Toronto Open Street Map

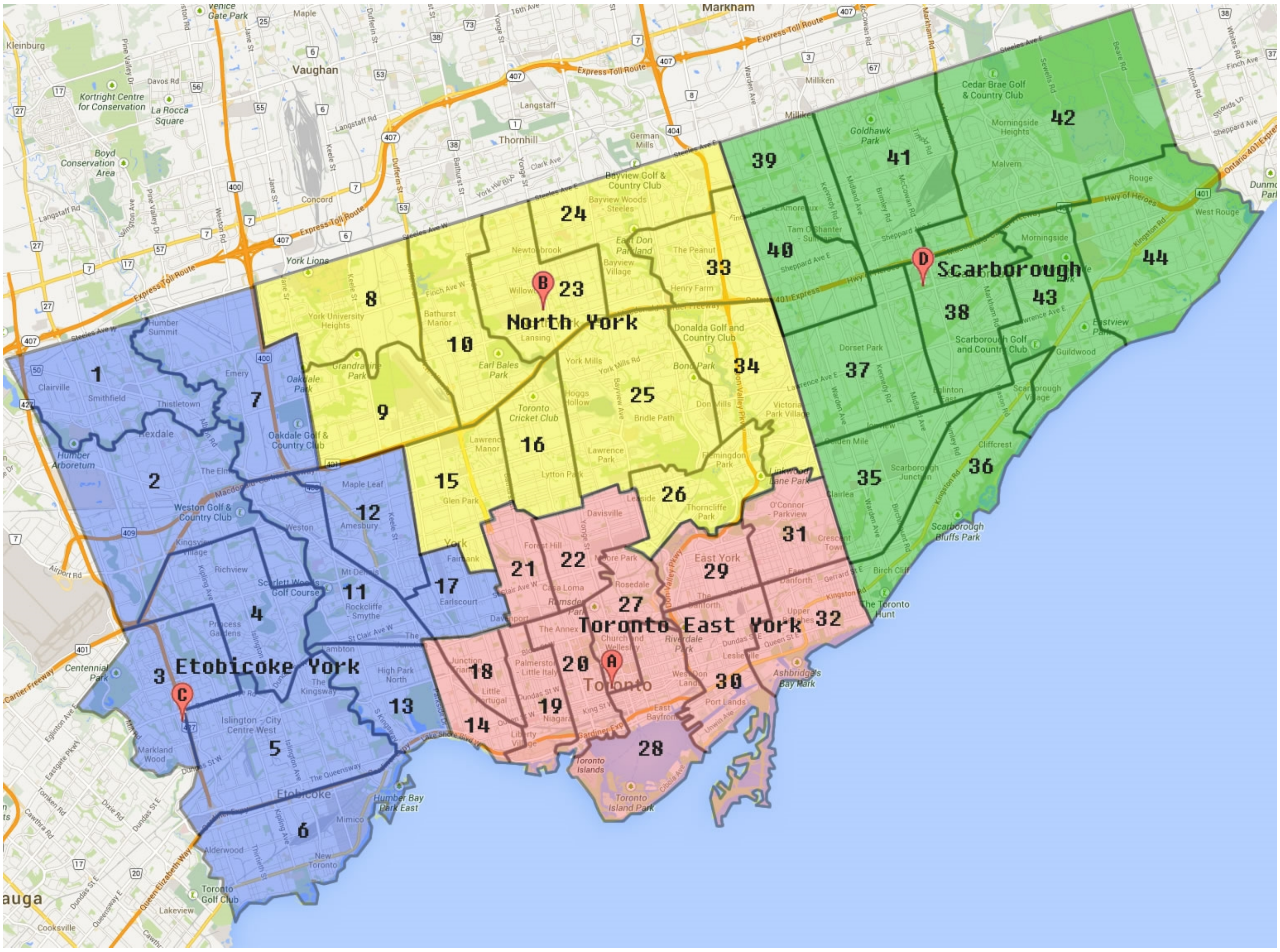
Dataset:

I chosed Torento, Canada ..as I planned to visit it in 2018.

> Toronto is the most populous city in Canada and the provincial capital of Ontario. With a population in 2016 of 2,731,571 Wikipedia (https://en.wikipedia.org/wiki/Toronto)

Note: I didn't use sample data set as I have issuee with slow internet connection, I tryed to download extract sample data set but the extraction size was more than 50MB so I decide to save time and used it as full data set.

- Full Data Set: (https://mapzen.com/data/metro-extracts/metro/toronto_canada/ (https://mapzen.com/data/metro-extracts/metro/toronto_canada/))



Project:

Files:

- Quiz folser/:has all snippets that I used to solve quizes
- Toronto.osm : dataset file
- Project snippits folder/ : has all codes that I used in project to parse and audit part of data in order to write final code.
- Audit.py : the final actual codes that parse Toronto.osm file, audit, shape data and then export to CSV files.
- CSVtoSQl.py: codes to read csv files then export to SQL database, after that do some quiries on it.
- Final Project 0.2 PDF
- schemaDB.py file

Files sizes:

- Toronto.osm :184 MB
- nodes.csv : 43.5 MB
- nodes_tags.csv : 27.3 MB
- ways.csv : 2.82 MB
- ways_nodes.csv : 32.9 MB
- ways_tags.csv : 27.7 MB

1. About:

OpenStreetMap, the project that creates and distributes free geographic data for the world.* (http://wiki.openstreetmap.org/wiki/Main_Page) the project transfer real world object into conceptual data model as following:

- **nodes:** define a specific point on earth surface.
- **ways:** define linear features such as rivers and roads and area boundaries.
- **relation:** multi-purpose data structure explain how other elements work together.
- **Tag:** all previous data element has tag that describe the meaning of particular element. OpenstreetMap data can export to many format, what I have to choose for this project is OSM XML file format. * (http://wiki.openstreetmap.org/wiki/Elements)

To understand the dataset more, I wrote code to count each elements in my dataset, the code is in file name UnqiueTags.py, it shows that my dataset has following:

- **member :** 45097 memebr elements
- **nd :** 1426164 nd elements
- **node :** 1227262 nodes
- **osm :** 1 element
- **relation :** 3377
- **tag :** 1524327
- **way :** 220214

That means the dataset has 122,726,2 nodes or in another words, places from Toronto.

2.Aduit:

From OpenStreetMap wiki page, it says that each Tag element has two free format text fields: a "key" and a "value". Key is unique, any elements cannot have 2 or more tags that hass the same key!
In order to understand Keys in Tgas elements more, I wrote code that show patterns of keys values and show the list of keys.In my dataset I have 42 keys values and 21 of them are valid lower case values, 18 one of them has colon in the name, and only 3 of them has one or more of problamtic characters for example ?%#$@. This result I had when I run the file name TagsType.py. Also, the code shows all Tags keys values as list:

- 'highway',
- 'addr:city',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:street',
- 'amenity',
- 'cuisine',
- 'name',
- 'outdoor_seating',
- 'phone',
- 'smoking',
- 'takeaway',
- 'addr:city',
- 'addr:country',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:state',
- 'addr:street',
- 'name',
- 'phone',
- 'shop',
- 'source',
- 'amenity',
- 'cuisine',
- 'name',
- 'highway',
- 'addr:housename',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:street',
- 'amenity',
- 'name', -'addr:housenumber',
- 'addr:street',
- 'addr:street:name',
- 'addr:street:prefix',
- 'addr:street:type',
- 'building', -'building:levels',
- 'chicago:building_id',
- 'restriction',
- 'type',
- 'highway',
- 'addr:city',
- 'addr:housenumber',

- 'addr:postcode',
- 'addr:street',
- 'amenity',
- 'cuisine',
- 'name',
- 'outdoor_seating',
- 'phone',
- 'smoking',
- 'takeaway',
- 'addr:city',
- 'addr:country',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:state',
- 'addr:street',
- 'name',
- 'phone',
- 'shop',
- 'source',
- 'amenity',
- 'cuisine',
- 'name',
- 'highway',
- 'addr:housename',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:street',
- 'amenity',
- 'name',
- 'addr:housenumber',
- 'addr:street',
- 'addr:street:name',
- 'addr:street:prefix',
- 'addr:street:type',
- 'building',
- 'building:levels',
- 'chicago:building_id',
- 'restriction',
- 'type',
- 'highway',
- 'addr:city',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:street',
- 'amenity',
- 'cuisine',
- 'name',
- 'outdoor_seating',
- 'phone',
- 'smoking',
- 'takeaway',
- 'addr:city',

- 'addr:country',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:state',
- 'addr:street',
- 'name',
- 'phone',
- 'shop',
- 'source',
- 'amenity',
- 'cuisine',
- 'name',
- 'highway',
- 'addr:housename',
- 'addr:housenumber',
- 'addr:postcode',
- 'addr:street',
- 'amenity',
- 'name',
- 'addr:housenumber',
- 'addr:street',
- 'addr:street:name',
- 'addr:street:prefix',
- 'addr:street:type',
- 'building',
- 'building:levels',
- 'chicago:building_id',
- 'restriction',
- 'type'

For audti, I will mainly foucus on Streey type, phone, and postcode.

2.1 Street Type:

The idea or concept behind AuditStreetType.py file is to serach Tags keys values pattern against Regex pattern and show the one who isn't correct in order to build Mapping dictionary to correct and update dataset later.
The issues I have in my dataset when I call audit function from AuditStreetTyep.py file:
1. Abbriviation:

- Ave
- E
- Pl
- Dr
- St
- W
- Rd
- Hwy
- Blvd
- St.
- E.
- ave
- Pkwy
- Pl
- ST
- W.
- Wynd
- st.
- rd

2. Lower case letters:

- avenue
- street
- west

To solve this issue, first I had to build mapping dictionary to use it to correct issues. The data for mapping dictionary I took it from City Of Toronto government webpage. They created road naming and classification docuemnt for Toronto.
Road Classification System- City of Toronto Street Name Index
(https://www1.toronto.ca/City%20Of%20Toronto/Transportation%20Services/Road%20Classification%20Syste
wide_index.pdf)

2.0 ABBREVIATIONS

Street Type	Full Name	Street Type	Full Name
Ave	Avenue	Sq	Square
Bdge	Bridge	St	Street
Blvd	Boulevard	Ter	Terrace
Crcl	Circle	Trl	Trail
Crct	Circuit	View	View
Cres	Crescent	Walk	Walk
Crt	Court	Way	Way
Cs	Close	Wds	Woods
Dr	Drive	Wood	Wood
Gdns	Gardens		
Grn	Green		
Grv	Grove		
Gt	Gate		
Hill	Hill		
Hts	Heights		
Lane	Lane		
Line	Line	**Street Direction**	**Full Name**
Lwn	Lawn	N	North
Mews	Mews	S	South
Path	Path	W	West
Pk	Park	E	East
Pkwy	Parkway		
Pl	Place		
Ramp	Ramp		
Rd	Road		
Rdwy	Roadway		

Mapping Dictionary:

In [3]:

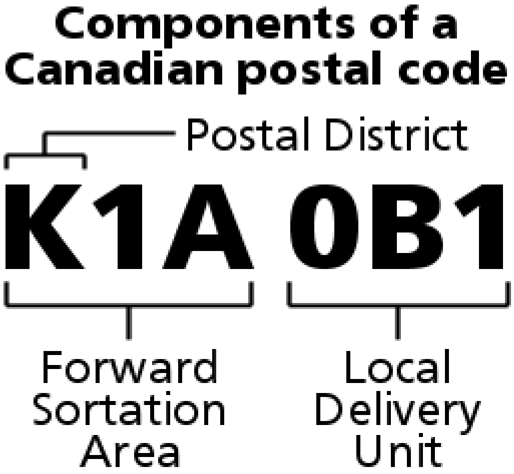
```
            #Before : After
mapping = {"Ave": "Avenue",
            "E": "East",
            "Pl": "Place",
            "Dr": "Drive",
            "St":"Street",
            "W":"West",
            "Rd":"Road",
            "Hwy":"Highway",
            "Blvd":"Boulevard",
            "St.":"Street",
            "E.":"East",
            "ave":"Avenue",
            "Pkwy":"Parkway",
            "ST":"Street",
            "W.":"West",
            "st.":"Street",
            "rd":"Road",
            "avenue":"Avenue",
            "street":"Street",
            "west":"West"
            }
```

I wrote update_name function in AuditStreetType.py file that will read each street type name and check against mapping dictionary, if it is there, it will update it with the new correct value.For example:
Ave, **ave**, **avenue** all will be correct to **Avenue** and so on. after running the code, one of the result is like that to comapre
Street type: avenue,street name befor: Ryerson avenue,and street name after: Ryerson Avenue

---

2.2 Postal Code:

Candian postal code is a six characters alphanumeric string, with a space between third and forth chracter.All letters are capital letters and Toronto postal codes strat with M leeter.Resource (https://en.wikipedia.org/wiki/Postal_codes_in_Canada)



I wrote a code that parse postal code values and check for issues, I encounter 3 issues:

- Some postal code values dooesn't have space between third and forth characters.
- Some postal code values doesn't start with M letter which indicated Toronto
- Other postal code values has more than one value.

How I solved them:

2.2.1: Postal code that doesn't have space:
I wrote code to add space between third and forth characters.

In [ ]:

```
elif postal_code[0]=="M" and len(postal_code)==6:
            postal_code=' '.join(postal_code[i:i+3] for i in range(0, 6, 3))
```

2.2.2: Postal code that doesn't start with M:
I wrote code to skip this value and not to add its element/tag/node to list as it doesn't belong to Toronto

2.2.3: Postal code that has more than one value:
I wrote code to split and take only the first value

---

3. Export to SQL database and some queries:

I wrote code to export CSV files to SQL databse in memeory in order to work on them later. File name is CSVtoSQl.py

Number of nodes:1227262 nodes

In [2]:

```
cur.execute('SELECT count(*) FROM nodes ')
print cur.fetchone()
```

```
(1227262,)
```

Number of ways:220214 ways

In [8]:

```
cur.execute('SELECT count(*) FROM ways ')
print cur.fetchone()
```

```
(220214,)
```

The most frequent Postal code is: M4P 1E4. When I serach about this postal code using Area Codes website (http://www.area-codes.net/canada) it shows that it is postal code for Yonge Eglinton Centre.

In [9]:

```
cur.execute('select value from (select value, count(1)  from nodes_tags where key="post
code" group by value order by count(1) desc) ')
print cur.fetchone()
```

```
('M4P 1E4',)
```

Number of cusines in dataset is 200 cusinie.

In [68]:

```
cur.execute('SELECT distinct(value) FROM nodes_tags where key="cuisine" ')
cuisine=cur.fetchall()
len(cuisine)
```

Out[68]:

```
200
```

The top 10 most frequent cuisines in Toronto are:

- Coffee Shop :510
- Pizza : 271
- Sandwich: 237
- Burger: 145
- Chinese: 97
- Japanese: 75
- Chicken: 55
- Indian: 53
- Italian: 51
- Sushi: 45

In [5]:

```
cur.execute('SELECT value, count(id) as count_cuisine FROM nodes_tags where key="cuisin
e"  group by value order by count_cuisine DESC Limit 10 ')
top_ten_cuisine=cur.fetchall()
top_ten_cuisine
```

Out[5]:

```
[('coffee_shop', 510),
 ('pizza', 271),
 ('sandwich', 237),
 ('burger', 145),
 ('chinese', 97),
 ('japanese', 75),
 ('chicken', 55),
 ('indian', 53),
 ('italian', 51),
 ('sushi', 45)]
```

I'm interest to check for all Pizza resturants in Toronto, there are 89 diffrent brand.

In [115]:

```
cur.execute('SELECT distinct(value) FROM nodes_tags where key="name"  and value LIKE "%
pizza%"  and value NOT LIKE "%www.%" and value NOT LIKE "%.c%" or value LIKE "%PIZZA%"
 and value NOT LIKE "%www.%" and value NOT LIKE "%.c%" or value LIKE "%Pizza%" and valu
e NOT LIKE "%www.%" and value NOT LIKE "%.c%"')
pizza=cur.fetchall()
pizza
```

Out[115]:

```
[('Pizza Pizza',),
 ('pizza',),
 ('Pizza Nova',),
 ('Papa Ceo Pizza',),
 ("Cora's Pizza",),
 ("Gino's Pizza Bar and Grill",),
 ('Diamond Pizza',),
 ('Mamma Pizza',),
 ('241 Pizza - gone bankrupt - franchisee wanted sign',),
 ('241 Pizza',),
 ("Mamma's Pizza",),
 ('Pizza Hut',),
 ('Pizza Rustica Restaurant & Bar',),
 ('Boston Pizza',),
 ("Domino's Pizza",),
 ('Pizza Hut/Wing Street',),
 ('Pizzaiolo',),
 ('Bona Pizza & Pasta',),
 ('Blaze Pizza',),
 ('Brass Taps Pizza Pub',),
 ('Zizi Pizza',),
 ('Pizza Lambretta',),
 ('Pizzaville',),
 ('Former 241 Pizza',),
 ('Pizza Park',),
 ("Regino's Pizza",),
 ('Pizza Pide',),
 ('Pizza to Go',),
 ("Angelo's Coal Fired Pizza",),
 ('Village Pizza n Burgers',),
 ('Pizza Mom',),
 ("Gino's Pizza",),
 ('Martino Pizza',),
 ('Queenslice Pizza & Pita',),
 ('Olympic 76 Pizza',),
 ('Danforth Pizza House',),
 ("Bigman's Gourmet Pizza",),
 ('Express Pizza',),
 ('Boccone Deli & Pizza Bar',),
 ('Express Pizza & Grill',),
 ("Yogi's Pizza and Wings",),
 ('Pizza Pan',),
 ('Fresh Pizza Plus',),
 ("Dino's Wood Burning Pizza",),
 ('Thyme 4 Pizza & Pasta',),
 ('Sempre Pizza and Pasta',),
 ("Romi's Pizza & Ristorante",),
 ('PI CO. Pizza Bar',),
 ('24 Pizza',),
 ('Pizza Hut Express',),
 ('Tim Hortons, Mr. Sub, Pizza Pizza, Hero Certified Burgers',),
 ("Chito's Pizza",),
 ('Bocconcini Pizza and Wings',),
 ('Fresh Slice Pizza',),
 ('Pizza Del Arte',),
 ('Mr. Pizza',),
 ('Gelato Pizza',),
 ('Pizza Gigi',),
 ('Pizza Depot',),
 ('yummy pizza',)
 ('yummy pizza',),
 ('Planet Pizza',),
 ('Fresca Pizza & Pasta',),
 ('Pure Pizza and Burger',),
 ('pizza_and_Burgers',),
 ("George's Pizza",),
 ("Papa John's Pizza",),
 ('City Fried Chicken & Pizza',),
 ("Pizza L'Amore",),
 ('World Famous Pizza',),
 ('Midland Pizza & Wings',),
 ('Pizza King',),
 ('Queen Margherita Pizza',),
 ('italian_pizza',),
 ('Pizza Hot Wings',),
 ("Papino's Pizza",),
 ('Parliament Panzerotti Pizza',),
 ('Turtle Pizza & Wing',),
 ("Ginoi's Pizza",),
 ('pasta;pizza',),
 ('Vivo Pizza + Pasta',),
 ('Maple Pizza',),
 ('Fast food restaurant, that serves delicious pizza, and offers good tast
ing wings, french fries; and cool and refreshing drinks.',),
 ('armenian;pizza',),
 ("Milano's Pizza",),
 ('Pizza E Pazzi',),
 ('italian;pizza',),
 ('Casual eatery with modern decor serving wood-fired, Neapolitan pizzas,
 pasta dishes & Italian wines.',),
 ("Robin's Pizza & Wings",),
 ('pizza;chicken',)]
```

The top Pizza resturants in Toronto based on number of branches is Yummy Pizza:

In [129]:

```
cur.execute('SELECT value FROM nodes_tags where key="name"  and value LIKE "%pizza%"  a
nd value NOT LIKE "%www.%" and value NOT LIKE "%.c%" or value LIKE "%PIZZA%"  and value
 NOT LIKE "%www.%" and value NOT LIKE "%.c%" or value LIKE "%Pizza%" and value NOT LIKE
 "%www.%" and value NOT LIKE "%.c%"')
pizza_resturant_list=cur.fetchall()
heapq.nlargest(1, zip(pizza_resturant_list))
```

Out[129]:

```
[(('yummy pizza',),)]
```

4. Suggestions:

After going through Toronto City official website, it shows clearly that the governor of Toronto and her/his team is put a lot of efforts to organize and provide full and consist information, so I think if they created a platform with reward program for thier citiznes to use to enter data about city and there is reward for them. The idea is simple: the city should create application for diffrent devices and anyone from Toronto city can download and validate her/his account by location and NID. After that, each user can caluclyate points for reward program by entering or validating/corecting data about Toronto map.

5. Conclusion:

When I started working on dataset, I noticed that this dataset was quite diffrent than example.osm dataset, as this dataset nodes doesn't has valuable information as wxample.osm dataset, for example Toronto.osm file doesn't have uid, user and changset information which caused to miss a huge amount of valuable information.

What to do to increase the quality of Toronto map data?As I suggest before, Toronto governers need to create application that work on all devices and intoduce reward program for thier people to help and enter data to increase qulaity of information. Benefits:

- Huge amounts of data will be gathered.
- Map will have infprmation about not famous places and restuants..etc
- Saving a lot of time and efforts.

Anticipated Issues:

- We will have huge amounts of data but if the process of collecting the data were correct and well defined and implemnt, we will have a lot of unwanted and unuseful data that would take a great deal of time in wrangling them.
- There's a high chance that people may not particioate or the collected data is not enough.