Project Summary

What is your name?

Liuyin Cheng

What E-mail address do you use to sign in to Udacity?

cliuyin@gmail.com

What area of the world you used for your project? Post a link to the map position and write a short description. Note that the osm file of the map should be at least 50MB.

URL:
http://www.openstreetmap.org/export#map=11/40.0155/-82.9234 (Columbus)
I chose this particular place because I have been living here for 3 years and I know it well and would like its map to be improved in quality!

Is there a list of Websites, books, forums, blog posts, github repositories etc that you referred to or used in this submission (Add N/A if you did not use such resources)?

Udacity
Python-phonenumbers: https://github.com/daviddrysdale/python-phonenumbers

Please carefully read the following statement and include it in your email:

*"I hereby confirm that this submission is my work. I have cited above the origins of **any** parts of the submission that were taken from Websites, books, forums, blog posts, github repositories, etc. By including this in my email, I understand that I will be expected to explain my work in a video call with a Udacity coach before I can receive my verified certificate."*

Is there any other important information that you would want your project evaluator to know?

Use this space to communicate with your project evaluator. Is there anything you would like to communicate? Feedback or suggestions?

Problems Encountered in the Map

After downloading the Columbus data, I took a look at the osm file and noticed there are two main problems with the data: over-abbreviated street names and inconsistent telephone number.

Over-abbreviated Street Names

Some address has full street names, like “Sunbury Road”, but others have abbreviated names, like “Morse Rd”. To make them more readable, I updated all substring in abbreviated names, such “Morse Rd” becomes “Morse Road”.

Inconsistent Telephone Number

The telephone numbers have different formats, and here are some of them: “(614) 476-7100”, “614-764-1144”, “6142883354”, “+1-614-299-4826”. I wanted them having a same and clear format for all telephone numbers. Using the python lib, “phonenumbers”, all valid phone numbers can be converted into “(xxx) xxx-xxxx” format or “(xxx) xxx-xxxx ext. xxxx” if it has extension. I added one function “update_phone(…)” in audit.py and called it in “shape_element(…)” function of “data.py” to update phone numbers.

Data overview & Mongo Shell Queries

I modified data.py and audit.py to meet my needs. Once Osm data was cleaned up, then converted into json file and finally imported into MongoDB, I can gather some basic statistics about the database.

File Size

openmap_columbus.osm -- 98.4MB
openmap_columbus.osm.json – 106MB

Number of Documents

```
> db.columbus.find().count()
454487
```

Number of nodes

```
> db.columbus.find({"type": "node"}).count()
408614
```

Number of ways

```
> db.columbus.find({"type": "way"}).count()
45866
```

Number of unique user

```
> db.columbus.distinct("created.user").length
```

```
535
```

Top 3 Contributors

```
> db.columbus.aggregate({$group: {_id: "$created.user", count: {$sum: 1}}},
                        {$sort: {count: -1}},
                        {$limit: 3})
{ "_id" : "woodpeck_fixbot", "count" : 212241 }
{ "_id" : "Vid the Kid", "count" : 71066 }
{ "_id" : "TIGERcnl", "count" : 22926 }
```

“woodpeck fixbot” is an automated edit used by Frederik Ramm. “Vid the Kid” is another main mapper to mostly central Ohio. “TIGERcnl” is another automated edit. They contribute 68% of data.

Other Ideas

In this section, I dug deeply into the supermarket data.

Number of shops

```
> db.columbus.find({"shop": {$exists: true}}).count()
377
```

Number of distinct shop categories

```
> db.columbus.distinct("shop", {"shop": {$exists: true}}).length
55
```

Top 5 categories of shops

```
> db.columbus.aggregate({$match: {"shop": {$exists: true}}}, {$group: {_id: "$shop", "count":
{$sum: 1}}}, {$sort:{"count": -1}}, {$limit: 5})
{ "_id" : "supermarket", "count" : 72 }
{ "_id" : "convenience", "count" : 39 }
{ "_id" : "mall", "count" : 37 }
{ "_id" : "car_repair", "count" : 18 }
{ "_id" : "doityourself", "count" : 16 }
```

Supermarket is the largest category of shop and I am very familiar to this data subset.

Number of supermarkets

```
> db.columbus.find({"shop": "supermarket"}).count()
72
```

Number of distinct supermarkets

```
> db.columbus.distinct("name", {"shop": {$exists: true}, "shop": "supermarket"}).length
```

Top 5 popular supermarkets

```
> db.columbus.aggregate({$match: {"shop": {$exists: true}, "shop": "supermarket"}}, {$group: {_id: "$name", "count": {$sum: 1}}}, {$sort: {"count": -1}}, {$limit: 5})
{ "_id" : "Kroger", "count" : 25 }
{ "_id" : "Giant Eagle", "count" : 12 }
{ "_id" : "Meijer", "count" : 4 }
{ "_id" : "Target", "count" : 3 }
{ "_id" : "Walmart Supercenter", "count" : 2 }
```

The data shows there are only 25 “krogeer” in Columbus area, which is much less than reality. At first, I thought “Kroger” may have inconsistent names and I listed relative supermarkets.

Kroger Shops

```
> db.columbus.aggregate({$match: {"shop": {$exists: true}, "shop": "supermarket", "name": {$regex: /Kroger/i}}}, {$group: {_id: "$name", "count": {$sum: 1}}}, {$sort: {"count": -1}})
{ "_id" : "Kroger", "count" : 25 }
{ "_id" : "Kroger Food & Drug", "count" : 1 }
{ "_id" : "Kroger Marketplace", "count" : 1 }
{ "_id" : "Morse Road Krogers", "count" : 1 }
```

Even counting other Kroger shops, the number of Korgers in this area is only 28, which is still much less than reality. Now it’s obvious that openmap data about Columbus area is incomplete, at least for supermarket part.

Whole Foods Market

I know there are two whole foods market in Columbus area, and wanted to check with openmap data.

```
> db.columbus.find({"name": {$regex: /whole foods/i}})
{ "_id" : ObjectId("55694db4c857e2351821067c"), "shop" : "supermarket", "name" : "Whole Foods", "created" : { "changeset" : "2611623", "user" : "Joe Inoh", "version" : "1", "uid" : "176089", "timestamp" : "2009-09-25T04:12:23Z" }, "pos" : [ 40.00668, -83.0521616 ], "type" : "node", "id" : "506867784" }
{ "_id" : ObjectId("55694db6c857e23518221838"), "shop" : "supermarket", "name" : "Whole Foods Market", "created" : { "version" : "1", "uid" : "909257", "timestamp" : "2013-05-26T11:09:39Z", "changeset" : "16291337", "user" : "Wrong Again" }, "pos" : [ 40.0983016, -83.0867522 ], "address" : { "city" : "Columbus", "street" : "West Dublin – Granville Road", "housenumber" : "3670", "postcode" : "43235" }, "type" : "node", "id" : "2320540467" }
```

There are two result about Whole Foods, one in 43235 and another incomplete one. But, their “pos” information are almost the same and they may be the same shop. As I know, there are only two whole foods in Columbus area and another one is in 43221 area with location of [39.96118, -82.99879]. Now it's clear that the returned results are duplicated records.

There may be some records limiter to each other or some duplicated records in this data set and reducing them can improve the data quality. However, there are two challenges doing it.

1. Without additional information, how can we know two or more similar records are mentioning the same data entity. Like in Whole Foods example, if I am not familiar with Columbus, I am not able to figure out problematic record.
2. Even we know the duplications, we don't know which one we can trust. Or how can we combine them to form one more complete record.

If we can implement this improvement, the data set would be more accurate.

Conclusion

After this review of data, I think Columbus data is incomplete and also has some duplicate information. This exercise gives me a good example on how messy the real data could be and how important it is to clean data before further analysis. Thank you to all Udacity staff and Openmap editors.