OpenStreetMap Project
Data Wrangling with MongoDB

Miri Cho

Map Area: Atlanta, GA, USA
https://mapzen.com/data/metro-extracts/#atlanta

1. Problems encountered in your map

1) 'k' attribute cleanup:

Problem: Parent level tags (node, way, relation) have a child level tag named "tag" which has 'k' and 'v' as attributes. These 'k' and 'v' attributes contain a lot of information but there are too many unique values.

```
node:  ['changeset', 'uid', 'timestamp', 'lon', 'version', 'user', 'lat', 'id']
        tag: ['k', 'v']
        k_attrib unique values:  424

relation:  ['changeset', 'uid', 'timestamp', 'version', 'user', 'id']
        member: ['role', 'ref', 'type']
        tag: ['k', 'v']
        k_attrib unique values:  465

way:  ['changeset', 'uid', 'timestamp', 'version', 'user', 'id']
        nd: ['ref']
        tag: ['k', 'v']
        k_attrib unique values:  621
```

Solution: Just like the way we handled “addr" in lesson 6, k attributes that have at least one colon(:) were put under a separate dictionary within a dictionary.

Example:

```
'NHD:ComID': '41260457',
'NHD:FCode': '46006',
'NHD:FType': 'StreamRiver',
'NHD:way_id': '41260457',

changed to:

'NHD': {'ComID': '41262231',
       'FCode': '46003',
       'FType': 'StreamRiver',
       'way_id': '41262231'},
```

2) Improving Street names

Problem: The street names were inconsistent and also there were too many unique types of streets (rd, dr, ln, etc)

Solution: While there are over 70 different street name types, I decided to use the 'mapping dictionary' to capture the common ones to replace. I modified the code from Lesson 6 to make the changes.

```
mapping = { "st": "street",
          "st.": "street",
          "rd": "road",
          "rd.":"road",
          "ave" : "avenue",
          "ave." : "avenue",
          "cir" : "circle",
          "blvd" : "boulevard",
          "ct" : "court",
          "sq" : "square",
          "pkwy" : "parkway",
          "ln" : "lane",
          "trl" : "trail",
          "dr." : "drive",
          "dr" : "drive",
          "hwy" : "highway",
          }

def update_name(name, mapping):
    lname = name.lower()
    for i in lname.split(" "):
        if i in mapping.keys():
```

```
        lname = lname.replace(i,mapping[i])
    return lname
```

3) Auditing

Problem: The toughest part of this project was understanding the structure of this raw data to plan out a way to clean and process the data. Auditing and reviewing the raw data structures involved several steps and a few days due to the size of the data.

Solution:
1) Find each tag type and number of each tag:
-Using the lesson 6 code, I created a dictionary of each tag type and count.

```
{'node': 11287705, 'nd': 12976083, 'bounds': 1, 'member': 35466, 'tag': 6105957, 'osm': 1, 'way':
764672, 'relation': 4168}
```

2) Identify parent level tags and child level tags:
-Not knowing how many levels there were and which were at each level, this step identified all first level tags and second level tags. There were no third-level tags.

```
first_level:  set(['node', 'relation', 'bounds', 'way'])
second_level:  set(['member', 'tag', 'nd'])
third_level:  set([])
```

3) Identify each relationship between one parent and its child-level tags:
-This step shows child level of each first level parent and returns a dictionary of each pair.

```
node:  set(['tag'])
relation:  set(['member', 'tag'])
bounds:  set([])
way:  set(['tag', 'nd'])
```

4) For each parent and child tag, show a sample to understand data:

```
for neighbor in root.findall('node')[:1]:
    print neighbor.tag, neighbor.attrib
    for i in neighbor.findall('tag'):
        print i.tag, i.attrib
```

2. Overview of the Data

File sizes

sample.osm .... 2.4 MB
sample.osm.json .... 3.5 MB

atlanta_georgia.osm ....2.42 GB
atlanta_georgia.osm.json …. 3.46 GB

# Number of documents

```
> db.atlanta.find().count()
12052377
```

# Sample Document

```
> db.atlanta.find_one()
```

```
{u'_id': ObjectId('55d1e6610a269d5d25071e85'),
 u'created': {u'changeset': u'3173372',
              u'timestamp': u'2009-11-21T04:30:48Z',
              u'uid': u'147510',
              u'user': u'woodpeck_fixbot',
              u'version': u'2'},
 u'id': u'52374108',
 u'pos': [32.87149, -85.193652],
 u'type': u'node'}
```

# Number of nodes

```
> db.atlanta.find({"type":"node"}).count()
11287611
```

# Number of ways

```
> db.atlanta.find({"type":"way"}).count()
764629
```

# Number of unique users

```
> len(db.atlanta.distinct("created.user"))
```

```
1646
```

# Top 2 contributing users

```
> cursor = db.atlanta.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
{"$sort":{"count":-1}}])
```

```
[{u'_id': u'Liber', u'count': 5599783},
 {u'_id': u'Saikrishna_FultonCountyImport', u'count': 2264410}]
```

# Number of users appearing only once (having 1 post and 2 posts)

```
> cursor = db.atlanta.aggregate([
        {"$group":{"_id":"$created.user", "count":{"$sum":1}}},
        {"$group":{"_id":"$count", "num_users":{"$sum":1}}},
        {"$sort":{"_id":1}},
                                ])
        list(cursor)[:2]
```

```
[{u'_id': 1, u'num_users': 276}, {u'_id': 2, u'num_users': 134}]
# 276 users have only 1 post and 134 users have only 2 posts
```

3. Other ideas about the datasets

Ideas on additional bitcoin adoption usage research & visualization
Using the region-wide or country-wide data regarding bitcoin (or other cyber money) payment adoption, it would be interesting to see the characteristics of surrounding areas where they accept cyber money including bitcoin. Also, it would be interesting to visualize this data and map it to make it available for users.

The current challenge in this dataset is that there is not enough data points to make this effort significant. One way to improve and increase the datasets on bitcoin usage would be understand the type of users who are currently contributing this type of data and encouraging them to contribute. Based on the contributor statistics, it seems clear that the type of data contributed is dependent on the contributor type. The following is the top 5 amenity types that the top user (Liber) has contributed, who has contributed over a half of this dataset.

```
[{u'_id': None, u'count': 11199554},
 {u'_id': u'parking', u'count': 2},
 {u'_id': u'place_of_worship', u'count': 2},
 {u'_id': u'police', u'count': 2},
 {u'_id': u'courthouse', u'count': 2}]
```

On the other hand, these are the contributors who contributed the bitcoin data. These users are not even in the top 10 contributor list in the total dataset. Researching more into these users' contribution activities and even method of contribution would help encourage this type of users and increase the datapoint on bitcoin adoption.

```
[{u'_id': u'Blobo123', u'count': 16},
 {u'_id': u'oldenburg69', u'count': 12},
 {u'_id': u'dreyzehner', u'count': 4},
 {u'_id': u'Robert Wilson', u'count': 4},
 {u'_id': u"Gino's Classic Barber Shoppe", u'count': 2}]
```

Contributor statistics

Top user contribution percentage (Liber) - 46.46%
Combined top 2 users' contribution (Liber and Saikrishna_FultonCountyImport) - 65.25%
Combined Top 10 users contribution - 88.46%

# Top 5 buildings that have wifi

```
db.atlanta.aggregate([{"$match":{"wifi":{"$exists":1}, "amenity":{"$exists":1}}},
                {"$group":{"_id":"$amenity", "count":{"$sum":1}}},
                {"$sort":{"count":-1}},
                {"$limit":5}
                ])
```

```
[{u'_id': u'cafe', u'count': 5},
 {u'_id': u'restaurant', u'count': 4},
 {u'_id': u'pub', u'count': 2},
 {u'_id': u'bar', u'count': 1}]
```

# Amenity types that accept bitcoin

```
db.atlanta.aggregate([{"$match":{"payment.bitcoin":{"$eq":"yes"}}},
                {"$group":{"_id":"$amenity", "count":{"$sum":1}}},
                {"$sort":{"count":-1}},
                {"$limit":5}
                ])
```

```
[{u'_id': None, u'count': 20},
 {u'_id': u'restaurant', u'count': 7},
 {u'_id': u'bar', u'count': 1},
 {u'_id': u'atm', u'count': 1},
 {u'_id': u'doctors', u'count': 1}]
```

#Top 5 Contributors who reported amenity information that accepts bitcoin

```
db.atlanta.aggregate([{"$match":{"payment.bitcoin":{"$eq":"yes"}}},
                      {"$group":{"_id":"$created.user", "count":{"$sum":1}}},
                      {"$sort":{"count":-1}},
                      {"$limit":5}
                      ])
```

```
[{u'_id': u'Blobo123', u'count': 16},
 {u'_id': u'oldenburg69', u'count': 12},
 {u'_id': u'dreyzehner', u'count': 4},
 {u'_id': u'Robert Wilson', u'count': 4},
 {u'_id': u"Gino's Classic Barber Shoppe", u'count': 2}]
```

#Top 5 amenity types that Saikrishna_FultonCountyImport reported

```
db.atlanta.aggregate([{"$match":{"created.user":{"$eq":"Saikrishna_FultonCountyImport"}}},
                     {"$group":{"_id":"$amenity", "count":{"$sum":1}}},
                     {"$sort":{"count":-1}},
                     {"$limit":5}
                     ])
```

```
[{u'_id': None, u'count': 4525766},
 {u'_id': u'place_of_worship', u'count': 2558},
 {u'_id': u'hospital', u'count': 196},
 {u'_id': u'parking', u'count': 126},
 {u'_id': u'social_centre', u'count': 48}]
```

#Top 5 amenity types that Liber reported

```
db.atlanta.aggregate([{"$match":{"created.user":{"$eq":"Liber"}}},
                      {"$group":{"_id":"$amenity", "count":{"$sum":1}}},
                      {"$sort":{"count":-1}},
                      {"$limit":5}
                      ])
```

```
[{u'_id': None, u'count': 11199554},
 {u'_id': u'parking', u'count': 2},
 {u'_id': u'place_of_worship', u'count': 2},
 {u'_id': u'police', u'count': 2},
 {u'_id': u'courthouse', u'count': 2}]
```

# Top 4 religion at place of worship

```
db.atlanta.aggregate([{"$match":{"amenity":{"$exists":1}, "amenity":"place_of_worship"}},
                      {"$group":{"_id":"$religion", "count":{"$sum":1}}},
                      {"$sort":{"count":-1}}, {"$limit":2}])
```

```
[{u'_id': u'christian', u'count': 4293},
 {u'_id': None, u'count': 1314},
 {u'_id': u'jewish', u'count': 9},
 {u'_id': u'muslim', u'count': 5}]
```

# Among the documents that have amenity, show the top 5 amenities in Atlanta

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

```
[{u'_id': u'place_of_worship', u'count': 5630},
 {u'_id': u'parking', u'count': 3082},
 {u'_id': u'school', u'count': 2355},
 {u'_id': u'grave_yard', u'count': 2180},
 {u'_id': u'restaurant', u'count': 1123}]
```

# Top 5 cuisine list from Atlanta restaurants

```
> db.atlanta.aggregate([
        {"$match":{"amenity":{"$exists":1}, "amenity":"restaurant"}},
        {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},
        {"$sort":{"count":-1}}, {"$limit":5}
        ])
```

```
[{u'_id': None, u'count': 398},
 {u'_id': u'american', u'count': 97},
 {u'_id': u'pizza', u'count': 83},
 {u'_id': u'mexican', u'count': 69},
 {u'_id': u'burger', u'count': 31}]
```