整理OpenStreetMap数据

## 所选地图区域：

中国，上海

<https://mapzen.com/data/metro-extracts/metro/shanghai_china/>

Metro Extracts ) Shanghai 
Yanche 
. Taizhou 
Nanjin Zhenj•ang • 
Changzhouo 
Shiji(l Hu 
;ling 
Wuhu 
Xuanzhou 
Wuxi 
Huzhbu 
Hangzhou 
Shanghai'. 
Ningbo , 
ttps://ma pze n. c o m/d ata/metro- extra Cts/ 

上海是中国的特大城市之一，也是我呆了将近十年的城市。对于这个城市，有熟悉，也有陌生，正好借着这个机会好好再深入了解下上海。

## 地图中发现的问题

通过选定其中的一个样本

<http://www.openstreetmap.org/export#map=12/31.2063/121.6245>

OpenStreetMap 
31 238g 
121 57 0 
31 ． 17 0 
许 可 协 议 
121 乇 760 
海 市 
0 
上 生 路 
洋 泾 街 道 
一 花 木 地 
一 》 北 蔡 镇 
《 南 枞 
GPSShi 迹 
52D A20 
52D 庶 20 
． 唐 
川 沙 新 帕 
用 户 日 记 
帮 助 
关 于 
Openst 「 “ tM 叩 数 据 采 用 开 放 数 据 共 車 开 放 据 库 许 可 
协 议 囤 DOL 脛 权 ． 
如 果 上 面 的 导 ± 工 且 铤 姣 了 ， 适 考 慮 便 甲 下 面 列 ± 的 
宗 源 宗 导 ± ： 
Overpass 角 P 《 
"k 0 的 是 ． 张 江 镇 ， 
康 侑 蕷 
引 501 孬 
G15D1 A30 

用data.py跑数据，发现的问题主要有：

1. 中英夹杂的城市名称
2. 中英文夹杂的道路名称
3. 邮编号码不正确，正确的邮编号码是6位数，个别错误的号码有5位和7位

### 中英夹杂的城市名称

将数据导入SQL之后，做一些基本的查询

select value, count(\*) from nodes\_tags where key='city' group by value order by count(\*) DESC limit 10;

"上海市" "414"

"上海" "287"

"Shanghai" "52"

"Hangzhou" "25"

"张家港市" "16"

"上海市闵行区" "13"

"Ningbo" "12"

"杭州市" "10"

"Jiading" "9"

"杭州" "8"

可以看到city的内容很乱，又没有统一的规则，但是postcode有规则，可以借助这个来清洗数据

### postcode整体情况

select length(value),count(\*) from tags where key='postcode' group by length(value) order by count(\*) desc;

"6" "978"

"5" "5"

"7" "2"

"9" "2"

可以看到6位数是最多的，也是正确的邮编长度，而剩下的就是错误的位数。将不是6位数的postcode删除。

清洗之后查看邮编的大致情况

select value,count(\*) from tags where key='postcode' group by value order by count(\*) desc limit 10;

"200240" "245"

"201620" "63"

"212003" "48"

"201203" "47"

"201315" "36"

"314211" "31"

"200231" "26"

"214121" "26"

"215600" "26"

"215000" "23"

根据邮编规则，20开头的邮编代表上海市，21/22开头的邮编代表江苏省，31开头的邮编代表浙江省。用这个关联来清洗开头city数据。由于本次清洗主要针对上海，所以，目前仅清洗开头20的数据。

select a.id,a.key,a.value,b.key,b.value from nodes\_tags as a,nodes\_tags as b where a.id=b.id and a.key='city' and b.key='postcode' and b.value like '20%' group by a.value;

"2559764097" "city" "Anting" "postcode" "201805"

"4114670192" "city" "Jiading" "postcode" "201821"

"603090750" "city" "Pudong District, Shanghai" "postcode" "201203"

"4628026169" "city" "Shaghai" "postcode" "200092"

"4452121885" "city" "Shanghai" "postcode" "201103"

"2408433892" "city" "West BDC, Shanghai" "postcode" "201803"

"4583821529" "city" "shanghai" "postcode" "200052"

"4406766180" "city" "上海" "postcode" "200093"

"3105891888" "city" "上海嘉定区" "postcode" "201807"

"4403840264" "city" "上海市" "postcode" "200443"

"4508518225" "city" "上海市闵行区" "postcode" "201100"

"4378607643" "city" "上海市静安区昌平路68号601" "postcode" "200041"

"2870580111" "city" "上海闵行" "postcode" "200240"

"2326827862" "city" "临港新城" "postcode" "201306"

"3498998329" "city" "松江区" "postcode" "201620"

可以看出，这些都属于上海市，因此修正city=“上海市”

update nodes\_tags set value='上海市' where key='city' and id in (select a.id from nodes\_tags as a,nodes\_tags as b where a.id=b.id and a.key='city' and b.key='postcode' and b.value like '20%' group by a.value);

update nodes\_tags set value='上海市' where key='city' and (value='上海' or value='Shanghai');

### 道路问题

在道路这个问题上，经常出现缩写字符，比如Rd，road，Raod等，规则不统一，因此用audit.py清洗这部分数据

先用street\_name=street\_name.split('(')[0]等将道路名称里面的附加内容删掉。

expected = ["Street", "Avenue", "Boulevard", "Drive", "Court", "Place", "Square", "Lane", "Road",'Market','Zone', 'District','Mall',

"Trail", "Parkway", "Commons","Block",'Garden','Highway','S308','Dong','Park','West']

根据实际情况更新audit的内容，最终清洗结果如下：

nanjing => Nanjing Road

WenSanLu DianZi XinXi JieQu => WenSanLu DianZi XinXi Block

Fengxian Lu => Fengxian Road

Nandang Dong Lu => Nandang Dong Road

Taian Lu => Taian Road

Zhenming Lu => Zhenming Road

ZhongShangNanEr Lu => ZhongShangNanEr Road

Chuanchang Lu => Chuanchang Road

Jindu Rd => Jindu Road

Yongfu Rd. => Yongfu Road

Fuquan Rd. => Fuquan Road

Changhua Rd. => Changhua Road

Bibo Rd. => Bibo Road

East Zhenchuan Rd. => East Zhenchuan Road

Chengnan Rd. => Chengnan Road

Wuzhong Rd. => Wuzhong Road

Qingyang Rd. => Qingyang Road

Pudong avenue => Pudong Avenue

Siping => Siping Road

yongkang rd => yongkang Road

fuyou rd => fuyou Road

Fuxing rd => Fuxing Road

yanan rd => yanan Road

wukang lu => wukang Road

yongfu lu => yongfu Road

fuzhou lu => fuzhou Road

fumin lu => fumin Road

CaoXi Bei lu => CaoXi Bei Road

Nachan lu => Nachan Road

Ling long lu => Ling long Road

hehuaxing => Hehuaxing Road

Xiangyang => Xiangyang Road

Yuanli Rd => Yuanli Road

E-Shan Rd => E-Shan Road

Wukang Rd => Wukang Road

Yining Rd => Yining Road

Inner Ring Rd => Inner Ring Road

Xingzhou Rd => Xingzhou Road

Xian Xia Rd => Xian Xia Road

Huashang Rd => Huashang Road

Zhenxun Rd => Zhenxun Road

Aomen Rd => Aomen Road

Wukang => Wukang Road

Haigang Ave. => Haigang Avenue

Wuchang Ave. => Wuchang Avenue

Huaihai West road Wellington garden => Huaihai West road Wellington Garden

Husong Hwy. => Husong Highway

Xiaoxinqiao Xiang => Xiaoxinqiao Lane

Hong Song East Raod => Hong Song East Road

xingjianlu => Xingjian Road

Wensan West Rode => Wensan West Road

Hangnan Gonglu => Hangnan Parkway

Yuyuan road => Yuyuan Road

Nanjing road => Nanjing Road

Baoshan road => Baoshan Road

Bixiu road => Bixiu Road

Xiang yang south road => Xiang yang south Road

Tianshan road => Tianshan Road

yindu road => yindu Road

## 数据总体概述

本案例用的数据为上海市数据，整个OSM数据下载下来有669M,体量较大，运行样本时的速度还好控制，但是运行整个数据集时会比较慢（本机用时24h+）。因此一些基本的练习还是用样本数据比较好，这样整个练习完成顺畅之后再用整个数据集运行。

运行data.py产生的5个csv文件以及最后由nodes\_tags和ways\_tags合成的tags文件大小分别如下：

select count(\*) from nodes;

3314303

262M

select count(\*) from nodes\_tags;

277788

9.55M

select count(\*) from ways\_nodes;

3916447

94.2M

select count(\*) from ways;

412613

23.4M

select count(\*) from ways\_tags;

889516

select count(\*) from tags;

1167304

37.7M

查看上海市的一些基础设施：

select value,count(\*) from tags where key='amenity' group by value order by count(\*) desc limit 10;

"bicycle\_rental"        "2598"

"school"        "1396"

"parking"        "1350"

"restaurant"        "1275"

"bank"        "602"

"toilets"        "422"

"cafe"        "378"

"fuel"        "378"

"fast\_food"        "363"

"hospital"        "342"

 可以看出上海的自行车租赁设施比较多，其次是学校，有1396所，教育资源很丰富。

上海作为国际大都市，饮食方面也比较多元化

select value, count(\*) from tags where key="cuisine" group by value order by count(\*) desc limit 10;

"chinese"        "164"

"burger"        "52"

"chicken"        "32"

"coffee\_shop"        "32"

"pizza"        "31"

"italian"        "15"

"american"        "14"

"japanese"        "14"

"asian"        "12"

"international"        "12"

结果显示，中餐占最大的比例，但burger和chicken等西餐厅也不少。美式、意式、日式等多元化。

OSM属于开放编辑，有大量的参与者共同绘制出我们看到的地图。

select count(distinct user) from ways;

1408

select count(user),user from ways group by user order by count(user) desc limit 10;

"93092"        "Chen Jia"

"20892"        "xiaotu"

"20373"        "aighes"

"12079"        "XBear"

"11664"        "zzcolin"

"9979"        "jamesks"

"9519"        "duxxa"

"9345"        "u\_kubota"

"8898"        "yangfl"

"8016"        "Stenive"

地图来源：

select count(\*) from tags where key='source';

135888

select value, count(\*),(count(\*)\*1.0/(select count(\*) from tags where key='source') )from tags where key="source" group by value order by count(\*) desc limit 10;

"PGS"        "86782"        "0.638628870834805"

"Bing"        "30324"        "0.223154362416107"

"GPS"        "6824"        "0.0502178264453079"

"bing"        "6608"        "0.0486282821146827"

"wind"        "499"        "0.00367214176380549"

"osm-gpx"        "480"        "0.00353232073472271"

"survey"        "384"        "0.00282585658777817"

"interpolation"        "324"        "0.00238431649593783"

"Yahoo imagery"        "249"        "0.00183239138113741"

"Bing 2010"        "244"        "0.00179559637348405"

 可以看到，’PGS’和’Bing’的数据份额占到了86%。

select count(\*) from tags where key="created\_by";

27810

select value, count(\*),(count(\*)\*1.0/(select count(\*) from tags where key="created\_by")) from tags where key="created\_by" group by value order by count(\*) desc limit 10;

select user,count(\*),(count(\*)/(select count(\*) from nodes/100.0)) from nodes group by user order by count(\*) desc limit 10;

"almien\_coastlines"        "22020"        "0.791801510248112"

"JOSM"        "5539"        "0.199172959367134"

"Potlatch 0.10f"        "64"        "0.00230133045667026"

"Merkaartor 0.12"        "57"        "0.00204962243797195"

"dkt\_GNS-import-1"        "47"        "0.00169003955411722"

"Potlatch 0.9a"        "27"        "0.000970873786407767"

"Merkaartor 0.11"        "13"        "0.000467457749011147"

"Potlatch 0.10e"        "13"        "0.000467457749011147"

"Potlatch 0.4a"        "7"        "0.00025170801869831"

"iLOE 1.8"        "5"        "0.000179791441927364"

98%的编辑是由"almien\_coastlines"和"JOSM"完成的，很少属于人类的编辑。

## 需进一步完善的地方

本次整理只能算是清洗的一小步，后续对于道路名称的详细清洗还需改进；城市的清洗也只进行了上海的部分，其他相邻城市不能简单地以邮编开头两位来区分，因此需要探索更好的方法来清洗数据。

对于OSM地图，这是一个很好的集合众人力量的实例，不过像中国的邮编这种，可以设置一些条件来规范编辑，比如限定6位数，这样能更好的减少人工犯错的机会。但是这样会缺少灵活性，如果用户首次输入出现错误提示，可能会打消其积极性。

另外，对于数据分析的部分，由于贡献度大的用户是电脑或者智能设备，这些相对来说数据有较高的一致性，而贡献很小的用户可能不太熟悉OSM，上传与其他用户重复的点的几率也较大，因此，在用数据进行分析的时候，建议优先取贡献大的用户提供的数据，这样能保证数据的一致性。不过这样也会出现问题，如果某个设备故障，采集的某一批次数据是错误的，而人工的是正确的，那就相当于错失了纠正的机会。