Data Wrangling with MongoDB

Project Summary

Name: Maureen Maria Rocas
Email: mau_rocas@hotmail.com
Map Area: Quezon City, Philippines
http://www.openstreetmap.org/export#map=12/14.6829/121.0618
Note: For some reason, the map boundaries are not consistent each time I clicked the URL above.
Please manually enter the following: latitude range: [14.5789 to 14.7868]
longitude range: [120.887 to 121.236]

Citations:

Open Street Map Wiki: https://wiki.openstreetmap.org/wiki/OSM_XML
Mongo DB Aggregation Reference: http://docs.mongodb.org/manual/reference/aggregation/
Others: Please check footnote section for the list of websites I referred to in this project.

*"I hereby confirm that this submission is my work. I have cited above the origins of **any** parts of the submission that were taken from Websites, books, forums, blog posts, github repositories, etc. By including this in my email, I understand that I will be expected to explain my work in a video call with a Udacity coach before I can receive my verified certificate."*

Quezon City is one of the cities that make up Metro Manila, the National Capital Region of the Philippines, on the island of Luzon. It is the most populous city in the country, and the largest city by area in Metro Manila.[1]

Quezon City is also the place of my birth. It is where I attended my primary and secondary schooling, where I went to church on Sundays and went shopping on weekends. Thus, I chose Quezon City because I know the streets and I have lived it. Furthermore, this project will allow me to make a contribution by making improvements to the quality of street maps of my beloved neighbourhood.

Problems Encountered

The Quezon City Open Street Map (OSM) is downloaded as an XML file. To audit this file, I created 2 Python programs, MapStructureAudit.py and MapContentAudity.py[2], which execute two types of audit: data structure and content. The data types and format of the nodes[3], ways[4] and relation[5] attributes are validated in the data structure audit; while the data content audit records all tag's "k" and "v" attribute and focuses on the address values: street, postal code and city.

Below are the some of the problems found:

---

[1] http://en.wikipedia.org/wiki/Quezon_City
[2] https://github.com/maurocas17/udacity/tree/master/UdacityDataWranglingProject. Contains the 3 python programs created to parse the OSM xml, add corrections and generate the json file. Please check the program's comment section for details of each class.
[3] https://wiki.openstreetmap.org/wiki/Nodes
[4] https://wiki.openstreetmap.org/wiki/Ways
[5] https://wiki.openstreetmap.org/wiki/Relations

- Unknown value for "k" and "v" attribute
- Uniformity of Street and City Names
- Incorrect Postal Codes and Incomplete data

Unknown value for "k" and "v" attribute

The "k" and "v" are attributes of the Tag element. There are 6 records found with empty "v" attribute. Referring to the DTD of OSM XML[6], "v" attribute is required; however, it is unclear that "v" can have an empty value. Based on Map Features[7], the "k" attribute only contain letters (a-z) and underscores (_). There are 20 records found that do not match this rule. Examples in the Quezon City data are: "street signs", "House No.", "MR.QUICKIE", and "Years in Business".

There is no concrete rule written in which to base the correctness of the "v" and "k" attributes. Nevertheless, it is good to note of such records so that they can be amended in the future. But for this project, records found were ignored and were not saved to MongoDB.

Uniformity of Street and City Names

There are a lot of uniformity issues found in the Street Names of Quezon City data, like using short or abbreviated format as oppose to long formats, using lower and upper case format, uniformity in street type suffix (i.e. Avenue, Street), and just containing the street name and nothing else. There are also misspelled street names found. Below are the details:

- Some street names have street types in abbreviated and non-abbreviated format; while some do not have the street type. To standardized street names, abbreviated street type will be converted to its long format (i.e. St to Street). Also, for streets that do not have street type, the type will be appended to them. However, this is a case to case basis. For example, the Quezon City data contains three formats for "Katipunan" street: "Katipunan", "Katipunan Ave." and "Katipunan Avenue". As I am familiar with Katipunan, I have corrected them to "Katipunan Avenue". However, this is not the case for "M L Quezon". Since I am not familiar with it and the downloaded map data contains conflicting formats: "M L Quezon", "M.L. Quezon Extension", and "M L Quezon St.", the street type was not added.
- Some streets are in abbreviated formats while some uses the long format. For example: EDSA[8] is short for Epifanio de los Santos Avenue. EDSA is corrected to use the long format.
- Some are just misspelled. Like "Tomast Morato" should be "Tomas Morato" and "Marcos Hoigway" should be "Marcos Highway".
- Some are all in lower case format like "old sauyo rd" which is corrected to "Old Sauyo Road".
- Some have values other than the street name.
    - Some have the city, housenumber, building, compound, and subdivision. Like "Durian Street, Palmera Homes Subd" and "537 EDSA, Cubao"
    - Some have "Along", like "Along Commonwealth Avenue".
    - Some have "corner", like "Gen. Luna cor. Santa Potenciana Sts.
    - Some have "Infront", like "A. Bonifacio Infront of Shell gas station"

[6] https://wiki.openstreetmap.org/wiki/API_v0.6/DTD

[7] https://wiki.openstreetmap.org/wiki/Map_Features

[8] http://en.wikipedia.org/wiki/EDSA_%28road%29, EDSA is short for Epifanio de los Santos Avenue

- Some have “Intersection” like “Tayuman St. Intersection Tomas Mapua St”

These records were corrected to only have the street name. This means that “Durian Street, Palmera Homes Subd” is corrected to “Durian Street”.

Below are some of the problems found in the city names:

- City names in lower case format, like “caloocan”, is corrected to “Caloocan”
- Most of the city names do not have “City” appended to it. To standardize the city names, “City” is removed and just the name was retained. Like “Antipolo City” is corrected to “Antipolo”.
- City names with wrong spelling were corrected. Like “Manil” is corrected to “Manila”

Incorrect Postal Codes and Incomplete Data

The postal codes of Quezon City consist of 4 digits which all begin in “11”[9].

There are 89 unique postal codes found

```
> db.osm.distinct("address.postcode").length
89
>
```

Out of 89, 63 are invalid postcode.

```
> db.osm.aggregate([
... {$match:{"address.postcode":{"$exists":1, "$nin":[/^11/] } } },
... {$group:{_id:"$address.postcode", count:{$sum:1} }} ,
... {$group:{_id:null, count: {$sum:1} } }
... ])
{ "_id" : null, "count" : 63 }
>
```

That is 71%, which is huge! However, looking at the invalid postcode, I found out that they reside just outside or around the boundaries of Quezon City.

Furthermore, the Quezon City map is surrounded by other cities in all its sides[10]. This means that there are bound to be areas that are outside the city but will be included in the downloaded map as long as the areas lie within the boundaries of latitude "14.5789" to "14.7868" and longitude "120.887" to "121.236". Therefore, the postcodes are not really invalid. Nevertheless, it is still alarming that there are more address tags available outside Quezon City than within Quezon City, especially, when Quezon City has the largest area in Metro Manila.

To emphasize the incompleteness of the data for Quezon City, I queried for the total number of a well-known Burger store, Jollibee[11], in Quezon City, and I got 6. This is way less! There are more than 60 stores found in the city[12].

Overview of the Data

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

---

[9] http://zip-codes.philsite.net/quezon_city.htm

[10] http://en.wikipedia.org/wiki/Barangays_of_Quezon_City#mediaviewer/File:Political_Divisions_of_Quezon_City.png

[11] http://en.wikipedia.org/wiki/Jollibee

[12] http://www.munchpunch.com/jollibee/quezon-city

File Sizes:

qc.osm - 201 MB

qc.osm.json - 230 MB

Number of documents

```
> db.osm.find().count()
1080796
```

Number of nodes

```
> db.osm.find({type:"node"}).count()
888720
>
```

Number of ways

```
> db.osm.find({type:"way"}).count()
192076
```

Number of Unique Users

```
> db.osm.distinct("created.user").length
932
>
```

Top 5 Contributors

```
> db.osm.aggregate([
... {$group:{_id:"$created.user", count:{$sum:1}}},
... {$project:{
... contributions:"$count",
... "% contributions":{$multiply:[
...   {$divide:["$count", db.osm.find().count()]}, 100]}
... }  },
... {$sort:{"% contributions":-1}},
... {$limit:5}
... ])
{ "_id" : "jmbangate", "contributions" : 478390, "% contributions" : 44.26274708
6406684 }
{ "_id" : "maning", "contributions" : 171041, "% contributions" : 15.82546567529
8576 }
{ "_id" : "ianlopez1115", "contributions" : 71416, "% contributions" : 6.6077224
56411756 }
{ "_id" : "irenepicache", "contributions" : 67696, "% contributions" : 6.2635316
93307526 }
{ "_id" : "Rally", "contributions" : 43200, "% contributions" : 3.99705402314590
37 }
>
```

Total Contributions per Year

```
> db.osm.aggregate([
... {$project:{year:{$substr:["$created.timestamp", 0, 4]}}},
... {$group:{_id:"$year", contributions_per_year:{$sum:1}}},
... {$sort:{_id:-1}}
... ])
{ "_id" : "2015", "contributions_per_year" : 3032 }
{ "_id" : "2014", "contributions_per_year" : 670285 }
{ "_id" : "2013", "contributions_per_year" : 71137 }
{ "_id" : "2012", "contributions_per_year" : 107877 }
{ "_id" : "2011", "contributions_per_year" : 140691 }
{ "_id" : "2010", "contributions_per_year" : 33619 }
{ "_id" : "2009", "contributions_per_year" : 30809 }
{ "_id" : "2008", "contributions_per_year" : 23346 }
>
```

Number of users whose contributions ranges from 1-10

```
> db.osm.aggregate([
... {$group:{_id:"$created.user", count:{$sum:1}}},
... {$group:{_id:"$count", total_users:{$sum:1}}},
... {$project:{total_users:"$total_users",
...     "% user":{$multiply:[
...       {$divide:["$total_users",db.osm.distinct("created.user").length]}
...       , 100]} } },
... {$sort:{_id:1}},
... {$limit:10}
... ])
{ "_id" : 1, "total_users" : 161, "% user" : 17.27467811158798 }
{ "_id" : 2, "total_users" : 83, "% user" : 8.905579399141631 }
{ "_id" : 3, "total_users" : 61, "% user" : 6.545064377682404 }
{ "_id" : 4, "total_users" : 53, "% user" : 5.686695278969957 }
{ "_id" : 5, "total_users" : 54, "% user" : 5.793991416309012 }
{ "_id" : 6, "total_users" : 24, "% user" : 2.575107296137339 }
{ "_id" : 7, "total_users" : 24, "% user" : 2.575107296137339 }
{ "_id" : 8, "total_users" : 22, "% user" : 2.3605150214592276 }
{ "_id" : 9, "total_users" : 21, "% user" : 2.2532188841201717 }
{ "_id" : 10, "total_users" : 15, "% user" : 1.6094420600858368 }
>
```

Additional Ideas

Open Street Map is a free project done by volunteers; anybody can enter anything he or she wishes.[13] There is a guideline in place that would increase the quality of the map, if only all volunteers follow and agree on this code of conduct. Unfortunately, this is not the case as I realized when parsing the downloaded map for Quezon City. I saw that format uniformity is not followed which makes data parsing more difficult. Thus, in my opinion, it would be great if there is some sort of validation in place for "k" and "v" attribute in order to eliminate invalid characters in "k" attribute and to remove the empty value for "v". There is not much can be done for correctness, but having these two validations will help improve the quality of the map being submitted.

It would be even better if there is an array of valid "k" values, rejecting map submission if "k" does not adhere to any of the valid values. There should also be a way for volunteers to submit new values for "k", allowing for the k's valid values to grow. However, this suggestion involves some management and coordination, a waiting time needed for OSM management to respond and add in the new "k" value.

Additional Data Exploration using MongoDB Queries

Top 5 amenities

```
> db.osm.aggregate([
... {$match:{"amenity":{$exists:1} } },
... {$group:{_id:"$amenity", count:{$sum:1} }} ,
... {$sort:{count:-1}},
... {$limit:5}
... ])
{ "_id" : "school", "count" : 1117 }
{ "_id" : "restaurant", "count" : 1010 }
{ "_id" : "bank", "count" : 931 }
{ "_id" : "place_of_worship", "count" : 775 }
{ "_id" : "parking", "count" : 709 }
>
>
```

[13] https://wiki.openstreetmap.org/wiki/Good_practice

Top 5 fast food chain

```
> db.osm.aggregate([
... {$match:{"amenity":"fast_food" } },
... {$group:{_id:"$name", count:{$sum:1} }} ,
... {$sort:{count:-1}},
... {$limit:5}
... ])
{ "_id" : "Jollibee", "count" : 115 }
{ "_id" : "McDonald's", "count" : 75 }
{ "_id" : "Chowking", "count" : 41 }
{ "_id" : "KFC", "count" : 39 }
{ "_id" : "Mang Inasal", "count" : 29 }
>
```

Jollibee and McDonalds are the top two competing Hamburger chain in the Philippines. Jollibee, being a Filipino store, tops McDonalds by 35%.

5 Burger Food Chain near 15th Avenue

```
> db.osm.aggregate([
... {$geoNear:{near:db.osm.findOne({"address.street":"15th Avenue"}).pos,
... distanceField:"dist.calculated", distanceMultiplier: 6371, minDistance:2,
... query:{"amenity":"fast_food", "cuisine":"burger", "name":{$exists:1},
...        "address.street":{$exists:1} } } },
... {$project:{_id:0, name:"$name",
...    distanceInKm:"$dist.calculated", street:"$address.street"}},
... {$limit:5}
... ])
{ "name" : "McDonald's", "distanceInKm" : 38.75992641562097, "street" : "Aurora
Boulevard" }
{ "name" : "McDonald's", "distanceInKm" : 43.76504164699948, "street" : "Aurora
Boulevard" }
{ "name" : "McDonald's", "distanceInKm" : 49.64542672465421, "street" : "Aurora
Boulevard" }
{ "name" : "Jollibee", "distanceInKm" : 76.83857683081676, "street" : "Epifanio
de los Santos Avenue" }
{ "name" : "McDonald's", "distanceInKm" : 107.74504123941388, "street" : "Katipu
nan Avenue" }
>
```

The query above requires the existence of a 2D index. To create 2D index in Mongo DB execute:

db.osm.ensureIndex({pos:"2d"})

Conclusion

Data that comes from human entries are prone to errors; consequently, data quality is greatly affected. This is clearly shown in the Quezon City map downloaded from OSM. Hence, auditing and cleaning of data are important steps before a viable analysis can be done. I believe that I was able to apply the data cleaning techniques learned in this course and with the use of MongoDB queries, I was able to gather as much statistics as I can. However, collecting map data would be more efficient if a validation process is added prior to submission of map data. Such validations may include checking the format of tag's k attribute, as well as ensuring that "k" and "v" do not have empty values. It would be even better if there is control on what values can be added to "k" attribute. Having these validations in place would greatly alleviate dirty data, lessen the time needed for data cleaning and improve the quality of maps submitted to Open Street Map.