# OpenStreetMap项目报告

# 1. 地图区域

地图范围： 中国，上海

主要选取了西至虹桥机场，东至浦东机场的范围。

下载地址： http://overpass-api.de/api/map?bbox=121.2869,31.1176,121.6434,31.2915 (http://overpass-api.de/api/map?bbox=121.2869,31.1176,121.6434,31.2915)

# 2. 数据清洗过程中碰到的问题

1.tags类型的节点中，邮政编码(key='postcode')的value部分：应该是6位数字，实际情况存在中文、或是数字位数不对的情况。

2.tags类型的节点中，value中存在特殊字符，无法导入数据库。

3.tags类型的节点中，电话(key='phone')的value部分，存在以下几种情况：
(a)多个电话、或是分机号
(b)不同格式(未必包含+86、未必包含+021、021写成21、是否包含'-'号的分隔符等)
(c)不同类型的电话类型（手机号、固话、400电话、五位全国通用电话等)

## 2.1. 邮政编码问题

国内统一的邮政编码位数为6位数字，因此考虑数据清洗时去除其他不匹配的情况。

In [ ]:
```
#用于去除错误邮编
REGULAR_POSTCODE =re.compile(r'^[0-9]{6}$')
def is_postcode(src_postcode, regular_postcode=REGULAR_POSTCODE):
    postcode = re.search(regular_postcode, src_postcode)
    if postcode:
        return True
    else:
        return False
```

执行后显示剔除如下情况的错误格式：
错误的邮编格式{'value': '2000080', 'key': 'postcode', 'type': 'addr', 'id': '4364315493'}
错误的邮编格式{'value': '201315 上海', 'key': 'postcode', 'type': 'addr', 'id': '148014167'}
错误的邮编格式{'value': '201315 上海', 'key': 'postcode', 'type': 'addr', 'id': '148014201'}
错误的邮编格式{'value': '2000080', 'key': 'postcode', 'type': 'addr', 'id': '293504473'}
错误的邮编格式{'value': '20032', 'key': 'postcode', 'type': 'addr', 'id': '307449542'}
错误的邮编格式{'value': '20032', 'key': 'postcode', 'type': 'addr', 'id': '307455604'}

剔除错误格式后再次查询邮政编码：

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

| value | count |
|---|---|
| 201203 | 48 |
| 201315 | 34 |
| 200231 | 25 |
| 200120 | 13 |
| 200040 | 11 |
| 200050 | 10 |
| 200032 | 9 |
| 200031 | 9 |
| 200135 | 8 |
| 201206 | 8 |

## 2.2 value中存在不可导入数据库的特殊字符

在导入数据库过程中，导入语句报错提示：“ 存在特殊字符：\xF0\x9F\x93\xAE for column 'value ”
使用notepadd++中的字符转换功能，转换后的字符为： ，在map文件中查找定位到邮局的nodes_tags, 推测是用户输入时复制图像进入到文本框内。 考虑对tags相关的csv文件中的value字段进行一次扫描，去除坏字符。

In [1]:
```
#用于替换value字段内的奇怪字符
VALUE_REPLACE_CHARS = ' '
def reject_bad_chars_of_value(src_string, badchars=VALUE_REPLACE_CHARS):
    result = re.sub(badchars, "", src_string)
    return result
```

执行后再次导入数据成功。

## 2.3 调整电话格式

地图中的电话标记存在很多问题，没有进行统一的格式化输入。
问题包括：
(a)一个电话标签确有多个电话（分隔符是用户自定的）、或电话标签内包含分机号；
(b)不规范的格式输入，包括多钟格式：未必包含+86、未必包含+021、021写成21、包含'-'号的分隔符等；
(c)不同类型的电话类型（手机号、固话、400电话、五位全国通用电话等)。

用下述SQL语句查询得到电话的一些例子：

In [ ]:
```
select t.`key`, t.`value`
from (select * from nodes_tags union all select * from ways_tags union all select * from rel_
tags) t
where t.`key`='phone';
```

862122163900
+86 6361 2898
021-63914848, 021-63522222
+86 21 38809988
2164312091
86-21-50559888
+2147483647
+862164712821
+18 13621675140
02162883030
+86-21-5160-7888
+86 138 1609 3747
(021) 3356-3996
021-63779282
+86 (0)21-68778787

分析后，考虑分为三步处理：
1.去除数字中的所有格式（空格，+号，-号，括号）；
2.正则匹配，去除国家号与城市号，获取纯粹的的电话号码（8位固话，11位手机号，7位400电话，5位全国通用电话）；
3.生成正规格式的电话。(分为固话、手机号、400电话、全国通用电话等类型)

使用的处理函数如下：

In [ ]:
```
#用于规范电话的格式
#如果校验通过，返回正确格式的电话，否则返回空字符串
REGULER_PHONE = re.compile(r'^(86)?(021|21)?([0-9]{8}|[0-9]{11}|400[0-9]{7}|[0-9]{5})$')
def audit_phone(src_phone, regular_phone=REGULER_PHONE):
    new_phone = re.sub(r'[+ \-()]', '', src_phone)
    phone = re.search(regular_phone, new_phone)
    if phone:
        phone_num = phone.group(3)
        phone_type = len(phone_num)
        #固话
        if phone_type==8:
            return '+86-021-' + phone_num
        #手机
        elif phone_type==11:
            return '+86-' + phone_num
        #400电话
        elif phone_type==10:
            return '+86-' + phone_num
        #全国通用电话
        elif phone_type==5:
            return '+86-' + phone_num
        else:
            return ''
    else:
        return ''
```

执行后显示剔除如下情况的错误格式：
错误的电话格式OrderedDict([('id', '477661623'), ('key', 'phone'), ('value', '021-63914848, 021-63522222'), ('type', 'regular')])
错误的电话格式OrderedDict([('id', '2345419578'), ('key', 'phone'), ('value', '+18 13621675140'), ('type', 'regular')])
错误的电话格式OrderedDict([('id', '3609090494'), ('key', 'phone'), ('value', '+86 8621 5118 1222'), ('type', 'regular')])
错误的电话格式OrderedDict([('id', '159787864'), ('key', 'phone'), ('value', '8008103088'), ('type', 'regular')])
错误的电话格式OrderedDict([('id', '293862126'), ('key', 'phone'), ('value', '+86 21 64874095*208'), ('type', 'regular')])
错误的电话格式OrderedDict([('id', '376223385'), ('key', 'phone'), ('value', '0862162838711'), ('type', 'regular')])
错误的电话格式OrderedDict([('id', '392578179'), ('key', 'phone'), ('value', '(+86)21/52068000'), ('type', 'regular')])

剔除错误格式的电话后，规范格式数据的例子如下所示：

phone +86-021-33533053
phone +86-021-64857333
phone +86-4008807729
phone +86-021-60548081
phone +86-021-68778787
phone +86-021-33680676
phone +86-021-61551988
phone +86-95580
phone +86-13818175006

# 3. 数据集的概述统计

## 3.1 文件大小

```
map.xml                77M
nodes.csv              27M
nodes_tags.csv         2M
relations.csv          1M
rel_members.csv        2M
rel_tags.csv           1M
ways.csv               3M
ways_nodes.csv         10M
ways_tags.csv          5M
```

## 3.2 唯一用户数量

In [ ]:
```
#唯一用户数量
select count(*) from
(select `user` from nodes union select `user` from ways union select `user` from relation)
t;
```

获得数量为：1185

In [ ]:
```
#每个用户的贡献次数（选取显示前20贡献度的用户）
select t.`user`, count(*) as count from
(select `user` from nodes union all select `user` from ways union all select `user` from rel
ation) t
group by t.`user`
order by count desc;
```

| user | count |
|---|---|
| zzcolin | 57497 |
| HWST | 41568 |
| Koalberry | 24914 |
| yangfl | 20434 |
| z_i_g_o | 18872 |
| Xylem | 17704 |
| u_kubota | 15014 |
| yhilan | 13038 |
| Esperanza36 | 11149 |
| aighes | 6966 |
| Knockerclot0715 | 6496 |
| Austin Zhu | 6464 |
| Mirarkitty | 5837 |
| KartaBY | 5512 |
| lukys1 | 5249 |
| DAJIBA | 4411 |
| chdr | 4391 |
| Poetini LEHARTEL | 4215 |
| zhongguo | 4027 |
| anthropologist | 3905 |

## 3.3 node, way, relation的数量

In [ ]:
```
#node节点的数量
select count(*) from nodes;
```

341258

In [ ]:
```
#way节点的数量
select count(*) from ways;
```

49340

In [ ]:
```
#relation节点的数量
select count(*) from relation;
```

652

## 3.4 其他节点类型数量统计

In [ ]:
```
#咖啡店的数量
select count(*) from
(select * from nodes_tags
union all select * from ways_tags
union all select * from rel_tags) t
where t.`value`='cafe';
```

254

In [ ]:
```
#便利设施、商店的数量
select count(*) from
(select * from nodes_tags
union all select * from ways_tags
union all select * from rel_tags) t
where t.`key` in ('amenity', 'shop');
```

4371

In [ ]:
```
#出现次数小于10的key值
select count(*) from
(select t.`key` as key_str from
(select * from nodes_tags
union all select * from ways_tags
union all select * from rel_tags) t
group by key_str
having count(*)<10) t2
```

422

# 4. 额外的改进建议

## 4.1 改进、分析数据的建议

审计过程中的一些不规范格式输入、特殊字符等情况，可以通过在页面前端的输入部分进行格式校验与控制。比如标签值仅限输入中英文字符、数字、和其他常用字符等；

## 4.2 实施改进的益处与问题

实施输入控制后，可以大幅减少错误字符、错误格式的数据量；但是引入输入控制比较困难，本身需要一套自己的维护工具。

# 5. 项目总结

本次项目是我第一次实施数据清理。在项目进行中，python编码、数据审计、数据库导入都是对我的挑战。期间不断的试错、上网查找问题解决办法等，对于我的能力提升也很有帮助。此外在处理公开数据集后，我也认识到数据清洗是多么需要花费时间与耐心的一项工作。

# 6.参考链接


CSV文件处理：
http://blog.csdn.net/ko_tin/article/details/72627266 (http://blog.csdn.net/ko_tin/article/details/72627266)
http://blog.csdn.net/huitailang1991/article/details/54946528 (http://blog.csdn.net/huitailang1991/article/details/54946528)

正则匹配：
https://docs.python.org/2/library/re.html (https://docs.python.org/2/library/re.html)

Cerberus库：
http://docs.python-cerberus.org/en/stable/ (http://docs.python-cerberus.org/en/stable/)

数据库相关问题：
http://blog.csdn.net/u013063153/article/details/53304261 (http://blog.csdn.net/u013063153/article/details/53304261)
http://blog.csdn.net/gvfdbdf/article/details/49455381 (http://blog.csdn.net/gvfdbdf/article/details/49455381)
http://bbs.csdn.net/topics/390661358 (http://bbs.csdn.net/topics/390661358)
https://stackoverflow.com/questions/6621530/1264-out-of-range-value-fix
(https://stackoverflow.com/questions/6621530/1264-out-of-range-value-fix)
http://www.oschina.net/question/1165991_2200405 (http://www.oschina.net/question/1165991_2200405)
http://blog.csdn.net/a14206149/article/details/35991455 (http://blog.csdn.net/a14206149/article/details/35991455)