Data Wrangling with Mongo DB
Open Street Map Project2 Report

Name - Punit Mehndiratta
Map Location - Philadelphia PA , United States

Files location on github -
https://github.com/pmehndi11/Data_wrangling_MongoDB_Project2

1. Problems encountered in the map

Initially i created a subset (10% of data) from the philadelphia.osm file and then audit and analyze this particular file set by using https://github.com/pmehndi11/Data_wrangling_MongoDB_Project2/blob/master/philadelphia_subset_audit_fix_wrangle.py

I encountered the following issues -
- Multiple abbreviated street names (ex- St, st., street, ST)
- Incorrect data in street names (ex- atreet instead of street)
- Consistency issues (ex- street names contains full addresses)
- Data outside of Philadelphia
- Incomplete data (ex- None values for many tags)

The problems are discussed in details below , much of these issues have been fixed for full data set by using

philadelphia-fullsetdata-audit_fix_wrangle.py

Multiple abbreviated street names

During the audit of data, before loading into MongoDB , i found out that the street names have multiple abbreviations for ex- Road is represented as rd, rd. , Rd, ROAD and road.
I have fixed these street abbreviations.

Incorrect data in street names

In many cases , the data in street names was found to be incorrect (ex- Street was written in many cases as Sstreet or Sreet or Areet ro Sts), in some cases the street was incorrectly mentioned (for ex- Instead of 'Spruce Street' , the addr:street contained 'Spruce'. Instead of

‘Spring Garden Street’ the addr:street mentioned ‘Spring Garden’). As a person familar with Philadelphia region , i corrected such incorrect and incomplete data using the mapping part of the update_name function.

Consistency issues

In many cases, instead of street names in the addr:street tag values , the complete address was mentioned . (Example - instead of ‘Mano Avenue’ as street name , the tag value mentioned ‘200 Manor Ave. Langhorne, PA 19047’ , the complete address of the node.
In other cases , the street name also mentioned the City name alongwith the street name.
(Ex - Instead of South Street , the tag value contained ‘South Street, Philadelphia, PA’)
I have corrected such issues within the Update_name function as well as Shape_element function for the full set of data before converting the file to json format.

Data Outside of Philadelphia

During the audit even before the data was loaded into MongoDB , it became clear based on Street names and City names that the data in the fullset file contains areas outside of Philadelphia, PA.

After the data was audited, corrected , transformed to JSON format and loaded into MongoDB , i used MongoDB queries and aggregate the data based on postal codes and name of cities to figure out what addresses the data represented.

The code is represented in the file below.

Philadelphia-dataset_postal_codes.py

or on the mongo.exe console

- Sort Postal codes by count , descending

```
>db.philly.aggregate[{"$match" : {"address.postcode" : {"$exists" : 1}}},
                    {"$group" : {"_id" : "$address.postcode",
                              "count" : {"$sum" : 1}}},
                    {"$sort" : {"count" : -1}},
                    {"$limit" : 100}]
```

Top 7 postal codes are represented below-

```
{u'_id': u'19130', u'count': 362}
{u'_id': u'19128', u'count': 108}
{u'_id': u'19104', u'count': 43}
{u'_id': u'08096', u'count': 37}
{u'_id': u'19355', u'count': 27}
{u'_id': u'19103', u'count': 25}
{u'_id': u'08071', u'count': 23}
```

Based on information about Philadelphia ,the city postal codes are from 19102 till 19154.

The postal codes in our philly collections (obtained from our philadelphia.osm file) ranges from 08021 to 08088 (South NJ area near to Philadelphia) to Philadelphia City and Philadelphia county to East PA near Philadelphia (postal code 18901,18929,18914), to Wst PA near Philadelphia (Montgomery County area) to Wilmington , DE area (postal codes 19803, 19810).

This shows that the philadelphia.osm file contains data not only about Philadelphia metropolitan area , but the larger Philadelphia-Reading-Camden, PA-NJ-DE-MD area
known as ‘**Greater Philadelphia Region**’ which spans around 40-50 miles outside of Philadelphia city and into 4 states. (PA-NJ-DE-MD)

Incomplete and missing data

Many of the tag have value missing or mentioned as ‘None’ (for ex- there are many ‘fast food’ restaurants with name value as ‘None’ . Similarly , many places of worship have religion as ‘None’.

2. Data Overview

I have analyzed the open street map of philadelphia metro city dataset.
Major Steps taken

- To audit, map, fix street names, wrangle philadelphia.osm (full set 509 MB file) into data model in json. code, run the file below using Python2 notebook.

https://github.com/pmehndi11/Data_wrangling_MongoDB_Project2/blob/master/philadelphia-fullsetdata-audit_fix_wrangle.py

This will generate the output file philadelphia.osm.json

- Now start the mongo db

```
C:\MongoDB\Server\3.0\bin\mongod.exe
```

- Next Import the json file

```
C:\MongoDB\Server\3.0\bin\mongoimport --db philly --collection philly --file
philadelphia.osm.json
```

```
Imported  2449819 documents
```

- Now start mongo:

```
C:\ MongoDB\Server\3.0\bin\mongo
> use philly
switched to db philly
```

Below we describe some basic statistics about our data set alongwith MongoDB queries used to gather them.

- **Original File sizes**

Original size of philadelphia .osm file -   501,130 KB
The size of json file generated for the same - 549,687 KB

- **Size of collection in MongoDB**

```
> db.philly.dataSize()
768344144
```

(about 768 MB size)

- **Number of documents in philly collection**

```
> db.philly.find().count()
2449819
```

- **Number of unique users**

1018 user have edited the greater philadelphia region map

```
> db.philly.distinct("created.user").length
1018
```

- **Top 10 contributing User**

As we can see below the Top 10 contributing users in the descending order

```
>db.philly.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
{"$group":{"_id":"$count", "num_users":{"$sum":1}}}, {"$sort":{"_id":1}}, {"$limit":10}])
```

```
{u'_id': u'dchiles', u'count': 805506}
{u'_id': u'woodpeck_fixbot', u'count': 633879}
{u'_id': u'NJDataUploads', u'count': 307818}
{u'_id': u'crystalwalrein', u'count': 93254}
{u'_id': u'Louise Belcher', u'count': 61722}
{u'_id': u'choess', u'count': 60486}
{u'_id': u'bot-mode', u'count': 59407}
{u'_id': u'NE2', u'count': 42215}
{u'_id': u'ceyockey', u'count': 37702}
{u'_id': u'TIGERcnl', u'count': 35596}
```

The user 'woodpeck_fixbot' indicates some automated scripts run to fix the data for PA.
The user 'NJDataUploads' also indicates some automated scripts for maintenance of NJ data.

- **Number of ways and nodes**

There are 2279599 nodes and 169,211 ways in the greater philadelphia region map.

```
> db.philly.find({type: "node"}).length()
2279599
> db.philly.find({type: "way"}).length()
169211
```

3. Additional Ideas about the Dataset

- **Number of Restaurants in the region**

There are 697 Restaurants in the Philadelphia region

```
> db.philly.find({amenity: "restaurant"}).count()
697
```

- **Top 10 cuisines in the region**

On analyzing the top cuisines are as below -
We find out that the top cuisines are Pizza, Italian, sandwich, chinese and american.

Also the fact that 'None' is top cuisine states that the data is incomplete whereby cuisine for Restaurants are not mentioned for around half of the restaurants (375 out of total 697 Restaurants).

```
> db.philly.aggregate([{"$match":{"amenity":{"$exists":1}, "amenity":"restaurant"}},
{"$group":{"_id":"$cuisine", "count":{"$sum":1}}},        {"$sort":{"count":-1}}, {"$limit":10}])

{u'_id': None, u'count': 375}
{u'_id': u'pizza', u'count': 64}
{u'_id': u'italian', u'count': 33}
{u'_id': u'sandwich', u'count': 33}
{u'_id': u'chinese', u'count': 27}
{u'_id': u'american', u'count': 26}
{u'_id': u'mexican', u'count': 15}
{u'_id': u'burger', u'count': 13}
{u'_id': u'indian', u'count': 10}
{u'_id': u'asian', u'count': 10}
```

- **Top 10 amenities in the region**

Interestingly the top amenity in our 'Greater Philadelphia Region' map is

‘Parking’(2986) followed by ‘Schools’(2168) followed by ‘Place of worship ,
‘Restaurant’(697) and ‘Fire Station’(505). Interestingly the number 10 amenity is
‘University’(177)

Philadelphia dataset (philly collection)top 10 amenities Mongo DB queries-
https://github.com/pmehndi11/Data_wrangling_MongoDB_Project2/blob/master/philly_top10_amenities.py

Output is

```
{u'_id': u'parking', u'count': 2986}
{u'_id': u'school', u'count': 2168}
{u'_id': u'place_of_worship', u'count': 731}
{u'_id': u'restaurant', u'count': 697}
{u'_id': u'fire_station', u'count': 505}
{u'_id': u'fast_food', u'count': 303}
{u'_id': u'post_office', u'count': 239}
{u'_id': u'bank', u'count': 209}
{u'_id': u'grave_yard', u'count': 197}
{u'_id': u'university', u'count': 177}
```

- **Number of Starbucks in the Philadelphia region**

There are 22 Starbucks cafe in the map area

```
> db.philly.find({amenity: "cafe",name: "Starbucks"}).count()
22
```

- **Biggest Religions in the Philadelphia region**

Here are the top 3 religions in the Philadelphia area

```
> db.philly.aggregate([{"$match":{"amenity":{"$exists":1}, "amenity":"place_of_worship"}},

{"$group":{"_id":"$religion", "count":{"$sum":1}}},

{"$sort":{"count": -1}}, {"$limit":3}])


Christianity : 637
None :          54
```

Jewish   : 34

None figure shows that there is incomplete data for place of worship.
Though majority of place of worship are Christian churches.

- **Top 5 Fast food places in the Philadelphia Region**

No surprise here , there are 42 Mcdonalds in the region , followed by ‘None(21) , Wendys (20), Dunkin Donuts (15) , Burger King (13) and Subway (10)

21 Fast food places with name : None shows incomplete data in the map.

Philadelphia dataset (philly collection) top 5 fast food restaurants Mongo DB queries-
https://github.com/pmehndi11/Data_wrangling_MongoDB_Project2/blob/master/philly_top5_fastfood_restaurants.py

Output is as below -
```
{u'_id': u"McDonald's", u'count': 29}
{u'_id': None, u'count': 21}
{u'_id': u"Wendy's", u'count': 20}
{u'_id': u'Dunkin Donuts', u'count': 15}
{u'_id': u'McDonalds', u'count': 13}
{u'_id': u'Burger King', u'count': 13}
{u'_id': u'Subway', u'count': 10}
```

We have combined the count "McDonald's" and 'McDonalds’.

- **Top 10 Leisure types areas in Philadelphia Region**

Philadelphia Region does have high number of Leisure areas especially Playgrounds and Parks as shown by the result of our query below.

Philadelphia dataset (philly collection) top 10 leisure areas Mongo DB queries
https://github.com/pmehndi11/Data_wrangling_MongoDB_Project2/blob/master/philly_top10_leisure-areas.py

Output is as below-
```
{u'_id': u'pitch', u'count': 1132}
{u'_id': u'park', u'count': 766}
{u'_id': u'playground', u'count': 208}
```

```
{u'_id': u'golf_course', u'count': 107}
{u'_id': u'sports_centre', u'count': 76}
{u'_id': u'swimming_pool', u'count': 66}
{u'_id': u'track', u'count': 57}
{u'_id': u'nature_reserve', u'count': 26}
{u'_id': u'garden', u'count': 25}
{u'_id': u'picnic_table', u'count': 24}
```

The pitch and Playground both represent open play areas and can be counted together as count of Playgrounds.

• Conclusion and additional ideas about the data

It is clear that the data for Greater Philadelphia area is incomplete , have lot of errors and is inconsistent. I have fixed a lot of the issues regarding the street names for the project completion. All of this data and need to be fixed at the Open Street Map for Philadelphia region. Also as we can see from analysis many nodes have missing information (such as Name of Restaurants , fast food places, places of worship (having the tag values as 'None').

One way to improve the open street map data is

- Compare the data with another external source such as mapquest or google maps for the same region.
- Edit then Open street map data based on the compared accurate data.

Potential Challenges for such approach of implementation -

- Legal and Ethical issues -We can extract the data from Google map with Google Maps API in order to compare with Open street maps data for same region. But legally based on information on Google Maps https://developers.google.com/maps/, it seems like the data is no for fixing other maps , but for apps and websites usages by individual and businesses.
- Implementation challenges - Even in absence of any legal and ethical challenges, the data from Google map will then need to be compared with .osm data and the fixes are then need to be loaded back into open street map using JOSM. (since Id editor is not for fixing a large amount of data). This itself can be a large project if legally allowed.
- In such case. also there may be questions regarding which particular dataset is correct in particular case.(Google or OSM) Based on my analysis, open street maps does have better reputation for rural area maps.

The above challenges points to reason why engaging as many users in developing the maps around their own neighbourhood or areas known to them seems a better way to improve the maps whether its open street map via JOSM / id editor OR google maps via google map maker.

4. Additional issues encountered during the project implementation

- **Memory and timeout issues with full data set parsing and conversion**

The Biggest issue encountered during the project implementation was related to Memory errors and timeouts (Kernel restarts of Ipython notebook) when the full dataset was used. Such issues were resolved by improving the code quality (removing unnecessary steps and print statements) and by using incremental parsing of XML tree by clearing out unneeded parts of the tree after processing is finished for them.