Project 3 Data Wrangling

## 1. Problems Encountered in the Map

After initially downloading a small sample size of the Charlotte area and running it against a provisional data.py file, I noticed three main problems with the data, which I will discuss in the following order:

1. Street name end with letter but not its type(From 'Avenue A' to 'Avenue Z')
2. Street name with different case for same type ( 'AVENUE':"Avenue" )
3. “Incorrect” postal codes('NY 10533')

### Street name case and ending problem

I find the problems with function “audit\_street\_type” and “audit”. I fixed these problems with “update\_name” and “output\_after\_mapping\_name”.

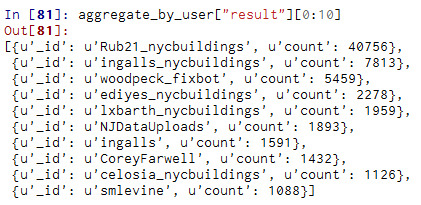
### Postal Codes problem

I find the problems with function “audit\_postcode”. I fixed these problems with “output\_after\_update\_postcode”.

# Sort users by count, descending

aggregate\_by\_user=db.nodes.aggregate([{"$group":{"\_id":"$created.user","count":{"$sum":1}}},{"$sort":{"count":-1}}])

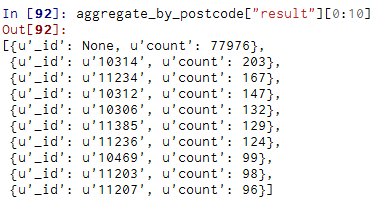
result:



This result shows that almost half of the nodes and ways are created by user “Rub21\_nycbuildings”

# Sort postcode by count, descending

aggregate\_by\_postcode=db.nodes.aggregate([{"$group":{"\_id":"$address.postcode","count":{"$sum":1}}},{"$sort":{"count":-1}}])

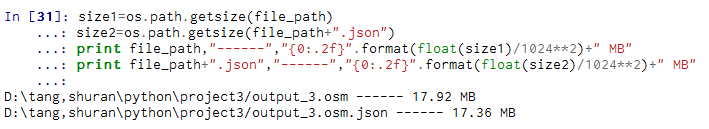


This result showe that, almost 90% documents do not have a postcode.

## 2. Data Overview

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

### File size



# Number of documents

> db.map\_collection.find().count()

# Number of nodes

> db.map\_collection.find({"type":"node"}).count()

1471349

# Number of ways

> db.map\_collection.find({"type":"way"}).count()

84502

# Number of unique users

> db.map\_collection.distinct({"created.user"}).length

336

# Top 1 contributing user

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

[ { "\_id" : "jumbanho", "count" : 823324 } ]

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

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

[ {"\_id":1,"num\_users":56} ]

## 3. Additional Ideas

### Additional data exploration using MongoDB queries

Sorted\_by\_city=db.map\_collection.aggregate([{"$match":{"address.city":{"$exists":1}}},{"$group":{"\_id":"$address.city", "count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":10}])

'N': set(['Avenue N', 'Horace Harding Expressway Sr N']),

'W': set(['Avenue W', 'Tarrytown Road Suite 1 W']),

find\_addr\_keys