OpenStreetMap Data Wrangling

By Piyush Goyal

Map Data : *Los Angeles, California*

Problems :

Section 1 : Data auditing and data cleaning

Tags and keys counts

Using the function inside the heading ‘Tags and keys counts’ in .ipynb file, the count occurrences of each tag was calculated.

```
{'bounds': 1,
 'member': 93198,
 'nd': 12782177,
 'node': 10955705,
 'osm': 1,
 'relation': 13471,
 'tag': 8246277,
 'way': 1224761}
```

Additional functionalities were added to above function to get the count of key attribute in tag. 20 of them with most occurrences are

```
[ ('building', 670742), ('highway', 566014), ('ele', 484618), ('height', 470715), ('lacounty:bld_id', 469648),
('lacounty:ain', 469468), ('start_date', 443346), ('building:units', 422001), ('name', 380151), ('tiger:county',
243155), ('tiger:cfcc', 242914), ('source', 226421), ('tiger:name_base', 225098), ('tiger:name_type',
```

203989), ('tiger:zip_left', 185387), ('tiger:zip_right', 179588), ('tiger:reviewed', 121028), ('tiger:source', 112912), ('tiger:tlid', 112329), ('tiger:separated', 106620)]

After getting the count of key occuring, to know what each key hold and what the value means, I wrote maximum of 20 values of each key into a json file. This gave insight of some inconsistent data format (like 40mph vs 40). But I didn't corrected it.

Special 'tag' keys values

So there were **704** tags with count equal 1 from which some were very specific while some were not necessary. So I manually made list of keys which are to be filter out before populating the database which also contained some key-value tags with count more than 1 which has redundant data.

Multiple Zip Codes

On observing the data, the zip codes were mapped to multiple keys (eg. tiger:zip_left, tiger:zip_right, etc) and also in all some of the keys multiple zip codes were present with ';' as a delimiter. So I mapped all the zip codes to the key 'zipcodes' which is a list of all the zipcodes observed.

Phone Numbers

Phone numbers were formatted inconsistently. It contained values like 'yes', so all these values were ignored and the all phone numbers were re-formatted to standard form (951) 587 2505 using phonenumbers module in python.

Street Names Abbreviated

There was too much inconsistency in street names. So I changed all the street names to titliesed for (avenue -> Avenue). Also all '.', ',' characters were removed. Also street names like 'Telephone Ave Suite 12' are changed to 'Telephone Avenue Suite 12'.

Section 2 : Data Overview

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

File Sizes

los-angeles_california.osm.json : ***2.5GB***
los-angeles_california.osm   : ***2.5GB***

Number of Documents

```
> db.losAngelesCA.count()
  12180466
```

Number of nodes and way

```
> db.losAngelesCA.find( { ‘type’ : ‘node’ } ).count()
  10955705

> db.losAngelesCA.find( { ‘type’ : ‘way’ } ).count()
  1224761
```

Number of Distinct Users

```
> db.losAngelesCA.distinct('created.user')
```

Top 10 contributing users

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

Number of Users contributing only once

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

Section 3 : Additional Ideas

With the queries conducted, there was more focus on the node places rather than ways. There was additional presence of relation and member tag, although auditing and analysing of those tag would have given us some more insight about the data but I left them out.

Data set can be improved if a sample of data set is displayed to the users or adding regex in the form so that I would only pass the accepted format.

The benefits of this implementation is now most of the street data is in a desired format, all the zip codes are in a key, all phone numbers are in standard form. But the problem is that most of the data is potentially outdated. According to TIGER WIKI, the most recent data was updated on 2014. It can be a good exercise to use GOOGLE MAPS API to validate the results.

Contributor statistics

- ➔ Top user contribution percentage (“schleuss_imports”) : ***8.65%***
- ➔ Combined top 2 users' contribution (“schleuss_imports” and “manings_labuildings”) : ***13.87%***
- ➔ Combined Top 10 users contribution : ***39.19%***
- ➔ Combined contribution of users contributing only once - ***0.006%***

Conclusion

After this review of the data it’s obvious that the ***Los Angeles California*** area is incomplete, though I believe it has been well cleaned for the purposes of this exercise.