整理 OpenStreetMap 数据

采用的地图

- https://www.openstreetmap.org/relation/165473
- https://mapzen.com/data/metro-extracts/metro/las-vegas_nevada/

拉斯维加斯是世界著名赌城，对该城市的地图数据比较好奇，所以选择此城市。

地图中存在的问题

运行 data.py 文件时发现以下问题：

- 邮政编码的格式不统一

```
<tag k="addr:postcode" v="89119"/>
<tag k="addr:postcode" v="NV 89149"/>
<tag k="addr:postcode" v="89108-7049"/>
```

数据清洗

用 audit.py 文件对数据进行清洗整理，把类似于 'NV 89031' 和 '89109-1907' 格式的邮政编码全部转换为 '89119' 这样的格式。
修正邮政编码的部分代码如下：

```
def update_name(name):
    m = re.search(postcode_NV_re, name)
    n = re.search(postcode_re, name)
    if m:
        name = (re.sub(postcode_NV_re, '', name)).strip()
    elif n:
        name = name.split('-')[0]
    return name
```

修正后的格式如下:
89108-7049 => 89108

NV 89149 => 89149

- 修正数据格式过程中遇到的问题:
  第一次输出修正后的数据时，只输出了 '89108-7049 => 89108',为了验证是否是 update_name 函数引起的错误，
  在执行 update_name 函数前把所有的邮政编码都输出了，结果还是缺少 ‘NV 89149’,最后通过在 sample.osm 文件中查找 ‘NV 89149’，发现其在顶级标签 relation 的子标签 tag 中，
  检查发现 audit_project.py 文件限定了查找标签为 'node' 和 'way'，删掉之后便得到了正确答案。

把修正后的数据写入到 CSV 文件中

- 此过程中遇到的问题:
  完成此步骤后发现 CSV 文件中的错误邮政编码数据格式并没有修正过来，检查后发现，在把数据写入到 CSV 文件前没有更新数据，添加部分代码后达到了预期结果。
- 添加的主要代码如下:

```
if child.attrib['k'] == 'addr:postcode':
    tags_attrib['value'] = update_name(child.attrib['v'])
else:
    tags_attrib['value'] = child.attrib['v']
```

把 CSV 文件导入到数据库中

查询数据库中的邮编格式是否正确

```
SELECT key, value FROM nodes_tags
WHERE key == 'postcode' AND value == '89108'
UNION
SELECT key, value FROM relation_tags
WHERE key == 'postcode' AND value == '89149'
```

```
postcode  89108
postcode  89149
```

数据概述

文件大小

```
las-vegas_nevada.osm.......... 219.04 MB
sample.osm.................... 9.12 MB
data_wrangling_schema.db...... 496.64 MB
nodes.csv..................... 84.73 MB
nodes_tags.csv................ 2.28 MB
ways.csv...................... 6.54 MB
ways_nodes.csv................ 29.08 MB
ways_tags.csv................. 14.19 MB
relation_tags.csv............. 0.10 MB
relations.csv................. 0.03 MB
```

唯一用户的数量

```
SELECT COUNT(DISTINCT(e.uid)) FROM
(SELECT uid FROM nodes UNION
SELECT uid FROM ways UNION
SELECT uid FROM relations) e;
```

```
1117
```

nodes 数量

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

```
1063059
```

ways 数量

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

```
115082
```

警察局的数量

```
SELECT COUNT(*) FROM nodes_tags
WHERE value == 'police';
```

```
16
```

中国餐馆的数量

```
SELECT COUNT(*) FROM nodes_tags
WHERE key == 'cuisine' and value == 'chinese';
```

```
21
```

店面最多的10中咖啡店

```
SELECT value , COUNT(*)
FROM nodes_tags
JOIN (SELECT DISTINCT id FROM nodes_tags WHERE value="cafe") nodes_ids
ON nodes_tags.id=nodes_ids.id
WHERE key="name"
GROUP BY value
ORDER BY COUNT(*) DESC
LIMIT 10;
```

```
Starbucks,37
"Dunkin' Donuts",3
"Coffee Bean & Tea Leaf",2
"Starbucks Coffee",2
"Baga Hookah",1
"Brooklyn Bagels",1
```

```
"Cafe Bellagio",1
"Cafe Belle Madeleine",1
"Cafe Pan",1
"Café Berlin",1
```

星巴克的分店最多，同时发现一个问题，星巴克的名称出现了 'Starbucks' 和 'Starbucks Coffee'两种写法。

改进数据建议

在分析过程中发现，更多的数据其本身并没有错误，更多的是数据的格式不统一，比如本次分析中发现的邮编格式，星巴克的店名,深入分析后发现，电话号码的格式也不统一。

益处:

会提高用户在使用地图时的用户体验，提高用户使用率。

预期的问题

因为该建议会增加提交数据者在提交前修改数据的次数。可能会较低提交者的积极性，使得数据贡献者的人数减少。

结论

通过本次项目，让我熟悉了数据清洗的基本流程，了解了 SQL 的基本使用，同时在清洗邮政编码格式时,更加熟悉了 Python 的基础知识。但是由于本次采用的是国外的地图，由于文化的差异，对地图中的一些信息了解的不是很清楚，
对项目的进行有一定的影响。