OpenStreetMap Project
Data Wrangling with MongoDB

Yeqing Zhang

Map Area: Manchester, England, United Kingdom
Open Street Map: https://www.openstreetmap.org/relation/146656
Download link: https://s3.amazonaws.com/metro-extracts.mapzen.com/manchester_england.osm.bz2

1. Problems Encountered in the Map

After a brief checking through the map data, there are a few major problems that I noticed in the map.

- Non-standardized postcodes: “CW84DA”, "M22-9QY"
- Inconsistent sources: "NPE;OS OpenData StreetView", "NPE;OS_OpenData_StreetView"

Postcode

Most of the postcodes in the dataset are well-formatted in a correct way as guided on UK Postcode Formatting. But there are a few inconsistent or ill-formatted postcode that does not follow the guidance. Below is the list of the major problems.

Lower cases
“cw84db"

No whitespace between the outcode and incode
“CW84DA"

Leading or trailing white spaces
" M41 6NA"

invalid characters
"M22-9QY"

After standardizing the postcodes,

```
>db.manchester_england.aggregate({$match:{'address.postcode':{$exists:true}}},{$gr
oup:{_id:'$address.postcode',count:{$sum:1}}},{$sort:{count:-1}}, {$limit:10})
{ "_id" : "OL9 0NT", "count" : 211 }
{ "_id" : "OL9 0NS", "count" : 193 }
{ "_id" : "SK9 5BT", "count" : 68 }
{ "_id" : "SK6 4AN", "count" : 56 }
{ "_id" : "SK23 7NS", "count" : 52 }
{ "_id" : "OL9 0NX", "count" : 47 }
{ "_id" : "CW9 5AY", "count" : 42 }
{ "_id" : "SK6 4PJ", "count" : 41 }
```

```
{ "_id" : "SK6 4PP", "count" : 39 }
{ "_id" : "SK6 4EG", "count" : 38 }
```

It is noticed that out of the top 10 postcodes, none starts with M, which means they are not in the Manchester city area. This means the dataset actually contains the data from the nearby areas, i.e. Greater Manchester. It can be verified by performing a top 10 city query against the dataset.

```
>db.manchester_england.aggregate({$match:{'address.city':{$exists:true}}},{$group:{
_id:'$address.city',count:{$sum:1}}},{$sort:{count:-1}}, {$limit:10})
{ "_id" : "Romiley", "count" : 768 }
{ "_id" : "Oldham", "count" : 656 }
{ "_id" : "Manchester", "count" : 322 }
{ "_id" : "Northwich", "count" : 274 }
{ "_id" : "Buxworth", "count" : 85 }
{ "_id" : "Macclesfield", "count" : 77 }
{ "_id" : "Pickmere", "count" : 52 }
{ "_id" : "Warrington", "count" : 36 }
{ "_id" : "Heald Green", "count" : 36 }
{ "_id" : "Salford", "count" : 35 }
```

It shows that the top 2 cities are “Romiley” and “Oldham”, which are two towns near Manchester. As explained by Wikipedia (https://en.wikipedia.org/wiki/Greater_Manchester), the top 1 city Romiley is within Stockport, which is the second region of Greater Manchster; 2nd city Oldham itself is the No. 4 main region.

Source

The sources of the dataset are inconsistent. The sources are named in a strikingly random and casual fashion, resulting in unicity and inconsistencies. For example, from the source of Bing search engine, a query with regex case insensitive match against “bing” can result in 183 different namings.

```
>db.manchester_england.aggregate({$match:{source:/bing/i}},{$group:{_id:'$source',
count:{$sum:1}}},{$group:{_id:null,count:{$sum:1}}})
{ "_id" : null, "count" : 183 }
```

Now perform a query to list 10 examples.

```
>db.manchester_england.aggregate({$match:{source:/bing/i}},{$group:{_id:'$source',
count:{$sum:1}}},{$sort:{_id:1}},{$limit:10})
{ "_id" : "BING", "count" : 2 }
{ "_id" : "BING & NPE", "count" : 1 }
{ "_id" : "Bing", "count" : 14784 }
{ "_id" : "Bing  /  survey", "count" : 1 }
{ "_id" : "Bing & OS Open Street Map", "count" : 1 }
{ "_id" : "Bing & foot survey", "count" : 1 }
```

```
{ "_id" : "Bing & ground survey", "count" : 44 }
{ "_id" : "Bing & survey", "count" : 2 }
{ "_id" : "Bing (portals and airvents); knowledge of curved end segments", "count" : 1
}
{ "_id" : "Bing /   OS hISTORIC MAPS", "count" : 1 }
```

In summary, the problems can be categorized as…

- **Case sensitivity issue**: “Bing”, “BING”
- **Over-detailed description**: "Bing (portals and airvents); knowledge of curved end segments"
- **Overlapping with other sources:** “BING & NPE”, "Bing+NAPTAN"
- **Nested structure:** { "source" : { "geometry" : "Bing", "name" : "NaPTAN" } }

The solution is to predefine a set of source categories and for each source name, performs a regex match (case insensitive) against each predefined source category. If matched, store the matched source category into an array, i.e. tagging.

After the standardization of the source names, an aggregation query on the source category can be performed.

```
>db.manchester_england.aggregate({$unwind:'$source_category'},
{$group:{_id:{source_category:'$source_category'}, count:{$sum:1}}},
{$sort:{count:-1}}).pretty()
{ "_id" : { "source_category" : "Bing" }, "count" : 23881 }
{ "_id" : { "source_category" : "NaPTAN" }, "count" : 13407 }
{ "_id" : { "source_category" : "OS Open Data" }, "count" : 13281 }
{ "_id" : { "source_category" : "Survey" }, "count" : 6185 }
{ "_id" : { "source_category" : "GPS" }, "count" : 3627 }
{ "_id" : { "source_category" : "Yahoo" }, "count" : 2895 }
{ "_id" : { "source_category" : "NPE" }, "count" : 1621 }
{ "_id" : { "source_category" : "Landsat" }, "count" : 607 }
{ "_id" : { "source_category" : "Local" }, "count" : 504 }
{ "_id" : { "source_category" : "Website" }, "count" : 83 }
{ "_id" : { "source_category" : "Campus" }, "count" : 78 }
{ "_id" : { "source_category" : "PGS" }, "count" : 35 }
{ "_id" : { "source_category" : "Observation" }, "count" : 32 }
{ "_id" : { "source_category" : "NLS" }, "count" : 30 }
{ "_id" : { "source_category" : "Wikipedia" }, "count" : 9 }
```

It shows that the top three sources are Bing, NaPTAN, OS Open Data.

2. Data Overview

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

File Sizes

manchester_england.osm 268MB
manchester_england.osm.json 305MB

Number of documents

```
> db.manchester_england.find().count()
1433928
```

Number of nodes

```
> db.manchester_england.find({"type":"node"}).count()
1254464
```

Number of ways

```
> db.manchester_england.find({"type":"way"}).count()
179464
```

Number of unique users

```
> db.manchester_england.distinct("created.user").length
1694
```

Top 3 contributing user

```
> db.manchester_england.aggregate({$group:{_id:'$created.user',
count:{$sum:1}}}, {$sort:{count:-1}}, {$limit:3})
{ "_id" : "RichardB", "count" : 258835 }
{ "_id" : "Steeley", "count" : 85614 }
{ "_id" : "RobChafer", "count" : 46620 }
```

Number of users appearing only once (having 1 post)

```
> db.manchester_england.aggregate({$group:{_id:'$created.user',
count:{$sum:1}}}, {$sort:{count:-1}}, {$match:{count:1}}, {$group:{_id:null,
count:{$sum:1}}})
{ "_id" : null, "count" : 301 }
```

Top 3 amenities

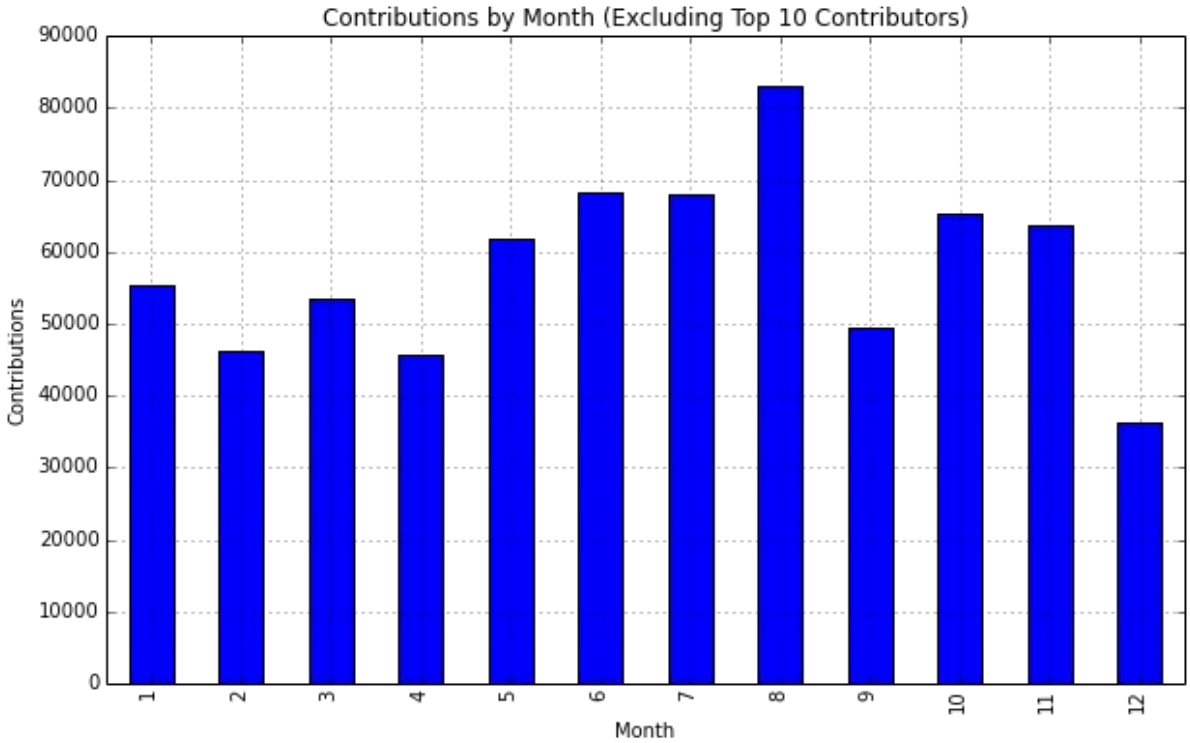
```
> db.manchester_england.aggregate({$match:{amenity:{$exists:true}}},
{$group:{_id:'$amenity', count:{$sum:1}}}, {$sort:{count:-1}}, {$limit:3})
{ "_id" : "parking", "count" : 3526 }
```

{ "_id" : "pub", "count" : 1518 }
{ "_id" : "school", "count" : 1007 }

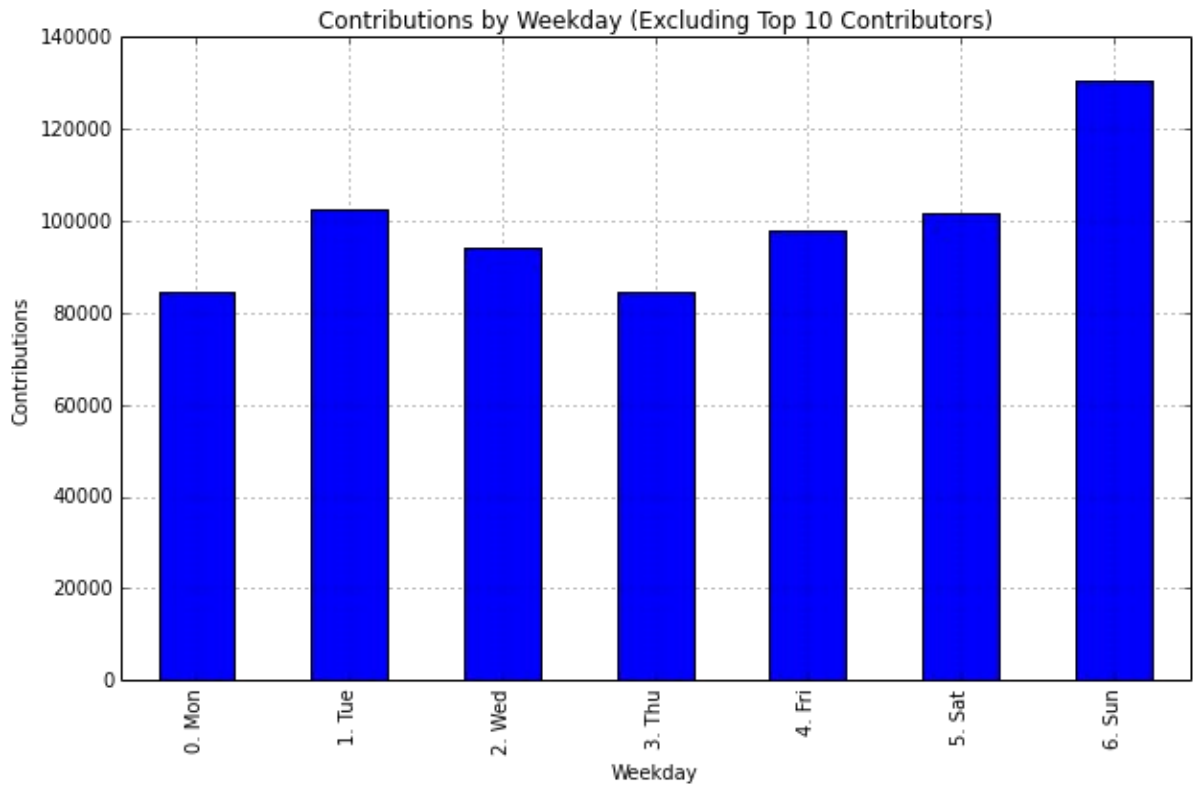
3. Additional Ideas

Seasonality suggestion

It seems that the number of contributions follow certain patterns (excluding top 10 contributors that contains 50% of total). By looking at the monthly contributions, it appears that the holiday months (April Easter, September, December Xmas) have lower contributions than other months. Also, by looking at the contributions by weekday data, it is noticed that most of the contributions come from Sunday. Perhaps the contributors prefer to travel and work on the mapping as a hobby over the weekends. In the summer holidays (August) when the weather is notably better than other seasons, the contributors tend to travel around and map the places where they travelled.



The contributions by month. April, September and December have lower contributions whilst August has the highest contributions.



Contributions by weekday. Sunday has the highest contributions.

Node Info

It is noticed that a notable amount of nodes can be considered as “basic nodes”, which only contains basic information about this node (id, pos, created, type). Nodes that contains any other extra info are not considered as basic nodes.

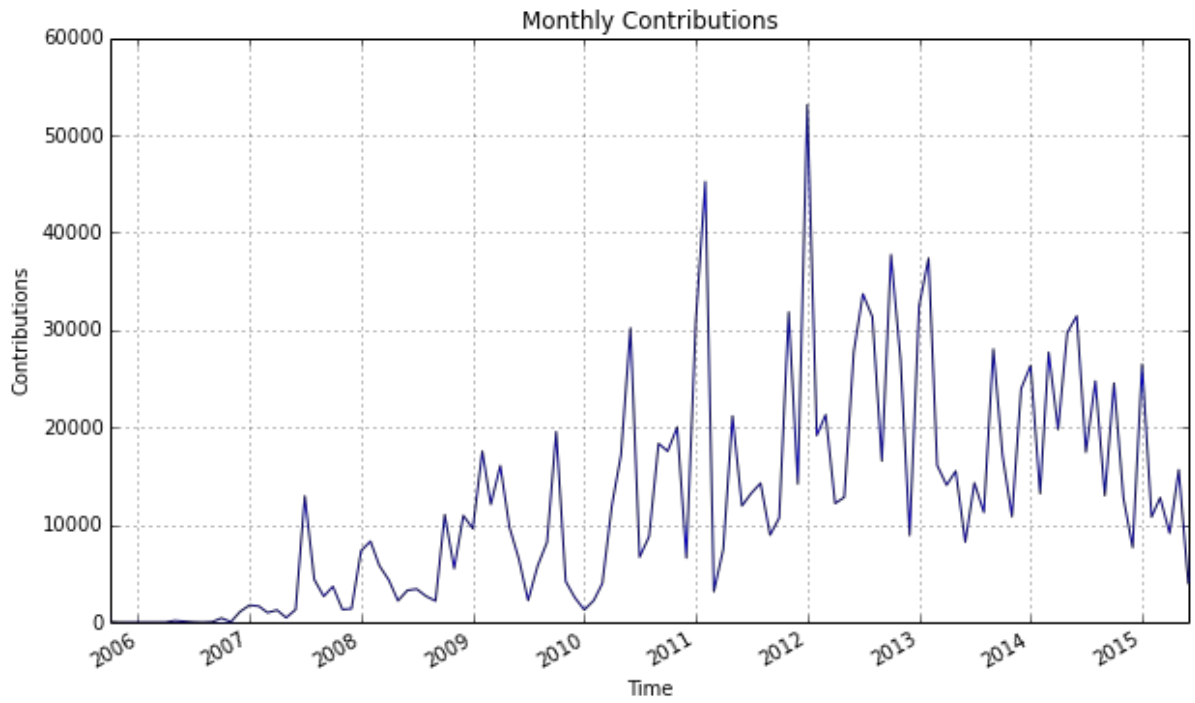
Below query performs a check on how many basic nodes the dataset contains.

```
>db.manchester_england.aggregate({$match:{type:'node'}},{$group:{_id:'$basic_nod
e',count:{$sum:1}}})
{ "_id" : false, "count" : 90364 }
{ "_id" : true, "count" : 1164100 }
```

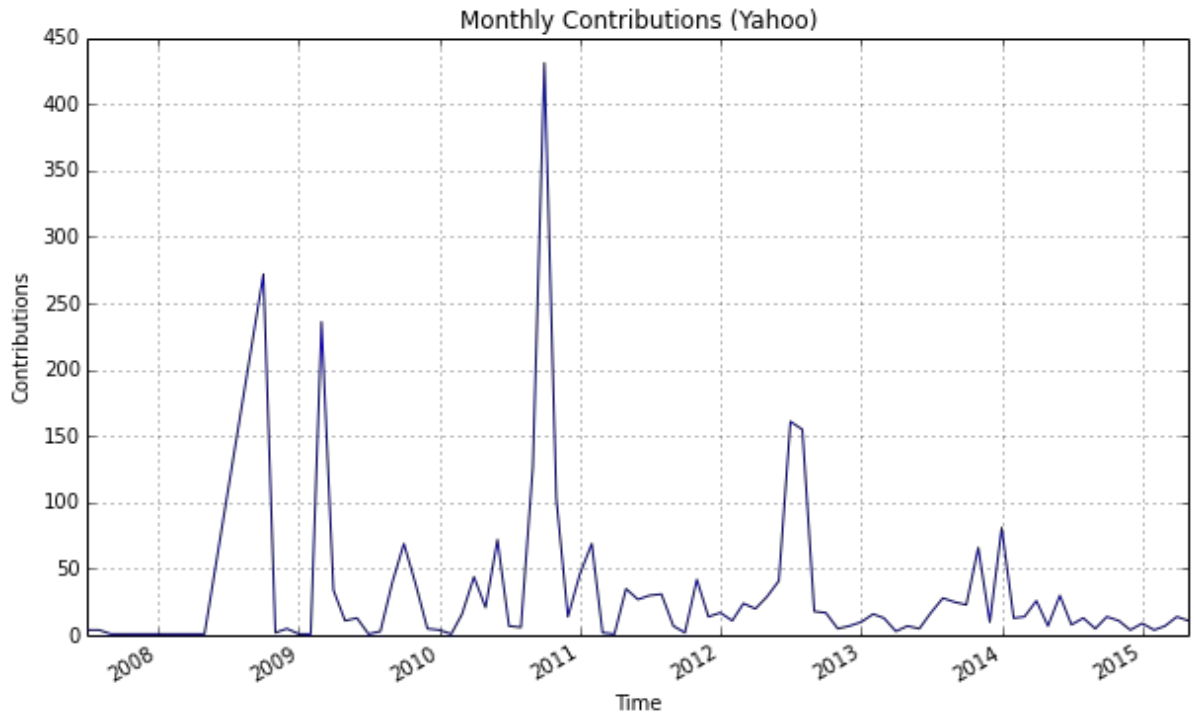
It is true that the majority of the nodes (93%) are basic nodes.

Contributions Overtime

Starting from 2006, the contributions overtime follows an increasing trend and peaked in 2012. Then the contributions follows an decreasing trend. It seems that the Great Manchester area on the OpenStreetMap is being improved.



Monthly contributions. The contributions overtime follows an increasing trend and peaked in 2012 and follows a declining trend after then until recent month. This is contributed by the integration of Bing Maps into OpenStreetMap which was introduced in November 2010 (http://wiki.openstreetmap.org/wiki/Bing).



Monthly Contributions (Bing). During 2012, it is noticed that the contributions from Bing started in 2011 reached around 1000 per month in 2012 and then has been declining since 2013.

Conclusion

After this review of the data, it is evident that the Great Manchester area on OSM is with minor inconsistencies on the postcodes and major issues of sources. The data, I believe, is well cleaned for the purpose of this exercise. It is interesting to know that the top source of the data comes from Bing, which is not surprising as contributors utilized most of the Bing Maps Services (Bing Aerial Imagery) since the end of 2010. I believe with an appropriate setup of Bing Maps API into OpenStreetMap, the data would be likely to integrate more cleaned data into OSM.