Problems encountered in my map

・Name inconsistency(Starbucks Coffee,Starbucks)

When I try to count cafe by descending order in mongoDB, I found that there are two different name for Starbucks Coffee.(Result is below).This result is of sample part of the map,so the number is small.

```
>db.sf.aggregate([{"$match":{"cuisine":{"$exists":1},"amenity":"cafe"}},
{"$group":{"_id":"$name","count":{"$sum":1}}},{"$sort":{"count":-1}}])

{ "_id" : "Starbucks Coffee", "count" : 4 }
{ "_id" : "Starbucks", "count" : 3 }
{ "_id" : "Espresso Roma", "count" : 1 }
{ "_id" : "Sun Maxim's", "count" : 1 }
...
```

The name inconsistency will result in the diffrent outcome, so I adjust "Starbucks" to "Starbucks Coffee". The result is below.

```
>db.sf_r.aggregate([{"$match":{"cuisine":{"$exists":1},"amenity":"cafe"}},
{"$group":{"_id":"$name","count":{"$sum":1}}},
{"$sort":{"count":-1}},{"$limit":10}] )

{ "_id" : "Starbucks Coffee", "count" : 66 }
{ "_id" : "Peet's Coffee & Tea", "count" : 13 }
{ "_id" : "Peet's Coffee and Tea", "count" : 5 }
{ "_id" : "Philz Coffee", "count" : 5 }
{ "_id" : "Peet's Coffee", "count" : 4 }
{ "_id" : "Quickly", "count" : 3 }
{ "_id" : "Beanery", "count" : 3 }
{ "_id" : "Highwire Coffee Roasters", "count" : 2 }
```

{ "_id" : "Tart to Tart", "count" : 2 }
{ "_id" : "Blue Bottle Coffee", "count" : 2 }

"Starbucks Coffee" and "Starbucks" are now only "Starbucks Coffee" and there's no more "Starbucks".

・Zip code inconsistency

Sorting the post code in descending order,I found that the post code includes Oakland that is not in SF.

>db.sf_r.aggregate([{"$match":{"address.postcode":{"$exists":1}}},{"$group":{"_id":"$address.postcode","count":{"$sum":1}}},{"$sort":{"count":-1}}])

{ "_id" : "94122", "count" : 4580 }
{ "_id" : "94611", "count" : 2982 }
{ "_id" : "94116", "count" : 2022 }
{ "_id" : "94610", "count" : 1349 }
{ "_id" : "94133", "count" : 1056 }
{ "_id" : "94117", "count" : 992 }
{ "_id" : "94127", "count" : 597 }
{ "_id" : "94103", "count" : 506 }
{ "_id" : "94109", "count" : 441 }
{ "_id" : "94063", "count" : 381 }
{ "_id" : "94587", "count" : 257 }
{ "_id" : "94114", "count" : 200 }
{ "_id" : "94061", "count" : 169 }
{ "_id" : "94110", "count" : 165 }
{ "_id" : "94102", "count" : 158 }
{ "_id" : "94123", "count" : 130 }
{ "_id" : "94108", "count" : 125 }

{ "_id" : "94131", "count" : 123 }
{ "_id" : "94105", "count" : 112 }
{ "_id" : "94501", "count" : 112 }

The post code starts with “941” is SF's post code. However, “946..” is Oakland's.
“940.." is Redwood City’s and “945..” is Alameda County's.
OpenStreetMap doesn’t classify the post codes perfectly.

・Zip code formatting inconsistency

Zip codes in the data also have problems in inconsistency. Some postal codes have state characters (e.g. "CA94122" "ca94611" and so on), have long ones (such as 94611-2111 or 946115555) or have erroneous ones (not 5 digits such as 1232).
Therefore, I removed state-characters and then, I kept the 5-digit zip code programmatically. (e.g. "CA94122" is "94122", "94611-2111" is "94611" and"946115555" is "94611". I removed the erroneous ones.) After the cleaning all of the zip codes are 5-digits (result is below).

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

{ "_id" : "94038", "count" : 1 }
{ "_id" : "94130", "count" : 1 }
{ "_id" : "94013", "count" : 1 }
{ "_id" : "90214", "count" : 1 }
{ "_id" : "94017", "count" : 1 }

Overview of the data

File Sizes
san-francisco_california.osm.....952MB
san-francisco_california.osm.json.....1.08 GB

#Number of documents
> db.sf_r.find().count()
4974618

#Number of unique users
> db.sf_r.distinct("created.user").length
2202

#Number of nodes
> db.sf_r.find({"type":"node"}).count()
4470070

#Number of ways
> db.sf_r.find({"type":"way"}).count()
504548

#Number of distinct amenities
> db.sf_r.distinct("amenity").length
170

Other ideas about the datasets

・Who contributed the most to make OpenStreetMap(OSM) in SF.

>db.sf_r.aggregate([{"$group":{"_id":"$created.user","count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":1}])

{ "_id" : "ediyes", "count" : 944015 }

“ediyes” has contributed the most to make OSM and the contribution is quite high (19.0%)

Top 3 users’ contribution account for 41.7% of OSM in SF and
top 10 user’s contribution account for 63.6%.

According to the data above, small number of member makes the most of OSM. The more people join the OSM project, the more erroneous data will be added. However, if many people join the project, errors will be likely to be modified. Many people have smart phones or smart watches that equip GPS. It means that they have potential to join OSM project. Giving proper incentive may result in the increase of active users. To keep the distance or calorie consumption will be possible since those equipment have GPS. If badges or rewards were presented depending on the distance they walk or the place where nobody walked before, it could attract more people.

Conclusion

OpenStreetMap are edited by human so it includes errors, typos and inconsistency. If more people join the project, the quality and quantity of data will enhance. So far, we need to audit and develop plan for cleaning and then, we need to write some codes to clean when we plan to utilize this map.