OpenStreetMap Sample Project

Data Wrangling with MongoDB

*Matthew DeLaney*

Map Area: Chattanooga, TN, United States

[*https://www.openstreetmap.org/relation/197273*](https://www.openstreetmap.org/relation/197273)

[*http://metro.teczno.com/#chattanooga*](http://metro.teczno.com/#chattanooga)

1. Problems Encountered in the Map

Over-­abbreviated Street Names

Postal Codes

2. Data Overview

3. Additional Ideas

Contributor statistics and gamification suggestion

Additional data exploration using MongoDB

Conclusion

# **1. Problems Encountered in the Map**

After downloading the Chattanooga osm and running it through the auditor I noticed 4 main problems

Many and inconsistent directional street names (E 3rd Street, Hermitage Avenue North)

* Street Numbers in the Street Name (5728 Tennessee 58, 5420 Brainerd Rd)
* Misspellings/Missing Suffix (Singal instead of Signal, Cone missing the Suffix (lane)
* Post Codes were not all from Chattanooga (30719,30720)

## **Inconsistent Directional Names**



Once the data was imported to MongoDB, some basic querying revealed that there was no consistent directional prefixes or suffixes. I edited the strings to make it so the full word was used at the beginning of the street name.

## **Street Numbers in the Street Name**

## **Post Code Errors**

Post Codes were mostly correct. There was one that had a state prefix and one that was too short. Quick transformations corrected the state prefix, and dropped the postal code that was too short when I realized there wasnt a code it was similar to.

Some postal codes looked very different that the majority of codes.

# Sort postcodes by count, descending

db.char.aggregate([{"$match":{"address.postcode":{"$exists":1}}}, {"$group":{"\_id":"$address.postcode", "count":{"$sum":1}}}, {"$sort":{"count":­1}}])

Here are the top two results, beginning with the highest count:

{u'\_id': u'37415', u'count': 633},

{u'\_id': u'37363', u'count': 283},

The vast majority of the codes were in Chattanooga, but there were definitely some from outside. The assumption is that they were likely from Georgia since Chattanooga was on the border. After checking the data online the data was allowed to stay.

# **2. Data Overview**

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

File sizes

chattanooga.osm ....... 117 MB

chattanooga.osm.json .. 135 MB

# Number of documents

> db.Chat.find().count()

636430

# Number of nodes

> db.Chat.find({"type":"node"}).count()

579301

# Number of ways

> db.Chat.find({"type":"way"}).count()

57065

# Number of unique users

> len(db.Chat.distinct("created.user"))

622

# Top 3 contributing users

> db.Chat.aggregate([{"$group":{"\_id":"$created.user", "count":{"$sum":1}}}, {"$sort":{"count": -1 }}, {"$limit":3}])

[{u'\_id': u'rjhale1971', u'count': 202349},

{u'\_id': u'T\_9er', u'count': 52777},

{u'\_id': u'woodpeck\_fixbot', u'count': 38683}]

# Number of users appearing only once (having 1 post)

> db.udacity.aggregate([  
 {"$group":{"\_id":"$created.user", "count":{"$sum":1}}},  
 {"$group":{"\_id":"$count", "num\_users":{"$sum":1}}},  
 {"$sort":{"\_id":1}},  
 {"$limit":1}  
])

[ {"\_id":1,"num\_users":64} ]

# “\_id” represents postcount

# **3. Additional Ideas**

**Joining additional databases**

**I think it would be interesting to join a accident and crime report database to this mapping database and see if there is any sort of correlation we can find between types of amenities and crime rate. Maybe graveyards are high crime, and areas with benches are low crime. I think a clustering analysis would be very interesting in finding out more about the data.**

**There are a few issues that could arise. First, it is unlikely that accidents and crime reports are given the street address where they are found. We would most likely have to come up with a way to write the street address. Possibly the middle of the street where it occurred and then use a map API to get the latitudes and longitudes. I think the map could provide some interesting benefits as well. If you were able to get an idea of crime rates and accidents you could get better deals on real estate, and also better allocate police patrols, or change speeds and zoning.**

## **Additional data exploration using MongoDB queries**

# Top 10 appearing amenities

> db.udacity.aggregate([  
 {"$match":{"amenity":{"$exists":1}}},  
 {"$group":{"\_id":"$amenity", "count":{"$sum":1}}},  
 {"$sort":{"count": -1}},  
 {"$limit":10}  
])

[{u'\_id': u'parking', u'count': 1244},

{u'\_id': u'place\_of\_worship', u'count': 541},

{u'\_id': u'school', u'count': 259},

{u'\_id': u'grave\_yard', u'count': 225},

{u'\_id': u'restaurant', u'count': 188},

{u'\_id': u'swimming\_pool', u'count': 165},

{u'\_id': u'fast\_food', u'count': 86},

{u'\_id': u'shop', u'count': 58},

{u'\_id': u'bench', u'count': 55},

{u'\_id': u'fuel', u'count': 54}]

# Biggest religion (no surprise here)

> db.udacity.aggregate([  
 {"$match":{"amenity":{"$exists":1}, "amenity":"place\_of\_worship"}},  
 {"$group":{"\_id":"$religion", "count":{"$sum":1}}},  
 {"$sort":{"count": -­1}},  
 {"$limit":1}  
])

[ {"\_id":"christian","count":507} ]

# Most popular cuisines

> db.udacity.aggregate([  
 {"$match":{"amenity":{"$exists":1}, "amenity":"restaurant"}},  
 {"$group":{"\_id":"$cuisine", "count":{"$sum":1}}},  
 {"$sort":{"count":-1}},  
 {"$limit":3}

[{u'\_id': None, u'count': 148},

{u'\_id': u'mexican', u'count': 5},

{u'\_id': u'american', u'count': 4}]

## **Conclusion**

After this review of the data it’s obvious that the Chattanooga area is incomplete, though I believe it has been well cleaned for the purposes of this exercise. i think an interesting thing to note is that parking is the top tag for amenities. This makes me think parking is a real issue. I also am curious on why there are more grave yards than restaurants. Likely many small plots. I think more work can be done with this site and will likely contribute in the future.