Problems encountered in your map
The first thing I explored in the data downloaded was any potential problems found in the dataset. During my initial audit of the dataset, I found the following issues:

- Inconsistent street names: Different abbreviations, and different formatting meant that many streets that should be the same i.e.: Street, St, St. were all being categorized separately.
- Inconsistent Zip Codes: Most zip codes were consistent but there were some that were just marked as “Disneyland”, some that began with the CA- prefix, and some that had the 4-digit extension at the end.

To address the street issue, I first put together a mapping of the abbreviations that would need to be updated, and what they should be updated to:

```
mapping = { "St": "Street",
            "St.": "Street",
            "Rd." : "Road",
            "Ave" : "Avenue",
            "Dr" : "Drive",
            "Dr." : "Drive",
            "Pkwy" : "Parkway",
            "Pkwy." : "Parkway",
            "Rd" : "Road",
            "Av" : "Avenue",
            "Ave." : "Avenue",
            "Blvd" : "Boulevard",
            "Blvd." : "Boulevard",
            "Ln." : "Lane",
            "WAY" : "Way"
            }
```

Once I had created this dictionary of mappings, I used a function to update the street names to their non-abbreviated versions.

I noticed the zip code issue after I had imported the data into MongoDB and run some initial queries. To address this issue, I created a function that would strip the “CA” off of the beginning of any zip codes, or strip off any trailing 4-digit extensions.

Overview of the data
I first investigated the size of the files:

orangecounty.osm	197.80MB
orangecounty.osm.json	212.60MB

Next, how many unique users exist in the dataset:

```
len(db.distinct("created.user"))
869
```

How many nodes?

```
	db.find({"type":"node"}).count()
	795519
```

How many ways?

```
	db.find({"type":"way"}).count()
	108177
```

Here is the query that was used to audit the zip codes:

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

Top 10 amenities:

```
pprint.pprint(list(db.aggregate([{"$match" :  {"amenity" : {"$exists" : 1}}}, \
		{"$group" : {"_id" : "$amenity", "count" : {"$sum" : 1}}}, \
		{"$sort" : {"count" : -1}}, \
		{"$limit" : 10}])))
```

```
[{u'_id': u'parking', u'count': 1061},  {u'_id': u'school', u'count'
: 640},  {u'_id': u'restaurant', u'count': 522},  {u'_id': u'fountai
n', u'count': 410},  {u'_id': u'fast_food', u'count': 361},  {u'_id'
: u'bench', u'count': 289},  {u'_id': u'place_of_worship', u'count':
 275},  {u'_id': u'toilets', u'count': 271},  {u'_id': u'fuel', u'co
unt': 189},  {u'_id': u'drinking_water', u'count': 188}]
```

Top 10 leisure:

```
pprint.pprint(list(db.aggregate([{"$match" : {"leisure" : {"$exists" : 1}}}, \
		{"$group" : {"_id" : "$leisure", "count" : {"$sum" : 1}}}, \
		{"$sort" : {"count" : -1}}, \
		{"$limit" : 10}])))
```

```
[{u'_id': u'garden', u'count': 1745},  {u'_id': u'park', u'count': 8
64},  {u'_id': u'pitch', u'count': 839},  {u'_id': u'swimming_pool',
 u'count': 273},  {u'_id': u'playground', u'count': 144},  {u'_id':
u'golf_course', u'count': 33},  {u'_id': u'picnic_table', u'count':
30},  {u'_id': u'sports_centre', u'count': 29},  {u'_id': u'track',
u'count': 21},  {u'_id': u'court', u'count': 15}]
```

Top 10 office types:

```
pprint.pprint(list(db.aggregate([{"$match" : {"building" : {"$exists" : 1},
				"building" : "commercial"}}, \
		{"$group" : {"_id" : "$office", "count" : {"$sum" : 1}}}, \
		{"$sort" : {"count" : -1}}, \
		{"$limit" : 10}])))
```

```
[{u'_id': None, u'count': 1744}, {u'_id': u'company', u'count': 1}]
```

This was disappointing, because it looks like not many of the commercial buildings have been tagged in exactly what type of office that they are.

How many sports facilities:

```
pprint.pprint(list(db.aggregate([{"$match" : {"sport" : {"$exists" : 1}}}, \
                    {"$group" : {"_id" : "$sport", "count" : {"$sum" : 1}}}, \
                    {"$sort" : {"count" : -1}}])))
```

```
[{u'_id': u'baseball', u'count': 360},  {u'_id': u'tennis', u'count'
: 258},  {u'_id': u'basketball', u'count': 114},  {u'_id': u'swimmin
g', u'count': 72},  {u'_id': u'soccer', u'count': 52},  {u'_id': u'a
merican_football', u'count': 20},  {u'_id': u'volleyball', u'count':
 19},  {u'_id': u'skateboard', u'count': 9},  {u'_id': u'equestrian'
, u'count': 8},  {u'_id': u'beachvolleyball', u'count': 6},  {u'_id'
: u'athletics', u'count': 5},  {u'_id': u'golf', u'count': 5},  {u'_
id': u'multi', u'count': 5},  {u'_id': u'billiards', u'count': 5},
{u'_id': u'football', u'count': 4},  {u'_id': u'hockey', u'count': 4
},  {u'_id': u'bowls', u'count': 3},  {u'_id': u'archery', u'count':
 3},  {u'_id': u'10pin', u'count': 2},  {u'_id': u'canoe', u'count':
 2},  {u'_id': u'tenns', u'count': 1},  {u'_id': u'horseshoe', u'cou
nt': 1},  {u'_id': u'bowling', u'count': 1},  {u'_id': u'horse_racin
g', u'count': 1},  {u'_id': u'diving', u'count': 1},  {u'_id': u'ska
ting;soccer;basketball', u'count': 1},  {u'_id': u'tennis;basketball
', u'count': 1},  {u'_id': u'handball', u'count': 1},  {u'_id': u'so
ftball', u'count': 1},  {u'_id': u'racquet', u'count': 1},  {u'_id':
 u'skating', u'count': 1},  {u'_id': u'basketball;volleyball', u'cou
nt': 1}]
```

Other ideas about the dataset:

As populated as this area is, it looks like the data is not very clean, and is still missing a lot of data. While doing queries on the different tags I noticed many "yes" entries, which indicated that there was not a more specific tag applied. One interesting project that I could see this data being used for is combining it with census population data for a certain zip codes or cities. You could compare the population and size of certain areas (zip codes or cities) and compare it to the amount of city provided amenities (parks, fire departments, walkways, etc.) to ensure that each population and area was provided enough amenities. You could probably look at the average amount of amenities for population, and then get a good idea of if it's enough by comparing each area to the average, and seeing if it is above or below. Another interesting idea would be to add an additional tag to the "sport" tag to indicate if the area was free / public to use or if it was a private facility that you must pay. This would create an interesting way to search the map and you wanted to see where you could go to play that sport. This would require significant user contribution and would probably have to be updated frequently as business came and went out of business. Because the granular data in the map is already so sparse, this seems like it would probably not be very successful.