Data Wrangle OpenStreetMaps Data

Alexander Baranov

Map Area: Nizhniy Novgorod, Russia

https://s3.amazonaws.com/metro-extracts.mapzen.com/nizhniy-novgorod_russia.osm.bz2

1. Problems Encountered in the Map

In general, I did not meet issues that I would expect to meet. There is no tricky abbreviation as it happens in other areas, no issues with postal code etc. Main issue I should fix was different order in street name. In Russian, street names 'улица Петрова' and 'Петрова улица' are the same street while for program they are definitely different. Therefore, I needed to use one of options and all names in another should be translated into it. I did it in update_name(name) function of convertToJson module.

2. Data Overview

File sizes

Filename	File size, MB
nizhniy-novgorod_russia.osm	184
nizhniy-novgorod_russia.osm.json	261
nizhniy-novgorod_russia_sample.osm	18,5
nizhniy-novgorod_russia_sample.osm.json	26,1

Number of documents

```
> db['nizhniy-novgorod_russia'].find().count()
```

952266

Number of nodes

```
> db['nizhniy-novgorod_russia'].find({"type":"node"}).count()
```

807533

Number of ways

```
> db['nizhniy-novgorod_russia'].find({"type":"way"}).count()
```

144482

Number of unique users

```
> db['nizhniy-novgorod_russia'].distinct("created.user").length
```

352

Top 1 contributing user

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

```
{ "_id" : "Kato Kontenta", "count" : 347569 }
```

Number of users appearing only once (having 1 post)

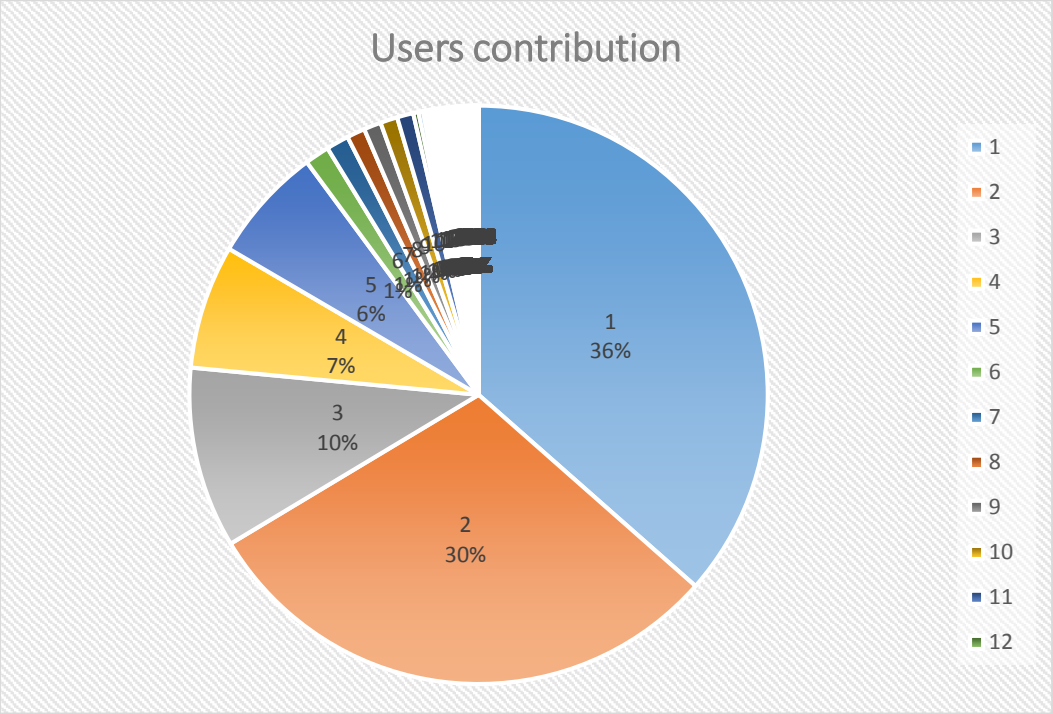
```
> db['nizhniy-novgorod_russia'].aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}}, {"$group":{"_id":"$count", "num_users":{"$sum":1}}}, {"$sort" : {"_id" : 1}}, {"$limit":1}])
```

```
{ "_id" : 1, "num_users" : 65 }
```

3. Additional Ideas

Users contribution

There are two main users accounted for 2/3 of contributions. 5 top users accounted for 89 % of contributions. The rest are minor contributors entered from one to few records.



Top 10 amenities

> db['nizhniy-novgorod_russia'].aggregate([{"$match":{"amenity":{"$exists":1}}}, {"$group":{"_id":"$amenity","count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":10}])

{ "_id" : "parking", "count" : 473 }

{ "_id" : "bench", "count" : 328 }

{ "_id" : "waste_basket", "count" : 276 }

{ "_id" : "kindergarten", "count" : 263 }

{ "_id" : "cafe", "count" : 206 }

{ "_id" : "waste_disposal", "count" : 172 }

{ "_id" : "school", "count" : 169 }

{ "_id" : "bank", "count" : 154 }

{ "_id" : "pharmacy", "count" : 150 }

{ "_id" : "fuel", "count" : 140 }

Chart of cuisines

> db['nizhniy-novgorod_russia'].aggregate([{"$match":{"amenity":{"$exists":1}, "amenity":"restaurant"}},{"$group":{"_id":"$cuisine","count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":10}])

{ "_id" : null, "count" : 64 }

{ "_id" : "japanese", "count" : 5 }

{ "_id" : "english", "count" : 1 }

{ "_id" : "italian", "count" : 1 }

{ "_id" : "european;japanese", "count" : 1 }

{ "_id" : "pizza", "count" : 1 }

{ "_id" : "russian", "count" : 1 }

{ "_id" : "ukrainian", "count" : 1 }

{ "_id" : "uzbek", "count" : 1 }

Conclusion

I have controversial feeling about the data. On one hand, entered data are cleaned enough. I did not spend much effort to fix them and to start using them. On the other hand, the data contains too much information that I would count as garbage like information about each bench and each wastebasket. It looks like local people entered information for local people which local people know already why I would like to see information from local people to tourists. However, this information is missed mostly. For instance most of restaurants (64 / 76) do not have cuisine field entered. There is still much work to enter needed information.