

Open Street Map Data Wrangling Project

Analysis of Lagny-Sur-Marne

The location that I chose to do the data wrangling project on was Lagny-Sur-Marne, a suburban location near Paris that some of my cousins, uncle and aunt live at.

Initial Look At Data

The first part of the Open Street Map (OSM) cleanup was to figure out what the tags were that we were working with. The exact numbers are shown below:

```
{'bounds': 1,  
  'member': 68230,  
  'meta': 1,  
  'nd': 382851,  
  'node': 279358,  
  'note': 1,  
  'osm': 1,  
  'relation': 393,  
  'tag': 122717,  
  'way': 44687}
```

This initial focus is important to figure out the different parts that the XML file from OSM contains. Looking through the documentation on <https://wiki.openstreetmap.org/wiki/Tags> shows that there are three major tags – nodes, ways, and relations.

Some initial checks that we can do before we start to clean the data is to look through and see what type of data we are working with. Within nodes and ways we can see that there are quite a few tags that people are contributing that make up the entire OSM file that we are going through. What do some of these tags look like?

```
{'lower': 118901, 'lower_colon': 3284, 'other': 532, 'problemchars': 0}
```


In the code, we went through and divided the tags into different parts – all lowercase, which is a simple surface-level tag about the node / way /relation, semicolon, which is a tag that has a subcategory, problematic, which has some punctuation we don't recognize, and other, which does not fall into any of these categories.

Since the “lower” and “lower_colon” (tags with semicolons) were so dominantly present in the dataset, I spent my time figuring out how to clean those rather than the content that would have to be more carefully looked over by hand and fixed.

Contribution of Users

```
set(['103253',  
    '1057876',  
    '10610',  
    '1065637',  
    '107257',  
    '1075986',  
    '1082660',  
    '1088559',  
    '10927',  
    '112548',  
    '11699',  
    '118014',  
    '118021',  
    '1186317',  
    '1198089',  
    '1200216',  
    '12022',  
    '1208699',  
    '1234481'...
```

Next, I looked over the number of unique users that had updated the data. To note, the number of unique users were around 356. They all ended up developing a file that was the size...

 OSM Lagny

6/30/2017 12:20 AM File

68,395 KB

Wow! 70 MB's large! There were almost a million lines of text in that file. Either one person did the lion's share of the contribution or each person contributed around 3,000 lines of nodes, ways, or relations to the data. There must be quite some content to be reviewed here because of the size of the content since the project must become unwieldy at a certain point before the data capped out at 70 MB's.

Cleaning Street Names

```
{'1940': set(['Place de l'Appel du 18 Juin 1940']),  
'34': set(['av. Thibault de Champagne, RN 34']),  
'Bellezane': set(['Rue de Bellezane']),  
'Bizeau': set(['Quai Bizeau']),  
'Bonnet': set(['Avenue Bonnet']),  
'Bordes': set(['Rue des Bordes']),  
'Briarde': set(['Avenue de la Ferme Briarde']),  
'Buffard': set(['Rue Edouard Buffard']),  
'Camus': set(['Avenue Albert Camus']),  
'Chabanneaux': set(['Avenue Chabanneaux']),  
'Champagne': set(['Avenue Thibaud de Champagne']),  
'Chantrennes': set(['Rue des Chantrennes']),  
u'Charbonni\xe8re': set([u'Rue de la Charbonni\xe8re']),  
u'Charon': set([u'All\xe9e du Clos Charon']),  
'Charpentier': set(['Boulevard Charpentier']),  
u'Ch\xe2teau': set([u'Rue du Ch\xe2teau']),  
u'Claie': set([u'All\xe9e du Pr\xe9 Claire']),  
'Clemenceau': set(['Avenue Georges Clemenceau']),  
u'Cocher': set([u'All\xe9e du Buisson Cocher'])...
```

The data we are working with here is in French, so there are a few street names that don't compare exactly to English. The code provided had a regular expression script that was supposed to pick out only names that we don't recognize, but seemed to be caught up with certain aspects of street names that I wasn't sure to interpret. For the others, however, we were able to find a few names that were lowercase when they weren't supposed to be (rue vs. Rue), abbreviated (av. vs. Avenue), and that was pretty much the only problematic ones that I found.

Querying the Data

Cursory Statistics

Time for one of the most interesting parts – finding out more about the data!

First, a rough count of the different statistics about the dataset.

```
db.OSMLagny.find().count()
```

Using MongoDB, we can see that the number of entries is around 324,045...

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

The number of users is around 337...

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

The number of nodes is around 279,355...

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

...and the number of ways is around 44,679.

Types of Amenities

Parking Structures

Time to find out more about Lagny.

What are the types of buildings and structures in Lagny?

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

First things first, we query MongoDB for all amenities in the area and sort them from highest count to lowest count. Here is a list of the top amenities in Lagny-Sur-Marne:

```
[{'_id': 'parking', 'count': 158},
 {'_id': 'bench', 'count': 62},
 {'_id': 'waste_basket', 'count': 58},
 {'_id': 'school', 'count': 51},
 {'_id': 'restaurant', 'count': 45},
 {'_id': 'toilets', 'count': 23},
 {'_id': 'fast_food', 'count': 22},
 {'_id': 'place_of_worship', 'count': 21},
 {'_id': 'post_box', 'count': 19},
 {'_id': 'pharmacy', 'count': 19}]
```

Apparently the most prominent structures in Lagny have to do with parking and benches. Interesting results for a suburb in France, where public transportation and walking can get you pretty much anywhere. How many cars should be able to fit in this location?

```
return list(db.OSMLagny.aggregate([{"$match":{"amenity":{"$exists":1}, "amenity":"parking"}},
 {"$group":{"_id":"parking", "count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":5}]))
```

Let's pull the top five types of parking structures to find out.

```
[{'_id': 'surface', 'count': 104},
 {'_id': None, 'count': 47},
 {'_id': 'underground', 'count': 4},
 {'_id': 'multi-storey', 'count': 3}]
```

Only three types of structures! Interesting. And only seven of them seem to be bigger than single-level parking. Okay, that makes more sense. Lagny may be pretty heavy on parking, but it doesn't have multi-level parking everywhere. If that was the case, then Lagny might be more likely classified as urban than suburban.

Schools

What type of a town is Lagny? Is it a place for families, or is it a university town where students go to get degrees in higher education? Let's take a look at the type of schools available to us.

```
return list(db.OSMLagny.aggregate([{"$match":{"amenity":{"$exists":1}, "amenity":"school"}},
 {"$group":{"_id":"school", "count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":5}]))
```

This query grabs the top five types of schools, which are:

```
[{'_id': None, 'count': 18},
 {'_id': {'FR': 'l\'mentaire'}, 'count': 13},
 {'_id': {'FR': 'lyc'}, 'count': 8},
 {'_id': {'FR': 'maternelle'}, 'count': 7},
 {'_id': {'FR': 'coll\ge'}, 'count': 5}]
```

The “None” category here is pretty disappointing, as it would have been nice to have one of the other school types at the top. The rest of the query, however, is nice. There is a heavy influence of the élémentaire school at 13 compared to the lycée, maternelle, and collège schools. For non-French speakers, that is elementary schools at 13, high school at 8, preschool at 7, and middle school at 5. It seems that Lagny focuses more on the basic schools for younger kids than the schools for older kids.

Restaurants

Finally, we have the type of restaurants at Lagny for a unique look into the lives of the people that live there. What are the most popular types of restaurants there?

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

This takes the top five restaurants, below:

```
[{'_id': None, 'count': 25}, {'_id': 'burger', 'count': 5}, {'_id': 'french', 'count': 4}, {'_id': 'asian', 'count': 3}, {'_id': 'seafood', 'count': 2}]
```

Interesting! Restaurants with burgers are the top counted, with French food being second to burgers. It seems that the stereotypical notion of French food being prevalent in France may not be the case here.

Possible Additions to Data Set

There seems to be quite a few “None” categories that appear on the amenities I dug deeper into. This could probably be fixed by scraping the Google Maps API by feeding the position of the nodes that have these characteristics and placing the results programmatically back into the database.

Potential positive aspects of pulling from Google Maps:

1. The data would be thoroughly cleaned by Google Maps already.
2. The data would probably be more complete than Open Street Maps.
3. Pulling the data would be automated.

Potential negative aspects of pulling from Google Maps:

1. The data coming in might not be readable to those that implement it as it may be in French.
2. Since the data is coming from the French suburb, it may not be any better than the Open Street Maps data.
3. Google Maps API's might not have the same position for the nodes as Open Street Maps

Conclusion

After looking through the queries, Open Street Maps has a pretty good rendition of what Lagny-Sur-Marne is like. Though there may be many parking structure, Lagny has the typical characteristics of a suburb – plenty of single-level parking, schools for children, and a small number of restaurants to eat at. Contrary to what might be thought of French people, the people of Lagny seem to have an interest in burger spots or Lagny as a whole may have an oversupply of them.

The data could always be improved through the use of Google Maps APIs, as long as the information is available and clean. Someone could do a great benefit to Open Street Maps by producing an automated function that supplies nodes with updated characteristics so that the description of the places in any location in the world has a better profile than before.