数据分析 - 开放街道地图(Open Street Map)

地区

中国上海
https://www.openstreetmap.org/relation/913067
上海是我的家乡，因此我对于数据探索的结果很感兴趣，同时也希望为上海的数据做出自己的贡献。我使用 Python 进行数据清洗，使用 SQLite 进行数据保存、探索和分析。

在地图中遇到的问题

- 某些英文地址中，存在使用简写的情况。比如：“Chengnan Rd.”、“Haigang Ave.”。
- 某些英文地址中，存在使用拼音 Lu/lu 表示路的情况。比如：“Pingjiang Lu”
- 某些地址中既包含中文地址，又包含英文地址，比如：<tag k="addr:street" v="巨鹿路/Julu Rd"/> <tag k="addr:street" v="柏庐南路 999 号（Bailunan Rd.）"/>。
- 某些地址中包含换行符，源自 xml 中的“
 ”实体，如<tag k="addr:street" v="Shennan
 Road"/>解析出来“
”就会变成换行。
- 地址的邮政编码和上海市的邮政编码不一致。例如：<tag k="addr:postcode" v="20032"/> <tag k="addr:street" v="中山南二路"/>。
- 某些邮政编码后面加上了“上海”两个字。例如：<tag k="addr:postcode" v="201315 上海"/>。

数据清洗的方案及问题预期

英文地址中使用简写的情况

使用全称，替换问题地址中的简称，比如将“Chengnan Rd.”，转换成“Chengnan Road”，将“Haigang Ave.”转换成“Haigang Avenue”。

改进的益处和问题

将数据导入到数据库中后，是通过地址关键字进行查询的。将关键字进行统一后，便于今后的查询。
但是英文简写多种多样，很难全部都列出来。所以清洗过的数据中，依然可能存在英文地址中有简写的情况，没有被有效的替换掉。因为使用了常用的英文简写替换列表，所以清理后能够保证大多数数据的一致性。

英文地址中使用拼音 Lu/lu 表示路的情况

将 Lu/lu 替换成 Road 的标准形式。由于路名中的拼音也可能包含 Lu/lu，为了避免错误替换，仅替换字符串结尾的 Lu/lu。

改进的益处和问题

把 Lu 改成 Road 后，更加符合英文的使用习惯，也能够更加保证数据的一致性。
为了防止将路名中的拼音也替换掉，清理只能替换尾部的 Lu/lu，这样其他位置的 Lu 仍然可能存在，清理的不够完全。好在，大部分的 Lu 应该都在尾部，因此这样的方式能够保证大多数数据的一致性。

对于地址中包含换行符的情况

使用正则表达式将地址中的换行和多余空格替换成一个空格字符串，使格式标准化

同时包含中文地址和英文地址的情况

使用正则表达式，提取中文地址，删除英文地址

改进的益处和问题

地址格式更加一致，方便以后的搜索和查询
但是中文和英文地址混合的格式多种多样，单凭一个简单的正则表达式恐怕无法将所有的情况都包括在内，因此某些数据仍然存在混合的情况。探索后发现，这种方式能够处理大部分的数据，只有少量数据因为复杂的中英文混合无法处理，所以这种方式可以接受

邮政编码后面加上了“上海”两个字的情况

只保留邮政编码中的数字部分，过滤掉非数字字符。

地址的邮政编码和上海市的邮政编码不一致的情况

上海地区的邮政编码都是六位数的，而且以 20 开头，数据探索过程中发现有些邮政编码多于或少于六位，有些不以 20 开头。所以，将不以 20 开头和非六位数的邮政编码全都过滤掉。

改进的益处和问题

能够过滤掉非上海的和错误邮政编码，但是由于某些标签不包含邮政编码数据，因此仍然会存在一些上海以外的地区数据。此外，对于用户误操作造成的错误邮编，比如多输入一位或少输入一位，这种方式也会造成有效数据丢失。在探索时发现，错误的邮政编码只占了很小的一部分，所以这种处理方式不会造成很大影响。

数据概览

所有数据清理以后，将其存为 5 个 csv 文件，然后导入 SQLite 数据库中

文件大小

shanghai_china.osm.....361MB
openstreetmap.db.....194MB
nodes.csv.....135MB
nodes_tags.csv.....5MB
ways.csv.....13MB
ways_tags.csv.....17MB
ways_nodes.csv.....49MB

node 的数量

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

1704748

way 的数量

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

227576

用户数量

```
SELECT
  COUNT(DISTINCT(t.uid))
FROM
  (
    SELECT
      uid
    FROM
      nodes
    UNION ALL
      SELECT
        uid
      FROM
        ways
  ) t;
```

1800

10 位贡献最多的用户

```sql
SELECT
  t.user,
  COUNT(*) AS num
FROM
  (
    SELECT
      user
    FROM
      nodes
    UNION ALL
      SELECT
        user
      FROM
        ways
  ) t
GROUP BY
  t.user
ORDER BY
  num DESC
LIMIT 10;
```

Austin Zhu	229166
xiaotu	159867
aighes	136997
zzcolin	82661
Koalberry	73845
yangfl	73783
Xylem	70293
duxxa	67545
Peng-Chung	60970
alberth2	45465

只贡献过一次的用户

```
SELECT
  COUNT(*)
FROM
  (
    SELECT
      t.user,
      COUNT(*) AS num
    FROM
      (
        SELECT
          user
        FROM
          nodes
        UNION ALL
          SELECT
            user
          FROM
            ways
      ) t
    GROUP BY
      t.user
    HAVING
      num = 1
  ) t;
```

425

额外的想法

贡献者信息统计

用户的贡献呈现出非常严重的倾斜。少量的用户贡献了绝大多数的地图信息。这里，有一些关于用户贡献的统计信息：

- 贡献最多的用户（Austin Zhu），占全部贡献的 11.86%
- 贡献最多的前 10 个用户（占全部用户的 0.56%），合计占全部贡献的 51.78%
- 贡献最多的前 100 个用户（占全部用户的 5.6%），他们的贡献占全部贡献的 91.79%
- 贡献最少的前 1500 个用户（占全部用户的 83.3%），他们的贡献只占全部贡献的 1.49%

我认为，对于贡献度高的用户来说，数据的一致性和可靠性有一定的保证；同时，Open Street 可以采取一些奖励措施(比如增加用户等级、授予徽章)，来促进更多的人来为此项目做出贡献，毕竟群众的力量是巨大的。

额外建议和预期的问题
做数据分析时可以考虑删除贡献较少的 1500 个用户提供的数据，这样做的好处是能够使数据更加一致和可靠。可能遇到的问题是，删除数据会造成数据的不完整，但考虑到这些用户的数据只占了 1.5%，所以只会删除很少的数据，利大于弊。

道路数据统计

```
SELECT
  count(DISTINCT(id))
FROM
  (
    SELECT
      *
    FROM
      nodes_tags
    UNION ALL
      SELECT
        *
      FROM
        ways_tags
  ) t
WHERE
  KEY = 'street'
```

4197

```
SELECT
  count(DISTINCT(id))
FROM
  (
    SELECT
      *
    FROM
      nodes_tags
    UNION ALL
      SELECT
        *
      FROM
        ways_tags
  ) t
WHERE
  KEY = 'postcode'
```

717

数据集中包含了 4197 条道路数据，但是邮编信息只有 717 条，仅占 17%，可见大部分道路没有提供相应的邮编数据。我认为邮编是相对比较重要的信息，可以考虑在用户输入道路的时候将邮编作为必选部分，提高数据的完整性。

使用 SQLite 进行额外的数据探索

出现次数前 10 的便利设施

```sql
SELECT
  value,
  COUNT(*) AS num
FROM
  nodes_tags
WHERE
  KEY = 'amenity'
GROUP BY
  value
ORDER BY
  num DESC
LIMIT 10;
```

restaurant	1015
bicycle_rental	517
bank	404
cafe	341
toilets	286
fast_food	270
parking	150
bar	131
fuel	128
atm	114

出现最多的餐厅

```sql
SELECT
  value,
  COUNT(*) AS num
FROM
  nodes_tags
JOIN (
  SELECT DISTINCT
    (id)
  FROM
    nodes_tags
  WHERE
    value = 'restaurant'
) t ON nodes_tags.id = t.id
WHERE
  nodes_tags."key" = 'name'
GROUP BY
  nodes_tags.value
ORDER BY
  num DESC
LIMIT 5;
```

肯德基	5
McDonald's	4
Pizza Hut	4
兰州拉面	3

吉祥馄饨	3

出现最多的快餐店

```
SELECT
  value,
  count(*) AS num
FROM
  nodes_tags
JOIN (
  SELECT DISTINCT
    (id)
  FROM
    nodes_tags
  WHERE
    value = 'fast_food'
) t ON nodes_tags.id = t.id
WHERE
  nodes_tags."key" = 'name'
GROUP BY
  nodes_tags.value
ORDER BY
  num DESC
LIMIT 5;
```

KFC	69
麦当劳	27
Burger King	4
Pizza Hut	3
Subway	3

出现最多的银行

```
SELECT
  value,
  count(*) AS num
FROM
  nodes_tags
JOIN (
  SELECT DISTINCT
    (id)
  FROM
    nodes_tags
  WHERE
    value = 'bank'
) t ON nodes_tags.id = t.id
WHERE
  nodes_tags."key" = 'name'
GROUP BY
  nodes_tags.value
ORDER BY
  num DESC
LIMIT 5;
```

招商银行	43
中国银行	25
建设银行	23
上海银行	20
中国农业银行 ABC	10

结论

经过本次清洗，数据在一致性和完整性上有了一定的提升。虽然清洗过的数据中，依然可能存在数据一致性和完整性的问题，但是我相信这次数据清洗已经很好地达到了本次练习的目的。

在本次清洗过程中，我注意到，Open Street Map 可以从 GPS 获取数据，还能让所有人都参与编辑地图、提供数据。这种形式能够增加数据的来源和多样性，也从一定程度上保持了准确性，但是也带来了数据一致性和完整性上的问题。随着对数据清洗，可以大大改善数据的质量。