Table of Contents

- 1 初步探索数据
  - 1.1 数据描述
  - 1.2 标签数量
  - 1.3 数据问题
  - 1.4 导入数据库
- 2 数据概述
  - 2.1 文件大小
  - 2.2 各种数量统计
- 3 关于数据集的其他想法
  - 3.1 改进或分析数据的建议
  - 3.2 所选节点类型的数量统计
- 4 总结
- 5 参考

初步探索数据

数据描述

选取了广州市的地图数据，因为广州市我最熟悉的城市，在清理和整理数据方面会更容易发现数据的问题，而不是通过猜测和估计，如一些地址和地名的问题。我选择广州市区的地图数据，主要包括海珠区、荔湾区、越秀区、天河区、白云区和番禺区的部分地区，为了使地图数据文件大小超过50MB。

地图链接

- https://www.openstreetmap.org/#map=13/23.1118/113.3265
- https://overpass-api.de/api/map?bbox=113.2193,23.0548,113.4338,23.1688

标签数量

In [8]:
```
! python tags.py
```

```
{'bounds': 1,
 'member': 90313,
 'meta': 1,
 'nd': 247396,
 'node': 211387,
 'note': 1,
 'osm': 1,
 'relation': 1470,
 'tag': 93489,
 'way': 28780}
```

数据问题

审查数据质量，发现地址的问题有这些：

In [9]:
```
! python key_problem.py
```

```
{'lower': 82253, 'lower_colon': 11123, 'other': 113, 'problemchars': 0}
```

In [1]:
```
! python audit.py
```

```
{'39': set(['Hong Road No 39']),
 u'510610': set([u'China, Guangdong Sheng, Guangzhou Shi, Tianhe Qu, TianHe GongYuan, Tianhe Rd, \u592a\u53e4\u6c47L307\u53f7
, \u90ae\u653f\u7f16\u7801: 510610']),
 'Bilu': set(['Lang wang Bilu']),
 u'East': set([u'\u73e0\u6c5f\u4e1c\u8def Zhujiang Road East']),
 'Guangdong,': set(['Room 322, Jintao Building,26# Guangyuanzhong Road, Baiyun District,Guangzhou, Guangdong,']),
 'Guangzhou': set(['Dishifu Road, Guangzhou',
                   'Room 1302, E2 Building, Jin Gui Yuan, #6 Jin Gui Jie, Jie Fang Bei Lu, Bai Yun District, Guangzhou']),
 u'Lu': set([u'\u6ee8\u6c5f\u4e1c\u8def Binjiang Dong Lu']),
 u'Rd': set(['Linhe West Cross Rd',
             'Liurong Rd',
             u'\u5185\u73af\u8def Inner Ring Rd']),
 u'Shop': set([u'\u6c99\u592a\u5357\u8def, upstairs in Health food Shop']),
 'St': set(['Yanyu S St']),
 u'West': set([u'\u73e0\u6c5f\u897f\u8def Zhujiang Road West']),
 'Xi': set(['Huang Pu Dadao Xi']),
 'ave': set(['baogang ave']),
 'road': set(['Huacheng road']),
 'road)': set(['Chigang Lu (road)'])}
沙太南路, upstairs in Health food Shop => 沙太南路, upstairs in Health food Shop
Room 322, Jintao Building,26# Guangyuanzhong Road, Baiyun District,Guangzhou, Guangdong, => Room 322, Jintao Building,26# Gua
ngyuanzhong Road, Baiyun District,Guangzhou, Guangdong,
Chigang Lu (road) => Chigang Lu (road)
Huang Pu Dadao Xi => Huang Pu Dadao West
珠江西路 Zhujiang Road West => 珠江西路 Zhujiang Road West
Room 1302, E2 Building, Jin Gui Yuan, #6 Jin Gui Jie, Jie Fang Bei Lu, Bai Yun District, Guangzhou => Room 1302, E2 Building,
Jin Gui Yuan, #6 Jin Gui Jie, Jie Fang Bei Lu, Bai Yun District, Guangzhou
Dishifu Road, Guangzhou => Dishifu Road, Guangzhou
Hong Road No 39 => Hong Road No 39
Yanyu S St => Yanyu S Street
内环路 Inner Ring Rd => 内环路 Inner Ring Road
```

```
Linhe West Cross Rd => Linhe West Cross Road
Liurong Rd => Liurong Road
Lang wang Bilu => Lang wang Bilu
滨江东路 Binjiang Dong Lu => 滨江东路 Binjiang Dong Road
baogang ave => baogang Avenue
珠江东路 Zhujiang Road East => 珠江东路 Zhujiang Road East
China, Guangdong Sheng, Guangzhou Shi, Tianhe Qu, TianHe GongYuan, Tianhe Rd, 太古汇L307号, 邮政编码: 510610 => China, Guangdon
g Sheng, Guangzhou Shi, Tianhe Qu, TianHe GongYuan, Tianhe Rd, 太古汇L307号, 邮政编码: 510610
Huacheng road => Huacheng Road
```

总结起来，主要有以下问题：

- 地址和名称的英文名的缩写和大小写问题，如('Linhe West Cross Rd')
- 地址和名称英文的翻译问题，如('Huang Pu Dadao Xi')
- 地方名称问题，如('中国工商银行'，'工商银行')

接下来更详细地讨论以上问题

地址英文名的缩写问题

地图数据中发现地址英文名存在有些缩写，有些没有缩写的情况，需要规范这些数据的统一性，统一不缩写，例如把'Linhe West Cross Rd'变成'Linhe West Cross Road'，同样大小写的问题也用相同的处理办法。

地址英文的翻译问题

英文翻译问题比较复杂，存在的问题如'China, Guangdong Sheng, Guangzhou Shi, Tianhe Qu, TianHe GongYuan, Tianhe Rd'，英文地址中某些部分直接是拼音，如Sheng, Shi, Qu, Jie, Lu...若考虑数据的统一性，把拼音部分转成英文需要判断该拼音是否是省、市、区、街、路等意思 ，存在一定困难。 折中办法是只转换地址最后一个字的拼音，这样处理没那么复杂，但地址的统一性还有待完善。

地方名称问题

发现有些名称存在别名或翻译成多个名称，如快餐店麦当劳存在几个名字，如MacDonald、MacDonald's，可统一名字为麦当劳；还有银行名称如中国工商银行，还有另外一种叫法如工商银行，同样的还有建设银行。这些地方名称需要统一起来。

导入数据库

运行data.py生成地图数据json文件，用mongoimport把json文件导入数据库。

In [ ]:
```
2018-02-26T16:23:43.801+0800 connected to: localhost
2018-02-26T16:23:43.801+0800 dropping: db_news.gz
2018-02-26T16:23:46.764+0800 [###.....................] db_news.gz 8.38MB/65.2MB (12.9%)
2018-02-26T16:23:49.818+0800 [######..................] db_news.gz 17.1MB/65.2MB (26.2%)
2018-02-26T16:23:52.764+0800 [########................] db_news.gz 25.8MB/65.2MB (39.5%)
2018-02-26T16:23:55.783+0800 [############............] db_news.gz 33.5MB/65.2MB (51.5%)
2018-02-26T16:23:58.764+0800 [###############.........] db_news.gz 41.6MB/65.2MB (63.9%)
2018-02-26T16:24:01.830+0800 [#################.......] db_news.gz 48.5MB/65.2MB (74.4%)
2018-02-26T16:24:05.584+0800 [####################....] db_news.gz 58.4MB/65.2MB (89.5%)
2018-02-26T16:24:07.386+0800 [########################] db_news.gz 65.2MB/65.2MB (100.0%)
2018-02-26T16:24:07.386+0800 imported 240167 documents
```

总共导入了240167个文档。

数据概述

文件大小

map.osm ..... 50.3 MB
map.osm.json .... 68.8 MB

各种数量统计

文件数量

In [ ]:
```
> db.gz.find().count()
240167
```

节点数量

In [ ]:
```
> db.gz.find({'type':'node'}).count()
211387
```

途径数量

In [ ]:
```
> db.gz.find({'type':'way'}).count()
28778
```

唯一用户数量

In [ ]:
```
> db.gz.distinct('created.user').length
387
```

最大贡献的用户

In [ ]:
```
> db.gz.aggregate([{'$match':{'created.user':{'$exists':1}}},
...                  {'$group':{'_id':'$created.user', 'count':{'$sum':1}}},
...                  {'$sort':{'count':-1}},
...                  {'$limit':1}])
{ "_id" : "羊角忠实黑", "count" : 60698 }
```

贡献只有1次的用户数量

In [ ]:
```
> db.gz.aggregate([{'$match':{'created.user':{'$exists':1}}},
...                  {'$group':{'_id':'$created.user', 'count':{'$sum':1}}},
...                  {'$group':{'_id':'$count', 'num_users':{'$sum':1}}},
...                  {'$sort':{'num_users':-1}},
...                  {'$limit':1}])
{ "_id" : 1, "num_users" : 106 }
```

数据源数量

In [ ]:
```
> db.gz.distinct('source').length
26
```

数据源前5

In [ ]:
```
> db.gz.aggregate([{'$match':{'source':{'$exists':1}}},
...                  {'$group':{'_id':'$source', 'count':{'$sum':1}}},
...                  {'$sort':{'count':-1}},
...                  {'$limit':5}])
{ "_id" : "Yahoo hires", "count" : 459 }
{ "_id" : "GPS", "count" : 154 }
{ "_id" : "bing", "count" : 124 }
{ "_id" : "Bing", "count" : 86 }
{ "_id" : "Yahoo", "count" : 73 }
```

关于数据集的其他想法

提交文件包含一个或多个用于改进或分析数据的额外建议

提交文件中详尽地讨论了实施改进的益处及一些预期的问题

改进或分析数据的建议

地图数据中最大的问题是地址名称的问题，因为所选的地区中地名或地址的标签name存在中文、拼音和英文几种格式，考虑数据统一性，需要统一一种格式。最理想的情况是统一为中文，因为若统一为英文，需要考虑翻译的准确性，若统一为拼音，存在表达意思不清晰的问题。统一成中文可以清晰表达地名或地址，但预期的问题是需要花费很多工作去处理转换，可能也会转换得不准确。

所选节点类型的数量统计

数量前10位的设施类型

In [ ]:
```
> db.gz.aggregate([{'$match':{'amenity':{'$exists':1}}},
...                  {'$group':{'_id':'$amenity', 'count':{'$sum':1}}},
...                  {'$sort':{'count':-1}},
...                  {'$limit':10}])
{ "_id" : "restaurant", "count" : 248 }
{ "_id" : "school", "count" : 206 }
{ "_id" : "bank", "count" : 152 }
{ "_id" : "parking", "count" : 128 }
{ "_id" : "toilets", "count" : 90 }
{ "_id" : "hospital", "count" : 85 }
{ "_id" : "fast_food", "count" : 74 }
{ "_id" : "bus_station", "count" : 62 }
{ "_id" : "cafe", "count" : 53 }
{ "_id" : "place_of_worship", "count" : 42 }
```

数量前10位的餐厅或快餐厅

In [ ]:
```
> db.gz.aggregate([{'$match':{'amenity':{'$in':['restaurant','fast_food']}}},
...                  {'$group':{'_id':'$name', 'count':{'$sum':1}}},
...                  {'$match':{'_id':{'$ne':null}}},
...                  {'$sort':{'count':-1}},
...                  {'$limit':10}])
{ "_id" : "麦当劳", "count" : 33 }
{ "_id" : "肯德基", "count" : 17 }
{ "_id" : "必胜客", "count" : 8 }
{ "_id" : "赛百味", "count" : 5 }
{ "_id" : "72街", "count" : 5 }
{ "_id" : "真功夫", "count" : 5 }
{ "_id" : "都城快餐", "count" : 5 }
{ "_id" : "Burger King", "count" : 3 }
{ "_id" : "萨莉亚", "count" : 2 }
{ "_id" : "同湘会", "count" : 2 }
```

最受欢迎的菜系

In [ ]:
```
> db.gz.aggregate([{'$match':{'amenity':{'$in':['restaurant','fast_food']}}},
...                  {'$group':{'_id':'$cuisine', 'count':{'$sum':1}}},
...                  {'$match':{'_id':{'$ne':null}}},
```

```
...                 {'$sort':{'count':-1}},
...                 {'$limit':3}])
{ "_id" : "chinese", "count" : 54 }
{ "_id" : "burger", "count" : 25 }
{ "_id" : "chicken", "count" : 13 }
```

数量前10位的银行

In [ ]:
```
> db.gz.aggregate([{'$match':{'amenity':'bank'}},
...                 {'$group':{'_id':'$name', 'count':{'$sum':1}}},
...                 {'$match':{'_id':{'$ne':null}}},
...                 {'$sort':{'count':-1}},
...                 {'$limit':10}])
{ "_id" : "建设银行", "count" : 22 }
{ "_id" : "中国银行", "count" : 21 }
{ "_id" : "农业银行", "count" : 20 }
{ "_id" : "工商银行", "count" : 19 }
{ "_id" : "招商银行", "count" : 10 }
{ "_id" : "广州银行", "count" : 6 }
{ "_id" : "交通银行", "count" : 6 }
{ "_id" : "广发银行", "count" : 6 }
{ "_id" : "民生银行", "count" : 5 }
{ "_id" : "兴业银行", "count" : 3 }
```

总结

此数据还有待整理和清理，需要处理的问题还有很多，例如地名或地址部分。但总的来说以上几个步骤练习了整理数据的过程。

- 用有效性、准确率、完整性、一致性和均匀性来评估数据的质量。
- 解析并且从.xml文件格式和收集数据。
- 处理来自大量文件和大型文件并且编程进行清理的数据。
- 使用 MongoDB存储、查询和聚合数据。

参考

- https://github.com/j450h1/P3-Data-Wrangling-with-MongoDB
- http://nbviewer.jupyter.org/github/jm974/openstreetmap/blob/master/OpenStreetMap.ipynb
- http://nbviewer.jupyter.org/github/tychen927/openstreetmap_mongoDB/blob/master/main.ipynb