OpenStreetMap San Francisco Project
Data Wrangling with MongoDB

Jennifer Tsou

Map Area: San Francisco, CA, USA

https://mapzen.com/data/metro-extracts/your-extracts/99d97e82a282

1. Problems Encountered in the Map

- After downloading the San Francisco map and converting it to sample size. I ran it against a provisional data.py file, and against tags.py to see if there are any potential problems, I noticed a few some problems.

1. Street names are abbreviated (st.)
2. Inconsistent postal code - some addresses do not have postal codes.

ABBREVIATED STREET ISSUE.

1. **Once I identified all the abbreviated ones, I converted them. (Missouri St => Missouri Street)**
   I decided to fix the street name by iterate each word into a dictionary to fix the inconsistency.

```
def update_name(name, mapping):
    after = []
    # Split name string to test each part of the name;
    # Replacements may come anywhere in the name.
    for part in name.split(" "):
        # Check each part of the name against the keys in the correction dict
        if part in mapping.keys():
            # If exists in dict, overwrite that part of the name with the dict value for it.
            part = mapping[part]
        # Assemble each corrected piece of the name back together.
        after.append(part)
    # Return all pieces of the name as a string joined by a space.
    return " ".join(after)

    return name
```

POSTAL CODE ISSUE

Some address lack postal code or have inconsistent format. Therefore, I decided to clean the format a lot, and then group them together based on the addresses that have postcode only.

```
def update_postcode(postcode):
    if re.match(r'$\d{5}^', postcode):
        return postcode
    try:
        return re.findall(r'^(\d{5})-\d{4}$', postcode)[0]
    except:
        pass
```

#Sort postcodes by count, descending

sf.aggregate([{"$match":{"address.postcode":{"$exists":1}}},{"$group":{"_id":"$address.postcode", "count":{"$sum":1}}}, {"$sort":{"count":-1}},{'$limit':5}])

```
{ "_id" : "94117", "count" : 4 }
{ "_id" : "94103", "count" : 3 }
{ "_id" : "94107", "count" : 2 }
{ "_id" : "94102", "count" : 2 }
{ "_id" : "94110", "count" : 2 }
```

This result struck me that if the highest count is 4, then we are lacking zip code severely. So I decided to get rid of LIMIT and reran the program... I am right.

```
>
sf.aggregate( [{"$match":{"address.postcode":{"$exists":1}}},       {"$group":{"_id":"$address.postcode",
"count":{"$sum":1}}},      {"$sort":{"count":1}}])
{ "_id" : "94103-3124", "count" : 1 }
{ "_id" : "94158", "count" : 1 }
{ "_id" : "94115", "count" : 1 }
{ "_id" : "94111", "count" : 1 }
{ "_id" : "94107", "count" : 2 }
{ "_id" : "94110", "count" : 2 }
{ "_id" : "94102", "count" : 2 }
{ "_id" : "94103", "count" : 3 }
{ "_id" : "94117", "count" : 4 }
```

Maybe the count of zip codes are so small because there are not a lot of address in the area. This is when I decided to see how many addresses are in the area, to see if I'm right. There were plethora of addresses.

#Sort address by count, descending

sf.aggregate([{"$match":{"address ":{"$exists":1}}},

{"$group":{"_id":"$address ", "count":{"$sum":1}}},

{"$sort":{"count":-1}}])

```
{ "_id" : { "housenumber" : "648" }, "count" : 1 }
{ "_id" : { "street" : "3rd Street", "housenumber" : "590",
"postcode" : "94107" }, "count" : 1 }
{ "_id" : { "street" : "Brannan Street", "housenumber" : "274" },
"count" : 1 }
{ "_id" : { "street" : "Montgomery Street", "housenumber" : "1200" },
"count" : 1 }
{ "_id" : { "street" : "16th Street", "housenumber" : "3149" },
"count" : 1 }
{ "_id" : { "street" : "Valencia Street", "housenumber" : "260" },
"count" : 1 }
{ "_id" : { "street" : "16th Street", "housenumber" : "3121" },
"count" : 1 }
{ "_id" : { "city" : "San Francisco", "street" : "Divisadero Street",
"housenumber" : "298" }, "count" : 1 }
{ "_id" : { "street" : "14th Street", "housenumber" : "494" },
"count" : 1 }
{ "_id" : { "street" : "Mariposa Street", "housenumber" : "2424",
"postcode" : "94110" }, "count" : 1 }
{ "_id" : { "city" : "San Francisco", "country" : "US", "state" : "CA",
"street" : "4th Street", "postcode" : "94103-3124", "housenumber" :
"70" }, "count" : 1 }
{ "_id" : { "city" : "San Francisco", "state" : "CA", "street" : "4th
Street", "housenumber" : "22", "postcode" : "94103" }, "count" : 1 }
{ "_id" : { "street" : "Haight Street", "housenumber" : "1601",
"postcode" : "94117" }, "count" : 1 }
{ "_id" : { "city" : "San Francisco", "state" : "CA", "street" :
"Bryant Street", "housenumber" : "1600", "postcode" : "94103" },
"count" : 1 }
{ "_id" : { "city" : "San Francisco", "street" : "4th Street",
"housenumber" : "960", "postcode" : "94158" }, "count" : 1 }
{ "_id" : { "street" : "Sacramento Street", "housenumber" : "500" }, "count" : 1 }
{ "_id" : { "street" : "Haight Street", "housenumber" : "530" }, "count" : 1 }
{ "_id" : { "city" : "San Francisco", "street" : "Columbus Avenue", "housenumber" : "155" },
"count" : 1 }
{ "_id" : { "housenumber" : "490" }, "count" : 1 }
{ "_id" : { "city" : "San Francisco", "street" : "King Street", "housenumber" : "298",
"postcode" : "94107" }, "count" : 1 }
Type "it" for more
```


2. Data Overview

This section contains information about dataset and MongoDB queries used to gather them.

FILE SIZES

san-francisco_california.osm 319.2MB

sanfran.osm.json 46.5MB

#Number of documents

```
> sf.find().count()
26975
```

#Number of nodes

```
> sf.find({"type":"node"}).count()
26975
```

#Number of ways

```
> sf.find( { type: "way" } ).count()
0
```

#Number of Unique Users (265 users have edited the map)

```
> sf.distinct("created.user").length
265
```

#Top 1 contributor

```
> sf.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":1}])
{ "_id" : "KindredCoda", "count" : 12819 }
```

 # Number of users appearing only once (having 1 post)

```
{ "_id" : 1, "num_users" : 87 }
```

ADDITIONAL DATA EXPLORATION

#Top 10 appearing amenity

```
 sf.aggregate([{'$match':{'amenity':{'$exists':1}}},{'$group':{'_id':'$a
menity','count':{'$sum':1}}},{'$sort':{'count':-1}},{'$limit':10}])
{ "_id" : "post_box", "count" : 50 }
{ "_id" : "pub", "count" : 26 }
{ "_id" : "restaurant", "count" : 13 }
{ "_id" : "bicycle_parking", "count" : 8 }
{ "_id" : "cafe", "count" : 7 }
{ "_id" : "parking", "count" : 4 }
{ "_id" : "fuel", "count" : 4 }
{ "_id" : "telephone", "count" : 4 }
{ "_id" : "toilets", "count" : 3 }
{ "_id" : "bar", "count" : 3 }
```

#TOP 10 APPEARING SHOPS

```
{ "_id" : "bicycle", "count" : 2 }
{ "_id" : "wine", "count" : 1 }
{ "_id" : "dry_cleaning", "count" : 1 }
{ "_id" : "clothes", "count" : 1 }
{ "_id" : "laundry", "count" : 1 }
{ "_id" : "outdoor", "count" : 1 }
{ "_id" : "hairdresser", "count" : 1 }
{ "_id" : "hardware", "count" : 1 }
{ "_id" : "supermarket", "count" : 1 }
```

3. Conclusion

After the review of the data, I've determined that the data is not complete. San Francisco is a big city, and to think that there is only 1 supermarket in the top 10 appearing shops alerted me that something is missing. I believe that my data is thoroughly cleaned and the analysis is done right for the project. However, the dataset can be improved by setting some guideline when users want to contribute to the map, such as upload in consistent format and have the pre-check mechanism like "incorrect format" or "incomplete information". The benefit of such audit would be to maintain a more complete dataset and prevent incomplete information, which makes the analysis hard. Although this implementation would make the life of analyst easier, it might impact the contributors' willingness to participate in the Open Street Map community. Sometimes people just do not have complete information to contribute.