Wrangle OpenStreetMap Data Project

Name: Ryan Helgeson
Map Area: I chose the metro area of Seattle, Washington. I chose this area because I have family members that live out there and have been thinking about moving there myself.

https://mapzen.com/data/metro-extracts/metro/seattle_washington/

Data Audit

Using the parse.py script I took a look at the types of tags in this dataset and how often they occur.

- ‘bounds': 1,
- 'member': 95773,
- 'nd': 8981937,
- 'node': 8023761,
- 'osm': 1,
- 'relation': 10396,
- 'tag': 4481271,
- 'way': 797329

Using the tags.py script I checked the patterns of the “k” value of each tag and put them in 4 categories: “lower” for tags containing only lowercase letters, “lower_colon” for tags with a colon in their names, “problemchars” for tags that contained problematic characters, and “other” for tags that don’t fall into the other categories.

- 'lower': 2360025
- 'lower_colon': 2051958
- 'problemchars': 0
- 'other': 69288

Problems Encountered

There were inconsistencies in street addresses that I encountered in the dataset. I used audit.py to identify inconsistent street names and update them to create a more consistent naming convention. I transferred this code into the data.py script so it cleans and updates the street names before converting the ism file into csv files and loads it into the SQL database. Here are some examples of inconsistent street names and their updated name.

Abbreviations:
- St --> Street
- Hwy --> Highway
- Ave --> Avenue

Capitalization:
- southeast --> Southeast
- west --> West

Punctionation:
- N. --> North
- S.E. --> SE
- SW, --> SW

Missing a Space:

- MainStreet --> Main Street

Another problem I encountered was that one or more nodes in the dataset did not have a 'user' element, throwing an error when I tried to run the data.py script. To deal with this I chose to remove those elements that were missing the 'user' element.

Overview of the data

File sizes:

- seattle_washington.osm: 1768.03004 MB
- project.db: 1271.783424 MB
- nodes.csv: 683.651264 MB
- nodes_tags.csv: 683.651264 MB
- ways.csv: 48.511062 MB
- ways_tags.csv: 122.519847 MB
- ways_nodes.csv: 212.85279 MB

Database queries:

Number of nodes

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

8023761

Number of ways

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

797329

Number of unique users

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

3677

Top contributing users

```
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;
```

```
Glassman|1306072
SeattleImport|735867
tylerritchie|655629
woodpeck_fixbot|567217
```

alester|367150
Omnific|366221
Glassman_Import|225779
CarniLvr79|211255
STBrenden|202918
Brad Meteor|195855

Number of Starbucks

```
SELECT COUNT(*) FROM nodes_tags WHERE value LIKE '%Starbucks%';
```

346

Top amenities

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

bench|3411
bicycle_parking|3408
restaurant|2971
waste_basket|1522
cafe|1449
fast_food|1232
school|837
parking|798
toilets|774
place_of_worship|735

Top fast food

```
select nodes_tags.value, count(*) as num from nodes_tags
join (select distinct(id) from nodes_tags where value='fast_food') i
on nodes_tags.id=i.id
where nodes_tags.key='name'
group by nodes_tags.value
order by num desc
limit 3;
```

Subway|218
McDonald's|59
Jack in the Box|49

Restaurants in each city

```
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 =
'city'
group by nodes_tags.value order by num desc LIMIT 8;
```

Seattle|812
Kirkland|47
Bellevue|29
Victoria|19
Lynnwood|18
Redmond|13
Bothell|11
Edmonds|11

Additional Ideas

This dataset is certainly not complete and would need some more user contributions to fill out the rest of the data for the Seattle metro area. For example in the dataset it says there is only 133 hotels:

```
select value, count(*) as num from nodes_tags
where nodes_tags.value = 'hotel';
```

hotel|133

But a quick search on hotels.com says there are over 400 hotels in Seattle. More information in the dataset would help tourists or people who live in Seattle to see what amenities are provided to them. This would just involve more people with the motivation to build out this dataset to completion.

More people contributing to this dataset would however lead to more data errors and inconsistency. So various data cleaning methods, like the one provided in this project for cleaning street names, would be needed to make sure that the data being entered is correct and consistent with the rest of the dataset.

Files

- seattle_washington.osm: Full dataset of the osm file
- seattle_sample.osm: sample data of the osm file
- audit.py: audits street names and updates them. Used for testing and inserted into data.py
- data.py: takes some code from audit.py to clean and update street names. Also builds the CSV files from the OSM file and shapes the data for SQL database insertion.
- database.py: creates the database and tables from the CSV files created in the data.py file
- parser.py: used to count how many of each tag there was
- file_size.py: prints out the file sizes of various files used in the project
- sample.py: created the seattle_sample.osm file
- schema.py: schema for the tables in the SQL database
- tags.py: Finds the pattern of each tag in the dataset and categorizes them

References

http://learnosm.org/en/osm-data/data-overview/

http://wiki.openstreetmap.org/wiki/OSM_XML

hotels.com

https://discussions.udacity.com/t/how-to-handle-missing-data-in-osm-file/242641/2

https://gist.github.com/carlward/54ec1c91b62a5f911c42