P3: Open Street Map Data

Bryony Miles

Map area:	Cardiff newport bristol bath, England
Link:	https://mapzen.com/data/metro-extracts/#cardiff-newport-bristol-bath-england
File size:	535MB

I chose to do this project in MongoDB for two reasons:
- I know SQL and SQLLite and it is good to get some working practice with new technologies
- The flexibility Mongo DB offers to evolve the database as new needs or new data emerges

Step 1: Auditing the Data

a) Key Facts

First I checked unique tags to get a general idea of size using *iterative.py*. Here are the results:

- 1 bound
- 1 osm
- 52975 members
- 2489219 nodes
- 2255 relations
- 1012005 tags
- 317867 ways

I then used *getusers.py* to establish there are 1992 users. I then had a quick browse through the data. Things looked pretty similar in format to the Chicago example.

I ran *tagtypes.py* to check the 'k' value of the tags and check in case there were any weird tags which wouldn't work as keys in Mongo DB or whether I was unintentionally missing any key data.

This returned

- 819633 lower case tags
- 149344 lower case tags with colons
- 10 problem chars
- 43018 others

The problem chars were

- 7 x 'station_no.'
- 'surface.material'
- 'Hare Krishna FOLK Centre'
- 'aIM Boundaries'.

The majority of the others were
- AREA,CODE,NUMBER.POLYGON
- FIXME - better look at this.
- or started with a prefix naptan (National Public Transport Access Nodes):, source: (used to indicate the origin of data), name:, ref:, turn:, or cycleway:

b) Looking a little further

First I had a look at the wiki entries to re-familiarise myself with the data structure:

node - single point in space - lat,long, nodeid, tags

- point features i.e amenities , can be part of a relation
- nodes on ways - nodes may form the path of a way.
    - intersection - share nodes or if you cross at different heights (ie bridge) they don't share
- block of nodes - the location of the reference systems, tags of each node

way - ordered list of nodes

- normally at least one tag, often included within a relation, between 2 and 2000 nodes
- either open or closed (ie polyline or area)

relation - defining logical/geographical relationships

- one or more tags OR an ordered list of one/more nodes or ways
- many different types such as street, route, site, bridge, waterway

I then ran *audit.py* to look at the street names.   The first round allowed me to expand the expected list to include the following:

["South","West","East","North","Way","Walk","View","Terrace","Row","Rise","Parade","Park","Mead","Mews","Hill","Green" ,"Grove","Gate","Gardens","Estate","Cottages","Crescent","Close","Street", "Avenue", "Boulevard", "Drive", "Court", "Place", "Square", "Lane", "Road", "Trail","Buildings", "Parkway", "Broadway","Kingsway","Queensway","Commons"]

I then identified a few consistency issues - see **Data Cleaning plan**.

I then went through a similar process the first two letters of postcode - see **Data Cleaning plan**.

c) Data formatting

Time to get the data into Mongo DB using *mung1.py*.

I reformatted the data using the same data model as the project case study as we are looking at the same type of data. If this doesn't suit later down the line I can change it.

```
{
"id": "660774542",
"created": {
            "user": "DaveF",
            "timestamp": "2014-07-10T12:47:37Z",
            "changeset": "24063039",
            "version": "4",
            "uid": "115894"
            },
"pos": [51.510174, -2.1964616], "type": "node", "id": "104779"}
"address": {
            "country": "GB",
            "housenumber": "13",
            "city": "Cardiff",
            "street": "Pethybridge Road"
            },
"amenity": "restaurant",
"type": "node",
"name": "Golf Clubhouse"
"node_refs": ["2946322217", "2946322224", "2946322207", "2946322206"],
}
```

d) a bit more searching in MongoDB

For code look at the file *mongo.py*. First some totals and an example entry.

```
Number of Tags: 2807086
Number of Nodes: 2489194
Number of Ways: 317828
```

ADD USER INFO HERE

```
{'_id': ObjectId('572a3773c086a2c66c114fef'), 'created': {'user':
'corshamjim', 'version': '6', 'timestamp': '2009-02-14T14:36:29Z',
'changeset': '418988', 'uid': '73972'}, 'pos': [51.4245682, -2.1979582],
'type': 'node', 'id': '104793'}
```

I then had a look at the amenity type - see **Data Cleaning Plan**.

Next I had a look at address - see **Data Cleaning Plan**.

I noticed that some addresses had no postcode or city so I did some checking on this.

```
Addresses with a City 20472
Addresses with no City 24069
Addresses with no Postcode 27367
Addresses with no City and no Postcode 0
```

This is fine as they all have some positional data which makes them of value on the map.

Step 2: Creating a Data Cleaning Plan

The fields has been audited for:

- accuracy
- completeness
- consistency
- uniformity

The next step is to put the Data Cleaning Plan into action. See mongoclean.py

Data Cleaning Plan

Data Field	Issues	Fixes
Address	As you'd expect the address data is pretty clean. I set up the following maps which recommended only 23 changes: 1. `Rear of the Llanover Arms, Bridge St => Rear of the Llanover Arms, Bridge Street` 2. `Rear of Llanover Arms, Bridge St => Rear of Llanover Arms, Bridge Street` 3. `High St => High Street` 4. `Victoria St => Victoria Street` 5. `Naas lane => Naas Lane` 6. `Greenhill => GreenHill` 7. `Doyle avenue => Doyle Avenue` 8. `Newbridge st => Newbridge Street` 9. `kingsway => Kingsway` 10. `Aberfawr Rd => Aberfawr Road` 11. `Old Gloucester Rd => Old Gloucester Road` 12. `Church Rd => Church Road` 13. `Coed Cae Rd => Coed Cae Road` 14. `Clyde Rd => Clyde Road` 15. `Lynx Crescent, => Lynx Crescent` 16. `Hope Chapel hill => Hope Chapel Hill` 17. `Guilford Cresent => Guilford Crescent` 18. `Hutton HIll => Hutton Hill` 19. `caerphilly road => caerphilly Road` 20. `pickwick road => pickwick Road`	mapping = { "Crescent,": "Crescent", "Cresent": "Crescent", "HIll": "Hill", "Rd": "Road", "St": "Street", "avenue": "Avenue", "hill":"Hill", "kingsway":"Kingsway", "lane":"Lane", "road":"Road", "st":"Street" }
Postcode (first 2 letters)	Data pretty clean again. Some issues with capitalisation. One issue with a web address rather than a postcode.	Pmapping = { "Bs": "BS", "Cf": "CF", "bs": "BS", "cf": "CF", "ta": "TA" } Manual change to osm file: - found postcode for St Mary's Priory Church, Monmouth on Google.
Amenity	Two unrecognisable amenity names: yes, pau `{'created': {'timestamp': '2014-09-11T19:03:45Z', 'version': '1', 'user': '-Anarchy-', 'changeset': '25375046', 'uid': '2304265'}, 'pos': [51.4600608, -3.1611394], '_id': ObjectId('572a37a2c086a2c66c2e327e'), 'name': "'Dr Who' Water Bus Stop", 'id': '3072293702', 'type': 'node', 'amenity': 'yes'}` `{'created': {'timestamp': '2015-11-14T19:11:04Z', 'version': '1', 'user': 'manor09', 'changeset': '35314282', 'uid': '182738'}, 'pos': [51.7247727, -3.0641434], '_id': ObjectId('572a37aec086a2c66c34f705'), 'name': 'Hair on Harpers', 'id': '3836219527', 'type': 'node', 'amenity': 'yes'}` `{'created': {'timestamp': '2009-12-04T11:30:23Z', 'version': '1', 'user': 'JonBritton', 'changeset': '3287552', 'uid': '48155'}, 'pos': [51.4842006, -3.2396283], '_id': ObjectId('572a377cc086a2c66c1715ce'), 'id': '580823638', 'type': 'node', 'amenity': 'lau'}` `Convalescent Centre, Dental Hospital, Health Centre, Manège, Pool, Pool Hall, Showroom, Triangle Walk, Truck Service Iveco, University (exists lower case)`	Unrecognisable Entries 1. change to "water_bus_stop" - this is not spam, they do exist. 2. Hair on Harpers does exist, I'll change it to "hairdresser" 3. Nothing under name so delete entry as not a meaningful entry. Change all of these to to lower case with an underscore instead of a space.

Address	**Capitals:** BRIDGEND PONTYCLUN PONTYPRIDD **With and without , Bristol** Almondsbury, Bristol Bradley Stoke, Bristol Congresbury, Bristol Thornbury, Bristol Yate, Bristol **Just , Bristol** Clifton Village, Bristol Clifton, Bristol Downend, Bristol Harbourside, Bristol Horfield, Bristol **Just , Cardiff** Cardiff Bay, Cardiff Roath, Cardiff **With and without , Cardiff** Pontprennau, Cardiff **With , Weston** Worle, Weston Super Mare Worle, Weston-super-Mare **Odd ones:** Cardiff, Wales Brisol Trefforest Treforest **Hyphenation:** Burnham on Sea Burnham-on-Sea Weston Super Mare Weston-Super-Mare Weston-super-Mare	Change to Sentence case There are some suburbs of Bristol and Cardiff that are standalone and others with ,Bristol, Weston or , Cardiff after. Decision: remove all ,Bristol all ,Cardiff for consistency Change to "Cardiff" Change to "Bristol" Change to "Treforest" Remove hyphens Change Super to Sentence case
FIXME	I printed these off and there are notes questioning their validity. There are 965.	Decided to delete these until confirmed as could cause confusion.

Step 7: Additional ideas

Now the data is cleaned, what could it be used for? All sorts, here are a few suggestions.

- Moving house - searching for amenities within a specified area?
- Coming on holiday - amenities near the hotel?
- Worried about speeding fines? Where are the cameras?
- Thinking of setting up a sweet shop? Where's the competition.

More than anything I'd say that the OpenStreetMap data really comes into play when it is linked with another dataset ie. improve services, solve problems (ie traffic jams, oversubscription to schools), looking at crime data, finding gaps in the market. The choice of MongoDB means that it should be easier to combine old/new datasets without going back to the drawing board.

Step 8: Other Thoughts

If I did this again I would try to perform the MongoDB import earlier as querying is much easier. Ideally, after having checked out the structure of the data and used that as a basis for a new **Data Format**.