Wrangle OpenStreetMap Data

Map Area

Shenzhen City, Guangdong, China

- http://www.openstreetmap.org/relation/3464353 (http://www.openstreetmap.org/relation/3464353)
- https://mapzen.com/data/metro-extracts/metro/shenzhen_china/ (https://mapzen.com/data/metro-extracts/metro/shenzhen_china/)

Shenzhen City is my working city currently, I have been working here for more than 6 years, I am interested to explore more about the city. I have many years experience on SQL and Relational database, here, I shall try to use MongoDB to complete the exercise.

Problems encountered in the Map

After downloading the Shenzhen City's OSM OpenStreetMap, I wrote python script ***review_and_audit.py*** to review some main fields I care about, I noticed some problems with the data, and I will make plan for cleaning for each one later.

1. Inconsistent Phone Number
   China's country code is **+86** and the area code of Shenzhen City is **755**. However, with reviewing the inputted of OpenStreetMap with many kinds of formats as below:
   +86-755-2771 8888
   +86 (755) 2693 6888
   +86 755 25662355
   +86 755 8305 0888
   +86 755-83220215
   +8675582330888
   (86 755) 8298-9888
   0755-26832759
   24560331
   As I shall build the collection for Shenzhen City, that doesn't need the country code and area code and only keep the 8 numbers without other chars.
2. After reviewing the street type in the street name, I found some abbreviated address type (e.g. St,Ave,Av and Rd) or start with lowercase (e.g. road), I will update them accoring to below mapping.
   mapping = {

```
"St": "Street",
"S.": "Square",
"road": "Road",
"Ave": "Avenue",
"Av": "Avenue",
"Rd": "Road"
}
```

3. Inappropriate Amenity contains problematic characters For example

bicycle_rental
swimming_pool
bus_station;bank

It should be updated as bellow (replace "_", ";" to space)

bicycle rental
swimming pool
bus station bank

Cleaning and Shape elements

I prepared python script ***openStreetMap.py*** which will audit and correct above mentioned problems, shapping element for belows, so that output the data into JSON format before I import to MongoDB.

In [ ]:

```
<tag k="name" v="麦当劳 湖贝分店"/>
<tag k="amenity" v="fast_food"/>
<tag k="cuisine" v="burger"/>
<tag k="name:en" v="McDonald&#39;s"/>
<tag k="addr:city" v="深圳市 Shenzhen"/>
<tag k="addr:street" v="湖贝路 Hubei Rd"/>"
<tag k="addr:housenumber" v="1002"/>
```

Should be turned into:

In [ ]:
```
{...
"address": {
    "housenumber": 1002,
    "street": "湖贝路 Hubei Road"
}
"amenity": "fast food",
...
}
```

For "way" specifically:

In [ ]:
```
<nd ref="305896090"/>
<nd ref="1719825889"/>
```

Should be turned into

In [ ]:
```
"node_refs": ["305896090", "1719825889"]
```

For "relation" specifically:

In [ ]:
```
<member type="relation" ref="911844" role="subarea"/>
<member type="relation" ref="913110" role="subarea"/>
```

Should be turned into

In [ ]:
```
"member":[
    {
            "ref" : "911844",
            "role" : "subarea",
            "type" : "relation"
        },
        {
            "ref" : "913110",
            "role" : "subarea",
            "type" : "relation"
        },
]
```

Data Overview and Additional Ideas

File sizes

shenzhen_china.osm.....167M
shenzhen_china.osm.json.....245M

Number of unique users

In [ ]:
```
> db.shenzhen.distinct("created.user").length
```

879

Number of ways

In [ ]:
```
> db.shenzhen.find({"type":{"$eq":"way"}}).pretty().count()
```

92677

Number of nodes

In [ ]:
```
> db.shenzhen.find({"type":{"$eq":"node"}}).pretty().count()
```

821393

Top 10 contributing users

In [ ]:
```
> db.shenzhen.aggregate(
    [{$match:{"created.user":{"$ne":null}}},
     {$group:{"_id":{user_name:"$created.user"},count:{$sum:1}}},
     {$sort:{count:-1}},
     {$limit:10}
    ]
)
```

```
{ "_id" : { "user_name" : "MarsmanRom" }, "count" : 169547 }
{ "_id" : { "user_name" : "HelioFelix" }, "count" : 133129 }
{ "_id" : { "user_name" : "hlaw" }, "count" : 102309 }
{ "_id" : { "user_name" : "Triomphe346" }, "count" : 49361 }
{ "_id" : { "user_name" : "Philip C" }, "count" : 27132 }
{ "_id" : { "user_name" : "samho1234567" }, "count" : 20811 }
{ "_id" : { "user_name" : "a1579" }, "count" : 20320 }
{ "_id" : { "user_name" : "bgard" }, "count" : 20176 }
{ "_id" : { "user_name" : "happy-camper" }, "count" : 18585 }
{ "_id" : { "user_name" : "ch40s" }, "count" : 13962 }
```

Additional Exploration

I want to study how users input the words while they editing in OpenStreetMap.org, for example, the supermarket name "Walmat", using the below inquiring, has different formats, the user MarsmanRom edited for two places and has the same format with the name.

In [ ]:
```
> db.shenzhen.find(
    {$and:[{"shop":{"$eq":"supermarket"}},{"name": /.*a.*l.*t.*/}]},
    {"_id":0,"created.user":1,"name":1}
)
```

{ "name" : "WalMart", "created" : { "user" : "francesco" } }
{ "name" : "Wal Mart", "created" : { "user" : "dcs" } }
{ "name" : "沃尔玛 Walmart", "created" : { "user" : "MarsmanRom" } }
{ "name" : "沃尔玛 Walmart", "created" : { "user" : "MarsmanRom" } }
{ "name" : "Walmart", "created" : { "user" : "tartanabroad" } }

Top 5 number of supermarkets

In [ ]:
```
> db.shenzhen.aggregate(
    [{$match:{$and:[{"type":{"$eq":"node"}},{"shop":{"$eq":"supermarket"}},{"n
     {$group:{"_id":{name:"$name"},count:{$sum:1}}},
     {$sort:{count:-1}},
     {$limit:5}
    ]
)
```

{ "_id" : { "name" : "惠康 Wellcome" }, "count" : 4 }
{ "_id" : { "name" : "百佳超級市場 ParknShop" }, "count" : 4 }
{ "_id" : { "name" : "Vanguard Supermarket" }, "count" : 2 }
{ "_id" : { "name" : "沃尔玛" }, "count" : 2 }
{ "_id" : { "name" : "华润万家" }, "count" : 2 }

Additional Ideas

The downloaded metro extrac from mapzen (https://mapzen.com/data/metro-extracts/metro/shenzhen_china/) is not readly the Shenzhen City only which I would explore and precess, that actually includes parts of the neighboring city (e.g Hong Kong). Therefore, as a future improvement, the first ,I will filter out the non-Shenzhen nodes and their references. This must improve the data accuracy and impact the results of data anlysis.

The second, as the OpenStreetMap is editing by many users, or other reasons, some tags were given very detailed information ("k","v"), however, we still can find some tags with little "k" & "v" olny, even missed the most important information. We may resolve this issue by cross-referencing/cross-validating missing data from other database like Google API according to coordinate (lattitude & longtitude), this must improve the data validity and completeness for us.

Conclusion

After few exploration on the data, actually, I find many problems on the data, even I have cleaned some for it. I need to iterate to clean and explore many many times, so that the data can be ultimately used for analysis.