Project: JOCON37 OpenStreetMap Data Case Study

Denver, Colorado - United States

https://github.com/jocon37/OpenStreetMap-Data-Wrangling
(https://github.com/jocon37/OpenStreetMap-Data-Wrangling)

Introduction

I chose this region - city for two reasons. First, I had a difficult time being able to get the right size file to extract from OpenStreetMap so I took what would work, but my family use to have a small ranch in a city near Denver, so it is a place I am familiar with.

Problems Encountered in the Map

The data throughout the XML should truly follow specific formatting and should be consistent and easy to read. I noticed a huge difference between when I ran the Audit cleaning against the primary XML file and the Sample XML file. When I ran it against the original file I had a lot of clean ups, but in the sample file I only had 1.

- There are overabbreviated street names like "Hwy", "Ave", "St", "S", "Blvd", etc. For example, Blvd has 1720 entries, Ave has 7945, and St has 11,199 entries on the file but also has 'st'. These were spelled out to be West, South, Avenue, etc.
- Correction of misspellings. Example Boulavard ==> Boulevard
- Correction of words that have been combined incorrectly like MainStreet, which should be Main Street in the primary file.

Originally when I ran my sample.py and submitted my project I had an interesting encounter. The sample generated had no way tags in it, only node tags. It was still a large sample file at 1.84MB. But I couldn't do much research on it and I ended up doing all my analysis on the primary XML file so that I could have some decent fun. However, since I had recevied my submission back to make a few updates, I decided to make more than just a few. This sample map became a problem area so I decided to tweak my sample.py and recreate the whole sample.osm as well as recreate my audit.py to see if I got a better sample and result on the second go round and I did!

Auditing and Clearing:

Originally, the main function I used for clearning the street names is:

def audit_street_type(street_types, street_name): m = street_type_re.search(street_name) if m: street_type = m.group() if street_type not in expected: street_types[street_type].add(street_name)

However, it appeared that this did not seem to work as well as it should have. So I decided to be a little more specific in my cleaning process. The above code looks to find all the streets that need to be changed, but to fix them I added the following

def update_name(name, mapping): name_split = name.split(' ') name_new = '' for item in name_split: if item in mapping: item_new = mapping[item] name_new = name_new + item_new + ' ' else: name_new = name_new + item + ' ' name = name_new.strip(' ') return name

Overview of the data

I extracted the OSM.XML ZIP file and ran data.py and audit.py, I created the .csv files. I then used SQLITE to create a database and load the .csv files into tables so I could query the tables.

- denver_colorado_osm.xml 714MB
- nodes.csv 271MB
- nodes_tags.csv 16MB
- ways.csv 25MB
- ways_nodes.csv 86MB
- ways_tags.csv 45MB

With the new 2nd sample OSM file (denver_colorado.osm) I have a better file with fuller .csv files. Originally, the sample OSM had all three ways csvs empty save the header row. With this 2nd file I have some files with teeth to them, even if only a few.

- denver_colorado.osm 7.31MB
- nodes.csv 2.77MB
- nodes_tags.csv 168KB
- ways.csv 254KB
- ways_nodes 892KB
- ways_tags 465KB

So let's do some querying and look at the data in this new OSM file. I'm creating my tables in SQLITE3 where I will be doing my analysis. What's really nice about SQLITE is that when you import files it will automatically create the tables for you.

To get an idea of the number of unique users in the ways.csv file we can look at the disntinct uid variable:

```
sqlite> select count(distinct(a.uid))
   ...> from(select uid from nodes union select uid from ways) as a;
543
sqlite>
```

There are 543 unique users in this section of sample file for Denver, Colorado.

We can look at the basic count of nodes and ways:

```
sqlite> select count(*)
   ...> from nodes;
62646
sqlite> select count(*)
   ...> from ways;
7920
sqlite>
```

Querying the nodes table returns a count of 62,646 and the ways table returns a count of 7,920.

We can look at various types of nodes out of the nodes_tags table by looking for different items. For example, let's look at how many BBQ places there are?

```
sqlite> select  count(*)
   ...> from nodes_tags
   ...> where key = 'amenity' and value = 'bbq';
0
sqlite>
```

It found none, but what if you want to know what your options are?

You can run a query to look at the distinct types of values within the key of amenity to see what options you have

```
sqlite> select distinct value
   ...> from nodes_tags
   ...> where key = 'amenity';
school
place_of_worship
fire_station
library
fast_food
restaurant
bicycle_parking
pub
bicycle_rental
cafe
bank
shelter
bar
fountain
parking
pharmacy
toilets
bench
post_box
swimming_pool
```

What about bicycle_parking? How many are in the area? Toilets? Swimming_pools?

```
sqlite> Select count(*)
   ...> from nodes_tags
   ...> where key = 'bicycle_parking';
0
sqlite> select count(*)
   ...> from nodes_tags
   ...> where key = 'toilets';
0
sqlite> select count(*)
   ...> from nodes_tags
   ...> where key = 'swimming_pool';
0
sqlite>
```

Wouldn't you know, 0 return on all three. That doesn't mean they don't exist, just don't exist in the sample. They are likely in the larger XML file.

We can look at the top 10 categories of shops through queries as well:

```
sqlite> select count(*), value
   ...> from nodes_tags
   ...> where key='shop'
   ...> group by value
   ...> order by count(*) desc
   ...> limit 10;
7,car_repair
4,alcohol
3,convenience
3,doityourself
3,supermarket
2,car
2,department_store
2,yes
1,antiques
1,bakery
sqlite>
```

- Car_repair 7
- Alcohol..... 4
- Convenience..... 3
- Doityourself 3
- Supermarket 3
- Car..... 2
- Department_store. 2
- Yes 2
- Antiques 1
- Bakery 1

The "Yes" shop means that the shop is unidentified or unspecified. The Doityourself shops include stores such as Lowes, HomeDepot and even some Self Storage.

It is interesting and rather humorous to see that there are more alcohol shops than supermarkets. Not sure what that says about us as a society. I can't help but laugh a little to my self and wonder if it is like that in my area as well. I also did look up the marijuana shops, out of curiosity, since it is legal in Colorado. I have to admit, I was half expecting it to be in the top 10 list (I say half humorly), however, I was super surprised to see there were none. I can only think that some of those shops are listed under "yes" - the unspecifed shops.

If you want to know what do in and around Denver, Colorado? We can look up tourism in the tables with key = "tourism".

```
sqlite> select distinct(value)
   ...> from nodes_tags
   ...> where key = 'tourism';
museum
guest_house
information
picnic_site
viewpoint
hotel
attraction
motel
artwork
trail_riding_station
camp_site
caravan_site
gallery
hostel
apartment
yes
sqlite>
```

You can look for items like Museum, Picnic_site, Viewpoint, Hotel, Artwork, Trail_riding_station, Camp_site and more. There is a lot of options available to look for.

Other Information

It might be helpful, as a suggestion, for OneStreetMap, to provide some guidelines or manuals for mappers to have to follow for when they are mapping. That way, they would have consistency when mappers are coding addresses and postal codes and it might cause less confusion.

It would be easier to read and be easier to search.

The challenge would be getting eveyrone to read the guidelines, plus people have their own preferences and style for doing things.

Starting with a group of top mappers to test the idea of set guidelines and implementing it. Sometimes getting the popular group doing something the rest will follow.

Make a game out of it by creating a rank system out of the guidelines.

Create a peer review system.

If we wanted to, we could look at the data to see who the top contributing users are to OneStreetMap with our queries.

```
sqlite> select a.user, count(*) as num
   ...> from (select user from nodes union all select user from ways) as a
   ...> group by a.user
   ...> order by num desc
   ...> limit 10;
"",35283
chachafish,12747
"Your Village Maps",7626
CornCO,1658
GPS_dr,1361
woodpeck_fixbot,1180
Stevestr,975
DavidJDBA,819
"RustProof Labs",598
EnigmaQuip,403
sqlite>
```

- "" 35283
- chachafish 12747
- "Your Village Maps" . 7626
- CornCO 1658
- GPS_dr 1361
- woodpeck_fixbot 1180
- Stevestr 975
- DavidJDBA 819
- "RustProof Labs". 598
- EnigmaQuip 403

I tried to look up who "" might be, if it was a person or a bot and couldn't find anything, but they sure seem to be a top editor. In doing some research it looks like chachafish is an actual person. They have quite a bit of activity in OneStreetMap's forum (https://forum.openstreetmap.org/viewforum.php?id=67) (https://forum.openstreetmap.org/viewforum.php?id=67)). "Your Village Maps" is also a very active individual as is CornCo and GPS_dr.

Approaching these individuals through the forums, creating a game type ranking system to promote the new guidelines would be a great start.