OpenStreetMap 数据集研究

地图区域

中国 / 湖北 / 武汉（https://mapzen.com/data/metro-extracts/metro/wuhan_china/）
(https://mapzen.com/data/metro-extracts/metro/wuhan_china/）)

所遇问题

因为样本量过大， 故先选择了一个小样本sample.osm进行预分析。可发现以下几个问题：

- 街道名称的缩写，例如：Sanjiaohu Rd
- 街道名称中包含中文和拼音，例如：车站村 chezhancun

纠正街道名

为了使街道名称后缀显示一致性，通过以下函数可以进行修正：

In [ ]:

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

更新后的街道名会更加整齐，例如Sanjiaohu Rd会更新为Sanjiaohu Road

postcode

In [1]:

```
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()
```

Out[1]:

```
[(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)]
```

从列举出的邮编可以看出邮编的格式基本没有问题，但是并不全面，虽然是武汉市的地图描述，但是很多地方的邮编并没有，由此可以发现地图只是武汉市的部分地区。

数据概述

文件大小

- 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

节点数量

In [3]:

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

Out[3]:

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

途径数量

In [5]:

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

Out[5]:

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

用户数量

In [13]:

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

Out[13]:

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

前十大贡献用户

In [4]:

```
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()
```

Out[4]:

```
[(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)]
```

出现一次的用户数量

In [4]:

```
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()
```

Out[4]:

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

其他想法

In [6]:

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

Out[6]:

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

计算可以发现:

- 用户贡献率最高(GeoSUN)达到32.13%
- 前两名用户贡献率(GeoSUN和Soub)为45.89%
- 前十大用户贡献率为74.5%

附加数据探索

店铺类型及数量

In [5]:

```
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()
```

Out[5]:

```
[(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'hairdresser', 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)]
```

休闲类型及数量

In [11]:

```
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()
```

Out[11]:

```
[(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)]
```

最受欢迎的菜系

In [7]:

```
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()
```

Out[7]:

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

结论

经过对数据的审查可以发现，数据不是完全干净，所以通过编程的方式清理街道。虽然武汉市的地图数据不够完整，但是从分析到的菜系，娱乐方式及店铺类型足可以发现其大致符合武汉的生活方式及特色。

由于分析的是中国地区的城市，城市地区名及相关街道信息都显示的是拼音或者是中文和拼音一起的格式，但是这样可能会造成同发音不同地址的情况，所以为避免出现错误，应该结合对当地的认识来清理街道数据，将对应的拼音翻译为中文名称。这样虽然要求分析者要对当地有一定的认识并且可能或者仍旧会出现极少的翻译错误，但是这种改进能使我们更好的了解数据。

In [ ]: