Project 3: OpenStreetMap Data Wrangling with SQL (San Francisco/Bay Area)

Data Audit

First thing I did was investigate how many unique tags were in the data set as well as the patterns in the said data set.

Unique Tags

```
{'bounds': 1,
 'member': 53772,
 'nd': 7448039,
 'node': 6278429,
 'osm': 1,
 'relation': 6012,
 'tag': 1994316,
 'way': 770438}
```

Tag Patterns

```
{'lower': 128299,
'lower_colon': 69108,
'other': 2547,
'problemchars': 13}
```

Data Auditing

I decided to audit street names, city names, postal codes, and amenities as I suspected that these values could be problematic based on what I learned from the Udacity data wrangling module. I found errors in three of the four categories (street names, city names, postal codes).

Street Names

Street names proved to be tricky to clean. Based on a familiarity of the area of San Francisco County, I tracked down the street names that were missing their suffix (i.e. St, Ave, Blvd) and inputted where possible (For example, Broadway became Broadway Avenue). I also went through all the abbreviated forms of street suffixes and expanded them to make them all standardized. Capitalization and punctuations were also an issue in the data. To tackle the capitalization, I standardized all street suffixes to an upper case leading letter followed by all lower case letters (STREET → Street or street → Street). Suite numbers were common to street entries in this dataset. For streets inputted in the form of "STREETNAME Street SUITENUMBER", I just removed the suite number. Incorrect spelling was also an issue that was easily fixed if the street suffix was misspelled (example: Avenie → Avenue).

Pending anomalies in the dataset that were noticed, but not cleaned completely, included decimals in streets ("SF 80 PM 4.5"), directional suffixes (Broadway Avenue East), single letter suffixes "Avenue E", intersections "Se Quad Sr 1/Crespi Dr Int"), and incomplete, vague entries such as "Vallejo" and "Wedemeyer".

City Names

Cities were relatively simple to wrangle in comparison to street names. Most of the inconsistencies in data entry came from unstandardized capitalization (all cities were standardized to the form of a leading uppercase letter followed by lower case letters: Alameda).

Some cities identified the state in the form of “Oakland, CA”. Since this data set was taken from a county in California, it was unnecessary to include the CA suffix, so I removed it from all city names. Incorrect spelling occurred in some entries and was fixed in all instances (Emeyville → Emeryville, Okland → Oakland).

Postal codes (or zip codes)

Zip codes came in the five-digit form (92805), nine digit form (92803-1333), state leading code (CA 93093), or in one case, “CA92309”). Everything was standardized to the five-digit form. Some zip codes were not consistent with the rest of the other zip codes (San Francisco has 9 as the first digit in the code; some entries lead with a 1). The zip codes that did not reside in San Francisco county were not modified as I will not be looking at those in my data analysis.

Cleaned Data Exploration (using SQLite)

After cleaning the data and writing it into .csv files, I loaded the newly formed .csv files into a datebase named “SANFRAN.db”. I initially counted the number of nodes and ways, the basic units of this data set, to get a feel for how large the dataset was.

Number of nodes query result:

```
770438
```

Number of ways query result:

```
2666
```

Unsurprisingly, we have less ways than nodes, as nodes are defined as single points that can make up a way. I also looked at the top contributors to the data set.

Top users query results:

```
[('andygol', 1288108),
 ('ediyes', 912011),
 ('Luis36995', 703573),
 ('dannykath', 445309),
 ('RichRico', 404022),
 ('Rub21', 393086),
 ('calfarome', 185572),
 ('oldtopos', 167231),
 ('KindredCoda', 149861),
 ('karitotp', 134929)]
```

I was interested in exploring the amenities of the Bay Area and found that restaurants were the top amenity in the dataset.

Top amenities query results:

```
[('restaurant', 2884),
 ('bench', 1162),
 ('cafe', 970),
 ('place_of_worship', 702),
```

```
 ('post_box', 684),
 ('school', 590),
 ('fast_food', 580),
 ('bicycle_parking', 558),
 ('drinking_water', 507),
 ('toilets', 401)]
```

Narrowing in on the food and drink offerings in the Bay Area, I wanted to see which fast food and beverage establishments had the biggest presence. Coffee shops and Subways lead the way to cater to a Bay Area population that runs on coffee and a health-conscious food diet.

Popular fast foods query results:

```
('Subway', 63)
("McDonald's", 26)
('Taco Bell', 22)
('Burger King', 21)
('Jamba Juice', 18)
('Chipotle', 11)
('Jack in the Box', 9)
('KFC', 8)
("Togo's", 7)
("Wendy's", 6)
```

Popular cafes query results:

```
('Starbucks', 126)
("Peet's Coffee & Tea", 24)
('Starbucks Coffee', 18)
("Peet's Coffee", 10)
('Philz Coffee', 10)
("Peet's Coffee and Tea", 8)
('Quickly', 7)
('Blue Bottle Coffee', 5)
('Jamba Juice', 5)
('Royal Ground Coffee', 5)
```

Conclusions

This data set has proven very useful in gaining a better understanding of the San Francisco Bay Area. However, it has several areas of improvement that needs to be addressed before the dataset can be considered clean for all uses and implementations. As mentioned, the street names data was cleaned for abbreviations and misspellings, but still contain vague, incomplete entries that need to be further investigated (see Street Names section in Data Auditing). Another area to look at would be to clean the values stored in the amenities tags. From the café query above, I observe that Starbucks and Peets come up in different results due to inconsistent spellings.

The bulk of future data cleaning as of now will be in the street names cleaning. An obstacle to ensuring complete standardization of the street data is the fact that there are so many inconsistencies in the data that it is almost impossible to map all the errors and fix them in a dictionary by brute force. Instead, the use of regular expressions and more detailed search and

sort methods, may be better alternatives to improving the street data in a more efficient way. Through exhaustive cleaning of the amenity values and street names, the database queries will return more accurate results that can make it easier to understand the data set as a whole.