DAND Project No.3: OpenStreetMap Project

Data Wrangling with MongoDB

By Tyler Jin

Map Area: Shanghai, China

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

Notes:

- This is my second submission. According to the feedback from first grader, I made following revision in this version:
    - Improve the 'Suggestion & Conclusion' part of this report.
    - Add docstring to specify the usage of each function in the coding file.
- Information of raw data and coding reference can be found in 'data source and coding reference.txt' in my assignment folder.
- All code for this project can be found in 'p3_osm_project_tyler_jin.ipynb' or equivalent html file. I put most of detail explanation in that file. Please review that file first and then this report.

1. Some issues about this dataset:

a. The same attribute sometimes is recorded in different languages (English, Simplified Chinese, Chinese Pinyi, Traditional Chinese and other languages).
b. The format of 'source' value is messy.

Solution to issue a:

Step 1: Before being imported to MongoDB, a name collection is set up for each item. Different languages are grouped inside this collection. (Code file part 3)

Step 2: After being imported to MongoDB, field names inside name collection are further cleaned through MongoDB update and rename methods. (Code file part 4)

After cleaning, major name languages are counted as below:

```
Simplified Chinese: 93275
English: 44866
Chinese Pinyin: 4796
Traditional Chinese: 1052
```

Solution to issue b:

Step 1: High frequency sources, such as ‘GPS’ and ‘Bing’ are grouped before data being imported to MongoDB. (Code file part 3)

Step 2: After being imported, other high frequency source types are inspected, similar words are standardized. (Code file part 4)

After cleaning , top cited sources are listed below:

```
pipeline=[
  {'$match':{'source':{'$exists':1}}},
  {'$group':{'_id':'$source','count':{'$sum':1}}},
  {'$sort':{'count':-1}},
  {'$limit':20}
]
list(collection.aggregate(pipeline))
```

```
{'_id': 'PGS', 'count': 87342},
{'_id': 'Bing', 'count': 37504},
{'_id': 'GPS', 'count': 6650},
{'_id': 'Estimation', 'count': 681},
{'_id': 'Yahoo', 'count': 643},
{'_id': 'survey', 'count': 534},
{'_id': 'osm-gpx', 'count': 471},
{'_id': 'Landsat', 'count': 349},
{'_id': 'GNS', 'count': 279},
{'_id': 'GISmaps/std', 'count': 156},
{'_id': "potlatch's P-key", 'count': 150},
{'_id': 'Lakewalker / Landsat', 'count': 106},
{'_id': 'Survey', 'count': 105},
{'_id': 'photograph', 'count': 93},
{'_id': 'Mapbox', 'count': 62},
{'_id': 'China Data Center, University of Michigan', 'count': 45},
{'_id': 'Strava', 'count': 36},
{'_id': 'PGS & Bing', 'count': 26},
{'_id': 'wild_guess', 'count': 22},
{'_id': 'scanaerial', 'count': 22}
```

2. Data Overview:

 # File Sizes:

```
shanghai_sample.osm: 6 M
shanghai_sample.osm.json: 10 M
shanghai_china.osm: 633 M
shanghai_china.osm.json: 961 M
```

 # Number of Documents:

```
>collection.find().count()
3524981
```

 # Number of Nodes:

```
>collection.find({'type':'node'}).count()

3136391
```

 # Number of Ways:

```
>collection.find({'type':'way'}).count()

385795
```

 # Number of Unique Users:

```
>len(collection.distinct('created.uid'))

2041
```

 # Top 5 Contributing Users:

```
>pipeline=[
        {'$match':{'created.user':{'$exists':1}}},
        {'$group':{'_id':'$created.user','count':{'$sum':1}}},
        {'$sort':{'count':-1}},
        {'$limit':5}
    ]
>list(collection.aggregate(pipeline))
  {'_id': 'Chen Jia', 'count': 661742},
  {'_id': 'aighes', 'count': 183143},
  {'_id': 'katpatuka', 'count': 137812},
  {'_id': 'XBear', 'count': 126490},
  {'_id': 'yangfl', 'count': 112884}
```

 # Number of users who only contribute once:

```
>pipeline=[
       {'$match':{'created.user':{'$exists':1}}},
       {'$group':{'_id':'$created.user','count':{'$sum':1}}},
```

```
        {'$match':{'count':{'$eq':1}}}
    ]
>len(list(collection.aggregate(pipeline)))
408
```

3. Additional Ideas:

 # Top 10 amenities:

Top 2 amenities are bicycle rental and parking, both of which are transportation related. It is interesting that toilet is in top 10.

```
>pipeline=[
    {'$match':{'amenity':{'$exists':1}}},
    {'$group':{'_id':'$amenity','count':{'$sum':1}}},
    {'$sort':{'count':-1}},
    {'$limit':10}
]
>list(collection.aggregate(pipeline))
 {'_id': 'bicycle_rental', 'count': 2436},
 {'_id': 'parking', 'count': 1323},
 {'_id': 'school', 'count': 1231},
 {'_id': 'restaurant', 'count': 1229},
 {'_id': 'bank', 'count': 588},
 {'_id': 'toilets', 'count': 410},
 {'_id': 'fuel', 'count': 357},
 {'_id': 'cafe', 'count': 354},
 {'_id': 'fast_food', 'count': 339},
 {'_id': 'hospital', 'count': 338}
```

 # Top 10 cuisines:

No surprise, top 1 cuisine is Chinese food, followed by burger/pizza/chicken. Popular fast food is the same in every big city.

```
>pipeline=[
```

```
    {'$match':{'cuisine':{'$exists':1}}},
    {'$group':{'_id':'$cuisine','count':{'$sum':1}}},
    {'$sort':{'count':-1}},
    {'$limit':10}
]
>list(collection.aggregate(pipeline))
 {'_id': 'chinese', 'count': 161},
 {'_id': 'burger', 'count': 51},
 {'_id': 'pizza', 'count': 31},
 {'_id': 'chicken', 'count': 31},
 {'_id': 'coffee_shop', 'count': 30},
 {'_id': 'italian', 'count': 14},
 {'_id': 'japanese', 'count': 14},
 {'_id': 'american', 'count': 14},
 {'_id': 'international', 'count': 12},
 {'_id': 'asian', 'count': 12}
```

 # Top 10 natural objects:

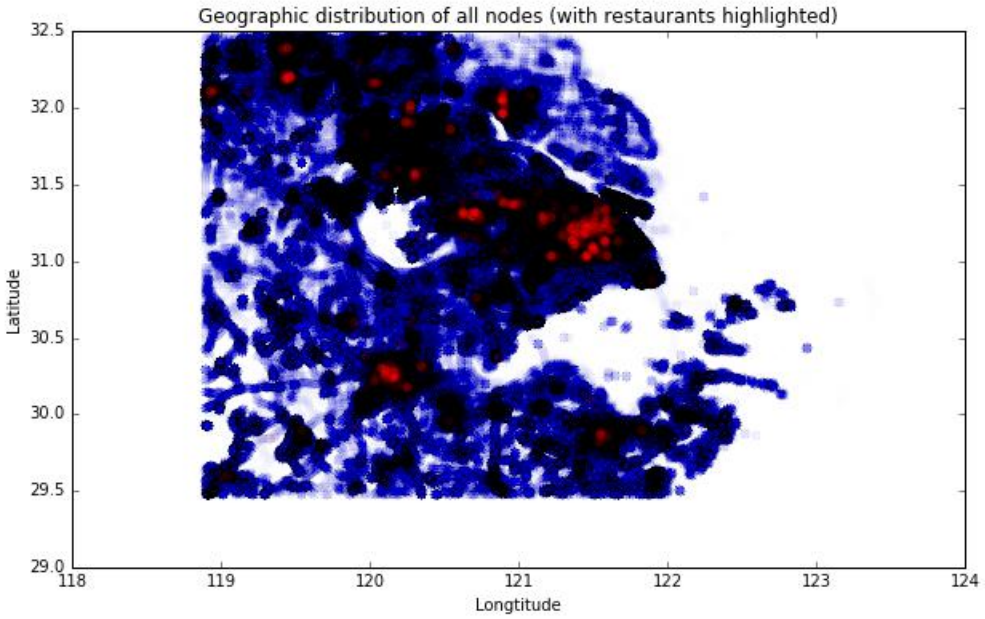
Water and trees are most common natural objects in this city. Since the east side of shanghai is sea, coastline is also frequently marked in OSM.

```
>pipeline=[
    {'$match':{'natural':{'$exists':1}}},
    {'$group':{'_id':'$natural','count':{'$sum':1}}},
    {'$sort':{'count':-1}},
    {'$limit':10}
]
>list(collection.aggregate(pipeline))
 {'_id': 'water', 'count': 6288},
 {'_id': 'tree', 'count': 3454},
 {'_id': 'coastline', 'count': 1631},
 {'_id': 'wood', 'count': 986},
```

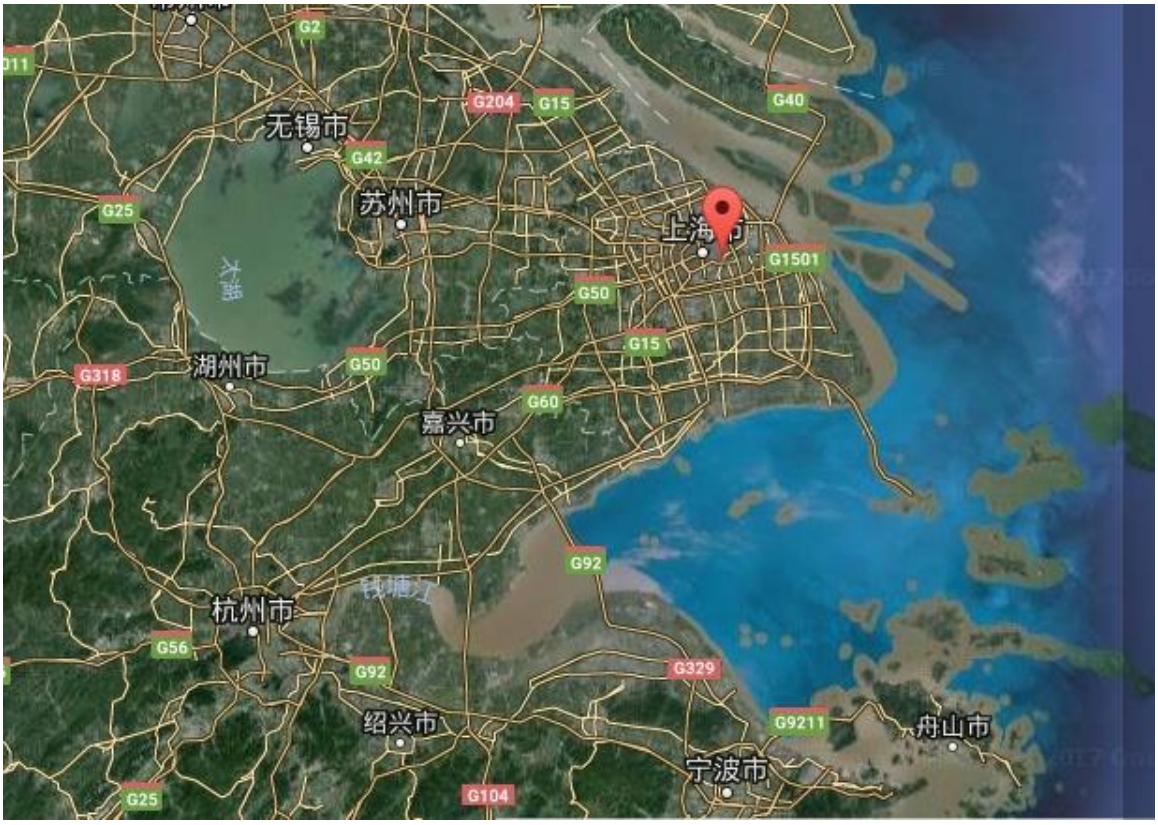
```
{'_id': 'scrub', 'count': 124},
{'_id': 'sand', 'count': 108},
{'_id': 'tree_row', 'count': 80},
{'_id': 'peak', 'count': 80},
{'_id': 'wetland', 'count': 50},
{'_id': 'grassland', 'count': 36}
```

 # Reconstruct map with latitude/longitude data:

I want to explore the geographic pattern of all nodes in this oms dataset, so I use longitude and latitude as x and y to generate a scatter plot, and highlight restaurants in it.



The shape of the scatter is basically the same as the real geographic shape of that area. The right side is completely blank, because the east side of Shanghai is sea. Some dense clusters of nodes (blue) can be observed, which is aligned with the cluster pattern of restaurants (red). I assume those areas are major towns for residence. I snapshotted a map picture with the same longitude/latitude frame. Turned out that OSM metro extract data package includes not only the data of downtown Shanghai, but also a broad area around Shanghai. The largest cluster in the middle is downtown Shanghai, and those small clusters correspond to other cities and towns near Shanghai. There is a hole on the left side of Shanghai, that is a famous lake – Tai Lake. In the left bottom corner of the scatter plot, some line pattern can be observed, that might be highway/national lane/railway.



4. Suggestion & Conclusion:

Potential limitation of current dataset:

- OpenStreetMap is not a well-known platform in China. Personally, I never heard of it before this project. The lack of popularity among local people is also reflected in the language usage in this dataset. Many manually generated nodes are recorded in English only (might be created by expats who live in Shanghai). The information of restaurants and other daily life facilities are far from being comprehensive: among 3.5 million nodes, only ~1000 nodes are restaurant record; The number of most popular nodes, bicycle rental and parking, is also small.

Suggestion to improve the dataset:

- Leverage other platforms' API: Google service is not available in China right now, but some popular local lifestyle platforms have API available. One good option is http://www.dianping.com. The most famous lifestyle recommendation platform including detail rating and information of local restaurants and entertainment places (Similar to Yelp or TripAdvisor). OSM's data variety will be significantly improved if it can import local data from Dianping's API.
- Develop mobile App to encourage user's participation: Encourage people to check-in on OSM mobile App (like what the platform 'Foursquare' did). Give out coupons of nearby places to motivate people's participation.

Benefits:

- Leveraging API of existing local platforms is the fastest way to collect massive user generated data.
- People spend more time on mobile phone than on desktop end. They are more likely to log in OSM if a convenient mobile App is available.

Anticipated problems:

- Data of the whole city is massive, it might take a long time to import existing data from another platform's API. Those platforms might also forbid or unable to sustain massive data extraction.
- People will not use an App which they never heard of. Some marketing campaign activities might be necessary to boost the popularity of OSM in Shanghai.