**Map Area: Syracuse, NY, United States**
File size: 60.4 MB

**Problems Encountered:**

*1. Postal Codes*

Initially I found no strange postcode values because my screening function needed some improvements. Below I printed out all of the postcodes that contain a “problematic character” or that didn’t have the expected 5 digits. Some of the strings contained "-" and I realized I needed to add this to my list of “problematic characters” to search for. I noticed some zip codes were long (i.e. missing a "-") so for these I cleaned the data by taking only the first 5 digits. I also noticed that at least one postal code did not have the correct 4 extra digits after the "-" it had 3 extra. In all cases my final cleaned postcode only contained 5 digits with the entire string converted to int type.

Below are the unique post codes:

set(['13066', '13202-1107', **'13224-1110'**, '13507', '13207', '13206', '13205', '13204', '13203', '13202', '13209', '13208', '13022', '13108', '13120', '13244', '13104', '13220', '13204-1243', '13224', '13164', '13039', '13206-2238', **'132059211'**, '132179211', '13041', **'13219-331'**, '13290', '13088', '13210', '13211', '13212', '13214', '13215', '13210-1203', '13210-1053', '13078', '13219', '13090', '13031', '13152', '13027', '13116', '13218-1185', '13057'])

*2. Inconsistent Pharmacy Names*

I discovered inconsistencies with pharmacy names while doing some aggregation in MongoDB. This was after my initial data cleaning using my Python script. I remedied this problem by writing and implementing an additional pharmacy name cleaning function before outputting to the database. In MongoDB I was grouping pharmacies in Syracuse by their name to figure out which pharmacy was most popular. Below I have output my query results from MongoDB before and after data cleaning.

*******BEFORE I CLEANED*******
{ "count" : 11, "pharmacy" : "Kinney Drugs" }
{ "count" : 11, "pharmacy" : "Rite Aid" }
{ "count" : 8, "pharmacy" : "Rite Aid Pharmacy" }
{ "count" : 3, "pharmacy" : "Walgreens" }
{ "count" : 2, "pharmacy" : "Wegmans Pharmacy" }
{ "count" : 1, "pharmacy" : "Rite Aid 10733" }
{ "count" : 1, "pharmacy" : "Main Street Pharmacy of Marcellus" }
{ "count" : 1, "pharmacy" : "Manlius Pharmacy" }
{ "count" : 1, "pharmacy" : "CVS" }
{ "count" : 1, "pharmacy" : "Gifford & West Pharmacy" }
{ "count" : 1, "pharmacy" : "Harvey's Pharmacy" }
{ "count" : 1, "pharmacy" : "Rite-Aid" }
{ "count" : 1, "pharmacy" : "Tops Pharmacy" }
{ "count" : 1, "pharmacy" : "Price Chopper Pharmacy" }
{ "count" : 1, "pharmacy" : "Kinney Drugs Pharmacy" }
{ "count" : 1, "pharmacy" : "Kinney's" }

{ "count" : 1, "pharmacy" : "Etain" }
{ "count" : 1, "pharmacy" : "Rite Aid 10766" }
{ "count" : 1, "pharmacy" : "Kinney Pharmacy" }

*******AFTER I CLEANED*******
{ "count" : 22, "pharmacy" : "Rite Aid Pharmacy" }
{ "count" : 14, "pharmacy" : "Kinney Drugs" }
{ "count" : 3, "pharmacy" : "Walgreens" }
{ "count" : 2, "pharmacy" : "Wegmans Pharmacy" }
{ "count" : 1, "pharmacy" : "Main Street Pharmacy of Marcellus"}
{ "count" : 1, "pharmacy" : "Manlius Pharmacy" }
{ "count" : 1, "pharmacy" : "CVS" }
{ "count" : 1, "pharmacy" : "Harvey's Pharmacy" }
{ "count" : 1, "pharmacy" : "Tops Pharmacy" }
{ "count" : 1, "pharmacy" : "Price Chopper Pharmacy" }
{ "count" : 1, "pharmacy" : "Etain" }
{ "count" : 1, "pharmacy" : "Gifford & West Pharmacy" }

**Data Overview:**

Here I present basic statistics about the dataset and the MongoDB queries used to acquire them.

Size of File
syracuse_new_york.osm (60.4 MB)
syracuse_new_york.osm.json (67.8 MB)

Number of Documents
```
db.syracuse.find().count()
```
310954

Number of Nodes
```
db.syracuse.find({"type":"node"}).count()
```
275753

Number of Ways
```
db.syracuse.find({"type":"way"}).count()
```
35182

Number of unique users
```
len(db.syracuse.distinct("created.user"))
```
240

Top contributor list
```
db.syracuse.aggregate([{"$group": {"_id":"$created.user","count":{"$sum":1}}},
                       {"$sort":{"count":-1}},{"$limit":1}])
```

{ "_id" : "zeromap", "count" : 155672 }

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

```
{ "_id" : "zeromap", "count" : 155672 }
{ "_id" : "woodpeck_fixbot", "count" : 75649 }
{ "_id" : "DTHG", "count" : 27597 }
{ "_id" : "yhahn", "count" : 8144 }
{ "_id" : "RussNelson", "count" : 8073 }
{ "_id" : "fx99", "count" : 4499 }
{ "_id" : "bot-mode", "count" : 4428 }
{ "_id" : "timr", "count" : 2951 }
{ "_id" : "TIGERcnl", "count" : 2077 }
{ "_id" : "Johnc", "count" : 2037 }
{ "_id" : "ECRock", "count" : 1853 }
{ "_id" : "OSMF Redaction Account", "count" : 969 }
{ "_id" : "JessAk71", "count" : 925 }
{ "_id" : "zephyr", "count" : 806 }
{ "_id" : "NYSDEClands", "count" : 786 }
{ "_id" : "FrederickRelyea", "count" : 779 }
{ "_id" : "NE2", "count" : 754 }
{ "_id" : "D_S_W", "count" : 728 }
{ "_id" : "cjp", "count" : 702 }
{ "_id" : "mjpelmear", "count" : 583 }
```

Count the number of schools

```
db.syracuse.aggregate([{"$match":{"amenity":{"$exists":1},"amenity":"school"}},
                       {"$group": {"_id": "null", "count": {"$sum": 1 }}}])
```

```
{ "_id" : "null", "count" : 191 }
```

Count the number of shops

```
db.syracuse.aggregate([{"$match":{"shop":{"$exists":1}}},
                       {"$group": {"_id": "null", "count": {"$sum": 1 }}}])
```

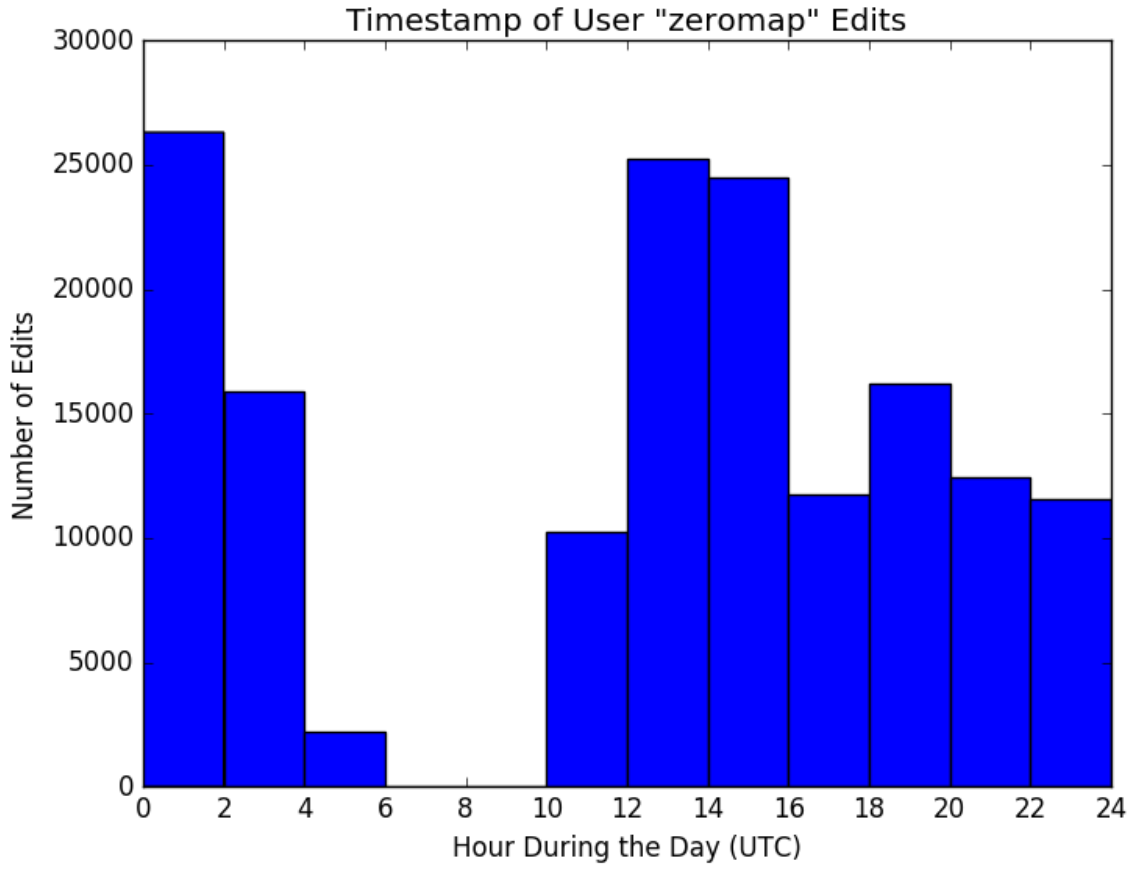
```
{ "_id" : "null", "count" : 858 }
```

**Additional Data Exploration:**

*1. Who are the contributors?*

It is clear that very few user names account for a huge proportion of the total number of contributions. The top 3 user names account for 83.1% of edits (see below). Automated routines (bots) offer a good explanation for this finding. It is interesting to look at the timestamp of the contributions from the most active user to decipher any temporal patterns in map editing. You can see after aggregating all of zeromaps contribution timestamps that this user's edits were not distributed evenly throughout the day. Most edits came between 0:00 and 02:00 UTC and between 12:00 and 16:00 UTC. No edits occurred between 06:00 and 10:00 UTC.

*TOP CONTRIBUTORS*
"mapzero" → 50% of the map contributions
"woodpeck_fixbot" → 24.3% of map contributions
"DTHG" → 8.8% of map contributions



*2. Top 20 amenities?*

```
db.syracuse.aggregate([{"$match":{"amenity":{"$exists":1}}},
                {"$group":{"_id":"$amenity", "count":{"$sum":1}}},
                {"$sort":{"count":-1}},{"$limit":20}])
```

```
{ "_id" : "parking", "count" : 922 }
{ "_id" : "school", "count" : 191 }
{ "_id" : "restaurant", "count" : 148 }
{ "_id" : "bench", "count" : 148 }
{ "_id" : "fast_food", "count" : 147 }
{ "_id" : "place_of_worship", "count" : 128 }
{ "_id" : "fuel", "count" : 116 }
{ "_id" : "bank", "count" : 63 }
{ "_id" : "post_box", "count" : 55 }
{ "_id" : "pharmacy", "count" : 49 }
{ "_id" : "bicycle_parking", "count" : 48 }
{ "_id" : "waste_basket", "count" : 37 }
{ "_id" : "cafe", "count" : 36 }
{ "_id" : "grave_yard", "count" : 36 }
{ "_id" : "library", "count" : 32 }
{ "_id" : "fire_station", "count" : 31 }
{ "_id" : "parking_entrance", "count" : 30 }
{ "_id" : "shelter", "count" : 29 }
{ "_id" : "toilets", "count" : 26 }
{ "_id" : "charging_station", "count" : 25 }
```

*3. What is the most popular pharmacy in Syracuse?*

```
db.syracuse.aggregate([{"$match":{"amenity":{"$exists":1},"amenity":"pharmacy"}},
                {"$group":{"_id":{"pharmacy":"$name"},"count":{"$sum":1}}},
                {"$project":{"_id":0,"pharmacy":"$_id.pharmacy","count":"$count"}},
                {"$sort":{"count":-1}}])
```

```
{ "count" : 22, "pharmacy" : "Rite Aid Pharmacy" }
{ "count" : 14, "pharmacy" : "Kinney Drugs" }
{ "count" : 3, "pharmacy" : "Walgreens" }
{ "count" : 2, "pharmacy" : "Wegmans Pharmacy" }
{ "count" : 1, "pharmacy" : "Main Street Pharmacy of Marcellus"}
{ "count" : 1, "pharmacy" : "Manlius Pharmacy" }
{ "count" : 1, "pharmacy" : "CVS" }
{ "count" : 1, "pharmacy" : "Harvey's Pharmacy" }
{ "count" : 1, "pharmacy" : "Tops Pharmacy" }
{ "count" : 1, "pharmacy" : "Price Chopper Pharmacy" }
{ "count" : 1, "pharmacy" : "Etain" }
{ "count" : 1, "pharmacy" : "Gifford & West Pharmacy" }
```

*4. Top 5 shop types?*

```
db.syracuse.aggregate([{"$match":{"shop":{"$exists":1}}},
                       {"$group":{"_id":{"Shop":"$shop"},"count":{"$sum":1}}},
                       {"$sort":{"count":-1}},
                        {"$limit":5}])
```

```
{ "_id" : { "Shop" : "convenience" }, "count" : 117 }
{ "_id" : { "Shop" : "car_repair" }, "count" : 56 }
{ "_id" : { "Shop" : "hairdresser" }, "count" : 54 }
{ "_id" : { "Shop" : "supermarket" }, "count" : 48 }
{ "_id" : { "Shop" : "clothes" }, "count" : 41 }
```

**Improving the data:**

I believe OSM's free tagging system that allows an unlimited number of attributes describing each feature needs to be more strictly managed. Formal standards should be set to improve the quality of information contained within the tag and to reduce inconsistencies and irregularities in the dataset. For example, during my data cleaning I found discrepancies in pharmacy naming that could be avoided if tighter constraints were implemented initially. These inconsistencies are an example of tags being unverifiable. One user may name a pharmacy "Rite-Aid" while another names the same pharmacy "Rite Aid Pharmacy". The name of a pharmacy on a map should be the full name provided by the company (i.e. Rite Aid Corporation) or should reflect an agreed upon name by the OSM community. I recommend a small group of individuals to be tasked with this. A list of acceptable pharmacy names, grocery store names, restaurants, etc. should be established for each city (or map region) and all "name" attributes should be vetted against these standard lists. This proposed change would remove ambiguity, facilitate aggregation queries for certain map regions, and reduce the need for extensive data cleaning. One potential shortfall I see in having lists of acceptable names for certain nodes is that a new business, restaurant, etc. could pop up and have no name stored yet. In this case, the new pharmacy name would need to be first passed to the OSM group tasked with maintaining proper naming standards. Maintaining this group could potentially be costly in the long-run if individuals do not wish to volunteer.