OpenStreetMap Sample Project
Data Wrangling with MongoDB

Vijay Pazheparampil

Map Area: Bengaluru, Karnataka, India

https://mapzen.com/data/metro-extracts/metro/bengaluru_india/

1. Problems Encountered in the Map

Postal Code

•It was found that Non numeric postal codes were present in the address. So these documents were ignored.

•Space separated codes were converted to correct 6 digit numeric numbers
Eg : *560 090* was converted to 560090

•Documents having Postal Codes with length less than 6 were also ignored
Eg : 79

Street Address

•At the time of processing the data, it was found that the city name *bangalore* was present at the end of many street addresses. So following regular expression was used to find and remove the presence of *bangalore* in the street address. Trailing spaces and commas were also removed.

```
re.compile('[,\s]*bangalore[,\s]*$')
```

•Abbreviations were used in the street addresses. So the following transformation mapping was used to find and replace the presence of such words.

```
{ 'Rd.' : 'Road', 'Rd' :'Road'}
```

2. Data Overview

The file as downloaded from Mapzen (ex_dAvZyUpXvZKSvGG5bDdec1YK9DCjC.osm) size was 652 MB.
After processing the file as a JSON file, making the above corrections and taking into account 'way' and 'node' tags, the size turned out to be 759 MB
Later, I have created a MongoDB Atlas cluster to upload my map data (.json). Howver there was a restriction on the maximum file size I could upload, which was 512 MB.
The upload was done with the help of the following command from Mongo Shell

```
mongoimport /uri "mongodb://admin:<PASSWORD>@cluster0-shard-00-00-cdlst.mongodb.net:27017,cluster0-shard-00-01-cdlst.mongodb.net:27017,cluster0-shard-00-02-cdlst.mongodb.net:27017/test?ssl=true&replicaSet=Cluster0-shard-0&authSource=admin" /collection bangaloreosm /drop /file
```

C:\Users\Vijay\Documents\DataAnalyst_Udacity\Assignments\MongoDB\Mapzen\ex_dAvZyUpXvZKSvGG5bDdec1YK9DCjC.osm.json

Collection Name : bangaloreosm
Database Name : test

The following analysis was done woth MongoDB Compass:

- Total number of Documents :

```
db.bangaloreosm.count()
2295000
```

- Total number of nodes :

```
db.bangaloreosm.find({'type':'node'}).count()
2294983
```

- Total number of way :

```
db.bangaloreosm.find({'type':'way'}).count()
0
(This might be due to the restriction on the MongoDB Atlas cloud size which was 512 MB)
```

- Total number documents with type not as node or way:

```
db.bangaloreosm.find({$or :[{'type':{$ne :'node'}, 'type' :{$ne : 'way'}}]}).count()
17
```

- Total number of uniques users :

```
db.bangaloreosm.distinct('created.user').length
1330
```

- Top contributing user :

```
pipeline = [  {'$group':{'_id':'$created.user','adds':{'$sum':1}}},
              {'$sort':{'adds':-1}},
              {'$limit':1} ]
db.bangaloreosm.aggregate(pipeline)
{ "_id" : "saikumar", "adds" : 90597 }
```

- Total users contributing to only one document :

```
pipeline=[    {'$group':{'_id':'$created.user','adds':{'$sum':1}}},
              {'$match':{'adds':1}},
              {'$group':{'_id':null,count:{'$sum':1}}} ]
db.bangaloreosm.aggregate(pipeline)
{ "_id" : null, "count" : 333 }

(So 25% of document is contributed by only one contributor)
(333 * 100)/1330 = 25%
```

- Top 10 contributing users :

```
pipeline = [  {'$group':{'_id':'$created.user','adds':{'$sum':1}}},
              {'$sort':{'adds':-1}},
              {'$limit':10} ]
db.bangaloreosm.aggregate(pipeline)
{ "_id" : "saikumar", "adds" : 90597 }
{ "_id" : "akhilsai", "adds" : 85092 }
{ "_id" : "jasvinderkaur", "adds" : 72355 }
{ "_id" : "shekarn", "adds" : 67311 }
{ "_id" : "Mohanrrb", "adds" : 63629 }
{ "_id" : "premkumar", "adds" : 62932 }
{ "_id" : "sdivya", "adds" : 58241 }
{ "_id" : "harishk", "adds" : 54051 }
{ "_id" : "sampath reddy", "adds" : 53363 }
{ "_id" : "vamshiN", "adds" : 52906 }
```

- Total document contrinuted by top 10 contributing users :

```
pipeline = [  {'$group':{'_id':'$created.user','adds':{'$sum':1}}},
              {'$sort':{'adds':-1}},
              {'$limit':10},
              {'$group':{'_id':null,'sum':{'$sum':'$adds'}}} ]
db.bangaloreosm.aggregate(pipeline)

{ "_id" : null, "sum" : 660477 }
```

(So 28.7% of documents is contributed by top 10 contributors)
*(660477 * 100)/2295000 = 28.7%*

- Total schools count :

```
pipeline=[    {'$match':{'amenity':{'$in':['school']}}},
              {'$group':{'_id':'$amenity','count':{'$sum':1}}} ]

db.bangaloreosm.aggregate(pipeline)
{ "_id" : "school", "count" : 285 }
```

- Total bar/liqour shop count :

```
pipeline=[    {'$match':{'amenity':{'$in':['bar','liqour','liquor shop']}}},
              {'$group':{'_id':'$amenity','count':{'$sum':1}}} ]
db.bangaloreosm.aggregate(pipeline)
{ "_id" : "bar", "count" : 60 }
```

- Documents having name in English Language and name in local language (Kannada)

```
db.bangaloreosm.find({'name:kn':{'$exists':true},'name':{'$exists':true}}).count()
3649
```

3. Ideas to improve the dataset

Improvement of Postal Codes:

As we had issues with the prostal pin codes, one way to improve it could be validate it against a more official repository coming from the Postal department itself. The link (https://data.gov.in/catalog/all-india-pincode-directory) provides a csv file with the postal codes with the latitude and longitude which could be used for cross verification and improvde the dataset.

Problems: The latlong information is only for some postal codes. (But at least it could be means to cross verify)

Improvement of Street Names:

Why dont we have a wikipaedia like control to edit important features like street names and addresses. This will ensure that the map data is clean and have a controlled evolution. After getting this idea, I just checked on the link (http://wiki.openstreetmap.org/wiki/Research/Ideas) and came to know that open street map management is already going in this direction. Good to know that!

4. Conclusion

Bangalore metro map is a rich source of geogrpahic information that can be made available to people at large. Mongodb allows an easy way to query this data set using very simple 'SQL-like' query. Initially I thought that this would be a tough task as I was working with Mongodb for the first time. However after doing this exrcise, it is not as tough as it looks. Setting up Mongodb and running queries is really very simple.

However we will always need to ensure that the data source is reliable as well as of good quality to derive meaningful insights.

5. References

1) **https://mapzen.com/data/metro-extracts/**

2) **https://cloud.mongodb.com/**

3) **https://docs.mongodb.com/manual/reference/**

4) h**ttps://stackoverflow.com/**