OpenStreetMap Project ,

Data Wrangling with MongoDB

By - Kamal Lochan Panigrahi

Map Area: Kolkata , West Bengal , India

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

The map is of city Kolkata . I have been to this city many a times .So many places are there to visit .Finally I got an opportunity to contribute to its improvement in openstreetmap.org.

1. Problems encountered :

- Over abbreviated and unusual Street names
- Inconsistent city names
- Incorrect postal codes

2. Data Overview

3. Additional ideas

1. Problems encountered in the map :

After downloading the osm file I sliced out a small sample of it and started parsing it .I found two main problems in the data set .

Over-abbreviated Street names :

By iterating through the dataset by using the method –

I found most of the street names were abbreviated e.g

Karbala Tank Ln.

Scott Ave.

Pathan St.

Then I used the update function to clean these names :

```
def update_name(name, mapping):
    if '(' in name:
            name = name.split('(')[0]
            name = name.strip()
    m = street_type_re.search(name)
    if m:
            if m.group() in mapping.keys():
            name = re.sub(m.group(), mapping[m.group()], name)
```

Finally I got the names as

Karbala Tank Lane

Scott Avenue

Pathan Street

Unusual street names :

Again when I iterated through the elements I found some of the names were mis spelt and some had extra brackets next to the names e.g

D.r A.k paul raod

I used the update name function and changed the inconsistent names to proper format e.g

D.r A.k paul road

Diamond Harbour Road etc.

Inconsistent City names :

During auditing I also found that cities were represented incorrectly i.e

K : addr : city v: Salt lake (Bidhan Nagar)

I made a correction by approaching a correct format i.e

K : City v: Salt lake

Incorrect Postal codes:

I found most of the postal codes were written incorrectly .eg

“700 027” ,

“700 095”

In data.py I made a cleaning approach to these postal codes. After cleaning I

Represented the postal codes as

700027

700095

Finally I put the nodes into a proper dictionary format using shape_element( ) .Then stored all the dictionaries into a json file so as to import it into mongodb for further analysis.

2. Data Overview :

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

File sizes

```
kolkata.osm ......... 81 MB
kolkata.osm.json .... 126 MB
```

# Number of documents

```
>db.kolkata.find().count()

448161
```

# Number of nodes

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

400906
```

# Number of ways

```
>db.kolkata.find({"type":"way"}).count()
47254
```

# Number of unique users

```
> db.kolkata.distinct({"created.user"}).length
146
```

# Top 1 contributing user

```
> db.kolkata.aggregate([{"$group":{"_id":"$create.user",
"count":{"$sum":1}}}, {"$sort":{"count":1}},
{"$limit":1}])

{"_id" : "Rondon237", "count" : 162603 }
```

3. Additional Ideas :

Contributor statistics :

The contribution of users looks skewed possibly due to automated vs manual map editing .Before giving some statistics let me put the initial figures i.e

Total no of documents :448161

No of unique users contributing :146

Top 10 contributers :

```
db.kolkata.aggregate ( [ { $group : { '_id': '$created.user', 'count':{ $sum : 1 } } },

        { $sort : { 'count' : -1 } },

                { $limit : 10 } ] )


                { "_id" : "Rondon237", "count" : 162603 }

                { "_id" : "sakthivel", "count" : 90623 }

                { "_id" : "maxsaurav", "count" : 76599 }

                { "_id" : "baigan", "count" : 16650 }

                { "_id" : "dmgroom_coastlines", "count" : 16323 }

                { "_id" : "sujandeb", "count" : 15774 }

                { "_id" : "iambibhas", "count" : 11198 }

                { "_id" : "Japa", "count" : 9505 }
                { "_id" : "Oberaffe", "count" : 9014 }
                { "_id" : "katpatuka", "count" : 6852 }
```

Contribution to the dataset by top user : 36.3%

Contribution to the dataset by top 2 user : 56.5%

Contribution to the dataset by top 10 user : 92.6%

As, we can see from the above trend only 10 out of 146 users contribute to around 92.6% of the entire data set . This shows that not many users are interested in supplying data for the OSM .

We can encourage more users to contribute to OSM project by :

-> giving them credit for adding the data by adding some points to their account .

-> maintaining a leader's board .

-> if they are rewarded for their contribution .

-> asking them to form groups and contribute to the OSM project , for the improvement of their city or state .

Additional Exploration:

Exploring top 10 amenities:

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

          { "_id" : "school", "count" : 110 }
          { "_id" : "hospital", "count" : 74 }
          { "_id" : "college", "count" : 57 }
          { "_id" : "fuel", "count" : 35 }
          { "_id" : "restaurant", "count" : 32 }
          { "_id" : "atm", "count" : 28 }
          { "_id" : "cinema", "count" : 28 }
          { "_id" : "bank", "count" : 25 }
          { "_id" : "place_of_worship", "count" : 24 }
          { "_id" : "university", "count" : 22 }
```

Most popular religion:

```
>db.kolkata.aggregate([{$match:{"amenity":{$exists:1},
     "amenity":"place_of_worship"}},{$group:{"_id":$religion,
          "count":{$sum:1}}},{$sort:{"count":-1}}, {$limit:1}])
```

```
     { "_id" : "hindu", "count" : 10 }
```

Top five sources :

```
>db.kolkata.aggregate([{$match : {'source' : {$exists:1 } } },
       { $group : { '_id' : '$source', 'count' : {$sum : 1 } } },
           { $sort : { 'count' : -1 } },
                { $limit : 5 }])
```

```
      {u'_id': u'PGS', u'count': 7674},
      {u'_id': u'Bing', u'count': 1853},
      {u'_id': u'Yahoo hires', u'count': 333},
      {u'_id': u'AND', u'count': 315},
      {u'_id': u'GPS', u'count': 124}
```

Benefits of Improving the OSM data :

-> As most of the people use smart phone these days , they can contribute to OSM using the ir GPS , which results in a more accurate data .

-> The data can also be used by government or private institutions to study about the geogr aphy of the place like the number of offices, buildings, schools, hospitals etc .

Anticipated problems in implementing the improvements :

-> Not many users will be ready to contribute to an opensource project like OSM by using their smart phone GPS , as they will have to do some work like moving from place to place for a more accurate result .

-> Some may design applications that automatically adds data to the OSM continuously , thus it always remains at the top of the leader board . This might be unfair for users who actually spend some time to contribute to the project

4. Conclusion

After this review of the data it’s obvious that the Kolkata area is incomplete, though I believe it has been well cleaned for the purposes of this exercise. It interests me to notice a fair amount of GPS data makes it into OpenStreetMap.org on account of users’ efforts, whether by scripting a map editing bot or otherwise. With a rough GPS data processor in place and working together with a more robust data processor similar to data.py I think it would be possible to input a great amount of cleaned data to OpenSreetMap.org.

People should be encouraged to put interest in contributing effort for this purpose .