OpenStreetMap Project
Data Wrangling with MongoDB

By: Jacob Kreisler

Map area: Chicago, IL

https://www.openstreetmap.org/relation/122604
https://mapzen.com/data/metro-extracts/metro/chicago_illinois/

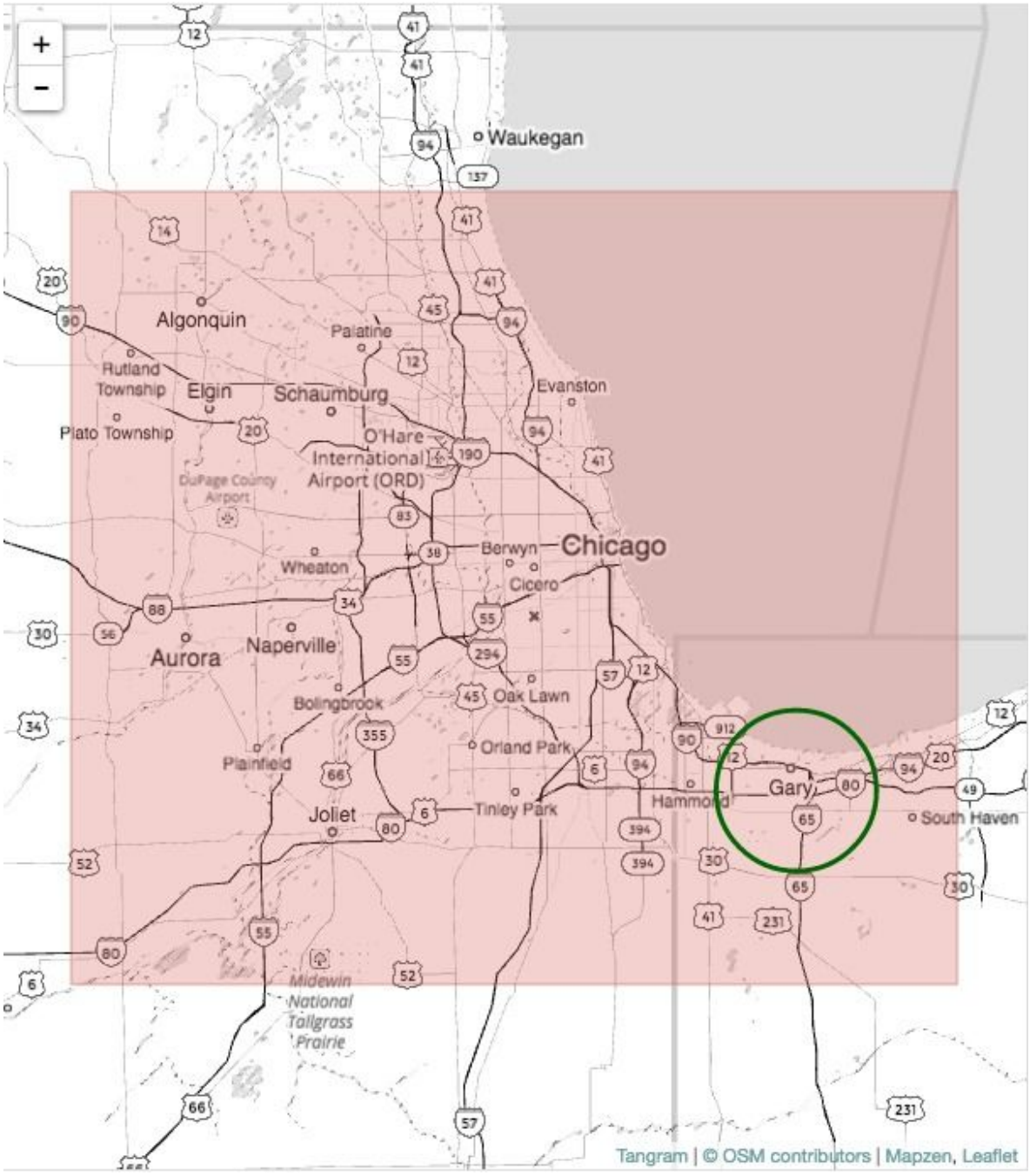
1. Problems Encountered in the Data

To begin, I created a sample of the larger map dataset (2.02 GB), and was then able to skim through the xml to get a better sense of how the information was organized. After running a query that returned tag and attribute values, three problems were revealed:

- **Abbreviated Street Names**
  One of the first problems that surfaced was inconsistencies in how street names were written. Some addresses were abbreviated, while others were not. For example, “Kostner Ave” compared to “Sherman Avenue.” To fix this, I defined a function that would take the abbreviations and extend them out. “Koster Ave” now became “Koster Avenue.”

```
def update_name(name, mapping):
    m = street_type_re.search(name)
    other_street_types = []
    if m:
        street_type = m.group()
        if street_type in mapping.keys():
            name = re.sub(street_type_re,mapping[street_type],name)
        else:
            other_street_types.append(street_type)
    return name
```

- **Map Coverage Area**
  While working on the postal code query, I noticed that several postcodes started with the number “46 - - - .“ Taking another look at the coverage area, you can see that the bottom right corner actually includes Indiana, as places like Gary, Indiana are often considered part of the Greater Chicago area. Because of this, I decided to leave these postcodes in the dataset, rather than filter out any city not named “Chicago.”



- **Postal Code Length**
  The query performed revealed postcodes written as a “ZIP +4 Code,” which included a “-” followed by four additional digits. To standardize these instances, I used a function that produces a standard 5 digit zip:

```
def update_postcode(postcode):
    match = re.match(r'^\D*(\d{5}).*', postcode)
    clean_postcode = match.group(1)
    return clean_postcode
```

- **Including State in City Name**
  Querying for city names revealed that many locations were tagged under “Chicago, IL” rather than “Chicago.” To correct for this, I added the following code:

```
if tag.attrib['v'] == "Chicago, IL":
    tag.attrib['v'] = "Chicago"
```

2. Data Summary

File sizes:
sample.osm . . . . . 204.5 MB
sample.osm.json 289.7MB

Number of documents:

```
> db.sample.find().count()
        1,691,831
```

Number of nodes:

```
> db.sample.find({"type": "node"}).count()
        1,685,558
```

Number of ways:

```
> db.sample.find({"type": "way"}).count()
        6,273
```

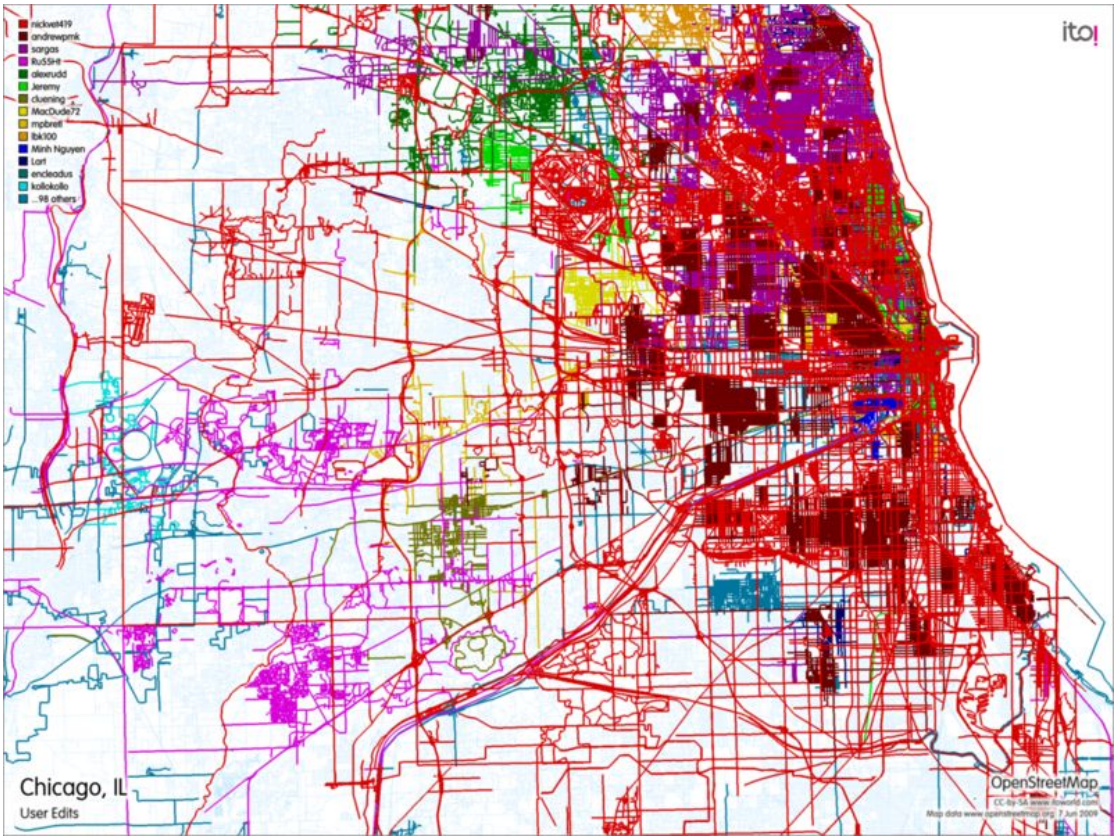
Top five contributing users:

```
> db.sample.aggregate(pipeline)

> pipeline = [{"$match":{"created.user":{"$exists":1}}},
              {"$group":{"_id":"$created.user", "count":{"$sum":1}}},
              {"$sort":{"count":-1}},{"$limit":5}]

                     [{u'_id': u'chicago-buildings', u'count': 968262},
                      {u'_id': u'Umbugbene', u'count': 205895},
                      {u'_id': u'alexrudd (NHD)', u'count': 45472},
                      {u'_id': u'woodpeck_fixbot', u'count': 44924},
                      {u'_id': u'TIGERcnl', u'count': 20750}]
```

The highest contributing user, “chicago-buildings” dominates the top five, with 75% of contributions. Due to the considerable difference between the users, I suspect the data for “chicago-buildings” is being updated automatically rather than manually.

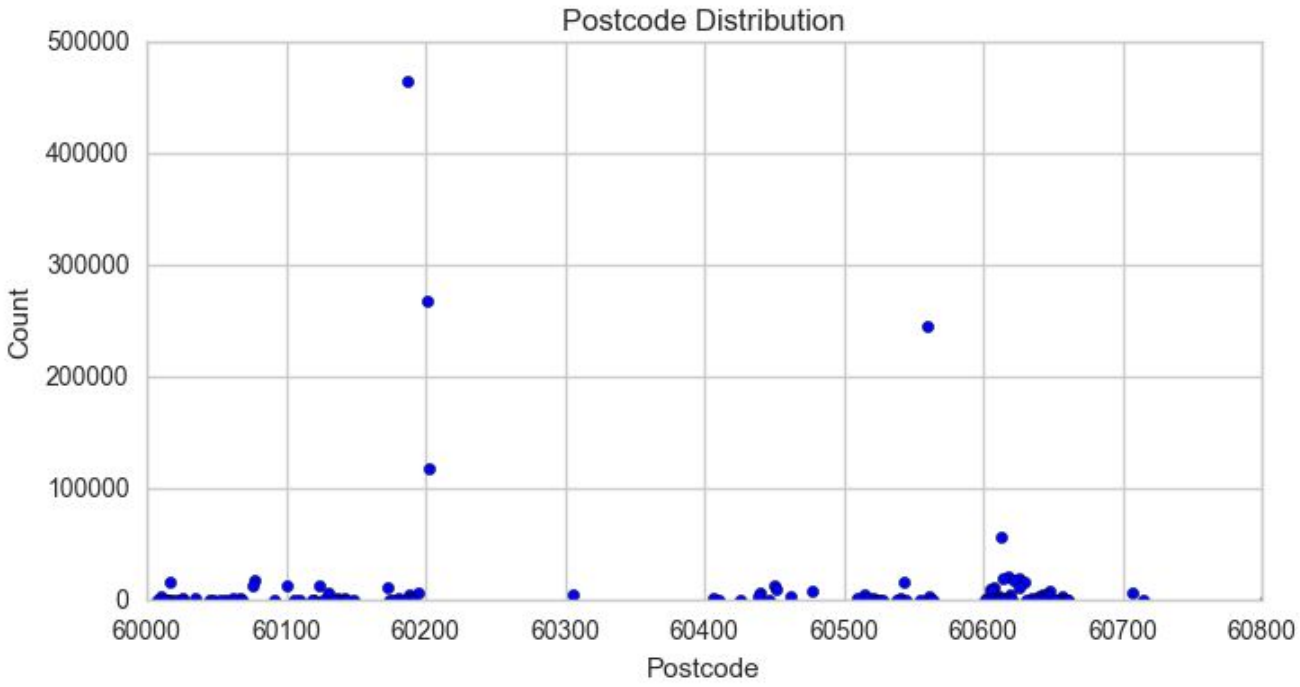


This image shows user contributions in Chicago. Each color represents a different user. Our third highest user, alexrudd, can be seen represented as dark green in the image.

-http://wiki.openstreetmap.org/wiki/File:OSMMapperChicagoUsers.png

3. Geographical Insights

Charting the postcodes (ex- Indiana) shows that the data represents a large coverage area. Although many postcodes are represented in the dataset, a few are represented substantially more:



Top five postcodes tagged:

```
> db.sample.aggregate(pipeline)
> pipeline = [{"$match":{"address.postcode":{"$exists":1}}},
              {"$group":{"_id":"$address.postcode", "count":{"$sum":1}}},
              {"$sort":{"count":-1}},{"$limit":5}]

                    [{u'_id': u'60187', u'count': 465764},
                    {u'_id': u'60201', u'count': 268338},
                    {u'_id': u'60560', u'count': 245802},
                    {u'_id': u'60202', u'count': 117848},
                    {u'_id': u'60613', u'count': 56422}]
```

Top five neighborhoods tagged:

```
> db.sample.aggregate(pipeline)
> pipeline = [{"$match":{"address.city":{"$exists":1}}},
              {"$group":{"_id":"$address.city", "count":{"$sum":1}}},
              {"$sort":{"count":-1}},{"$limit":5}]

                     [{u'_id': u'Evanston', u'count': 1111476},
                     {u'_id': u'Chicago', u'count': 251007},
                     {u'_id': u'Skokie', u'count': 34520},
                     {u'_id': u'Woodridge', u'count': 23056},
                     {u'_id': u'Schaumburg', u'count': 20238}]
```

Top five amenities tagged:

```
> db.sample.aggregate(pipeline)
> pipeline = [{"$match":{"amenity":{"$exists":1}}},
              {"$group":{"_id":"$amenity", "count":{"$sum":1}}},
              {"$sort":{"count":-1}},{"$limit":5}]

                     [{u'_id': {u'amenity': u'restaurant'}, u'count': 247434},
                      {u'_id': {u'amenity': u'fuel'}, u'count': 239436},
                      {u'_id': {u'amenity': u'atm'}, u'count': 72034},
                      {u'_id': {u'amenity': u'fast_food'}, u'count': 58023},
                      {u'_id': {u'amenity': u'parking'}, u'count': 44115}]
```

Top five cuisines tagged:

```
> db.sample.aggregate(pipeline)
> pipeline = [{"$match":{"cuisine":{"$exists":1}}},
              {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},
              {"$sort":{"count":-1}},{"$limit":5}]

                     [{u'_id': {u'cuisine': u'burger'}, u'count': 266045},
                     {u'_id': {u'cuisine': u'sandwich'}, u'count': 220459},
                     {u'_id': {u'cuisine': u'american'}, u'count': 160037},
                     {u'_id': {u'cuisine': u'mexican'}, u'count': 25811},
                     {u'_id': {u'cuisine': u'pizza'}, u'count': 19777}]
```

4. Conclusion

I would argue that the most important aspect of this type of project is to have an abundance of accurate, current data. So, as others have suggested, I believe gamification would be a great way to incentivize users to contribute more frequently. An example of this is Waze, a “community-based traffic and navigation app,” Waze relies on its users to contribute information in real time, sharing anything from traffic updates, police sightings, and even the latest gas prices. These reports are substantiated by other users and the user that created the report is rewarded with points that allows them to climb up in rank. The social aspect of the app adds an element that makes sharing data fun and interactive, but perhaps it creates a weakness as well. Due to the competitive nature of earning points and climbing in rank, some users may be tempted to abuse the reporting system for personal gain, adding inaccurate reports that cannot be discredited in time.