Problems encountered in your map

Initially I was thinking to select Atlanta or Chennai both of them were of big size, so my machine could not handle such a big data set. So I decided to choose some other city data set which of size less than 200 MB. I am planning to attend Nebraska next week for warren buffet conference, so thought it would be a better idea to choose. That way I will know about the city before I visit (I in fact have some idea about it as I had read about it before reserving travel and stay.

The second problem I faced was to unzip the file as it was a bz2 compressed format. I downloaded the 7zip and unzipped the file to get the OSM file.

Though i tried to follow the approach I used to solve the Lesson 6:case study problems, when I changed the data set to read new file I had few data locations issues. I solved them to point the right as I had both the bz2 and unzipped file.

I also explored the other parsing methodologies and settled with iterative parsing.

I also had tough time in resolving the street names to bring it to common format. As it had variety of names like blvd,st and numbered street. Took some time and explored to settle the final mapping list.

It took some time to load into mongo db as well, though I had a certification on mongodb this time I have to load a big file so it took sometime load in fact the process just ran over night.

Overview of the Data

I took the Omaha, Nebraska dataset for analysis, to generate the .json file from the data set I used the file wrangling_tojson.py from iPython. I gave the omaha_nebraska.osm file as an input to the wrangling_tojson.py file.

The output file has been named as omaha_nebraska.osm.json

I had the mongo db installed in machine already, as I did the mongo db certification in the same machine. So I just have to connect db and load the data.

First, we start up MongoDB:

```
$ mongod --dbpath ~/data/db
```

Next, we import the data:

```
$ mongoimport --db map --collection map --file ./data/ omaha_nebraska.osm.json
```

Now start mongo:

```
$ mongo
> use test
switched to db test
```

Size of the file

The original OSM file is 161 MB. The JSON file generated from the OSM file is 233MB.

Let's look at the size of the actual collection:

> db.wrang.dataSize()

235210640

Number of unique users:

319 users have edited this map.

> db.wrang.distinct("created.user").length;

319

Number of nodes and ways:

The omaha map contains 694,980 nodes and 79.232 ways:

> db.wrang.find({type:"node" }).length();

694980

> db.wrang.find({type:"way" }).length();

79232

number of chosen type of nodes

```
There are 34 cafes in omaha
```

> db.wrang.find({amenity:"cafe" }).length();

34

```
There are 645 shops in omaha
```

> db.wrang.find({shop:{$exists:true}}).count();

645

- **Other ideas about the datasets**

Though it's covering most of the business locations, it would be good if it can cover all of the business locations in the map. But it would be very tedious and time-consuming to add the appropriate information to all of them. I wonder whether it would be possible to programmatically get that data from the Google Maps API. But using information from Google Maps to add information to OpenStreetMap seems like data stealing and might violate the terms of agreement for Google Maps. May be if there is a way to get the accidents happening in different locations, we can analyze which junction or area has frequent accidents. Also, if there is a way to predict the most used roads that will be good.

wrangling_namechange file updates :

```
mapping = { "St": "Street",
            "St.": "Street",
            "Ave": "Avenue",
            "Rd.": "Road",
            '106': '106',
            '330': '330',
            '370': '370',
            'A': 'A',
            'Ave': 'Avenue',
            'Blvd': 'Blvd',
            'Broadway': 'BroadWay',
            'Circle': 'circle',
            'Dr': 'Drive',
            'Hascall': 'Hascall',
            'Highway': 'Highway',
            'Maple': 'Maple',
            'North': 'North',
            'Plaza': 'Plaza',
            'Q': 'Q',
            'Rd': 'Road',
            'STREET': 'Street',
            'South': 'South',
            'St': 'Street',
            'St.': 'Street',
            'Way': 'Way',
            'bing': 'bing'

            }
```