Ana Javed

OpenStreetMap: New Delhi Case Study

I. Map Location Information

New Delhi, India
Capital of India
Population: 317,797
Longitude: 77.166313
Latitude: 28.571945

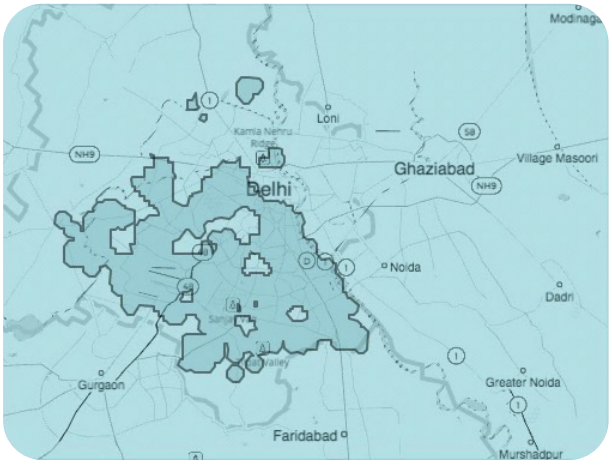

Image courtesy of Mapzen.com

New Delhi, India is where my family and I are from, and I wanted to further explore and wrangle the data available on OpenStreetMap.org. The map was created with a custom extract on Mapzen.com to include neighboring cities (such as Ghaziabad and Noida) in order to have a thorough outlook of the location.

The original .OSM file (670.1 MB) was shortened to a smaller file (27.1 MB) to audit and conduct analyses on. The four CSV files created to import the data into a SQL database were:

nodes.csv	8.2 MB
nodes_tags.csv	39 KB
ways.csv	1.1 MB
ways_tags.csv	889 KB

II. Specific Issues in the Dataset

a. **Different Languages:** There were a variety of languages encountered in the data, from Hindi to Russian, however English was predominately used by users editing the data. I anticipated this and to account for the discrepancy, I excluded any keys under the "node" tags that began with "name" and did not have ":en" (in indicate English) following after it. In addition, I parsed any outlier texts that were not in English, such as the left image below.

```
k: name v: Bay 6
k: name v: Bay 31
k: name v: Bay 131
k: name v: ต้องไปนอนแถวนี้
k: name:hi v: ต้องไปนอนแถวนี้
k: name v: Vijaylakshmi Apartment
k: name:en v: Saffron
k: name v: MLA office
k: name:en v: MLA office
```

```
name:diq Delhiyo Newe
name:dv [illegible]
name:el Νέο Δελχί
name:en New Delhi
name:eo Nov-Delhio
name:es Nueva Delhi
name:et New Delhi
name:eu New Delhi
name:ext Nueva Delhi
name:fa دهلی نو
name:fi New Delhi
name:fr New Delhi
```

b. **Tags with photo links:** Certain tags discovered in the data contained links referencing the shop or monument mentioned. I did not want to include such data in my queries (since it was not relevant to the information I was seeking) and they were excluded during the wrangling process.

```
image https://www.flickr.com/photos/131866138@N03/18430789763/in/datetaken/
image https://www.flickr.com/photos/131866138@N03/19025651076/in/datetaken/
image https://www.flickr.com/photos/131866138@N03/18951060170/in/dateposted-public/
image https://www.flickr.com/photos/131866138@N03/1913
image https://www.flickr.com/photos/131866138@N03/18952763030/in/dateposted-public/
image https://www.flickr.com/photos/131866138@N03/19010334759/in/dateposted-public/
image https://www.flickr.com/photos/131866138@N03/19199901451/in/dateposted-public/
image https://www.flickr.com/photos/131866138@N03/19196207935/in/dateposted-public/
image https://www.flickr.com/photos/131866138@N03/19290428241/in/datetaken/
image https://www.flickr.com/photos/131866138@N03/19280598012/in/datetaken/
image https://www.flickr.com/photos/131866138@N03/18777871813/in/dateposted-public/
image https://www.flickr.com/photos/131866138@N03/20677635029/in/dateposted-public/
```

c. **Inconsistent names for values:**
In order to complete meaningful analyses and submit applicable queries in SQL, amenities with varying names but of the same type were converted to a standard name. For example, “Axis Bank”, “Citibank”, and “ICICI Bank” values were converted to the string “bank”. Other amenities that were standardized include schools, pharmacy, cinemas, and hospitals.

This was also done for tags describing the “state” value. “NCR” appeared many times, along with the abbreviation DL, and stands for National Capital Region. These values were also converted since “NCR” is a metropolitan area name for Delhi.

```
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: NCR
addr:state v: NCR
addr:state v: DL
addr:state v: DL
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
addr:state v: Delhi
```

```
Axis Bank
bank
Axis Bank
Citibank
bank
HDFC Bank
CITI Bank
bank
ICICI Bank
bank
Andhra Bank
bank
State Bank of India
bank
HDFC Bank
bank
bank
bank
OBC Bank
bank
HDFC Bank
```

d. **Phone number audits:** For this process, regular expressions or regex was utilized to ensure the numbers had a standard format. In India, phone numbers contain 12 digits that are preceded by a plus sign, in contrast to the United State’s 10 number format. Below is the code utilized to conduct the audit and to clean the phone numbers to have no spaces or dashes.

```
    if re.match(r'^\+([0-9]+\s)*', phone_value):
        if "-" in phone_value:
            sep_dashes= phone_value.split("-")
            newphone="".join(sep_dashes)
            each_dict['value']=newphone
        elif " " in phone_value:
            sep_spaces= phone_value.split(" ")
            newphone="".join(sep_spaces)
            each_dict['value']=newphone
```

```
'+91-120-3997600',
'+91 97 17 896108'
'+911244541800',
```



```
+911203997600
+919717896108
+911244541800
```

III. SQL Queries

a. Counting Data From "nodes" and "ways" tags:

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

COUNT(*)
121458

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

COUNT(*)
24993

b. Top 10 Amenities Query

Value	Num
Bank	12
Fuel	12
Food	11
ATM	6
Place of Worship	6
Cafe	4
Hospital	4
Cinema	3
Embassy	3
Park	3

```
SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10;
```

This query demonstrates the significance of standardizing and consolidating data in the "amenity" key, since it allow for proper counts. From this, it is easily determined that banks, fuel stations, and food locations are most abundant in the mapped area.

c. Top Postcodes:

Value	Num
"110087"	"19"
"100006"	"3"
"110075"	"3"
"122001"	"3"
"110019"	"2"

```
SELECT tags.value, COUNT(*) as num
FROM (SELECT * FROM nodes_tags UNION
ALL SELECT * FROM ways_tags) tags
WHERE tags.key='postcode'
GROUP BY tags.value
ORDER BY num DESC;
```

Since I was including outer cities into the data, I wanted to examine which postcodes were most represented. "110087" is New Delhi, India and the most occurring. The following postcodes also include South West Delhi and Gurugram (another district in the map data), however "100006" is the postcode for Beijing, China and should be excluded from the dataset in the future.

d. Number of Unique Users

COUNT(DISTINCT(uid))
512

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

This is significant information because it illustrates how a total of 512 distinct user entered information on OpenStreetMap to makeup the data contained in the shortened XML file.

e. Top 10 Users for "ways" and "nodes" Tables

Ways:

User	num
"Oberaffe"	"1410"
"premkumar"	"1349"
"saikumar"	"1153"
"sdivya"	"1038"
"anthony1"	"927"
"anushap"	"876"
"sathishshetty"	"832"
"himabindhu"	"786"
"Apreethi"	"716"
"harisha"	"674"

```
SELECT user, COUNT(*) as num
FROM ways GROUP BY user
ORDER BY num desc limit 10
```

Nodes:

User	num
"Oberaffe"	"6563"
"premkumar"	"4841"
"Naresh08"	"4592"
"saikumar"	"4422"
"anushap"	"4204"
"sdivya"	"4164"
"anthony1"	"3759"
"himabindhu"	"3616"
"pvprasad"	"3518"
"Apreethi"	"3359"

```
SELECT user, COUNT(*) as num
FROM nodes GROUP BY user
ORDER BY num desc limit 10;
```

Users "Oberaffe" and "premkumar" mostly entered the data in both tables. Other users that appear in both tables are "saikumar", "sdivya", "Apreethi", "anthony1", and "himabindhu".

f. Years When Data was Most Entered

Timestamp	Num
"2015"	"80527"
"2016"	"28439"
"2012"	"4301"
"2014"	"3302"
"2010"	"1944"

```
SELECT timestamp, COUNT(*)
as num  FROM nodes GROUP
BY timestamp ORDER BY num
DESC LIMIT 5;
```

When wrangling and cleaning the data, I converted the timestamp values to display only the year in order to determine when data was most entered. It is not surprising that 2015 and 2016 are the

top two years since technology and the Internet becomes more widely available to remote areas as time continues.

g. Top 5 Region Types

Value	Num
"locality"	"33"
"Sunder Vihar" (residential neighborhood)	"20"
"suburb"	"6"
"village"	"5"
"hamlet"	"2"

```
SELECT tags.value, COUNT(*) as
num  FROM (SELECT * FROM
nodes_tags UNION ALL SELECT *
FROM ways_tags) tags WHERE
tags.key= 'place' GROUP BY
tags.value ORDER BY num DESC
limit 5;
```

This information provides insight regarding what types of regions exist in the mapped area. Localities can often mean cities in India, and it is most occurring type in the list. Second is a residential neighborhood called “Sunder Vihar”. Interestingly villages are the fourth most occurring regions, which can indicate that rural areas are still in proximity of the capital.

IV. Limitations

There were portions of the dataset that were not wrangled or standardized completely, due to the objectives of this case study. To improve the data and its analysis for further use, I recommend conducting thorough cleaning and wrangling. Not only could more SQL queries be conducted, but the dataset would also be more concise.

In addition, I also advocate for OpenStreetMap to adapt a method to standardize the data inserted by users to either ensure the information is all in one language or is encoded if there are ascii characters. One benefit of this implementation is that it would enable all users to insert data comfortably for international areas where the predominant language is not English. However a foreseeable challenge to this implementation is when words of other languages do not have an exact translation in English and lose its meaning in the process.

V. Sources

a. https://mapzen.com/data/metro-extracts/wof/102029189.geojson
b. https://mapzen.com/data/metro-extracts/
c. http://www.openstreetmap.org/#map=11/28.6701/77.3513
d. https://en.wikipedia.org/wiki/National_Capital_Region_(India)
e. https://developers.google.com/edu/python/regular-expressions