Data_wrag

November 1, 2017

1 OpenStreetMap Data Case Study

1.0.1 Map Area

Map area comprises of two locations.

1) Blacksburg, Virginia, United States

2) Christanburg, Virginia, United States

Blacksburg is the college town in Virginia,US where Virgina Tech is located. I am currently enrolled in the school and want to explore the area. Christanburg is one of the relatively bigger city near Blacksburg I frequently visit.

1.0.2 Problems Encounterted in the Map

After initially downloading the sample file and analyzing it, I could find the following problems,

1) Inconsistent Street Names

2) Inconsistent Postal Codes

Inconsistent Street Names The sample data was manually explored to analyze the street names and inconsistencies were visibile. Further, all the street names were extracted. Accordingly, a list of expected streen names was created. Also a mapping dictionary was created to map a inconsistent street name with a correct one. Code below analyzes street name to check if the name is as per expected list. If no, it changes it to expected list as per mapping dictionary.

Inconsistent Postal Codes Since the area does not have many postal codes, it was easy to identify the postal codes. Following distict postal codes are present in the data : '24060-3348', 'VA 24060', '24061-9517'. We can see that they are not consistent with each other. The code below corrects the inconsistencies in the postal code as well.

```
In [4]: #List of the expected street names
        expected = ["Street", "Avenue", "Boulevard", "Drive",
                    "Court", "Place", "Square", "Lane", "Road",
                    "Trail", "Parkway", "Commons","Northeast",
                    "Run","Plaza","Terrace","Southeast","Northwest",
                    "Southwest","Circle","North",
                   "Crossing","Way","South","Pass"]

        #List of expected postal codes
        codes = ["24060" , "24061", "24062", "24063","24068","24073"]

        #Mapping to correct incorrect street names to correct ones.
        mapping = { "St": "Street",
                    "St.": "Street",
                    "Ave": "Avenue",
                    "Rd.": "Road",
                    "Rd": "Road",
                   "Blvd": "Boulevard",
                   "dorm": "dorm",
                   "Dr.": "Drive"

                    }

        def update_name(name, mapping):

            temp = name.split(" ")
            temp1 = len(temp)
            last = temp[-1]
            rem = temp[0:(len(temp)-1)]
            key = last

            if key in mapping: # Helps to ensure that we have a systematic error at hand
                name = ' '.join(rem) + " " + mapping[key]
            return name


        def update_postals():
            postals = audit(OSMFILE)[1]

            for i in range(0,len(postals)):
                if "-" in postals[i]:
                    postals[i]=postals[i].split("-")[0]
                elif " " in postals[i]:
                    postals[i]=postals[i].split(" ")[1]
```

1.0.3 Data Overview and Addition Ideas

1.0.4 File Sizes

1) Blacksburg.OSM.....................58.5 mb

2) mydb...............................4.19 mb

3) nodes.csv..........................447 KB

4) nodes_tags.csv......................111 KB
5) ways.csv............................26 KB
6) ways_nodes.csv......................134 KB
7) ways_tags.csv.......................53 KB

```
In [5]: import sqlite3
        import csv
        from pprint import pprint

        sqlite_file = 'mydb.db'

        # Connect to the database
        conn = sqlite3.connect(sqlite_file)
        conn.text_factory = lambda x: unicode(x, 'utf-8', 'ignore')

        # Get a cursor object
        cur = conn.cursor()
```

1.0.5 Number of Nodes

```
In [6]: cur.execute("SELECT COUNT(*) FROM nodes;")
        pprint("Number of nodes")
        pprint(cur.fetchall())
```

```
'Number of nodes'
[(32880,)]
```

1.0.6 Number of Ways

```
In [7]: cur.execute("SELECT COUNT(*) FROM ways;")
        pprint("Number of ways")
        pprint(cur.fetchall())
```

```
'Number of ways'
[(2646,)]
```

1.0.7 Number of Unique Users

```
In [8]: cur.execute("SELECT COUNT(DISTINCT(e.uid)) \
        FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;")
        pprint("Number of unique users")
        pprint(cur.fetchall())
```

```
'Number of unique users'
[(93,)]
```

1.0.8 Top 10 contributing users

```
In [9]: cur.execute("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")
        pprint("Top 10 contributing users")
        pprint(cur.fetchall())
```

```
'Top 10 contributing users'
[(u'mutantmonkey', 18984),
 (u'jumbanho', 6756),
 (u'Evanator', 2832),
 (u'woodpeck_fixbot', 1770),
 (u'dcat', 636),
 (u'Spesh', 552),
 (u'Jonah Adkins', 540),
 (u'Chris Lawrence', 384),
 (u'jbvejle', 336),
 (u'Dayton_Poff', 276)]
```

1.0.9 Top 10 common amenities in the area

```
In [10]: cur.execute("SELECT value, COUNT(*) as num \
         FROM nodes_tags \
         WHERE key='amenity' \
         GROUP BY value \
         ORDER BY num DESC \
         LIMIT 10;")
         pprint("Top 10 common amenities in the area")
         pprint(cur.fetchall())
```

```
'Top 10 common amenities in the area'
[(u'bicycle_parking', 42),
 (u'bench', 28),
 (u'place_of_worship', 21),
 (u'recycling', 7),
 (u'restaurant', 7),
 (u'school', 7)]
```

1.0.10 Common Sports played in the area

```
In [12]: cur.execute("SELECT value \
         FROM nodes_tags \
         WHERE key='sport';")
         pprint(cur.fetchall())
```

```
[(u'disc_golf',),
 (u'disc_golf',),
 (u'disc_golf',),
 (u'disc_golf',),
 (u'disc_golf',),
 (u'disc_golf',),
 (u'disc_golf',)]
```

1.0.11 Scope of improvement

It is important to note that openstreetmap data for my area isn't complete and accurate enough. For instance, there are many other sports that is played which I could not find when I was querying for sports. Similarly, area does have many restaurants and cafes. However, I could not find them as well. I think additing more of this data to the dataset would really make this dataset information rich.

One of the unique ways to improve the dataset particularly for this location is to engage the student community that is available in the Blacksburg town due to presence of Virginia Tech. There are students who have required skills to perform data improvement activities. Students can not just be encouraged but awared points for taking up this task in the form of extra credit or in fact this exercise can be the part of the coursework in one or more of the relevant classes.

1.0.12 Benefits and Anticipated problems

Primary benefit of using student community to improve data quality is that it will be a continuous process. It couple of semester time we can significantly improve data quality. Secondly, so many students will get the feel of real data rather than using canned data.

One of the problem is the struggle that students might have to go through to work with real data which is not easy to handle and manipulate. Secondly, this task would be a part of some project work and students are still learning the ropes, there won't be standardization in inputs. While instructions can be given to make the data standardised, validation is really not possible for so many students. Thus, final data quality might require further iterative improvements.

1.0.13 Conclusion

This was by first attempt at data wrangling. I chose a very small data file (50MB) so that it is easier to handle the data. For the purpose of exercise, it was a good experience to clean data using some form of program. It would be interesting to explore how one can handle non-systematic problems in the data. I understand that the result of the project does not produce the tidiest data one can get. I would want to work on identifyting more problems and cleaning them using a larger data file.