OpenStreetMap Data Case Study

Guo, Anqi

Map Area

Honolulu, HI, United States

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

Honolulu is a beautiful city that I wish to visit, so I would like to explore the database about it in advance.

Problems Encountered in the Map

I have noticed several problems regarding this osm file, and they are listed below:

- **Abbreviated street types:**

Some of the street types are in abbreviated form (eg: Ave, Blvd, St) while others aren’t, and I want all of them in full form.

```
def audit_street_type():
    osm_file.seek(0)
    expected = [’Street’,’Avenue’,’Boulevard’,’Drive’,’Court’,’Place’,’Parkway’,
            ’Highway’,’Circle’,’Lane’,’Road’,’Loop’,’Way’,’Walk’,’Square’,
            ’Trail’,’Commons’,’Place’,’Terrace’,’Promenade’]
    for event, elem in ET.iterparse(osm_file, events=(’start’,)):
        if elem.tag == ’way’ or elem.tag == ’node’:
            for tag in elem.iter(’tag’):
                if tag.attrib[’k’]==’addr:street’:
                    street_split = tag.attrib[’v’].split(’ ’)
                    if street_split[-1] not in expected:
                        incorrect_street[street_split[-1]].add(tag.attrib[’v’])
    return incorrect_street
```

So I use the function below to correct street types:

```
    mapping = {”Ave”: ”Avenue”,
               ”Blvd”: ”Boulevard”,
               ”St”: ”Street”,
               ”Blvd.”: ”Boulevard”,
               ”Hwy.”: ”Highway”,
               ”Rd”: ”Road”,
               ”Pl”: ”Place”,
               ”St.”: ”Street”,
               ”street”: ”Street”}
    expected = [’Street’,’Avenue’,’Boulevard’,’Drive’,’Court’,’Place’,’Parkway’,
            ’Highway’,’Circle’,’Lane’,’Road’,’Loop’,’Way’,’Walk’,’Square’,
            ’Trail’,’Commons’,’Place’,’Terrace’]

    if tag.attrib[’k’] == ’addr:street’:
        street_split = tag.attrib[’v’].split(” ”)
        for key in mapping.keys():
            if key == street_split[-1]:
                TAG[’value’] = tag.attrib[’v’].replace(key, mapping[key])
```

- **Uncapitalized street type**

"street" in "Marchant street" is not capitalized, so I use the function above to correct this typo.

- **Incorrect street type**

Some values are not street types, examples are "Enchanted Lakes Shopping Center", "McCarthy Mall", "Fort Street Mall", etc. So I use the function below to revise the "key" value from "street" to "name".

```
        last_word = tag.attrib['v'].split(' ')
        expected = ['Center','Mall']
        if last_word[-1] in expected:
            TAG['key'] = 'name'
            TAG['type'] = 'regular'
```

- **Wrong street name**

A space is missing in "S KingSt", and street type is missing in "Piikoi", "Ala Moana", and "Ala Napunani", and I correct those typos by the function below:

```
        missing = {'Ala Moana': 'Ala Moana Boulevard',
                   'Ala Napunani': 'Ala Napunani Street',
                   'Piikoi': 'Piikoi Street'}

        if tag.attrib['v'] == "S KingSt":
            TAG['value'] = 'S King Street'

        elif tag.attrib['v'] in missing.keys():
            TAG['value'] = missing[tag.attrib['v']]
```

- **Inconsistent state name**

Second level "k" attribution with value "addr:state", and some corresponding "v" attribution are "HI", while others are "Hawaii" :

```
def audit_state_name():
    osm_file.seek(0)
    for event, elem in ET.iterparse(osm_file, events=('start',)):
        if elem.tag == 'way' or elem.tag == 'node':
            for tag in elem.iter('tag'):
                if tag.attrib['k']=='addr:state':
                    if tag.attrib['v'] != 'HI':
                        incorrect_state[tag.attrib['v']] += 1
    return incorrect_state
```

so I use the function below to achieve consistency:

```
    if tag.attrib['k'] == 'addr:state':
        TAG['value'] = 'HI'
```

- **Second level "k" tags with the values "addr:postcode" and "postal_code" have same meaning but different names**

I use the following function to correct this typo:

```
    if tag.attrib['k'] == 'postal_code':
        TAG['key'] = 'postcode'
        TAG['type'] = 'addr'
```

- **Inconsistent postal codes**

```
def audit_postcode_type():
    osm_file.seek(0)
    for event, elem in ET.iterparse(osm_file, events=('start',)):
        if elem.tag == 'node' or elem.tag == 'tag':
            for tag in elem.iter('tag'):
                if tag.attrib['k']=='addr:postcode' or tag.attrib['k']=='postal_code':
                    if len(tag.attrib['v']) != 5:
                        incorrect_postcode[tag.attrib['k']].add(tag.attrib['v'])
    return incorrect_postcode
```

All postal codes are five digits except "96815-2834", so I use the following function to delete the hyphen and last four digits:

```
if tag.attrib['k']=='addr:postcode' or tag.attrib['k']=='postal_code' and len(tag.attrib['v'])>5:
    TAG['value'] = tag.attrib['v'][:5]
```

- **Inconsistent phone numbers**

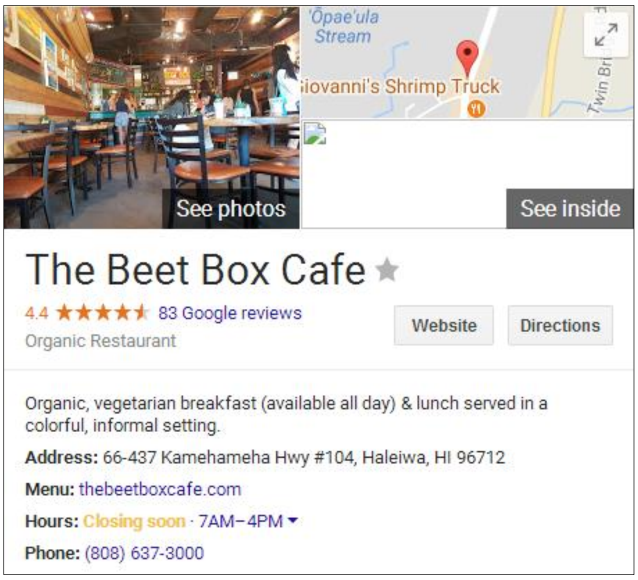
Some phone numbers have plus sign, country code, area code, dots, or hyphen, while others don't:

```
def audit_phone_number():
    phone_re = re.compile(r'^\d$')
    for event, elem in ET.iterparse(osm_file, events=('start',)):
        if elem.tag == 'way' or elem.tag == 'node':
            for tag in elem.iter('tag'):
                if 'phone' in tag.attrib['k']:
                    if not phone_re.search(tag.attrib['v']) or len(tag.attrib['v'] == 11):
                        incorrect_phone[tag.attrib['k']].add((tag.attrib['v']))
    return incorrect_phone
```

So I use the function below to unify the phone number form:

```
    if 'phone' in tag.attrib['k']:
        digit = ''.join(filter(lambda x: x.isdigit(),tag.attrib['v']))
        if len(digit)== 11:
            TAG['value'] = '+' + digit
        elif len(digit) == 7:
            TAG['value'] = '+1808' + digit
        else:
            TAG['value'] = '+1' + digit[:11]
```

Note: phone number of "Beet Box Café" lacks area code, so I googled it and added up area code "808" to it.



The Beet Box Cafe
4.4 ★★★★★ 83 Google reviews
Organic Restaurant

Organic, vegetarian breakfast (available all day) & lunch served in a colorful, informal setting.

Address: 66-437 Kamehameha Hwy #104, Haleiwa, HI 96712

Menu: thebeetboxcafe.com

Hours: Closing soon · 7AM–4PM

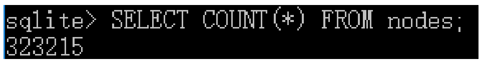
Phone: (808) 637-3000

Data Overview and Additional Ideas

File Sizes

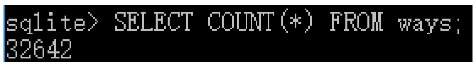
honolulu_hawaii.osm ············ 66.0MB
honolulu_hawaii.db ··············· 37.3MB
nodes.csv ································· 26.0MB
nodes_tags.csv ························ 664KB
ways.csv ···································· 1.85MB
ways_tags.csv ··························· 3.69MB
ways_nodes.csv ························ 8.75MB

Number of nodes



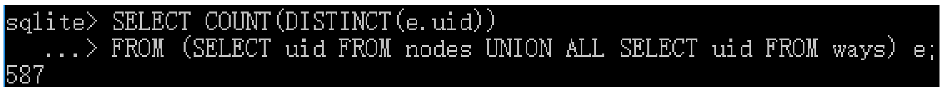
```
sqlite> SELECT COUNT(*) FROM nodes;
323215
```

Number of ways



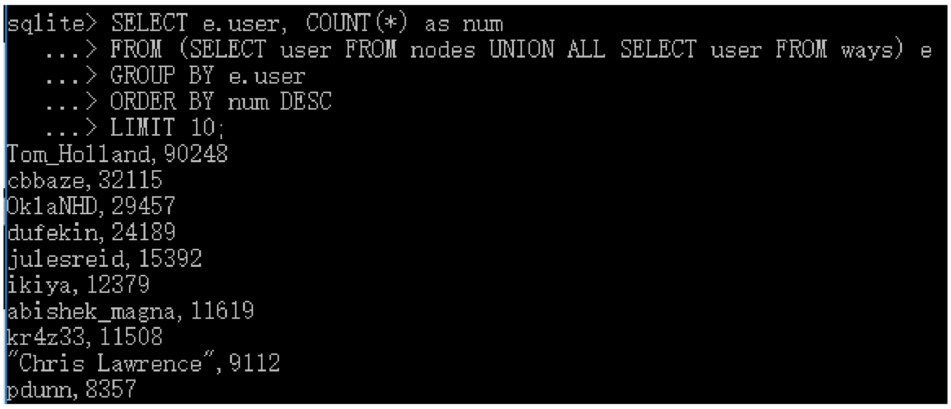
```
sqlite> SELECT COUNT(*) FROM ways;
32642
```

Number of unique users



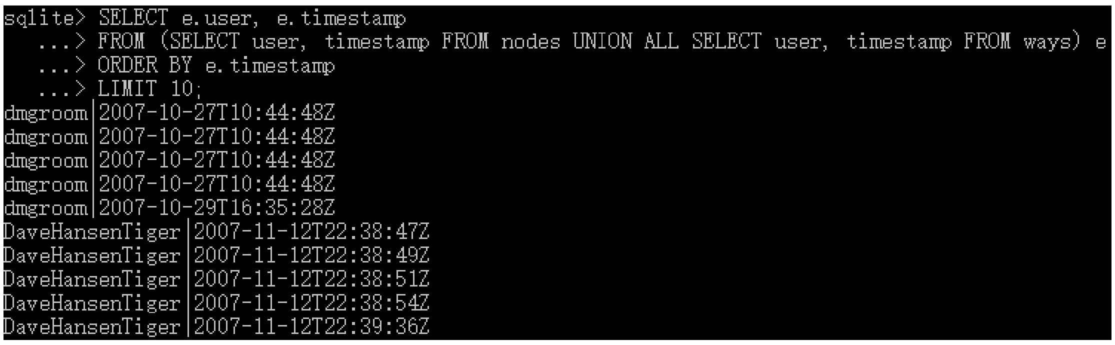
```
sqlite> SELECT COUNT(DISTINCT(e.uid))
   ...> FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
587
```

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;
Tom_Holland,90248
cbbaze,32115
Ok1aNHD,29457
dufekin,24189
julesreid,15392
ikiya,12379
abishek_magna,11619
kr4z33,11508
"Chris Lawrence",9112
pdunn,8357
```

Top 10 oldest post



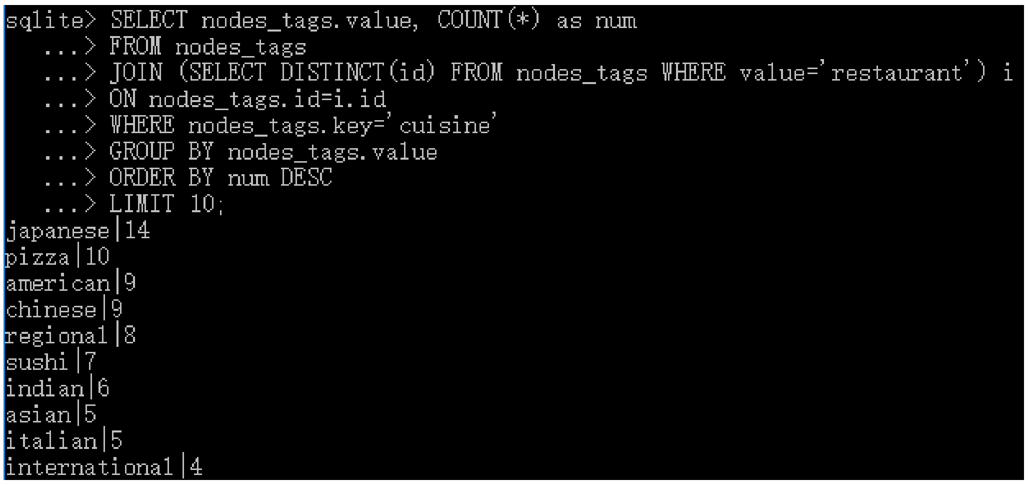
```
sqlite> SELECT e.user, e.timestamp
   ...> FROM (SELECT user, timestamp FROM nodes UNION ALL SELECT user, timestamp FROM ways) e
   ...> ORDER BY e.timestamp
   ...> LIMIT 10;
dmgroom|2007-10-27T10:44:48Z
dmgroom|2007-10-27T10:44:48Z
dmgroom|2007-10-27T10:44:48Z
dmgroom|2007-10-27T10:44:48Z
dmgroom|2007-10-29T16:35:28Z
DaveHansenTiger|2007-11-12T22:38:47Z
DaveHansenTiger|2007-11-12T22:38:49Z
DaveHansenTiger|2007-11-12T22:38:51Z
DaveHansenTiger|2007-11-12T22:38:54Z
DaveHansenTiger|2007-11-12T22:39:36Z
```

Additional Ideas

I googled the history of OpenStreetMap, and found that it is launched on 2004, but the query shows that oldest post is on 2007, so I guess that the website was not really "open" during the first 3 years.

Additional Data Exploration

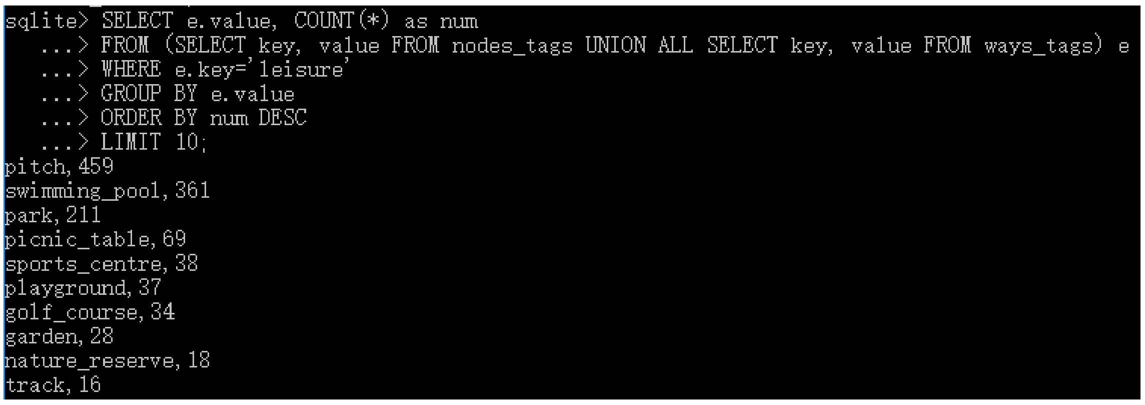
Most popular cuisines



```
sqlite> 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 10;
japanese|14
pizza|10
american|9
chinese|9
regional|8
sushi|7
indian|6
asian|5
italian|5
international|4
```

Japanese is the 2nd largest ethnic group in Hawaii, which explains the large amount of Japanese restaurants in Honolulu.

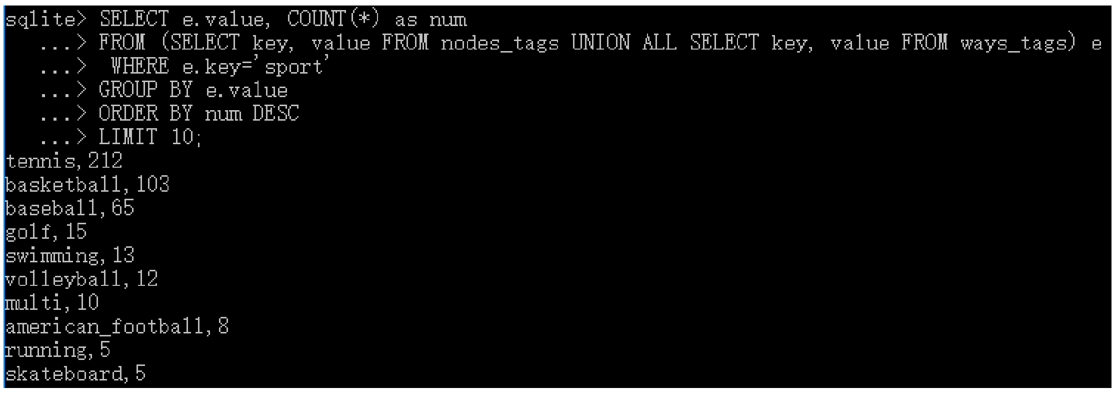
Most popular leisure spots



```
sqlite> SELECT e.value, COUNT(*) as num
   ...> FROM (SELECT key, value FROM nodes_tags UNION ALL SELECT key, value FROM ways_tags) e
   ...> WHERE e.key='leisure'
   ...> GROUP BY e.value
   ...> ORDER BY num DESC
   ...> LIMIT 10;
pitch,459
swimming_pool,361
park,211
picnic_table,69
sports_centre,38
playground,37
golf_course,34
garden,28
nature_reserve,18
track,16
```

The amount of swimming pools in Honolulu did shock me, since I feel that it is better and cooler to swim in the ocean nearby.

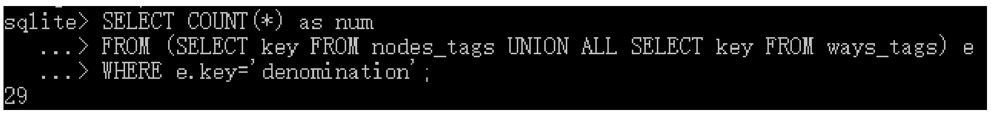
Most popular sports



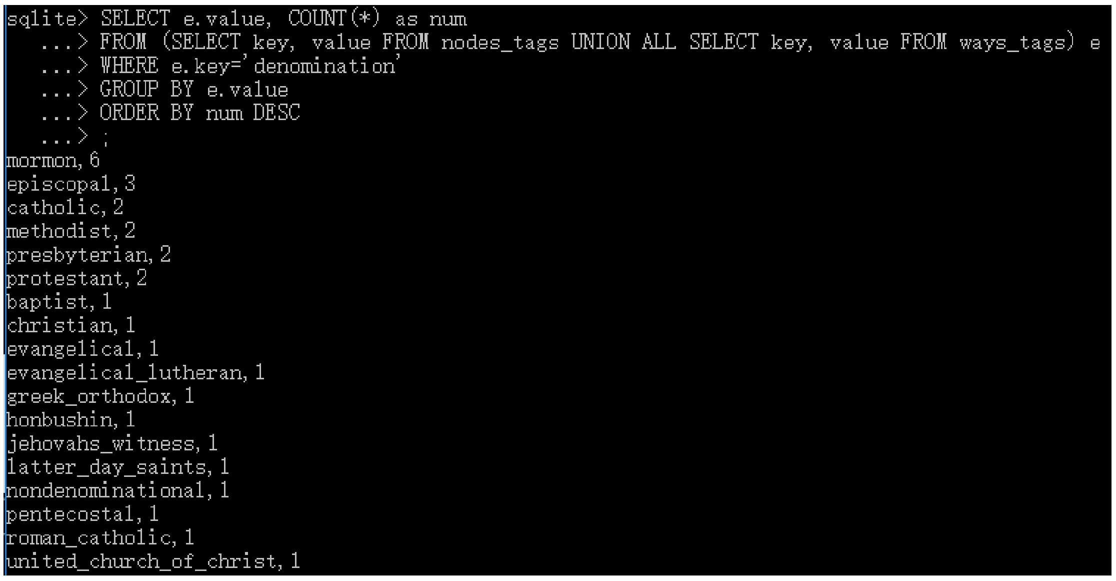
```
sqlite> SELECT e.value, COUNT(*) as num
   ...> FROM (SELECT key, value FROM nodes_tags UNION ALL SELECT key, value FROM ways_tags) e
   ...>  WHERE e.key='sport'
   ...> GROUP BY e.value
   ...> ORDER BY num DESC
   ...> LIMIT 10;
tennis,212
basketball,103
baseball,65
golf,15
swimming,13
volleyball,12
multi,10
american_football,8
running,5
skateboard,5
```

Living in island surrounded by ocean, people in Honolulu still prefer land sports like tennis and basketball.

Denominations of Churches



```
sqlite> SELECT COUNT(*) as num
   ...> FROM (SELECT key FROM nodes_tags UNION ALL SELECT key FROM ways_tags) e
   ...> WHERE e.key='denomination';
29
```



```
sqlite> SELECT e.value, COUNT(*) as num
   ...> FROM (SELECT key, value FROM nodes_tags UNION ALL SELECT key, value FROM ways_tags) e
   ...> WHERE e.key='denomination'
   ...> GROUP BY e.value
   ...> ORDER BY num DESC
   ...> ;
mormon,6
episcopal,3
catholic,2
methodist,2
presbyterian,2
protestant,2
baptist,1
christian,1
evangelical,1
evangelical_lutheran,1
greek_orthodox,1
honbushin,1
jehovahs_witness,1
latter_day_saints,1
nondenominational,1
pentecostal,1
roman_catholic,1
united_church_of_christ,1
```

I am surprised to find that out of 29 churches in Honolulu, 7 are Mormon (Latter_Day_Saints is Mormon too) which preaches false teaching of Gospel.

Conclusion

After reviewing the data about Honolulu, I feel that it is pretty dirty and needs further cleaning. Take phone number for example, incomplete phone numbers are invalid as well, not to mention consistency issue. Therefore, I suggest setting specialized format for users while inputing data.

References

https://classroom.udacity.com/nanodegrees/nd002-cn-advanced/parts/7f46cd58-8041-4d9d-88a5-4b7c6f7be63e/modules/63f680db-5dc5-4dce-acdb-ac4909d2db2e/lessons/5436095827/concepts/54908788190923