OpenStreetMap Project
Data Wrangling with MongoDB

Kien Nguyen

Map Area: Des Moines, IA, United States
https://s3.amazonaws.com/metro-extracts.mapzen.com/des-moines_iowa.osm.bz2

1. Problems Encountered in the Map

After downloading the entire Des Moines area and running some experimentation on the data, I noticed three main problems with the data, which I will discuss in the following order:

- Inconsistent postal codes ("50023", "50313-2798")
- Inconsistent date formats ("10/04/2004", "4/30/1979")
- Over-abbreviated street types ("St", "Rd", "Pkwy")

a. Inconsistent postal codes

Among the 5291 data points that have information about postal codes, 5263 of them are formatted as 5-digit postal code, and the other 28 points include the lasting 4 digits at the end. It will be hard to query the data if we have both types of format in MongoDB. For example, if I want to count all the data points that have postal code of 50023, I will miss all the cases when the postal code has the trailing 4 digits such as 50023-2421.

Therefore, in order for the data to be consistent, I decided to strip those extra 4 digits from those 28 data points.

b. Inconsistent date format

There are 576 data point that contain information about the date created for the Geographic Names Information System (GNIS) of that data point. Among those 571 agree with the format as "mm/dd/yyyy". However, there are 5 data points which don't want to keep the month and day information as two digits each. Those "strange" data points are: "4/30/1979", "4/30/1979", "1/1/2000", "4/30/1979", "10/4/2004".

Similar to the inconsistency of postal code, I decided to convert those 5 date data to the normal "mm/dd/yyyy" format as the other data points.

c. Over-abbreviated street types

In our data, there are 9 data points where users input the abbreviated versions for street types such as St or ST for street or Ct. for Court.

In order to make the street data more consistent, I decided to correct those abbreviated name before insert the date into MongoDB.

2. Data Overview

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

File size:

- des-moines_iowa.osm ... 145.5 MB

Numbers of documents / data points

- db.desMoines.find().count()
  739926

Number of nodes

- db.desMoines.find({"type": "node"}).count()
  684196

Number of ways

- db.desMoines.find({"type": "way"}).count()
  55723

Number of unique users

- db.desMoines.distinct("created.user").length
  297

Number of users appearing only once (having one post)

- db.desMoines.aggregate([{"$group": {"_id": "$created.user",
  "count": {"$sum": 1}}},
  {"$group": {"_id": "$count",
  "num_users": {"$sum": 1}}},
  {"$match": {"_id": 1}}])

  {"_id": 1, "num_users": 45}

Number of shops

- db.desMoines.find({"shop": {"$exists": 1}}).count()
  1099

Number of nodes with amenity as café

- db.desMoines.find({"amenity": "cafe"}).count()
  45

Top 1 contributing user

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

  {"_id": "Jeff Ollie", "count": 313174}

3. Additional Ideas

Contributor statistics:

- Top user contribution percentage ("Jeff Ollie"): 313174 / 739926 = 42.33%
- Top 2 users contribution percentage ("Jeff Ollie" and "iowahwyman"): (313174 + 178406) / 739926 = 66.44%
- Combined top 10 users contribution: 708187 / 739926 = 95.71%

```
➢ db.desMoines.aggregate(
   [{"$group": {"_id": "$created.user",
               "count": {"$sum": 1}}},
   {"$sort": {"count": -1}},
   {"$limit": 10},
   {"$group": {"_id": "sum10", "res":
               {"$sum": "$count"}}}])

  {"_id": "sum10", "res": 708187}
```

Additional Data Exploration

Top 10 appearing amenities

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

[{u'_id': u'parking', u'count': 3499},
{u'_id': u'place_of_worship', u'count': 419},
{u'_id': u'restaurant', u'count': 317},
{u'_id': u'school', u'count': 268},
{u'_id': u'fast_food', u'count': 223},
{u'_id': u'shelter', u'count': 205},
{u'_id': u'bench', u'count': 191},
{u'_id': u'fuel', u'count': 159},
{u'_id': u'bank', u'count': 137},
{u'_id': u'toilets', u'count': 122}]
```

According to the data above, parking space occupies the majority of the amenities in Des Moines, which makes sense since Des Moines is the biggest city in Iowa and is a crowded place, so people need more space to park.

Top 3 biggest religions

```
> db.desMoines.aggregate([{"$match": {"religion": {"$exists": 1}}},
                {"$group": {"_id": "$religion",
                      "count": {"$sum": 1}}},
                {"$sort": {"count": -1}},
                {"$limit": 3}])

[{u'_id': u'christian', u'count': 398},
{u'_id': u'buddhist', u'count': 4},
{u'_id': u'jewish', u'count': 3}]
```

It is not surprised that Christian is the most popular religion in Iowa as it is in the United States.

```
# Top 3 most popular highway types
> db.desMoines.aggregate([{"$match": {"highway": {"$exists": 1}}},
                        {"$group": {"_id": "$highway",
                                "count": {"$sum": 1}}},
                        {"$sort": {"count": -1}},
                        {"$limit": 3}])
```

```
[{u'_id': u'service', u'count': 12779},
{u'_id': u'residential', u'count': 9285},
{u'_id': u'path', u'count': 3816}]
```

It appears to have a lot of services on the highway in Des Moines, which also makes sense since Des Moines is the biggest city in Iowa, and services are important factors in the development of the city's economy.

Conclusion

The OpenStreetMap data for Des Moines area is of course not complete. There are tons of data points that need to be filled in the information such as address or Geographical Name Information System. Although there are still some places where there are some inconsistencies, the users have been following mostly uniform formats of date and postal code. With the support of Python and the aggregation framework, it is possible to programmatically clean the data for OpenStreetMap.