OpenStreetMap Project
Data Wrangling with MongoDB

Rajesh Nair

Map Area: Mumbai, India

http://www.openstreetmap.org/export#map=19/18.95238/72.83271
https://mapzen.com/metro-extracts/

1. Problems Encountered in the Map

After downloading the Map data for Mumbai and running it against tags.py, audit.py and data.py, I noticed the following problems with the data, which I will discuss in the following order:

- **Incorrect key names for tags:** Key names ike "nerul","Laxmi Tower CHS","Mahesh Jain" are names of Place, Apartment, Person and does not indicate any attribute about the data
- **Inconsistent postal codes**: The Postal code field has values in the following formats “43”, “40060”, “400 601”, “400076, India”, “4oooo89” which are valid but either has some missing / additional characters or have a different format
- **Incorrect postal codes:** *Mumbai area zip codes all begin with “4” however a few of the zip codes like “boundry”, “110092”, “123” were either incorrect or outside this region*
- **Inconsistent Street Names:** The street names do not follow a defined naming pattern and there are a few streets which use abbreviation / different naming convention which were to be standardized. For eg: “(East)” has the following different variations “(E)”, “Easr”, “East”, “East,”
- **Few tags have “type” as the key:** The Nodes, Ways and relations are identified by using the “type” key in the JSON Document which is being created by data.py. In the Tags element there is a key called “type”. So the value of the type used for indicating the type of node was getting updated with the value of the “type” key in the tags element.
- **The Postal code is represented by 2 different names for the key:** In addition to the “addr:postcode” key the post code is also appearing as “postal_code” key. Hence, we are having 2 different names for the same attribute which needs to be standardized
- **The key “internet_access”is having more than 2 values:** The key “internet_access” in the tags element used to denote if Internet Access is available or not at a facility has more than 2 values. There are 3 values for this key “yes”, “no” and “wlan”.

Incorrect key names for tags

The Incorrect key names ike "nerul","Laxmi Tower CHS","Mahesh Jain" do not define any property for the given data and hence was ignored during processing of the tags by having all such keys included in a Python List called “IGNORE_KEYS” and ignoring these keys while processing the tags

Inconsistent postal codes

The Postal codes in India are of 6 characters in length and for Mumbai the postal code starts with “4”. Some of the postal codes had a space after the first 3 characters (For e.g., 400 601 was updated to 400601).
There were postal codes which had either one of the zeroes missing or extra thus making it a 5 character or a 7 character code. The 5 character postal codes were corrected by adding additional zero to the string and the 7 character postal codes were corrected by removing the extra zero (For e.g., 40060 was fixed by adding a zero to correct it to 400060 and 4000072 was corrected by removing the extra zero to correct it to 400072).
There were postal codes which had “o” in place of Zero and were corrected by replacing the “o” with Zero “0” (For e.g., “4oooo89” was corrected to “400089”)
Sometimes the 2 digit postal codes are used when the preceding 4 characters would be “4000”. For e.g., “43”. This was corrected by prefixing it with “4000” and it got corrected to “400043”

Incorrect postal codes

The Postal codes which didn’t start with “4” and could not be corrected were updated by prefixing it with “Invalid Post code =>” string (For e.g., 63103 was updated to “Invalid Post code => 63103”)

Inconsistent Street Names

The Street names in Mumbai do not have a defined structure. Hence, most often the street name will also contain the locality information along with the street name. For e.g., “Tulinj Road, Radha Niwas, Nalasopara Easr”. There is not much that we can do to remove these inconsistencies. There were some other inconsistencies like the Key word “East” was appearing as “Easr”, "East", "East,", “ (E)”, “(East)”. All of these were updated as “(East)”

Few tags have “type” as the key

The Key “type” has been used to identify if the data in the JSON document is for a Node, Way or a relation. For certain tags there is a key called “type”. The value of this key tag was updating the value of the “type” key and the identity of the document as to if it is a “Node”, “Way” or a “Relation” was getting lost. This issue was overcome by updating the “type” key in the tags element with the name “tag_type”

The Postal code is represented by 2 different names for the key

The Postal code is being identified by 2 keys i.e., "addr:postcode" key and the "postal_code" key. Mostly, for the "Way" element the Postal code is appearing as "postal_code" instead of "addr:postcode". To have standardization in representing the postal codes the postal_code key has been updated to addr.postcode key while creating the JSON document

The key "internet_access"is having more than 2 values

The key "internet_access" in the tags element is used to denote whether Internet access is available at a given facility or not and should have just 2 values "yes" or "no". But, in the date it has 3 values "yes", "no" and "wlan". To fix this problem while creating the JSON document we update the value of this field such that if the value is "no" then it will be updated with "no" and for anything other than "no" it is updated to "yes"

2. Data Overview

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

File sizes

mumbai_india.osm .... 104.98 MB
mumbai_india.osm.json .... 120.79 MB

# Number of documents

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

584743
```

# Number of nodes

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

536724
```

# Number of ways

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

47816
```

# Number of relations

```
> db.mapdata.find({"type":"relation"}).count()

203
```

# Number of unique users

```
> db.mapdata.aggregate([{"$group":{"_id":"$created.user", count":{"$sum":1}}},
{"$group":{"_id": "","count":{$sum:1}}},{$project:{"_id":0,"Count of Unique
Users":"$count"}}])

{ "Count of Unique Users" : 862 }
```

# Top 20 contributing user

```
>db.mapdata.aggregate([{"$group":{"_id":"$created.user","count":{"$sum":1}}},{"$
sort":{"count": -1}}, {"$limit":20}])

{ "_id" : "parambyte", "count" : 87963 }
{ "_id" : "PlaneMad", "count" : 65553 }
{ "_id" : "balaji88", "count" : 57993 }
{ "_id" : "MJL Wood", "count" : 57816 }
{ "_id" : "udaya", "count" : 43474 }
{ "_id" : "smith_dsm", "count" : 38340 }
{ "_id" : "Giyavudeen", "count" : 28975 }
{ "_id" : "Heinz_V", "count" : 21717 }
{ "_id" : "indigomc", "count" : 19208 }
{ "_id" : "singleton", "count" : 15548 }
{ "_id" : "shekhar", "count" : 13113 }
{ "_id" : "Moorthy1", "count" : 11754 }
{ "_id" : "PremK", "count" : 10007 }
{ "_id" : "Oberaffe", "count" : 9416 }
{ "_id" : "gaurav jain", "count" : 7234 }
{ "_id" : "Meghanand", "count" : 5990 }
{ "_id" : "dmgroom_coastlines", "count" : 5860 }
{ "_id" : "Aadesh", "count" : 4827 }
{ "_id" : "jain zachariah", "count" : 4512 }
{ "_id" : "Shekhar11", "count" : 4451 }
```

3. Additional Ideas

Data Exploration and insights about the data using Python programs

The number of tag elements in the Open Street Map for Mumbai suing mapparser.py

```
{'bounds': 2,
 'member': 4738,
 'nd': 1201206,
 'node': 1073448,
 'osm': 2,
 'relation': 406,
 'tag': 256012,
 'way': 95632}
```

Additional insights about the data and how they have been shaped in to a JSON document using the data.py program

After exploring the data, I felt that the following fields give information about the Amenity and hence created a sub document called amenity and used a Python List called AMENITY to hold the attributes describing Amenity

```
AMENITY = ["religion", "denomination", "cuisine","opening_hours",
"capacity","smoking","covered", "designation",
"toilets:disposal","social_facility","social_facility:for"]
```

The Value for the amenity was converted to lower case as there were cases where the Religion and other values were in mixed case strings

I found that some of the keys were not useful for describing the data and can be ignored while creating the JSON document. I used a Python List called IGNORE_KEYS to maintaining the list of such keys which were not to be processed

```
IGNORE_KEYS =["gns", "WDPA", "communication", "Sector", "diplomatic", "mangeshi
dham","mangeshi dham", "Mangeshi Dham", "business Park", "contact", "leaf_cycle",
"nearyouu", "place:", "ship:type", "Cable TV Provider", "EState Consultants",
"earthquake:damage", "Golden Park", "mtb:scale", "oneway:bicycle", "alt_name:",
"namePREAM SHARMA COMMUNICATION","name:pt", "name:pl",
"name:jbo","turn:lanes","wikipedia:","population:","nerul","Laxmi Tower CHS","Mahesh
Jain"]
```

The Address components are starting with addr word. All of the address components have been made part of the address subdocument

The name has different regional language notation and they are stored in keys after “name:”. I have saved these under the native_names subdocument. There were keys like 'name:pt','name:pl','name:jbo' which I felt are not giving much information and chose to not include in the JSON document

The alternative names for a node is appearing with the following different keys "new_name","old_name","alt_name","alternative_name". I used a Python List

ALT_NAMES to maintain these keys and used the “alt_names” array to store all the alternative names used for a given node

Additional data exploration using MongoDB queries

# Total count of Railway Stations

```
>db.mapdata.aggregate([{$match:{"type":"node","railway":"station"}},{$group:{"_i
d":"",count:{$sum:1}}},{$project:{"_id":0,"Count of Railway Stations identified in
the Map:": "$count"}}])
```

```
{ "Count of Railway Stations identified in the Map:" : 148 }
```

Railways is one of the Backbone of Mumbai Transportation system and is evident from the count of Railway stations identified in the Open Street Map data for Mumbai

# Top 10 appearing amenities

```
>db.mapdata.aggregate([{"$match":{"amenity":{"$exists":1}}},
{"$group":{"_id":"$amenity.type","count":{"$sum":1}}},          {"$sort":{"count":-1}},
{"$limit":10}, {"$project":{"_id":0,"Amenity":"$_id","Count":"$count"}}])
```

```
{ "Amenity" : "place_of_worship", "Count" : 354 }
{ "Amenity" : "restaurant", "Count" : 241 }
{ "Amenity" : "school", "Count" : 234 }
{ "Amenity" : "bank", "Count" : 199 }
{ "Amenity" : "hospital", "Count" : 154 }
{ "Amenity" : "fuel", "Count" : 123 }
{ "Amenity" : "parking", "Count" : 121 }
{ "Amenity" : "bus_station", "Count" : 112 }
{ "Amenity" : "cafe", "Count" : 107 }
{ "Amenity" : "college", "Count" : 94 }
```

The top amenity appearing in the list is Place of Worship followed by restaurant, followed by school. This shows that the users have added more Place of worship as compared to the restaurants and school. In reality in Mumbai the count of schools and restaurants are more than the place of worship

# Place of Worship by Religion

```
>db.mapdata.aggregate([{"$match":{"amenity.type":"place_of_worship"}},
{"$group
":{"_id":"$amenity.religion","count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$pr
oject":{"_id":0,"Religion":"$_id","Count of Place of Worship":"$count"}}])
{ "Religion" : "hindu", "Count of Place of Worship" : 122 }
{ "Religion" : "muslim", "Count of Place of Worship" : 95 }
{ "Religion" : "christian", "Count of Place of Worship" : 52 }
{ "Religion" : null, "Count of Place of Worship" : 49 }
{ "Religion" : "jain", "Count of Place of Worship" : 9 }
{ "Religion" : "zoroastrian", "Count of Place of Worship" : 9 }
{ "Religion" : "buddhist", "Count of Place of Worship" : 7 }
{ "Religion" : "sikh", "Count of Place of Worship" : 6 }
{ "Religion" : "jewish", "Count of Place of Worship" : 4 }
{ "Religion" : "hare_krishna", "Count of Place of Worship" : 1 }
```

As expected, the count of place of worship for Hindus is maximum followed by Muslim and then followed by Christian. There are many Place of worship which are missing the religion and accounts for 14% of the Place of worship data

# Top 10 Cuisines

```
> db.mapdata.aggregate([{"$match":{"amenity.type":"restaurant"}}, {"$group":{"_id":"$amenity.cuisine","count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$project":{"_id":0,"Cuisine":"$_id","Count":"$count"}},{$limit:10}])
{ "Cuisine" : null, "Count" : 119 }
{ "Cuisine" : "indian", "Count" : 40 }
{ "Cuisine" : "regional", "Count" : 15 }
{ "Cuisine" : "vegetarian", "Count" : 12 }
{ "Cuisine" : "pizza", "Count" : 9 }
{ "Cuisine" : "italian", "Count" : 7 }
{ "Cuisine" : "chinese", "Count" : 7 }
{ "Cuisine" : "international", "Count" : 5 }
{ "Cuisine" : "burger", "Count" : 5 }
{ "Cuisine" : "seafood", "Count" : 2 }
```

Majority of the Restaurants around 119 does not have the cuisine populated in the Map data.

# Count of Nodes having Post code, Street and Address

Count of Nodes where Address exists

```
>db.mapdata.aggregate({$match:{"address":{"$exists":1},type:"node"}},{"$group":{"_id":"",count:{$sum:1}}})
```

```
{ "_id" : "", "count" : 867 }
```

Count of Nodes where Postal Codes exists

```
>db.mapdata.aggregate({$match:{"address.postcode":{"$exists":1},type:"node"}},{"$group":{"_id":"",count:{$sum:1}}})
```

```
{ "_id" : "", "count" : 615 }
```

Count of Nodes where Street exists

```
>db.mapdata.aggregate({$match:{"address.street":{"$exists":1},type:"node"}},{"$group":{"_id":"",count:{$sum:1}}})
```

```
{ "_id" : "", "count" : 629 }
```

4. Conclusion

Following are my findings from the map data of Mumbai

Since, Street names are not standardized it is difficult to ascertain if the address is complete for the nodes which have addresses. There are many nodes where the address is missing or incomplete. As noted above out of the 867 nodes which have address, only 615 of them have Postal code and 629 of them have Street address

There are 862 distinct users who have contributed to the Mumbai Map data. The user contribution is skewed. The top 10 users account for 75% of the map data and the top 20 users account for around 88% of the map data. The remaining 842 users account for only 12% of the data

From the map data the count of place of worship exceeds the count of schools and restaurants. In reality the number of schools and restaurants far exceeds the count of place of worship in Mumbai

In the data it can be seen that most of the restaurants are missing the Cuisine information. Also, the Religion information is missing in almost 14% of the Place of worship data. Hence, lots of information is missing in the Open Street Map data for Mumbai region

The Map data appears to be very sparse and there is a lot of scope for adding more nodes to the Open Street Map data for the Mumbai Region