OpenStreetMap Case Study - London, UK

The location to be studied is London, UK and nearby areas, which I am particularly interested in as I plan to move there in the near future.

***All values used in this case study are from my data sample and not the full data set.*

To view the area, select:

https://www.openstreetmap.org/relation/65606
http://metro.teczno.com/#london

Problems Encountered in the Map

After performing an analysis on a sample of the data, I have encountered the following issues:

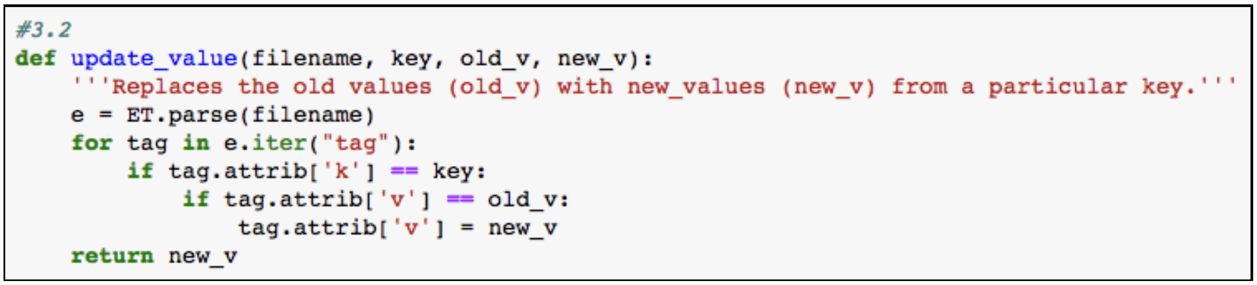
1. Most of the cities are not London, and some of London's neighbourhoods and boroughs are listed as cities.
2. Speed limits are improperly formatted.
3. Some postal codes are improperly formatted.
4. There are variations in the spelling of street types.
5. The sources of information was very inconsistent.

Cities:
The data that was used for this case study included cities surrounding London, so it was not surprising to find other cities, but I was surprise by the ratio of other cities to London.

Top 5 cities in sample:

Reading	102
London	39
Swanley	18
Maldon	10
Horsham	9

The neighbourhoods of Twickenham and Wembley, as well as the boroughs of Lewisham and Greenwich were listed as cities. I changed their values to London, using the function below, because London is the city where the node resides in.



```
#3.2
def update_value(filename, key, old_v, new_v):
    '''Replaces the old values (old_v) with new_values (new_v) from a particular key.'''
    e = ET.parse(filename)
    for tag in e.iter("tag"):
        if tag.attrib['k'] == key:
            if tag.attrib['v'] == old_v:
                tag.attrib['v'] = new_v
    return new_v
```

**I used this function to update values for speed limits, postal codes, and sources of information, in addition to cities.*

Speed Limits:
The typical format for the speed limit is '30 mph.' There were variations, such as leaving out the 'mph,' not putting a space between the number and mph, and having very improbably speed limits, such as '129.'

Postal Codes:
Many of the postal codes were correctly formatted, but "RG291AL" and "TW196AQ" needed to have spaces added: "RG29 1AL," "TW19 6AQ." Other postal codes were incomplete, such as "E3," "HA4," "W5," etc. They were left as is, as they provide some information about the node/ way's location.

Sources of Information:
Although many nodes and ways did not have a source, the ones that did came with great variations. 'Bing' for example, came in the following forms: 'Bing (2015-12-16),' 'Bing 2012,' 'bing,' and 'Bing.' This and many other sources were rewritten to create more homogeneity.

Street Types:
Although most of the street types were consistent, there was one case where 'Ave' needed to be corrected to Avenue.

Overview of the Data

Size of Files

sample_london.osm	5.2 MB
london_england.osm	2.55 GB
nodes_tags.csv	142 KB
nodes.csv	1.9 MB
ways_nodes.csv	654 KB
ways_tags.csv	311 KB
ways.csv	192 KB

Number of Nodes

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

22638

Number of Ways

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

3238

Number of Unique Users

```
sqlite> SELECT COUNT(DISTINCT(sources.uid))
   ...> FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways)
sources;
```

1415

Other Ideas about the datasets

Top 10 Users

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

```
941 - ”Eriks Zelenka”
927 - ”The Maarssen Mapper”
892 - TimSC_Data_CC0_To_Andy_Allan
743 - ca_hoot
734 - busdoc
733 - Johnmb
608 - Essex_Boy
524 - DanGregory
476 - c2r
409 - “Tom Chance”
```

Contribution Percentages

3.63% - Top user
7.21% - Top 2 users
16.35% - Top 5 users
26.96% - Top 10 users
50.20% - Top 32 users
***See python code for this formula, under #1.1 - #1.3*

Number of Users with 1 Post

```
sqlite> SELECT COUNT(*) FROM
   ...> (SELECT sources.user, COUNT(*) as num
   ...> FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways)
        sources
   ...> GROUP BY sources.user
   ...> HAVING num = 1) a;
```

620 - Number of users

The contributions made to this dataset are very skewed. Although there is not a ‘main’ user who is providing most of the information, there is a group of 32 users, or 2.3% percent of users, providing half the information.

Cities

```
sqlite> SELECT value, COUNT(*) as num
   ...> FROM (SELECT * FROM nodes_tags UNION ALL
   ...> SELECT * FROM ways_tags) tags
   ...> WHERE tags.key = 'city'
   ...> GROUP BY tags.value
   ...> HAVING num > 1
   ...> ORDER BY num DESC;
```

102 - Reading
45 - London
18 - Swanley
10 - Maldon
9 - Horsham
5 - Walthamstow
4 - Burnham-On-Crouch
3 - Colchester
3 - Crawley
3 - Luton
3 - Redhill
2 - Chelmsford
2 - Heybridge
2 - Leybourne
2 - Rochester
2 - Slough
2 - "St. Albans"
2 - Woking

Despite London being the primary focus of this case study, Reading is sourced as a city more than twice as often as London. Perhaps London was referenced more regularly in the relations tags.

Amenities

```
sqlite> SELECT value, COUNT(*) as num
   ...> FROM nodes_tags
   ...> WHERE key = 'amenity'
   ...> GROUP BY value
   ...> HAVING num > 1
   ...> ORDER BY num DESC;
```

33 - post_box
19 - bench
17 - pub
14 - bicycle_parking
8 - telephone
7 - parking
6 - restaurant
5 - cafe
5 - waste_basket
4 - atm
3 - bank
3 - fuel
3 - pharmacy
3 - place_of_worship
2 - college
2 - fast_food
2 - school

I thought it was very interesting to see what was referenced most frequently. Since we are looking at an area in the UK, it is not surprising to see ‘pub’ so often, but I am a little surprised to see school and restaurant referenced as infrequently as they are. Perhaps some people called what could be a pub/restaurant a pub.

Types of Nature

```
sqlite> SELECT value, COUNT(*) as num
   ...> FROM (SELECT * FROM nodes_tags UNION ALL
   ...> SELECT * FROM ways_tags) tags
   ...> WHERE tags.key = 'natural'
   ...> GROUP BY tags.value
   ...> ORDER BY num DESC;
```

197 - tree
85 - wood
23 - water
7 - scrub
5 - tree_row
1 - coastline
1 - grassland
1 - marsh
1 - peak

Relative to the amenities’ values, it is interesting to see how often types of nature are cited. The number of trees, for example, outnumber the total amount of amenities that were sourced.

Types of Leisure

```
sqlite> SELECT value, COUNT(*) as num
   ...> FROM (SELECT * FROM nodes_tags UNION ALL
   ...> SELECT * FROM ways_tags) tags
   ...> WHERE tags.key = 'leisure'
   ...> GROUP BY tags.value
   ...> ORDER BY num DESC;
```

25 - pitch
9 - garden
9 - playground
6 - park
2 - golf_course
2 - swimming_pool
1 - club
1 - common
1 - nature_reserve
1 - recreation_ground
1 - stadium

No big surprises here, given that this data is about London and area, I expected to see a number of pitches. I also thought that there would have been references to ‘alternative sports,’ such as bowling, or (ice or field) hockey.

Most common sport

```
sqlite> SELECT value, COUNT(*) as num
   ...> FROM (SELECT * FROM nodes_tags UNION ALL
   ...> SELECT * FROM ways_tags) tags
   ...> WHERE tags.key = 'sport'
   ...> GROUP BY tags.value
   ...> ORDER BY num DESC;
```

6 - soccer
5 - tennis
2 - golf
1 - basketball
1 - bowls
1 - cricket
1 - multi
1 - sailing
1 - skateboard

As expected, soccer is the most common sport listed. Somewhat surprisingly, is how close tennis is to soccer, and that basketball is listed, despite there not being a ‘court’ or something similar in the leisure category. The leisure key for basketball is pitch.

Common Speed Limits

```
sqlite> SELECT value, COUNT(*) as num
   ...> FROM (SELECT * FROM nodes_tags UNION ALL
   ...> SELECT * FROM ways_tags) tags
   ...> WHERE tags.key = 'maxspeed'
   ...> GROUP BY tags.value
   ...> ORDER BY num DESC;
```

127 - “30 mph"
72 - “20 mph"
24 - “40 mph"
16 - “60 mph"
10 - “70 mph"
7 - “50 mph"
2 - “10 mph”
2 - “5 mph"
2 - “90 mph"
1 - “45 mph”
1 - “25 mph”
1 - 129

Given that most of the data comes from urban areas, it makes sense that the most common speed limit is 30 mph. I was surprised to see 90 mph as a value, but after looking into it, that is the speed limit for two segments of the railway.

Ideas to Improve the Data

1. Include a ‘borough’ key.

Much like how there is a key for the city or street, there could also be a key for which borough an object resides in. Since London, and other cities, is divided into boroughs that help to provide information about location, this should be useful for users of the data. Given that not all cities have boroughs or are divided into subsegments, this would not be useful for all cities. In addition, if an object is near the boundary of two boroughs, a user could make the mistake of recording an object in the wrong borough.

2. Add a verification tags.

Similar to how skills can be verified by connections on LinkedIn, nodes, tags, and relations could be verified by other users via a tag. If a user agrees that all the tags about a node, tag, or relation are correct, they could add their user id to a verification tag. This should help to make more of the data more reliable as it can be approved by numerous users. Issues could arise if people verify inaccurate information, as this would mislead other users about the data. It could also hinder users from making changes to inaccurate information if it has received multiple verifications; users could second guess their correct information.