Rahel Ghebrekidan

Udacity - Data Analyst Nondegree

Project 3: Data Wrangling with Mongodb (Open Street Map)

Data: July 22, 2016

Santa Monica Open Street Map

1. Introduction

Open Street Map is a free geographic data which was created by Steve Coast in 2004 in United Kingdom. Later many people have contributed in amending different places.

I have used the Santa Monica, California, USA open street map data for this project. I have chosen Santa Monica first because it is in California and it will be either for me to know some facts about it. Second, I have a plan to visit Los Angeles and Santa Monica is one of the cities in Los Angeles county. So it is a great opportunity for me to see how the city looks like in this dataset.

The objectives of this project are:-

A. To parse xml files in python
B. To update if there are any problems with the dataset
C. To change XML files to JSON in order to manipulate it in MongoDB and
D. To come up with some analysis of the dataset.

I have found problems with Street Names, Postcode, Telephone Numbers and City in the dataset and I have updated them. I have used MongoDB and python in the data wrangling and analysis.

2. Dataset Overview

The dataset is extracted from open source https://s3.amazonaws.com/metro-extracts.mapzen.com/santa-monica_california.osm.bz2 . The dataset of Santa Monica is originally an OSM file and has been converted to JSON file during the analysis. The OSM file is 111.15 Mb and the JSON file is 125.78 Mb.

Unique element types

{'bounds': 1,
 'member': 4167,
 'nd': 544106,
 'node': 486074,
 'osm': 1,

'relation': 792,
'tag': 320481,
'way': 51875}

Number of unique users

> len(db.St_Monica_CA.distinct('created.user'))
Result: 341

3. Problems Encountered in the Dataset

I surveyed several information in the dataset and I focused in Street names, Postcode, Phone number and city.

3.1 Street Names

The dataset have some problems in naming the streets. It has abbreviations in the street type. The streets with abbreviations in the street type are audited by defining the expected street types and mapping and resulted as follows.

Walnut Ln => Walnut Lane
Main St. => Main Street
Ocean Bd. => Ocean Boulevard
Olive Ave => Olive Avenue
Ohio Ave => Ohio Avenue
Montana Ave => Montana Avenue
Third Street Promenade => Third Street Promenade
Santa Monica Bvd => Santa Monica Boulevard
Pico Blvd => Pico Boulevard
Lincoln Blvd => Lincoln Boulevard
Wishire Blvd => Wishire Boulevard
W Washington Blvd => W Washington Boulevard
Wilshire Blvd => Wilshire Boulevard
W Pico Blvd => W Pico Boulevard
Washington Blvd => Washington Boulevard
Santa Monica Blvd => Santa Monica Boulevard
Olive ave => Olive Avenue
Entrada Dr => Entrada Drive
Donald Douglas Loop South => Donald Douglas Loop South
Olympic Blvd. => Olympic Boulevard

3.2 Postcodes

I noticed some Postcodes had nine digits with hyphen after the fifth digit and some has state(CA) prefix. The US post uses 4 codes after the five digit zipcode for efficient allocation especially

when sorting mails. Although it does not mean that the additional four digits are wrong numbers, for consistency in the dataset, I have decided to omit them.

```
def update_zipcode(zipcode):
    # Deleting the digits and the hyphen after the fifth digit and the prefix "CA"
    Ext_check = re.findall('(\d{5})-\d{4}', zipcode)
    if Ext_check:
        return re.sub('(\d{5})-\d{4}', '\\1', zipcode)
    else:
        return re.sub('[^0-9]', '', zipcode)[:5]
for zipcodes, ways in audit_zipcode.iteritems():
    for zipcode in ways:
        better_zipcode = update_zipcode(zipcode)
        print zipcode, "=>", better_zipcode
```

```
90025-9998 => 90025
CA 90291 => 90291
90401-2405 => 90401
CA 90272 => 90272
CA 90404 => 90404
90272-3719 => 90272
CA 90401 => 90401
CA 90405 => 90405
90064-1508 => 90064
90291-3879 => 90291
```

3.3 Phone Number

According to most common way of telephone number writing, I assumed telephone number formatting valid as follows: - +1 345 346 3597, 345 346 3597, (345) 346 3597, 345 - 346 – 3597 and 1- 345- 346-3567. The invalid phone numbers are updated as follows.

```
def update_phone(phone_number):
    valid_phone = re.sub('[^0-9]+', '', phone_number)
    phone_pattern = re.sub("(\d)(?=(\d{3})+(?!\d))", r"\1-", "%d" % int(valid_phone[:-1])) + valid_phone[-1]
    return phone_pattern
for phone_number, ways in audit_phone_number.iteritems():
    for phone in ways:
        better_phone_num = update_phone(phone)
        print phone, "=>", better_phone_num
```

```
+01-310-260-6308 => 1-310-260-6308
310. 444. 0045 => 310-444-0045
310.473.1447 => 310-473-1447
310.478.3545 => 310-478-3545
310.826.2229 => 310-826-2229
```

3.4 City

The city name is not showing Santa Monica only. There are other cities mentioned there, some with lower case, and some with CA suffix.

```
city = db.St_Monica_CA.aggregate( [
    { "$match" : { "address.city" : { "$exists" : 1} } },
    { "$group" : { "_id" : "$address.city", "count" : { "$sum" : 1} } },
    { "$sort" : { "count" : -1}},
      ])
print(list(city))
```

```
[{u'count': 92, u'_id': u'Santa Monica'}, {u'count': 34, u'_id': u'Los Angeles'}, {u'count': 19, u'_id': u'Marina Del
Rey CA'}, {u'count': 13, u'_id': u'Marina Del Rey'}, {u'count': 10, u'_id': u'Venice'}, {u'count': 7, u'_id': u'Marin
a del Rey'}, {u'count': 3, u'_id': u'Los Angeles-Venice'}, {u'count': 2, u'_id': u'West Los Angeles'}, {u'count': 1,
u'_id': u'Venice CA'}, {u'count': 1, u'_id': u'Marina del Ray'}, {u'count': 1, u'_id': u'santa Monica'}, {u'count':
1, u'_id': u'Pacific Palisades'}]
```

Santa Monica is one of the Los Angeles county city which is located in west of Los Angeles. People who live in Santa Monica can say that their city name is Los Angeles. In addition Marina Del Rey, Venice, Pacific Palisades are also in Los Angeles county. Hence, I did not prefer to

replace all the city names with Santa Monica. I have only updated the names to be consistent in writing (matching cases in the names). City names West Los Angeles and Los Angeles are updated as Santa Monica. Los Angeles- Venice is updated as Venice and the Suffix CA is omitted.

The process of updated the city name is done in MongoDB.

4. Data Analysis with MongoDB

The OSM(XML) file is changed to JSON in order to be suitable to be analyzed in MongoDB and all the updates made were included in the dataset.

A person who wants to visit any place is most likely interested in tourist sites, type of Cuisines in the city and amenities. Below are those of summarized information regarding tourist sites, types of cuisines and amenities of Santa Monica.

4.1 Tourist Sites

```
Tourism_Sites = db.St_Monica_CA.aggregate( [
    { "$match" : { "tourism" : { "$exists" : 1} } },
    { "$group" : { "_id" : "$tourism", "count" : { "$sum" : 1} } },
    { "$sort" : { "count" : -1}},
      ])
print(list(Tourism_Sites))
```

```
[{u'count': 54, u'_id': u'hotel'}, {u'count': 12, u'_id': u'attraction'}, {u'count': 9, u'_id': u'motel'}, {u'count':
4, u'_id': u'museum'}, {u'count': 3, u'_id': u'picnic_site'}, {u'count': 3, u'_id': u'hostel'}, {u'count': 2, u'_id':
u'artwork'}, {u'count': 1, u'_id': u'theme_park'}, {u'count': 1, u'_id': u'viewpoint'}, {u'count': 1, u'_id': u'infor
mation'}, {u'count': 1, u'_id': u'apine_hut'}]
```

4.2 Types of Cuisines

```
cuisine = db.St_Monica_CA.aggregate([{"$match":{"amenity":{"$exists":1},
                                     "amenity":"restaurant",}},
                        {"$group":{"_id":{"Food":"$cuisine"},
                                     "count":{"$sum":1}}},
                        {"$project":{"_id":0,
                                     "Food":"$_id.Food",
                                     "Count":"$count"}},
                        {"$sort":{"Count":-1}},
                        ])
print(list(cuisine))
```

```
[{u'Food': None, u'Count': 43}, {u'Food': u'american', u'Count': 9}, {u'Food': u'mexican', u'Count': 5}, {u'Food':
u'italian', u'Count': 5}, {u'Food': u'burger', u'Count': 4}, {u'Food': u'regional', u'Count': 2}, {u'Food': u'thai',
u'Count': 2}, {u'Food': u'steakhouse,_italian,_mediterranean', u'Count': 2}, {u'Food': u'sandwich', u'Count': 2},
{u'Food': u'pizza', u'Count': 2}, {u'Food': u'indian', u'Count': 1}, {u'Food': u'french', u'Count': 1}, {u'Food': u'v
egan', u'Count': 1}, {u'Food': u'greek', u'Count': 1}, {u'Food': u'argentinian', u'Count': 1}, {u'Food': u'vegetaria
n', u'Count': 1}, {u'Food': u'international', u'Count': 1}, {u'Food': u'seafood', u'Count': 1}, {u'Food': u'thai;chin
ese', u'Count': 1}, {u'Food': u'asian', u'Count': 1}]
```

4.3 Amenities

```
amenity = db.St_Monica_CA.aggregate([{'$match': {'amenity': {'$exists': 1}}}, \
                                     {'$group': {'_id': '$amenity', \
                                                 'count': {'$sum': 1}}}, \
                                     {'$sort': {'count': -1}}, \
                                     {'$limit': 10}])
print(list(amenity))
```

```
[{u'count': 176, u'_id': u'parking'}, {u'count': 86, u'_id': u'restaurant'}, {u'count': 83, u'_id': u'bicycle_renta
l'}, {u'count': 57, u'_id': u'school'}, {u'count': 50, u'_id': u'place_of_worship'}, {u'count': 42, u'_id': u'cafe'},
{u'count': 41, u'_id': u'drinking_water'}, {u'count': 23, u'_id': u'fast_food'}, {u'count': 20, u'_id': u'toilets'},
{u'count': 17, u'_id': u'fuel'}]
```

5. Limitation

This project is limited in updating problems with street names, phone numbers, postcode and city. My knowledge in the city is limited and it was difficult in updating the city as I was not sure if I had to replace the city names like Venice with Santa Monica.

6. Conclusion

In this project I have updated some problems with street names, postcodes, phone numbers and city names of Santa Monica. Making the source open is essential for inputs from several interested entities. In this work I have made some improvements and if several individuals and institutions attempt to refine and correct the OSM data of the Santa Monica City beneficiaries would have complete and comprehensive correct dataset. In this Information age availing completely correct data is both challenging and very essential. Hence, local people has to take initiatives.

I suggest that it would be better if there is a means where each volunteer who makes any update in any place to make notes on why the changes were made. The tool can be a kind of readme notepad. Although it might be discouraging for some people to make updates, it is good that every to be responsible in making any changes so that the data trustful worthy to be used.

Reference:

1. Data design Goup.(2014).JSON : Conver XML to JSON: Retrieved from http://convertjson.com/xml-to-json.htm
2. Sudirman,J.(2013). Open Street Map: Retrieved from http://napitupulu-jon.appspot.com/posts/wrangling-openstreetmap.html
3. Allanbreyes.(2015). Udacity Data Science:P2: Retrieved from https://github.com/allanbreyes/udacity-data-science/tree/master/p2
4. Bogotobogo.(2016). MongoDB with Pymongo I: Retrieved from http://www.bogotobogo.com/python/MongoDB_PyMongo/python_MongoDB_pyMongo_tutorial_installing.php
5. Stack over flow: Retrieved from http://stackoverflow.com/questions/2577236/regex-for-zip-code
6. Regular expression101: regular expression tester : Retrieved from https://regex101.com/r/lV5yU9/1
7. MongoDB (2008). MongoDB. Retrieved from: https://docs.mongodb.com/manual/reference/command/getLastError/