整理 OpenStreetMap 数据

更新信息

相比于上一次提交：

> - *在代码中添加了注释信息；*
> - *修改 data.py 中的 shape_tag 函数，将清理后的数据保存到 csv 文件中；*
> - *在本报告中添加了“关于数据集的其他想法”的内容。*

区域选择

湖北省，武汉市 数据来源：

- http://www.openstreetmap.org/relation/3076268#map=9/30.6696/114.3873
- https://mapzen.com/data/metro-extracts/metro/wuhan_china/

选择武汉的原因是，这是我上大学并现在居住的地方，我想通过本次数据整理项目对这个区域的资料进行整理，为以后进行有趣的探索做准备。 本来我的家乡（山东某小城）也在考虑范围内的，然而是个小城市，数据集太小，不符合要求。

数据集中的问题

做数据审查时，我主要观察了数据集的两个方面，分别是数据类型是否恰当和数据的值是否合理，并拿出改进方法。本工作的代码在`audit.py`文件中。本次数据审查和整理主要针对两类问题：

1. 街道名的缩写问题（例如“Huang Xiao He Rd”）；
2. 邮政编码错误（例如 k="addr:postcode" v="Wuhan Hankou"）

需要说明的是，我选择了中国的城市**武汉**来进行此次项目，所以`addr:street`是中文的，而 Python 2 中的编码问题使问题处理起来变得非常复杂。而数据集中还包含街道的英文名`name:en`，此字段也包含了街道的信息，所以我在审查数据时用的不是`addr:street`而是`name:en`。

街道名的缩写问题

在街道的`name:en`中，有“Huang Xiao He Rd”、“Haohu Ave”等带缩写的街道名称，此处通过匹配，将缩写改为全称：

```
mapping = {
            "jie": "Street",
            "lu": " Road",
            "road": "Road",
            "Bldg": "Building",
            "Ave": "Avenue",
            "Rd": "Road",
            "Lu": "Road",
            "St.": "Street",
            "Str": "Street",
            "Rd.": "Road"
        }
```

经过代码处理：

```
def update_way_names(name, mapping):
    for k, v in mapping.items():
        if k in name:
            name = name.replace(k, mapping[k])
            return name
    return name
```

缩写改为全称的英文：

```
Huang Xiao He Rd -> Huang Xiao He Road
Xiang Gang Rd -> Xiang Gang Road
Haohu Ave -> Haohu Avenue
```

邮政编码错误

通过匹配邮编的格式把错误的邮编排除：

```
def check_postcode(postcode):
    if parse_int(postcode) is None or len(postcode) != 6
or not postcode.startswith('43'):
        return False
    else:
        return True
```

类似 k="addr:postcode" v="Wuhan Hankou" 这样的错误邮编就被剔除了。

数据概览

审查和清理完毕后，用 `csv2db.py` 将各 csv 文件导入 `wuhan_osm.db` SQL 数据库中。

文件大小

- wuhan.osm: 62.2 MB;
- wuhan_osm.db: 44.7 MB;
- nodes.csv: 24.7 MB;
- ways.csv: 1.9 MB;
- nodes_tags.csv: 489 KB;
- ways_tags.csv: 2.43 MB;
- ways_nodes.csv: 9 MB.

Number of nodes

```
SELECT COUNT(*) FROM nodes;
```

结果: 314887

Number of ways

```
SELECT COUNT(*) FROM ways;
```

结果: 34260

Number of unique users

```
SELECT COUNT(DISTINCT(T.uid)) FROM
(SELECT uid FROM nodes UNION ALL
SELECT uid FROM ways) as T;
```

结果: 513

贡献最多的用户 Top 10

```
SELECT T.user, COUNT(*) AS num FROM
(SELECT user FROM nodes UNION ALL SELECT user FROM ways)
as T
GROUP BY T.user
ORDER BY num DESC
LIMIT 10;
```

结果：

```
GeoSUN|112204

Soub|48069

jamesks|24414

Gao xioix|17901

katpatuka|17298

dword1511|13558

samsung galaxy s6|10603

flierfy|5715

hanchao|5289

keepcalmandmapon|5123
```

再来看一下这 10 位顶级贡献者，一共贡献了多少数据：

```
SELECT SUM(NUM.num) FROM
(SELECT T.user, COUNT(*) AS num FROM
    (SELECT user FROM nodes UNION ALL SELECT user FROM
ways) as T
    GROUP BY T.user ORDER BY num DESC LIMIT 10) as NUM;
```

结果：260174

总共的数据为 `314887 + 34260 =349147` ,这 10 位用户贡献了本数据集总量的 74.5% 的数据。

探索性分析

武汉最多的 10 项生活设施是什么，分别有多少？

```
SELECT value, COUNT(*) as num FROM
nodes_tags WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10;
```

结果：

```
restaurant|159
school|152
bank|129
townhall|75
parking|72
fast_food|60
fuel|58
bicycle_parking|35
hospital|32
atm|27
```

以上就是武汉最多的十项生活设施啦，然而最多的 restaurant 居然才 159 个，这数据是值得怀疑的，毕竟武汉这么大，成千上万个 restaurant 也是正常的。导致这个问题的原因可能是数据集不够全面，或者标签不足，没有标出来。

作为吃货，还想知道这些 restaurant 中，类型最多的是什么餐馆。

餐馆类型 TOP 10

```
SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags JOIN
(SELECT DISTINCT(id) FROM nodes_tags WHERE
value='restaurant') as T
ON nodes_tags.id=T.id
WHERE nodes_tags.key='cuisine'
GROUP BY nodes_tags.value
ORDER BY num DESC
LIMIT 10;
```

结果：

```
chinese|14
asian|2
barbecue;chinese|1
burger|1
chinese;american|1
chinese;oriental|1
```

从结果来看，最多的还是我中国菜啦，上榜的还有烤肉、汉堡等类型。当然此类型数据比较少，不具有统计意义。

关于数据集的其他想法

通过本次数据整理项目，我开始意识到了数据清洗工作的繁杂性和重要性。在看视频课程的时候，老师讲数据收集和数据整理过程可能要占整个数据分析流程大约 70% 的时间，这是毫不夸张的。数据整理，要审查数据集的有效性、精度、完整性、一致性、统一性等方面，这是一个繁复可能也有些无聊的任务，却是十分重要的，后面探索性分析结论的准确和可靠与否，都建立在数据整理是否到位之上。

分析数据的额外建议

在数据集中，有很多标签的值为中文，如 `<tag k="addr:city" v="武汉"/>` 、 `<tag k="name" v="汉川市"/>` 等。在这些标签中，其实也有一些数据清洗的工作要做。例如，在 `k="addr:city"` 中，正确的 v 值应该是“武汉”，即 `v="武汉"` ,但在数据集中，有 `v="武汉市"` 、 `v="Wuhan"` 、 `v="唐家墩街道"` 等不统一的问题，在接下来的分析中，有必要对这些问题进行清理。

实施改进的益处

清理值为中文的重要标签，如 `name` ,可以提高数据集的有效性、一致性等，对于数据集的进一步探索分析很重要，如用 SQL 探索数据集的时候，就可以使用这些中文标签作为分类的依据。

预期问题

Python 2 的编码问题导致其处理中文比较复杂，这也是我在清理 `way` 的数据时，选择 `name:en` 代替 `addr:street` 的原因。在数据清理和将 OSM 文件输出到 csv 文件的过程中，都可能导致中文变成编码。所以在接下来的清洗和分析中，我会采用 Python 3 来进行下一步的工作。

我这次数据整理项目还存在一些问题。首先是数据集的完整性不足，跟动辄几个 G 的 osm 文件相比，我这 62 MB 的数据集可以说是非常袖珍了。我对该数据集的审查也可以更加深入，本次整理主要针对街道名称和邮编的问题，可能其他方面还有问题需要后面整理。