Open Street Map Project

Author : Görkem Berk Şahan
Map Area : Istanbul
https://mapzen.com/data/metro-extracts/metro/istanbul_turkey/ (https://mapzen.com/data/metro-extracts/metro/istanbul_turkey/)

Why I choose this location is actually I know and I live in Kocaeli, near of Istanbul city.

Creating Sample Data File

Firt of all we create an sample file from actual data file with taking sample data, After take a look sample data, we can see some problems like that ;

TAKE A LOOK STREETS :

Mh.
O.S.B.
Sahra Mah.
İstiklal Cad.
a Parkı Cad.
ilköy Mah.
Cad.
Cd.
Merkez Mah. Halkalı Cad.
Sok.
Sk.
Cad.
al Fevzi Çakmak 3. Sk.
a Sk.
Yolu Sk.
Pasa Sk.
Paşa Sk.
Sultan Mehmet Cd.
Sk.
cad.
lıç Ali Pasa Mescidi Sk.
//www.istiklalsthouse.
Mah. Dereboyu Cad. Fulye Sok.
kışla Cad.
Kahveci Mah. Yavuz Sultan Selim Bulv.
Caddesi 10.
Batıyol Sk.

INVALID POST CODES :

3400

INVALID CITY NAMES :

İSTANBUL
Ümraniye/İstanbul
Çekmeköy
İstanbul/Sultanbeyli
İstasyon Mh. / Kocaeli
İstanbul
Üsküdar/İstanbul
Şişli/İstanbul
Çatalca/İstanbul
İstanbul Çekmeköy

1. Problems Encountered In Map Data

```
a. Some street names shorten like below ;
    Sokak as Sok.,Sk.
    Cadde as Cd., Cad.
    Mahalle as Mah.
    Bulvar as Bulv.
b. There is invalid Post codes like 3400
c. Some city names has county names with city name like Üsküdar/İstanbu
l so "/" or " "(space) used as delimeter and one of its items is city n
ame and other one is county.
d. City names is not same format such as camel case ( Istanbul or ISTAN
BUL )
```

2. Cleaning Data and Creating Json File and MongoDb

After audit data, some desicions to clean data is like that;

- Street names has shorten values, correct them
- Turkish characters is a open issue for compare,group beacuse some users use traditional char but some doesnt, so we clear it to english chars.
- Some address values is inconsistent, such as it contains county instead city in city key, correct it if we know the true one
- Some street values has an website address so we clear it
- Some postcodes is inconsistent of standart postcode, for ex. postcodes of Istanbul city must start 34 , so we can check it

We will create four correction fuction to audit and correct city, street and Post code values :

- Chaning Turish characters to English characters :

- We will create a dict to use in changing process and if Street,City names has an Turish character(one of keys i dict.) it will be changed to dict. value. ;

```
list = {"İ":"I",
        "ı":"i",
        "ö":"o",
        "Ö":"O",
        "ş":"s",
        "Ş":"S",
        "ü":"u",
        "Ü":"U",
        'Ğ':'G',
        'ğ':'g',
        'ç':'c',
        'Ç':'C'
       }
```

- Audit City names and correct :
  - Beacuse of selected map area, city names must not have an special characters '/',' ',... etc, so if it contain one of this we must check it, after checking process I realize that some people enter county/city ort county city
  - We will have and dict. so that audit data and changing to correct one, to create this dictionary, I used results above audt result.

```
city_names =
{  'istanbul' : 'istanbul',
   'kocaeli':'kocaeli',
   'gebze':'kocaeli',
   'avcilar':'istanbul',
   'sariyer':'istanbul',
   'beylikduzu':'istanbul',
   'sancaktepe':'istanbul',
   'cekmekoy':'istanbul',
   'beyoglu':'istanbul',
   'bakirkoy':'istanbul',
   'sultanbeyli' : 'istanbul',
   'pendik' : 'istanbul',
   'kadikoy' : 'istanbul',
   'sisli' : 'istanbul',
   'tuzla' : 'istanbul',
   'esenyurt' : 'istanbul',
   'uskudar' : 'istanbul',
   'sile' : 'istanbul',
   'atasehir' : 'istanbul',
   'istambul' : 'istanbul',
   'kartal' : 'istanbul',
   'kagithane' : 'istanbul',
```

```
'heybeliada' : 'istanbul',
'maltepe' : 'istanbul',
'dilovasi' : 'istanbul',
'sultanahmet' : 'istanbul',
'bahcelievler mahallesi' : 'istanbul',
'yenibosna' : 'istanbul',
'bayrampasa' : 'istanbul',
'darica' : 'istanbul',
'kilyat' : 'istanbul',
'fatih-istanbul' : 'istanbul',
'basaksehir' : 'istanbul',
'cayirova' : 'istanbul',
'ora' : 'istanbul',
'umraniye' : 'istanbul',
'zeytinburnu' : 'istanbul',
'kavacik' : 'istanbul',
'istanbbul' : 'istanbul',
'buyukada' : 'istanbul',
'besiktas' : 'istanbul',
'selinpasa' : 'istanbul',
'umraniye/istanbu' : 'istanbul',
'rumeli' : 'istanbul',
'eyup' : 'istanbul',
'elmadag/sisli' : 'istanbul',
'sekerpinar' : 'istanbul',
'istanbus' : 'istanbul',
'balat' : 'istanbul',
'kumburgaz' : 'istanbul',
'topkapi' : 'istanbul'
}
```

- Audit Street names and correct :
  - When looking sample data I saw some shorten values, and create an dictionary and additional control if it has web site in street name. To decide what we change, I write an regex to find shorten values xxx yyy zz. -> zz.
  - **re.compile("\w+.(?P.+\.)?",re.IGNORECASE)**

```
map_Street = {
    "Sok.":"Sokagi ",
    "Sk.":"Sokagi ",
    "Cad.":"Caddesi",
    "Cd.":"Caddesi ",
    "Mah.":"Mahallesi ",
    "Mh.":"mahallesi ",
    "Bulv.":"Bulvari "
}
```

- Audit Post code :

- Because of standart of this locations post codes, its prefix must be same for city , and map location for Istanbul it must starts with 34 and for Kocaeli 41

when correction , some result like that :

```
city old : Sütlüce \ Istanbul
city new : Istanbul

city old : Üsküdar - Istanbul
city new : Istanbul

street old : Salancak, Üsküdar
street new : Salancak, Uskudar

street old : Yıldız Posta Caddesi
street new : Yildiz Posta Caddesi

street old : Büyükdere Cad.
street new : Buyukdere Caddesi

city old : ISTANBUL
city new : Istanbul
```

suggestions for improving the data

When we look data structure, we can see that tag values kept as parentkey:childkey and value, and if there is no child value so it kept as key:value so taking this data take more time, and as I understand there is no data validation for example if city name contains special characters / or - , it may be asking user to confirm this is true.

We can correct data with some methods but we can not sure if we dont create inconsistent data, because user enter the data wrongly and we change it by our supposing, for ex. when we change street names we cannot sure it is shorten or really street name.

Changing Values :

- Benefits :
  - data will be standart to analyse and using
  - data size may increase if it has wrong type and not normalized like city property has county value too
- Anticipated Issues :
  - we may change data to incorrect form such that incomprehensible from other people
  - user that gave data maybe made a mistake and we suppose to improve it
  - it is hard to improve unpatterned datas such as write failured such as Istnbul -> Istanbul

3. Data Overview

istanbul.osm file 252 MB	istanbul.osm.json file 274 MB

```
>>db.osmMapData.distinct("created.uid")

Unique user count :  2333
```

```
>>db.osmMapData.aggregate([{"$match":{ "type":{ "$in":["way","node"]} }
},{"$group":{"_id":"$type","count":{"$sum":1}}},{"$sort":{"count":-1}}]
)

node and way count
{u'count': 1156118, u'_id': u'node'}
{u'count': 191232, u'_id': u'way'}
```

```
>>db.osmMapData.aggregate([{"$group":{"_id":"$addr.city","count":{"$sum
":1}}},{"$sort":{"count":-1}}])

city counts
{u'count': 1345191, u'_id': None}
{u'count': 2133, u'_id': u'Istanbul'}
{u'count': 51, u'_id': u'Kocaeli'}
```

```
>>db.osmMapData.aggregate([{"$match":{"type":{"$in":["node","way"]}}},{
"$group":{"_id":"$addr.street","count":{"$sum":1}}},{"$sort":{"count":-
1}},{"$limit":10}])

top streets that has data
{u'count': 1345727, u'_id': None}
{u'count': 52, u'_id': u'Fatih Sultan Mehmet Caddesi'}
{u'count': 48, u'_id': u'Kilyos Caddesi'}
{u'count': 46, u'_id': u'Mese Sokak'}
{u'count': 36, u'_id': u'Senay Sokak'}
{u'count': 35, u'_id': u'Cam Sokak'}
{u'count': 33, u'_id': u'Zambak Sokak'}
{u'count': 31, u'_id': u'Ardic Sokak'}
{u'count': 27, u'_id': u'Erguvan Sokak'}
{u'count': 27, u'_id': u'Manolya Sokak'}
```

4. Additional Ideas

```
all users groped their name
db.osmMapData.aggregate( [{"$group":{ "_id":"$created.user","count":{"$
sum":1} }},{"$sort":{"count":-1}} ,{"$limit":10}])


top 10 users
db.osmMapData.aggregate( [{"$group":{ "_id":"$created.user","count":{"$
sum":1} }},{"$sort":{"count":-1}} ,{"$limit":10}])

all user statistics :
              count
count   2337.000000
mean     576.540436
std     4002.836227
min        1.000000
25%        1.000000
50%        6.000000
75%       47.000000
max    90043.000000

Top 10 users statistis :
              count
count     10.000000
mean   51483.800000
std    22416.471919
min    25341.000000
25%    36944.750000
50%    48834.000000
75%    60116.750000
max    90043.000000


total point count :  1347375


Nesim added about  6 % of total points
bigalxyz123 added about  6 % of total points
Cicerone added about  4 % of total points
Ckurdoglu added about  3 % of total points
katpatuka added about  3 % of total points
JeLuF added about  3 % of total points
EC95 added about  2 % of total points
canTurgay added about  2 % of total points
```

```
Sakthi20 added about  2 % of total points
turankaya74 added about  1 % of total points
```

- Top user Nesim has added 90043 points 6% of total points
- Top 10 users has added 32% of all points

Conclusion

After this analysis, we can see that data that inserted by users couldnt be validated. Even if clean data , some city or street data is incorrect.There are so many case to check data if really work long time on this. OSM must collect validation rules like data, so data will be clean and standart.

Resources : N/A