

# OpenStreetMap Data Case Study

## Map Area

### Wuhan,Hubei,China

- <https://www.openstreetmap.org/export#map=11/30.5962/114.2993>
- Used QGIS to download .OSM document.

**This map is my hometown,I was born and studied in there.There are many delicious food restaurant in Wuhan,and I 'd like an opportunity to contribute to its improvement on OpenStreetMap for my hometown.**

## Problems encountered in the map

**After initially downloading a small sample size of the WUHAN area and running sample.osm.Two main problems as below:**

- Overabbreviated street names("Sanjiahu Rd")
- Including Chinese and spell name ("车站村 chezhancun")

## Corrected the street name

### corrected the street name ("Rd -> Roade")

```
python def update_street_name(name, mapping): for ma in mapping.keys(): if name.endswith(ma): name=name.replace(ma,mapping[ma]) return name
```

## explored the zipcode

```
python import sqlite3 con = sqlite3.connect('map.db') cur = con.cursor() cur.execute("select tags.value, count(*) as num \ from (select * from nodes_tags union all select * from ways_tags) as tags \ where tags.key = 'postcode' \ group by tags.value \ order by num desc") cur.fetchall()
```

```
[(u'430079', 49),  
(u'430074', 9),  
(u'430070', 8),  
(u'430056', 5),  
(u'430062', 5),  
(u'430000', 3),  
(u'430064', 3),  
(u'430100', 3),  
(u'430072', 2),  
(u'430101', 2),  
(u'', 1),  
(u'430022', 1),  
(u'430065', 1),  
(u'430071', 1),  
(u'430073', 1),  
(u'430080', 1)]
```

**the Zipcode looks good**

## Data overview and additional ideas

### File sizes

- wuhan\_china.osm ..... 62.3 MB
- map.db ..... 33.6 MB
- nodes.csv ..... 24.8 MB
- nodes\_tags.csv ..... 0.48 MB
- ways.csv ..... 1.96 MB
- ways\_tags.csv ..... 2.4 MB
- ways\_nodes.cv ..... 9 MB

### Number of nodes

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

```
[(314952,)]
```

### Number of ways

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

```
[(34267,)]
```

## Number of unique users

```
python cur.execute('select count(distinct(e.uid)) from (select uid from ways union all select uid from nodes) as e') cur.fetchall()
```

```
[(515,)]
```

## Top 10 contributing users

```
python cur.execute('select e.user, count(*) as num \ from (select user from ways union all select user from nodes) as e \ group by e.user \ order by num desc \ limit 10') cur.fetchall()
```

```
[(u'GeoSUN', 112201),  
(u'Soub', 48069),  
(u'jamesks', 24414),  
(u'Gao xioix', 17901),  
(u'katpatuka', 17298),  
(u'dword1511', 13558),  
(u'samsung galaxy s6', 10603),  
(u'flierfy', 5715),  
(u'hanchao', 5289),  
(u'keepcalmandmapon', 5122)]
```

## Number of users appearing only once(having 1 post)

```
python cur.execute('select count(*) \ from (select e.user, count(*) as num \ from (select user from nodes union all select user from ways) as e \ group by e.user \ having num = 1) as u') cur.fetchall()
```

```
[(92,)]
```

## Additional ideas

```
python cur.execute('select count(*) as num \ from (select user from nodes union all select user from ways) as e') cur.fetchall()
```

```
[(349219,)]
```

**Here are some user percentage statistics:**

- Top user contribution percentage("GeoSUN")32.13%
- Combined top users'contribution("GeoSUN" and "Soub")45.89%
- Combined Top 10 users contribution 74.5%

## Additional Data Exploration

### Numbers of store

```
python cur.execute("select e.value, count(*) as num \ from (select * from nodes_tags union all select * from ways_tags) as e \ where e.key = 'shop' \ group by e.value \ order by num desc") cur.fetchall()
```

```
[(u'supermarket', 83),  
(u'convenience', 26),  
(u'mall', 18),  
(u'books', 17),  
(u'bakery', 10),  
(u'garden_centre', 9),  
(u'yes', 9),  
(u'car', 7),  
(u'bicycle', 6),  
(u'clothes', 6),  
(u'department_store', 5),  
(u'copyshop', 4),  
(u'wine', 4),  
(u'computer', 3),  
(u'florist', 3),  
(u'optician', 3),  
(u'greengrocer', 2),  
(u'hairstylist', 2),  
(u'laundry', 2),  
(u'mobile_phone', 2),  
(u'seafood', 2),  
(u'beverages', 1),  
(u'bookmaker', 1),  
(u'butcher', 1),  
(u'furniture', 1),  
(u'gift', 1),  
(u'hardware', 1),  
(u'jewelry', 1),  
(u'photo', 1),  
(u'photo_studio', 1),  
(u'sports', 1),  
(u'travel_agency', 1)]
```

## Numbers of leisure space

```
python cur.execute("select e.value, count(*) as num \ from (select * from  
nodes_tags union all select * from ways_tags) as e \ where e.key = 'leisure'  
\ group by e.value \ order by num desc") cur.fetchall()
```

```
[(u'pitch', 281),  
(u'park', 232),  
(u'track', 77),  
(u'common', 26),  
(u'sports_centre', 24),  
(u'playground', 19),  
(u'stadium', 15),  
(u'recreation_ground', 12),  
(u'swimming_pool', 7),  
(u'garden', 4),  
(u'golf_course', 4),  
(u'sauna', 1),  
(u'water_park', 1),  
(u'yes', 1)]
```

## The most popular food

```
python cur.execute("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")  
cur.fetchall()
```

```
[(u'chinese', 14),  
(u'asian', 2),  
(u'barbecue;chinese', 1),  
(u'burger', 1),  
(u'chinese;american', 1),  
(u'chinese;oriental', 1)]
```

## Conclusion

**After this review of Wuhan map data it was shows that area isn't complete,even I though I have been cleaned the map data of this project.It was interest me to notice many kind of food,leisure space and store which correspond with Wuhan people life style and customs.**

**Because of the city of Wuhan is China city, the city of street is combine with Chinese and Chinese spell name that will error for same pronounce but actually different street. So improvement and accurate translation the name of street will help us to better understanding the map data.**