

OpenStreetMap Data Project - Wrangling Map Data for Sydney, Australia

February 6, 2018

1 OpenStreetMap Data Project: Sydney, Australia

1.0.1 Geographic Area

Sydney, New South Wales, Australia

https://mapzen.com/data/metro-extracts/metro/sydney_australia/

<https://www.openstreetmap.org/relation/5750005>

The map dataset chosen for this project is the Sydney, Australia metro area dataset from MapZen. I decided on this location because of a study abroad program in Australia I participated in during my undergrad. I lived in and traveled the country for six months, but was unable to trek the Harbour City. I have friends and family currently living in Sydney and would like to visit the area soon. In this project I'm looking to explore the dataset in an effort to contribute to the improvement of map data on the OpenStreetMap database website, gain some additional geographic familiarity and uncover some interesting facts about the city.

1.0.2 Creating a Sample Dataset

Before auditing and cleaning the original dataset, I found it useful to create a smaller sample file in order to validate and work through any issues discovered. This allowed for a more streamlined process when writing, running, and correcting a number of python scripts and SQL queries. The sample file created was 6.9 MB in size and this process proved to be a time saver, especially in the case of the CSV conversion script. Once the code was suitable and successfully run on the sample file, it was applied to the original uncompressed OSM XML file and resulting tables created from the CSV conversion files.

The method was utilized in assessing the quality of the data for validity, accuracy, completeness, consistency and uniformity and ultimately the effort to clean and standardize the dataset. I identified several language errors in the location data including spelling, abbreviations, and typos.

In []: *#See sampleconversion.py for code details.*

#Taking a systematic sample of elements from the original DSM region (Sydney, Australia)

1.1 Data Audits

Street Name Errors Python is used to audit and print out street names in the sample dataset, containing some unusual values. The function below is used to identify and correct unexpected spelling errors, street name abbreviations and typo inconsistencies. The function takes a string with `street_name` as an argument and returns the corrected name, iterating over each word in an address.

In some cases mapping out more specific changes was required, especially considering the inimitable street names found nowhere else. A bit more verification was required in order to limit hypercorrections, using a popular map application. From the provided list the script audits and make changes to the 'mapping' variable in the OSM file. Therefore "ApplegumCrescent" is corrected to "Applegum Crescent".

In []: #See sydneyProcessing.py for code details.

State Code Audit Additional auditing was performed to verify correct use and consistency of the state code. The state code for New South Wales is NSW. The results of this audit show there were no inconsistencies.

In []: #See scode.py for code details.

A SQL query performed once the CSVs have been loaded into SQL tables confirms all 15 instances including state code are correct.

```
SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='state_code'
GROUP BY key;
```

```
NSW 15
```

Postal Code Audit Auditing process involving numeric field performed to verify consistency among postal code, which in Sydney are 4 digits long.

In []: #See postalcode_audit.py for code details.

XML Data Primitives The code below explores the data further by giving a count of the data primitive included in the `sydney_australia.osm` file. The code checks the "k" value for each "" to identify potential problems. Four regular expressions check for certain patterns in the tags, and whether we have any tags with problem characters.

The function 'key_type', provides a count of each of four tag categories in a dictionary. The four categories are "lower" for tags that contain only lowercase letters and are valid, "lower_colon" for otherwise valid tags with a colon in their names, "problemchars" for tags with problematic characters, and "other" for other tags that do not fall into the other three categories.

In []: #See sydneyProcessing.py for code details. Code output below.

```
{'problemchars': 8, 'lower': 745578, 'other': 8614, 'lower_colon': 104685}
```

CSV Conversion After the auditing and cleaning steps were performed, the OSM file was converted to CSV files for inclusion in a SQL database as tables. The script below details the conversion to CSV format, which can then be converted in SQL tables.

```
In [ ]: %load 'schema.py'
```

```
#See saConversion.py for details
```

1.2 Dataset Overview Statistics

Size of Files

```
sydney_australia.osm .... 344.2 MB
sydneyAu.db ..... 188.8 MB
nodes.csv ..... 127.3 MB
nodes_tags.csv ..... 6.2 MB
ways.csv ..... 12.4 MB
ways_tags.csv ..... 23.4 MB
ways_nodes.cv ..... 44 MB
```

Number of Unique Users

```
sqlite> SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;

count = 2382
```

Number of Unique Nodes

```
sqlite> SELECT COUNT(*) FROM nodes;

count = 1529440
```

Number of Unique Ways

```
sqlite> SELECT COUNT(*) FROM ways;

count = 208933
```

1.2.1 Exploration of Additional Nodes

I chose to explore a number of additional node types to further the geographical exploration of the map data for consideration in the planning of a potential holiday trip. I started with a query into the top ten amenities Sydney has to offer. It is interesting to note the top amenity in Sydney is the communal bench.

Amenities:

```
sqlite> SELECT value, COUNT(*) as num
...> FROM nodes_tags
...> WHERE key='amenity'
...> GROUP BY value
...> ORDER BY num DESC
...> LIMIT 10;
```

bench	1465
restaurant	1015
cafe	884
drinking_water	836
parking	764
toilets	660
fast_food	604
bicycle_parking	588
post_box	423
place_of_worship	397

Everybody gets hungry. The next query provides a breakdown of the types of restaurants that call Sydney home.

Cuisine:

```
sqlite> SELECT nodes_tags.value, COUNT(*) as num
...> FROM nodes_tags
...> JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='restaurant') i
...> ON nodes_tags.id=i.id
...> WHERE nodes_tags.key='cuisine'
...> GROUP BY nodes_tags.value
...> ORDER BY num DESC
...> LIMIT 20;
```

thai	74
chinese	64
italian	55
pizza	45
japanese	41
indian	34
vietnamese	16
korean	15
greek	12
sushi	10
regional	9
asian	8
international	8
malaysian	8
burger	7

asian;sushi;japanese	6
lebanese	6
seafood	6
fish_and_chips	5
french	5

This query explores data identifying religious structures, which could be used to identify potential architectural wonders worth visiting.

Liturgy, Top 3:

```
sqlite> SELECT nodes_tags.value, COUNT(*) as num
...> FROM nodes_tags
...> JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='place_of_worship') i
...> ON nodes_tags.id=i.id
...> WHERE nodes_tags.key='religion'
...> GROUP BY nodes_tags.value
...> ORDER BY num DESC
...> LIMIT 3;
```

christian	338
buddhist	9
muslim	9

The following query shows many different tourist attractions worth visiting. I'm impressed by the 327 viewpoints located throughout the city. Many of which provide glorious beach vistas.

Tourism, Top 10:

```
sqlite> SELECT nodes_tags.value, COUNT(*) as num
...> FROM nodes_tags
...> JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE key='tourism') i
...> ON nodes_tags.id=i.id
...> WHERE nodes_tags.key='tourism'
...> GROUP BY nodes_tags.value
...> ORDER BY num DESC
...> LIMIT 10;
```

information	341
viewpoint	327
picnic_site	232
hotel	130
attraction	123
artwork	121
museum	51
hostel	43
camp_site	21
motel	17

City art is explored in this query to provide some potential mid-point attractions while walking the city streets from one attraction to the next. City art also draws a nice contrast with artwork contained in museums.

City Art

```
sqlite> SELECT nodes_tags.value, COUNT(*) as num
...> FROM nodes_tags
...> JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE key='artwork_type') i
...> ON nodes_tags.id=i.id
...> WHERE nodes_tags.key='artwork_type'
...> GROUP BY nodes_tags.value
...> ORDER BY num DESC;
```

sculpture	39
statue	20
mural	6
interactive	3
gates	1
graffiti	1
installation	1
sculptor	1
streetart	1

In the interest of exploring how the data is added to the map dataset I thought we could investigate the top contributing users. From the results below, we can see the top ten contributing users out of the 2382 total unique users. Based on these findings, user “balcoath” made 18.7% of the total contributions for the top ten users.

1.2.2 Top Contributing Users

```
sqlite> SELECT e.user, COUNT(*) as num
...> FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
...> GROUP BY e.user
...> ORDER BY num DESC
...> LIMIT 10;
```

balcoath	117344
inas	89241
TheSwavu	74266
aharvey	66423
ChopStiR	60487
ozhiker2	51861
"Leon K"	46508
cleary	41601
Rhubarb	40735
AntBurnett	37524

1.2.3 Potential Areas for Improvement

Data integrity and the standardization of data is paramount in being able to perform reliable data analysis, and some of the more common errors in this dataset appear to be the result of human error. As is likely the case with the street name errors identified. Perhaps creating a data-entry template, Readme file, or a data type verification process via a java application could be utilize when users are manually entering data into the OpenStreetMap databases.

This would likely prevent any mistakes before they are entered into the database, making the need to routinely audit and clean the data less cumbersome. However, some anticipated problems with this suggestion include the likelihood that some users wouldn't read the supporting materials or even ignore the instructions altogether. This might be avoided with solid and complete input from a GPS device.

1.2.4 Conclusions

In the effort to reveal insights into the diversity of religious structures in the city, the "places of worship" value tag in the nodes_tag table was explored. For the purposes here, we could use these results to gain additional insight in the makeup of religious structures, and potentially identify some of the more popular religious structures worth visiting.

While reviewing the results I couldn't help consider the lack of representation of those who don't affiliate with any religion. This is likely the result of this population not having a central place of worship. Gathering additional data in the proportional religious affiliation of the citizens of Sydney for comparison would provide a better glimpse into the religious makeup of the city.

All the data explored in this project has provided some interesting results, which are sure to prove helpful in the planning of a future holiday to Sydney, Australia.