**Data Audit**

**all\_tags.py**

To get a dictionary with the tags as key and number of times the tags can be encountered in the map as value I have used iterative parsing to process the map file. It is important to get the feeling of how much data can be expected to have in the map. The following is the output :

*Defaultdict(<class ‘int’>,{ ‘member’ : 42581, ‘meta’ : 1 , ‘nd’ : 2479830 , ‘node’ : 2172781 , ‘note’ : 1, ‘osm’ : 1 , ‘relation’ : 947 ,’remark’ : 1, ‘tag’ : 374482 , ‘way’ : 168695 })*

**tag\_types.py**

I have used regular expression to classify the ‘k’ attributes of each tag and see if there are any potential problems. Here I have used 3 regular expressions and by using them tags are classified in 4 categories (lowercase, lowercase with colon, problematic charaters and others) and the output is the count of tags in each group:

*{ ‘lower’ : 364509 , ‘lower\_colon’ : 9525 , ‘other’ : 448 , ‘problemchar’ : 0}*

**sampler.py**

I have used this code to take a systematic sample of elements from the original osm. It extracts the kth top level element . The size of the sample file can be adjusted by changing value of K.

**tag\_tags.py**

I have used a sample.osm file, which was created with the help of sampler.py , for this code. It creates a dictionary with the set of ‘v’ attributes and ‘k’attribute as key. With the help of the dictionary we could see the data more clearly and find problem with data. For example I found that there are two different ‘k’ attributes for postal code (‘addr:postcode’ and ‘postal\_code’).

**uniqueUsers.py**

It helps to find out how many users have contributed to the dataset. It returns a set uid’s of users who have contributed to dataset.

**name\_audit.py**

It is used to audit names in ‘k’ attribute. It provides the code to find all the abbreviations that are need to be included in mapping dictionary. Mapping dictionary is used to create a better name for better comprehension.

**create\_csv.py**

Used to parse, clean and shape osm data. It is used to create csv and its validation using **schema.py** .

**schema.py**

Defines schema used for validation.

**CaseStudy.pdf**

The Report of the project which includes sql queries to provide statistics about the dataset.

**sample100.osm**

Sample osm file created using sampler.py

**AREA DESCRIPTION.docx**

Provides description of the map data used for case study.

**References.txt**

Provides links to websites, github repositories etc used as reference for the case study.