Wrangling OpenStreetMap Data

MAP AREA :

Greater Seattle, King county boundary

https://www.openstreetmap.org/export#map=12/47.5979/-122.1549

I have been living in vibrant and diverse area of Greater Seattle from the past 3 years. Seattle and its sister cities in King county offers non-exhausting list of things to do and to explore. This project has provided yet another avenue to explore and know the city through the perspective of real data about the city make up.

CLEANSING THE DATA:

After downloading the OSM XML file for the area chosen and eyeballing it , I was surprised to see the data was much cleaner than I had expected. So I converted it to csv and then used SQLite DB queries to find any inconsistencies in the data.

Auditing Queries:

1> To check for inconsistencies in city name

```
SELECT tags.value, COUNT(*) as count
 FROM (SELECT * FROM nodes_tags  UNION SELECT * FROM ways_tags) tags
 WHERE tags.key='city' GROUP BY tags.value Limit 8
```

It returned the following results

```
"BELLEVUE"   "1"
"Belevue"    "1"
"Bellevue"   "482"
"Bellevue, WA""29"
"Clyde Hill"  "1"
"Hunts Point" "194"
"Issaquah"   "211"
"Issaquah, WA""1"
```

2> To audit the street names

```
SELECT tags.value
FROM (SELECT * FROM nodes_tags
    UNION
    SELECT * FROM ways_tags) tags
WHERE tags.key='street' and tags.value like '% NE'
Limit 5
```

```
"127th PL NE"
"148th Ave NE"
"Bellevue Way NE"
"148th Ave NE"
"107th Avenue NE"
```

3> Checking for the postcodes longer than 5 digit

```
SELECT tags.value
FROM (SELECT * FROM nodes_tags
    UNION
    SELECT * FROM ways_tags) tags
WHERE tags.key='postcode' and length(tags.value)>5
Limit 10
```

```
"98106-1499"
"98004-5903"
"98004-5983"
"98109-5210"
"98027-5305"
"98195-2350"
"98033-7722"
"98052-6088"
"W Lake Sammamish Pkwy Northeast"
"98004-4452"
```

Here are the observations from audit that need cleansing and the steps I took for the same:

Wrangling functions:

1> City names have been written in different formats. Eg: seattle, SEATTLE or Seattle, WA and so on. It will be appropriate to have all versions of a city name in the same format while we run queries with city names.

I used the following function to make all city names consistent . The capwords() function puts all strings in the format “Abcd”

```
def audit_city_name(name):
   if ', WA' or ',WA' in name:
       name = name.rstrip ( ', WA')
   return string.capwords(n ame)
```

2> Street names have the directions in different formats. Eg: NE and Northeast. The ‘mapping’ dictionary below is used in the function to audit the direction part of the street name

```
def audit_street_name( name, mapping):
   m = street_type_re.search( name)
   if m:
       street_type = m.group()
       if street_type  in list(mapping.keys()):
           better_street_type = mapping[street_type]
           name = street_type_re.sub(better_street_type,  name)
   return name
```

3> Most postcodes are valid 5 digit zip codes . Others were in the valid format 12345-6789; In which case the first 5 digits were extracted. Any other format of postcode was marked invalid and later used to prefix invalid postcode with tag ‘fixme’.

```
postcode_format_re= re.compile( r'^\d{5}(?:[-\s]\d{4})?$')

def update_postcode(postcode, invalid =  True):
   """Check if postcode is of standard format 12345 or 12345-6789. If it neither then mark invalid as TRUE. If valid, Return just the first 5 digits"""
   m = postcode_format_re.s earch(postcode)
   if m:
      invalid =  False
       postcode= postcode[: 5]
   return (invalid, postcod e)
```

I used these functions in wrangle.py to wrangle city name, directrion abbrevations and postcode before writing the data to csv.

Once these changes are performed, the data is now ready to be used for analyzing and answering questions about what interests me various aspects of the city.

OVERVIEW OF THE OSM DATA:

FILE SIZES :

map.osm	310.6 MB
osmdb.db	175.1 MB
nodes.csv	112.2 MB
nodes_tags.csv	11.6 MB
ways.csv	9.2 MB
ways_tags.csv	24.4MB
ways_nodes.csv	35.2 MB

NUMBER OF NODES:

SELECT count(*) FROM nodes

1346976

NUMBER OF WAYS:
SELECT count(*) FROM ways

157158

TOP 5 CITIES IN THE DATA:

```
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags
    UNION
    SELECT * FROM ways_tags) tags
WHERE tags.key='city'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 5
```

```
"Seattle"      "86889"
"Kirkland"     "7081"
"Bellevue"     "482"
"Redmond"      "230"
"Issaquah"     "211"
```

THE MOST USED SOURCES :

```
SELECT value, count(*) as count
FROM (SELECT * FROM nodes_tags UNION SELECT * FROM ways_tags) tags
WHERE tags.key = "source"
GROUP BY value
ORDER by count DESC
LIMIT 10
```

```
"King County GIS;data.seattle.gov"   "75175"
"King County GIS"                    "27549"
"data.seattle.gov"                   "23650"
"bing"                               "5813"
"Bing"                               "3522"
"PGS"                                "1843"
"Yahoo_wms"                          "1768"
"tiger_import_dch_v0.6_20070830"     "1751"
"SDOT Bike Rack Import 2012"         "1740"
"data.seattle.gov;King County GIS"   "474"
```

TOP 5 LEISURE:

```
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags
    UNION ALL
    SELECT * FROM ways_tags) tags
WHERE tags.key='leisure'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 5
```

```
"pitch"             "744"
"Park"              "581"
"Playground"        "325"
"swimming_pool"     "222"
"garden"            "142"
```

The result said the top leisure is “pitch”. This is in sync with my observation around the city that people here are very much into active and fit lifestyle and sports makes a big part of it. I dig in further to see what exactly “pitch” means. I do a SELF JOIN to get a sample detail about “pitch”

```
SELECT tags.id, tags.key, tags.value, nodes_tags.*
FROM (SELECT * FROM nodes_tags WHERE key='leisure' and value = 'pitch' limit 3) tags JOIN nodes_tags
ON tags.id=nodes_tags.id
```

```
"1655859302" "leisure"     "pitch"  "1655859302" "leisure"    "pitch"        "regular"
"1655859302" "leisure"     "pitch"  "1655859302" "sport"      "tennis"       "regular"
"1655859302" "leisure"     "pitch"  "1655859302" "surface"    "asphalt"      "regular"
"2459904692" "leisure"     "pitch"  "2459904692" "leisure"    "pitch"        "regular"
"2459904692" "leisure"     "pitch"  "2459904692" "sport"      "Rock Wall"    "regular"
"3581745093" "leisure"     "pitch"  "3581745093" "leisure"    "pitch"        "regular"
"3581745093" "leisure"     "pitch"  "3581745093" "name"       "Seattle Center Skatepark"    "regular"
"3581745093" "leisure"     "pitch"  "3581745093" "sport"      "skateboard"   "regular"
```

So this shows any infrastructure needed for various sports is tagged as pitch. May it be tennis or skateboarding.

FIRST 3 ENTRIES IN ways :

```
SELECT id, user, uid, timestamp FROM ways
ORDER BY timestamp
LIMIT 3
```

```
"4921383"     "btb"              "10608"      "2007-07-19T01:37:46Z"
"6316942"     "DaveHansenTiger"  "7168" "2007-09-17T02:56:08Z"
"6317315"     "DaveHansenTiger"  "7168" "2007-09-17T02:57:46Z"
```

First entry was in july 2007, which seems a little unexpected to me, as the wikipedia page says the OSM was launched in 2004

ADDITIONAL IDEAS:

The OSM XML data tags had many(396 to be specific) keys called ‘fixme’. Few of the values for such ‘fixme’ keys were as follows:

"resurvey"
"bad address range"
"addresses missing buildings"
"What's in this site now?"
"verify alley turns here"
"Where does this road come from?"
"Survey needed to see where this road ends. Aerial obscured by motorways."
"Was this moved to the new MOHAI, or to their storage/library? It is obviously not here now the building is demolished."
"Location is approximate"

The data that is ambiguous needs to be handled in a more systematic way. The ‘fixme’ tag values need to be categorized into predetermined options which will facilitate their easy update in future. For Eg: A category called word “Resurvey” can accommodate the entire “Survey needed to see where the road ends”.

This will make it easy to come up with a strategy to resolve a single kind of ‘fixme’ tag and apply the same strategy to all the similar ‘fixme’ tags.

The downside to it could be the loss of details stating the exact problem, but this can be more appropriately documented in a separate key value pair called ‘comments’ .

CONCLUSION:

Analysing the OSM data is a great learning experience and also can be very useful to know a city better. As an example, I was amazed to know that the city amenity that stood second after the ‘car-parking’ was ‘bike-parking’ ! It tells a lot about the culture and vibe of the city. I am eager to explore some more cities and see how it stands in comparison to Seattle in various aspects.