Map Area:

Singapore, Singapore

It will be interesting to see what information OSM can tell about my country

Reference Websites:

- Udacity Forums
- http://www.openstreetmap.org/relation/536780#map=11/1.3044/103.8459&layers=G
- https://mapzen.com/data/metro-extracts/metro/singapore/
- https://en.wikipedia.org/wiki/Postal_codes_in_Singapore
-

Wrangling Process:

- Perform pre-cursory check on downloaded XML file
- Noticed that there were issues with street names and postal codes (More information can be found below)
- Cleaned up abbreviated street names
- Cleaned up postal code format

Dataset Overview:

File Size:

- singapore.osm 297 MB
- singapore-sample.osm 11.9MB
- nodes.csv 110 MB
- nodes_tags.csv 3.55 MB
- ways.csv 12.3 MB
- ways_nodes.csv 41.5 MB
- ways_tags.csv 17 MB

Number of unique contributors: 1753

```
cur.execute('''select count(distinct(uid)) from (select uid from nodes union all select uid from ways);''')
```

```
['count(distinct(uid))']
[(1753,)]
```

Number of nodes: 1402943

```
cur.execute('''select count(*) from nodes;''')
```

```
['count(*)']
[(1402943,)]
```

Number of ways: 214195

```
cur.execute('''select count(*) from ways;''')
```

```
['count(*)']
[(214195,)]
```

Total data points: 1617138

Data Issues:

After a pre-cursory check and examination of the OSM file, the following ware noticed:

- Abbreviated street names
- Wrong postal code format
- Wrong city entered
- Addresses from beyond Singapore

Abbreviated Street Names

Such as dr vs drive, st vs street

While glancing through the OSM file, some of the street names were abbreviated possibly by contributors who were used to shortening common words.

For Singapore, street names may end with numbers such as Bedok North Avenue 3 (My address). Also depending on how the contributors submit the data, the below code that was used to convert to full street names may not be fully applicable and I may need to amend the code in the future.

```
#Update street names
def update_name(name, mapping):
    '''
    Args:
        name: street value of attrib['v']
        mapping: dictionary to replace found string/character

    Returns:
        Updated street name'''
    for map in mapping:
        if map in name:
            name = re.sub(r'\b' + map + r'\b\.?', mapping[map], name)
    return name
```

Wrong Postal Code Format

Such as 'Singapore 408564' or '81200https://www.youtube.com/watch?v=m92AUlOMP2k'

(As street name auditing was performed in the project preparation quiz, the code was omitted)

audit_pc.py was used to check on the format of the postal codes. A list of postal codes which did not match the 6-digit format was returned. Function update_pc was then used to update the postal codes to match the format for example, Singapore 408564 to 408564. Exception cases are where the strings did not contain 6 continuous digits. They were retained as part of the dataset.

```
import re
import xml.etree.cElementTree as ET
from collections import defaultdict
import pprint

osm_path = "C:\Users\Leo\Anaconda3\P3 Project OSM\singapore.osm"
post_re = re.compile(r'\d{6}')

def check_pc(osmfile):
    '''Checks if postal codes matches regular expression.
        Adds attributes which do not match to list and returns it.'''
    osm_file = open(osm_path, "r")
    pc_codes = []
    for event, elem in ET.iterparse(osm_file, events=("start",)):
        if elem.tag == "node" or elem.tag == "way":
            for tag in elem.iter("tag"):
                if tag.attrib['k'] == 'addr:postcode' and not test_re.search(tag.attrib['v']):
                    pc_codes.append(tag.attrib['v'])
    osm_file.close()
    return pc_codes

pprint.pprint(check_pc(osm_path))
```

```
#Update postal code based on continous 6-digit format in Singapore
#Searches entire string for match
def update_pc(pc):
    '''
    Args:
        pc: postcode value of attrib['v']

    Returns:
        Updated postcode'''
    if post_re.search(pc):
        pc = post_re.search(pc).group()
    return pc
```

Addresses From Beyond Singapore

A query for cities listed in the dataset showed that most were from Johor Bahru (Malaysia) which is North of Singapore which coincides with the high number of Postal Codes from Johor Bahru.

Possible causes for this could be:

- Extraction boundary limit set by Metro Extracts on non-member accounts which caused Johor Bahru to be included
- Boundary classification errors causing nodes in Malaysia to be set under Singapore
- Tagging errors by contributors when submitting but given the high number, this would be the least likely

```
cur.execute('''select a.key, a.value, count(*) as num
from (select * from nodes_tags union all
select * from ways_tags) a
where a.key = 'city'
group by a.key, a.value
order by num desc
limit 20;''')
```

```
['key', 'value', 'num']
[(u'city', u'Johor Bahru', 13836),
 (u'city', u'Singapore', 11302),
 (u'city', u'Batam', 41),
 (u'city', u'SKUDAI', 35),
 (u'city', u'Masai', 13),
 (u'city', u'Ulu Tiram', 10),
 (u'city', u'Iskandar Puteri', 7),
 (u'city', u'Skudai', 7),
 (u'city', u'singapore', 5),
 (u'city', u'Kulai', 4),
 (u'city', u'Nusajaya', 4),
 (u'city', u'Sembawang', 4),
 (u'city', u'Batam Kota', 2),
 (u'city', u'Johor bahru', 2),
 (u'city', u'#01-05', 1),
 (u'city', u'#01-06', 1),
 (u'city', u'#01-33', 1),
 (u'city', u'#01-38/40/42', 1),
 (u'city', u'#01-44', 1),
 (u'city', u'#01-46', 1)]
```

Interesting Data:

Top amenities

- Top result shows restaurants which fits in perfectly with Singaporeans as we love to eat
- Followed by places of worship, parking lots, cafes, taxi stands, fast food restaurants, atms, banks, toilets and fuel stations
- Delving further on places of worship showed that the top ranking was Muslim however, the results may have be skewed by the data from Johor Bahru. I would expect that the data would be more spread out evenly among the religions

```
cur.execute('''select key, value, count(*) as num
from nodes_tags where key = 'amenity'
group by key, value
order by num desc limit 20;''')

cur.execute('''select nodes_tags.value, count(*) as num
from nodes_tags
join (select distinct (id) from nodes_tags where value = 'place_of_worship') a
on nodes_tags.id = a.id
where nodes_tags.key = 'religion'
group by nodes_tags.value
order by num desc
limit 20;''')
```

```
['key', 'value', 'num']
[(u'amenity', u'restaurant', 1414),
 (u'amenity', u'place_of_worship', 579),
 (u'amenity', u'parking', 527),
 (u'amenity', u'cafe', 366),
 (u'amenity', u'taxi', 337),
 (u'amenity', u'fast_food', 336),
 (u'amenity', u'atm', 218),
 (u'amenity', u'bank', 203),
 (u'amenity', u'toilets', 197),
 (u'amenity', u'fuel', 196),
 (u'amenity', u'school', 162),
 (u'amenity', u'shelter', 150),
 (u'amenity', u'police', 143),
 (u'amenity', u'food_court', 116),
 (u'amenity', u'bar', 103),
 (u'amenity', u'doctors', 93),
 (u'amenity', u'post_office', 78),
 (u'amenity', u'bench', 76),
 (u'amenity', u'parking_entrance', 75),
 (u'amenity', u'bus_station', 73)]
```

```
['value', 'num']
[(u'muslim', 410),
 (u'christian', 87),
 (u'buddhist', 20),
 (u'hindu', 4),
 (u'jewish', 3),
 (u'taoist', 2),
 (u'sikh', 1)]
```

Contributors Data

Top contributors are as below:

JaLooNz: 366226 approximately 22.65% of the data While second place berjaya contributed approximately 7.29% of the data

With the huge disparity between the contributors, I can't help but suspect that automated applications/programs were used to submit the data besides manual means.

OSM does not state which collection and submission methods are allowed but if they were to introduce reward mechanisms, automation may have to be disallowed in-lieu of fairness.

```
cur.execute('''select user, count(*) as num
from (select user from nodes union all select user from ways)
group by user
order by num desc
limit 20;''')
```

```
['user', 'num']
[(u'JaLooNz', 366226),
 (u'berjaya', 117874),
 (u'rene78', 80294),
 (u'cboothroyd', 74481),
 (u'Luis36995', 41876),
 (u'ridixcr', 40352),
 (u'kingrollo', 39691),
 (u'lmum', 38867),
 (u'Sihabul Milah', 37310),
 (u'calfarome', 34677),
 (u'jaredc', 26493),
 (u'nikhilprabhakar', 25557),
 (u'Jothirnadh', 24139),
 (u'manings', 22892),
 (u'yurasi', 22361),
 (u'matx17', 21466),
 (u'zomgvivian', 20110),
 (u'poornibadrinath', 19116),
 (u'fusionstream', 18010),
 (u'singastreet', 17642)]
```

Conclusion

After exploring the data, it's clear that the information for Singapore incomplete and with errors (Barring the mix of data from Johor Bahru).

The concept of users contributing to the project is intriguing and allows the data to grow much faster than if done only by company staff however, possibilities of errors or format issues will always exist. I tried to filter and correct the data as much as possible but there were still some error entries left for streets and cities.

A possible suggestion could be to provide users with a format list in the application which they can choose from or, verify the format of the submissions before accepting.

Benefits of using a formatted list/verification before submission:

- Users will not have to worry about submitting wrong data or, remembering the format
- People working or using the data will not have to worry about consistency or errors

Anticipated Problems:

- Standardizing of postal code formats which differs country to country, localized help will be necessary
- Depending on the number of concurrent submissions, the performance of the application/backend may be affected