Overview

This is an exploration of the OpenStreetMap data for the New York metro area.

Problems Encountered

When running an audit of the downloaded data of the metro New York area, I observed the following issues:

- Inconsistent street suffixes (Ave. vs Avenue, St. vs Street, etc)
- Bad street names
- Inconsistent/missing city/state information
- Inconsistent zip codes (“11201”, “NY 11201”, “11201-1234”)
- Inconsistent state names
- Missing/inconsistent Borough information

Inconsistent street suffixes

Similar to the data from class, the NY metro extract has a lot of inconsistencies in street suffixes but the list of variations was a lot longer. The list of mismatches was generated by parsing the original OSM file and collecting variations on street suffixes.
The data was cleaned by defining a “canonical” suffix map and using that to map street suffixes before importing into MongoDB.

Bad street names

Some of the street names contained extraneous information like suite numbers and building names. The number of occurrences was low so I chose to leave them as is. Some other examples of bad street names included intersections (“A St. and B St.”) and repeated names (“A st; A st”), but this was infrequent as well.

Inconsistent/missing city/state information

As the metro New York is not limited to New York, having city and state information is important for performing many analysis. Running some sample queries over the MongoDB data showed that many of the documents had zip codes but not city or state.
I found a zip code database and added a step to the processing script to audit the city/state information given a zipcode. Auditing showed that the zipcode database was almost always more correct than the OSM data, so I used the zipcode database as the source of city/state when possible.
The zip code data was obtained from http://www.boutell.com/zipcodes/ which is based on 2000 census data.

Even after filling in city, state using zip code data there were still some number of addresses without city or state information. One possible idea worth exploring in the future is using the lat/long data to obtain city/state information.

Inconsistent zip codes

Normalizing the zip codes was important in order to perform the city/state lookup described above. I used a regular expression to extract the first sequence of 5 digits from the postcode field.

Inconsistent state names

Some simple MongoDB aggregation queries showed that the state names were inconsistent in terms of casing and abbreviation. This became less of an issue once the state data was being looked up in the zip code database, but it was still important to normalize the names in order to effectively compare OSM values to the zipcode database.

Missing/inconsistent Borough information

When looking at data in the New York area, it's often interesting to explore/compare data by boroughs. The OSM data for New York does not appear to have sufficient borough information and frequently city/borough are used interchangeably (e.g. Queens vs Forest Hills). I did not attempt to address this in the analysis.

Data Overview

Data files

new-york_new-york.osm	2.1G
new-york_new-york.json	2.2G

Basic data statistics

Number of documents

```
> db.nyc.find().count()
9792801
```

Number of nodes

```
> db.nyc.find({"type":"node"}).count()
8405988
```

Number of ways

```
> db.nyc.find({"type":"way"}).count()
1386813
```

Breakdown of addresses by state

```
> db.nyc.aggregate({ $match: { "address": { $exists: true }}}, {
"$group": { "_id": "$address.state", "count": { "$sum": 1 } } }, {
$sort: { count: -1 }})
{ "_id" : "NY", "count" : 917752 }
{ "_id" : null, "count" : 5609 }
{ "_id" : "NJ", "count" : 2497 }
{ "_id" : "CT", "count" : 183 }
{ "_id" : "KY", "count" : 1 }
{ "_id" : "SC", "count" : 1 }
{ "_id" : "PA", "count" : 1 }
{ "_id" : "VA", "count" : 1 }
```

Still quite a few addresses without a state.

What kind of cuisines can be found (top 20)?

```
> db.nyc.aggregate({$match: { cuisine: { $exists: true }}}, {$group:
{ _id: "$cuisine", count: { $sum: 1 }}}, { $sort: { count: -1 } })
{ "_id" : "burger", "count" : 217 }
{ "_id" : "coffee_shop", "count" : 215 }
{ "_id" : "italian", "count" : 156 }
{ "_id" : "pizza", "count" : 153 }
{ "_id" : "american", "count" : 149 }
{ "_id" : "mexican", "count" : 120 }
{ "_id" : "sandwich", "count" : 84 }
{ "_id" : "chinese", "count" : 76 }
{ "_id" : "donut", "count" : 46 }
{ "_id" : "japanese", "count" : 46 }
{ "_id" : "ice_cream", "count" : 44 }
{ "_id" : "indian", "count" : 40 }
{ "_id" : "french", "count" : 37 }
{ "_id" : "thai", "count" : 36 }
{ "_id" : "chicken", "count" : 26 }
{ "_id" : "asian", "count" : 26 }
{ "_id" : "regional", "count" : 21 }
{ "_id" : "sushi", "count" : 20 }
{ "_id" : "steak_house", "count" : 19 }
{ "_id" : "seafood", "count" : 18 }
```

Fast food is over-represented. Non-fast food data has poor coverage.

What's the most popular Pizza chain/store name (top 20)?

```
> db.nyc.aggregate({$match: { cuisine: "pizza"}}, {$group: { _id:
"$name", count: { $sum: 1 }}}, { $sort: { count: -1 } })
{ "_id" : null, "count" : 5 }
{ "_id" : "Domino's Pizza", "count" : 3 }
{ "_id" : "Patsy's Pizzeria", "count" : 3 }
{ "_id" : "2 Bros Pizza", "count" : 3 }
{ "_id" : "Saporito", "count" : 2 }
{ "_id" : "Motorino", "count" : 2 }
{ "_id" : "Ben's Pizzeria", "count" : 2 }
{ "_id" : "Papa John's", "count" : 2 }
{ "_id" : "Joe's Pizza", "count" : 2 }
{ "_id" : "Famous Famiglia", "count" : 2 }
{ "_id" : "Pizza Hut", "count" : 2 }
{ "_id" : "Natale's Pizzeria", "count" : 1 }
{ "_id" : "Union Pizza Works", "count" : 1 }
{ "_id" : "Helen's Pizza", "count" : 1 }
{ "_id" : "Dentist", "count" : 1 }
{ "_id" : "Mini Munchies Pizza", "count" : 1 }
{ "_id" : "Tomato Pie", "count" : 1 }
{ "_id" : "Aurora Pizza", "count" : 1 }
{ "_id" : "Forino", "count" : 1 }
{ "_id" : "Brooklyn Central", "count" : 1 }
```

Poor coverage again. Surprisingly low counts for pizza chains and very few duplicate names (which is not the reality in NY).

Which city has most pizza joints (Top 20)?

```
> db.nyc.aggregate({$match: { cuisine: "pizza"}}, {$group: { _id:
"$address.city", count: { $sum: 1 }}}, { $sort: { count: -1 } })
{ "_id" : null, "count" : 74 }
{ "_id" : "New York", "count" : 44 }
{ "_id" : "Brooklyn", "count" : 15 }
{ "_id" : "West Milford", "count" : 2 }
{ "_id" : "Sunnyside", "count" : 2 }
{ "_id" : "Howard Beach", "count" : 1 }
{ "_id" : "Woodside", "count" : 1 }
{ "_id" : "Ridgewood", "count" : 1 }
```

```
{ "_id" : "Cos Cob", "count" : 1 }
{ "_id" : "Paramus", "count" : 1 }
{ "_id" : "Jersey City", "count" : 1 }
{ "_id" : "Waldwick", "count" : 1 }
{ "_id" : "Orange", "count" : 1 }
{ "_id" : "Belleville", "count" : 1 }
{ "_id" : "Jackson Heights", "count" : 1 }
{ "_id" : "Long Beach", "count" : 1 }
{ "_id" : "Forest Hills", "count" : 1 }
{ "_id" : "Kew Gardens", "count" : 1 }
{ "_id" : "Piscataway", "count" : 1 }
{ "_id" : "Astoria", "count" : 1 }
```

Again, the numbers are very low.

Other Ideas

In the first project, we looked at NYC subway data in the context of how weather affects ridership. It would be interesting to use the NYC ridership data in combination with the OSM data to create a visualization of how the entry/exit counts fluctuate at different stations throughout the day.

Again using the NYC subway data, it would be interesting to explore the relationship between ridership and presence of amenities near a certain station (e.g. shopping, food, commercial hubs, other public transport). Unfortunately, there might be issues with such an analysis due to the poor coverage of the area in terms of amenities.