Udacity Project: OpenStreetMap Data Analysis Using SQL

Usman Rizwan

1 Introduction

I decided to look at the OpenStreetMap (OSM) data for Calgary, AB in Canada. As a Calgarian I am quite familiar with the street names and postal codes in Calgary. So I have a pretty good idea about how to fix the data types. The data set I used is 151 MB in size.

2 Data Wrangling

As soon as I downloaded the data I noticed that the street names were problematic. Most of the streets in Calgary end with NW, NE, SW and SE. These are the acronyms that are used on the street signs around the city but people usually write these names in a variety of ways. In the data set I found many instances of $Northeast$, $NorthEast$, $N.E.$, $N.E$, *ne*, *n.e.*, etc. I decided to write a function to standardize all addresses with cardinal directional to NW, NE, SW and SE.

```
mapping_to_right_name={‘Northeast’: ‘NE’,‘N.E.’: ‘NE’, ‘n.e.\n’: ‘NE’, ‘Northeast’: ‘NE’, ‘nw’:
‘NW’, ‘Northwest’: ‘NW’, ‘N.W’:‘NW’, ‘Northwest’: ‘NW’, ‘N.W.’: ‘NW’, ‘Southwest’: ‘SW’, ‘S.W.’:
‘SW’, ‘South-west’: ‘SW’, ‘Southeast’: ‘SE’, ‘S.E’: ‘SE’,  ‘South-east’: ‘SE’, ‘se’: ‘SE’}

#There is only once that a space in is used to specify South East
Wrong_names_with_spaces={‘South East’: ‘SE’}

#This function will correct the name and return it.
def update_name(name, mapping, map_names_with_spaces):
    name = name.split(" ")
    for i in range(len(name)):
        if name[i] in mapping:
            name[i] = mapping[name[i]]
    name = " ".join(name)
    for key in map_names_with_spaces:
        if key in name:
            name = name.replace(key, Wrong_names_with_spaces[key])
    return name
```

Another problem I noticed was with the postal codes. Postal codes in Calgary (and in Canada) are usually written as B2W 5N6. But in the OSM data, postal codes are written in a variety of ways, for e.g. B2W5N6, B2W-5N6, b2w 5n6, etc. I chose to standardize all the postal codes as B2W 5N6 because that is format used on Government of Canada's mail. The function I used to standardize the postal code was:

```
#This function will correct the postal code and return it.
def update_postal_code(postal_code):
    LOWER_COLON = re.compile(r’^[a-z]|_-+:’)
    if(len(postal_code) == 6):
        postal_code = [postal_code[:3],postal_code[3:] ]
    else:
        postal_code = postal_code.split("-")
    postal_code = " ".join(postal_code)
    postal_code = postal_code.upper()
    return postal_code
```

These were the two main problems with the data sets and both were easily progmattically fixable. The next step was to read the data into SQL tables. For this I constructed five csv files. The details of how the csv tables were constructed are given in the $shape\_elements(elem)$ and $audit(osmfile)$ functions in the code. The csv files were then read into SQL. See Fig. 1 to see the schema of the tables used.

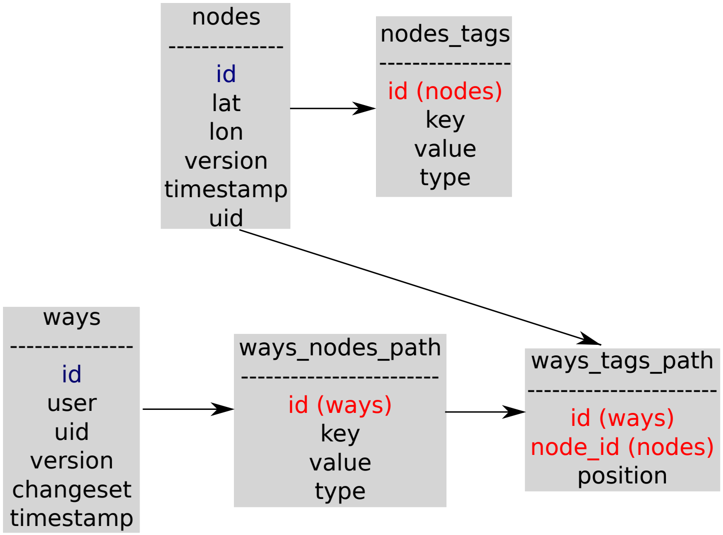

Figure 1: A diagram showing the relationship between schemas used for data analysis. Blue is for primary keys and red is for foreign keys.

3 Results

The first I looked at in the SQL table was the total number of nodes in the data set using the query:

```
SQL_query: ‘‘‘SELECT COUNT(id) FROM nodes’’’
```

There are 700200 nodes in the data set.

The next thing I looked at was the number of ways in the data set using the query:

```
SQL_query: ‘‘‘SELECT COUNT(id) FROM ways’’’
```

There are 85816 ways in the data set.

The next thing I was curious about was the number of Tim Hortons and Starbucks in the city. I also wanted to know their longitudinal and latitudinal locations for some data visualization. To find the number of Tim Hortons in the city I used the following query:

```
SQL_query: ‘‘‘SELECT COUNT(*) FROM (SELECT id, key, value FROM nodes_tags WHERE value = ‘Tim
Hortons’ GROUP by id’’’
```

There are 51 Tim Hortons in Calgary according to OSM. To get the location of each Tim Hortons in the city I used the following query:

```
SQL_query: ‘‘‘SELECT lon, lat
FROM nodes
JOIN nodes_tags ON nodes.id = nodes_tags.id
WHERE value = ‘Tim Hortons’
GROUP BY nodes.id’’’
```

The result from this query was converted into pandas data frame. To get the number of Starbucks locations and their coordinates the following queries were used:

```
SQL_query: ‘‘‘SELECT COUNT(*) FROM (SELECT id, key, value FROM nodes_tags WHERE value =
‘Starbucks’ GROUP by id’’’
SQL_query: ‘‘‘SELECT lon, lat
```

```
FROM nodes
JOIN nodes_tags ON nodes.id = nodes_tags.id
WHERE value = ‘Starbucks’
GROUP BY nodes.id’’’
```

There are 27 Starbucks in the city. As for Tim Hortons locations, the locations of Starbucks were written into pandas data frame. I used the folium package in python to visualize the locations of the Tim Hortons and Starbucks in the city (see Fig. 2).

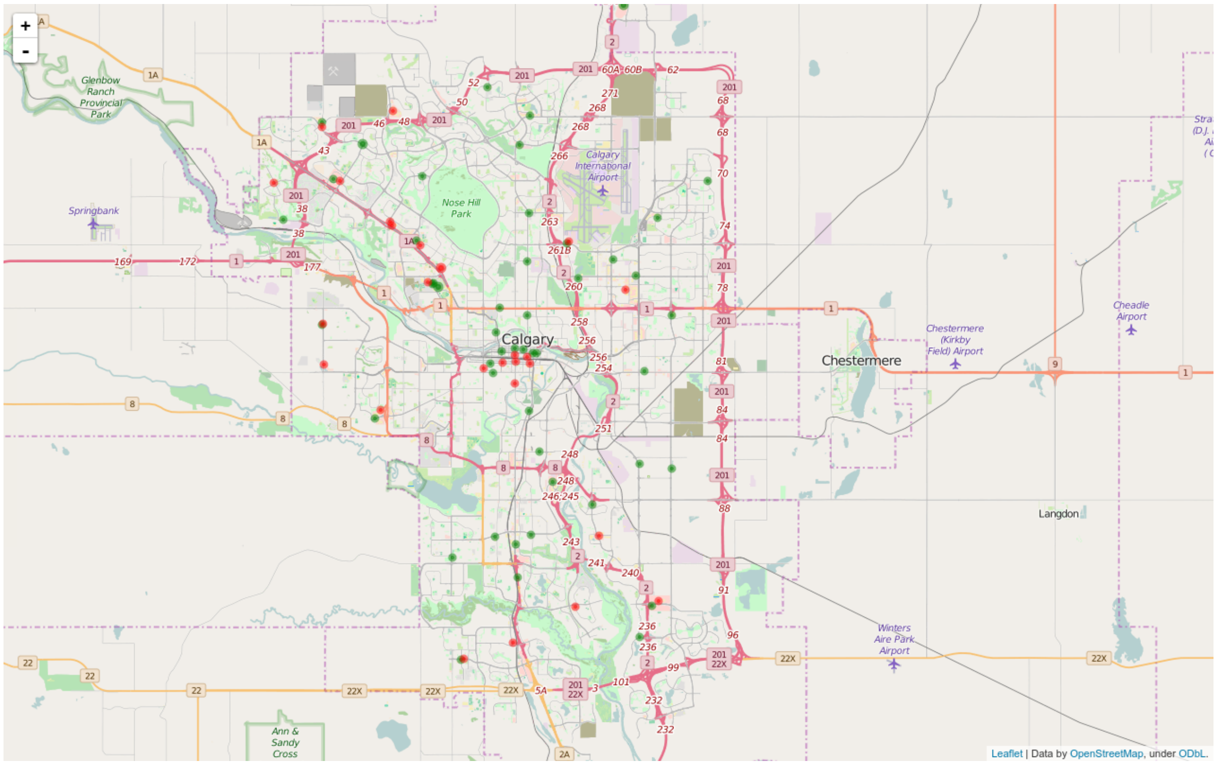
+
-
Glenbow Ranch Provincial Park
Springbank
Nose Hill Park
Calgary International Airport
Calgary
Chestermere
Chestermere (Kirkby Field) Airport
Cheadle Airport
Langdon
Winters Aire Park Airport
Ann & Sandy Cross
Leaflet | Data by OpenStreetMap, under ODbL.

Figure 2: Locations of Tim Hortons and Starbucks in Calgary, AB. Tim Hortons locations are shown as green dots while Starbucks locations are shown as red dots.

As expected downtown area has the highest density of Tim Hortons and Starbucks. The OSM data suggests that there are no Tim Hortons or Starbucks in or around the International airport but I know this to be untrue as I have personally seen and bought coffee from the Tim Hortons and Starbucks in the Calgary International airport.

The next thing I wanted to look at was the total number of contributors. To find the total number of contributors I used the following query:

```
SQL_query: ‘‘‘SELECT COUNT(DISTINCT uid) FROM ways;’’’
```

This query gives the correct result because uid is unique to each user, so any user who has made at least one commit to the OSM data of Calgary is counted. The total number of contributors to OSM map of Calgary 594. This number is surprisingly low. I wanted to see how much contributions each contributor was making. So I plotted a histogram of the number of contributions made by contributors (see Fig. fig:CvsCo). It seems most people make make less than 100 contribution and then there are very few dedicated users who make 1000’s of contributions.

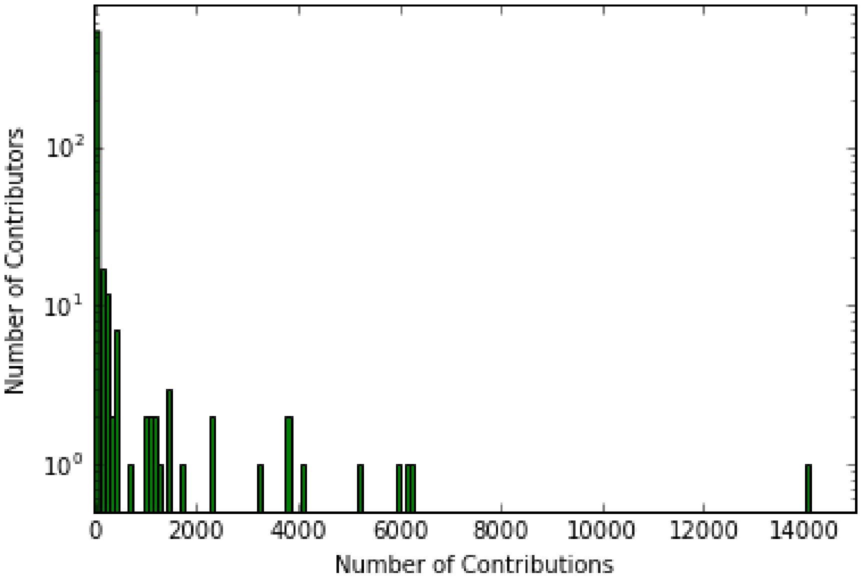

Figure 3: A histogram of the number of contributions made by each contributor. Bin size = 100

4 Conclusion

I was surprised to learn about the number of Starbucks in the city. I have seen very few Starbucks around town. Most of the Starbucks are around the downtown area in Calgary, which is not surprising. Though it is surprising how few people in Calgary have contributed to the OSM project. The data has very few details especially about the suburbs away from the city center.

The data seems to contain very few locations that are labelled as to whether they are wheelchair accessible or not. Labelling wheelchair accessible places would be an important contribution to the project and would also be a great help for people who use wheelchairs.

The data also seems to be lacking information about public transport. This would be something that the Calgary public transport should take an interest in and fill in the gaps regarding this information.

5 References

- I used a variety of pages on http://stackoverflow.com/ for help.
- I also used a variety of help pages on folium to figure out how to make the map. I got the idea to use the folium package from this blog https://blog.dominodatalab.com/creating-interactive-crime-maps-with-folium/