OpenStreetMap Data Case Study¶

Map Area¶

San Francisco,CA,United States

- https://www.openstreetmap.org/relation/111968
- https://mapzen.com/data/metro-extracts/metro/san-francisco_california/

San Francisco is one of the most beautifiul cities that I have even been to. So I choose San Francisco to do more investigation and to see if I can provide improvement on OpenStreetMap.org.

Data Audit¶

Unique Tags¶

After veiwing the dataset, we first wants to find different types of tag and count the numbers of unique tags. After we run Audit 1.py, we can see informtion below:

```
'bounds': 1,

'member': 54146,

'nd': 7527776,

'node': 6348477,

'osm': 1,

'relation': 6045,

'tag': 2005366,

'way': 779061
```

Patterns in the Tags¶

Before I process the data and add it into our database, I want to check the "k" value for each tag and see if there are any potential problems.I created 3 regular expressions to check for certain patterns in the tags. Using Audit 2.py, I have counted each of four tag categories.

```
'lower': 1293565,
'lower_colon': 685834,

'other': 25837,

'problemchars': 130
```

Problems Encountered in the Map¶

The orginal dataset is 1.25GB. We used code in Sample1.py to take a systematic sample from onginal dataset to do test first. I notice some problems which I will discuss in the following order:

- Overabbreviated street names: "Ave.","St.","Plz"
- Incorrect POstal code format:"515","1087","CA 94030"

Overabbreviated street names¶

The first problem I find in this dataset is Overabbreviated street names. We will use the audit_3.py to clean these data.

```
Abbreviations: Rd -> Road,Dr-> Drive

LowerCase :STREET->Street

Misspelling :socity -> Society

UpperCase Words: Ehs->EHS

Extra_words:By-pass->Bypass
```

Incomplete and incorrect postal codes¶

The zipcode of San Francisco begins with "94". We find some zipcode use incorrect 5 digit formats,so first, we will find all zipcode to see what infomation we need to correct. We will use audit 4.py to clean all postal codes.Now we will clean zipcode by following function. We will change all format into 5 digit standard formats. We also find some zipcode with string "CA", we will remove them.

```
Incorrect zipcode: ca => None,CA => None

Extra 4 digit zipcode :94002-2121 => 94002

Removing extra string :CA 94544 => 94544
```

Data Overview¶

After we audit and clean the data, we will save the new data and use following codes to import data into SQL database through Audit_5.py. This section contains basic statistics about the San Francisco OpenStreetMap dataset and SQL queries, and also some additional ideas about the data in context.

File sizes¶

```
san_francisco.osm ......... 1.25 GB

sf_sample.osm ............. 6.5 MB

nodes.csv ................. 514 MB

nodes_tags.csv ............ 9.17 MB

ways.csv .................. 45.2 MB

ways_tags.csv ............. 58.1 MB

ways_nodes.csv ............ 179 MB
```

Number of nodes¶

```
sqlite> SELECT COUNT(*) FROM Nodes;
```

6347454

Number of ways¶

```
sqlite> SELECT COUNT(*) FROM Ways;
```

785006

Number of unique users¶

```
sqlite> SELECT COUNT(DISTINCT(e.uid))

       FROM (SELECT uid FROM Nodes UNION ALL SELECT uid FROM Ways) e;
```

2673

Top 10 contributing users¶

```
sqlite> SELECT e.user, COUNT(*) as num
```

```
        FROM (SELECT user FROM Nodes UNION ALL SELECT user FROM Ways) e

        GROUP BY e.user

        ORDER BY num DESC

        LIMIT 10;

andygol,1293352

ediyes,912008

Luis36995,703533

dannykath,518968

RichRico,403972

Rub21,393065

calfarome,185558

oldtopos,167223

KindredCoda,149671

karitotp,134912
```

Number of users appearing only once (having 1 post)¶

```
sqlite> SELECT COUNT(*) FROM

        (SELECT e.user, COUNT(*) as num

         FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e

         GROUP BY e.user

         HAVING num=1)  u;
```

653

Additonal Ideas¶

Most popular cuisines¶

```
sqlite> SELECT nodes_Tags.value, COUNT(*) as num

        FROM nodes_Tags

            JOIN (SELECT DISTINCT(id) FROM nodes_Tags WHERE value='restaurant')
 i
```

```
            ON nodes_Tags.id=i.id
        WHERE nodes_Tags.key='cuisine'
        GROUP BY nodes_Tags.value
        ORDER BY num DESC
        LIMIT 10;
mexican,192
chinese,156
pizza,143
japanese,138
italian,129
thai,105
american,98
vietnamese,71
burger,56
indian,55
```

Most popular bank¶

```
sqlite> SELECT nodes_Tags.value, COUNT(*) as num
        FROM nodes_Tags
            JOIN (SELECT DISTINCT(id) FROM nodes_Tags WHERE value='bank') i
            ON nodes_Tags.id=i.id
        WHERE nodes_Tags.key='name'
        GROUP BY nodes_Tags.value
        ORDER BY num DESC
        LIMIT 5;
"Wells Fargo",57
"Bank of America",50
Chase,43
Citibank,27
"US Bank",14
```

List of Top 20 Amenities in San Francisco¶

```
sqlite> SELECT value, COUNT(*) as num
          FROM nodes_tags
          WHERE key='amenity'
          GROUP BY value
          ORDER BY num DESC
          LIMIT 20;
restaurant,2891
bench,1163
cafe,972
place_of_worship,700
post_box,684
school,590
fast_food,579
bicycle_parking,564
drinking_water,511
toilets,401
bank,369
bar,318
parking,276
fuel,265
car_sharing,225
waste_basket,211
atm,208
pub,201
post_office,162
pharmacy,151
```

Conclusion:¶

The San Francisco dataset is quite large and messy. It is clear that even though I made data cleanning, it's still not 100% clean. Since there are thousands of contributing users, so it is inevitable to have so many human input error. I'm thinking openstreetmaps could create a standard infomation format for users adding and updating information. When users views maps online, we could add an link to encourage users to find errors and report new locations. Users who did most update will be awarded a small gift every month. We also could build a standard information add-on screen and only let users just put in the detail information in each column, like location name, street name, Apt number,only 5-digit-zipcode, update reason etc. It may cost money to build up this small system, but it will save lots of time to do data wrangling in next step.