

OpenStreetMap Data Wrangling: Case Study - Auditing & Cleaning

Map Area

 [Boston, MA, United States](#)

- [Boston on OpenStreetMap](#)
- [Boston Metro extract](#)

This is the city that welcomed me to the States, where I studied and graduated. I have a particular fondness for Boston, and I'm interested in seeing how good the data is and if it can be improved.

Problems encountered with the data

- A lot of street names were abbreviated, using different abbreviations (`Street`, `street`, `St.`, `St ...`)
- Some street names only consisted of names of numbers, so I had to identify what they really referred to (`#1302`, `3`, `Elm`, `Albany ...`)
- Some street names contained unexpected characters or contained more than one street name (`Church Street, Harvard Square`, `Massachusetts Ave; Mass Ave`)

Analysis process

We needed to extract the data so we can start working with it.

The first thing to do was to look at the `boston_massachusetts.osm` file, and get a sense of its structure.

It follows an XML structure, humanly readable. The way it works is simple. There are several tags, and we were interested in three in particular:

- **nodes**, which consist of "a **single point in space** defined by its *latitude*, *longitude* and *node id*. Nodes can be used to define standalone point features. In this case, a node will normally have at least one tag to define its purpose. Nodes are often used to define the shape or "path" of a way."
- **ways**, which are an "**ordered list of nodes** and normally also has at least one tag". To make it simple, these are streets, avenues...
- **relations**, which are "used to model logical (and usually local) or **geographic relationships between objects**". To make it simple, these are areas made of several ways.

Nodes are represented with the tag `<node></node>`. They can have the following attributes:

- `id` - integer, the unique ID of the node
- `lat` - integer, the latitude
- `lon` - integer, the longitude
- `version` - integer, the number of edits
- `timestamp` - W3C datetime format, time of the last modification
- `changeset` - integer, the changeset number in which the object was created or updated. A changeset consists of a edits made by a single user over a short period of time
- `uid` - integer, the unique ID of the user who last edited
- `user` - string, the pseudonym of the user who last edited

Ways are trepresented with the tag `<way></way>`. They can have the following attributes:

- `id` - integer, the unique ID of the node
- `version` - integer, the number of edits
- `timestamp` - W3C datetime format, time of the last modification
- `changeset` - integer, the changeset number in which the object was created or updated. A changeset consists of a edits made by a single user over a short period of time
- `uid` - integer, the unique ID of the user who last edited
- `user` - string, the pseudonym of the user who last edited

Relations are represented with the tag `<relation></relation>`. They can have the same attributes as ways.

Nodes that make up the **ways** are represented with the tag `<nd />`. They can only have a `ref` attribute, which basically references the `id` of the node as an integer.

Ways and **nodes** in **relations** are represented with the tag `<member />`. They can have three attributes:

- `type` - string, the type of the object (node, way)
- `ref` - integer, the references to the unique ID of the object
- `role` - string, the role it plays in the relation (`via`, `from`, `to`)

Tags are represented with the tag `<tag />`. They are used as child elements of nodes, ways and relations to bring additional information. They can have two attributes:

- `k` - string, the key of the tag
- `v` - string, the value of the tag

First, we had to know our dataset, its 436.2 MB of data and its 5,920,723 lines of code. Here are the different tags it contains:

- 1,939,872 nodes

- 310,285 ways
- 1,263 relations
- 10,894 members
- 2,335,749 nds

We can also thank 1,402 users for making this dataset possible.

Auditing the data

After auditing the data, we ended up with the following dictionary:





```
{'Avenue': 851, 'Street': 4137, 'Drive': 77, 'Parkway': 27, 'street': 2, 'Road': 248, 'Ave': 163, 'Ave.': 21, 'St': 252, 'St.': 1, 'Broadway': 98, 'Garage': 1, 'Place': 83, 'Lane': 27, 'Pasteur': 3, 'Square': 68, 'St.': 42, 'Boulevard': 12, 'Rd': 29, 'Way': 25, 'Pkwy': 10, 'Ct': 8, 'Fellsway': 15, 'Wharf': 5, 'Elm': 1, 'Highway': 13, 'Hwy': 1, 'Artery': 1, 'Center': 15, 'Newbury': 2, 'Holland': 1, 'Lafayette': 1, 'Street.': 1, 'Floor': 1, '1702': 1, '6': 1, 'South': 2, 'LEVEL': 1, 'rd.': 1, 'Greenway': 2, 'Corner': 1, 'Row': 6, 'Sq.': 1, '303': 1, 'floor': 2, '1100': 1, '846028': 1, 'Hall': 1, 'Turnpike': 2, 'st': 1, 'Park': 52, 'Terrace': 17, 'Jamaicaway': 2, 'Plaza': 2, 'Court': 9, 'Building': 1, 'Market': 1, '#1302': 1, '#12': 1, '#501': 1, 'Dartmouth': 1, '104': 1, 'Yard': 1, 'Mall': 3, 'Boylston': 2, 'Driveway': 1, 'Winsor': 1, 'ST': 1, 'Cambrdige': 2, 'Albany': 3, 'Fenway': 5, 'Hlghway': 1, 'Windsor': 2, 'Ext': 1, 'Circle': 7, 'place': 1, 'Pl': 1, 'Brook': 1, 'Hampshire': 1, 'Longwood': 1, 'Dr': 1, '3': 1, 'H': 1}}
```

We could see a number of issues here:

- we had a lot of numbers that should probably correspond to `addr:housenumber`.
- we had a lot of different abbreviations for different street types
- we had names of streets and parks instead of the type: Boylston is a Street, Broadway is a highway, Fenway is a park, and so on
- we had some typos: 'Hlghway', 'Cambrdige'...


We defined a list containing the correct, expected values, and a dictionary mapping incorrect abbreviations to their correction.

We audited the data a second time to map incorrect values to the string they were contained in (`'#12': ['Harvard St #12']`), in order to get a better understanding of the bad data. We were able to identify these specific issues:

1. The numbers we saw were either:
 1. house numbers, that belong to the tag `addr:housenumber`
 2. suite numbers, that belong to the tag `addr:suitenumber`
 3. train track numbers, that we don't necessarily need (knowing where to find South Station is sufficient)
 4. a PO Box that happens to be on Albany Street. The `PO Box 846028` information appears in the `addr:housenumber` tag already
2. `Albany`, `Boylston`, `Cambridge` (written `Cambrdige` here with a typo), `Dartmouth`, `Elm`, `Hampshire`, `Holland`, `Newbury` are Boston street names
3. `Lafayette`, `Longwood` and `Winsor` are Boston avenue names
4. `Pasteur` corresponds to Avenue Louis Pasteur
5. The `South Market Building` is located on  `4 South Market Street`
6. All `Center` s are correct, we added `Center` to our expected list
7. All `Circle` s are correct, we added `Circle` to our expected list
8. `Coolidge Corner` should not have appeared here
9. `Museum of Science Driveway` is correct, we added `Driveway` to our expected list
10. `B Street Ext` corresponds to B Street
11. `Fellsway` is a parkway
12. `Fenway` is of course Fenway Park
13. `Boylston Street, 5th Floor` should only be Boylston Street, the 5th Floor information should appear in the tag `addr:floornumber`
14. `Stillings Street Garage` is a garage located on  `11 Stillings Street`
15. `East Boston Greenway` is correct
16. `H` is actually  `605 Hancock Street`
17. `Faneuil Hall` is located  `4 South Market Street`
18. `Jamaicaway` is correct
19. `LOMASNEY WAY, ROOF LEVEL` should only be Lomasney Way
20. `Cumington Mall` is correct, we added Mall to our expected list
21. `Faneuil Hall Market` is 1 Faneuil Hall Square
22. `Two Center Plaza` should have been 2 Center Plaza
23. `Park Plaza` is correct
24. `Charles Street South` should have simply been Charles Street
25. `Sidney Street, 2nd floor` should have only been Sidney Street, the 2nd Floor information should have appeared in the tag `addr:floornumber`
26. `First Street, 18 floor` should have only been First Street, the 18th Floor information should have appeared in the tag `addr:floornumber`
27. `Windsor` corresponds to Windsor Place in Somerville

We then ran a third audit in order to identify any street name containing incorrect characters.

Almost all the entries returned we considered correct, except for two:

- Church Street, Harvard Square corresponds to a parish located on  [1446 Massachusetts Avenue](#)
- Massachusetts Ave; Mass Ave should have only been Massachusetts Avenue

Cleaning the data

Once we had identified all the incorrect names, we created dictionaries to update incorrect values. We could now clean the data and save it all in a final dictionary.

This final dictionary contained all the incorrect values and their correction, and we used it to prepare the insertion of the data into a SQL database. We exported the corrected the data into 5 csv files:

- nodes.csv
- nodes_tags.csv
- ways.csv
- ways_nodes.csv
- ways_tags.csv

We used these files to create a boston.db database containing five tables corresponding to the five csv s. Then we were ready to start querying.