Michael C. Villadelrey

OpenStreetMap Data Case Study

Map Area

Honolulu, Hawaii, United States

- https://mapzen.com/data/metro-extracts/metro/honolulu_hawaii/
- https://www.openstreetmap.org/relation/119231

Being largest City of Hawaii, Honolulu is the place where you can get the most things offered in Hawaii. Historical landmarks, world class shopping, awesome beaches, parks, falls, great local delicacies, almost everything Hawaii can offer is in Honolulu. This place is one of the destinations I want to have a vacation with my family. So it is a good opportunity to extract data from mapzen and have a good idea on the places on Honolulu.

Problems Encountered in the Map

After checking the csv files from the output of the provided data.py file, I have noticed some problems in the data:

1. Different street abbreaviations (St. , St , Rd, Road, Dr, Drive, etc)
2. Inconsistent postal code (“96850” , “96734-9998” , “HI 96819”)

Problems with Street names:

Both the nodes_tag and the ways_tag have key = ‘street’. And upon checking the values, I saw inconsistencies in the naming convention of the street names. To deal with this problem, I checked the different conventions in the csv and created list of accepted street names and a dictionary mapping to correct those with unacceptable street names. I have created a function where the street names will be corrected before it is added in the dictionary.

```
# This is the list of acceptable street names
expected = ["106", "Avenue", "Boulevard", "Center", "Circle", "Drive", "Highway", "Ike",
            "Lane", "Loop", "Mall", "Parkway", "Place", "Road", "Street", "Walk", "Way",
            "Honolulu", "Kailua", "King", "Momi", "Terrace"]
```

```
# the keys of this mapping was obtained from checking unique keys
# in the csv files resulted from the provided data.py
mapping = { "St": "Street",
            "St.": "Street",
            "Rd.": "Road",
            "Ave": "Avenue",
            "Blvd": "Boulevard",
            "Dr": "Drive",
            "Hwy": "Highway",
            "Pkwy": "Parkway"
            }
```

```
def update_name(name, mapping):
# this function will set the street name to an acceptable street name
    badname = street_type_re.search(name).group()
    pos = name.find(badname)

    if badname in mapping:
        goodname = mapping[badname]
        name = name[:pos]+goodname
    return name
```

Sample Corrected Street Names under Ways Tag		
id	**value (bad street name)**	**value (corrected street name)**
268795384	Lusitania St.	Lusitania Street
280348190	Ala Pumalu St	Ala Pumalu Street
62541758	Kalakaua Ave	Kalakaua Avenue

Sample Corrected Street Names under Nodes Tag		
id	**value (bad street name)**	**value (corrected street name)**
367803601	Kipapa Dr	Kipapa Drive
4198265289	Ala Moana Blvd	Ala Moana Boulevard
4255639689	Kamehameha Hwy	Kamehameha Highway
4223222595	Meheula Pkwy	Meheula Parkway

Problems with the Postal:

From Wikipedia: "The basic format consists of five numerical digits. An extended **ZIP+4** code, introduced in 1983, includes the five digits of the ZIP Code, a hyphen, and four additional digits that determine a more specific location within a given ZIP Code."

Since most of the values for postal codes in the dataset is five numerical digits, I will adopt this as standard for postal code values.

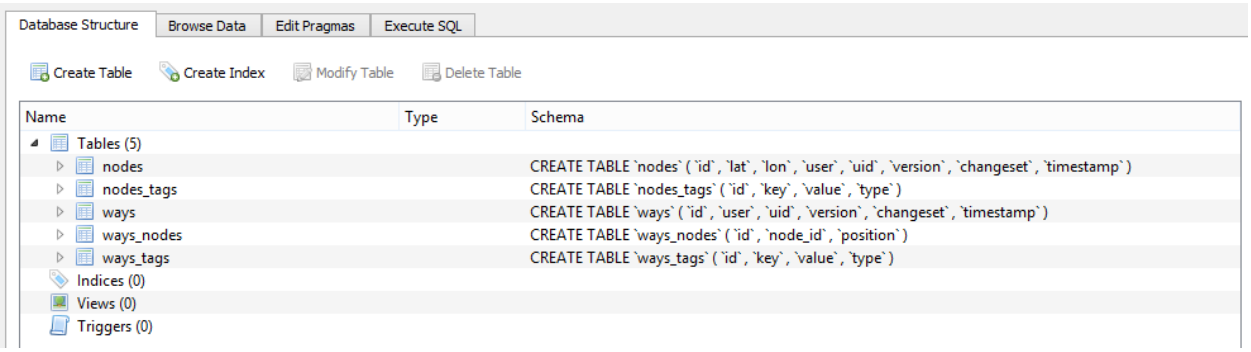
I used regular expression to address this issue:

```
# this function will set the postalcode values to the standard/acceptable values
def update_postalcode(num):
    num = re.findall('[0-9]+',num)[0]
    return num
```

Sample Corrected Postcodes under Ways Tag		
id	**value (bad postcode)**	**value (corrected postcode)**
232136444	96825-9998	96825
269984254	96826-4427	96826
302171171	96734-9998	96734
436080000	96815-2830	96815
440166239	96817-1713	96817

Sample Corrected Postcodes under Nodes Tag		
id	**value (bad postcode)**	**value (corrected postcode)**
2609285445	HI 96819	96819
4263793619	96712-9998	96712
4338125893	96815-2518	96815
4339585890	96815-2834	96815
4339585891	96815-2834	96815

After the data wrangling stage, I created a database in DB Browser for SQLite and named it as “Honolulu_Osm.db”. I also imported the resulting csv files in SQLite to serve as tables of my Database.



Data Overview

File sizes

Name	Date modified	Type	Size
Honolulu_Osm	10/27/16 5:01 PM	Data Base File	2 KB
Honolulu_Osm.db-journal	10/27/16 5:17 PM	DB-JOURNAL File	4 KB
Final Project 3.ipynb	10/27/16 5:17 PM	IPYNB File	14 KB
nodes	10/27/16 5:16 PM	Microsoft Excel C...	20,084 KB
nodes_tags	10/27/16 5:16 PM	Microsoft Excel C...	457 KB
ways	10/27/16 5:16 PM	Microsoft Excel C...	1,480 KB
ways_nodes	10/27/16 5:16 PM	Microsoft Excel C...	6,770 KB
ways_tags	10/27/16 5:16 PM	Microsoft Excel C...	3,473 KB
honolulu_hawaii.osm	10/26/16 10:43 PM	OSM File	51,674 KB

Number of Nodes:

```
import sqlite3
import pandas as pd
```

```
conn = sqlite3.connect('Honolulu_Osm.db')
```

```
c = conn.cursor()
```

```
c.execute('SELECT COUNT(*) FROM nodes')
print c.fetchone()
```

```
(242889,)
```

Number of Ways:

```
c.execute('SELECT COUNT(*) FROM ways')
print c.fetchone()
```

```
(25540,)
```

Number of Unique Users:

```
c.execute('''SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e''')
print c.fetchone()
```

```
(468,)
```

Number of Users appearing only once:

```
c.execute('''SELECT COUNT(*)
FROM
    (SELECT e.user, COUNT(*) as num
     FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
     GROUP BY e.user
     HAVING num=1)  u''')
print c.fetchone()
```

```
(103,)
```

I created a function for queries to display a table:

```
def result(query):
    c.execute(query)
    rows = c.fetchall()
    names = [description[0] for description in c.description]
    df = pd.DataFrame(rows)
    df.columns = names
    print df
```

Top 10 Contributing Users:

```
result('''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''')
```

	user	num
0	Tom_Holland	102158
1	cbbaze	14989
2	ikiya	12807
3	kr4z33	9435
4	Chris Lawrence	9214
5	pdunn	9072
6	aaront	8510
7	woodpeck_fixbot	8446
8	bdiscoe	5097
9	Mele Sax-Barnett	4617

Top 10 appearing amenities:

```
result('''SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10''')
```

	value	num
0	restaurant	196
1	fast_food	102
2	parking	75
3	cafe	62
4	toilets	61
5	fire_station	30
6	waste_basket	29
7	library	24
8	bank	22
9	bench	22

Most popular cuisines:

```
result('''SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
    JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='restaurant') i
    ON nodes_tags.id=i.id
WHERE nodes_tags.key='cuisine'
GROUP BY nodes_tags.value
ORDER BY num DESC Limit 9''')
```

	value	num
0	japanese	7
1	pizza	6
2	american	5
3	chinese	5
4	regional	5
5	international	4
6	asian	3
7	italian	3
8	thai	3

Top 10 leisure:

```
result('''SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags UNION ALL
      SELECT * FROM ways_tags) tags
WHERE tags.key LIKE '%leisure'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 10''')
```

	value	count
0	pitch	338
1	swimming_pool	138
2	park	137
3	picnic_table	65
4	golf_course	36
5	sports_centre	35
6	track	22
7	garden	19
8	nature_reserve	18
9	playground	17

Religion:

```
result('''SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags UNION ALL
      SELECT * FROM ways_tags) tags
WHERE tags.key = 'religion'
GROUP BY tags.value
ORDER BY count DESC''')
```

	value	count
0	christian	25
1	buddhist	7
2	muslim	1

Suggestions for Improving Data:

I think it would be best if Mapzen has a “clean up” tool where users can upload corrections for their dataset. In this way, users who notice corrections will be given a chance to correct the dataset themselves. Correction files can be in a form of xml guided by an acceptable schema.

Benefits:

- It will encourage more users to participate in the objective of having a good and clean dataset for free;
- Cleaning will be faster since more will contribute in the cleaning process.

Anticipated Problems:

- It might result to a more problematic dataset if restrictions on uploading corrections is not properly set;
- Conflicts between users may arise on deciding which correction is better.