PROJECT WRANGLE OPEN STREET MAP DATA

ANNECY AREA (FRANCE)

Table of Contents

1. INTRODUCTION

Project “wrangle Open Street Map data” intends to practice data wrangling in a large dataset.

We will first proceed to **data acquisition**. For this, we need to understand how data is built, by whom and with what guidelines.
We will need to master various tools to extract data and gather it in a file that is

- Accurate enough : that we can explore with a reasonable confidence ;
- Easy to query ;
- Complete but not to heavy (just the data needed, nothing too complicated)

In a second step, we will **clean dataset**, that is identify main errors or problems and replace them with valid data, in a cleaned dataset for further exploration.
This cleaning must be automated as much as possible, to make our query reproducible: it would be a shame to clean a big dataset, with no possibility to adapt tool created (python file) to other queries.
Also, as this is an iterative process : errors are found by groups, beginning on small sized dataset : hence the need for an automated process.

Third phase consists in **exploring dataset**, and, as for me it is the funniest part !

2. DATA ACQUISITION

2.1 OSM OPEN STREET MAP – HOW IS IT BUILT ?

Dataset is extracted from Open Street Map project.
OSM intends to create a free editable map of the world with information gathered on voluntary basis. Project was created by Steve Coast, in the UK in 2004. Since then more than six millions users had contributed to populate information until May 2020 (source Wikipedia)

Volontary contribution can lead to unequal quality of information.
However quality is said to be good, for most places in the world.

Indeed Open street map is organized so that data is organized : contributions can be traced to their contributer and features are guided in a limited number of keys.

OSM is structured by elements:

.

- **nodes** (points in space surface defined by its latitude and longitude and node id) ;

Example:
Id 14726846
Latitude 45.8680353
Longitude 6.0610645

.

- **ways** (linear features and area boundaries) A way is an ordered list of between 2 and 2,000 nodes that define an open line or a loop ; each way has a list of referenced nodes, defined via `<nd ref=""/>` attributes:
-

Examples : a street, with nodes being shops, or houses

- **relations** (relations between nodes, or ways)
Examples : administrative boundaries, hiking or cycling routes..
Nodes rel

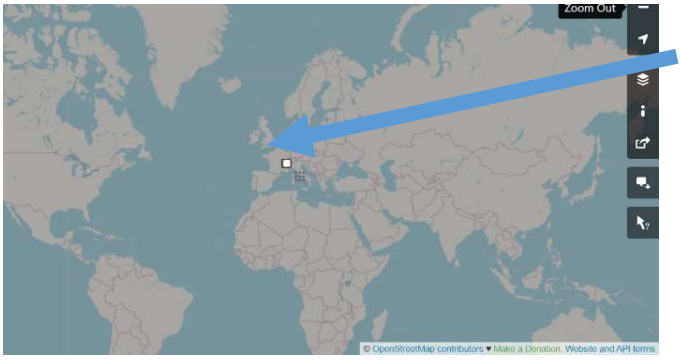
Nodes, ways and relations can have tags. Tags are detailed information about elements. They are made of a key and a value.

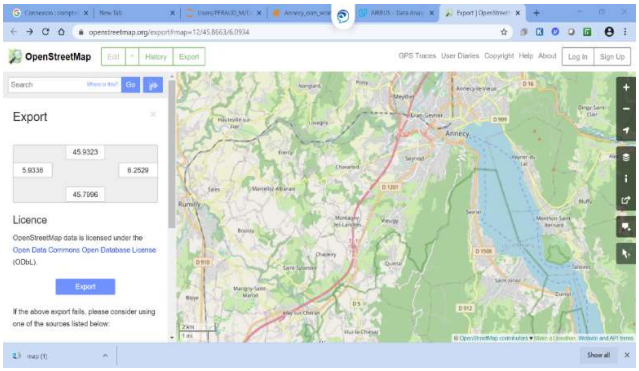
OpenStreetMap's free tagging system allows the map to include an unlimited number of attributes describing each feature. There is a guideline (https://wiki.openstreetmap.org/wiki/Map_Features) about tags that can be used. Each user is free to create new tags and use existing tags.

2.2 ANNECY IN FRANCE DATA ACQUISITION

In order to acquire Annecy dataset, we connected to address below :
https://www.openstreetmap.org/#map=5/51.500/-0.100

We chose Annecy area : Annecy is a town near Switzerland border in the Alps. Part of my family leaves there. It's a beautiful area, with a lake near ski resorts and full of history.

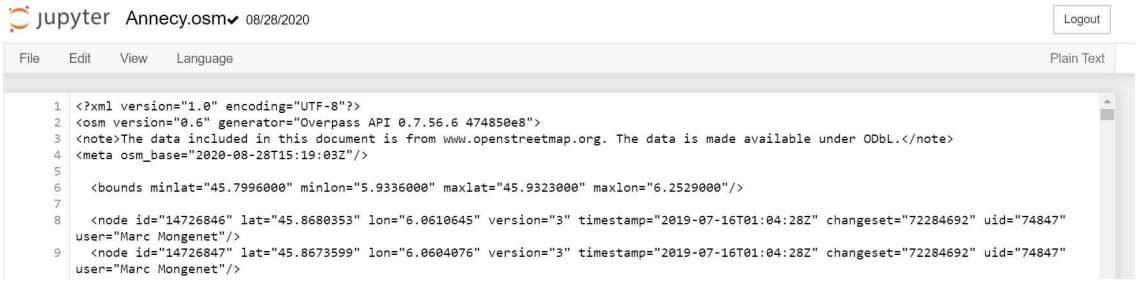




We then used Overpass API tool to export data to an .os Annecy.osm.

Information about latitude and longitude is as follows:
minlat="45.7996000" minlon="5.9336000" maxlat="45.9323000" maxlon="6.2529000"
size is 152Mb, according to Overpass API tool.

No that we have Annecy.osm file, we wanted first to check that file was the one we expected : look onto first lines of the file in jupyter looks good : it's a xml file, encoded in Unicode Latitude and longitude references are those we are expecting.



Via a python query, we checked size, number of elements per type, and number of users:

```
Size of Annecy.osm file is 152 Mb, so higher than 50Mb requested

Number of tags per type
                0
osm         1
note        1
meta        1
bounds      1
node        614821
tag         322728
way         91637
nd          807410
relation    1049
member      45585

926 users until August 28th 2020
```

Size of data set seems large enough to get significant information.
Python query can be used over any osm dataset.Thus, it is re-usable.

There seems to be “ready made” python modules to explore xml files, called osmium. I will try to install it and run it when I have time because it seems powerful, to investigate osm files. There seems to be hundreds of library in python that can save time for users.

3. DATASET CLEANING

3.1 STREET TYPE NAMES CLEANING

We focused on cleaning street type names.
French street names are beginning with streetname type,
Rue Lafayette or Allées Wilson etc.
In another node, you can find housenumber
Decoding of Annecy street names has to be a little different from that of US street names.
When iterating on the query, we will see that there are many valid possible entries other than rue (=street).

```
"Routes","Maison","ZA","ZI","Palais","Parc","Angon","Escaliers","Le","L
es","Lieudit","Chemin","Esplanade","Faubourg","Passage","Pont","Port","
Rue",
"Boulevard","Place","Allée","Avenue","Impasse","Col","Côte","Quai","Ram
pe","Route","Promenade","Square","Voie"
```

Here are first lines of list of wrong entries (wrong case or over abbreviated)

```
'13': {'13 Rue JEAN MOULIN'},
 'Georges': {'Georges Paccard'},
 'allée': {'allée des Fournais', 'allée des gentianes'},
 'avenue': {"avenue d'Aix Les Bains",
            "avenue d'Aix les Bains",
            'avenue de France',
            'avenue de johnathan',
            'avenue de la Mavéria',
            'avenue des trois fontaines'},
 'chemin': {'chemin des Glaisiers', 'chemin de Bellevue', 'chemin de be
llevue'},
```

And first lines for modification proposal (full data in file Annecy_osm_wrangling.ipynb

```
**QUICK CHECK OF MODIFICATIONS PROPOSAL**
0
rue du Pré Faucon       Rue du Pré Faucon
rue des écoles          Rue des écoles
rue Sommeiller          Rue Sommeiller
rue des Glières         Rue des Glières
rue Cassiopée           Rue Cassiopée
rue de la préfecture    Rue de la préfecture
rue du Président Favre  Rue du Président Favre
rue du Vieux Moulin     Rue du Vieux Moulin
rue de la Garr          Rue de la Garr
rue Henry Bordeaux      Rue Henry Bordeaux
rue du Travail          Rue du Travail
```

13 and Georges are not changed at this stage
13 should go in another tag (housenumber) and name of the street type should be added to street_nametag at a later stage; Lower case is replaced and abbreviation fg is replaced by Faubourg.
If users are not paying attention to street names, can we consider dataset as reliable ? It depends on what we are looking for. Where I live, I use open street map for recommended itineraries by bicycle, so street names are not so important for me. What is more important is what I can find on my way (shops, parks...).

What we would like to check is quality of features: check that tags attributes are in limited number and correspond to what is recommended so that, If we query on two different areas, we will find equivalent contents for equivalent tags. Attributes.

Comparing list of existing “shops” information, we can find 97 different entries, among which 5 times “Shop =yes”
Looking content of “shop= yes” nodes, that is not a referenced node, it seems that it could be replaced by sport (2808761221) or by confectionery (5317939707)

```
0   (2808761221, city, Talloires)
```

```
1   (2808761221, housenumber, 39)
2   (2808761221, name, Les passagers du vent)
3   (2808761221, postcode, 74290)
4   (2808761221, shop, yes)
5   (2808761221, sport, free_flying)
6   (2808761221, street, Chemin de Pré Monteux)
7   (4360383476, city, Talloires)
8   (4360383476, housenumber, 150)
9   (4360383476, name, Les Grands Espaces)
10  (4360383476, opening_hours, Mo-Su 09:00-18:00)
11  (4360383476, phone, +33 4 50 60 79 06)
12  (4360383476, postcode, 74290)
13  (4360383476, shop, yes)
14  (4360383476, sport, free_flying)
15  (4360383476, street, Chemin de Pré Monteux)
16  (4360383476, website, http://grandsespaces-parapente-annecy.com/)
17  (5317939707, name, Chocolaterie)
18  (5317939707, shop, yes)
19  (5424257736, name, Indeeep)
20  (5424257736, shop, yes)
21  (5606465259, name, Alp Isolation)
22  (5606465259, shop, yes)
```

And some ambiguities : what is the difference between shop :"car" and shop :"car-repair" ? Between shop "deli" and shop "convenience".
If we would have to do a precise analysis of car retailers in France and their geographical position, we would have to cross check information with other sources of information like business directories or direct enquiry with car retailers, to check accuracy of information.

	"Shop" keys value	Number of occurrence (decreasing order)	
0	clothes	84	
1	bakery	74	
10	books	14	
11	car_repair	12	repairs only ?
14	car	11	car dealer
23	kiosk	7	News agent ? Convenience
24	greengrocer	7	
25	alcohol	7	
26	estate_agent	6	
27	cheese	6	shops selling exclusively cheese
28	yes	5	
29	tobacco	5	
30	deli	5	Shop focused on selling delicatessen , possibly also fine wine. Not to be confused with the US delis.
31	travel_agency	4	

In order to make sql queries, we have to convert Annecy.osm file to a .csv file and then to a sql data basis.

4. SQL DATABASIS CREATION AND SQL QUERIES

4.1 CREATION OF SQL DATABASIS

We will convert to a sql databasis all nodes and nodes tags, all ways and ways tags of Annecy.osm file. (Relations are not exported)

For this there are on line existing tools, for example: my geodata converter. Capacity of this tool is however limited to 50MB, which is a bit too small for our 152MB file.

Let's create five csv files, with appropriate content, let's change in street names, wrong entries identified previously. Thus, databasis will be cleaned for this aspect. It will be a pity to identify problems and import those problems in the data basis we want to query.

Five csv files are created with appropriate data to be converted in sql tables in a data basis.

See python file Annecy_osm_wrangling.ipynb for detail.

Csv files are converted in a sql databasis: Annecy.db

4.2 ACCURACY AND COMPLETENESS CHECKS

- To check completeness and accuracy of sql databasis, that is to check that all we found with etree tool is correctly reflected in Annecy.db, we check number of nodes and ways and we check on two tests that two entries "rue Cassiopée" and "chemin de Bellevue" are corrected in sql databasis
  Point of ambiguous classification for shops or amenities will still exist because we did not correct anything in this field.
  See python file Annecy_osm_wrangling.ipynb for detail.

I Indeed, it seems important to us to check that dataset is still complete and as accurate as possible when we come to further queries.

- We can see that sql users are less 793, that our first count in etree.926. We consider that it is due to the fact that we did not import all data of Annecy.osm (no relation imported).

- As regards main contributors : "*Emmanuel Pacaud* contributes for 50% to all entries in selected area".In case of further enquiry, we would contact him and check his osm blog. We can see on OSM information about users that Emmanuel Pacaud has added a lot of GPS traces(mountain bike, hiking..)
  Second contributor is JFK73, who seems to be also an active contributor in the field of ski and mountain bike https://www.salzburgresearch.at/blog/alpine-ski-world-championship-meets-openstreetmap-who-are-the-mapping-champions/

Once sql databasis is built, queries are much easier for us than wrangling with etree, and less time consuming as well. Perhaps because we did not get used to python since long enough. (first contact with python in May 2020).
We can see that sql databasis can be updated as well. So it could be a nicer solution to change more data with sql queries, rather than etree queries.
https://www.w3schools.com/sql/sql_update.asp

5. CONCLUSION

Data wrangling is really time consuming, but also rewarding.
Trying to understand logic and content of dataset as well as checking accuracy, completeness all along the process seems to represent 80% of the total workload.

Using existing tools for wrangling (geodata converter, PyOsmium…) seems a way to reduce associated workload.

Then checking data quality cannot be a stand alone process. As regards validity of keys, as we saw for shops, fine cleaning would need support of local users in the town together with Other sources of information.

We must know what we are looking in and what we are looking for, not to waste to much time. Wrangling tools are so powerful, that we can waste a lot of time, just exploring data out of curiosity. (We did that…)
We can say that Data wrangling is time consuming but also makes fun.