Open Street Map Analysis - Oahu

Introduction

For this analysis, data from the island of Oahu in the State of Hawaii was exported from the Open Street Map ('OSM') database.

http://www.openstreetmap.org/#map=10/21.4901/-158.0795

Having lived on Oahu for several years, I was interested in performing an analysis on the area. Perhaps I will learn a something new about the island!

To make testing quicker, I extracted a sample of data from the original to see what information is contained within the file. While the original .osm file is 67.1 megabytes, the sample I took was only 3.42 megabytes in size. This is much easier to manage.

Auditing Data

To audit the information within the .osm file, I first took a look at what key values were stored in the tags of the file. To do this, I wrote a script that recorded and counted the number of times a key appeared. Within the sample file, there were 230 types of keys.

Looking through the list, I noticed that a number of these keys were labeled with "addr:". As these keys contain address information, I audited some of the address information to see if there were any inconsistencies in the data.

Examining this information revealed data quality issues with regards to City and State Names.

For example, when looking at the City Names, the following entries appeared:
"Hale'iwa": 19,
'Haleiwa': 8,
'Hale‘iwa': 3

The inconsistent spelling of Hale'iwa would impact a data analysis because not all records would be retrieved if "Hale'iwa" was searched for.

The same issue with consistency appeared when the state was looked at:
'HI': 1167,
'Hawaii': 65,
'Hi': 1,
'hawaii': 3,
'hi': 4

A third issue arose with regard to street names. Utilizing the sample code from **Data Quality: Example Using our Blueprint**, multiple issues appeared regarding the street types. A few of these issues consist of:

1.) Overabbreviated street names ('St.' instead of 'Street')
2.) The lack of "street type" in the street name ('King')
3.) Typos and lack of capitalization ('highway')
4.) City name in the Street name ('Honolulu')

For the purpose of this analysis, issues 1 and 3 will be addressed. These issues can be corrected via a script which can check for the incorrect values within a mapping dictionary and correct them. Issue 2 will need a more manual process where the street names that are missing a street type will need to be validated with another data source to ensure their validity. Similarily, tags that fall under issue 4 will need to be further examined for validity. Additionally, there is a need for further inspection to ensure that a tag has the appropriate "addr:city" value contained within other tags within the node or way.

Cleaning the Data

To start cleaning the data, the first question to be asked is "Which data needs to be cleaned?" From the results of the audit process above, the following pieces of information need to be fixed before it gets imported into a SQL datbase for further analysis:

1. Overabbreviated Street Names
2. Typos and lack of capitalization in street names
3. Inconsistent spelling of City names
4. Inconsistent spelling and abbreviation of the State

The second question that now needs to be asked is "How will this data be corrected?"

Several python scripts will be utilized which will iterate through the .osm file, and identify tags which contain "addr:street", "addr:city", and "addr:state". A mapping dictionary will also be created for each attribute, so any incorrect values will be updated with the correct one. This will ensure uniformity within the data set.

The main body of code used to clean and convert the OSM data into several CSV files was provided in the **Open Street Map Case Study: Preparing for Database**. This starter code was then enhanced to include several functions and mapping dictionaries to clean up the data and ensure consistency and uniformity throughout the dataset.

The script created five different files based on the tag elements: *nodes*, *nodes_tags*, which are tags nested within a node, *ways*, *ways_tags* which are tags nested within a way, and *ways_nodes* which are node references nested within a ways element.

Building the SQL Database

After the data has been exported into the .csv files, they need to be imported into a sqlite3 database, oahu.db. Before these files could be imported, a schema needs to be created. Five tables were created; one for each file. After the tables have been created, the .csvs were imported utilizing the .import command.

Analyzing the Data

Verifying the cleaning was successful

With the data loaded into the sqlite3 database, analysis can now be done.

The first query is looking at tags that contain a state. This is to verify that the cleaning process was successful before the data was exported into the .csv.

In [ ]:

```
select value, count(*)
from Way_Tags
where type = 'addr'
and key = 'state';
```

In [ ]:

```
Output:
HI,914
```

As 'HI' is the only value returned for state, the cleaning process was successful as no other values appeared. Now that the cleaning process has been verified, the data can be analyzed.

Most Tags

From processing the .osm file, there were a couple of things I was curious about, and I can now query the database to get some answers.

One of the questions I wanted to know the answer to is "Which Nodes have the most tags?"

To find this answer, I wrote the following query:

In [ ]:

```
Select n.id, count(*)
from Nodes as n, Node_Tags as nt
where n.id = nt.id
group by n.id
order by count(*) desc
limit 10;
```

In [ ]:

```
Output:
316949921,43
3831920861,35
```

```
21442033,24
150921582,17
150919907,16
150920535,16
368391861,16
150920073,15
150920482,15
4175109984,15
```

This query showed that node 316949921 has the most tags, with 43. What does node 316949921 point to?

In [ ]:

```
Select * from node_tags where id = 316949921;
```

In [ ]:

```
Output (sample):
316949921,abbreviation,Hawaii,name
316949921,be,"Ð"Ð°Ð²Ð°Ñ-",name
316949921,cdo,Hawaii,name
316949921,continent,"North America",is_in
316949921,country,USA,is_in
316949921,country_code,US,is_in
316949921,cs,Havaj,name
316949921,date,2014-01-01,population
316949921,en,Hawaii,name
316949921,eo,Havajo,name
316949921,es,"HawÃ¡i",name
316949921,fr,"HawaÃ¯",name
316949921,fy,"HawaÃ¯",name
```

It looks like node 316949921 points to the basic information for the State of Hawaii. It does look like there are a few issues, as shown above. It appears that some of the tags contain the foreign name for Hawaii in multiple languages. As not all languages use the Latin character set, some of the values appear unreadable.

Referenced Streets

In [ ]:

```
select count(distinct t.value)
from (select * from node_tags UNION select * from way_tags) as t
where t.key = 'street';
```

In [ ]:

```
Output:
362
```

The above query looks at the total number of unique streets that are identified within the nodes_tags and ways_tags. Which streets are referenced the most?

In [ ]:

```
select t.value, count(*)
from (select * from node_tags UNION select * from way_tags) as t
where t.key = 'street'
group by t.value
order by count(*) desc
limit 5;
```

In [ ]:

```
Output:
"South King Street",67
"Ala Moana Boulevard",55
"Kamehameha Highway",53
"Ala Wai Boulevard",46
"KalÄkaua Avenue",44
```

The above query and output shows the five most referenced streets on the island of Oahu. This makes sense as each street is a very popular roadway.

There is an issue with "Kalakaua Avenue" that can affect the quality of the data set. The Hawaiian language and names are very commonly used as street names throughout the State of Hawaii, and the use of the okina (`), and overlines are frequently used with names. This can cause an issue with consistency within the data set, as some people may choose to use ` instead of ', or ignore any accent marks over the letters.

Fast Food Restaurants

In [ ]:

```
select f.value, count(*)
```

```
from (select * from node_tags UNION select * from way_tags) as f

where f.id in (select t.id from (select * from node_tags UNION select * from way_tags)
 as t where value = 'fast_food')

and (f.key = 'name' OR f.type = 'name')

group by f.value

order by count(*) desc

limit 10;
```

In [ ]:

```
Output:

McDonald's|20

Subway|18

Jack in the Box|9

Zippy's|9

Burger King|8

Taco Bell|8

Panda Express|4

Jack in the box|3

Jamba Juice|3

Aloha Burrito Shop|2
```

The above query searches for the names of the most common fast food restaurants. As shown above, McDonald's has the most restaurants on Oahu. However Jack in the Box is listed twice; this is because of inconsistent spelling of the restaurants name, resulting in it appearing twice. While this does not change it's ranking, it does change the number of stores, making it appear that there are less restaurants than there are.

Biggest Contributors

In [ ]:

```
select count(distinct u.uid)

from (select id, uid from nodes UNION select id, uid from ways) as U;
```

In [ ]:

```
Output:

572
```

The above query looks at the total number of contributors to this data set. Of the 572 users providing information, who are the biggest contributors?

In [ ]:

```
select u.uid, u.user, count(*)
from (select id, uid, user from nodes UNION select id, uid, user from ways) as U
group by u.uid, u.user
order by count(*) desc
limit 10;
```

In [ ]:

```
Output:
574654|Tom_Holland|90479
2257893|cbbaze|32248
302871|OklaNHD|29785
30521|dufekin|24212
4501014|julesreid|15360
69966|ikiya|12400
5412936|abishek_magna|11622
604586|kr4z33|10999
36121|Chris Lawrence|9145
32360|pdunn|8415
```

User 574654, or Tom_Holland contributed the most information, with over 90,000 entries!

Room for Improvement

While the Open Street Map data is certainly impressive, there is much room for improvement to ensure data quality. As discussed throughout my analysis, some of the key issues with data quality are as follows:

1. Foreign language translation
2. Inconsistent capitalization of names
3. Inconsistent use of accent marks over letters

In an area like Hawaii, where street names, buildings, parks, and locations utilize the Hawaiian language, there is an issue with consistency in the data. For words and names with the okina, people may enter the name with an apostrophe, a grave accent, or just omit the character entirely. This can lead to duplicate entries, impacting the quality of data.

One proposed solution for the capitalization issue is to create a system that automatically capitalizes the first letter of each word. This ensures that there will be no capitalization issues. In the event that there is a reason for a name to not be capitalized, a dictionary of exceptions can be made to correct the value afterwards.

In regards to the issue with the accent marks used within words, a function could be created to eliminate the use of them. So if a character is an 'a' with an overline or a tilde, it'll automatically be converted into an 'a'. While this will ensure uniformity of the data, it may also upset people who want the correct accent marks to appear over the letters. Similarily with the capitalization issue, this could be resolved if after a standardizing process is run, a dictionary of exceptions could be used to ensure correctness data.

While the idea of a dictionary of exceptions is certainly interesting, it would not be efficient in processing. This analysis only looked at a very small area, an island! Because of the language, there would be a lot of exceptions required because a lot of names utilize special characters. With over 100 other languages in the world, this dictionary would become very large very quickly.

The Open Street Map data set is very impressive, however due to its information being crowdsourced, the data quality issues become very apparent. Consistency and uniformity appear one of the biggest issues here, as there may be typos, misspellings, and the use of incorrect keys to describe a value.