# P3: Wrangle OpenStreetMap Data

Ying Wu

Map area: Santa Cruz, California, United States
https://s3.amazonaws.com/metro-extracts.mapzen.com/santa-cruz_california.osm.bz2

This is a nice beach area near where I grew up.

## Problems encountered in your map

During the audit I found a couple of issues:

- Postal codes sometimes had CA prefixed to the number and sometimes had 4 trailing digits expected: “95066” found: “CA 95065” and “95066-5121”
- Some have addr:street but are missing addr:housenumber or vice versa.
- Some have whole address (including house number) in addr:street field
- Abbreviation for Highway 9 (Hwy 9, Hwy. 9)
- Some keys were prefixed with “tiger:” and seemed to be in a different format

This dataset did not have any street names with directional abbreviation (N/S/E/W)

The code I used to create the json will fix the zip code and Highway issues. For the keys that started with “tiger:” I took the zip codes (“tiger:zip_left”) and converted those into “address:postcode”. (See attached python file)

## Overview of the data

Commands

```
mongoimport --db test --collection city --drop --file santa-cruz_california.json
mongo

> # within mongo shell
> use test
> sc = db.city
```

### Size of the file

```
santa-cruz_california.osm is 54800278 bytes (53MB)
santa-cruz_california.json is 61974898 bytes (60MB)
```

### Number of unique authors

```
> sc.distinct("created.uid").length
401
```

### Number of nodes and ways

```
> sc.find({"type": "node"}).count()
257661
> sc.find({"type": "way"}).count()
20954
```

### Total number of entries

```
> sc.find().count()
278695
```

### Number of different hiking trails by difficulty

Hiking trails are described here: http://wiki.openstreetmap.org/wiki/Key:sac_scale

```
python
>>> import pymongo
>>> client = pymongo.MongoClient("mongodb://localhost:27017")
>>> sc = client.test.city

>>> pipeline = [{"$match": {"sac_scale": {"$ne": None}}},
                       {"$group": {"_id": "$sac_scale", "count": {"$sum": 1}}},
                       {"$sort": {"count": -1}}] # only 6 possible so no limits
>>> result = sc.aggregate(pipeline)
>>> pprint.pprint(list(result))
[{u'_id': u'hiking', u'count': 87},
 {u'_id': u'mountain_hiking', u'count': 48},
 {u'_id': u'alpine_hiking', u'count': 1}]
```

### Top nature sites

```
>>> pipeline = [{"$match": {"natural": {"$exists": 1}}},
            {"$group": {"_id": "$natural", "count": {"$sum": 1}}},
            {"$sort": {"count": -1}},
            {"$limit": 10}]
>>> result = sc.aggregate(pipeline)
>>> pprint.pprint(list(result))
[{u'_id': u'tree', u'count': 827},
 {u'_id': u'wood', u'count': 220},
 {u'_id': u'water', u'count': 86},
 {u'_id': u'cliff', u'count': 55},
 {u'_id': u'beach', u'count': 50},
 {u'_id': u'coastline', u'count': 39},
 {u'_id': u'sand', u'count': 25},
 {u'_id': u'peak', u'count': 19},
 {u'_id': u'wetland', u'count': 17},
 {u'_id': u'spring', u'count': 10}]
```

## Other ideas about the datasets

When I was going through the OSM documentation, one thing that struck me as odd was that latitude / longitude is captured but elevation is not.  There is an entry on the wiki on this subject: http://wiki.openstreetmap.org/wiki/Altitude

```
>>> sc.find({"ele" : {"$exists": 1}}).count()
290
```

Since very few elements in the mapping data for this city have elevation information and often times cities are quite flat, adding this information for more elements could be an area of improvement.

Since Santa Cruz is along the coast, there are many scenic areas to visit. It could be interesting to have a 'vista' locations recorded signifying places ideal for scenic views / picture taking. I found that this is already implemented using the 'tourism' tag with the value of 'viewpoint', however, even though Santa Cruz is a pretty tourist friendly city, there are only 38 viewpoint elements in the OSM dataset. One way to populate additional viewpoints in an automated fashion could be to leverage the fact that often times GPS data is available as metadata for pictures posted online. By aggregating lat/long of places where people tend to take pictures, additional points of interest could be added to the OSM project. Of course, one must implement some quality control measures to avoid undesirable results such as filtering out concert venues and limiting multiple similar locations from one author.

**References**

https://wiki.openstreetmap.org/wiki/OSM_XML

http://wiki.openstreetmap.org/wiki/TIGER

https://wiki.openstreetmap.org/wiki/Key:tourism

(and many more pages from that wiki)

https://docs.mongodb.org/manual/reference/mongo-shell/

https://docs.python.org/2/library/collections.html