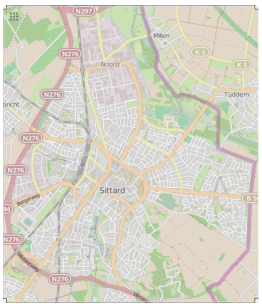
OpenStreetMap Sample Project
Data Wrangling with MongoDB

Mark Boon
Map Area: Sittard, Netherlands, Europe

About Sittard



Though not specifically related to data wrangling, I choose Sittard, because I live here and enjoy very much living here. Little town in the south of the Netherlands, next to the German border, but also very close to the Belgium border and about 25 km from Maastricht. For more info http://en.wikipedia.org/wiki/Sittard (and you are very welcome to visit)!.

Processing of the Map

Source data

The source data is based upon the following URL (including coordinates):
http://overpass-api.de/api/map?bbox=5.8370,50.9783,5.9065,51.0289

This resulted in a dataset 57 MB uncompressed dataset. As the area is a rectangle this also covers other villages including small part of Germany.

Auditing source data

Cities

The cities in this dataset showed inconsistencies. Partly this is related to display special characters and use of Python on Ubuntu ('Selfkant - T\xc3\xbcddern' should have been

‘Selfkant - Tüddern’), however also minor inconsistencies are noted (eg “Selfkant - Millen” vs “Selfkant-Millen”)

```
mbo@home:~/PythonScripts/datawrangling/assignment$ python step_unique.py sittard.xml
addr:city
Get unique list from XML file sittard.xml from addr:city
Start: 2015-03-22  11:31:00.522923

{'Born': 1,
 'Geleen': 415,
 'Limbricht': 735,
 'Munstergeleen': 226,
 'Selfkant': 13,
 'Selfkant - Millen': 11,
 u'Selfkant - T\xfcdderen': 2,
 u'Selfkant - T\xfcddern': 8,
 'Selfkant-Millen': 4,
 u'Selfkant-T\xfcddern': 11,
 'Sittard': 21658,
 'sittard': 1}

End: 2015-03-22  11:31:20.502125
Delta: 0:00:19.979202
```

After applying remapping in the processing script, this resulted in the following cities:

```
mbo@home:~/PythonScripts/datawrangling/assignment$ python step_unique.py sittard.xml
addr:city
Get unique list from XML file sittard.xml from addr:city
Start: 2015-03-22  15:54:09.738695

{'Born': 1,
 'Geleen': 415,
 'Limbricht': 735,
 'Munstergeleen': 226,
 'Selfkant': 13,
 'Selfkant - Millen': 15,
 'Selfkant - T\xc3\xbcddern': 21,
 'Sittard': 21659}

End: 2015-03-22  15:54:29.764841
Delta: 0:00:20.026146
```

Streets

Though very surprisingly, I did not detect any inconsistencies or abbreviations in street names. Typical dutch street name endings like “straat”, “laan”, “weg” are never abbreviated. Even common prefixed are not abbreviated like “Burgemeester”, “Dokter”, while it is very common to have these abbreviated when referring to such addresses.

Postal codes

The postal code showed minor inconsistencies. Dutch postal codes are formed by four numbers and two letters. An initial query showed the following set

```
{'52538': 49,
 '6131 AE': 1,
 '6131 AS': 1,
 '6131 AX': 1,
 '6131 BA': 1,
 '6131 CV': 1,
 '6131 EC': 1,
 '6131AA': 22,
 '6131AB': 7,
 '6131AC': 92,
 '6131AD': 5,
 '6131AE': 4,
 '6131AG': 23,
 .....
```

The postal code with only numbers is a German postal code. In the processing scripts blanks are moved from postal codes.

Scripts for correction of issues during auditing

The following snippet show the primary correction of postal code and cities:

```
remapCity={"Selfkant-Millen":"Selfkant - Millen","sittard":"Sittard", "Selfkant - Tüdderen":"Selfkant - Tüddern","Selfkant-Tüddern":"Selfkant - Tüddern"}

def remapping(key,value):
    value=value.encode("utf-8")
    if key=="addr:postcode":
        value=value.replace(" ","")
    elif key=="addr:city":
        if value in remapCity:
            value=remapCity[value]
    return value
```

Steps of importing to MongoDB

The following steps are used:

- processing to json using Python script including data manipulation
- mongoimport for importing the json to MongoDB

```
 mongoimport --db udacity2 --collection 20150327 --file sittard.xml.json --jsonArray
connected to: 127.0.0.1
2015-03-27T07:08:58.622+0100    Progress: 99124/79174410        0%
2015-03-27T07:08:58.622+0100        400     133/second
2015-03-27T07:09:01.183+0100    Progress: 172386/79174410       0%
2015-03-27T07:09:01.183+0100        700     116/seco
…
…
...
2015-03-27T07:43:10.050+0100    Progress: 76597121/79174410     96%
2015-03-27T07:43:10.050+0100        250100  121/second
2015-03-27T07:43:13.085+0100    Progress: 77993440/79174410     98%
2015-03-27T07:43:13.085+0100        252600  122/second
```

```
2015-03-27T07:43:14.078+0100 check 9 255138
2015-03-27T07:43:14.078+0100 imported 255138 objects
```

Data Overview

The original dataset:

```
-rw-rw-r-- 1 mbo mbo 59869489 mrt 22 07:07 sittard.xml
-rw-rw-r-- 1 mbo mbo 79174410 mrt 27 05:29 sittard.xml.json
```

The database is udacity2 and the collection is named 20150327.

```
> db.stats()
{
        "db" : "udacity2",
        "collections" : 3,
        "objects" : 255142,
        "avgObjSize" : 319.42845944611236,
        "dataSize" : 81499616,
        "storageSize" : 86327296,
        "numExtents" : 12,
        "indexes" : 1,
        "indexSize" : 7464688,
        "fileSize" : 201326592,
        "nsSizeMB" : 16,
        "dataFileVersion" : {
        "major" : 4,
        "minor" : 5
        },
        "extentFreeList" : {
        "num" : 0,
        "totalSize" : 0
        },
        "ok" : 1

> show collections
20150327
system.indexes
>
```

Stat	Command and result
Number of documents	> db.getCollection("20150327").find().count() 255138
Cities in dataset	> db.getCollection("20150327").aggregate([{"$match":{"address.city":{"$exists":1}}},{$group:{"_id":"$address.city", "count":{"$sum":1}}},{"$sort":{"count":-1}}]) { "_id" : "Sittard", "count" : 21659 } { "_id" : "Limbricht", "count" : 735 } { "_id" : "Geleen", "count" : 415 } { "_id" : "Munstergeleen", "count" : 226 } { "_id" : "Selfkant - Tüddern", "count" : 21 } { "_id" : "Selfkant - Millen", "count" : 15 } { "_id" : "Selfkant", "count" : 13 } { "_id" : "Born", "count" : 1 }

	Note that: - UTF8 handling is correct. During auditing the “ü”-character was not displayed as expected. This showing correct now: Tüddern
Top 10 Contributors	> db.getCollection("20150327").aggregate([{$group:{"_id":"$created.user", "count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":10}]) { "_id" : "Martin Borsje_BAG", "count" : 189985 } { "_id" : "3dShapes", "count" : 13390 } { "_id" : "gepeix", "count" : 12806 } { "_id" : "ARWIE", "count" : 10460 } { "_id" : "Martin Borsje", "count" : 5267 } { "_id" : "Willem1", "count" : 4489 } { "_id" : "Christoph Lotz", "count" : 3676 } { "_id" : "It's so funny_mechanical", "count" : 3431 } { "_id" : "AND", "count" : 2791 } { "_id" : "rivw", "count" : 1107 }
Top 10 Contributors of nodes that have a city	> db.getCollection("20150327").aggregate([{"$match":{"address.city":{"$exists":1}}},{$group:{"_id":"$created.user", "count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":10}]) { "_id" : "Martin Borsje_BAG", "count" : 18229 } { "_id" : "Christoph Lotz", "count" : 3669 } { "_id" : "rivw", "count" : 978 } { "_id" : "Willem1", "count" : 116 } { "_id" : "Maarten Deen", "count" : 33 } { "_id" : "Glenn Plas", "count" : 19 } { "_id" : "gepeix", "count" : 13 } { "_id" : "Dreamseer", "count" : 12 } { "_id" : "It's so funny_mechanical", "count" : 4 } { "_id" : "Stationskwartier_Sittard", "count" : 2 }
Shops	> db.getCollection("20150327").aggregate([{"$match":{"address.city":"Sittard"}},{"$match":{"shop":{"$exists":1}}},{$group:{"_id":"$shop", "count":{"$sum":1}}}]) { "_id" : "clothes", "count" : 1 } { "_id" : "gift", "count" : 1 } { "_id" : "chemist", "count" : 1 } { "_id" : "supermarket", "count" : 2 }
Amenities	> db.getCollection("20150327").aggregate([{"$match":{"address.city":"Sittard"}},{"$match":{"amenity":{"$exists":1}}},{$group:{"_id":"$amenity", "count":{"$sum":1}}}]) { "_id" : "fuel", "count" : 1 } { "_id" : "parking", "count" : 1 } { "_id" : "atm", "count" : 6 } { "_id" : "bank", "count" : 5 }

Additional ideas

As I am living for a while in Sittard, I know Sittard and the data does really not represent what Sittard offers in shopping, culture and facilities. Sittard has definitely more shops than the 5 (!?) from the query. The 5 shops in the data are:

```
> db.getCollection("20150327").aggregate([{"$match":{"address.city":"Sittard"}},{"$match":{"shop":{"$exists":1}}},{"$group":{"_id":{"shop":"$shop","name":"$name"}}}])
{ "_id" : { "shop" : "clothes", "name" : "H&M" } }
{ "_id" : { "shop" : "gift", "name" : "WW Sittard" } }
```

```
{ "_id" : { "shop" : "chemist", "name" : "Etos" } }
{ "_id" : { "shop" : "supermarket", "name" : "Albert Heijn TOGO" } }
{ "_id" : { "shop" : "supermarket", "name" : "Albert Heijn" } }
```

I could imagine that the company owner “Etos” or “Albert Heijn” could updated this list however this is inconsistent; e.g. there are more “Albert Heijn” supermarkets in Sittard than the listed two.

Similar applies to amenities:

```
>
db.getCollection("20150327").aggregate([{"$match":{"address.city":"Sittard"}},{"$match":{"amenity":{"$exists":1}}},{"$gr
oup":{"_id":{"amenity":"$amenity","name":"$name"}}}])
{ "_id" : { "amenity" : "atm", "name" : "Rabobank" } }
{ "_id" : { "amenity" : "atm", "name" : "ING" } }
{ "_id" : { "amenity" : "fuel", "name" : "Tinq" } }
{ "_id" : { "amenity" : "parking", "name" : "Q-Park Transferium" } }
{ "_id" : { "amenity" : "bank", "name" : "ING" } }
{ "_id" : { "amenity" : "bank", "name" : "Postbank" } }
{ "_id" : { "amenity" : "atm" } }
{ "_id" : { "amenity" : "bank", "name" : "Rabobank" } }
{ "_id" : { "amenity" : "atm", "name" : "Fortis" } }
{ "_id" : { "amenity" : "bank", "name" : "SNS Bank" } }
{ "_id" : { "amenity" : "bank", "name" : "ABN AMRO" } }
```

When querying through a specific street (“Oude Markt”) that have remarkable historical landmarks using query, only “technical” data is shown of addresses and postal codes.

```
db.getCollection("20150327").aggregate([{"$match":{"address.city":"Sittard","address.street":"Oude Markt"}}]),
```

Conclusion

This Sittard Openstreet dataset is technically fairly clean. There are minor inconsistencies on naming of cities and few on postal codes. The main lack is any data describing what is behind addresses as all seem to be private housing. There are sufficient datasets to have this enriched such chambre of commercial activities, church organizations for holy places and government for school, sports, cultural places. However I can fully imagine that such organizations do currently not consider OpenStreet a sufficiently known tool to reach audiences and effort of enriching (and more important, maintaining), may be considered not worth the effort. It could be a good case for a course exercise!