OpenStreetMap Data Case Study

Map Area

Phoenix, AZ, United States

• https://www.openstreetmap.org/export#map=12/33.5184/-112.0623

This map is of my hometown, so I'm more interested to see what database querying reveals, and I'd like an opportunity to contribute to its improvement on OpenStreetMap.org.

Problems Encountered in the Map

After initially downloading a small sample size of the Phoenix area and running it against a provisional data.py file, I noticed a couple main problems with the data, which I will discuss in the following order:

• Overabbreviated street names ("N. Cave Creek Rd.")

• Inconsistent postal codes ("85032-7702", "850822", "AZ 85004")

Overabbreviated Street Names

After auditing the data and cleaning it up, it was imported into a sqlite database. Some basic querying revealed street name abbreviations and postal code inconsistencies. I used regular expressions to handle correcting street names. See the following function:

```
def update_street_name(name, mapping):

    name = name.replace('  ', ' ')
    m = street_type_re.search(name)
    if m:
        street_type = m.group()
        if street_type in mapping.keys():
            name = re.sub(street_type, mapping[street_type], name)

    return name
```

This updated all substrings in the audited (problematic) address strings. Example: "N. Cave Creek Rd." becomes "North Cave Creek Road"

Postal Codes

Due to the nature of the postcodes, I was able to use a regular expression that only kept the main 5 digit code of the zip, stripping anything after the 5th digit, including "-####", extra digits, and anything before the 5 digits, like "AZ ". Here's the function used for it:

```
def update_postcode(postcode):

  search = re.match(r'^\D*(\d{5}).*', postcode)
  # select the group that is captured
  if search:
    clean_postcode = search.group(1)
    return clean_postcode
  else:
    pass
```

Overview Statistics

With the cleaned csv files imported into a database, I ran a number of queries in SQL to gain a better understanding of the dataset.

• Size of the file

• Number of unique users

• Number of nodes and ways

• Number of chosen type of nodes, like cafes, shops etc.

Size of the file

map.osm ..... 72.7 mb

map.db ..... 41.1 mb

new_nodes ..... 24.4 mb

new_nodes_tags ..... 2.44 mb

new_ways ..... 2.45 mb

new_ways_nodes ..... 8.15 mb

new_ways_tags ..... 7.47 mb

Number of Unique Users

```
SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM Nodes UNION ALL SELECT uid FROM Ways) e;
```

374

Number of Nodes

```
SELECT COUNT(*) FROM nodes;
```

302078

Number of Ways

```
SELECT COUNT(*) FROM ways;
```

42840

Top 10 Amentities

```
SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10;
```

place_of_worship 238

bench 227

waste_basket 174

restaurant 153

fast_food 129

school 113

fuel 104

swimming_pool 64

bicycle_rental 50

vending_machine 43

Top 10 Contributing Users

```
SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
ORDER BY num DESC
LIMIT 10;
```

Dr Kludge 232965

TheDutchMan13 32230

rayn 15732

dwh1985 7461

namannik 5222

Sundance 3251

Your Village Maps 2597

adenium 2511

ShatteredArm 2475

DesertNavigator 1977

Number of Users Appearing Only Once (Having 1 post)

```
SELECT COUNT(*)
FROM
  (SELECT e.user, COUNT(*) as num
   FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
   GROUP BY e.user
   HAVING num=1)  u;
```

92

Additional Ideas

Contributor statistics and usage suggestion / potential problems with suggestion

The contributions of users seems incredibly skewed, possibly due to automated versus manual map editing. Here are some user percentage statistics:

• Top user contribution percentage (“DrKludge”) 77.12%

• Combined number of users making under 1% contribution of posts 368 (about 98% of all users)

Given how skewed the user percentage distributions are, a more consistent participation across the board of users would help level set the data. For this particular region, the "DrKludge" user made the most contributions of map data. Obviously, this user does not need any sort of additional motivation - it's likely they fall under the demographic of the user community highlighted on the OpenStreetMap site - of the enthusiast mapper, engineers running the OSM servers, etc. For everyone else, incentives such as a leaderboard amongst their peers could be helpful in sparring competition and lead to the creation of more efficient bots, where more than 2% of the users make contributions of at least 1%.

A recently popular app – Pokémon Go – has mastered this gamification concept by exploiting one of the most addicting games in game history. Players are incentivized by leveling up their Pokémon, finding new exotic ones, hatching eggs, and battling at the gym. I think integration of the OSM database with an app such as this one would help lead to improvisations in the data. Another idea would be to create a running app that gives users rewards for the more area they cover in their routes. Perhaps, users can stack up or collect resources as they run, then create / upgrade a central base and battle it out with other users of the app. An Age of Empires / StarCraft spin to it. All the while, generating more accurate data for the OpenStreetMap.org.

While the phone apps would be a nifty approach towards getting people to generate more data for the OSM project, it also runs the risk of inaccurate input. However, I believe a validation system could be put into place to secure accurate data entry. Example, someone running with the run app open passes an In n Out Burger. They input American cuisine for the restaurant category. It’s validated when another person either likes this entry or puts in an entry of their own that matches the same info.

Additional Data Exploration

Taking a deeper dive into the top 10 amenities list, here are some of the most popular values for their respective keys:

Most Popular Cuisine

```
SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
  JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='restaurant') i
  ON nodes_tags.id=i.id
WHERE nodes_tags.key='cuisine'
GROUP BY nodes_tags.value
ORDER BY num DESC
LIMIT 1;
```

mexican 18

If you've ever been to Arizona, I recommend a diet consisting of tacos and burritos. Yet, this is still a surprise in that there aren't any other cuisines in the dataset. This again ties back to the skewed nature of these OSM datasets. With better data sources this would likely be a different cuisine list in the future as I'm sure there are other types of restaurants in the area.

Most Popular Religion (in this region)

```
SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
  JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='place_of_worship') i
  ON nodes_tags.id=i.id
WHERE nodes_tags.key='religion'
GROUP BY nodes_tags.value
ORDER BY num DESC
LIMIT 1;
```

christian 229

Conclusion

After reviewing the data of my hometown in Phoenix, AZ, it's evident the data isn't entirely complete. Although for the means of this project, it's ideal for allowing the auditing / cleaning up of plenty of data. I think a combination of more user input for less-skewed user contributions and a better protocol for actively cleaning the data would help make OpenStreetMap.org a more robust and reliable dataset.