Project: Wrangle OpenStreetMap Data

Map Area

The area I have chosen for this project is New York, NY, United States.

- https://mapzen.com/data/metro-extracts/metro/new-york_new-york/
- https://www.openstreetmap.org/relation/175905

Problems in the Map

I noticed 3 problems during the auditing process. These are listed below along with solutions.

1. **Inconsistencies in street names**
   Example: Ave, avenue, Avenue
   Cir, Circle
   Cres, Crescent
   Parkway, Pkwy

   The sample osm file revealed abbreviated street names. The below helper function maps these abbreviated street names to the appropriate version, thereby creating consistency in street names.

```
def update_name(name, mapping):
    m = street_type_re.search(name)
    if m:
        street_type = m.group()
        if street_type in mapping.keys():
            name = re.sub(street_type, mapping[street_type], name) #substitute abbreviated street name with mapping[street name]
    return name
```

2. **Different representations of phone numbers**
   Example: (718)333-9850, +1-201-716-2827, 212.736.5000, 4147
   There was no consistency in the representation of phone numbers. The following function updates all phone numbers to one standard representation, i.e., (xxx)xxx-xxx.

```
def update_phone(phone):
    phone1 = re.sub(r'\D', "", phone)
    phone2 = re.match(r'^(\d{3})(\d{3})(\d{4})$', phone1) #updates the phone no representation
    if phone2:
        return "(" + phone2.group(1) + ") " + phone2.group(2) + "-" + phone2.group(3)
```

3. **Zip codes outside New York state**
   This SQL query revealed certain zip codes that were not within the state of New York. All New York zip codes begin with the number '1'.

```
SELECT A.value
FROM (SELECT * FROM nodes_tags
      UNION
      SELECT * FROM ways_tags) A
WHERE A.key='postcode' and value NOT LIKE '%1%'
GROUP BY A.value;
```

A part of the result is shown below:
06807
06820
06830
06850
06853
06854
06855
06870
06878
07002

Also, the query revealed some anomalies in the representation of zip codes. While some zip codes had a 4-digit extension, some had 2-letter state name prefixed with the zip code. For example,

08854-5603
08854-5622
08854-5627
08854-5659
08854-5695
08854-8000
08854-8002
08854-8003
08854-8004

NJ 07024
NJ 07086
NJ 07652
NJ 07747
nj 07652

A simple solution would be to ignore records not starting with 1 in the zip code.

Overview of the data

new-york.osm ..... 2.62 GB
newyork.db ..... 2.09 GB
nodes.csv ..... 990 MB
nodes_tags.csv ..... 27.6 MB
ways.csv ..... 113 MB
ways_tags.csv ..... 287 MB
ways_nodes.cv ..... 345 MB

Number of nodes
SELECT COUNT(*) FROM nodes;
Result: 11502185

Number of ways
SELECT COUNT(*) FROM ways;
Result: 1802724

Number of unique users
SELECT COUNT(DISTINCT(A.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) A;

Result: 4847

Tourism places in New York

```
SELECT A.value, COUNT(*) as num
FROM (SELECT * FROM nodes_tags
        UNION ALL
      SELECT * FROM ways_tags) A
WHERE A.key='tourism'
GROUP BY value
ORDER BY num DESC;
```

Result:

hotel|533

attraction|269

artwork|198

museum|196

picnic_site|158

viewpoint|152

information|79

camp_site|31

motel|30

guest_house|22

hostel|20

theme_park|16

gallery|12

zoo|12

yes|9

Hotel|5

chalet|5

caravan_site|3

local knowledge|3

aquarium|2

sightseeing|2

apartment|1

bed_and_breakfast|1

historic|1

picnic_area|1

Most visited restaurants

```
SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='restaurant') A
ON nodes_tags.id=A.id
WHERE nodes_tags.key='name'
GROUP BY nodes_tags.value
ORDER BY num DESC
LIMIT 5;
```

Result:

Applebee's|11

IHOP|10

Panera Bread|10

Bareburger|9

Chipotle|8

Additional ideas:

1. While auditing the restaurants in New York city, I noticed that some restaurants had too many details associated with them while some had few. For example, certain restaurants had name, detailed address, and cuisine associated with it. On the other hand, some restaurants just the name. It would be helpful if there was a standard way to provide details for any place entered in the open street map.

2. It would be helpful if there was a section for users to review a location and add their comments. Other users could upvote or downvote the review thereby preventing overly skewed reviews.

   Benefits:
   1. Provision to provide a review might lead to increase in user involvement.
   2. Consistency in the details provided by the users helps with the authenticity of data.

   Anticipated Problems:
   1. If too many details are expected from users, they may forego writing reviews. This may inadvertently draw negative opinion on some places due to lack of reviews.
   2. Generally, people have an inherent bias to voice out bad experiences than good experiences. There may be more bad reviews than good ones. The upvoting and downvoting would mitigate this.

Conclusion:

I could see three clear discrepancies in the open street map of New York area. A) Inconsistencies in street names, B) Multiple representations of phone numbers and C) Wrong state data. All the discrepancies can be corrected through the solutions described above. The data quality can be improved by standardizing and validating input. Additional features such as reviews can be added to maximize the experience of open street map.