

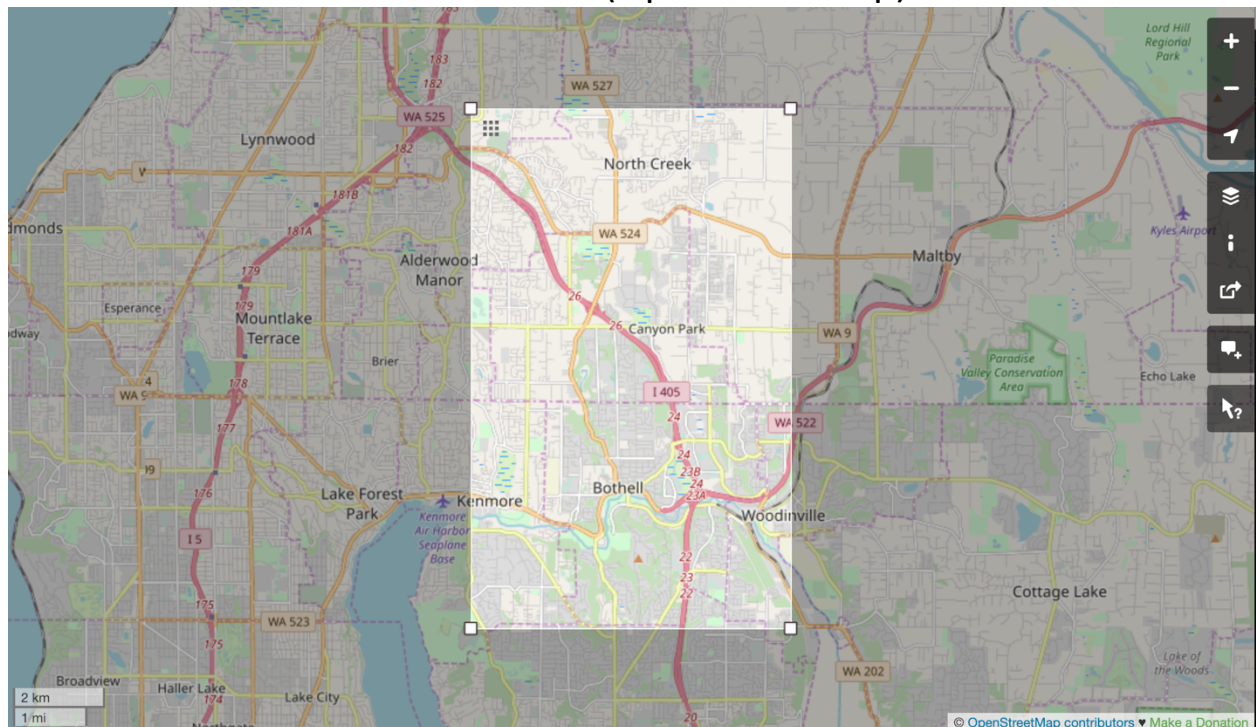
OpenStreetMap Case Study

Map Area

I looked at Bothell, WA and some area surrounding it. The latitude and longitudinal coordinates of my area of interest is as shown below.

	47.8337	
-122.2493		-122.1566
	47.7324	

The area of focus as shown below (OpenStreetMap):



I recently shifted to Canyon Park in Bothell, WA and wanted to explore the area. This project gave me a unique opportunity to do so while I have the ability to return my improvements to the OpenStreetMap community.

Problems Encountered in the Map

Looking through the map file I downloaded, I noticed three main improvements I wished to make. I shall discuss them in the following order:

- Street names were over-abbreviated
 - *228th St SE Suite 1, 19115 112th Ave NE, Bothell-Everett Hwy*
- Postal codes were in a few cases were 9 digits rather than 5.
 - *9802133, 98012-7134*
- Phone numbers were not uniform.
 - *+1 425 485 1363, 425 398 8900, (425) 492-1911*

Street names that were over-abbreviated

I audited the street names with the help of the `audit_street_names.py` which uses regular expressions to populate the uncommon types of street names. However, for the updating I did a word by word check in the street name and updated it rather than just looking at the last words.

```
def update_name(name, mapping):
    namearray=name.split(" ")
    newname=[]
    for element in namearray:
        for key in mapping.keys():
            if key == element:
                element=element.replace(key,mapping[key])
        newname.append(element)

    return " ".join(newname)
```

This resulted in updating street names like *228th St SE Suite 1* to *228th Street Southeast Suite 1*.

Postal codes uniformity.

Postal codes were mostly five digits but there were a few inconsistent to that format. These were audited and updated using the `audit_zipcode.py` . I removed all the trailing characters beyond the length of 5 characters thereby successfully updating zipcodes like *98012-7134* to *98012*.

I imported the files in to a database and looked at the counts for each postcode after ordering them using the following command:

```
select unified.value, count(*) as count
from
  (select * from nodes_tags
   union all
   select * from ways_tags) unified
where unified.key="postcode"
group by unified.value
order by count desc;
```

Here is the result beginning with the highest count:

Postcode	Count
98011	6785
98021	4641
98034	1361
98012	212
98072	204

Postcode	Count
98028	68
98503	2
98041	1
98125	1

These zipcodes are expected for this area ¹.

Non-uniformity of phone numbers

The phone numbers were not uniform. Some of them began with “+1” but some only had the ten digits. Their formatting was also not uniform, with some hyphenated, some bracketed, etc. I decided to update all phone numbers as “+1-XXX-XXX-XXXX”. This was done using the following update code:

```
def update_phone(phone):
    phonetmp=phone.replace(" ", "")
    phone=phonetmp.replace("(", "")
    phonetmp=phone.replace(")", "")
    phone=phonetmp.replace("-", "")

    if len(phone)==10:
        phone="+1"+phone
    elif len(phone)==11:
        if phone.startswith("+"):
            phone=phone.replace("+", "+1")
        else:
            phone=""+phone

    newphone=""
    for i,char in enumerate(phone):
        if i==2 or i==5 or i==8:
            newphone=newphone+"-"
            newphone+=char
        else:
            newphone=newphone+char

    return(newphone)
```

This resulted in updating the phone numbers like +4254814844 to +1-425-481-4844.

Data Overview and Additional Ideas

File sizes:

map.osm	50 MB
output	51 MB
db_map.db	28 MB
MyMapNodes.csv	19 MB
MyMapNodeTags.csv	597 KB
MyMapWays.csv	1.7 MB
MyMapWaysTags.csv	3.4 MB
MyMapWaysNodes.csv	5.5 MB

Number of nodes:

```
sqlite> select count(*) from nodes;
```

222853

Number of ways:

```
sqlite> select count(*) from ways;
```

26977

Number of unique users:

```
sqlite> select count(distinct(users.uid))  
from (select uid from nodes  
      union all  
      select uid from ways) users;
```

400

Top 10 contributing users:

```
sqlite> select users.user, count(*) as num  
from (select user from nodes  
      union all  
      select user from ways) users  
group by users.user  
order by num desc  
limit 10;
```

User	Number of contributions
patricknoll_import	170811
seattlefyi_import	10454
bdp	9774
STBrenden	9287
Natfoot	7959
compdude	4174
"patrick noll"	3432
zephyr	1752
Glassman	1742
LostSparky	1579

The other end of the spectrum are individuals who have contributed a single post.

```
sqlite> select count(*) from
...> (select users.user, count(*) as num
...> from (select user from nodes
...>        union all
...>        select user from ways) users
...> group by users.user
...> having num=1) count1user;
```

84

Additional Data Exploration

Amenities that are more than 5 in count

```
sqlite> select value, count(*) as num
...> from nodes_tags
...> where key="amenity"
...> group by value
...> having num > 5
...> order by num desc;
```

Amenity	Count
restaurant	40
fast_food	17
school	17
cafe	15
bank	9
fuel	9
bench	7
place_of_worship	6

The restaurants and fast-food take the top ranks.

Cuisines of the restaurants and fast-food with more than 2 in count

```
sqlite> select nodes_tags.value, count(*) as num
...> from nodes_tags
...>   join (select distinct(id) from nodes_tags
...>         where value="restaurant"
...>         or value="fast_food") eateries
...> on nodes_tags.id = eateries.id
...> where nodes_tags.key = "cuisine"
...> group by nodes_tags.value
...> having num >= 2
...> order by num desc;
```

Cuisine	Count
mexican	6
pizza	6
burger	5
thai	5
japanese	3
sandwich	3
asian	2
teriyaki	2

Religion in places of worship

```
sqlite> select nodes_tags.value, count(*) as num
...> from nodes_tags
...>   join (select distinct(id) from nodes_tags
...>         where value="place_of_worship")
...>         worshipplaces
...> on nodes_tags.id = worshipplaces.id
...> where nodes_tags.key = "religion"
...> group by nodes_tags.value
...> order by num desc;
```

Only 5 out of 6 of the places of worship had a religion tag associated and they were all “christian”. However, I know that there is Hindu place of worship in the area and is not included as a node in this map.

Other improvements

One of the main improvements that comes to my mind is a review rating like that found in Google Maps as a tag in the data. Another improvement would be to extract more node information from other sources.

Conclusion

From the review of this area of Bothell and surrounding areas, it is clear that the data associated is incomplete. I understand the data has been sourced and entered by users. This area probably needs more data sources and users to enter and update the information. Considering the availability of sources like Google Maps, Yelp, etc, the effort might not be worth it.

References

1 <https://www.zipcodestogo.com/Washington/>

