OpenStreetMap Data Case Study

Map area¶

San Jose, California - United States

http://www.openstreetmap.org/relation/112143
https://mapzen.com/data/metro-extracts/metro/san-jose_california/

This is a map of where I live, so I'm interested to see what information the database holds for this city and also contribute to its improvement on openstreetmap.org.

Problems Encountered in the Map

After downloading a small sample of the dataset, I noticed the following problems which I shall address in the following order -

1. Over abbreviated street names
2. Inconsisent postal code (95014-4641, 95014)
3. Incorrect street types

Over abbreviated street names

Using Python's regular expression, I audited for abbreviated street names such "Rd" for "Road" and "Blvd" for "Boulevard". I derived a small sample of the dataset to catch some of these over abbreviated street names and corrected them using the function below -

In [1]:

```
mapping = {
        "Ave":"Avenue",
        "Rd":"Road",
        "Blvd":"Boulevard",
        "court":"Court"
        }
 def update_name(name, mapping):
    m = street_type_re.search(name)
    x = m.group()
    if x in mapping.keys():
        name = name.strip(x)
        name = name + mapping[x]

    return name
```

Inconsistent postal codes

Some postal codes were represented by the 5 digit zipcode but some had a hyphen "-" followed by 4 digit extension number. I decided to strip the hyphen followed by the 4 digit and just use the 5 first digits in order to have more consistent queries.

The cleaning of postal codes was done by the function below -

In [4]:

```python
def update_postcode(postcode):
    split_post = postcode.split("-")
    return split_post[0]
```

Taking alook at the top 10 postal codes, looks like some postal codes are not San Jose but areas in cupertino, Sunnyvale etc which are areas adjacent to San Jose when grouped by this aggregator.
SELECT tags.value, COUNT*() as count*
FROM (SELECT FROM nodes_tags UNION ALL SELECT * FROM ways_tags) tags
WHERE tags.key='postcode' GROUP BY tags.value ORDER BY count DESC;

95014|659

95037|25

95070|18

94087|16

94086|15

95051|9

95129|9

95127|8

95035|7
6 out of the top 10 postal codes don't even belong to San Jose. These seem to be very high in number and so I looked at the cities -
SELECT tags.value, COUNT*() as count FROM (SELECT* FROM nodes_tags UNION ALL SELECT * FROM ways_tags) tags
WHERE tags.key LIKE '%city'
GROUP BY tags.value
ORDER BY count DESC;

Sunnyvale|237

San Jose|45

Morgan Hill|25

Santa Clara|20

```
Saratoga|17

San José|10

Los Gatos|8

Milpitas|6

1|5

Campbell|4

4|3

6|3

12|2

Cupertino|2

15|1

150000 gallon|1

35|1

474|1

50|1

8|1

9|1

cupertino|1

san jose|1
```

All these areas belong to the Santa Clara county and surrounding San Jose. It is possible that this map area comprises of surrounding areas of San Jose, especially around streets such as Stevens Creek Boulevard and El Camino Real which are very long and span multiple cities. This isn't incorrect but definitely unexpected and worth a mention.

Incorrect street types

A lot of streets are very unique in that they don't belong to a certain "type" of street. For example, Robles and Marino. They were revealed during the street type audit in the code below -

In [5]:

```
def audit_street_type(street_types, street_name):
    m = street_type_re.search(street_name)
    if m:
        street_type = m.group()

        street_types[street_type] += 1
```

Since they are few and far between, it best be left as is.

Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the SQL queries used to gather them, and some additional ideas about the data in context.

File sizes

ls -lash command on the terminal to get sizes in MB -

```
13M ways.csv

44M ways_nodes.csv

20M ways_tags.csv

344M san-jose_california.osm

24M mydb.db

132M nodes.csv

2.8M nodes_tags.csv
```

Number of nodes

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

110847
```

Number of ways

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

15121
```

Number of unique users

```
sqlite> SELECT COUNT(DISTINCT(e.uid)) FROM (SELECT uid FROM nodes UNION ALL
SELECT uid FROM ways) e;

744
```

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|19732

nmixter|19016

mk408|9832

Bike Mapper|5993

samely|5446

RichRico|5019

dannykath|4982

karitotp|4186

MustangBuyer|3572

Minh Nguyen|3176
```

Number of users having only 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) ;

214
```

Additional data exploration

Top 10 appearing amenities

```
sqlite> SELECT value, COUNT(*) as num FROM nodes_tags WHERE key='amenity' GROUP
BY value ORDER BY num DESC LIMIT 10;

restaurant|58

fast_food|27

bench|18

bicycle_parking|13

cafe|12

fuel|11
```

```
place_of_worship|11

school|9

parking_space|8

dentist|7
```

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;

mexican|7

japanese|4

pizza|3

sandwich|3

sushi|3

italian|2

american|1

barbecue|1

breakfast|1

chinese;vietnamese|1

french;vietnamese|1

greek|1

indian|1

korean|1

malaysian|1

mediterranean|1

regional|1

seafood|1

steak_house|1

thai|1
```

vietnamese|1

Contributor statistics

I see a lot of input to openstreetmaps coming from bots (especially woodpeck_fixbot) -

sqlite> select user,count(*) as num from (select user from nodes union all select user from ways) where user like "%bot%"group by user order by num desc; woodpeck_fixbot|15311 xybot|101 bot-mode|1 tippobot|1

However,

- The top contributor (andygol) has a contribution percentage of 15.6%
- The top 2 contributors (nmixter and andygol) combined have only a 30% contribution percentage.
- The top 10 contributors combined have a contribution percentage of 64.3%

It appears we have a fairly small number of contributors to the map.

sqlite> select count(distinct user) from (SELECT user FROM nodes UNION ALL SELECT user FROM ways); 1339

Given that more than half the contributions come from just 10 users, something like a badge/leaderboard situation can arise that helps everyone be aware of this scenario and encourage them to contribute. Also, more efficient bots can be created.

Google Map API and use of Pokemon Go can be some ways to improve the data set. Another suggestion would be to have some of the education system incorporate it as part of coursework and also have something like kaggle competitions for certain areas to get more users into improving the system.

Other thoughts

The San Jose area needs improvement but appears fairly cleaned for the purposes of this project. In the short term, utilizing google map API and pokemon go is I would guess to be a relatively feasible option to implement compared to incorporating this as a project in schools (several challenges arise in navigating through existing education system) which is more a long term recommendation for getting more users to contribute. Clearly varying levels of encouragement (competitions/badges etc) needs to be brought in to get the required input in improving the dataset.

References

1. https://discussions.udacity.com/t/wrangle-osm-data-project/291893/15
2. https://discussions.udacity.com/t/preparing-for-database-help-getting-started/202844/16
3. https://discussions.udacity.com/t/p3-project-combining-auditing-cleaning-and-csv-creation/231037/2
4. https://wiki.openstreetmap.org/wiki/Map_Features
5. https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md