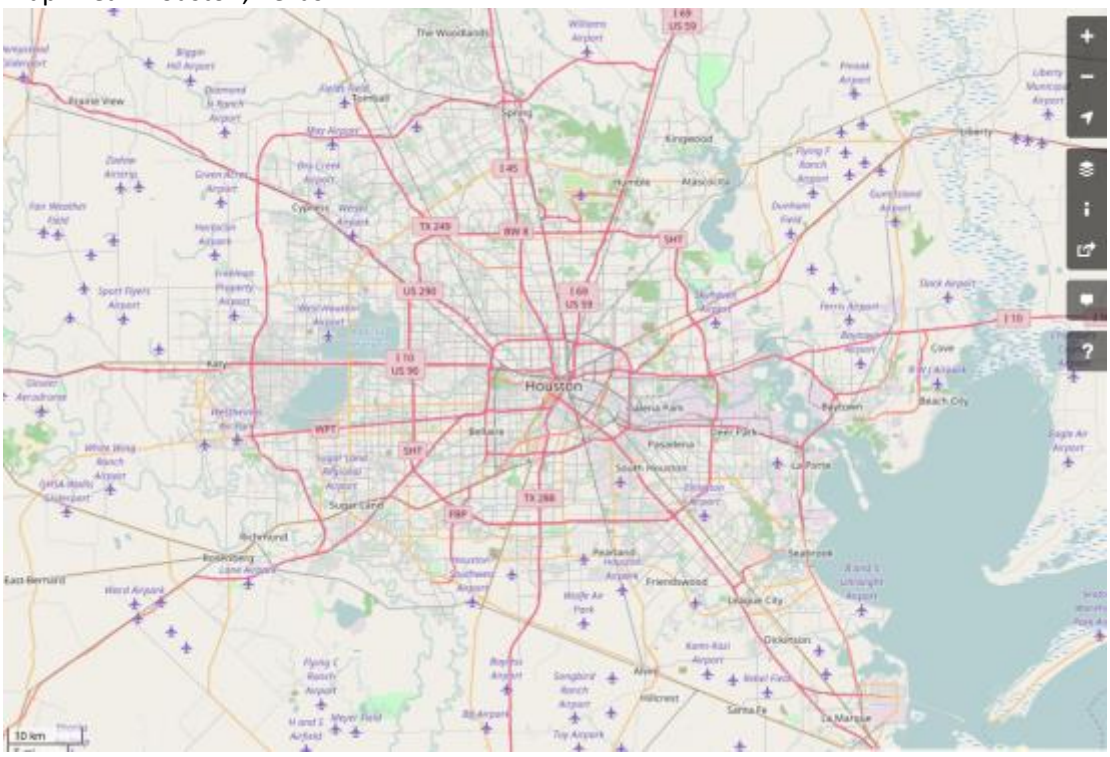
Data Wrangling with SQl

Name: Swetcha Karnati
Map Area: Houston, Texas.



Project Summary:

This project, I used data wrangling techniques to clean Houston, Texas map area from openstreetmap data. The reason to choose this city is because it is my current city and I got my master degree in this city. Cleaned data in XML is converted to CSV and imported into a SQL database.

DATA Audit

Tags:

parsing through the Houston dataset with ElementTree and count the unique element type by using count_tags function I got the number of unique tags as follows.

```
{'bounds': 1,
 'member': 27102,
 'nd': 3621611,
 'node': 3028397,
 'osm': 1,
```

'relation': 2462,
'tag': 2087706,
'way': 367024}

Tag Patterns:

Checking 'k' value for each tag. There are three regular expression, lower is for tags that contain only lowercase letters and are valid. lower_colon is for other valid tags with a colon in the value. problem is for tags with problematic characters. The results are as below:

{'lower': 892280, 'lower_colon': 1141158, 'other': 54265, 'problemchars': 3}

Problems in the DATA

After auditing the sample size of Houston map, I met one main problems with the data: Abbreviated street names

To correct the abbreviated street names following code is implemented, this code finds and corrects the abbrvaited street names. Functions used:

- audit_street_type: Checks if the street name is in the expected list.
- street_name: Check whether the attribute k = "addr:street".
- audit: Returns a list which match the previous two function conditions.
- update_name: Updates the old street name with a new one.

This update is done by the code in the audit.py file.

DATA OVERVIEW

The over view of the data is presented here including the file sizes and SQl quires used to gather some additional data.

File Sizes:

houston_texas.osm: 688.8 MB
nodes.csv: 252.3 MB
nodes_tags.csv: 6 MB
ways.csv: 21.8 MB
ways_tags.csv: 68.3 MB
ways_nodes.csv: 86.5 MB
houston_texas.db: 497.4 MB

Number of nodes:

sqlite> SELECT COUNT(*) FROM nodes;

3028397

Number Of ways:

```
sqlite> SELECT COUNT(*) FROM ways;
367024
```

Number of unique users:

```
SELECT COUNT(distinct(uid)) FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways)
```

Output: 1616

Top Contributing user:

```
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 1
```

Output: 'woodpeck_fixbot', 568987

Biggest religion:

```
SELECT nodes_tags.value, COUNT(*) as num FROM nodes_tags \
                JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value="place_of_worship") i \
                ON nodes_tags.id=i.id \
                WHERE nodes_tags.key="religion" \
                GROUP BY nodes_tags.value \
                ORDER BY num DESC\
                LIMIT 1';
```

Output:'christian', 2154

popular amenity:

```
SELECT value, COUNT(*) as num \
                FROM nodes_tags \
                WHERE key="amenity" \
                GROUP BY value \
                ORDER BY num DESC \
                LIMIT 1;
```

Output: 'place_of_worship', 2221

Conclusion:

After Auditing the data, it seemed clean except for small errors in street names like the short forms. As part of this project the data is further cleaned, but there are lots of improvements needed in this data set. The dataset contains very less amount of additional information such as amenities, tourist attractions etc. and it is also looked comparatively old than the recent map sources like google.

Suggestion:

- Maps might be more helpful for new visitors to the city if there is a “Attractions” tab in amenities.
- Solution to implement this could be importing from other sources.
- Potential problem is matching the existing data with new data.

References:

- http://puwenning.github.io/2016/02/10/P3-project-openstreetmap-data-case-study/