

OpenStreetMap 项目

地区 - 上海

中国，上海市

- https://mapzen.com/data/metro-extracts/metro/shanghai_china/

- 位置简介：

上海市，简称沪，别称申，[中华人民共和国直辖市](#)，中国大陆经济、金融、贸易中心城市，国家财政收入支柱城市。其[港口](#)为世界最大的[集装箱港](#)，主要产业包括[商贸流通](#)、[金融](#)、[信息](#)、[制造](#)。上海位于中国东部弧形海岸线的正中间，地处[长江三角洲](#)最东部，东向[东海](#)，隔海与[日本九州岛](#)相望，南濒[杭州湾](#)，西部与[江苏](#)、[浙江](#)两省相接，最北部为处于[长江](#)入海口中的[崇明岛](#)。上海是中国人口最多的城市之一，常住人口2419.70万，其中[本地户籍人口](#)为1439.50万人，约占总人口数的59%[\[参 1\]](#)。[江南的吴越文化](#)传统与各地移民带入的多样文化融合，逐渐形成了特有的[海派文化](#)。 — 来自 [维基百科](#)

- 选择原因：我现在在上海工作，对上海比较熟悉。

数据清洗

验证邮政编码

我使用以下代码对 Node 中的邮政编码进行验证：

```
rex_postcode = re.compile(r'[1-9]\d{5}(?!\\d)')
with open("shanghai.osm", 'rb') as f:
    soup = BeautifulSoup(f.read(), "lxml")
    postcodes = soup.find_all('tag', k='addr:postcode')
    for post in postcodes:
        if rex_postcode.fullmatch(post['v']) == None:
            print(post['v'])

print("Done!")
```

发现 6 条异常值：

```
2000080
201315 上海
201315 上海
2000080
20032
20032
```

验证经纬度

经纬度数据大部分为小数点后面保留 7 位，有少部分小数点后面只有 1~6 位，可能会影响数据的准确性。我使用以下代码来判断异常经纬度数据。

```
with open("shanghai.osm", 'rb') as f:
    soup = BeautifulSoup(f.read(), "lxml")
    nodes = soup.find_all('node')

    print("lat")
    for node in nodes:
        if len(node['lat']) < 6:
            print(node['lat'])

    print("lon")
    for node in nodes:
        if len(node['lon']) < 7:
            print(node['lon'])

    print("Done!")
```

小数点后少于 3 位的数据如下：

```
lat:
31.22
31.14
31.14
31.14

lon:
121.37
121.34
121.66
121.65
```

数据概览

Shanghai.osm 137.9MB
sample.osm 1.4MB

文件	大小	行数
Node.db	46.5MB	625646
Way.db	5.5MB	90315
Relation.db	96KB	1046

行数统计

由于我使用了 peewee 这个 python 库，所以没有直接使用原生 SQL 语句，在将数据都插入到 MySQL 以后，使用以下代码可以统计 node, way, relation 的行数：

```
nodes = SelectQuery(Node).select()
ways = SelectQuery(Way).select()
relations = SelectQuery(Relation).select()

print("nodes count = ", nodes.count())
print("ways count = ", ways.count())
print("relations count = ", relations.count())
```

独立用户统计

我统计了 Node 表的独立用户数，代码如下，和 SQL 语句 `GROUP BY` 的作用是一样的：

```
nodes = Node.select(Node.user,
    fn.Count(Node.user).alias("user_count")).group_by(Node.user)
```

独立用户数为：1163

排名前 10 的用户以及他们贡献数据的数量

以下数据从 Node 表中统计得出，代码如下：

```
query = Node.select(Node.user,
    fn.Count(Node.user).alias("user_count")).group_by(Node.user)
query = query.order_by(SQL('user_count DESC'))

for node in query[0:10]:
    print(node.user_count)
    print(node.user)
```

对应的数据为：

序号	user	count
1	zzcolin	66286
2	Xylem	62368
3	Stenive	49075
4	aighes	43568
5	yangfl	38028
6	HWST	36256
7	alberth2	19116
8	u_kubota	18795
9	z_i_g_o	18263
10	duxxa	13172

排名前 100 的用户数据总数

查询代码如下：

```
query = Node.select(Node.user,
fn.Count(Node.user).alias("user_count")).group_by(Node.user)
query = query.order_by(SQL('user_count DESC'))

num = 0
for node in query[0:100]:
    print(node.user_count)
    print(node.user)
    num += node.user_count

print('num = ', num)
print('per = ', num/625646)
```

数量总数为：582789 占比：93.15%

仅有一条数据的用户数

查询代码如下：

```
query = Node.select(Node.user,
fn.Count(Node.user).alias("user_count")).group_by(Node.user).having(fn.Count
(Node.user) == 1)
```

结果为：288

数据量小于 100 的用户数

查询代码如下：

```
query = Node.select(Node.user,
    fn.Count(Node.user).alias("user_count")).group_by(Node.user).having(fn.Count(
    Node.user) < 100)
```

结果为：934

Node 中 Tag 为 fast_food 的数量

fast_food 数量的查询代码如下：

```
with open("shanghai.osm", 'rb') as f:
    soup = BeautifulSoup(f.read(), "lxml")
    fastfood = soup.find_all('tag', k='amenity', v='fast_food')

    sum = 0
    for food in fastfood:
        tags = food.find_next_siblings()
        for tag in tags:
            if tag['k'] == 'name:en':
                sum += 1
                print(tag['v'])

    print("Done!")
    print("sum = ", sum)
```

结果为：75，大多为 KFC 和 McDonald's。

改进建议

原始数据中，有 93% 的数据由数据量总数排名前 100 的用户统计而来，有多达 288 个用户提供的数据仅有一条。同时这些数据中有部分数据的经纬度数据并不精确，所以这部分数据的准确性有待进一步验证。另外，邮政编码中，也有部分数据不规范，有的只有 5 位，有的超过了 6 位，甚至包含其他无效字符串，这部分数据也需要进行修复。Node 中 fast_food 的数量比较少，仅有 75 个，但是就实际情况来看，75 个未免也太少了，猜测应该是数据统计不全面或者缺失造成的。

参考资料

- <https://github.com/coleifer/peewee>
- <http://docs.peewee-orm.com/>
- <https://docs.python.org/2/library/decimal.html>
- <http://stackoverflow.com/questions/14010551/how-to-convert-between-bytes-and-strings-in-python-3>
- <http://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000/00>

- http://www.ruanyifeng.com/blog/2007/10/ascii_unicode_and_utf-8.html