Milan Patel

Selected map area: Bogotá, Colombia

(https://s3.amazonaws.com/metro-extracts.mapzen.com/bogota_colombia.osm.bz2)

Overview

I selected the city of Bogotá for a number of reasons. First of all, I know the city fairly well, having lived there for 4 months. I have a specific interest in the developing world, and understanding & building the potentially vast but uncollected data for those parts of the world. Compared to most well-developed countries, which have a high proportion of the population that are connected to the Internet who serve as active users and contributors, cities in developing countries have fewer users to develop and utilize the content. A cursory read of the source website above reveals this gap in available data: the size of the zipped XML file for Bilbao, Spain (which has a metro population of almost 1 million and an urbanized area of 17.35 km^2)[i], is 14.4 MB; Bogotá, which has a population of almost 8 million and urbanized area of 307 km^2,[ii] has a zipped XML file size of 8.4MB. The potential for collecting additional data is clear, and it is likely that the existing data set would benefit from a thorough analysis and cleanse.

Problems Encountered in the Map

Through an audit of the data, I encountered the following issues with the data:

- Abbreviated street names
- Incomplete street names
- Misuse of street address field to contain other types of data

Overview of Analysis process

Using the audit code to analyze the data for Bogota required a number of modifications to apply to the context of the city. First of all, since the street name data is in Spanish, the list of street types is different than that of the US; there are no “Streets” or “Avenues”, but there are many “Calles” and “Avenidas”. I had to iteratively run the audit code to identify the complete list of street types, and build the “Expected” list. Second, the street type comes at the beginning of the name instead of the end (“Calle 1” instead of “1st Street”), and I had to amend the regular expression to adjust for this fact.

Modified code is contained in “Bogota – Initial analysis.py”.

Problem: Abbreviated Street Names

I used the audit() function to show all street names that are not in the “expected” list—often because an abbreviation was used for the street type.

This table shows the number of standard street types that were correct before and after cleansing:

Street type	Before cleanse	After cleanse
Autopista	2	2
Avenida	42	55
Calle	162	334
Carrera	128	174
Diagonal	5	5
Transversal	3	3

The statistics for the “before” and “after” cleanse indicates that there are a number of items that use different variations of the expected street type. Creating a mapping library for these streets and cleansing the street names significantly improves the consistency of the data.

This “mapping” process is implemented in “Cleanse data and create JSON.py”, which modifies the street name when the JSON is created.

Incomplete Street Names

Note that the cleanse was not able to completely eliminate all issues. Correcting the remaining issues will require manual analysis and correction—for example, review of other map sources, or perhaps even looking at the OpenStreetMap and considering the value in context to make a quick correction. For example, there is a location whose street is simply called “11”; it is likely that this location is a “Calle 11” or “Avenida 11”, but further information is required to make a correction.

Misuse of street address field to contain other types of data

The street address field often contains data that clearly is not street address data.

For example, 16 distinct URLs and 9 email addresses were populated into that field.

The “mapping” process which updates the street names for abbreviations also corrects for this issue.

2. Data Overview

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

File sizes

```
Bogota_colombia.osm......... 113 MB
Bogota_colombia.osm.json .... 125 MB
```

# Number of documents

```
Query:db.streetdata.find().count()
Result: 587285
```

# Number of nodes

```
Query: db.streetdata.find({"type":"node"}).count()
Result: 469522
```

# Number of ways

```
Query: db.streetdata.find({"type":"way"}).count()
Result: 117763
```

# Number of unique users

```
Query: db.streetdata.distinct({"created.user"}).length
Result: 241
```

# Top 1 contributing user

```
Query: db.streetdata.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":1}])
Result:  [{u'count': 79840, u'_id': u'Federico Explorador'}]
```

With 79840 contributions, user Federico Explorador is top contributor.

# Number of users appearing only once (having 1 post)

```
Query: db.streetdata.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}},{"$group":{"_id":"$count",
"num_users":{"$sum":1}}}, {"sort":{"_id":1}}, {"$limit":1}])
```

```
Result: {u'ok': 1.0, u'result': [{u'num_users': 241, u'_id': 1}]}

241 users appearing only once
     # “_id” represents postcount
```

3. Additional Ideas

Improving details about amenities serving food

A number of amenities are establishments that serve food; such amenity types include “restaurant” (576 instances) “fast_food” (225 instances), and “cafe” (163 instances).

A key tag associated with this amenity is “cuisine”. Not all food-serving amenities have “cuisine” information; 66% of restaurants, 62% of fast food establishment, and 86% of cafes do not have cuisine information.

This data can be improved using the following methods:

Ensure cuisine categorization is internally consistent

The most numerous cuisine type within the “restaurant” categorization is “regional”; either these restaurants are from other regions in the world (which each have their own cuisine types), or is regional cuisine of Colombia (more likely). Other cuisines that are probably Colombian food are “colombiana” and “Typical” (used in “fast food” amenity category). Most likely, these should be placed in a “Colombian” food category.

Decide whether amenity category should be reclassified

Specifically, it should probably be decided if a clearer distinction between amenity “cafe” and “restaurant”; if a café is serving food, perhaps it should be classified as a “restaurant”. Note that the vast majority of locations with amenity “café” have cuisine “coffee shop”; perhaps cafes should be limited to places that primarily serve coffee.

Use the name of the amenity to determine what the cuisine of the restaurant is

This process may imply a very manual process; however, perhaps some insights can be obtained easily for a fair percentage of those amenities. For example, if an amenity has “pollo” in the name, it is almost certain to be a chicken-focused establishment. Similarly, “hamburguesas” or “burger” indicates a place for burgers.

The following query results indicates the restaurants with “pollo” in the name; these can probably be classified with cuisine “chicken”.

```
[{u'count': 1, u'_id': u'Asadero Kroc Pollo'}, {u'count': 1, u'_id': u'CombiPollo'}, {u'count': 1, u'_id': u'Mc Pollo'}, {u'count': 1, u'_id': u'Pollo Pizza Carne PPC'}, {u'count': 1, u'_id': u'El portal del pollo'}, {u'count': 1, u'_id': u'El Pollo Exitoso'}, {u'count': 1, u'_id': u'Pollomax'}, {u'count': 1, u'_id': u'El Rey del Pollo'}]
```

Additional data exploration using MongoDB queries

Top 20 amenities

Query: `db.streetdata.aggregate([{"$match":{"amenity":{"$exists":1}}}, {"$group":{"_id":"$amenity", "count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":20}])`

Results:
```
[{u'count': 841, u'_id': u'parking'}, {u'count': 790, u'_id': u'school'}, {u'count': 576, u'_id': u'restaurant'}, {u'count': 409, u'_id': u'fuel'}, {u'count': 366, u'_id': u'place_of_worship'}, {u'count': 365, u'_id': u'bank'}, {u'count': 290, u'_id': u'pharmacy'}, {u'count': 225, u'_id': u'fast_food'}, {u'count': 168, u'_id': u'hospital'}, {u'count': 163, u'_id': u'cafe'}, {u'count': 137, u'_id': u'police'}, {u'count': 127, u'_id': u'university'}, {u'count': 104, u'_id': u'bar'}, {u'count': 98, u'_id': u'college'}, {u'count': 95, u'_id': u'atm'}, {u'count': 67, u'_id': u'public_building'}, {u'count': 63, u'_id': u'bicycle_parking'}, {u'count': 62, u'_id': u'bus_station'}, {u'count': 59, u'_id': u'library'}, {u'count': 50, u'_id': u'theatre'}]
```

Data for analysis of cuisines

Amenities with cuisine populated

```
{u'count': 198, u'_id': u'restaurant'}, {u'count': 84, u'_id': u'fast_food'}, {u'count': 22, u'_id': u'cafe'}, {u'count': 1, u'_id': None}, {u'count': 1, u'_id': u'pub'}, {u'count': 1, u'_id': u'bar'}]
```

Cuisines for each amenity type

Restaurant:

Query: db.streetdata.aggregate([{"$match":{"cuisine":{"$exists":1}, "amenity":"restaurant"}}, {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":20}])

Result:
Restaurants by cuisine: [{u'count': 32, u'_id': u'regional'}, {u'count': 20, u'_id': u'pizza'}, {u'count': 20, u'_id': u'vegetarian'}, {u'count': 17, u'_id': u'steak_house'}, {u'count': 15, u'_id': u'burger'}, {u'count': 10, u'_id': u'italian'}, {u'count': 8, u'_id': u'chicken'}, {u'count': 7, u'_id': u'chinese'}, {u'count': 6, u'_id': u'international'}, {u'count': 6, u'_id': u'french'}, {u'count': 5, u'_id': u'asian'}, {u'count': 5, u'_id': u'mexican'}, {u'count': 3, u'_id': u'japanese'}, {u'count': 3, u'_id': u'fish'}, {u'count': 3, u'_id': u'spanish'}, {u'count': 2, u'_id': u'peruvian'}, {u'count': 2, u'_id': u'colombiana'}, {u'count': 2, u'_id': u'arab'}, {u'count': 1, u'_id': u'thai'}, {u'count': 1, u'_id': u'arepas.'}]

Fast food:

Query:
db.streetdata.aggregate([{"$match":{"cuisine":{"$exists":1},"amenity":"fast_food"}}, {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":20}])

Results:

Fast food cuisine: [{u'count': 39, u'_id': u'burger'}, {u'count': 23, u'_id': u'pizza'}, {u'count': 6, u'_id': u'sandwich'}, {u'count': 5, u'_id': u'chicken'}, {u'count': 5, u'_id': u'mexican'}, {u'count': 2, u'_id': u'regional'}, {u'count': 2, u'_id': u'vietnamese'}, {u'count': 1, u'_id': u'Typical'}, {u'count': 1, u'_id': u'italian'}]

Cafe:

Query:

db.streetdata.aggregate([{"$match":{"cuisine":{"$exists":1}, "amenity":"cafe"}}, {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":20}])

Results:

Cafe by cuisine: [{u'count': 8, u'_id': u'coffee_shop'}, {u'count': 2, u'_id': u'spanish'}, {u'count': 2, u'_id': u'Cafeteria'}, {u'count': 2, u'_id': u'ice_cream'}, {u'count': 2, u'_id': u'international'}, {u'count': 1, u'_id': u'coffee_shop,_bread_store'}, {u'count': 1, u'_id': u'almojabanas'}, {u'count': 1, u'_id': u'Donuts'}, {u'count': 1, u'_id': u'pasteleria'}, {u'count': 1, u'_id': u'Chocolate_shop'}, {u'count': 1, u'_id': u'regional'}]

Restaurants without cuisine attribute with “pollo” in name

Query:
db.streetdata.aggregate([{"$match":{"cuisine":{"$exists":0}, "amenity":"restaurant", "name":{"$regex": "[Pp]ollo"}}},

```
{"$group":{"_id":"$name", "count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":20}])
```

Result:

```
[{u'count': 1, u'_id': u'Asadero Kroc Pollo'}, {u'count': 1, u'_id': u'CombiPollo'}, {u'count': 1, u'_id': u'Mc Pollo'}, {u'count': 1, u'_id': u'Pollo Pizza Carne PPC'}, {u'count': 1, u'_id': u'El portal del pollo'}, {u'count': 1, u'_id': u'El Pollo Exitoso'}, {u'count': 1, u'_id': u'Pollomax'}, {u'count': 1, u'_id': u'El Rey del Pollo'}]
```

Conclusion

This Project Overview demonstrates that the data for Bogotá would benefit from cleansing and enhancement. Specifically, the street name data can be corrected by changing street abbreviations to the full street name. Additionally, MongoDB can be used to search through the names of eating establishments to identify the cuisine served.

The analysis that leads to these findings only scratches the surface of the data: further analysis will likely find many other cleaning opportunities.

[i] http://en.wikipedia.org/wiki/Bilbao
[ii] http://en.wikipedia.org/wiki/Bogotá