OpenStreetMap

Map Area

Melbourne, Victoria, Australia

https://www.openstreetmap.org/node/21579127#map=11/-37.8139/144.9632

This area is Melbourne city which I'm interested to complete my Master degree in its university , so this project is a good opportunity to discover this area.

Problems in the Map

After I download Melbourne map I take a small sample using Sample.py, and I run audit.py I notice some issue in the map:

Inconsistent street names:

some street has misspelling of street name "stree" or abbreviation "St.", another issue is the order of street name the direction is come in the last name so we will rearrange them:

- Change Graham Stree to Graham Street.
- Change Kirkham Road to West Kirkham Road.

```
street_type_re = re.compile(r'\b\S+\.?$', re.IGNORECASE)
street_dirc_re = re.compile(r'(\b\S+) (\b\S+) (\b\S+)', re.IGNORECASE)


#The Expected street type
expected = ["Street", "Avenue", "Boulevard", "Drive", "Court", "Place", "Square", "Lane", "Road","Parade",


        "Trail", "Parkway", "Commons","Circle","Circuit","Highway","Grove","Close","Terrace","Crescent","Way","Gardens"]


dir_expected = ["North","West","East","South"]


#street mapping
mapping = { "St": "Street",
            "St.": "Street",
             "Stree":"Street",
             "Ave": "Avenue",
            "Rd.": "Road"
            }
''''
```

```
,,,,,,,
def audit_street_type(street_types, street_name):
        m = street_type_re.search(street_name)
        if m:
            street_type = m.group()
            #check if the street in the expected list or not
            if street_type not in expected:
                if street_type in dir_expected
            #if it in diraction list ex: Kirkham Road West
                    update_name(street_name)
                else:
                    street_types[street_type].add(street_name)




def update_name(name):
    street = street_dirc_re.search(name)
    name = street_dirc_re.sub(r'\3 \1 \2', name)#audit street name by reorder the nams
    return name
```

Inconsistent state names:

The state is written in different ways abbreviation VIC or the city Melbourne instead of state ,so I will change it to one format:

- Change VIC to Victoria .
- Change Melbourne to Victoria.

```
state_exp=["Victoria"]
stat_mapping={
     "VIC": "Victoria",
     "VICTORIA": "Victoria",
     "Melbourne":  "Victoria"


}
,,,,
def updat_state_name(name,stat_mapping):
    if name in stat_mapping.keys():
        print name
        name = stat_mapping[name]
        return name
```

```
##cleaning stages
```

```
elif LOWER_COLON.match(child.attrib["k"]):  # code for 'id', 'key' and 'type' keys ...
                node_tag['type'] = child.attrib['k'].split(':',1)[0]
                '''
                  '''
                ''''
                if child.attrib["k"] == 'addr:street':
                        node_tag["value"] = update_name(child.attrib["v"])
                elif child.attrib["k"] == 'addr:state':
                        # check to see if the cleaning function returns a value:
                        if  updat_state_name(child.attrib["v"],stat_mapping):
                            # if it does, add it to the dictionary
                  node_tag["value"] = updat_state_name(child.attrib["v"],stat_mapping)
                        else:
           # if it doesn't, the attribute is uncleanable so move to the next element
                            continue
```

Data Overview and Additional Ideas:

File sizes

melbourne_australia.osm.....802 MB
Melbourne_DB.db.....417 MB
nodes.csv 303MB
nodes_tags.csv.....26.2MB
ways.csv.....29.6MB
ways_nodes.csv.....104 MB
ways_tags.csv.....46.1MB

number of unique users

```
sqlite> SELECT COUNT(DISTINCT(e.uid))
   ...> FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
2388
```

Number of nodes

```
sqlite> select count(*) from nodes;
3764775
```

Number of ways

```
sqlite> select count(*) from ways;
518803
```

Top 10 contributing users

```
,,,,,,,
sqlite> 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;

CloCkWeRX,         1185333
Leon K,            432274
melb_guy,          292440
Glen,              172088
AlexOnTheBus,      136104
stevage,           95097
Canley,            90444
dssis1,            79850
nickbarker,        74688
Rhubarb,           71896
```

Most common amenities

```
,,,,,,,
sqlite> SELECT value, COUNT(*) as num
   ...> FROM nodes_tags
   ...> WHERE key='amenity'
   ...> GROUP BY value
   ...> ORDER BY num DESC
   ...> LIMIT 5;

restaurant|1614
bench|1392
cafe|1341
fast_food|1240
toilets|1095
```

Top 10 appearing school

```
,,,,,,,
qlite> SELECT value
   ...>         FROM nodes_tags
   ...>             JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='school') i
   ...>             ON nodes_tags.id=i.id
   ...>         WHERE nodes_tags.key='name'
   ...>         LIMIT 10;

Pascoe Vale South Primary School
Coburg West Primary School
St Fidelis Primary School
East Doncaster Secondary College
Syndal Pre-School
Tally Ho Preschool
Christ Our Holy Redeemer Primary School
St Johns Pre-School
Nara Pre-School
Sydenham-Hillside Primary School
```

Top 10 popular bank

```
,,,,,,,
sqlite> SELECT nodes_tags.value, COUNT(*) as num
   ...>         FROM nodes_tags
   ...>             JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='bank') i
   ...>             ON nodes_tags.id=i.id
   ...>         WHERE nodes_tags.key='name'
   ...>         GROUP BY nodes_tags.value
   ...>         order by num desc
   ...>         LIMIT 10;



Commonwealth Bank|98
ANZ|57
Westpac|56
NAB|53
Bendigo Bank|43
Bank of Melbourne|20
Bank of Queensland|5
HSBC|4
ANZ Bank|3
```

```
Bankwest|3
```

Top 5 popular fast food

```
,,,,,,,
sqlite> SELECT nodes_tags.value, COUNT(*) as num
   ...>     FROM nodes_tags
   ...>     JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='fast_food') i
   ...>     ON nodes_tags.id=i.id
   ...>     WHERE nodes_tags.key='cuisine'
   ...>     GROUP BY nodes_tags.value
   ...>     order by num desc
   ...>     LIMIT 5;



burger|196
chicken|159
pizza|154
fish_and_chips|131
sandwich|104
```

Additional Ideas:

After we analyzing the data we can see a lot of misspelling and inconsistent values of the same key, to avoid that we can build an automated parser to audit and clean the user input. Another thing we could build a script to extract the information for a popular places like university institute , library ,mall and organizations and load it in the map this is so important to user especially people who comes from outside the country their priority is to have a full information of the location they want to go to it like business hours ,contact information and so on. The benefit of this improvement is to have the complete information and the data will be more consistence because it comes from its original source ,However not all organizations will allow to share its data or give permission to accessing it's data . Another issue is the different naming between the original source and the nod in OpenStreetMap. However we could overcomes these issues by explain the values of improvement for the organizations . One of these values is rising number of visitor to the place which has a clear and complete information in addition we will set rules and procedure to the extraction systems to protect the privacy and security of data, Although we will reviewing and handling the loading data to be sure its consistency and there is no duplicated data.

The open street map depends on users contribute so we should motivate them by doing different competition to clean and Auditing the map .

Conclusion:

I download the Melbourne dataset which is large city and has a lot of data then I analyze it and clean it but we can't say it has cleaned 100% ,However I believe it was sufficiently cleaned for the purposes of this project which was focus in auditing the data. Through this project I see the power of data it display a details of country that I did not go to but I have now a clear pictures of live in Melbourne.

My future works is to clean the other variable like phone and house number to have a consistent values .

Sources:

https://en.wikipedia.org/wiki/Category:Streets_in_Melbourne

http://melbourne-vic.street-map.net.au/

http://udacit.com

https://github.com/pratyush19/Udacity-Data-Analyst-Nanodegree/blob/master/P3-OpenStreetMap-Wrangling-with-SQL/Quiz/audit.py