OpenStreetMap Sample Project
Data Wrangling with SQL

Wafic Fahme

Map Area: Greater London, United Kingdom

*https://*mapzen.com/data/metro-extracts/your-extracts/c14328a6d43e

1. Problems Encountered in the Map

After requesting and downloading the data set from mapzen, I extracted a sample and noticed the below

- Abbreviation of Saint to St in addr:Street
- Tags with key 'fix me' to be ignored
- Inconsistent Home numbers with comma and hyphens, convert comma to hyphen
- In Wikipedia, remove additional character 'en:' from string

Then I went to fix the above as detailed below but then again I noticed 2 issues when querying the SQL which I was intimidated to neglect but couldn't, among these issues that didn't show at the beginning of wrangling this data set was the below 2 issues:

- I can see lots of "randomjunk_bot" in keys which I need to drop from my datasets.
- I can see that there is a lot of "en" in keys which causes the key to be useless

Abbreviation of Saint to St in addr:Street:
Lots of the street have 'St' in the beginning and with some research, I noticed that this is not abbreviation of street but of Saint, so I wrote a python script that search for all abbreviations of Saint and replaced it with the word "Saint" as clear by the below example:
<tag k="addr:street" v="St Andrew's Hill" /> became <tag k="addr:street" v="Saint Andrew's Hill" />

Tags with key 'fix me' to be ignored:
Too many tags where left with the title "fixme" by users, when looking at these tags, they looked useless for that I ran a python script that dropoed them from our dataset

Inconsistent Home numbers with comma and hyphens, convert comma to hyphen:
Home numbers are a tricky for someone living in Beirut as we don't number our houses, then I noticed that the format was mostly alphanumerical "11A", but then saw some houses with this type of number "111-117", which is mostly a company that is hosted at several combined houses, here I saw many houseaddresses like Swarovski store that had this wrong format "137,139", for that I wrote a python script to replace the comma with hyphen and make it consistent looking like "137-139"

In Wikipedia, remove additional character 'en:' from string:
In all data coming from wikipedia, I noticed that they had the word "en:" which was useless, so I

wrote a script that removed first 3 characters from any Wikipedia entry converting "en:Memorial to the Great Exhibition" to "Memorial to the Great Exhibition"

I can see lots of "randomjunk_bot" in keys which I need to drop from my datasets.
Lots of users used bots whom generated junk like the above string making the entire entry useless, so I dropped the entire record as it hold little to no value.

I can see that there is a lot of "en" in keys which causes the key to be useless:
This was a bit an error generated by my own cleaning method, because whenever there is a colon in the tag, the python script will consider the first to be the key and the second to be the type of the key but when the tag is “species:en”, the key becomes “en” and the type which doesn’t make sense, for that I wrote a script that will break the normal rule when the key is an “en”, making the key the second value after the colon and the type “regular.

2. Data Overview

This section contains basic statistics about the dataset and the SQL queries used to gather them.

File sizes

greater_london.osm ..... 52 MB
greater_london.db ..... 29 MB
nodes.csv ..... 16 MB
ways.csv ..... 22 MB
nodes_tags.csv ..... 38 MB
ways_tags.csv ..... 4 MB
ways_nodes.csv ..... 6 MB

Number of nodes:

```
cur.execute('SELECT count(*)\
        FROM nodes')
cur.fetchall()
```

Result: 19,7114 nodes

Number of ways:

```
cur.execute('SELECT count(*)\
        FROM ways')
cur.fetchall()
```

Result: 37,393 ways

Number of unique node tags types:

```
cur.execute('SELECT COUNT(DISTINCT(key))\
        FROM nodes_tags')
```

```
cur.fetchall()
```
Result: 468 Unique nodes

Number of unique node tags types:

```
cur.execute('SELECT COUNT(DISTINCT(key))\
        FROM ways_tags')
cur.fetchall()
```
Result: 563 unique node tags

Unique number of types of Amenities

```
cur.execute('SELECT value\
        FROM nodes_tags\
        WHERE key = "amenity"\
        GROUP BY value')
len(cur.fetchall())
```
Result: 104 Unique Amenities

3. Additional Ideas

Top 10 Amenities in nodes:

Below I extracted the top 10 amenities in the greater London area and it was a bit suprising to see bikes parking at the top spot as visible below

1. bicycle_parking: 507
2. bench: 466
3. restaurant: 368
4. post_box: 298
5. telephone: 290
6. cafe: 270
7. waste_basket: 248
8. pub: 166
9. fast_food: 151
10. atm: 103

Top 10 Amenities in ways:

1. parking: 270
2. restaurant: 225
3. cafe': 167
4. pub: 136
5. school: 126
6. place_of_worship: 114
7. fast_food: 91
8. bank: 35
9. bar: 31
10. community_centre: 27

Historic categories and numbers available in Greater London:

This is a good indicator how ancient this city is and how diverse.

1. archaeological_site: 2

2. blue_plaque: 4
3. cannon: 2
4. castle: 14
5. citywalls: 3
6. fence: 1
7. footway: 1
8. icon: 1
9. industrial: 1
10. memorial: 79
11. monastery: 1
12. monument: 13
13. police_telephone: 1
14. relic: 1
15. retaining_wall: 4
16. roman_road: 22
17. ruins: 4
18. ship: 2
19. yes: 3

Top 10 contributing users

1. Tom Chance: 23100
2. Paul The Archivist: 16674
3. Amaroussi: 13199
4. Ed Avis: 12128
5. Derick Rethans: 10707
6. abc26324: 9941
7. peregrination: 8529
8. Harry Wood: 7314
9. ecatmur: 6361
10. moyogo: 5543

 # Total number of entries recorded in this database

234507

 # Total number of entries recorded in this database from top 10 contributors

113496

 # Total number of users

1940

Conclusion on this section

Looking the above data and knowing that we have

- Total entries 234,507 entries
- 113,496 of 234,507 are from the top 10 contributors which is almost 50% of the entries From the total number of users:
- 10 are contributing 50% of data
- 1930 are contributing the resulting 50%

Types of buildings in Greater London:

- ETON_PLACE: 1
- AIR_SHAFT: 14
- APARTMENTS: 1077
- BANDSTAND: 2
- BLOCK: 5
- BOAT: 1
- BRIDGE: 1
- BUS_GARAGE: 1
- CASTLE: 13
- CHAPEL: 1
- CHIMNEY: 1
- CHURCH: 34
- CIVIC: 1
- CLOCK_TOWER: 1
- CLUBHOUSE: 1
- COLLEGE: 4
- COMMERCIAL: 219
- CONSTRUCTION: 9
- CONVENT: 1
- COUNCIL_FLATS: 3
- DATA_CENTER: 3
- DEPOT: 1
- DORMITORY: 1
- FACADE: 1
- FACULTY: 4
- FARM: 1
- FARM_AUXILIARY: 1
- FERRY_TERMINAL: 1
- FLATS: 24
- GALLERY: 1
- GARAGE: 39
- GARAGES: 42
- GASOMETER: 2
- GRANDSTAND: 1
- GREENHOUSE: 3
- HALL_OF_RESIDENCE: 1
- HOSPITAL: 12
- HOTEL: 15
- HOUSE: 3511
- HOUSES: 2
- HUT: 6
- INDUSTRIAL: 69
- KINDERGARTEN: 1
- KIOSK: 1
- LIGHT_INDUSTRIAL: 1

- MOSQUE: 1
- MULTIPLE: 1
- NO: 14
- OFFICE: 77
- OFFICES: 3
- OUTBUILDING: 23
- PART: 12
- PAVILION: 1
- PLACE_OF_WORSHIP: 1
- PORTACABIN: 1
- PRISON: 1
- PUB: 3
- PUBLIC: 7
- RESIDENTIAL: 1342
- RETAIL: 228
- ROOF: 37
- SCHOOL: 70
- SEMIDETACHED_HOUSE: 2
- SERVICE: 1
- SHED: 11
- SHIP: 2
- SHOP: 7
- STADIUM: 1
- STATION: 16
- STUDENT_ACCOMODATION: 1
- STUDENT_RESIDENCE: 1
- SUBSTATION: 1
- TERRACE: 361
- TOWER: 6
- TRAIN_STATION: 14
- TUNNEL_ENTRANCE: 1
- TUNNEL_SHAFT: 1
- UNIVERSITY: 24
- UTILITY: 1
- VIADUCT: 7
- VILLAGE_HALL: 1
- WAREHOUSE: 3
- YES: 11961

We have 19,379 entries with key building, but if we look closer we will notice that out of these records, 11,961 (almost 60%) are labeled "yes" which is good but not good enough knowing that the data can be more specific
An improvement that can be made here is be more specific on the type of the building especially for tourists who might want to know if this building is a hotel or residential.

Top 10 type of shops in Greater London:

1. CLOTHES: 152
2. CONVENIENCE: 151

3. HAIRDRESSER: 78
4. SUPERMARKET: 45
5. ESTATE_AGENT: 42
6. JEWELRY: 33
7. BEAUTY: 31
8. DRY_CLEANING: 31
9. NEWSAGENT: 31
10. FURNITURE: 29

Looking at the results, the data looks logical knowing that Greater London is renowned for its shopping experience and due to being the most expensive real estate in EU, having estate agent shops makes a lot of sense.

4. Additional Ideas

For a person in the loyalty industry I believe a rewarding the users can be the best idea to better enhance the data quality, we've seen it in other industries especially the airlines industry where people tend to be very price sensitive. Looking above we notice that the majority of the contributers on this dataset are less than maybe 30 users, for that we can use a rewarding system that might reward them points for their contributions on this issue and other issues.

This rewarding programs can reward these users to abide by strict rules when it comes to updating the maps rather than using broken bots or other quick fixes, where they can get a point and when accumulating several points they can redeem them for gadgets which openstreet project can fund by asking some specific suppliers (like go pro and Trek) whom will be more than glad to have access to such audience.

The power of such an approach has 2 major benefits, first openstreet project will be appearing for the self interest of every individual where every user will feel that he is getting something more than normal in return and most important is that this will create some sort of competition between top-noch users thus creating a culture where its about the quality of the work what will count.

Some of the problems in implementing such a system can be with gold diggers whom might work hard till they get the reward and then disappear forever causing the map to become outdated.

5. Conclusion

I believe the data is not finalized and I don't think it will ever be 100% ready as street, shops houses and others are in continuous change especially at big congested cities like London. I think they need to be more specific on lots of issues like the type of building and shops and better use bots, but then we won't have anything to work on as data analysts. I honestly think clean data is something that will never happen and there will always be error and that is why we learned data wrangling.