Data Wrangling

I have choosen Ahemdabad because it is one of the most commercial cities in India .

Its OSM File is big enough for cleaning . It is a 109 MB file

Code Resources

Code Library

Basic resources for this project are software libraries for python and MondoDB.

Code for Researching the Imported Files and Creating the Data.

Code snippets of the courses "Intro to Relational Databases", "Data Wrangling with MongoDB" (udacity.com) have been used here for downloading, analysing and cleaning the data. On this basis, several useful functions were built for these goals too.

Task carried out are :

1)Auditing .

2)Coverting OSM data to CSV files.

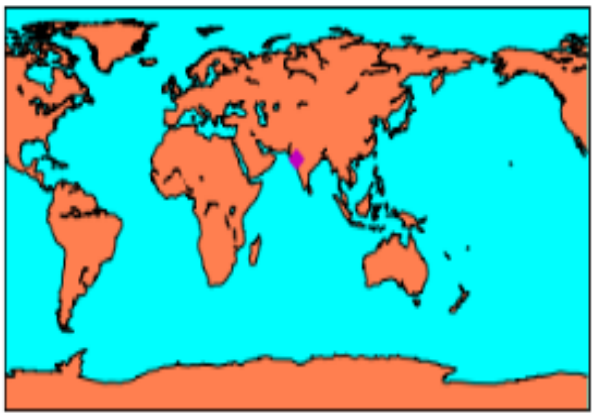
3)Using CSV data to create database.

4)Using database to extract important information.

5)I have plotted the city of Ahemdabad on world map using Folium , Matplotlib , and Basemap libraries.

Basemap library is just an extension of Matplotlib.

Plotting City on Map :



The Magenta colored Dot represents the location of Ahemdabad on world map ,which is at west of India .

Below plots are drawn using Folium:


Udaipur
Palanpur
Mahesana
Gandhidham
Ahmedabad
Ujjain
Dewas
Indore
Gujarat
Jamnagar
Rajkot
Vadodara
Bhavnagar
Porbandar
Junagadh
Surat
Jalgaon
Dhule
Valsad
Malegaon

Auditing :

Problems with the data :

1) The Data had wrongly spelled words and many abbrevations used which had to be replaced . Hence , Rd. Ft. , St. etc with Road , Feet , Saint and so on.

2) S.H and N.H means State Highway and National Highway.

3)Abbreviations: \S+[A-Z] Can be used for follows [CG Road very important abb to clean] Name Replace list : 1)CG -Chimanlal Girdharlal Road , (In some places CG road OR some places Chimanlal Girdharlal Road OR C.G. Road) 2)Sarkhej–Gandhinagar Highway - S.G. Highway

4)Sosciety,Socity,soc : as: Society

Results of few example can be seen as follows:

BEFORE AUDITING	AFTER AUDITING
132 Ft. Ring Road	132 Feet Ring Road
St. Xavier's College Road	Saint Xavier's College Road
100 Feet Rd.	100 Feet Road
Dutt Sosciety Bhattha, Paldi	Dutt Society Bhattha, Paldi
shiv-sankalp soc. amikunj cross road	shiv-sankalp Society amikunj cross road
SH-41	State Highway-41
NH8	National Highway8
CG Road C.G Road	Chimanlal Girdharlal Road
S.G	Sarkhej-Gandhinagar

Coverting OSM data to CSV files:

FILE	SIZE
nodes.csv	44,430KB
nodes_tags.csv	243KB
ways.csv	4799 KB
ways_nodes.csv	15096 KB
ways_tags.csv	2943KB

Creating Database:

Now there are two types of creating data base using Command line and using sqlite library in Jupyter Notebook.

I preferred using Sqlite library in Jupyter notebook.

Extracting Information

I extracted information like :

1) Number of stations

2) Number of Tourist places

3) Number of Religion places

4) Number of Amenities

5) Categorizing Amenities

Finding number of Railway Stations.

(As printscreen of my laptop is not working I have copy pasted the section of code.)

```
QUERY = ( '"SELECT DISTINCT COUNT(id)
        FROM nodes_tags
        WHERE value = 'station' '")
cur.execute(QUERY)
all_rows = cur.fetchall()
import pandas as pd
df = pd.DataFrame(all_rows)
print("Number of stations are")
pprint(df)
```

Result:

```
Number of stations are
    0
0  28
```

Finding number of tourist places

```
QUERY = ( '"SELECT DISTINCT COUNT(id)
        FROM nodes_tags
        WHERE key = 'tourism' AND value = 'attraction'  '")
cur.execute(QUERY)
all_rows = cur.fetchall()
```

```
import pandas as pd

df = pd.DataFrame(all_rows)

print("Number of Tourist places are")

pprint(df)
```

RESULT :

```
Number of Tourist places are
    0
0  68
```

Finding number of religion places

```
QUERY = ( '''SELECT DISTINCT COUNT(id)

        FROM nodes_tags

        WHERE key = 'religion' AND value = 'hindu'  ''')

cur.execute(QUERY)

all_rows = cur.fetchall()

import pandas as pd

df = pd.DataFrame(all_rows)

print("Number of Religion Places are")

pprint(df)
```

RESULT:

```
Number of Religion places are
    0
0  60
```

Counting number of amenities:

QUERY = ('"SELECT value,count(*)as num from (select value,key from nodes_tags UNION ALL select value,key from ways_tags)

where key='amenity'

group by value

order by num desc '")

```
cur.execute(QUERY)

all_rows = cur.fetchall()

print("Number of Amenities are")

pprint(all_rows)
```

```
Number of Amenities are
                      0   1
0                school  45
1              hospital  38
2               college  36
3            restaurant  34
4                  fuel  24
5     place_of_worship  24
6           bus_station  22
7                cinema  21
8               parking  19
9             fast_food  16
10          marketplace  13
11                 bank  10
12                 cafe  10
13           university   9
14               police   8
15         fire_station   6
16              library   6
17              toilets   6
18       drinking_water   5
19                  atm   4
20           courthouse   4
21             pharmacy   4
22          post_office   4
```

```
23     public_building   4
24                taxi   4
25             theatre   4
26               bench   3
27            fountain   3
28            post_box   3
29              studio   3
30            car_wash   2
31    community_centre   2
32           ice_cream   2
33               plaza   2
34              prison   2
35       swimming_pool   2
36            townhall   2
37         arts_centre   1
38                 bar   1
39  motorcycle_parking   1
```

 # CATEGORIZING THE AMENITIES BASED UPON THEIR FUNCTIONALITY :

QUERY = ('''SELECT value,count(*)as num from (select value,key from nodes_tags UNION ALL select value,key from ways_tags)

Where key='amenity' AND value IN ('school','hospital','college','restaurant','fuel','place_of_worship','bus_station','parking', 'marketplace','bank','university','fire_station' , 'library', 'toilets' , 'drinking_water' ,'atm' , 'pharmacy' , 'post_office' ,'public_building', 'taxi' , 'bench' , 'fountain', 'post_box' , 'car_wash' , 'community_centre' , 'plaza','swimming_pool', 'townhall' , 'arts_centre' , 'motorcycle_parking')

''')

```
cur.execute(QUERY)

all_rows = cur.fetchall()

print("Number of classified Amenities as Public are:")

for i , j in all_rows:

   print j
```

RESULT:

```
Number of classified Amenities as Public are:
338
```

```
QUERY = ( '''SELECT value,count(*)as num from (select value,key from nodes_tags  UNION ALL select value,key from ways_tags)

        where key='amenity' AND value IN ('prison','police','courthouse' )

         ''')

cur.execute(QUERY)

all_rows = cur.fetchall()

print("Number of classified Amenities as Goverment are:")

for i , j in all_rows:

    print j
```

RESULT:

```
Number of classified Amenities as Goverment are:
14
```

```
QUERY = ( '''SELECT value,count(*)as num from (select value,key from nodes_tags  UNION ALL select value,key from ways_tags)

        where key='amenity' AND value IN ('restaurant' , 'fast_food' ,'cafe' , 'drinking_water', 'ice_cream' , 'bar' )

         ''')

cur.execute(QUERY)

all_rows = cur.fetchall()

print("Number of classified Amenities as Food are:")

for i , j in all_rows:

    print j
```

RESULT:

```
Number of classified Amenities as Food are:
68
```

```
QUERY = ( '''SELECT value,count(*)as num from (select value,key from nodes_tags  UNION ALL select value,key from ways_tags)

        where key='amenity' AND value IN ('theatre' , 'studio' )

         ''')

cur.execute(QUERY)

all_rows = cur.fetchall()

print("Number of classified Amenities as Entertainment are:")

for i , j in all_rows:

    print j
```

RESULT:

```
Number of classified Amenities as Entertainment are:
7
```

Suggestions:

I think gamification is a good way for contribution regarding the context of Open Street Maps, if user knowledge were a lot of conspicuously displayed, maybe others would take associate degree initiative in submitting a lot of edits to the map. And, if everybody sees that solely a few of power users area unit making 80-90% of given map, that may spur the creation of a lot of economic bots, particularly if bound gamification components were gifts, like rewards , badges, or a leader board.

Accent marks and imprecise translations have the potential to add more dirty data and I think there could be a community page for each region in the OSM wiki with information on how to deal with it. This would make it easier for contributors and users to know what information to expect while still allowing for the community to have ownership over local data.

Advantages and Disadvantages :

This website can be viewed as a testing ground of interaction of a large number of people (including non-professionals) to create a unified information space. The prospects of such cooperation can not be overemphasized. The success of the project will allow to implement the ambitious plans in the field of available information technologies, the creation of virtual reality and many other areas.

The main advantage of Open Street map data is that it is open source and therefore free to use . This means any one can use data and create their own maps (and then use services like Mapbox to design host customized map tiles). This means developer does not have to work within Googles constraints.

The only imaginable downside to me is quality. Get me right, 99% is the good stuff, but as all crowd sourced data it's hard to maintain consistent quality control. Of course is free to alter and complete the data, but there's no guarantees as there would with a company behind it.

Conclusion :

The OSM data was audited , cleaned and underwent analysis, Sql queries helped in extracting many useful things .As you can see from the project file Data wrangling process cycle of auditing , cleaning and repeat.