OpenStreetMap Data Case Study

Map Area

Tri Cities, Washington, United States

- https://www.openstreetmap.org/export#map=11/46.2426/-119.0142

This map is of a metropolitan area to the north of my hometown, as my hometown was under 50 mb.

Problems Encountered in the Map

After downloading the map and analyzing it, the main issue that I came to find was that of street types having inconsistent names, such as ave vs avenue.

```
#This is used to replace bad street types

mapping = { "Ave": "Avenue",

            "Pl": "Place",

            "St": "Street",

            "St.": "Street",

            "Steet": "Street",

            "ave": "Avenue",

            "Ct": "Court",

            "Dr": "Drive",

            "Blvd": "Boulevard",

            "ST": "Street",

            "Dri": "Drive"

            }
```

```
#Checks to see if the element is a street name

def is_street_name(elem):

    return (elem.attrib['k'] == "addr:street")



#This function will update the street name if necessary to the correct one.

def update_name(name, mapping):

    m = street_type_re.search(name)

    o = street_name_re.search(name)

    if m:

        street_type = m.group()

        if street_type in mapping.keys():

            name = mapping[street_type]

            if o:

                firstname = o.group()
```

Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the MongoDB queries used to gather them, and some additional ideas about the data in context.

File sizes

```
TriCities.xml: 84.7 mb
TriCities.xml.json: 90.1 mb
```

Number of nodes

```
#Get a count of distinct nodes
len(db.StreetData2.distinct("id", {"type" : "node"}))
```

```
1176
```

Number of ways

```
#Get a count of distinct ways
len(db.StreetData2.distinct("id", {"type" : "way"}))
```

```
47753
```

Number of unique users

```
#Get the number of distinct users
len(db.StreetData2.distinct("created.user"))
```

```
468
```

Types of Amenities

```
#What kinds of amenities are there?
db.StreetData2.distinct("amenity", {"type": "node"})
```

```
['Columbia Park East Boat Ramp',
 'atm',
 'bank',
 'bar',
 'bbq',
 'bell',
 'bench',
 'bicycle_parking',
 'bureau_de_change',
 'bus_station',
 'cafe',
 'car_rental',
 'car_sharing',
 'car_wash',
 'charging_station',
 'childcare',
 'clinic',
 'clock',
 'compressed_air',
 'dentist',
 'dojo',
 'drinking_water',
 'fast_food',
 'fire_station',
 'food_court',
 'fountain',
 'fuel',
 'ice_cream',
 'kindergarten',
 'letter_box',
 'library',
 'loading_dock',
 'marketplace',
 'music_school',
 'parking',
 'parking_entrance',
 'parking_space',
 'pharmacy',
 'place_of_worship',
 'police',
 'polling_station',
 'post_box',
 'post_office',
 'prison',
 'pub',
 'public_bookcase',
 'recycling',
 'register_office',
 'restaurant',
 'school',
 'shelter',
 'shower',
 'smoking_area',
 'social_facility',
 'telephone',
 'theatre',
 'toilets',
 'university',
 'vacuum_cleaner',
 'vending_machine',
 'veterinary',
 'waste_basket',
'waste_disposal',
'water_point']
```

In [105]:

List of Distinct Banks

```
#What are the names of different banks in town?
db.StreetData2.distinct("name", {"amenity": "bank"})
```

```
['Bank of America',
 'Bank of Eastern Washington',
 'Bank of the West',
 'Banner Bank',
 'Chase',
 'Columbia Bank',
 'Community First Bank',
 'First Bank',
 'GESA Credit Union',
 'Gesa Credit Union',
 'HAPO Community Credit Union',
 'Hapo Community Credit Union',
 'Hapo Credit Union',
 'KeyBank',
 'Numerica Credit Union',
 'Spokane Teachers Credit Union',
 'Sterling Savings Bank',
 'U.S. Bank',
 'Wells Fargo',
 'Yakima Federal Savings and Loan']
```

Additional Ideas

I saw some issues in the data, where there were different types of data such as nodes and ways. It seems as if there were also the data types “public” “restriction” and “roundabout.” Is this how things are supposed to be? Further exploration is necessary.