OpenStreet 项目分析(mongodb)

1、项目中遇到的问题

由于下载数据文件较大，选择用代码生成了样本数据，在检查样本数据时，未发现数据异常，但是在数据清理获取数据时，每一次获取数据时，先要设计一个数据类型检查脚本，即用正则方式匹配数据，如果有异常数据则输出异常数据，在检测数据时发现了一个主要问题：

- ZipCode 与通用 ZipCode 不同。如：name:Faulkland Road zipcode:19805:19808
- 原文件过大，清洗数据时验证不方便，导入数据库时比较费时

处理异常 ZipCode

普通 ZipCode 是五位数字，在检查样本时发现一些 ZipCode 格式为 19805:19808，记录这些异常 Zip 所在的区域，经过上网查证，这些地区存在多个 ZipCode，断定该异常值代表的是 ZipCode 的起始值与终止值。处理方式：将该异常值按":"分组，用 for 循环获取之间的数值，存在列表中，以下是清理完毕后导入数据库所查询到的结果：

查看 ZipCode

```
>db.getCollection('zip_code').find({"name":"Faulkland Road
"})
```

城市名为：Faulkland Road 的 ZipCode

```
{
    "_id" : ObjectId("5a8d44ffe7d27669403ee861"),
    "name" : "Faulkland Road",
    "zip_code" : [
```

```
            "19805",
            "19806",
            "19807",
            "19808"
        ]
    }
```

查询结果如上图所示，**Faulkland Road** 的异常值（**19805:19808**）已经成功转化成正常值，由于此项数据是由客户端 **Robo 3T** 查询所得，结果展示位文档形式。

处理原文件过大

用代码生成一个小的样本文件，处理小文件的数据以及异常值，处理完毕后，将文件替换成大文件，生成 json 文件时，可以按照数据分成不同的 json 文件

2、数据集统计

此章节用于记录数据库的各项数据统计

文件大小

delaware-latest.osm.....203 MB
node.json.....86 MB
user.json.....77 MB
way.json.....4 MB
zip_code.json.....832 KB
restaurant.json.....12 KB

#Node 数量

```
>db.getCollection('node').find({}).count()
```

```
941707
```

#Way 数量

```
>db.getCollection('way').find({}).count()
```

```
92076
```

#唯一用户数量

```
>db.getCollection('user').distinct("user").length
```

```
936
```

#贡献最多的用户

```
>db.getCollection('user').aggregate([{"$group":{"_id":"$us
er", "count":{"$sum":1}}},
{ "$sort":{"count": -1}}, {"$limit":1}])
```

```
/* 1 */
{
    "_id" : "ceyockey",
    "count" : 221266.0
}
```

#只贡献一次的用户

```
>db.getCollection('user').aggregate([{"$group":{"_id":"$us
er", "count":{"$sum":1}}},
```

```
{"$group":{"_id":"$count", "num_users":{"$sum":1}}}, {"$so
rt":{"_id":1}},

{"$limit":1}])


/* 1 */

{

    "_id" : 1.0,

    "num_users" : 177.0

}
```

#餐馆数量

```
>db.getCollection('restaurant').find({}).count()


129
```

#使用次数最多的 zipcode

```
>db.getCollection('zip_code').aggregate([{"$unwind":"$zip_
code"},{"$group":{"_id":"$zip_code","count":{"$sum":1}}},
{"$sort":{"count":-1}},{"$limit":5}])


/* 1 */

{

    "_id" : "19720",

    "count" : 1069.0

}


/* 2 */
```

```
{
    "_id" : "19711",
    "count" : 963.0
}

/* 3 */
{
    "_id" : "19702",
    "count" : 859.0
}

/* 4 */
{
    "_id" : "19808",
    "count" : 712.0
}

/* 5 */
{
    "_id" : "19701",
    "count" : 709.0
}
```


3、数据集的改进建议

贡献数据集的建议

贡献数据的用户比较集中在某些人身上，以下书关于该项的分析：

- 贡献最多用户所贡献的比例（ceyockey）：21.3%
- 前两名用户所共享比例（tlt83 和 ceyockey）:39.3%
- 前十名贡献用户所占比例：64.7%

通过对贡献地图百分比的统计，发现用户普遍不愿意参与贡献地图数据，猜测原因有两个：

- 该网站缺少奖励机制，导致用户不愿意参与到贡献地图中来
- 用户不了解贡献地图数据的方法

以下是改进建议：

- 推行一些奖励机制，比如奖金，网站的勋章

  益处：

  1、使人们参与到网站地图数据更新中来
  2、由于奖励机制的推出，会增加网站的知名度。

  预期的问题：

  1、网站的内容可能需要重新布局，需要编写新增模块，但是预计工作量不会很大。
  2、由于奖励机制的推出，有的用户可能为获取奖励提交不实信息，这需要设计算法，新的审核模式。

- 开设一个页面普及绘测地图数据的说明以及方法，该项举措能够减少错误数据的产生

  益处：
  提高绘制地图的准确率，明确奖惩机制。

  预期问题：
  现阶段评估不会产生不良问题。

结论

在重新审视这些数据，发现对周边节点类型处理的不是很好，在原代码中只筛选了餐馆的数据，所以导入数据库后只有餐馆数据，缺少了灵活性，由于不了解各个标签的含义与规则，会导致检查数据是否存在非法数据时不够准确，清洗后的数据能够根据城市查询餐馆信息，以及城市所用 zipcode