Map Area

New York, NY, United States of America

http://www.openstreetmap.org/relation/175905

https://mapzen.com/data/metro-extracts/metro/new-york_new-york/

I chose to study New York because of its proximity to my house (roughly 1.5 hours via local bus). Additionally, I love the hustle and bustle of the city, the culture that different portions of the city offer, as well as the global importance of the "financial capital of the world". I hope that my contributions can help improve the OpenStreetMap (OSM) project.

Auditing Data

After downloading the data from MapZen, I wanted to see some of the unique tags used in a sample file of smaller size. A sample file is used instead of the original file because the large OSM data file would be difficult to process and would take a significant amount of time to run each script. I used the "mapparser.py" script to count the number of unique tags.

{'node': 1146732, 'relation': 925, 'way': 179638}

Tag Patterns

Before processing the data and adding it to a database, it is important to check the "k" value for each <tag> to see if there are any potential problems. The "tags.py" script was used to check for certain patterns in tags. Specifically, three regular expressions are used to examine the tags.

{'lower': 375607, valid tags that contain all lowercase letters

'lower_colon': 578015, tags that are valid except that they have a colon in their name

'problemchars': 2504, tags with problematic characters,

'other': 10637, tags that don't fit in the other three categories}

Problems Encountered in the Map

After downloading a small sample size of the New York OSM map, the sample file was run against the "data.py" script. This created five different CVS files: "nodes", "nodes_tags", "ways", "ways_nodes", and "ways_tags". Each CSV files corresponds to a table of the same name in the SQL database created by the "database.py" script.

Examining the CSV files and querying the SQL database tables revealed three main problems:

- Inconsistent street and avenue names
- Inconsistent telephone numbers
- Incorrect Zip Codes

Street Name:

Correcting street and avenue names was accomplished using regular expressions, as seen in "audit.py". The main issue was the inconsistency between different entries which led to multiple expressions meaning the same thing. For example, **West 84th Street** could be written as: **W. 84th Street**, **West 84th St.**, or **W. 84th St.** However, the "audit.py" script standardized the street and direction names so that it would be only be in the **West 84th Street** format.

Some examples changes to street and avenue names are shown in the table below (edited into table for convenience purposes).

Old	New
W. 84th Street	West 84th Street
Liberty St.	Liberty Street
Mill lane	Mill Lane
Bruckner Blvd.	Bruckner Boulevard
12th Avenue	Twelfth Avenue
6 Ave	Sixth Avenue

Phone Number:

All New York City phone numbers should have a country code of "+1" and an area code of "212" (United States Zip Codes). It is common for people to include parentheses for the area code, but that is not the convention for large metropolitan areas because the area code is required in what is referred to as ten-digit dialing (Wikipedia). Furthermore, for international callers, the country code of "+1" is needed. The code in "audit.py" was used to format the telephone numbers to include the country and area codes and to be in the following format: "+1 212-XXX-XXXX". See below for an example for some of the changes performed by the code.

Old	New
1 (212) 721-6500	+1 212-721-6500
12127216500	+1 212-721-6500
212-721-6500	+1 212-721-6500

Zip Code

The next type of data I was interested in examining was the zip code. I did this by running the SQL query below (also in "queries.py" script):

```
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
LIMIT 15;
```

The query resulted in the following table, with the zip codes in yellow indicating zip codes out of the NY city limits (edited for clarity):

Zip Code	Count
10314	2299
11234	1987
10312	1788
10306	1645
11236	1528
11385	1526
11746	1484
11706	1466
11743	1404
11757	1318
11229	1253
11203	1181
10469	1159
11208	1124
11207	1115

All the incorrect zip codes were similar so I wanted to take a count of the cities. I ran the following query:

```
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
LIMIT 15;
```

Which query outputted the following results, with the problem cities highlighted in yellow (edited for cla rity):

City	Count
West Babylon	1150
Commack	1148
West Islip	911
Brentwood	856
Huntington Station	820
Deer Park	817

Lindenhurt	805
East Northport	673
Dix Hills	635
Huntington	635
Bay Shore	618
Copiague	617
New York	556
North Babylon	511
Greenlawn	394

I was surprised that 14 or the top 15 cities were out of New York City. However, I noticed that all these cities were from Suffolk County, Long Island. Suffolk County is not one of the five counties, or boroughs, of NY city. The five boroughs of NY city are Manhattan, Bronx, Brooklyn, Queens, and Staten Island, which correspond to the **New York, Bronx, Kings, Queens, and Richmond counties**. I ran the following query to see what other counties were present in the dataset:

```
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags UNION ALL
SELECT * FROM ways_tags) tags
WHERE tags.key LIKE '%county'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 15;
```

Which produced the following results, with problem counties highlighted in yellow (edited for clarity):

County	State	Count
Nassau	NY	2545
Bergen	NJ	1600
Queens	NY	1379
Westchester	NY	1271
Middlesex	NJ	1243
Suffolk	NY	1036
Essex	NJ	876
Union	NJ	722
Morris	NJ	708
Kings	NY	669
Passaic	NJ	647
Monmouth	NJ	581
Bronx	NY	534
Richmond	NY	528
New York	NY	453

The five counties of NY city were all in the top 15, but none of them was the county with the most entries. Instead, three of the five counties had the least entries of the top 15 counties in the dataset. The errors likely occurred because of misunderstanding of what constitutes New York city. However, it

does come as a surprise that counties such as Monmouth, which is an hour and a half bus ride, (from my house to Port Authority, which is the main bus terminal of NYC), away, would be included in the dataset. These entries can be deleted from database using the **"DELETE FROM"** statement, but this could prove tedious (see more in Additional Ideas section).

Data Overview and Additional Ideas

File Sizes

New_York.osm……. 2,741,038 KB
Sample.osm……. 277,646 KB
New_York.db……. 141,588 KB
Nodes.csv……. 101,131 KB
Nodes_tags.csv……. 2,967 KB
Ways.csv……. 11,603 KB
Ways_tags.csv……. 29,406 KB
Ways_nodes.csv……. 35,233 KB

Number of Unique Users

Code:

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

Output:

2497

Number of Nodes

Code:

```
SELECT COUNT(*) FROM nodes
```

Output:

1,146,732

Number of Ways

Code:

```
SELECT COUNT(*) FROM ways
```

Output:

179638

Top 10 Contributing Users

Code:

```
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;
```

Output:

Username	Count
Rub21_nycbuildings	488738
ingalls_nycbuildings	93547
'MySuffolkNY'	62697
woodpeck_fixbot	62033
SuffolkNY	58047
minewman	49439
Northfork	41330
ediyes_nycbuildings	27258
lxbarth_nycbuildings	23378
smlevine	22029

Most Popular Cuisines

Code:

```
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;
```

Output:

Cuisine	Count
Italian	22
American	17
Pizza	14
Mexican	13
Indian	10
Chinese	8
Sushi	7
Japanese	6

Thai	6
Asia	4

Top 5 Places of Worship

Code:

```
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 5;
```

Output:

Place of Worship	Count
Christian	260
Jewish	23
Muslim	3
Buddhist	2
Hindu	1

Five Museums in NY

Code:

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

Output

Name
American Folk Art Museum
Bartow-Pell Mansion Museum
Dyckman Farmhouse Museum
Falaise Museum
Skyscraper Museum

Top Ten Amenities

Code:

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

Output:

Amenity	Count
Bicycle Parking	490
School	341
Places of Worship	296
Restaurant	285
Café	109
Fast Food	87
Bank	61
Fire Station	55
Bench	53
Bicycle Rental	45

Additional Ideas

There are several problems that I encountered while working with this dataset. One specific issue needing to be addressed is the data from outside of the New York City region. As mentioned above, several data points come from counties outside of NYC. OpenStreetMap should limit user inputs to the five NY city counties. Another issue, which is addressed above, is the lack of consistency for the format of the data. One solution is to only accept data inputs of a specific format. For example, phone numbers can only be in "+1 212-xxx-xxxx" format. Phone numbers written in a different way would not be accepted as part of the project. Stricter guidelines may, however, negatively impact users' willingness to contribute to the OSM project. Similarly, it may be difficult to decide on a set data format for certain values. For example, some people prefer to include phone number area codes in parentheses whereas others find it unnecessary. It is important to consider these drawbacks before any improvements are implemented. I hope to see the dataset continue to be cleaned and improved.

Resources:

https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md

https://github.com/davidventuri/udacity-dand/blob/master/p3/report.md

https://github.com/pratyush19/Udacity-Data-Analyst-Nanodegree/tree/master/P3-OpenStreetMap-Wrangling-with-SQL

http://www1.nyc.gov/nyc-resources/service/2123/new-york-city-counties

https://www.unitedstateszipcodes.org/

https://en.wikipedia.org/wiki/Ten-digit_dialing