Project 3

1. **Data issues**:
   a. Street names differ - addr:street
   b. Zip codes differ - addr:postcode
   c. addr:state

- After downloading an Open Street Map provided area of South Lake Tahoe I've run some data correction checks against the data, below are some of the data quality issues that I've uncovered.
   a. Street names have different endings. For example, 'Market St' instead of 'Market Street' or 'Heybourne Rd' instead of 'Heybourne Road'.
      i. 'St': set(['Market St'])
      ii. 'Rd': set(['Hawkins Ranch Rd', 'Heybourne Rd', 'River Ranch Rd'])
      iii. 'Ln': set(['Library Ln'])
      iv. 'Blvd': set(['7081 N. Lake Blvd', 'Lake Tahoe Blvd'])
      v. '27.4': set(['Hwy 80 PM 27.4'])

   All of the above are updated to match the majority, which in this case are full names. Additional, minor issue with the street names is that Highway is sometimes added to the end sometimes to the middle or beginning of the word.

   b. Some of the Zip codes have state abbreviations attached. The majority of the Zip codes don't, therefore I assume that the below are incorrect and remove the state abbreviations
      i. CA 96143: 2
      ii. CA 96161: 4
      iii. NV 89511: 2
      iv. NV 89413: 2
      v. NV 89449: 2
      vi. NV 89511: 2
      vii. NV 89701: 2
      viii. NV 89703: 2
      ix. NV 89706: 2
      x. RG6 1LT: 2 <= does not look like a US Zip code at all, after short Googling found that this is a UK zip code, therefore I will update this value to NA
         1. Zip codes were cleaned programmatically by re.findall(r'\d{5}', postcode) which finds 5 digits, if it does not I set the postcode to NA.
   c. Lastly, some of the state names are abbreviated and some are written in full
      i. CA: 354
      ii. California: 2
      iii. Nevada: 2
      iv. NV: 248

Since the majority of the names are abbreviated I will update the full names to be abbreviated as well

- **File description**:

a. Size:

```
south-lake-tahoe_california.osm   circa 116 mb
south_taho.json                   circa 132 mb
```

b. Number of nodes: 1168946

```
collection.find({"type":"node"}).count()
```

c. Number of ways: 72086

```
collection.find({"type":"way"}).count()
```

d. Number of unique users: 601

```
len(collection.distinct("created.uid"))
```

e. Top 2 users:

nmixter: 207054, theangrytomato: 152952

```
collection.aggregate([
        {"$group":{"_id":"$created.user", "count":{"$sum":1} } },
        {"$sort":{"count":-1} },
        {"$limit":2} ] )
```

f. Top restaurants by type:

```
i.   [{u'_id': u'american', u'count': 10},
ii.   {u'_id': u'pizza', u'count': 8},
iii.  {u'_id': u'mexican', u'count': 8},
iv.   {u'_id': u'thai', u'count': 6},
v.    {u'_id': u'sushi', u'count': 4}]
```

```
collection.aggregate([
            {"$match":{"amenity":{"$exists":1}, "amenity":"restaurant",
"cuisine":{"$exists":1} } },
            {"$group":{"_id":"$cuisine", "count":{"$sum":1} } },
            {"$sort":{"count":-1} },
            {"$limit":5} ] )
```

g. Top banks:

```
i.   [{u'_id': u'Bank of America', u'count': 8},
```

ii. {u'_id': u'El Dorado Savings Bank', u'count': 4},
iii. {u'_id': u'U.S. Bank', u'count': 2},
iv. {u'_id': u'Nevada State Bank', u'count': 2},
v. {u'_id': u'Wells Fargo', u'count': 2}]

```
collection.aggregate([

          {"$match":{"amenity":{"$exists":1}, "amenity":"bank",
"name":{"$exists":1} } },

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

          {"$sort":{"count":-1} },

          {"$limit":5} ] )
```

- **Other stats from the data set**:
  a. Top 5 amenities are:
     i. Parking: 728
     ii. School: 192
     iii. Toilet: 152
     iv. Restaurant: 128
     v. Post office: 100
  b. Top 3 fast food places:
     i. Taco Bel: 6
     ii. McDonald’s: 4
     iii. KFC: 4
     iv. Subway: 4
     v. Quiznos: 2
  c. Top 1 user, as mentioned above is nmixter and his contribution is 16.6% of total contributions, however the user who has added the most amenities is theangrytomato (202) which is 11.3% of total amenities

```
collection.aggregate([

          {"$match":{"amenity":{"$exists":1} } },

          {"$group":{"_id": "$created.user", "count": {"$sum": 1}}},

          {"$project": {"ratio": {"$divide" :["$count",collection.find({ "amenity":{"$exists":1}
}).count()] },

                                   "count_":"$count"} },

          {"$sort": {"ratio": -1} },

          {"$limit": 1} ])
```

Conclusion

South Lake Tahoe data, has quite a few inconsistencies, taking into account that I've only analyzed few attributes and in all was able to find correction points. I think working with OpenstreetMap data would be quite a challenge, at least in the data processing phase, as there is a lot of improvement to be done.

One of the path for data improvement, would be to demand users to review potentially inconsistent data. For example, before a user submits his data to openstreetmap she would be asked to review one big set of tag attributes and mark some of them as potentially incorrect. If N amount of users do that, the attribute could be removed.

The idea, if implemented could create a N-eye principle, meaning that multiple people would be able to spot potential inconsistencies. However, in order for it to work there should be an acceptably large number of people and as I've noticed during the project, the majority of contribution comes from circa 20 people, thus an extra work load would land on their shoulders, which could potentially discourage them from contributing further. In that case, another solution could be to ask new users review the contributions before letting them contributing, as in this case they would gain experience and a better understanding.

I think there are potentially many useful projects that would benefit from openstreetmap data. I would to build an app which would generate interesting and fun running paths in Amsterdam. Openstreetmap data contains a lot of information about surrounding areas, therefore one could build the paths based on:

- Distance from car/ bicycle traffic
- Existing running trails
- Locations near outside training facilities
- Path with in incline
- Forest trail
- Etc.

One potential issue would be the data validity and/ or availability. For example, outside training facilities, or road incline or forest trails might not be mapped at all, therefore extra effort would need to be put in in order to generate this information. In addition, existing running trail might not be up to date or not valid due to existing road works.

In addition, moving outside Amsterdam and the Randstad area (Amsterdam, The Hague and Rotterdam) the density of population decreases significantly, therefore other areas might have very little information available about points of interest.