OpenStreetMap Data Case Study

Map Area

Chattanooga, TN

- https://www.openstreetmap.org/export#map=13/35.0494/-85.2282

This map is the closest city to where I live that is large enough to meet the 50MB requirement. My home is about 30 minutes northeast of Chattanooga.

Problems Encountered in the Map Data

Potentially any human added data area has the potential for needing cleaning. With so many areas of possible work needing to be done I decided to focus on the two areas below.

- Street Types—addr:street
  As expected there are many different variations of a street type. For example a street type can be "Street", "St.", "street", "st", etc.
- City Name—addr:city
  There are also variations of the city name. Some examples are the first letter capitalized, entire city name in lower case, abbreviations used, etc.

Street Type Cleanup

After auditing the file for street names it became apparent that a lot of cleanup would be needed. Using the code below provided by the course I discovered that there was quite a bit of work needed. Four variations of the type "Avenue". Three variations of the type "Street". Five variations of the type "Drive". A total of 47 different street types were found with many variations of commonly used types. The challenge is to determine which street types are acceptable (Loop, South, etc.) and which would need to be cleaned (Dr, Hwy, etc.).

```
def print_sorted_dict(d):
    keys = d.keys()
    keys = sorted(keys, key=lambda s: s.lower())
    for k in keys:
        v = d[k]
        print("%s: %d" % (k, v))

def is_street_name(elem):
    return (elem.tag == "tag") and (elem.attrib['k'] == "addr:street")

def audit():
    for event, elem in ET.iterparse(osm_file):
        if is_street_name(elem):
            audit_street_type(street_types, elem.attrib['v'])
    print_sorted_dict(street_types)
```

#102
58
A
ave
Ave
ave.
Avenue
Blvd
Boulevard
Cir
Circle
Court
Ct
Dr
dr.
Dr.
Drive
drive
East
Hartman
Highway
Hwy
Lane
Lilac
Ln
Loop
N
North
Parkway
Pike
Place
Rd
rivermont
road
road
School
South
st
St
Street
Strikers
Terrace
Trail
View
vine
Way
West

The following function was used to update the street types:

```
def update_street_type(name):
    for old_type in mapping:
        if name.endswith(old_type):
            new_type = mapping[old_type]
            name = name[:(len(name) - len(old_type))] + new_type
    return name
```

City Name Cleanup

Auditing the city name showed there was work that needed to be done there as well. Like many larger cities Chattanooga has many areas that are considered to be suburbs of the city. Chattanooga is no exception. I discovered 19 different entries for the city name including six variations of the name “Chattanooga” alone. Abbreviations, misspellings and missing capitalization dominated the results. Luckily all the names below are legitimate areas of Chattanooga or derivatives of the city name itself.

```
def get_city_name():
    for element in get_element(OSM_FILE):
        if element.tag == "node" or element.tag == "way":
            for tag in element.iter("tag"):
                if is_city(tag):
                    if tag.attrib['v'] not in expectedcities:
```

```
                cities.add(tag.attrib['v'])
    pprint.pprint(cities);
```

```
Brainiard              Lookout Mountain
Ch                     Ooltewah
Chatanooga             Red Bank
Chattanooga, TN        Red bank
Chattanoooga           brainerd
Chattaoooga            east Ridge
chattanooga            hixson
East Ridge             ooltewah
Harrison               red bank
                       redbank
```

The following function was used to clean the city names:

```
def update_city_name(name):
    if name not in expected:
        for old_name in mapping:
            if name == old_name:
                name = mapping[old_name]
    return name;
```

Data Overview

File Sizes

Chattanooga_OSM.osm	62.465 MB
Nodes	23.675 MB
Nodes_Tags	0.465 MB
Ways	2.118 MB
Ways_Nodes	8.369 MB
Ways_Tags	3.918 MB

Number of Nodes

```
SELECT COUNT(*) FROM [Nodes];
```

285446

Number of Ways

```
SELECT COUNT(*) FROM [Ways];
```

35247

Number of Unique Users

```
SELECT    COUNT(DISTINCT(a.[uid]))
FROM      (SELECT [uid] FROM [Nodes]
          UNION ALL
```

```
        SELECT [uid] FROM [Ways]
        ) a;
```

674

Top 10 Contributing Users

```
SELECT  Top 10 a.[user], COUNT(*) Count
FROM    (SELECT [uid] FROM [Nodes]
         UNION ALL
        SELECT [uid] FROM [Ways]
        ) a;
```

user	**count**
rjhale1971	110402
T_9er	25293
ELadner	14050
Thad C	12793
booc0mtaco	7737
bobby22	6480
Your Village Maps	4570
maxerickson	4049
Leah	3774
chattGamer	3665

I suspect with 110,402 entries rjhale1971 may be a bot. That is an incredible number of entries.

Number of Users Having Only 1 Post

```
SELECT  COUNT(b.*)
FROM    (
         SELECT a.[user], COUNT(*) Number
         FROM    (
                 SELECT [user]
                 FROM    [Nodes]
                 UNION ALL
                 SELECT [user]
                 FROM [Ways]
                 ) a
        GROUP BY [user]
        HAVING COUNT(*) = 1
        ) b
```

74

Additional Data Exploration

Top 10 Amenities

```
SELECT  [Value], COUNT([Value]) Number
FROM    [Nodes_Tags]
WHERE [Key] = ‘amenity’
GROUP BY [Value]
```

```
ORDER BY Number DESC
```

```
place_of_worship   193
parking            129
restaurant          84
school              60
bench               35
fast_food           33
fuel                29
bicycle_rental      27
grave_yard          24
waste_basket        24
```

Being located firmly within the Bible Belt it was no surprise that places of worship was the top spot at 193. What was surprising is that waste_basket hit the top ten. I was not aware that finding locations to dispose of trash was such a high priority.

Number of Unique Restaurants

```
SELECT   COUNT(DISTINCT [Value])
FROM     [Nodes_Tags]
WHERE    [id] IN (
                   SELECT DISTINCT [id]
                   FROM [Nodes_Tags]
                   WHERE [Value] = ‘restaurant’
               )
AND      [key] = ‘name’;
```

```
79
```

Number of Unique Religions

```
SELECT   DISTINCT [Value]
FROM     [Nodes_Tags]
WHERE    [id] IN (
                   SELECT DISTINCT [id]
                   FROM [Nodes_Tags]
                   WHERE [Value] = ‘place_of_worship’
                   )
AND      [Key] = ‘religion’;
```

```
1        Christian
```

This result was both unsurprising and simultaneously disappointing. The one religion listed was ‘Christian’. I know from my travels in Chattanooga that there is a diverse religious community of many faiths. I was disappointed that none of them had been added.

Additional Ideas

After auditing the data I realized that everywhere data is entered by humans has the potential to be disorganized, incorrect or misleading. I believe that OSM would greatly benefit by possibly adding drop down menus in the data entry section. For example it may not be feasible to try and account for every street type but a drop down menu with the most common examples would help greatly. The user could still be allowed to enter the street type manually if the menu did not cover their specific case. City names could also benefit from this when keyed on the zip code entered by the user.

Another area of improvement would be to highlight areas that are missing data or don't fully represent the area. I am reminded of the unique religions above. My area of study has more religions than Christianity and if that discrepancy were somehow highlighted users could focus in on that area. This would provide a better overall picture and improve user experience.

Conclusion

After auditing the data I was able to clean the street types and city names to make them more presentable. I was able to create files to upload to a database to find statistical data about the extract. I discovered there are areas of improvement for the Chattanooga extract. More data fields could use with cleaning. If the improvements mentioned above were to be instituted the data in the extracts would be more representative of the area and in better condition.