Data analysis on London OSM

Map chosen

> London is the capital and most populous city of England and the United Kingdom.--wikipedia

1. Encountered problems

There are several problems popped up through the data analysis process, including bad formatted street names and phone number, and KeyError caused by elements with missing attributes.

1.1 Bad street names

The bad street names are usually caused due to bad street types, such as using abbreviations, or typo. According to the codes provided in case study lesson, the cleaning process is:

First, using regular expression to locate tags that contain street names.

Second, checking against the street names with an expected list of street types such as Street, Road to find names to be fixed.

Third, based on the returned list, updating the expected list (i.e. add unexpected street types such as Grove or Mews into the list); and make a mapping dictionary to correct the bad street types into the good ones (i.e. turning abbreviations like st. into Street, and typo like Sreet into Street.)

Fourth, creating an update function to update street names using the mapping dictionary.

Fifth, updating the street names when shaping elements so that the fixed data will be written into the CSV files.

Raised issue

This process was made easier thanks to the codes provided. But there are still some issues, such as some street name fields contain house number （11-13 charlotte Street）or even was not actually a street but part of a postcode（5A）.

Solution

I think this can be fixed by **changing the key name into FIXME during the shaping in the future**.

- Benefits
    - the problem street names will be filtered out. This will make the data cleaner, particularly when query for address in the database using aggregation.
    - it will be easier to figure out how many problem issues are there to be fixed in the database, so the analysts may decide if the manually fix these issues.
- Issues
    - if simply change the problem street names' key in database from "street" to "FIXME", it might result in confusion for the next user to figure out what type of "FIXME" it requires. The user would unable to tell if the to be fixed row is a street name/ a phone number or something else, given by a simple "5A" in the value. It might take more effort to double check, particularly if the "type" field is not "addr".

1.2 Bad phone number

The phone number are of various formats which made the data very dirty.
To solve the issue, I researched UK phone number system via wikipedia, and found that the most commonly seen 44 is the country code for the UK, and 02 is the geographic code for London. Other cities or regions in the UK have their own geographic codes, but also start with 0.

Bye-bye 44, 0 and ( )-+

So I decided to get rid of the country code like (44, or 4) and zero in the beginning of the phone number (00xxxxxx, or 0xxxxxx).I also removed all non-numeric digit from the value, like ()+- using regular expression.

Raised Issue

One issue raised is that some users input more than one phone numbers in a single field, and separating them with semi-colon “;” . (44 1234567; 44 7654321)
The regular expression to get rid of signals `re.sub("\D", "", v)` would mingle the independent phone numbers together in the same field. (44 1234567; 44 7654321–->1234567447654321)

But these are individual situations. To fix this,

Solution #1:

separating the phone numbers in the osm file by adding new node_tag "phone_1" with the second phone number

- Benefits:
    - the bad phone number will be separated without extra codes written.
- Issues:
    - will be time consuming if there are many such problem phone numbers.
    - if the file is very large, ordinary computer might take rather long time to search the target node_tag and change them, which can be annoying.

Solution #2:

inserting new rows into SQL database directly with the key as “phone_1” and value as the second phone number;

- Benefits:
    - would be easier and quicker to find the problem phone number using DB Browser than search in the OSM file directly.
- Issues:
    - still time consuming if there are multiple such situation.

Solution #3:

inserting new nodes_tags in shaping and csv programatically.

- Benefits:
    - the problem phone numbers will be handled problematically, which not saves manual work.
    - it can be apply in other osm database.
- Issues:
    - The codes required will more complicated than previous cleaning code as it requires inserting extra rows through CSV writing process.
    - The current codes iterparse the tags through .iterparse(), which means they can easily change values in existing field.

1.3 Bad key names

After writing the CSVs into database, I found that some keys referring to the same thing are under different names, such as “postalcode” and “postcode”, “fixme” and “FIXME”. This is partly attribute to different habits of the contributors. Such unified key names would cause extra work in SQL queries later.
So I updated these key names through shaping process as well.

1.4 Missing attribute

Some elements do not have attributes such as user or uid, which made it unable to apply {f : el.attrib[f] for f in node_attr_fields} when creating node_tag dictionary through shaping. The KeyError would raise.
I decided not to ignore these elements in case they also contain important tag information(as the informative nodes_tags are linked to nodes tables in the database).
I therefore chose to put “Missing Attributes” into the dictionary if there is missing attribute of the corresponding field.

2. Overview of the data:

2.1 What’s the OSM data’s size?

Project map: 144.293905 MB
Sample map: 5.507262 MB

2.2 How many nodes and tags are there in the OSM?

A total of 531302 node, 98374 way, 3879 relation, 724002 tag and 720629 nd in the OSM.

2.3 How many users contributed to the OSM?

A total of 2342 unique users contributed to the OSM.

Who is the most contributive user?

User “Paul The Archivist” is the most contributive user in both nodes and ways in the map.

Top three most contributive user in nodes:

'Paul The Archivist' contributed to 72336 nodes, followed by 'Tom Chance'(48877) and 'Ed Avis', 33216).

Top three most contributive user in ways:

'Paul The Archivist' also marked 15678 ways in the map, followed by 'Tom Chance' (5572), and 'Derick Rethans' (5401).

2.4 How many theatres are marked in the map?

A total of 85 theatres were marked in the map, and 18 of them sit in the West End area (sample map).

(I found through the sample map that ways_tags table documented more theatres than nodes_tags table, which was out of my surprise. So I queried for theatres from both the two tables using SQL and used `.set()` to avoid duplication.
The SQL queries in the two tables have some slight difference though.
From node_tags, it's better to search via key-name to get the theatre name, while from the way_tags tables, it is better to search for the wikipedia attribute.)

2.5 How many bicycle parking spots are there in the map and how about their total capacity?

1774 bicycle parking spots are documented in the map's nodes.
A total of 1444 of the bicycle parking spots can provide 14843.0 parking spaces. The rest 330 spots did not provide capacity information in the OSM.

Additional Improvement #1

China's two bike sharing giants have made their move overseas, and London is one of the target cities. And many Chinese cities have encountered issues such as bikes occupying sidewalks and affecting other pedestrians.
I checked up the bicycle parking in the OSM map as I'm interested in if the public facilities in London is enough to handle a potential wave of shared bikes.
So, in addition to the parking facility, I'd also like to know in future analysis:

Query #1:

How is the existing bike-rental service network in London?

```
Query1 = '''select count(*) as num from nodes_tags where key = "amenity" and value
= “bicycle_rental")'''
```

- Benefits:
  - will be able to get the figure
- Issues:
  - users could have missing bike rental spots in the osm. As bike rental service are usually big companies boasting about their coverage, the operator might release their exact number of rental spots, or at least we can try to check my result against the released number.

Query #2:

How bicycle friendly London is?
How many cycleways are there in London? How many of them are proposed and how many are under construction.

```
proposed_way = '''select count(*) as num from ways_tags where key = "proposed" and
value = "cycleway")'''
c.execute(proposed_way)
proposed_ways = c.fetchall()
print "proposed cycleways", proposed_ways
```

- Benefits:
  - will be able to get the idea of the bicycle environment in general in London.
  - particularly with the proposed cycleway information, we can expect the city's future attitude to green transportation in the future.
- Issues:
  - I tried codes as above, but syntax error was raised: sqlite3.OperationalError: near ")": syntax error. **Will research more on that.**

Additional Improvement #2

More visualization codes can be added to better illustrate the analysis result.
- Benefits:
- will be more effective in illustrating the result and communication

- Issues:
  - time costs