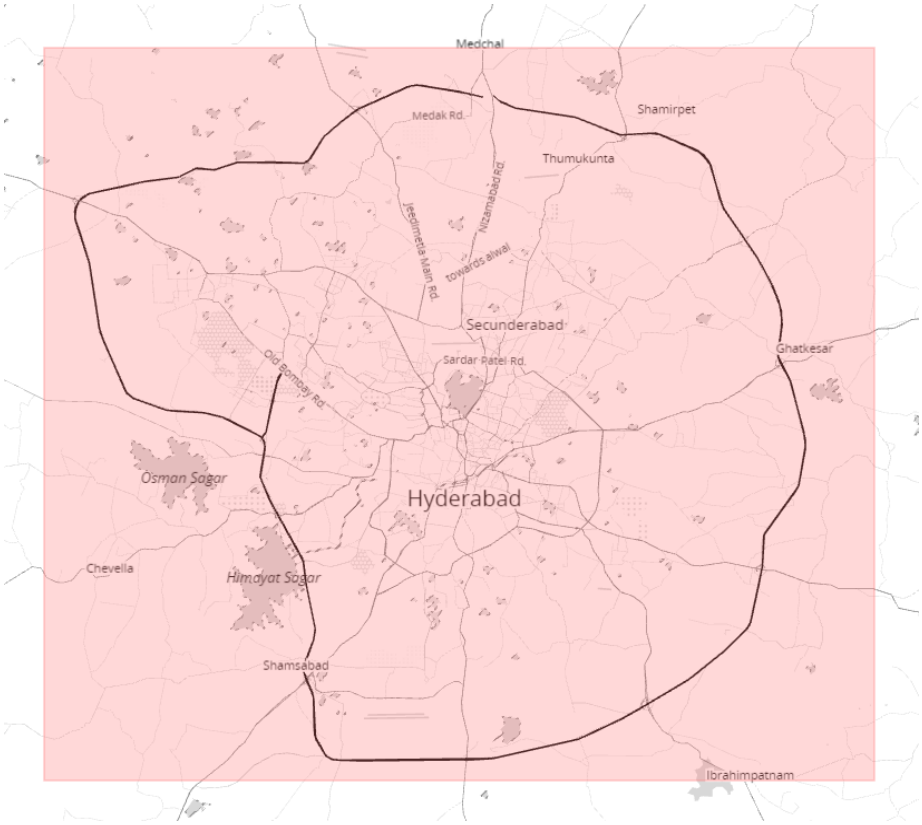
WRANGLE OPEN STREET MAP DATA

Submitted by: **Sai Saketh Boyanapalli**

In this project I will go through the process of data wrangling to audit, clean and set up this open street map data into SQL to uncover various insights about the area.

Area Selected**: Hyderabad**, Telangana, India



Why? I am from Hyderabad, I grew up there, I would love to learn more about my city and see if there is something interesting that I missed.

Data extracted from: https://mapzen.com/data/metro-extracts/metro/hyderabad_india/

Data Auditing

An Initial look:

I have looked at sample data and I see that there are many top-level tags in our data

Let's look at how many different top-level tags are available in our OSM file and their count.

For this I have used ***mapparser.py***

{'**bounds**': 1,

'**member**': 10304,

'**nd**': 4083254,

'**node**': 3230182,

'**osm**': 1,

'**relation**': 2318,

'**tag**': 866480,

'**way**': 771006}

We can see that's a very big file with file size over **700mb** and there are over 3.2 million node's, 700k nodes and over 2k relation's in our maps section.

What is a **node**? it's a single point in space defined by latitude and longitude. These points also have key, value pairs. features for the point are indicated by it.

What is a **way**? A way is simply collection of nodes, the tags here indicate the combined features of node.

Example.: key: amenity, value: school

Sample file

I have created a sample file using ***sample.py*** to see what kind of errors are present in our data.

I have used ***tags.py*** to see if there are any potential problems with tag 'key' (k) which are used to describe a topic, category, type or feature.

I will be using 3 regular expressions to see how many of the keys fall into these category

"lower", for tags that contain only lowercase letters and are valid,

"lower_colon", for otherwise valid tags with a colon in their names,

"problemchars", for tags with problematic characters, and

"other", for other tags that do not fall into the other three categories.

For the sample file I found only **2** tags with **problemchars** so, nothing major there.

Using Audit.py to look for error's in street addresses,

I have found some common inconsistencies here:

- 'St', 'St.' instead of 'Street'
- 'Rd', 'Rd.', 'road' instead of 'Road'
- Many Lower-case instances such as 'society', 'colony', 'office'

All the errors found in the sample set were fixed using the **update_name** function from **audit.py**

Implementing fixes

Now that our code works let's see what kind of errors we come across when running through

Accessing Problems

*Using **tags.py*** this time on whole data.

We can see what pattern the value of each 'k' value falls into

```
process_map(OSM_FILE)
```

```
{'lower': 861095, 'lower_colon': 5080, 'other': 257, 'problemchars': 48}
```

"lower": 861095, tags that contain only lowercase letters and are valid,

"lower_colon": 5080, for otherwise valid tags with a colon in their names,

"problemchars": 48, for tags with problematic characters, and

"other": 257, for other tags that do not fall into the other three categories.

Accessing Inconsistencies

Using ***audit.py*** to look at any inconsistencies with street names

- 'street' instead of 'Street'
- 'x-roads', 'x;road' instead of 'Cross Road' junctions in Hyderabad are usually called this way.
- 'raod', 'RD', 'ROAD' for 'Road'
- 'No-92', 'No.3', 'No.7' different types of road numbers these needed to be changed to 'Number 92'
- 'Marg' definition of Road in 'Hindi' one of our local languages
- 'Hydera' shortly spelling Hyderabad common in some areas
- 'Gachibowlo', 'Balkumpet' another type error for 'Gachibowli', 'Balkampet'

- ‘Colony, ’ another minor error for ‘Colony’
- ‘COLONY,AMBERPET’ this one is all caps and there is no space between two there is only one such error.

These are all the errors I could find in case of street addresses, given that this is a huge city I am surprised to see only few errors.

I will update all the errors with the corrected names in mapping variable than using the function ***update_name*** from ***audit.py*** and write changes to the osm file using the **data.py**

In total there were 28 changes.

Postal Codes

Next, I have updated my ***audit.py*** to see if there are any potential problems with postal code.

I have first run it on sample.osm to see if the code is working and saw that they were 2 errors.

- ‘50’ – this one is tricky the postal codes for Hyderabad are 6 digit’s starts with 500 so, this one can be start or ‘500 050’ shortly written has 50.
- ‘50046’ – In general India does not have postal codes with 5 digits’ so, this can be anything.

Now that my code was working I have run **audit.py** on entire data. This time I see 2 different errors.

- ‘500 032’ there is nothing wrong with this one, this style is commonly adapted in India and can be fixed.
- More than 6 digits like ‘5021377’, ‘5000000’, ‘34500034’ I am guessing these to be 502137, 500000, 500034 but cannot be certain for sure.

users.py to see no of unique users contributed to Hyderabad area Map.

And we see that there were **1028** unique users who have contributed. I am surprised by this no given the vast data that we have for this city.

Once we have gotten an idea on what data we have. I have converted the osm data to csv files using ***data.py*** and ***schema.py,*** This converts the data into 5 csv files which are imported into python and then a database is created using them.

- **Hyderabad_sample.osm** osm data file size 72Mb
- **nodes.csv** information on nodes file size 26Mb
- **node_tags.csv** information on node tags file size 89Kb
- **ways.csv** information on ways file size 4.6Mb
- **ways_nodes.csv** information on way nodes file size 9.6Mb
- **way_tags.csv** information on way tags file size 2.8Mb

Using the ***CsvtoDatabase.py*** file I have created ***hyderabad.db*** data base file of size 38MB.

Data Overview and Additional Ideas

*** The analysis was performed on sample file as the main file was too big to handle.

NO OF UNIQUE USERS

```
'SELECT COUNT(DISTINCT(uid)) \
  FROM (SELECT uid FROM nodes\
  UNION ALL SELECT uid FROM ways);'
```

598 users have contributed for this sampled map of Hyderabad

NO OF NODES

```
'SELECT COUNT(*) FROM nodes'
```

NO OF WAYS

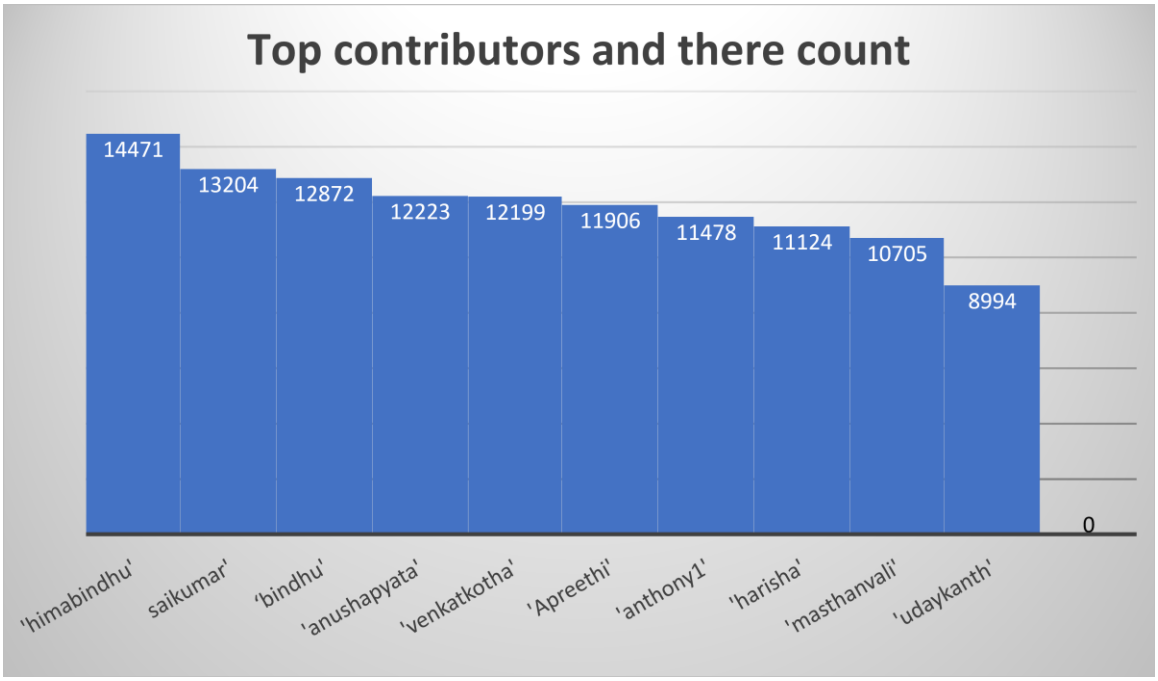
```
'SELECT COUNT(*) FROM ways'
```

77100

TOP CONTRIBUTIONS

```
'SELECT user, COUNT(*) as num \
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) \
GROUP BY user \
ORDER BY num DESC \
LIMIT 10;'
```

Top Contributors	Contribution
'himabindhu'	14471
saikumar'	13204
`bindhu'	12872
'anushapyata'	12223
'venkatkotha'	12199
'Apreethi'	11906
'anthony1'	11478
'harisha'	11124
'masthanvali'	10705
'udaykanth'	8994



ONE THING TO NOTICE IS THAT THE DISTRIBUTION IS NOT SKEWED

USERS CONTRIBUTING JUST ONCE

```
'SELECT COUNT(*) \
FROM (SELECT user, COUNT(*) as num \
    FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) \
    GROUP BY user \
    HAVING num=1);'
```

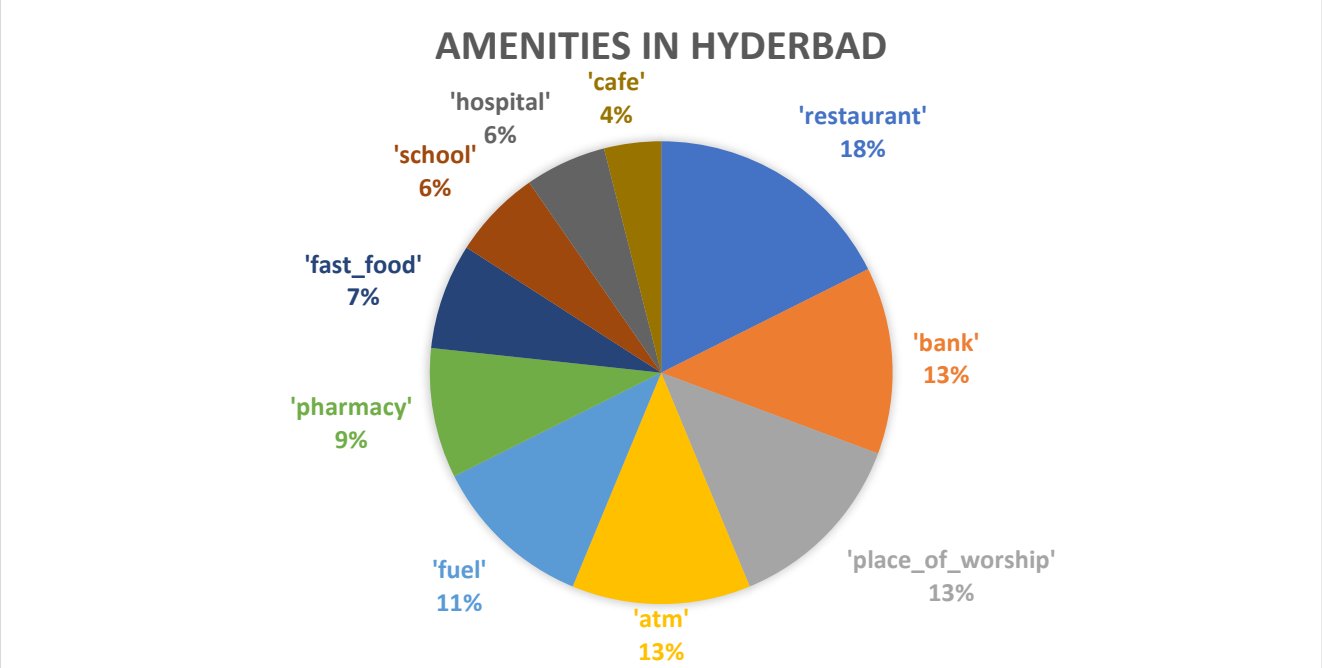
187 User's out of 598 users contributed just once.

ADDITIONAL INFORMATION

AMENITIES

```
'SELECT value, COUNT(*) as num \
FROM nodes_tags \
WHERE key = "amenity" \
GROUP BY value \
ORDER BY num DESC \
```

Amenities	Count
'restaurant'	31
'bank'	23
'place_of_worship'	23
'atm'	22
'fuel'	20
'pharmacy'	16
'fast_food'	13
'school'	11
'hospital'	10
'cafe'	7



MOST FOLLOWED RELIGION

```
'SELECT nodes_tags.value, COUNT(*) as num \
FROM nodes_tags \
JOIN (SELECT DISTINCT(id) \
   FROM nodes_tags WHERE value="place_of_worship") i \
   ON nodes_tags.id=i.id \
WHERE nodes_tags.key="religion" \
GROUP BY nodes_tags.value \
ORDER BY num DESC \
LIMIT 1;'
```

AS I FROM HYDERABAD, I AM CERTAINLY SURE THERE ARE PLENTY OF CUISINES LET'S SEE WHAT WE GOT HERE

CUISINES

```
'SELECT nodes_tags.value, COUNT(*) as num \
FROM nodes_tags \
   JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value="restaurant") i\
   ON nodes_tags.id=i.id \
WHERE nodes_tags.key="cuisine" \
GROUP BY nodes_tags.value \
ORDER BY num DESC;'
```

CUISINE	COUNT
'indian'	3
'regional'	3
'chinese'	2
'South_Indian'	1
'asian;noodles;chinese'	1
'fish;chicken;kebab;grill'	1
'fish;noodles;chicken;kebab;vegetarian'	1
'indian;regional;chinese'	1
'sushi'	1

Just looking at the count this makes me question the completeness and accuracy of the data. One of the main thing that Hyderabad stands apart is its diverse people from all over the country migrated to this place so, there are numerous cuisines available in Hyderabad and to my surprise we can see that here the data is incomplete.

Looking at the cuisines we can see that there is food menu in there instead of cuisine itself. This tells us that the data is not accurate.

Conclusion:

After going through the process of data wrangling I can say that what data we have is well structured, there were only few errors, when we looked at processing street addresses. But what I question is the **accuracy**, **completeness** and **validity** of the data we have for Hyderabad. Given the vast city and we look at few information we have for cuisines, amenities, religion that we have obtained from the data it is clear that the data is not updated properly.

Given that I have sampled through only 10% of the original data. As I have lived there for almost 20 years I can certainly say that there is lot more to it.

Another interesting thing there were only 598 contributors of which 187 have contributed just once that means 411 users have contributed for large amount of data. Top 10 users have contributed more than 30% of data.

Dealing with postal codes to be accurate I must look at the latitude and longitude coordinates and impute the correct code for the ones which are incorrect.

One thing is clear open street map is lacking strong user base in Hyderabad.

It should come up with innovative ideas to encourage users to contribute.

To fix errors it should include some restrictions with data entry.

Currently google maps is very popular and has tons of information which is very accurate. So, open street map should come up with functionality that google maps does not provide, like maps in regional language.

The problem with implementing this suggestion is one needs to convert the data to regional language which is very tedious and requires someone who has lived there and is fluent with the language.

FILES USED

- *sample.py* - This is used to convert main osm file and sample it to take every kth level element.
- *mapparser.py* – To see different top-level elements in our file
- *tags.py* – To look at what pattern's the k tag's and value fall into
- *audit.py* – To address inconsistencies with street address
- *data.py* - To convert xml to csv using schema
- *schema.py* – Schema used for creating csv files
- *CSVtoDatabase.py* – Converting CSV files to single database
- *query.py* – All queries performed on above created database
- *user.*py – To count unique user id's in osm file

References:

https://mapzen.com/data/metro-extracts/metro/hyderabad_india/

https://gist.github.com/swwelch/f1144229848b407e0a5d13fcb7fbbd6f

https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md

https://stackoverflow.com/questions/2887878/importing-a-csv-file-into-a-sqlite3-database-table-using-python

https://pythonspot.com/en/mysql-with-python/

http://support.esri.com/en/technical-article/000011656