Neil Amin
Project 4 Report

Problems encountered in map

In my project there were a few problems that I encountered. Some of these problems were technical problems that I encountered in my code and others were problems that I encountered with some of the logic involved. I also had some errors with the getting the OSM data that would be best for my project.

While trying to get the OSM file from the Openstreetmap site, I encountered a few problems. The first problem I had was trying to get an area that was perfect for my project. I wanted to get an area that had street names, amenities, etc. that I would be able to recognize and understand what the expect result should be. I also had to get an area that fit my size constraints of about 50-80 MB. Honolulu ended up being my best option. The next problem I had was with trying to create a sample file to test on. However, after a little bit of configuring on the code provided by Udacity, I was able to get it working.

In the actual dataset I found a problem with the street names. For example, the same street names would have multiple abbreviations for the same thing such “main st” or “main street”. I standardized these by using create a dictionary and writing a function to update the streets with these issues.

Another problem in the actual dataset was lack of standardization of the amenities. Some amenities were lowercase “restaurant” and some were uppercase “RESTAURANT”. So I updated it the same way as the street names to only capitalize the first letter of each.

Another error that I had was with the utf-8 encoding. The utf-8 encoding was adding the letter b in front of every element in my csv. This was due to a change in Python 3 that I do not think was accounted for in the shell code given by Udacity. I was able to fix this by: 1) removing the entire class UnicodeDictWriter 2) adding *encoding='utf8'* as a parameter to all 5 of the *codecs.open()* statements 3) immediately below the codecs.open() code chunk, replacing *UnicodeDictWriter* with *csv.DictWriter*.

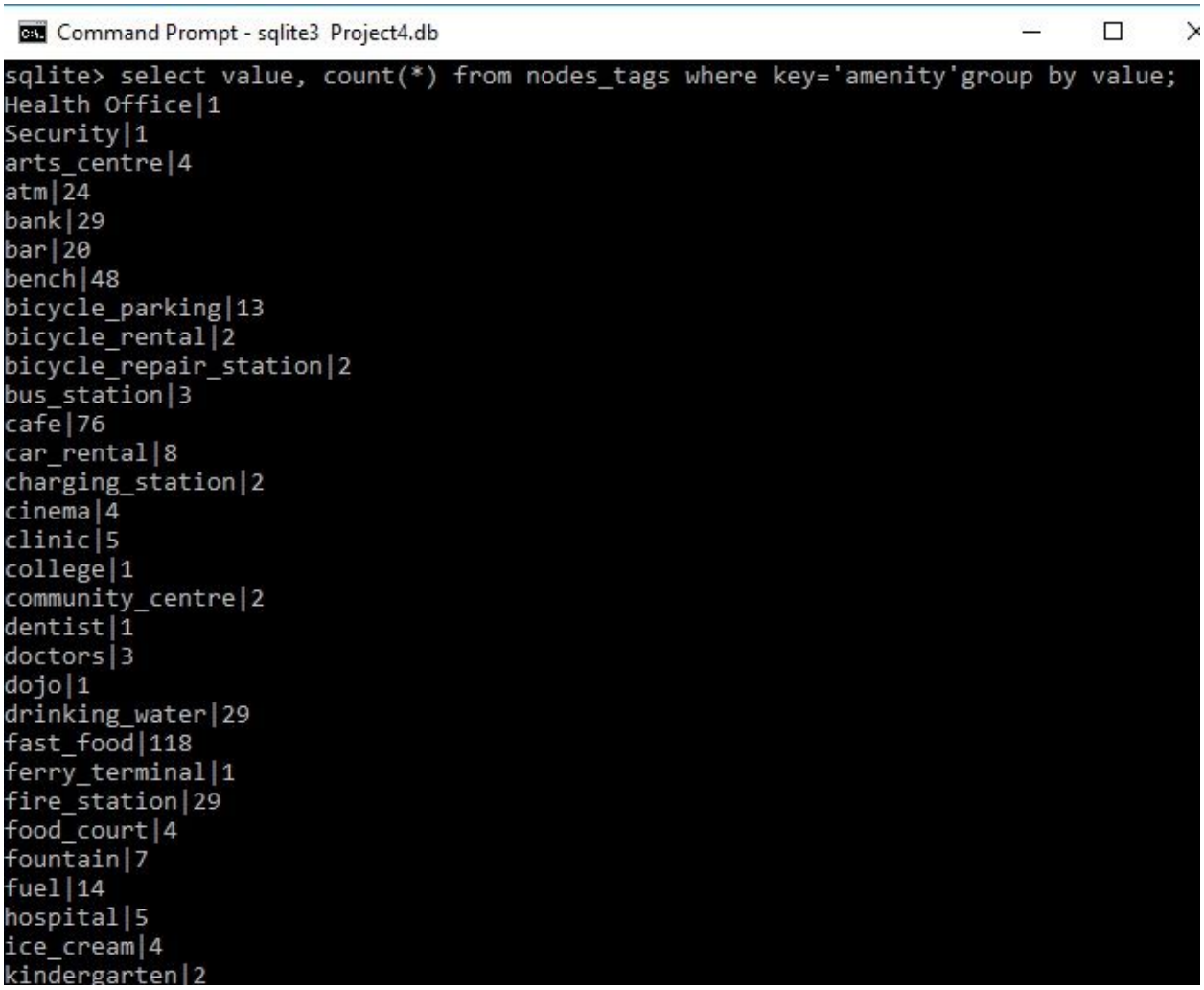
```
264 # ================================================== #
265 #               Main Function                        #
266 # ================================================== #
267 def process_map(file_in, validate):
268     """Iteratively process each XML element and write to csv(s)"""
269 
270     with codecs.open(NODES_PATH, 'w', encoding = 'utf8') as nodes_file, \
271          codecs.open(NODE_TAGS_PATH, 'w', encoding = 'utf8') as nodes_tags_file, \
272          codecs.open(WAYS_PATH, 'w', encoding = 'utf8') as ways_file, \
273          codecs.open(WAY_NODES_PATH, 'w', encoding = 'utf8') as way_nodes_file, \
274          codecs.open(WAY_TAGS_PATH, 'w', encoding = 'utf8') as way_tags_file:
275 
276         nodes_writer = csv.DictWriter(nodes_file, NODE_FIELDS)
277         node_tags_writer = csv.DictWriter(nodes_tags_file, NODE_TAGS_FIELDS)
278         ways_writer = csv.DictWriter(ways_file, WAY_FIELDS)
279         way_nodes_writer = csv.DictWriter(way_nodes_file, WAY_NODES_FIELDS)
280         way_tags_writer = csv.DictWriter(way_tags_file, WAY_TAGS_FIELDS)
```

Another major error that I encountered was importing my CSVs into the database. When I imported my CSVs using *.import name.csv name* my columns were all merging into one column. I fixed this by adding *.separator ,* which let SQL know that a comma was the separator for my data.

Overview of the Data

The data had some interesting trends that were found via useful SQL queries. I included some screenshots of some the highlights.

Count of each type of amenity:



Number of Unique Users:

```
sqlite> select count(distinct user) from nodes;
503
```

Count of nodes and ways:

```
sqlite> select (select count(distinct user) from ways) + (select count(distinct user) from nodes);
893
```

<u>Other ideas about the datasets</u>

Another idea that I had for the dataset was to add in a feature for joint complexes such as apartments or shopping centers. Many of these places may be in the same complex as others which is why they should be jointly held somehow in the data most easily by adding another column for shopping center. This could be a huge benefit because it could allow for more advanced queries such as finding all the breakdown of the number of each amenity in

each shopping center. The main downfall to this idea is that it could expand the size of data which could lead to a slowdown in a query speed and an increase in storage costs.