# OpenStreetMap Data Wrangling Project

## Map Area

Knoxville, TN, USA

I chose to work with data from Knoxville, as I have famiy in the area and have visited many times.

## Problems Encountered in the Map

Three problems with the data immediately stuck out to me:

- Some street types were not capitalized (i.e., "lane" vs. "Lane")
- Some street types were inconsistently abbreviated, such as "Rd." for "Road"
- Some streets did not have types at all, just names - for example, "Maloneyville"

```
 'Turnpike': {'Oak Ridge Turnpike'},
 'West': {'Main Street West'},
 'lane': {'Bearden View lane'},
 'way': {'university commons way'}}
701: {'701'}
70E: {'Hwy 70E'}
Ave: {'S Illinois Ave'}
Broadway: {'North Broadway'}
Edgemoor: {'Edgemoor'}
Hwy: {'Asheville Hwy'}
lane: {'Bearden View lane'}
Maloneyville: {'Maloneyville'}
Northeast: {'Henrietta Drive Northeast'}
Northwest: {'Palmwood Drive Northwest'}
Parkway: {'Parkway'}
Rd: {'Boatdock Rd'}
Southeast: {'Dandridge Avenue Southeast', 'Langford Avenue Southeast'}
Turnpike: {'Oak Ridge Turnpike'}
way: {'university commons way'}
West: {'Main Street West'}
```

### I chose to address the first two issues using data cleaning methods

Using mapping and the update_name function, I fixed inconsistent abbreviation and capitalization. The update_name function used regular expressions and utilized the re() method.

In [4]:
```
expected = ["Lane", "Street", "Pike", "Avenue", "Circle", "Pass", "Boulevard", "Highway",
            "Freeway", "Drive", "Court", "Place", "Way", "Road"]

mapping = {"St": "Street", "Hwy": "Highway", "lane": "Lane", "Rd": "Road"}
```

In [5]:
```
def update_name(name, mapping):

    m = street_type_re.search(name)
    if m:
        street_type = m.group()
        if street_type not in expected:
            name = re.sub(street_type_re, mapping[street_type], name)

    return name
```

This fixed the problematic street types, such as changing Boatdock Rd to Boatdock Road.

## Exploring the Data

I conducted various database queries using SQLite to get a statistical overview of the data.

### File sizes:

```
Knoxville.osm ..................... 71 MB
Knoxville.db ...................... 1.5 MB
nodes.csv ......................... 522 KB
ways.csv .......................... 37 KB
ways_nodes.csv .................... 164 KB
nodes_tags.csv .................... 23 KB
ways_tags.csv .....................19 BYTES
```

### Number of nodes and ways:

sqlite> SELECT COUNT(*) FROM ways;
Result: 616

sqlite> SELECT COUNT(*) FROM nodes;
Result: 6237

### Number of contributing users:

sqlite> SELECT COUNT(DISTINCT(uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways);

Result: 345

### Number of users having only one post:

sqlite> SELECT COUNT() *FROM (SELECT e.user, COUNT*() as num FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e GROUP BY e.user HAVING num=1) u;

Result: 16

### Top 10 contributing users

sqlite> SELECT e.user, COUNT(*) as num FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e GROUP BY e.user ORDER BY num DESC LIMIT 5;

Result:

```
zebanshee          72
wward              61
AndrewSnow         54
yurasi             42
DaveHansenTiger    26
```

## Additional ideas

The top 5 contributers make up nearly 74% of all contributions (73.9%). This suggests that not many total individuals have contributed to the Knoxville map. I feel that the data would be more complete, robust, and thorough if more people contributed to it.

### Investigative query: recent map changes

I conducted two SQL queries (listed below) to discover when the most recent changes took place on this map. It turns out that the map has not been updated since January of 2019, almost two years ago. This presents some concerns, as the road topography has most likely changed since then. It's very likely new businesses, side streets, driveways, road modifications, etc. have been implemented in that time.

sqlite> SELECT * FROM nodes ORDER BY timestamp DESC LIMIT 1;
Result: 2019-01-14T21:58:17Z

sqlite> SELECT * FROM nodes ORDER BY timestamp DESC LIMIT 1;
Result: 2019-01-09T03:36:53Z

#### Ideas for improvement

The Knoxville map could be improved by having a larger number of contributers and by those contributers continuing to add to the map year after year. A good way to accomplish this would be providing some sort of reward system for contributers. Award badges or virtual rewards for folks that edit the Knoxville map. Maybe even have the city (or an individual area business) provide localized rewards, like discount coupons for goods an services in the area, to Knoxville residents that participate in editing the map.

#### Benefits of improvements

The benefits of doing this would be more complete and up-to-date map data.

#### Anticipated problems of improvements

Drawbacks to incentivizing folks to participate in curating the street map is that more contributers means more errors and typos. This would require data analysts to do more in-depth cleaning to fix the mistakes, resulting in a less-efficient process.

## Conclusion

In conclusion, the Knoxville portion of the OpenStreetMap dataset is very helpful to locals and visitors in the area. However, due to the small amount of contributers to the data set, the information may be outdated and incomplete. With more robust auditing and cleaning, however, it would be very possible to greatly refine the present data and create a more complete map.

## Sites referenced:


http://wwww.w3schools.com/python/
https://docs.python.org/3.9/
https://www.classroom.udacity.com/
https://www.w3schools.com/sql/