OpenStreetMap Data Case Study

1. Introduction

1.1 Map Area

Honolulu, HI, United States - [https://mapzen.com/data/metro-extracts/metro/honolulu_hawaii/] - [https://www.openstreetmap.org/search?query=Honolulu#map=12/21.3255/-157.8014]

Honolulu is of the capital of state of Hawaii which is the most recent state to have joined the United States. It is also the only U.S. state located in Oceania and outside North America.

1.2 File Size

Based on the above output, we can read that the size of our dataset is **66.2 megabytes** which is greater than the required 50 megabytes.

2. Problem Encountered in the Map

We noticed that there are several problems with this dataset, which I will discuss in the following parts:

- Inconsistent ("HI 96819", "96817-1713", etc) and incorrect postal codes (Honolulu area postal codes can be found from this USPS page)
- Inconsistent phone number format ("+1 808 536 2236", "808-923-7024", etc)
- Street abbreviation ("Meheula Pkwy", "N King St", etc. Street suffix abbreviations can be found from this USPS page)

The file `data_wrangling.py` and `schema.py` will help us clean the dataset and convert it into `.csv` files in this part.

2.1 Postal Code

A postal code is a series of letters, digits or both, sometimes including spaces or punctuation, included in a postal address for ther purpose of sorting mails. In the United States, postal code is also called zip code which is composed of five numbers. There is an extended format for U.S. zip code which contains the basic five-digit zip code, followed by a hyphen and four additional digits.

In this project, we will use the five-digit format. First, I would like to print out some zip code examples that are included in our OSM file, which will give us a clearer understanding of the potential problems of this variable.

- 9
- 96701
- 96712-9998
- 96734-9998
- 96814
- 96815
- 96815-2518
- HI 96819
- ...

Honolulu area has 41 different zip codes, 12 of which are general zip codes while the rest are for particular PO Boxes or companies. They can be found from this USPS page. Now, we will try to clean the dataset that we have in hand based on the following three criteria.

- If zip code has five digits but not one of the 41 Honolulu area zip codes or is just 9, we remove the corresponding record.
- If zip code has 9 digits, we firstly will keep the first five digits only. Then we will see if the new zip code is one of the 41 Honolulu area zip codes. If yes, we update old zip code to new ones. If no, we remove the corresponding record.
- If zip code is recorded as "HI 96819", we update old zip code to "96819"

The output tells us that 163 records were removed here, and the rest 1323 records all follow our criteria. We will save the cleaned file as `cleaned_postal_file.xml`. We will use this one in the next part instead of the original dirty OSM file.

2.2 Phone Number

The U.S. phone number generally follows the format of `+1 808 XXX XXXX`. The first digit `+1` is the country code of U.S. The three digits `808` from the first space to second space is the area code within the United States. Different digits stand for different areas and `808` which stands for Honolulu area is an example here. The last seven digits are phone numbers which vary from individual to individual, and we usually add a space between the third and fourth digit.

There are different kinds of phone number formats here. Sometimes, we choose to ignore the country code `+1` or the plus sign `+`, replace the space with hyphen, use parentheses to emphasize the area code, etc. Due to the amount of phone numbers here, we will only list a few examples to have a clearer understanding of phone number formats.

- +1 808 536 2236
- 808-923-7024
- +1 808 432 2000
- +1 (808) 733-1540
- 1-808-955-7470
- 1.888.236.0799
- 6373000
- +18089234852
- 8088458044
- ...

In this project, we would like to convert every phone number into the format of `808-536-2236`. The country code `+1` is omitted since there is no part of Honolulu that is out of the United States. The criteria of cleaning our phone number format is listed below.

- First, we would like to convert every phone numbers into digit-only format by using regular expression.

- For those values with length equal to 10, we just convert it into `XXX-XXX-XXXX` format directly.
- For those values with length equal to 11, we remove the first country code and then covert it into `XXX-XXX-XXXX` format.

- There are two kinds of special points. The first kind has only one point which contains the value "80828412708082714836" which stands for two phone numbers and we convert it into "808-284-1270, 808-271-4836". The second kind only contains the last 7 digits. Here, we help add the area code `808` and then convert it.

We will save the cleaned file as `cleaned_phone_file.xml`. We will use this one in the next part instead of the `cleaned_postal_file.xml` file.

2.3 Addres

In the United States, we have many ways to record our street names because of the abbreviation. For example, when we try to write down the street name of Chicago O'Hare International Airport, we can choose *10000 W O'Hare Ave*, *10000 West O'Hare Avenue*, or some other possible abbreviations, all of which are correct.

In this project, we can also find out that different records here were recorded in different ways. Some of them are using abbreviations while the rest are not. Here, we would like not to use abbreviation here, which means that *10000 West O'Hare Avenue* is going to be the example format that we will have after cleaning the data. Because for people whose cannot speak fluent English may not be able to read or understand abbreviations clearly. For example, "St." can either stand for "Street" or "Saint" even though this will not be a problem in our dataset here.

The following are several examples. The left-hand side names are original street names while the right-side ones are updated names.

- Ena Rd => Ena Road
- kinalau Pl => kinalau Place
- Kipapa Dr => Kipapa Drive
- Meheula Pkwy => Meheula Parkway
- N King St => North King Street
- ... => ...

We will save the cleaned file as `cleaned_street_file.xml`. We will use this one in the next part instead of the `cleaned_phone_file.xml` file.

2.4 Output csv Files

In this part, we will convert our cleaned `cleaned_street_file.xml` file into several `.csv` files.

The **node** field should hold a dictionary of the following top level node attributes: - id - user - uid - version - lat - lon - timestamp - changeset

The **node_tags** field should hold a list of dictionaries, one per secondary tag. Secondary tags are child tags of node which have the tag name/type: "tag". Each dictionary should have the following fields from the secondary tag attributes: - id: the top level node id attribute value - key: the full tag "k" attribute value if no colon is present or the characters after the colon if one is. - value: the tag "v" attribute value - type: either the characters before the colon in the tag "k" value or "regular" if a colon is not present.

The **way** field should hold a dictionary of the following top level way attributes: - id - user - uid - version - timestamp - changeset

The **way_tags** field should again hold a list of dictionaries, following the exact same rules as for "node_tags".

Additionally, the dictionary should have a field **way_nodes**. **way_nodes** should hold a list of dictionaries, one for each nd child tag. Each dictionary should have the fields: - id: the top level element (way) id - node_id: the ref attribute value of the nd tag - position: the index starting at 0 of the nd tag i.e. what order the nd tag appears within the way element

Finally, we save these five dictionaries as five `.csv` files which can be loaded into SQL.

- The size of our nodes.csv file is 26.4 megabytes.
- The size of our nodes_tags.csv file is 0.7 megabytes.
- The size of our ways.csv file is 1.9 megabytes.
- The size of our ways_nodes.csv file is 9.1 megabytes.
- The size of our ways_tags.csv file is 3.8 megabytes.

3. SQL Database

3.1 Total Users

In this database, `uid` stands for the user ID of the last person who helped edit the corresponding record. We would like to see the frequency of our last editors and sort the result by descending order.

```
SELECT COUNT(DISTINCT(c.uid)) AS count
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) as c
WHERE c.uid != "";
```

The output shows that there are 583 total users which made contributions here.

3.2 Nodes and Ways

```
SELECT COUNT(*) FROM nodes WHERE id != "";
```

```
SELECT COUNT(*) FROM ways WHERE id != "";
```

The outputs show that there are 323776 nodes and 32697 ways.

3.3 Number of Schools

```
SELECT COUNT(t.value) AS count
FROM (SELECT * FROM nodes_tags UNION ALL SELECT * FROM ways_tags) as t
WHERE t.value = "school"
ORDER BY count DESC;
```

The output shows us that there are 363 schools in Honolulu area.

4. Additional Ideas

4.1 Postal Code

Here, we would like to see how many postal codes out of the 41 are included here and their corresponding frequencies. If the frequencies of some particular postal codes are not high, we may think about combining some postal codes together into a new one, which increases the efficiency of postal service.

```
SELECT t.value, COUNT(*) AS count
FROM (SELECT * FROM nodes_tags UNION ALL SELECT * FROM ways_tags) as t
WHERE t.key = "postcode"
GROUP BY t.value
ORDER BY count;
```

```
value count
96841   1
96825   4
96816   13
96821   16
96818   18
96817   46
96819   54
96813   99
```

```
96826   116
96814   196
96815   317
96822   415
```

We notice that frequencies of postal code `96841`, `96825`, `96816`, `96821` and `96818` are all less than 20 and total count of these five are 52 only. We read that code `96841` is reserved for a specific company and the rest four codes are for general areas instead of PO Boxes. We may take a look at the geometric area to see if combining these codes together is possible.

There are several issues here. The dataset may not be able to reflect the true distributions of USPS mails to these 41 postal codes in Honolulu area. If we really conbine some postal codes together based on it, it may be harder for mailmen to sort the mails and it may lead to delayed delivery.

4.2 Contributors

We have found that there are 583 contributors in the datasets. We can also try to rank the contributions that they have made to this dataset by calculating the count of their records and sorting the result by descending order.

We can think that the higher count (frequency) one user is, the more contributions that he/she has made. Based on the rank, we can easily know that which individuals tend to make more contributions and which ones make less contributions. For those individuals who make more contributions, we can try to figure out ways to keep them working on and making more. For those individuals who make less contributions, we should find some ways to encourage them to do this.

Like the previous part, we also have a potential issue here. Here, we just rank different contributors based on how many contributions that they have made instead of their qualities of contributions. We really do not want to have someone who just messes with our dataset, which will definitely influence the number of our customers.

5. Conclusion

The dataset of Honolulu, HI area is not clean at the beginning. There are inconsistent or incorrect postal codes, inconsistent phone number formats, etc. But this dataset is still be able to provide us some interesting information that we can learn about Honolulu area, like how many schools are there in Honolulu, if postal codes can be combined together, etc.

Reference

- Element, OpenStreetMap Wiki, https://wiki.openstreetmap.org/wiki/Elements
- Data Wrangling, Udacity, https://www.udacity.com/
- Street Suffix Abbreviations, United States Postal Services, https://pe.usps.com/text/pub28/28apc_002.htm
- Zip codes of Honolulu, HI, United States Postal Services, https://tools.usps.com/go/ZipLookupResultsAction!input.action?items=30&page=1&companyName=&address1=&address2=&city=Honolulu&state=HI&zip=96795