# OpenStreetMap Data Project

# Map Area

Hong Kong, China

- https://www.openstreetmap.org/node/2833125787
- https://mapzen.com/data/metro-extracts/

This is an amazing city where I am living with my family. I have been staying here for more than ten years, so I'm interested to see what database querying reveals, and I would like to take this opportunity to contribute to its improvement on OpenStreetMap.org.

# Problems Encountered in the Map

After initially downloading a small sample size file and running it against a provisional data.py file, I noticed two main problems with the data, which I will discuss in the following order:

- Overabbreviated street names *("d'Aguilar St")*
- *"Incorrect" postal codes (No zip code for Hong Kong, but a small count of numbers begin with "51" indicate areas outside this region.)*

## Overabbreviated Street Names

Once the data was imported to SQL, some basic querying revealed street name abbreviations and postal code incorrect. To deal with the over abbreviations, I tried to iterate over each word in an address, correcting them to their respective mappings in audit.py using the following function:

```
def update_name(name, mapping):
    words = name.split()
    for w in range(len(words)):
        if words[w] in mapping:
           #print words[w]
           words[w] = mapping[words[w]]
           name = " ".join(words)
    return name
```

This updated all substrings in problematic address strings, such that: *“1 Stewart Rd”* becomes: *“1 Stewart Road”; “d'Aguilar St”* becomes: *“d’Aguilar Street”.*

## Postal Codes

Because Hong Kong is a Special Administrative Region, no postal code is available for this place. However, SQL query still returned some inputs as “incorrect postal code”:

```
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags
      UNION ALL
      SELECT * FROM ways_tags) tags
WHERE tags.key='postcode'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 10;
```

Here are the top ten results, beginning with the highest count:

```
value,count
518000,11
510623,7
510180,6
519000,6
510290,5
518048,5
519087,5
518038,4
518067,4
"QBML 2”,4
```

From the postal codes beginning with the number of 510, I suspected that the mistakes might come from miscounting neighboring cities in mainland China as part of Hong Kong, such as Shenzhen and Zhuhai. To explore this, I performed another aggregation to verify a certain suspicion...

## Sort cities by count, descending

```
sqlite> SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags UNION ALL
      SELECT * FROM ways_tags) tags
WHERE tags.key LIKE '%city'
GROUP BY tags.value
```

```
ORDER BY count DESC
LIMIT 10;
```

And, the results, edited for readability:

```
"香港 Hong Kong",363
"屯門 Tuen Mun",128
"荃灣 Tsuen Wan",75
"Sai Kung",53
"广东省深圳市",50
"Hong Kong",49
"Ta Kwu Ling",41
"深圳",40
Zhuhai,37
Shenzhen,31
```

These results confirmed my suspicion that this metro extract would perhaps also include surrounding cities near Hong Kong.

# Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the SQL queries used to gather them, and some additional ideas about the data in context.

## File sizes

```
524381500 May 13 12:25 hong-kong_china.osm
433334272 May 16 15:17 hongkong.db
205059894 May 16 12:44 nodes.csv
6170640 May 16 12:44 nodes_tags.csv
15059326 May 16 12:54 ways.csv
70380086 May 16 12:54 ways_nodes.csv
23481938 May 16 12:54 ways_tags.csv
```

## Number of nodes

```
sqlite> SELECT COUNT(*) FROM nodes;
```

2506176

## Number of ways

```
sqlite> SELECT COUNT(*) FROM ways;
```

256776

## Number of unique users

```
sqlite> SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
```

1645

## Top 10 contributing users

```
sqlite> SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
ORDER BY num DESC
LIMIT 10;
```

```
hlaw,501492
MarsmanRom,234505
Popolon,160601
sn0wblind,121198
fsxy,98243
katpatuka,96604
KX675,79276
fdulezi,79028
rainy3519446,58106
tomlee721,52607
```

## Number of users only having 1 post

```
sqlite> SELECT COUNT(*)
FROM
    (SELECT e.user, COUNT(*) as num
     FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
     GROUP BY e.user
     HAVING num=1)  u;
```

295

## Top 10 appearing amenities

```
sqlite> SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10;

restaurant,769
shelter,660
bus_station,567
bank,561
parking,557
fast_food,329
fuel,238
taxi,230
place_of_worship,221
```

## Most popular cuisines

```
sqlite> SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
    JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE
value='restaurant') i
    ON nodes_tags.id=i.id
WHERE nodes_tags.key='cuisine'
GROUP BY nodes_tags.value
ORDER BY num DESC
LIMIT 10;

chinese,137
japanese,21
indian,15
italian,9
pizza,9
regional,9
international,8
burger,7
thai,7
french,6
```

# Additional Ideas

## Contributor statistics

Similar to other regions in the OpenStreetMap project, the contributions of users seems skewed, as expected. Here are some user percentage statistics:

- Top user contribution percentage (“hlaw”) 18.15%
- Combined Top 10 users contribution 53.63%

## Amenity statistics

Restaurants was the top appearing of amenities in Hong Kong. If combining restaurants and fast_food, there were total 4132 counts, 26.57% of the top 10 appearing amenities. In addition, bank ranked the 4th in the amenities, indicating Hong Kong as an international cuisine and financial center.

# Conclusion

After this review of the data I noticed that the Hong Kong area is incomplete, though I believe it has been well cleaned for the purposes of this exercise. OpenStreetMap project is a very good start for data wrangling beginners like me. I have been making progress on learning basic codings for data wrangling, as well as using SQL to review the database.