Udacity DAND Project 3 – Wrangle OpenStreetMap Data

Introduction

In this project, I'll use XML data downloaded from OpenStreetMap to perform some data analysis on Chicago.

The reason I choose Chicago is that I'm a huge fan of Milton Friedman who was once a professor in the university of Chicago, and I'm planning to pay a visit to Chicago this year.

In this project, I'll clean the data first and then store the data into Sqlite. After that, I'll perform some data analysis using the data stored in Sqlite.

My goal is to try to figure out what I can eat , play and entertain there.

map link (https://s3.amazonaws.com/metro-extracts.mapzen.com/chicago_illinois.osm.bz2)

Problems Encountered In The Map

The Chicago_illinois.osm file is roughly 2GB

As suggested during the case study, First I use the code in sample_file.ipynb to get a sample file of Chicago,which is 1/100 the size of the original file.

and here is a brief summary of fisrt level tag of my sample file

```
{'member': 732,
 'nd': 101177,
 'node': 86948,
 'osm': 1,
 'relation': 48,
 'tag': 67374,
 'way': 12258}
```

Then, I follow the procedure in case study to check if any problematic data exits in keys

```
{'lower': 20427, 'lower_colon': 30978, 'other': 15969, 'problemchars':
0}
```

To get a better understanding of the keys, i also print out the keys, and took a glimps of these keys.

There are tree values I think I'll need to update

- street name

For street names,

after using audit_file.py for our sample file, I find quite a lot of street names are written in abbreviation for example

```
'Ave': {'1st Ave',
        'Alabama Ave',
        'Arkansas Ave',
        'California Ave',
        'Cicero Ave',
        'Deleware Ave',
        'Hawaii Ave',
        'Lake Ave',
        'Meridian Ave',
        'N Cumberland Ave',
        'New York Ave',
        'S Western Ave',
        'W 133rd Ave',
        'W North Ave',
        'West Maple Ave'},
```

The Ave should be changed to Avenue. so **'*Alabama Ave*'** would become **'*Alabama Avenue*'**

And I updated the following abrreviation such as str. ave. to street and avenue with the following mapping.

```
street_type_mapping = { "St": "Street",
                        "St.": "Street",
                        "Ave": "Avenue",
                        "Ave." : "Avenue",
                        "Rd" : "Road",
                        "Rd." : "Road",
                        "Cir" : "Circle",
                        'Ln' : "Line",
                        "Dr" : "Drive",
                        "Ct" : "Court",
                        "Trl" : "Trail"
                        }
```

- post code

For post code, which come as a little bit suprise to me, Chicago post code system is simple and nite , it's usually in the form of a five digit number,and sometimes with a 4 digit exitension

```
{'60618'
 '60018-2627',
 '60160-1607',
 '60183 / 60174',
 '60521-2101',
 '60615-5299'}
```

,I find there is no need to change this type of data, since these data are arealy cleaned, so I'll just leave it the way it is.

- phone number

For the phone number , the format isn't consistent from one to another. For example,

```
{'(630) 444-1700',
 '(847) 825-4231',
 '+1 630 7392999',
 '+1 630 7590333',
 '+1 630 7839240',
 '+1-312-7874030',
 '+1-847-696-2800',
 '1 630 682 2105',
 '773 486 9010'}
```

Some phone number has "(",")","+"or "-" between numbers while the others don't, So I'll convert them into 10 digits or 11 digits number, depending on whether there is "1" in the front of the number, and get rid of all the special characters by using the following mapping

```
phone_number_mapping = {"(":"",
                        ")":"",
                        "-":"",
                        "+":"",
                        " ":""
                        }
```

here is code snippet for the above changes.

```
def update_phone_number(value):
    value = re.sub(r'\D', "", value)
    return value
```

So the result for the phone number '***(847) 825-4231***' would become '***8478254231***'

File Size

```
'chicago_illinois.osm': '1978.33MB',
'chicago_illinois_sample.osm': '19.98MB',
'mydb.db': '1160.1MB',
'nodes.csv': '729.74MB',
'nodes_tags.csv': '11.95MB',
'ways.csv': '75.95MB',
'ways_nodes.csv': '234.79MB',
'ways_tags.csv': '212.82MB'
```

Import Data To Database

After the above cleaning process, I then decide to move on to update the data in the original file and import it into Database

I process the data through audit_file.py and updates.py,and write_data.py

audit_file.py identifies if any problematic character or unformated data occurs, and updates.py will update it.

write_data.py are mainly

Perform Data Analysis

Code snippets for performing data analysis

In [16]:

```
cursor.execute("ANALYZE;")
cursor.execute("SELECT tbl, stat FROM sqlite_stat1;")
rows = cursor.fetchall()
db_stats = pd.DataFrame(rows[::-1], columns=['Table','Stat'])
db_stats
```

Out[16]:

	Table	Stat
0	nodes_tags	342727
1	nodes	8694751
2	ways_nodes	10267457
3	ways_tags	6375985
4	ways	1225807

In [17]:

```
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
table_names =cursor.fetchall()
table_names=[i[0] for i in table_names]
col_names={}
for i in table_names:
    cursor.execute('PRAGMA TABLE_INFO({})'.format(i))
    col_names[i]=[tup[1] for tup in cursor.fetchall()]
pprint.pprint(col_names)
```

```
{'nodes': ['id',
           'lat',
           'lon',
           'user',
           'uid',
           'version',
           'changeset',
           'timestamp'],
 'nodes_tags': ['id', 'key', 'value', 'category'],
 'sqlite_stat1': ['tbl', 'idx', 'stat'],
 'ways': ['id', 'user', 'uid', 'version', 'changeset', 'timestamp'],
 'ways_nodes': ['id', 'node_id', 'position'],
 'ways_tags': ['id', 'key', 'value', 'category']}
```

Top 10 Contributors

In [25]:

```
total_nodes_ways = int(db_stats[db_stats['Table']=='nodes']['Stat']) + \
                   int(db_stats[db_stats['Table']=='ways']['Stat'])
cursor.execute("SELECT uid, user, sum(count) as count FROM \
               (SELECT uid, user, count(*) as count FROM nodes GROUP BY uid \
               UNION \
               SELECT uid, user, count(*) as count FROM ways GROUP BY uid) \
               GROUP BY uid \
               ORDER BY count desc LIMIT 10;")
rows = cursor.fetchall()
users = pd.DataFrame(rows, columns=['User ID', 'User', 'Count'])
percentages = (users['Count'] / total_nodes_ways * 100).round(decimals=2)
users = pd.concat([users, percentages], axis=1)
users.columns = ['User ID', 'User', 'Contributions', 'Percentages of Total']
users
```

Out[25]:

	User ID	User	Contributions	Percentages of Total
0	674454	chicago-buildings	5606367	56.51
1	567034	Umbugbene	1093941	11.03
2	130794	alexrudd (NHD)	226078	2.28
3	147510	woodpeck_fixbot	220460	2.22
4	169600	patester24	105172	1.06
5	238419	g246020	104412	1.05
6	187130	mpinnau	103701	1.05
7	5387019	Oak_Park_IL	102949	1.04
8	522978	asdf1234	101109	1.02
9	120146	TIGERcnl	97078	0.98

Top ten cuisines to Eat in Chicago

In [28]:

```
cursor.execute("SELECT value, sum(count) as count FROM \
               (SELECT value, count(*) as count FROM nodes_tags WHERE key = 'cui
sine' \
               GROUP BY value \
               UNION \
               SELECT value, count(*) as count FROM ways_tags WHERE key = 'cuisi
ne' \
               GROUP BY value) \
               GROUP BY value ORDER BY count desc LIMIT 10;")
rows = cursor.fetchall()
cuisines = pd.DataFrame(rows, columns=['Cuisine', 'Count'])
cuisines
```

Out[28]:

	Cuisine	Count
0	burger	548
1	mexican	245
2	pizza	239
3	sandwich	205
4	american	159
5	coffee_shop	153
6	chinese	89
7	italian	89
8	chicken	86
9	ice_cream	48

Leisures can be found in Chicago

In [21]:

```
cursor.execute("SELECT value, sum(count) as count FROM \
               (SELECT value, count(*) as count FROM nodes_tags WHERE key = 'lei
sure' \
               GROUP BY value \
               UNION \
               SELECT value, count(*) as count FROM ways_tags WHERE key = 'leisu
re' \
               GROUP BY value) \
               GROUP BY value ORDER BY count desc LIMIT 10;")
rows = cursor.fetchall()
Leisure = pd.DataFrame(rows, columns=['Leisure', 'Count'])
Leisure
```

Out[21]:

	Leisure	Count
0	pitch	5926
1	park	4045
2	playground	2960
3	recreation_ground	1287
4	swimming_pool	506
5	golf_course	328
6	sports_centre	277
7	garden	215
8	picnic_table	149
9	nature_reserve	135

Shop Categories In Chicago

In [23]:

```
cursor.execute("SELECT value, sum(count) as count FROM \
               (SELECT value, count(*) as count FROM nodes_tags WHERE key = 'sho
p' \
               GROUP BY value \
               UNION \
               SELECT value, count(*) as count FROM ways_tags WHERE key = 'shop'
 \
               GROUP BY value) \
               GROUP BY value ORDER BY count desc LIMIT 10;")
rows = cursor.fetchall()
Shop = pd.DataFrame(rows, columns=['shop', 'Count'])
Shop
```

Out[23]:

	shop	Count
0	supermarket	578
1	convenience	347
2	clothes	252
3	car_repair	187
4	car	144
5	department_store	136
6	hairdresser	124
7	alcohol	108
8	bakery	100
9	bicycle	96

An Idea On User contribution

As we can observe over the data, currently , more than half of the data are contributed by a single person. This situation brought benefits as well as trouble to the website.

An obvious benefit of current situation is that, data contributed by a few contributors will make the overall data structure more formality, thus can be easily cleaned. But it also make project hard to scale since there are only a few people to contribute.

In order to scale up the project as well as maintaining the formality of the data structure. I come up with the following solutions.

Firstly, some data should be specified to certain type only. For example, data like phone number and post code data should be required to input digits only. One obvious advantage of this method is that, no matter how many people joined the project, the formality will remain relatively clean and tidy. Yet, there are also short comes with this method, for instance, people who follow this project would have to read much more guidelines before getting their hands wet. This will surely discourage people from joining the project.

Another method i'm considering is to bring some incentives for people to participate this project. For normal people, it’s hard to see the rewards coming from this open data source project, so a lot people won't have the incentive to join this project. why not take a interview with these top contributors? After the interview, editors could pick up share some stories about why these top contributors are willing to contribute to this project. The website may also tell people some benefits they can get from this projects on their home page. By this method, it will definitely help drive the contributor number and motivate them to join the project. As we can see, this method requires huge amount of work, the website may not have enough resource to interview these contributors and to select stories.

Lastly, I recommend adding easy understanding animated video tutorials to teach people how they can contribute to the project . This will reduce the barrier of entry , thus bring more contributors to the project. But this also requires time and money similar as previous suggestion.

Summary

After this project, I find it quite difficult to get detailed info I needed, but in general, it gives me an overall impression of Chicago and at least I find that there are a lot of golf_course and swimming pools there, so i might google further to find a good swimming pool.

Not sure how the user "chicago-buildings" can have such a big contribution to this project, it did amazed me,I feel hard to believe someone can contribute such a high percentage to project like this. Really want to know how he/she achieved that.

I'm a big fan of cafe,So I'll dig deeper to find delicious cafe shops in Chicago since there are lots of cofe shop there.

In [ ]: