OSM Project - Data wrangling and cleaning

Content

Map area

The area covers the city of Mannheim, a city in the South-west of Germany with about 300.000 inhabitants. I downloaded the data from mapzen

- http://www.openstreetmap.org/node/240060919 (http://www.openstreetmap.org/node/240060919)

I choose this city because I am currently living there since several years and I know the area well. This makes it easier for me to spot errors and problems in the map. Futhermore, as an OSM user, I benefit from any improvement of the map directly.

Before I turn to the analysis of potential problems with the data quality, I describe the dataset in more detail.

Number of nodes, ways, node tags and unique users

In [ ]:
```
SELECT count(*) FROM nodes
SELECT count(*) FROM ways
SELECT count(distinct(e.uid)) FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM
ways) e
SELECT count(key) FROM nodes_tags

Number of nodes: 889120
Number of ways: 189316
Number of unique users: 1248
Number of node tags: 143347
```

There are 889,120 nodes, 189,316 ways and 143,437 node tags. These entries were contributed by 1,248 users, but only a handful of users has contributed the majority of entries. From a total of 1,078,336 the two top users *wegavision* and *BernhardW* contributed 481,295 or about 45% of all entries.

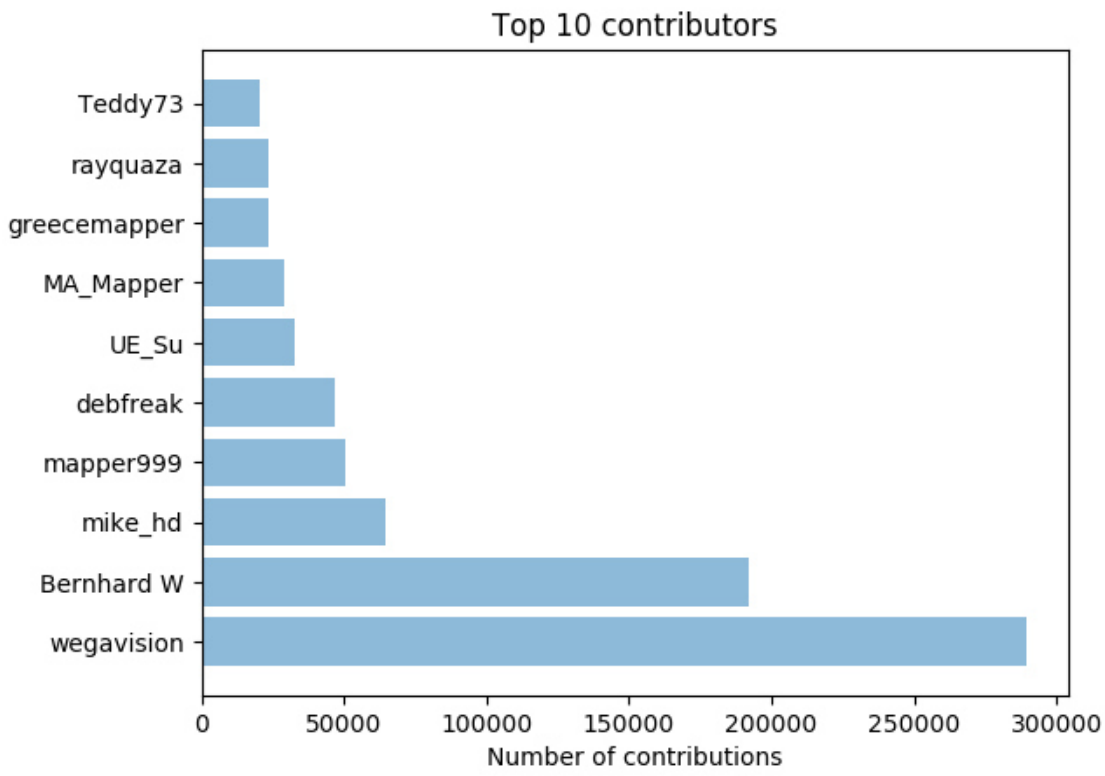
In [ ]:
```
## File sizes
mannheim.osm.....227 mb
osm.db..........130 mb
nodes.csv.........71 mb
nodes_tags.csv.....5 mb
ways.csv..........11 mb
ways_tags.csv.... 25 mb
ways_nodes.csv....29 mb
```

In [5]:
```
SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
ORDER BY num DESC
LIMIT 10;
```

```
wegavision 289493
Bernhard W 191802
mike_hd 64680
mapper999 50628
debfreak 47120
UE_Su 32520
MA_Mapper 29059
greecemapper 23534
rayquaza 23512
Teddy73 20568
```

In [3]:
```
from IPython.display import Image
Image("img/bar.png")
```

Out[3]:



Problems in the map

I used a small sample of every 10th entry of the overall map to assess the data quality. Overall, I was surprised of the overall very good consistency of the data. I have checked in particular the following entries:

- Consistency of company names
- Phone and fax numbers
- Postal codes, street names
- Opening hours of shops

On closer look only the two first points were problematic and included errors.

Consistency of company names

A SQL query of "name"-keys in the table *nodes_tags* revealed some inconsistancies.

- The branches of a backery chain were named *Grimminger* while others were named *Grimminger Bäckerei* (Bäckerei = backery).
- The branches of another chain were named *Görtz* and *Bäckerei Görtz*
- Shops of the supermarket chain were named *Penny Markt* and *Penny* I did not see any reason why different branches of the same company have different names and this might lead to problems if users search for a particular company. Therefore, I implemented a cleaning procedure that consolidated the names. I have choosen the name that was used more often for each company respectively.

Phone and fax numbers

Another SQL query showed that the formatting of telephone numbers were inconsistent. Most numbers started with the international prefix for Germany *+49* but others did not. Furthermore, some entries used () or - but others did not. I decided to use one format for all numbers. Apparently, there are several ways to format numbers (https://en.wikipedia.org/wiki/National_conventions_for_writing_telephone_numbers#Germany (https://en.wikipedia.org/wiki/National_conventions_for_writing_telephone_numbers#Germany)) I decided to use the formatting of **(+49) XXX XXXXX**. As an example, 0621-8374624 was programatically changed to (+49) 621 8374624.

Postal codes, street names

I checked for problems with postal codes but no problems occured. Some of the postal codes belong to neighboring cities of Mannheim but I did not expect the OSM extract to be entirely in line with the postal code areas. I also run a SQL query for streetnames and could not see any inconsistencies (only the first 20 are shown). In German, streets are often abbreviated with *Str.* instead of *Straße* but this was not the case in the OSM file.

In [6]:
```
# Most frequent postalcodes
SELECT value, count(value) FROM ways_tags WHERE key='postal_code' GROUP BY value OR
DER BY count(value) DESC LIMIT 20
```

```
68169 90
67067 55
68239 43
68259 35
67065 30
68519 14
68723 3
68219 2
68549 2
67063 1
67071 1
68229 1
68526 1
68535 1
68782 1
```

In [7]:
```
# Most frequent streenames
SELECT value, count(value) FROM ways_tags WHERE key='name' GROUP BY value ORDER BY
count(value) DESC LIMIT 20
```

```
Frankenthaler Straße 134
Riedbahn 129
Waldstraße 127
Mannheimer Straße 116
Friedrich-Ebert-Straße 103
Diffenéstraße 79
Neckarauer Straße 76
Hauptstraße 74
Speyerer Straße 71
Seckenheimer Landstraße 66
Bismarckstraße 61
Friedrichsring 59
Rhenaniastraße 57
Luisenring 56
Ludwigshafener Straße 54
Magdeburger Straße 54
Maudacher Straße 54
Tannhäuserring 54
Seckenheimer Straße 49
Weinheimer Straße 48
```

Opening hours of shops

Lastly, I had a look at opening hours. The German OSM wiki suggests several posibilities to format opening hours (http://wiki.openstreetmap.org/wiki/DE:Key:opening_hours (http://wiki.openstreetmap.org/wiki/DE:Key:opening_hours))

- 24/7
- 08:00-18:00 --> open daily
- Mo 10:00-12:00,12:30-15:00; Tu-Fr 08:00-12:00,12:30-15:00; Sa 08:00-12:00 --> different opening times for each day
  The name of the days is supposed to be in English. Nearly all entries were in line with these rules. The only inconsistency I found were different options for 24/7: 0:00-24:00, 00:00-24:00. I changed these to 24/7 as suggested by the Wiki.

Additional ideas and data exploration

New data structure for opening hours

The opening hours are largely consistent, but further calculations are very difficult when the data is stored in this way. For example, 06:00-17:00, **Mo-Fr** 6:00-17:00, Mo-Fr **0**6:00-17:00 indicate the same opening times in different ways. Furthermore, in this format it is difficult to calculate if somebody wants to know which coffee shops are open at the moment (such as in Google maps). To allow for this function, a better data structure would be:

- Mo 06:00-17:00
- Tu 06:00-17:00
- We 06:00-17:00
- Th 06:00-17:00
- Fr 06:00-17:00
- Sa closed
- Su closed

Benefits of this improvement:

- In such a format, opening times were more accessible programatically. For example, it would be easier to check which coffee shops are currently open, because we do not need to decode the unstructured entry.
- It is also easier to read for users.

Anticipated Problems:

- There would be a need for a standardized form for users to enter opening hours.
- If we want to change the existing entries to the new standardized format, we would need handle many exceptions and inconsistencies in the data.

Opening times of cafes

The DB can be used to assess the opening times of all Coffee shops. However, for only few of them opening hours are available. To implement a feature like "which cafe is open atm?" there would be a need for more detailed information on opening hours.

In [11]:
```
# Coffee shops and opening times
SELECT DISTINCT c.value, b.value
FROM nodes_tags as a, nodes_tags as b, nodes_tags as c
WHERE a.id = b.id AND b.id = c.id AND
a.value = "cafe" AND b.key = "name" AND c.key = "opening_hours
```

```
10:00-23:00 Eiscafe Riviera
Mo-Sa 09:30-20:00 Starbucks Coffee
Mo-Sa 09:30-20:00 Tchibo
Mo-Su 04:30-22:00 Wiener Feinbäckerei
Tu-Su 09:00-18:00 Konditorei Christmann
Mo-Sa 08:30-20:00 Tchibo Filiale
Mo-Fr 9:00-20:00 Tchibo
Mo-Fr 08:00-16:00 Euro Bistro
Tu-Sa 10:00-19:00 Café Sammo
Mo-Su 09:00-18:00 Cafè Meerwiesen
Mo-Sa 09:00-20:00 Tchibo Filiale
We-Mo 14:00-1:00 Cafe Secret
Mo-Su 10:00-22:00 Eiscafé Piazza
Mo-Su 6:00 - 18:00 Herzhaft-Süß
8:30-1:00 dolceamaro
Tu-Su 11:00-18:00 Café Mohrenköpfle Konditorei
9:00-18:30 Konditorei Café Wissenbach
Mo-Su 9:00-18:00 Kleines Café
9:00-1:00 Lavandou
Tu-Su 12:00-20:00 Eismanufaktur Zeitgeist
Mo-Sa 07:00-18:00 Bäckerei Kaya
Mo-Su 10:00-22:00 Eiscafe Gelateria-Forum
Tu-Sa 10:00-18:00 Kaffeerösterei Lauri
11:00-22:00 Eiscafe Adria
Mo-Su 11:00-22:00 Eis-Café Philip Franck
Mo-Su 09:00-01:00 Starks
Mo-Sa 10:00-20:00 Nespresso Boutique Mannheim
10:00-22:00 Eiscafé La Pallina
```

Wheelchair accessibility

The DB can also be used to see how accessible cafes and restaurants are for handicapped persons (yes/no/limited). For restaurants: yes 22%, limited 26%, no 52% For cafes: yes 31%, limited 25%, no 43%

Overall, 477 restaurants and 229 cafes are tagged for their accessibility. This data could be used for a more detailed analysis of accessibility in different cities and of public transport.

Benefits of this improvements:

- We can assess how wheelchair friendly a certain city or area is. This could be interesting for individuals with handicap before they move.

Anticipated Problems:

- It is not clear how reliable this data is. There are some rough guidelines on the OSM website, but users who enter data might have different perceptions of what is accessible and what is not.

In [18]:
```
# Cafes and wheelchair accessability
SELECT COUNT(b.value), b.value
FROM nodes_tags as a, nodes_tags as b
WHERE a.id = b.id AND
a.value = "cafe" AND b.key = "wheelchair"
GROUP BY b.value;

# Restaurants and wheelchair accessability
SELECT COUNT(b.value), b.value
FROM nodes_tags as a, nodes_tags as b
WHERE a.id = b.id AND
a.value = "restaurant" AND b.key = "wheelchair"
GROUP BY b.value;
```

```
Cafes
(58, u'limited')
(99, u'no')
(2, u'survey')
(72, u'yes')
Restaurants
(124, u'limited')
(250, u'no')
(1, u'unknown')
(103, u'yes')
```