The data for Toronto boroughs and neighbourhoods came from Wikipedia website (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M).
- The data related to Toronto's demographic data was pulled out form (https://en.wikipedia.org/wiki/Demographics_of_Toronto_neighbourhoods).
- I used Beautiful Soup 4 library to scrape the information from the websites. Beautiful Soup is a Python library for pulling data out of HTML and XML files.

Data acquisition and cleaning (continue)

- During the process of cleaning the data any cell has a borough but a Not assigned neighbourhood, then the neighbourhood will be the same as the borough.
- I used the Geopy library to get the Latitude and Longitude values of Toronto.
- I used folium library to build a map for Toronto with the neighbourhoods superimposed on top.
- Folium library builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the leaflet.js library.

Toronto's neighbourhoods superimposed on top of the map



The approach and methodology

- Foursquare APIs were used to explore the neighbourhoods and to segment them based on the top ten venues.
- Foursquare is a social location service that allows users to explore the world around them.

The approach and methodology (Continue)

- The venues were counted for each Borough.

```
In [31]: 1 toronto_venues.groupby('Neighbourhood').count()
```

Out[31]:

Neighbourhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Adelaide, King, Richmond	100	100	100	100	100	100
Bathurst Quay, CN Tower, Harbourfront West, Island airport, King and Spadina, Railway Lands, South Niagara	16	16	16	16	16	16
Berczy Park	56	56	56	56	56	56
Brockton, Exhibition Place, Parkdale Village	21	21	21	21	21	21
Business Reply Mail Processing Centre 969 Eastern	20	20	20	20	20	20
Cabbagetown, St. James Town	43	43	43	43	43	43
Central Bay Street	88	88	88	88	88	88
Chinatown, Grange Park, Kensington Market	100	100	100	100	100	100
Christie	15	15	15	15	15	15
Church and Wellesley	87	87	87	87	87	87
Commerce Court, Victoria Hotel	100	100	100	100	100	100

The approach and methodology (Continue)

- Then the top 5 most common venues in each neighbourhood were found.
- k-means cluster was run to cluster the neighbourhood into 10 clusters.

k-means cluster

```
In [31]: 1 toronto_venues.groupby('Neighbourhood').count()
```

Out[31]:

Neighbourhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Adelaide, King, Richmond	100	100	100	100	100	100
Bathurst Quay, CN Tower, Harbourfront West, Island airport, King and Spadina, Railway Lands, South Niagara	16	16	16	16	16	16
Berczy Park	56	56	56	56	56	56
Brockton, Exhibition Place, Parkdale Village	21	21	21	21	21	21
Business Reply Mail Processing Centre 969 Eastern	20	20	20	20	20	20
Cabbagetown, St. James Town	43	43	43	43	43	43
Central Bay Street	88	88	88	88	88	88
Chinatown, Grange Park, Kensington Market	100	100	100	100	100	100
Christie	15	15	15	15	15	15
Church and Wellesley	87	87	87	87	87	87
Commerce Court, Victoria Hotel	100	100	100	100	100	100

k-means cluster (Continue)

- The following is the visualization of the resulting clusters

Demographic data analysis

- The demographic data for Toronto was scrapped. There was no missing data points in the cells. All of the data types were object types, so we needed to convert some types to either float or integer types for further analysis of the data.

```
Data columns (total 13 columns):
Name                                                175 non-null object
FM                                                  175 non-null object
Census Tracts                                       175 non-null object
Population                                          175 non-null object
Land area (km2)                                     175 non-null object
Density (people/km2)                                175 non-null object
% Change in Population since 2001                   175 non-null object
Average Income                                      175 non-null object
Transit Commuting %                                 175 non-null object
% Renters                                           175 non-null object
Second most common language (after English) by name 175 non-null object
Second most common language (after English) by percentage 175 non-null object
Map                                                 175 non-null object
dtypes: object(13)
```

Demographic data analysis (Continue)

- We can see that we have some columns with commas in their cells. The commas will create issues for us when we convert the type from object to either float or integer. So, we need to remove the commas and convert the data type. The following is the data type after conversion.

Name	object
FM	object
Census Tracts	object
Population	int32
Land area (km2)	object
Density (people/km2)	float64
% Change in Population since 2001	object
Average Income	float64
Transit Commuting %	object
% Renters	object
Second most common language (after English) by name	object
Map	object
Percentage	float64
Language	object

Demographic data analysis (Continue)

- Some of the unnecessary data was dropped from the data frame and the Demographic data was appended with the coordinates.

Neighbourhood	Borough	Latitude	Longitude	Cluster Labels	Population	Density (people/km2)	Average Income	Transit Commuting %	Percentage	Language
The Beaches	East Toronto	43.676357	-79.293031	7	20416	5719.0	67536.0	13.8	0.7	Cantonese
Lawrence Park	Central Toronto	43.728020	-79.388790	7	6653	1828.0	214110.0	8.3	0.8	French
Davisville	Central Toronto	43.704324	-79.388790	0	23727	7556.0	55735.0	26.0	1.5	Persian
Rosedale	Downtown Toronto	43.679563	-79.377529	7	7672	2821.0	213941.0	11.3	1.0	Unspecified Chinese
Church and Wellesley	Downtown Toronto	43.665860	-79.383160	6	13397	24358.0	37653.0	25.1	1.8	Spanish
St. James Town	Downtown Toronto	43.651494	-79.375418	0	14666	63765.0	22341.0	27.4	8.1	Filipino

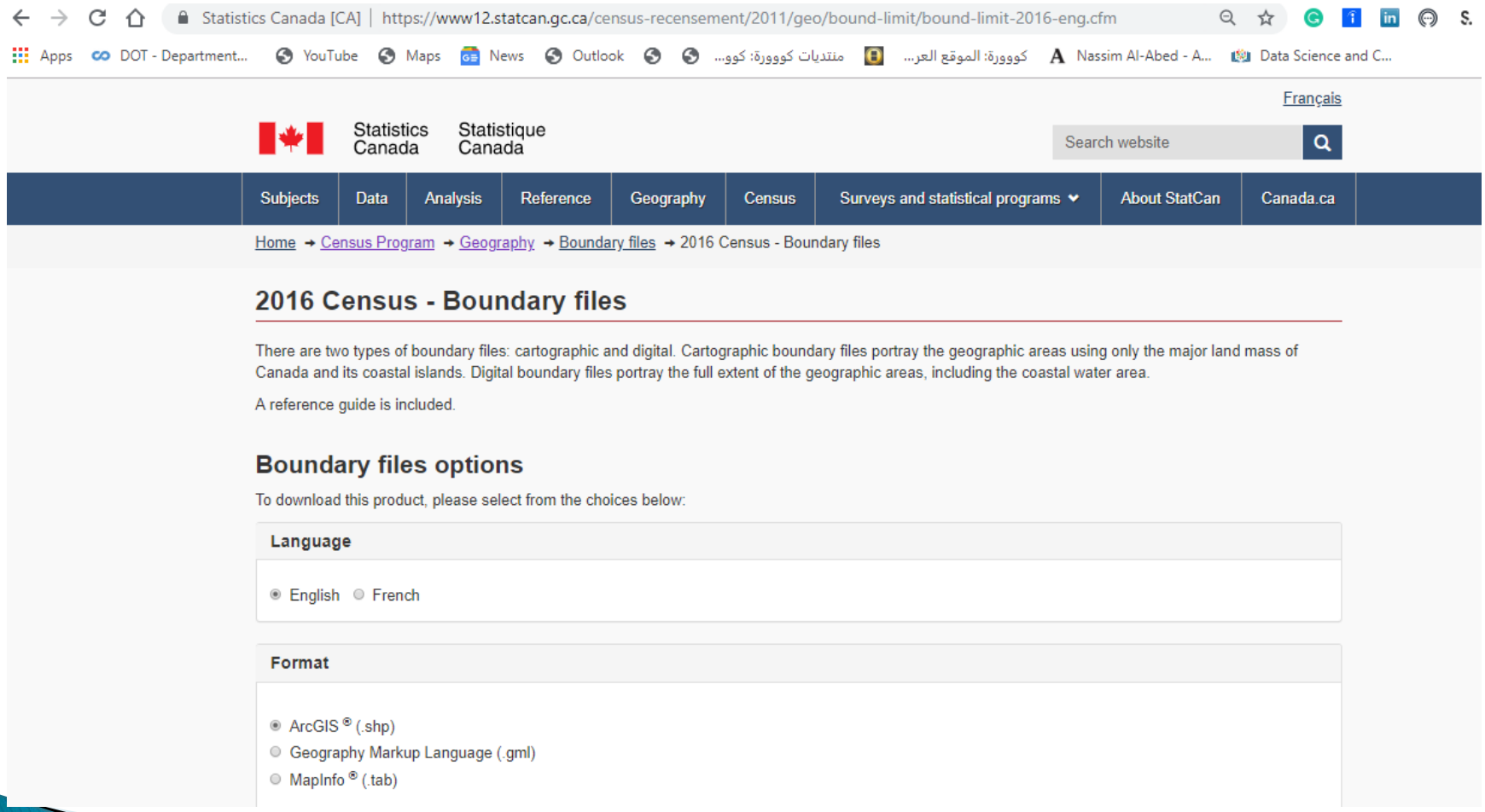
Creating choropleth map of population density

- In order to utilize the Python folium library in building a choropleth map of Toronto's population density we needed a Geojson file for Toronto. This file was not available and should be created.
- Canada Census population data from 2016 was downloaded and used in this project to create the Geojson file.
- Creating choropleth map for Toronto demographic data organized based on FSA (Forward Sortation Area—the first three digits of the Canadian Postal Code) utilizing Folium for creating these kind of maps requires a GeoJSON file as an input.
- Statistics Canada does publish some Census Boundary Files data for Canadian FSAs (available from this site: https://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/bound-limit-2016-eng.cfm).

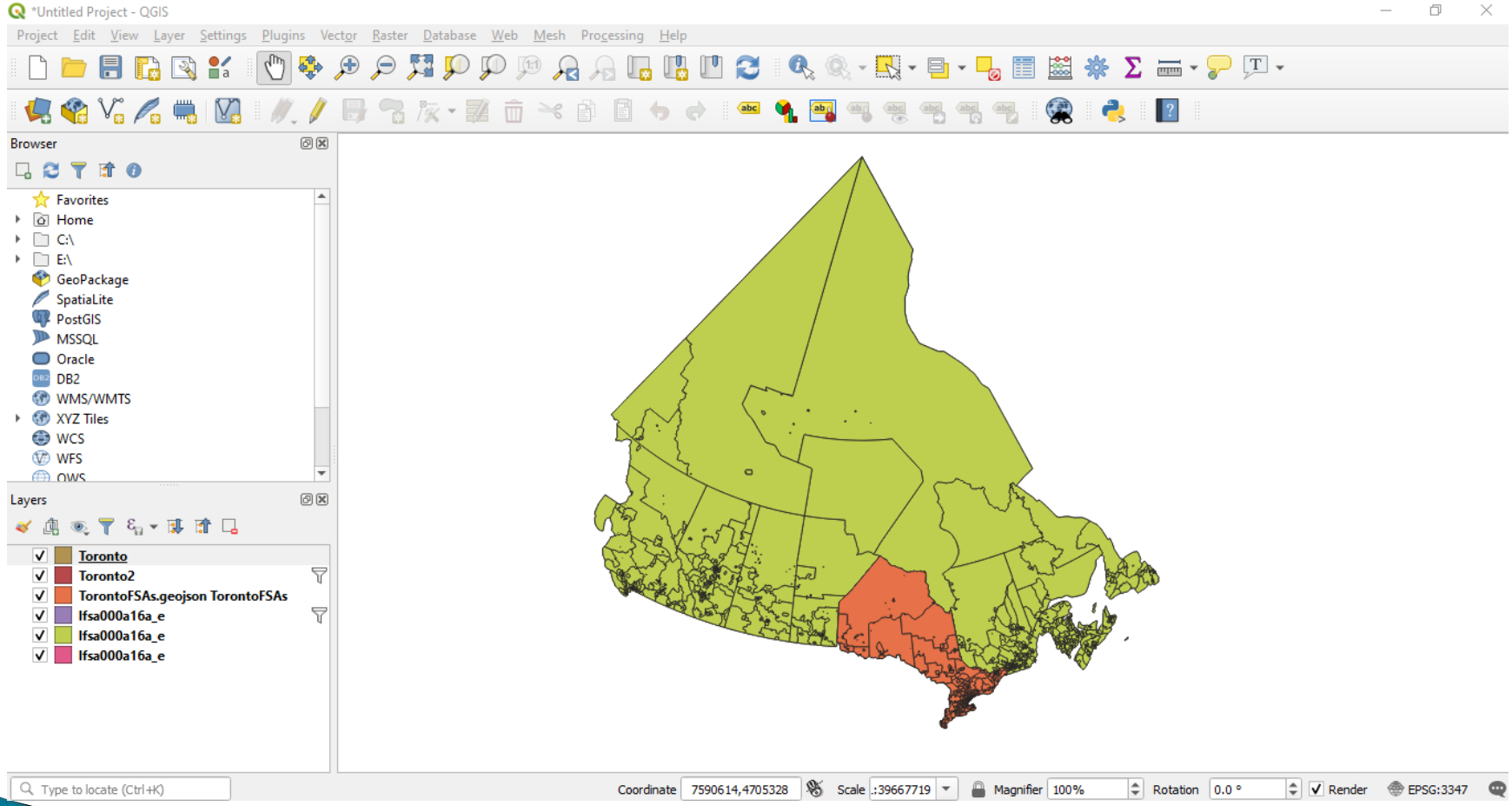
Creating choropleth map of population density (Continue)

- On Statistics Canada site I selected to save the option for downloading the data for the Forward Sortation Areas in ArcGIS shape file format.
- In ArcGIS software there is no direct method to convert the shape format to Geojson format. But, fortunately we can use QGIS software which is a free software to convert the shape file to Geojson.
- The downloaded file contains all of the Canadian statistics for 2016 and utilizing QGIS we had to filter the data for Ontario province and for Toronto.

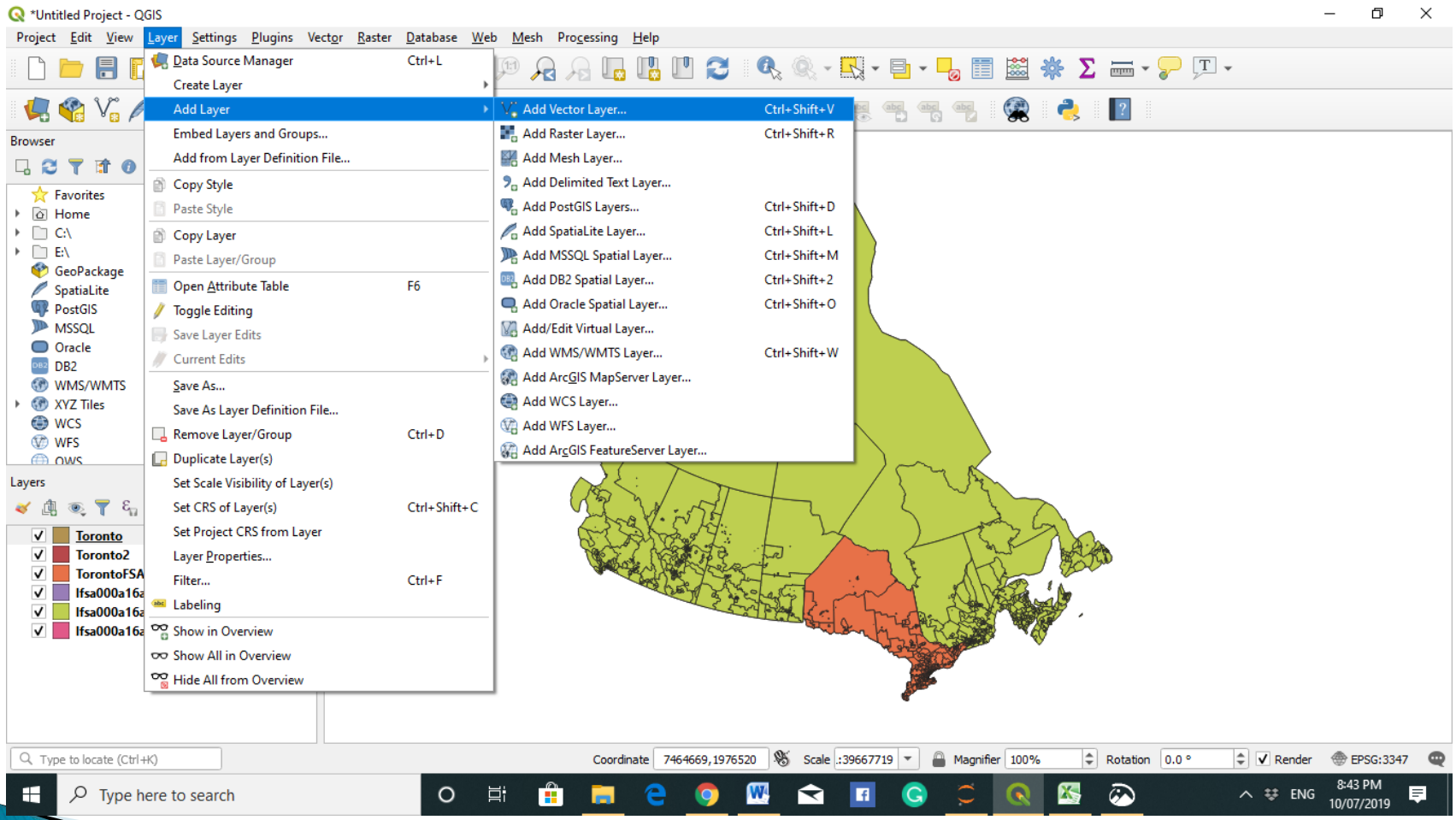
Downloading the 2016 census data



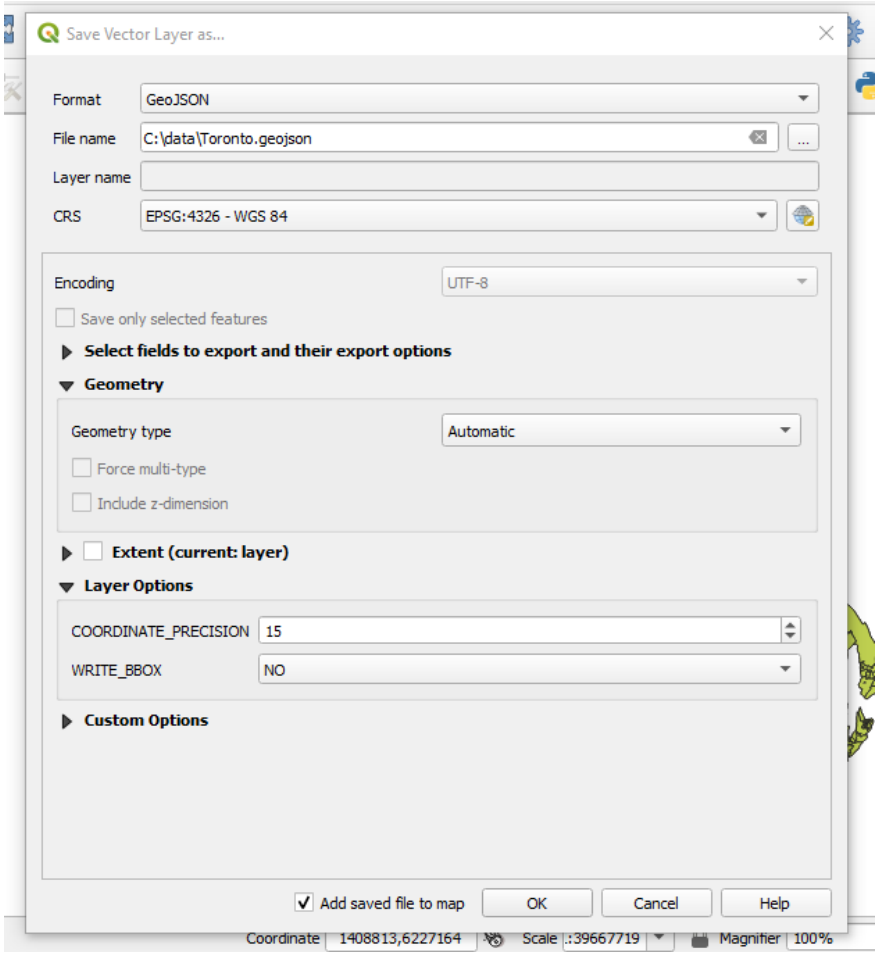
QGIS Software used to open and create a GeoJSON file for Toronto



Adding the shapefile to QGIS as a vector layer



Selecting the Geojson file format in QGIS



Conclusion and future directions

- The demographic data analysis of Toronto showed that the Chines language was spoken by more than 10% in North York.
- Spatial analysis utilizing folium library showed that North York had the highest population density in Toronto, even though there are around 30 Chinese restaurants in the area, North York fulfilled the conditions set at the start of the analysis.

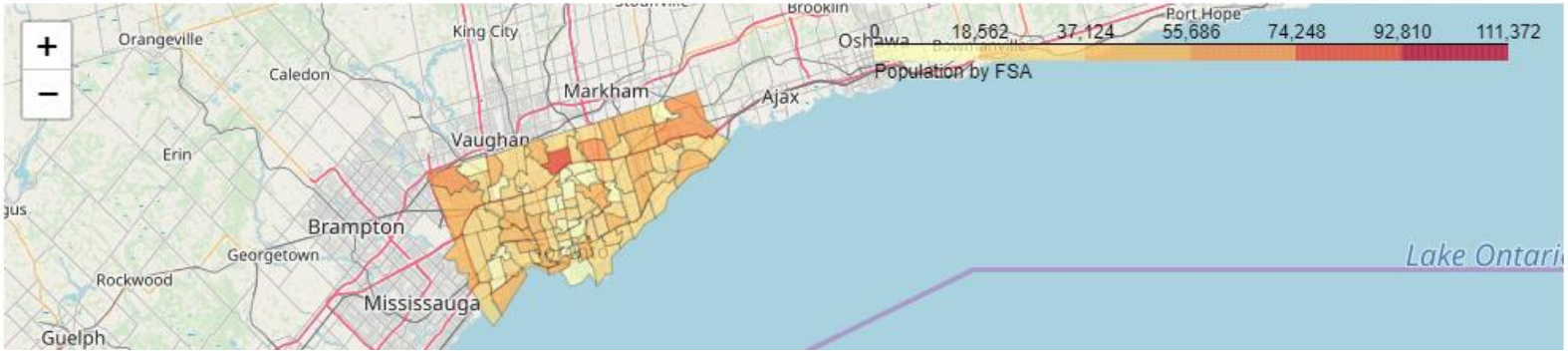
Drawing the population density for Toronto

In [133]:

```
#ontario_geo = "./data/NEIGHBORHOODS_WGS84.json"import pandas as pd
os.chdir('c:\\data')
map_toronto = folium.Map(location=[43.653963, -79.387207], zoom_start=10)
ontario_geo = "Toronto.geojson"
map_toronto.choropleth(geo_data=ontario_geo,
    data = df_pop,
    columns=['Geographic code','Population, 2016'],
    key_on='feature.properties.CFSAUID',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Population by FSA')

map_toronto
```

Out[133]:



Conclusion and future directions (continue)

- This project showed that data science could be used to conduct spatial data analysis utilizing Python and folium library.
- Utilizing QGIS software was necessary because I could not find any available Geojson file for Toronto. Geojson files are necessary in order to create choropleth maps.
- North York satisfied the selected site selection criteria set at the start of the project for the new restaurant.

Conclusion and future directions (continue)

- The utilization of Foursquare API was necessary in order to find Toronto's venues.
- Selecting the exact location of the new restaurant inside the Forward Sortation Area (FSA) was outside the scope of this project.