Project 3 Resubmittal
Data Wrangling with MongoDB

Derek Gurney

Map Area: Vancouver, BC, Canada

https://mapzen.com/data/metro-extracts/#vancouver-canada

Note: updates are in "Overview of the Data" and "Other ideas about the datasets"

1. Problems encountered in my map

An inspection of the data revealed several problems with the map data: abbreviated, inconsistent, or misspelled street types; misspelled city names; inconsistent province name; and misformatted postal codes. All of these were cleaned programmatically; additional issues are discussed below.

Abbreviated, inconsistent, or misspelled street types

Street types in the data set were abbreviated; for example, "Street" was represented as "St." The abbreviations were also inconsistent, with "Street" sometimes abbreviated as "St" and sometimes as "St." Lastly, street types were sometimes misspelled; for example "venue" instead of "Avenue."

Misspelled city names

"Vancouver" was sometimes spelled as "Vancovuer" and was sometimes uncapitalized.

Inconsistent province name

“British Columbia” was alternately represented as “BC”, “B.C.”, and “bc”.

Misformatted postal codes

Postal codes in Canada are formatted as “LetterNumberLetter NumberLetterNumber”, but many of the postal codes were represented either without capitalization or without the middle space.

Unaddressed issues

A number of other issues were left unaddressed: city names that are outside of the map area (e.g. Dresden), provinces outside map area (e.g. Ontario), and incomplete postal codes (e.g. V6C).

2. Overview of the Data

File sizes

Vancouver_canada.osm: 175.8 MB
Vancouver_canada.osm.json: 203.3 MB

Number of documents

```
> db.yvr.find().count()
3235218
```

Number of nodes

```
> db.yvr.find({"type":"node"}).count()
2772348
```

Number of ways

```
> db.yvr.find({"type":"way"}).count()
462801
```

Number of unique users

```
> db.yvr.distinct("created.user").length
1147
```
https://workflowy.com/#/6c1576358cfd

Number of cities

```
> db.yvr.distinct("address.city").length
24
```

Number of documents with amenity = “cafe”

```
> db.yvr.find({"amenity":"cafe"}).count()
353
```

Number of cafes, found a different way

```
> db.yvr.distinct( "id" ,{"amenity" :"cafe"}).length
353
```

Number of Crematoriums

```
> db.yvr.distinct( "id" ,{"amenity" :"Crematorium"}).length
1
```

Number of Starbucks (or Starbucks’)

```
> db.yvr.find({"name":/^Starbucks/}).count()
81
```

Number of Tim Hortons, a competing chain

```
> db.yvr.find({"name":/^Tim Hortons/}).count()
23
```

How are Tim Hortons classified?

```
> db.yvr.aggregate(
   [ { $match : { name : /^Tim Hortons/ } },{ $group : { _id : "$amenity" , number : { $sum : 1 }}} ]
);
{ "_id" : "cafe", "number" : 14 }
{ "_id" : "fast_food", "number" : 9 }
```

Top 10 cafes in Vancouver, including Tim Hortons, based on number of stores
First attempt, revised below

```
>db.yvr.aggregate(
```

```
   [ { $match : {$or:[{amenity:"cafe"},{name:/^Tim Hortons/}]} },{ $group : { _id : "$name" ,
number : { $sum : 1 }}} ,{$sort:{number:-1}},{$limit:10}]
);

{ "_id" : "Starbucks", "number" : 61 }
{ "_id" : "Tim Hortons", "number" : 23 }
{ "_id" : "Starbucks Coffee", "number" : 19 }
{ "_id" : "Blenz Coffee", "number" : 14 }
{ "_id" : null, "number" : 12 }
{ "_id" : "JJ Bean", "number" : 8 }
{ "_id" : "Blenz", "number" : 6 }
{ "_id" : "Caffè Artigiano", "number" : 3 }
{ "_id" : "Cafe Artigiano", "number" : 3 }
{ "_id" : "Waves Coffee House", "number" : 3 }
>
```

Since there are non-standarized names and some null entries, I updated the database and modified the query:

```
db.yvr.updateMany(
  { name: "Starbucks"},
  {
    $set: { name: "Starbucks Coffee"},
  }
)


db.yvr.updateMany(
  { name: "Blenz"},
  {
    $set: { name: "Blenz Coffee"},
  }
)

db.yvr.updateMany(
  { name: "Cafe Artigiano"},
  {
    $set: { name: "Caffè Artigiano"},
  }
)

db.yvr.updateMany(
  {$and:[{name: /^Waves/},{amenity:"cafe"}]},
```

```
  {
    $set: { name: "Waves Coffee House"},
  }
)

db.yvr.updateMany(
  { name: "JJBean"},
  {
    $set: { name: "JJ Bean"},
  }
)

db.yvr.aggregate(
   [
   { $match : { $and: [{$or:[{amenity:"cafe"},{name:/^Tim Hortons/}]},{name: {$exists: true }}] }},
   { $group : { _id : "$name" , number : { $sum : 1 }}} ,{$sort:{number:-1}},{$limit:10}
   ]
);

{ "_id" : "Starbucks Coffee", "number" : 80 }
{ "_id" : "Tim Hortons", "number" : 23 }
{ "_id" : "Blenz Coffee", "number" : 20 }
{ "_id" : "JJ Bean", "number" : 10 }
{ "_id" : "Waves Coffee House", "number" : 9 }
{ "_id" : "Caffè Artigiano", "number" : 6 }
{ "_id" : "Bean Around The World", "number" : 3 }
{ "_id" : "Sciué", "number" : 2 }
{ "_id" : "Musette Caffè", "number" : 2 }
{ "_id" : "Beyond Coffee", "number" : 1 }


Top 10 amenities
> db.yvr.aggregate(
   [
   { $match : {amenity: {$exists: true }}},
   { $group : { _id : "$amenity" , number : { $sum : 1 }}} ,{$sort:{number:-1}},{$limit:10}
   ]
);

{ "_id" : "parking", "number" : 1020 }
{ "_id" : "bench", "number" : 659 }
{ "_id" : "restaurant", "number" : 634 }
{ "_id" : "cafe", "number" : 353 }
```

{ "_id" : "fast_food", "number" : 287 }
{ "_id" : "bicycle_parking", "number" : 227 }
{ "_id" : "post_box", "number" : 202 }
{ "_id" : "bank", "number" : 148 }
{ "_id" : "school", "number" : 138 }
{ "_id" : "toilets", "number" : 104 }

Other ideas about the dataset

The map data can provide additional insights about the geography and economics of the area. Continuing the comparison of Starbucks and Tim Hortons, it would be interesting to compare how the two chains locate themselves:

- Does Starbucks locate in areas with younger people?
- Are Tim Hortons more likely to be located near highways?
- How close are Starbucks located together?

Combined with user data, the map data can also provide insights into how errors are introduced in the map and how they might be prevented. Questions in this vein would include:

- Do most new users create errors when they are new and then improve, or do only some users create most of the errors, both while they are new and later? The answer to this question would inform how to train new users: broadly, or targeted at users who have started with errors. The answer may also suggest editing policy: limit the ability of all new users to make changes from the beginning, or limit new users' ability to make changes only after they've made a certain number of errors.
- What types of geographical entities are most associated with errors? Ways or nodes? Addresses or names? Amenity classification or location? Answers to these questions would help site administrators develop "guard rails" that limited errors while encouraging participation. Of course, what an administrator might consider an error may turn out to be the best way to present data, given the idiosyncrasies of the facts on the ground. After all, the map is not the territory. An alternative to a "guard rail" system, which restricts what users can input, would be to flag an error as idiosyncratic and prioritized for review.