Dublin OpenStreetMap Data Wrangling Project

Map Area: Dublin, Leinster, Ireland
https://www.openstreetmap.org/relation/1109531#map=12/53.3547/-6.2510

I chose this city because I live here now and I would like to learn it better. Also, English is not my native language, so it is a great possibility to learn more about the street naming here.

Auditing and correcting the dataset

First, I used *'count tags.py'* to count the tags in Dublin dataset.
Output:

```
{'bounds': 1,
'member': 87272,
'nd': 1917485,
'node': 1392771,
'osm': 1,
'relation': 4887,
'tag': 1022937,
'way': 254675}
```

There are ~ 1.4M of nodes and ~250,000 of ways in the dataset.

Using *'count address types.py'* I counted the types of addresses.
Output:

```
street : 104968
housenumber : 88452
city : 18223
country : 4980
housename : 3033
postcode : 2534
interpolation : 979
block : 61
unit : 39
street:ga : 30
terracename : 30
city:ga : 28
county : 18
postal_district : 17
place : 15
apartments : 14
housename:ga : 9
inclusion : 6
flats : 5
suburb : 2
town : 1
floor : 1
terrace : 1
```

```
housenumber:source : 1
loc8 : 1
```

The results show, that mostly the street name and house number are used in the addresses. We can also notice that there are 3033 house names, which are used instead of house numbers - you can see it often in Ireland, indeed. Postcodes are only 2534, so they are used for only 2% of addresses, and there is an explanation for this too. Ireland did not have postcodes at all and they were created recently. It is not required to use postcodes for letters and parcels, so many people do not know their postcode.

Using *'street types.py'* I counted all the unique street types and print the examples. Regular expressions were used to find the last word which is usually used as street type.
Example of output:

```
road : 22627  [ Station Road, Station Road, Rock Road ]
park : 11557  [ Seabury Park, Seabury Park, Seabury Park ]
avenue : 10664  [ Seapoint Avenue, Sydney Parade Avenue, Ballinteer Avenue ]
drive : 6440 [ Blackthorn Drive, Blackthorn Drive, Wynnsward Drive ]
street : 4744  [ Main Street, Store Street, Main Street ]
grove : 3589  [ Corbawn Grove, Corbawn Grove, Corbawn Grove ]
...
upper : 1255  [ Georges Street Upper, Sheriff Street Upper, O'Connell Street Upper ]
...
avevnue : 1  [ Kill Avevnue ]
```

There are a lot of street types in Dublin, I will use special sources for validating them.
After the audit and validation I created the list of valid street types:

```
Road, Park, Avenue, Drive, Street, Grove, Crescent, Court, Lawn, Green, Close, Terrace,
Rise, Place, Way, Gardens, Heights, View, Walk, Wood, Lane, Lawns, Estate, Square, Vale,
Hill, Woods, Manor, Row, Quay, Parade, Glen, Mews,Meadows, Hills, Boulevard, Brae, Mount,
Valley, Brook, Well, Plaza, Alley, Crossing, Rest, Field, Bridge, Dales, Bypass,Cove,
Haven, Cross, Yard, End, Corner, Point
```

Correcting mistypes and data harmonization must be done for this list:

```
Ave, St., Rd., Roafd, St, Rd, Avevnue, Nouth
```

And there are also some strange things for investigation for these street types:

```
lower : 1742  [ Kilmacud Road Lower, O'Connell Street Lower, Kilmacud Road Lower ,...]
upper : 1255  [ Georges Street Upper, Sheriff Street Upper, O'Connell Street Upper ,...]
north : 506  [ Merrion Square North, Terenure Road North, Terenure Road North ,...]
west : 504  [ Essex Street West, Mountjoy Square West, Mountjoy Square West ,...]
east : 434  [ Lombard Street East, Essex Street East, Essex Street East ,...]
nouth : 54  [ Bayside Boulevard Nouth, Bayside Boulevard Nouth, Bayside Boulevard Nouth
,...]
middle : 22  [ Gardiner Street Middle, Gardiner Street Middle, Gardiner Street Middle
,...]
great : 12  [ Ship Street Great, Ship Street Great, Ship Street Great ,...]
little : 7  [ Mary Street Little, Mary Street Little, Mary Street Little ,...]
w : 6  [ O'Brien's Place W, O'Brien's Place W, O'Brien's Place W ,...]
. : 5  [ ., ., . ,...]
11 : 2  [ Unit 6, North Park Business Park, Finglas, Dublin 11, James Business Park, St
Margaret's Road, Finglas North, Dublin 11 ]
ride : 2  [ Leixlip Louisa Bridge Park & Ride, Leixlip Louisa Bridge Park & Ride ]
```

```
2 : 1  [ Dame Court, Dublin 2 ]
ireland : 1  [ CHQ epic Ireland ]
dublin : 1  [ Applewood Main St, Applewood, Swords, Co. Dublin ]
27-31 : 1  [ Supple Park 27-31 ]
airport : 1  [ Dublin Airport ]
```

We can see that we have some certain mistakes in the addresses in the street field, like "." or "Applewood Main St, Applewood, Swords, Co. Dublin". These things could be cleaned manually. But I would like to investigate more frequent cases: additional street types: "Lower", "Upper". "West" etc. These street types are correct, but they must have another street type before (e.g."Street Upper"), so I will check these street names to insure that common street types are correct in these cases.

Using *'additional street types.py'* I audit all the street names with additional street types.
Example of output:

```
East
{'Parnell Square East', 'Merrion Square East', "James's Place East", 'Tivoli Terrace
East', 'Bow Lane East',
'Bayside Square East'....
```

All the street names are correct here and they don't need to be improved.

Interesting fact!
There in only one MIDDLE street in our dataset, but there are almost 3000 Upper and Lower streets. This is Gardiner Street Middle, it also has Upper and Lower sisters, and it is a very tiny little part of the street close to Mountjoy Square in the Dublin City Center.

For correcting the dataset in *'improve.py'* mapping was used:

```
mapping = {'Ave': 'Avenue',
           'St.': 'Street',
           'Rd.': 'Road',
           'Roafd': 'Road',
           'St': 'Street',
           'Rd': 'Road',
           'Avevnue': 'Avenue',
           'Nouth': 'North'}
```

After this dataset was converted from XML to CSV format using *'parse to csv.py'*.
Two csv files containing street names were cleaned in *'clean street names.py'.*
Then the cleaned csv files were imported into a SQL database using Udacity schema and *'create SQL database.py'*

Data Overview

File sizes

```
dublin_ireland.osm .... 320 MB
dublin.db ............. 181 MB
nodes.csv ............. 109 MB
nodes_tags.csv ........ 5.6 MB
ways.csv ............... 14 MB
ways_tags.csv .......... 28 MB
ways_nodes.cv .......... 45 MB
```

Number of nodes

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

```
1392771
```

Number of ways

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

```
254675
```

Number of unique users

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

```
1511
```

TOP 3 contributed users

```
sqlite> SELECT u.user, count(u.id) as contributed FROM
(SELECT user, id FROM nodes UNION ALL SELECT user, id FROM ways) as u
GROUP BY u.user ORDER BY contributed DESC LIMIT 3;
```

```
Nick Burrett     218783
Brianh           197613
mackerski        197192
```

How many users contributed only once?

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

```
375
```

How many pubs are in Dublin?

```
sqlite> SELECT count(*) FROM nodes_tags WHERE key = 'amenity' and value
= 'pub';
```

```
456
```

There must be 740 pubs in Dublin, so there are some pubs that are not on the maps yet.

Additional data exploration

What are the ‘natural’ elements in the dataset?

```
sqlite> SELECT value, count(*) FROM nodes_tags WHERE key = 'natural'
GROUP BY value ORDER BY count(*) DESC;
```

```
tree   8060
peak     50
wood     12
bay       6
spring    4
cliff     1
rock      1
stone     1
tree_row  1
```

Who contributed natural stone to the map?

```
sqlite> SELECT nodes.user
FROM nodes JOIN nodes_tags ON nodes.id = nodes_tags.id
WHERE nodes_tags.key = 'natural' AND nodes_tags.value = 'stone';
```

```
IrlJidel
```

How many contributions were made by the user who contributed natural stone to the map?

```
sqlite> SELECT s.user, count(s.id) FROM
(SELECT id, user FROM nodes UNION ALL SELECT id, user FROM ways) as s,
(SELECT * FROM nodes JOIN nodes_tags ON nodes.id = nodes_tags.id
WHERE nodes_tags.key = 'natural' AND nodes_tags.value = 'stone') as u
WHERE s.user = u.user;
```

```
IrlJidel   15472
```

This user made 0.94% of all the contributions.

Conclusions

After the review of the data for Dublin city I would like to mention several conclusions and suggestions.

1. Additional data cleaning is needed for the address information.
Conclusion: Although there are almost no mistyping in the dataset, some data cleaning is necessary. There are many cases where all the lines of address or some parts of address (street and house number) were input in the ‘street’ field.
Benefits: We will have cleaner and richer dataset. We will have more values for house numbers and city.
Anticipated issue: Big part of this data cleaning must be done manually, because addresses , except of street name, can include one or several data of: house number or several numbers, second (bigger) street, city or town, area code, county, country. This will have a higher cost.

2. Creating required fields.
Suggestion: Users must be encouraged to input at least suburb name (e.g. ‘Sandymount’), if possible also area code (e.g. ‘Dublin 4’), but postcodes can be skipped as they are complicated and not used widely. This information can be useful for comparing suburbs.
Benefit: Richer database, which will allow users to make better searches and receive better addresses. Data analysis will have more options.
Anticipated issue: Required fields can create negative user’s experience. This may stop some users from contributing. Also wrong information may be input into these fields, so it will be good to improve code for collecting these fields from already existing elements nearby.

3. Improve the contribution
Suggestion: 25% of all users contributed only once. It is a big part of users. It would be nice to investigate user’s experience during contribution and find the way to keep users interested in contributing again. As we could see from number of pubs, many of amenities are not contributed yet. Users need to be encouraged!
Benefit: We will have better and up-to-date maps.
Anticipated issue: Cost.