OpenStreetMap Data 案例研究

地图区域

San Antonio, TX, United States

- https://www.openstreetmap.org/relation/253556 (https://www.openstreetmap.org/relation/253556)
- https://mapzen.com/data/metro-extracts/metro/san-antonio_texas/ (https://mapzen.com/data/metro-extracts/metro/san-antonio_texas/)

圣安东尼奥马刺队是我喜欢看的NBA球队之一，也是美国南部一座规模较小的城市，城市中河流比较多，人口中墨西哥及拉美裔的比例较高。我选择尝试清理这个城市的地图数据，看看有没有一些有趣的发现。

地图中遇到的问题

通过对数据的初步清理，发现了存在以下问题：

- 街道名称中有过度缩写："Rd", "St" "W", "N", "Ave", "Hwy"
- 街道名称字段中有的值并不是街道的名称，比如"Frogs Leap", "Alomosa Falls"
- 邮编不一致："78217", "78217-1341", "TX 78006"
- 在第二层`"k"`标签中的一部分街道名来自于Tiger GPS数据，并且被分开为几部分，呈现出以下形式：

街道名称的过度缩写

在对数据进行审查的过程中，发现部分街道名称存在过度缩写的问题。为了修正街道名称，采用以下的函数对不规范的街道名称通过加入字典mapping进行修正：

In [ ]:
```
mapping = { "Rd": "Road",
            "St": "Street",
            "Ave":"Avenue",
            "Hwy":"Highway",
            "Hiwy":"Highway",
            "W"  : "West",
            "N"  : "North",
            "E"  : "East",
            "S"  : "South"
            }

def update_name(name, mapping):
    words = name.split()
    for w in range(len(words)):
        if words[w] in mapping:
            if words[w-1].lower() not in ['avenue']:
                words[w] = mapping[words[w]]
                name = " ".join(words)
    return name
```

邮编不一致

数据集中的邮编并不是按照如"78217"一样的5位数字形式，一些邮编出现了不一致的形式，如："78217", "78217-1341", "TX 78006"。通过对不一致的邮编形式进行修正，通过SQL语句查询后的邮编结果如下表：

In [ ]:
```
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags
     UNION ALL
     SELECT * FROM ways_tags) tags
WHERE tags.key='postcode'
GROUP BY tags.value
ORDER BY count DESC LIMIT 10;
```

In [ ]:
```
78255 | 2171
78155 | 1398
78251 | 1124
78230 | 745
78249 | 497
78253 | 376
78216 | 301
78240 | 282
78006 | 187
78666 | 164
```

将序排列城市的频数

In [ ]:
```
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags UNION ALL
      SELECT * FROM ways_tags) tags
WHERE tags.key LIKE '%city'
GROUP BY tags.value
ORDER BY count DESC;
```

In [ ]:
```
San Antonio | 6459
Seguin | 1345
Boerne | 172
San Marcos | 158
New Braunfels | 88
La Vernia | 56
1 | 55
2 | 38
Schertz | 34
Converse | 32
Universal City | 23
10 | 19
5 | 19
4 | 18
12 | 16
20 | 14
30 | 13
Floresville | 13
Helotes | 13
6 | 11
```

绝大部分的节点和道路都是圣安东尼奥市的，但仍然有一部分属于其他城市，比如距圣城市中心56公里(Wikipedia)的Seguin市。

数据描述和其他的想法

这部分是关于数据集的基本统计，使用sqlite查询获得结果

文件大小

In [ ]:
```
san-antonio_texas.osm ......... 275.1 MB
database-sa.db ..........  150.8 MB
nodes.csv ..............  102.9 MB
nodes_tags.csv .........  2.9 MB
ways.csv ...............  8.3 MB
ways_tags.csv .........  23.7 MB
ways_nodes.cv .........  34.7 MB
```

节点数量

In [ ]:
```
sqlite> SELECT COUNT(*) FROM nodes;
```

1228878

道路数量

In [ ]:
```
sqlite> SELECT COUNT(*) FROM ways;
```

142862

唯一用户的数量

In [ ]:
```
sqlite> SELECT COUNT(DISTINCT(e.uid)) FROM (SELECT uid FROM nodes U
NION ALL SELECT uid FROM ways) e;
```

811

贡献前10名用户

In [ ]:
```
sqlite> SELECT e.user, COUNT(*) as num FROM (SELECT user FROM nodes
UNION ALL SELECT user FROM ways) e
GROUP BY e.user ORDER BY num DESC LIMIT 10;
```

In [ ]:
```
kre3d|453261
woodpeck_fixbot|345776
Bellhalla|157130
homeslice60148|135639
Vaderf|23588
TexasNHD|14188
GoldenStar365|12516
25or6to4|11846
balrog-kun|11802
happy5214|11692
```

只出现一次的用户数

In [ ]:
```
sqlite> SELECT COUNT(*) FROM
(SELECT e.user, COUNT(*) as num FROM (SELECT user FROM nodes UNION
ALL SELECT user FROM ways) e
 GROUP BY e.user HAVING num=1) u;
```

167

从排名前10的用户贡献数来看，排名第一的用户"kre3d"做出了33%的贡献，排名第二的用户做出了25.2%的贡献，合集占总数量的58.2%。可见一小部分“核心用户”为这个城市的地图做出了大部分的贡献。

其他的想法

其他的数据探索

大学校园的数量

In [ ]:
```
sqlite> SELECT count(*) FROM ways_tags WHERE value = 'university';
```

136

前10位流行的美食

In [ ]:
```
sqlite> SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
    JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='restaura
nt')
i
    ON nodes_tags.id=i.id WHERE nodes_tags.key='cuisine'
GROUP BY nodes_tags.value
ORDER BY num DESC LIMIT 10;
```

In [ ]:
```
mexican 18
pizza 13
american 12
burger 9
thai 8
barbecue 4
chinese 4
italian 4
asian 3
greek 3
```

果然，作为美国墨西哥裔人口比例较大的南方地区，墨西哥美食在圣安东尼奥最为流行。

Conclusion

在审查过程中发现的最为突出的问题是由于部分OpenStreetMap的数据来源于GPS的数据，导致一些道路以及节点名称的混乱。对于这样的GPS数据如果可以根据其数据结构，在数据输入时使用脚本程序进行修正，可以在很大程度上减少手动清理的复杂度，使得数据更为“干净”。此外，一些地名以及少量的邮政编码存在着格式不规范的问题。如果可为贡献者提供自动纠正格式不规范的插件或者示例说明，可以很大程度上避免这类问题的出现。当然，由于地名的复杂性，不可能穷尽所有的地名格式。