OpenStreetMap Wrangling - Kolkata, WB, India

> *Ashish Sahu*

For this project, I’ll be analyzing the OpenStreetMaps data for city of **Kolkata, India** obtain from Metro Extracts. The datafile was approximately 686 MB. This is a sprawling urban centre with more than **4.5 million** people. I am interested to see some of the descriptions for the data in the city and how much data has been contributed to date.

I started by loading the Python library for parsing XML files incrementally (`cElementTree`). In addition, I imported some more library to help with several other tasks as part of the data wrangling process including regular expressions, `json` (for file writing), `pretty print` (to print results that can be read easily), `string` (for string manipulation), `ast` (to convert a string variable into a dictionary during the file read process), and `pymongo` for communicating with mongodb.

1 Data Overview

File Size

kolkata_India.osm : 685.6 MB

kolkata_sample.osm: 65.3 MB

kolkata_india.osm.json: 1.03 GB

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

Basic statistics about the dataset

1. Number of documents: 577099
   ```
   > db.kolkata_map.find().count()
   ```
2. Number of nodes: 516166
   ```
   > db.kolkata_map.find({"type":"node"}).count()
   ```
3. Number of ways: 60932
   ```
   > db.kolkata_map.find({"type":"way"}).count()
   ```
4. Number of unique users: 254
   ```
   > db.kolkata_map.distinct('created.user').length
   ```

Other informations:

- Top Most contributed user
  ```
  > db.kolkata_map.aggregate([{$group:
  {'_id':'$created.user','count':{$sum:1}}}, {$sort:
  {'count':-1}}, {$limit:1}])
  {"_id":"Rondon237","count":255943}
  ```

- Number of users appearing only once (having 1 post)
  ```
  > db.Kolkata_map.aggregate([{$group:
  {'_id':'$created.user','count':{$sum:1}}}, {$group:
  {'_id':'$count','num_users':{$sum:1}}}, {$sort:
  ```

```
{'_id':1}},{$limit:1}])
```

```
{"_id":1,"num_users":47}
```

Additional Idea

Top 10 Users contributed to the OSM - Kolkata, India

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

```
{ “_id” : “Rondon237”, “count” : 255943 }
{ “_id” : “sakthivel”, “count” : 88362 }
{ “_id” : “maxsaurav”, “count” : 73081 }
{ “_id” : “baigan”, “count” : 16635 }
{ “_id” : “dmgroom_coastlines”, “count” : 16180 }
{ “_id” : “Heinz_V”, “count” : 15812 }
{ “_id” : “sujandeb”, “count” : 15422 }
{ “_id” : “iambibhas”, “count” : 15306 }
{ “_id” : “Japa”, “count” : 10719 }
{ “_id” : “Oberaffe”, “count” : 9701 }
```

- Contribution to the dataset by top user : 44.3%
- Contribution to the dataset by top 2 user : 59.6%
- Contribution to the dataset by top 10 user : 89.6%

As, we can see from the above trend only 10 out of 254 users contribute to around 89.6% 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 asking them to post on social networking sites , so that more people will become aware about OSM and start contributing or asking them to form groups and contribute to the OSM project ,for the improvement of their city or state .

Benefits of Improving the OSM data :

1. As most of the people use smart phone these days , they can contribute to OSM using their GPS , which results in a more accurate data .
2. With more accurate data about the place available , developers can use it to design applications like, to suggest best restaurants near a place or where can they find a nearest petrol pump to the end user.
3. The data can also be used by government or private institutions to study about the geography of the place like the number of offices, buildings, schools, hospitals etc .

Anticipated problems in implementing the improvements :

1. ot 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 .

2. ome users may not be willing to post to the social networking sites about their contribution to the project .
3. evelopers may design an application 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 .

Additional data exploration using MongoDB queries

Diversity of Religion

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

```
{ "_id" : "hindu", "count" : 17 }
{ "_id" : null, "count" : 11 }
{ "_id" : "christian", "count" : 7 }
{ "_id" : "muslim", "count" : 5 }
{ "_id" : "Irrespective of religion", "count" : 1 }
{ "_id" : "buddhist", "count" : 1 }
{ "_id" : "sikh", "count" : 1 }
```

Popular Sports

```
> db.kolkata_map.aggregate([{ $match : { 'sport' : { $exists : 1 } } }, { $group : { '_id' : '$sport', 'count' :
```

```
{ $sum : 1 } } }, { $sort :{ 'count' : -1 } } ])
```

```
{ “_id” : “tennis”, “count” : 11 }
{ “_id” : “swimming”, “count” : 11 }
{ “_id” : “cricket”, “count” : 3 }
{ “_id” : “soccer”, “count” : 2 }
{ “_id” : “basketball”, “count” : 1 }
```

Top 10 appearing amenities

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

```
{ “_id” : “school”, “count” : 137 }
{ “_id” : “hospital”, “count” : 75 }
{ “_id” : “college”, “count” : 68 }
{ “_id” : “restaurant”, “count” : 57 }
{ “_id” : “bank”, “count” : 51 }
{ “_id” : “place_of_worship”, “count” : 43 }
{ “_id” : “fuel”, “count” : 43 }
{ “_id” : “atm”, “count” : 41 }
{ “_id” : “cinema”, “count” : 30 }
{ “_id” : “university”, “count” : 26 }
```

2 Problems encoutered

After running `Street_Parser,py`, I found there were three main problems in the data set . I will be discussing each problem in the following order :

- Irregularity in street names ("'D.r A.k paul raod'")
- Inconsistent postal codes ("700 027", "700095")
- Incorrect postal codes (K olkata area zip codes all begin with 700. however a portion of all d ocumented zip codes were starting with 743.)

Irregularity in Street Names
After observing the Irregularities in the Street name from the sample dataset , I tried to audit the complete data set for the city of Kolkata, India , that I downloaded from Mapzen . The following irregularities were found in the street names :

- Some of them were misspelled ("D.r A.k paul raod")
- Some were abbreviated ("Karbala Tank Ln")
- Unnecessary extra information was provided in some cases- ("Major Arterial Rd, Action Area IID, New Town")
- There was no consistency in the Street names as the data came from various sources ("RAJA SUBODH CHANDRA MULLICK ROAD","Daud Ali Dutta Sarani(Sukea Row)")

After auditing the complete data set, I formulated a data cleaning plan for the street names . And then , I wrote a program `audit.py` that iteratively parsed over each of the street names in the kolkata_map.osm file and performed the cleaning action .

After cleaning , the following things were corrected :

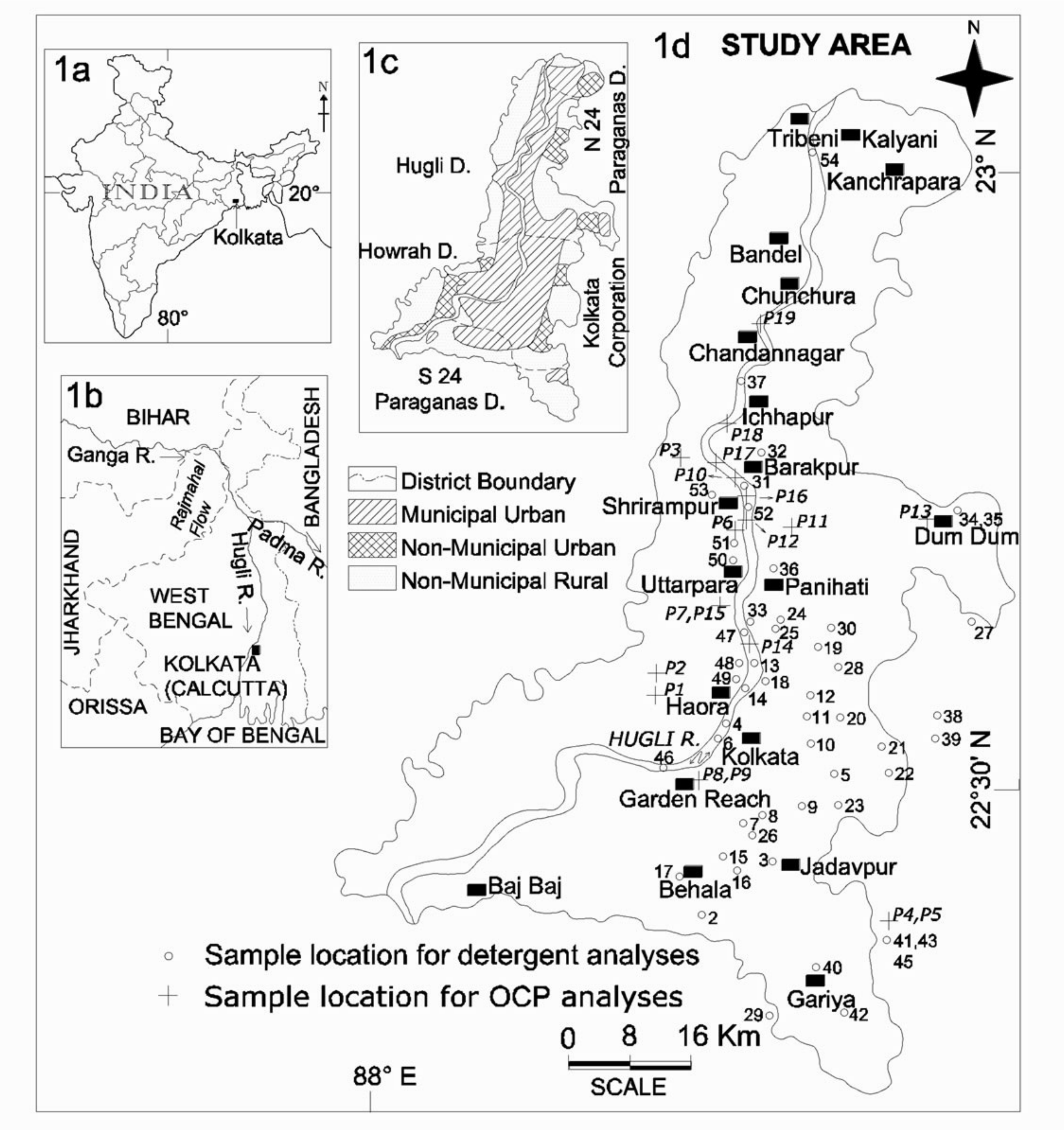
- “'D.r A.k paul raod” => “D.r A.k Paul Road”
- “Karbala Tank Ln” => “Karbala Tank Lane”
- “Major Arterial Rd, Action Area IID, New Town” => “Major Arterial Road”
- “RAJA SUBODH CHANDRA MULLICK ROAD” => “Raja Subodh Chandra Mullick Road”
- “Daud Ali Dutta Sarani(Sukea Row)” => “Daud Ali Dutta Sarani”

Postal Codes

Postal code strings posed a different sort of problem, they were mentioned in different formats like “700 027” , “700095” ,”7000 026” I performed a cleaning action in the `audit.py` to represent them all in a consistent format .
When I tried to view all the postal codes after cleaning , grouped in descending order of their counts using kolkata_map_overview.py , I found a two completely different kinds of problem .
out put of `kolkata_map_overview.py`

```
700064 => 911
741235 => 71
743363 => 57
712306 => 31
700107 => 22
```

The output of postal_codes.py shows top 5 postal codes for the city of Kolkata . But when I googled over it, I found that the postal codes for all the cities of Kolkata start with (700) , but there were many cities in the data set whose postal codes start with (741) . Doing further analysis over this , I found that (741) belongs to the city of kalyani in West Bengal , that falls under the Kolkata Metropolitan Area . There were also other cities in the data set whose postal codes start with (743) , (711) , (741) , (721) which included some other cities in the KMA (Kolkata Metropolitan Area) like Hoogly and Howrah . Thus the postal codes of the data set were not incorrect but they were of the cities in the KMA , hence the dataset could be more aptly named as KMA data set .



There were cities with postcode (9400) which seemed unique .

So , I tried to insert the data into MongoDB after cleaning and converting the data into proper dictionary format using `Data.py` and performed the following query:

Find cities with postcode 9400

```
> db.kolkata_map.aggregate([ { $match : {
```

`'address.postcode' : { $exists : 1 }, 'address.postcode' : '9400' } }, { $project : { ‘_id’ : 0, 'address.city' : 1, 'address.postcode' : 1 } } ])`
Result of the query :
{ “address” : { “postcode” : “9400”, “city” : “Satkhira” } } { “address” : { “city” : “Satkhira”, “postcode” : “9400” } }

Satkhira is a district in Bangladesh . It shares its boundary with West bengal, India . So , I removed the documents with postal code (9400) as it would have been inappropriate to include these cities in the Kolkata Map dataset .

3 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 data for the city of Kolkata comes from Prototype Global Shoreline (PGS) .
The Prototype Global Shoreline (PGS) is a datasource which OSM have used for a lot of coastline data. The data was extracted from the public domain Landsat imagery via an image recognition algorithm (Vectorisation software) . As Kolkata is located in the banks of river Hoogly, it can be expected that most of it’s data comes from PGS source .
People of Kolkata who really care can be encouraged to fixup the PGS data to refine the accuracy. This may be possible using better imagery, ontheground GPS surveying, or other techniques. It is also possible (globally) using Landsat again, but this time manually. With a bit of

human judgement we can work out the artefacts e.g. to achieve more natural lines/curves instead of zigzags.