OpenStreetMap Data - Case Study

Map Area

Bengaluru, India

I've been working in this city for the last two years as a data analyst. Was really curious to see what the OSM data looks like here

Problems encountered in the map

- Incorrect postal codes. ie 6 digit codes which do not start with 56
- Inconsistent postal codes. ie postal codes with characters and spaces
- City name - The majority of the city names had "Bangalore' as value.. City name was changed in 2006 from Bangalore to Bengaluru.

Incorrect postal codes + Inconsistent postal codes

Came across different types of erroneous zip codes while looking through the data.
We saw city names and other phrases creep in: eg. 'Bengaluru', 'iam in bang',
we saw numbers less than 6 digits (indian zip codes are 6 digits) : eg. '79'
we saw numbers with other characters: eg. '560001ph'
we saw numbers in other languages: eg. '೫೬೦೦೬೦'

When storing data in csv, we took care of some of the above problems by
taking only digits from the 'addr:postcode' string values --> Making sure they are not empty strings --> Type casting them into integers and then back to string; to take care of numbers in local languages --> Then whould check if length of string is equal to six and the first two characters are '56', if yes, we would return the processed value. Else, that tag would be ignored.

In [1]:
```
def audit_and_clean(postcode):
    onlydigits = ''.join(re.findall("\d+", postcode))
    if(onlydigits!=''):
        onlydigits = str(int(onlydigits))
        if(len(onlydigits)==6 and onlydigits[:2]=='56'):
            return onlydigits
```

City name not being "Bengaluru"

in the addr:city fileds, the top counts were for :

[('Bangalore', 4965),
('Bengaluru', 1262),
('bangalore', 1233),
('bengaluru', 144),
('BANGALORE', 104),
('BENGALURU', 67)]

In 2006, the city name was changed from 'Bangalore' to 'Bengaluru'. While storing data as csv. we stored them as 'Bengaluru'

In [ ]:
```
if (element.tag =='way' or element.tag =='node'):
    for child in element:
        if(child.tag == 'tag'):
            kval = child.attrib['k']
            if(PROBLEMCHARS.match(kval)):
                continue
            vval = child.attrib['v']
            if(kval == 'addr:city'):
                vval = 'Bengaluru'
```

Data overview and additional ideas

File sizes

bengaluru_india.osm --> 620 MB

nodes.csv -----------> 233 MB
nodes_tags.csv ----> 3.70 MB
ways.csv ------------> 39.1 MB
ways_nodes.csv ---> 85.4 MB
ways_tags.csv -----> 23.7 MB

Number of unique users

In [ ]:
```
#Number of unique users
SELECT COUNT(DISTINCT user) FROM
(SELECT user FROM nodes UNION SELECT user FROM ways) AS sub;
```

COUNT(DISTINCT user)
2032

Number of nodes

In [ ]:

```
#Number of nodes
select count(*) from nodes;
```

COUNT(*)
2887846

Number of ways

In [ ]:

```
#Number of ways
select count(*) from ways;
```

COUNT(*)
661844

Top amenities in the city with count

In [ ]:

```
#Top 10 amenities with count in the city
SELECT COUNT(value) AS cnt, value
FROM
(SELECT value FROM nodes_tags
WHERE `key` = 'amenity'
 UNION ALL
SELECT value FROM ways_tags
WHERE `key` = 'amenity') as tags_amenity
GROUP BY value
ORDER BY cnt DESC
LIMIT 10;
```

1774 restaurant
1099 place_of_worship
823 atm
816 bank
716 school
591 hospital
561 pharmacy
557 fast_food
371 cafe
326 fuel

Top cuisines in the city

In [ ]:

```
#Top 10 cuisines with count in the city
SELECT COUNT(value) AS cnt, value
FROM
(SELECT value FROM nodes_tags
WHERE `key` = 'cuisine'
 UNION ALL
SELECT value FROM ways_tags
WHERE `key` = 'cuisine') as tags_cuisine
GROUP BY value
ORDER BY cnt DESC
LIMIT 10;
```

391 regional
312 indian
92 pizza
90 vegetarian
81 chinese
58 ice_cream
54 coffee_shop
46 burger
35 international
28 italian

Its very interesting to note that we see more 'pizza' places than 'vegetarian' places in an Indian city.

Top contributing users to the OSM bangalore dataset

In [ ]:

```
#Top 10 contributing users
SELECT user, COUNT(user) cnt FROM
(SELECT user from nodes UNION ALL
SELECT user FROM ways) sub
GROUP BY user
ORDER BY cnt DESC
LIMIT 10;
```

jasvinderkaur 124889
akhilsai 118677
premkumar 115877
saikumar 114906
shekarn 98116
PlaneMad 95053
vamshikrishna 94258
himalay 88176
himabindhu 86842
sdivya 84983

Additional Exploration and comments

Most widely seen postcodes

In [ ]:

```
#Most widely seen postcodes
SELECT tags.value, COUNT(*) as count
FROM
(SELECT * FROM nodes_tags
UNION ALL
SELECT * FROM ways_tags) tags
WHERE tags.key='postcode'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 5;
```

value -- count
560066 - 271 ---> Bangalore East
560037 - 234 ---> Bangalore East
560003 - 202 ---> Bangalore North
560103 - 181 ---> Bangalore South East
560040 - 161 ---> Bangalore North West

Its observed that the top 5 most widely repeating zip codes (implicitly implying larger areas covered) are not in the central, heart of the city. These areas have developed in recent times. Earlier large outskirts areas would have come under one or two post offices. Then as the area developed, more buildings, offices and residential areas would have come up, creating more tags in the OSM dataset. The top two postcodes are in Bangalore East, which is the home to most of the tech companies and IT proffessionals in the city.

Max speeds in bangalore

```
In [ ]: #Max speeds in bangalore
        SELECT tags.value, COUNT(*) as count
        FROM (SELECT * FROM nodes_tags
                  UNION ALL
              SELECT * FROM ways_tags) tags
        WHERE tags.key='maxspeed'
        GROUP BY tags.value
        ORDER BY tags.value DESC
        LIMIT 3;
```

value (kmph)	**count**
80	227
60	287
50	113

max speed limit observed in bangalore is 80 kmph. (100kmph being max speed limit on certain roads in India)

Most used editors in the city

```
In [ ]: #Most widely seen editors
        SELECT tags.value, COUNT(*) as count
        FROM (SELECT * FROM nodes_tags
                  UNION ALL
              SELECT * FROM ways_tags) tags
        WHERE tags.key='created_by'
        GROUP BY tags.value
        ORDER BY count DESC;
```

value ------------------- **count**
JOSM -------------------474
Potlatch 0.10f------------- 233
Potlatch 0.10e -----------------108
Potlatch 0.7b -------------------20
Potlatch 0.9a -------------------16
Potlatch 0.10d------------------- 10
Potlatch 0.9c -------------------5
Potlatch 0.9b -------------------4
Potlatch 0.10c------------------- 2
Potlatch 0.9 -------------------2
Merkaartor 0.12 -------------------1
Vespucci 0.6.5 -------------------1
iLOE 1.9 -------------------1
cap4access -------------------1

We see that JOSM and potlatch are the most used editors by users contributing

Most widely seen sources

In [ ]:

```
#Most widely seen sources of OSM data
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags
          UNION ALL
      SELECT * FROM ways_tags) tags
WHERE tags.key='source'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 5;
```

value ------------------- **count**
Bing ---------------------1199
bing sat ---------------------407
GPS --------------------- 262
survey --------------------- 216
landsat --------------------- 76

In November 2010 it was announced that Bing has granted the right to trace from their aerial imagery for the purpose of contributing content to OpenStreetMap.
In the most used editors (JOSM, potlatch 2, iD), Bing aerial imagery opens as background imagery.

Most widely seen religions

In [ ]:

```
#Most widely seen religion
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags
          UNION ALL
      SELECT * FROM ways_tags) tags
WHERE tags.key='religion'
GROUP BY tags.value
ORDER BY count DESC
LIMIT 3;
```

value -------------count
hindu ---------------666
christian ----------157
muslim ------------121

New editor encouragement problem

In [ ]:

```
#Number of unique users
SELECT COUNT(DISTINCT user) FROM
(SELECT user FROM nodes UNION SELECT user FROM ways) AS sub;
```

COUNT (distinct user)
2032

In [ ]:

```
#finding 1 edit users
select COUNT(user)
FROM
(SELECT user, COUNT(user) cnt FROM
(SELECT user from nodes UNION ALL
SELECT user FROM ways) sub
GROUP BY user
HAVING cnt = 1
ORDER BY cnt DESC) sub1;
```

COUNT
517

We see that almost one 1/4th of the unique users in the bangalore OSM dataset, only contributed 'once' to the OSM dataset.

```
In [ ]: #sum of all entries by users
        SELECT sum(cnt)
        FROM
        (SELECT user, COUNT(user) cnt FROM
        (SELECT user from nodes UNION ALL
        SELECT user FROM ways) sub
        GROUP BY user
        ORDER BY cnt DESC) sub1;
```

COUNT
3,549,690

```
In [ ]: #sum of TOP 20 contributors
        SELECT sum(cnt)
        FROM
        (SELECT user, COUNT(user) cnt FROM
        (SELECT user from nodes UNION ALL
        SELECT user FROM ways) sub
        GROUP BY user
        ORDER BY cnt DESC LIMIT 20) sub1;
```

COUNT
1,726,772

We see that almost 50% of the contributions were made by the top 20 contributors alone. ie
50% of the contributions were made by the top 1% of contributors alone.

OSM urges new people to contribute to the map. Editing the data can be challenging for a beginner. We see a lot of incorrect mapping and tag values. Once changes are made, they are applied without a review process. This may lead to bad edits being made to the map and may often be left undiscovered, if removed, the original editor does not usually know why. The latter can be very discouraging.

If a new map editor could contribute and make changes, and have the changes peer reviewed before being applied, we may have had higher quality data in the OSM project and a mentorship model between experienced and new contributors.