Wrangle OpenStreetMap Data Project

by: Renee Cothern

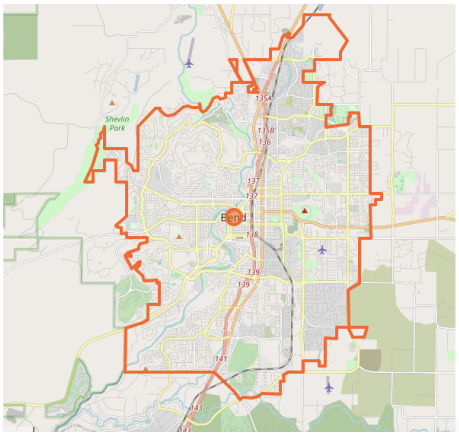
Table of Contents

Map Area

Area investigated: Bend, OR (Central Oregon)

https://www.openstreetmap.org/relation/186761#map=12/44.0613/-121.3153

This is the area I went to High School and College, so I was interested in how it has grown and what new items there are to see.



Problems Encountered in Map

Below shows the different areas I looked at in my data map. I looked through the data first, just to get an idea of the type of issues I would have to deal with.

- **Tags of different types and tag problems** - I will looked at to get at the individual tag in my XML file to get an understanding on how much of which data I can expect. The result shows that the k attributes for the tag fields have no problem characters

  - "lower", for tags that contain only lowercase letters and are valid
  - "lower_colon", for otherwise valid tags with a colon in their names
  - "problemchars", for tags with problematic characters, and
  - "other", for other tags that do not fall into the other three categories.

```
{'lower': 74946, 'lower_colon': 21794, 'other': 1085, 'problemchars': 0}
```

- **Testing for zip codes that don't conform** - I used the following website to find all the valid zip codes for the area. All zip codes conformed.: https://www.zip-codes.com/city/or-bend.asp

- **Testing for wrong entries in "city"**- All information should show the city to be Bend. I found one incorrect entry.

- **Testing for inconsistent Street Names (types and directional information)** - All information should show consistent street types, such as Avenue vs. Ave. Not all the data had consistent street types, though most of it did.

```
{'110': set(['Northwest Crossing Drive #110']),
 'Ave': set(['SW Simpson Ave']),
 'Blvd': set(['NW Riverside Blvd']),
 'Dr': set(['NE Azure Dr', 'SW Upper Terrace Dr']),
 'Pl': set(['NW Prairie Pl']),
 'St': set(['NE Division St', 'NW Union St', 'NW Wall St'])}
NW Wall St => NW Wall Street
NE Division St => NE Division Street
NW Union St => NW Union Street
Northwest Crossing Drive #110 => Northwest Crossing Drive #110
NW Riverside Blvd => NW Riverside Boulevard
SW Simpson Ave => SW Simpson Avenue
SW Upper Terrace Dr => SW Upper Terrace Drive
NE Azure Dr => NE Azure Drive
NW Prairie Pl => NW Prairie Place
```

The function I used to update street names is below, and will ultimately be ran from the the shape element function prior to exporting to csv file:

```
#ran to update the street names to the correct names
def update_name(name, mapping): #individual address, and mapping dictionary above
    found_type = street_type_re.search(name)
    if found_type:
        street_type = found_type.group()
        if street_type in mapping:
            #find the street_type in the mapping file, and put value in new_street_type
            new_street_type = mapping[street_type]
            #find the wrong street type in address, and replace with new
            name = name.replace(street_type, new_street_type)

    return name #return correct address and place in better_name located below
```

In addition, for directional information, all street names should show consistent compass street directional indicators, such as Northwest, instead of NW.

```
('Old street name: ', 'SW Simpson Ave')
('New street name: ', 'Southwest Simpson Ave')
('Old street name: ', 'SE Scott Street')
('New street name: ', 'Southeast Scott Street')
('Old street name: ', 'NE Cushing Drive')
('New street name: ', 'Northeast Cushing Drive')
('Old street name: ', 'NE Cushing Drive')
('New street name: ', 'Northeast Cushing Drive')
('Old street name: ', 'NW Prairie Pl')
('New street name: ', 'Northwest Prairie Pl')
('Old street name: ', 'SW Upper Terrace Dr')
('New street name: ', 'Southwest Upper Terrace Dr')
('Old street name: ', 'NE Division St')
('New street name: ', 'Northeast Division St')
('Old street name: ', 'NE Azure Dr')
('New street name: ', 'Northeast Azure Dr')
('Old street name: ', 'NW Wall St')
('New street name: ', 'Northwest Wall St')
('Old street name: ', 'NW Riverside Blvd')
('New street name: ', 'Northwest Riverside Blvd')
('Old city name: ', 'ch')
('New city name: ', 'Bend')
('Old street name: ', 'NW Union St')
('New street name: ', 'Northwest Union St')
('Old street name: ', 'NE Lancaster Street')
('New street name: ', 'Northeast Lancaster Street')
```

This will be used to update the street directional types, and the one incorrect city entry. This will be ran from the shape element function:

```
def audit_street_compass(elem):
    #osm_file = open(osmfile, "r")
    #for event, elem in ET.iterparse(osm_file, events=("start",)):

        if elem.tag == "node" or elem.tag == "way":
            for tag in elem.iter("tag"):
                if is_street_name(tag):
                    street_name = tag.attrib['v']
                    matches = ["NW", "SW", "NE", "SE"]
                    if any(x in street_name for x in matches):
                        print ("Old street name: ", street_name)
                        street_name = street_name.replace('NW', 'Northwest')
                        street_name = street_name.replace('SW', 'Southwest')
                        street_name = street_name.replace('NE', 'Northeast')
                        street_name = street_name.replace('SE', 'Southeast')
                        print ("New street name: ", street_name)
                #remember the one entry that was bad in 'Testing for wrong entries in "city"'?  Fixing it!
                if is_city(tag):
                    city_name = tag.attrib['v']
                    if 'ch' in city_name:
                        print ("Old city name: ", city_name)
                        city_name = city_name.replace('ch', 'Bend')
                        print ("New city name: ", city_name)
```

Overview of the Data¶

Preparing for Database

The next step was to prepare the data to be inserted into a SQL database. I used a python script that parses the elements in the OSM XML file transforming then to a tabular format, thus making it possible to write to .csv files. These csv files can then easily be imported to a SQL database as tables.

File Sizes

- mapBend2.osm: 58.7MB

Wrangle OpenStreetMap Data Project

by: Renee Cothern

- BendOR.db: 30.5MB
- nodes.csv - 22.8MB
- nodes_tags.csv - 277KB
- ways.csv - 1.9MB
- ways_tags.csv - 3.4MB
- ways_nodes.csv - 7.5MB

Evaluating the Dataset

The next task was to see what was included in the dataset and what could be improved. I imported by database, and started running queries. I used pandas dataframe to format my results in a more readable format.

Number of Nodes and Ways

The below gets some general statistics about how many "nodes" is in this dataset, as well as how many "ways".

```
QUERY = "SELECT count(*) AS num FROM nodes;"
QUERY2 = "SELECT count(*) AS num FROM ways;"
```

```
 Total Nodes
      268214
 Total Ways
      31847
```

Number of Users

The below gets the number of users who contributed to this dataset.

```
QUERY = '''
SELECT COUNT(DISTINCT(nodes.uid))
FROM nodes
LEFT JOIN ways ON nodes.id = ways.id;
'''
```

```
       Total Users Who Contributed
Count                          258
```

Top 10 Contributors

The next information shows the top contributors to creating this dataset.

```
QUERY = '''
SELECT subquery.user, COUNT(*) AS num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways)
AS subquery GROUP BY subquery.user ORDER BY num DESC LIMIT 10;
'''
```

Wrangle OpenStreetMap Data Project

by: Renee Cothern

	User Name	Number of Contributions
1	dkunce	80844
2	beej71	47388
3	Timothy Smith	36106
4	Peter Dobratz	20670
5	btwhite92	12631
6	lemmiwinks	10893
7	michaeldugger	9904
8	emilyeros	8656
9	Jeff Barlow	7659
10	cowdog	4588

Top 10 Node Types and Way Types

I was curious of what type of data was collected. in order to determine that I had to see the top number of nodes collected.

```
QUERY = '''
SELECT key, COUNT(*) AS Count
FROM nodes_tags
GROUP BY key
ORDER BY Count DESC
LIMIT 10;
'''
QUERY2 = '''
SELECT key, COUNT(*) AS Count
FROM ways_tags
GROUP BY key
ORDER BY Count DESC
LIMIT 10;
'''
```

	Node Type	Count
1	highway	1153
2	power	619
3	name	505
4	amenity	435
5	state	336
6	housenumber	322
7	street	318
8	postcode	313
9	city	311
10	phone	268

	Way Type	Count
1	building	20050
2	source	11831
3	highway	9254
4	name	5126
5	surface	4200
6	maxspeed	3084
7	service	2752
8	county	2454
9	cfcc	2451
10	name_base	2095

Building Types

I noticed above that 20,050 of the Way Types had a label of building. I found that to be odd because it would seem like that would be a node. So I decided to investigate.

Wrangle OpenStreetMap Data Project

by: Renee Cothern

```
QUERY = '''
SELECT value, COUNT(*) AS Count
FROM ways_tags
WHERE key="building"
GROUP BY value
ORDER BY Count DESC
LIMIT 10;
'''
```

	Building	Count
1	yes	18998
2	house	424
3	detached	227
4	commercial	124
5	retail	54
6	residential	41
7	apartments	35
8	school	32
9	roof	29
10	garage	21

Tourism

Trying to find places to go.

```
QUERY = '''
SELECT value, COUNT(*) AS Count
FROM nodes_tags
WHERE key="tourism"
GROUP BY value
ORDER BY count DESC
LIMIT 10;
'''
QUERY2 = '''
SELECT value, COUNT(*) as Count
FROM nodes_tags
WHERE key="cuisine"
GROUP BY value
ORDER BY Count
LIMIT 10;
'''
```

	Places to Tour	Count
1	artwork	35
2	information	27
3	picnic_site	22
4	viewpoint	7
5	hotel	6
6	museum	1

	Types of Cuisine	Count
1	american;vegetarian	1
2	breakfast;coffee_shop	1
3	cajun	1
4	chicken	1
5	chinese	1
6	french	1
7	indian	1
8	irish	1
9	juice; vegan; vegetarian	1
10	middle_eastern	1

Wrangle OpenStreetMap Data Project

by: Renee Cothern

Ideas about the Dataset

Ideas

There are **two ways** I can see to improve this dataset. To improve the numbers of viewers and to gain popularity of this resource would be to include fun places to visit. This could range from anything to restaurants, as well as places to tour. This data seems to be lacking in that area as noted from the above query labeled "Tourism". Although, I understand it's a map, the idea is to provide a hook for those to return to the map. The other way to improve it is more from a technical aspect. It seems like the building label (see query labeled: "Top 10 Node Types and Way Types") should be listed under "node" but instead was listed under "way" which is for point-to-point destinations. There were 20,050 building types under "way", and upon further destination, 95% of those were listed with a generic "yes" value. Perhaps because the data is still incomplete. Here is some further information about the "yes"
designation: https://wiki.openstreetmap.org/wiki/Key:building

Anticipated problems with implementing the improvement

Both ideas listed cost time to change. For the first improvement, a contributor would have to really research each building, and determine what it is, and add the necessary keys and values that would reflect further information, which potentially could be done with Google maps. The same problem exists with having to change buildings to move it from a way association to a node. However being that one would have to "start over" it would be an excellent opportunity to really populate some useful information about those buildings.

Conclusions

After reviewing this information, I noticed, despite the large number of contributors, the data was fairly consistent. I did have to make some changes to allow for further consistency, however for the most part, contributors did a good job of making sure the data was complete.

Also it's obvious that the information of the Bend, OR area is incomplete or could use more information of a node's "interest". And, in order to populate a lot of this information would take a large amount of time. I think it's really interesting the amount of time that goes into it, and to sacrifice that level of time is an indicator of a love of an area or a love of just mapping in general, especially to go the extra mile and put in the areas of interest.