Problems encountered

Python file used:
clean_and_ingest.py: parse/clean xml, and, load into MongoDB collection

Inconsistent use of abbreviations for street types

Similar to the dataset in the course materials this dataset used inconsistent abbreviations for street types. I did some initial checks of the data before it was loaded into MongoDB for further analysis.

Examples:

```
'Cirtcuit': set(['Champion Cirtcuit'])
'Circuit': set(['Waterbrooke Circuit', 'Joyner Circuit', 'Wimbledon Circuit', 'Currell Circuit', 'Jagera Circuit'])

'Tce': set(['Landsborough Tce'])
'Terrace': set(['Lower River Terrace', 'River Terrace', 'William Terrace', 'Prospect Terrace', 'Beaconsfield Terrace', 'Astor Terrace', 'Allan Terrace', 'Mowbray Terrace', 'Bowen Terrace', 'Bay Terrace', 'Lorimer Terrace', 'Ascog Terrace', 'Junction Terrace', 'Brighton Terrace', 'Lamington Terrace', 'Eversley Terrace', 'Normanby Terrace', 'Wickham Terrace', 'Wilton Terrace', 'Skyring Terrace', 'Waterview Terrace', 'Victoria Terrace', 'Vernon Terrace', 'Petrie Terrace', 'Eagle Terrace', 'Heaslop Terrace', 'Dornoch Terrace', 'Stanley Terrace', 'Raymond Terrace', 'Railway Terrace', 'Gregory Terrace', 'Denham Terrace']),
```

From the output of this I was able to create a mapping from the all abbreviations in the data set to the full name.

Invalid postcodes

Postcodes in Australia is always a numeric value that is 4 digits long. Some of the postcodes in the data contained non-numeric characters the more common ones included trailing/leading spaces, commas and the string “QLD” (state that Brisbane is in). To clean the postcodes all non-numerical characters were removed.

Examples:

```
QLD, 4101
QLD 4119
```

Invalid city fields

There were city values that also contained the state state name. These were cleaned out where the string “QLD” was removed from all city values.
Example:

```
Underwood, QLD
```

The data in the city field are actually all suburbs. The city field needs to be set to “Brisbane” and the data merged into the suburb field where city is used if the suburb does not exist (not implemented).

Invalid and inconsistent formatting of Phone numbers

There were phone numbers that are not valid in the file. Phone numbers were also in a very wide number of formats, country/area codes were inconsistently included, different separators were used and some did not use separators at all. All phone numbers were parsed and reformatted in international format: +61 7 12345678. Phone numbers that were not valid were removed from the data.

It also became apparent that there can be more than one phone number per node, the phone field in the document was changed to an array to accommodate this.

Examples:

```
07 3193 1072
+61 7 3425 3314
33501333
+61 (0)7 3263 1607
+61 (07) 3341 3344
1300groove
+61 7 3206 8143;+61 7 3206 7011
```

Incomplete address Data

The queries in the next section show that the address data is incomplete where:

- 21% of addresses do not have a street address
- 66% of addresses do not have a suburb or city
- 70% of addresses do not have a postcode

To fill in the missing data, the data will need to be enriched with another source such as Google Maps where the latitude and longitude values can be used for reverse geocoding to get the address information. This has not been implemented.

Overview of the Data

Size of file: brisbane_australia.osm 203mb
The queries which builds the pipeline separately were run via pymongo, refer to mongo_queries.py.

Number of unique users

```
db.map.distinct("created.user").length
```

```
967
```

Number of nodes

```
db.map.find({"type":"node"}).count()
```

```
944836
```

Number of nodes ways

```
db.map.find({"type":"way"}).count()
```

```
135635
```

Number of apartment buildings

```
db.map.find({"building":"apartments"}).count()
```

```
216
```

Number of addresses

```
db.map.count( {"address" : {"$exists" : 1}}  )
```

```
16691
```

Number of addresses by city/suburb:

```
pipeline = [
{"$match": {"address" : {"$exists" : 1}}},
{"$group": {"_id": {"city": "$address.city", "suburb" : "$address.suburb" }, "count"
: {"$sum" : 1}}},
{"$sort": {"count" : -1}},
{"$limit" : 100}
]
result = db.map.aggregate(pipeline)
```

Results (truncated):

```
[{u'_id': {}, u'count': 11072},
 {u'_id': {u'suburb': u'Carindale'}, u'count': 1006},
 {u'_id': {u'city': u'Loganholme'}, u'count': 689},
 {u'_id': {u'city': u'Tanah Merah'}, u'count': 422},
 {u'_id': {u'city': u'Thornlands'}, u'count': 385},
 {u'_id': {u'city': u'Banyo'}, u'count': 355},
 {u'_id': {u'city': u'Wellington Point'}, u'count': 315},
 {u'_id': {u'city': u'Chapel Hill'}, u'count': 231},
 {u'_id': {u'city': u'Brisbane'}, u'count': 180},
 {u'_id': {u'city': u'Rochedale South'}, u'count': 165},
 {u'_id': {u'city': u'Nudgee'}, u'count': 149},
 {u'_id': {u'city': u'Kelvin Grove'}, u'count': 147},
 {u'_id': {u'city': u'Beenleigh'}, u'count': 121},
 {u'_id': {u'city': u'Newmarket'}, u'count': 107},
 {u'_id': {u'city': u'Brisbane', u'suburb': u'Carindale'}, u'count': 86},
 {u'_id': {u'city': u'Ormiston'}, u'count': 84},
 {u'_id': {u'city': u'Capalaba'}, u'count': 72},
 {u'_id': {u'city': u'Alberton'}, u'count': 66},
```

Due to the large number of addresses without a suburb or a city, the number of addresses per suburb or city are very low.

Number of addresses without streets

```
pipeline = [
{"$match": {"address" : {"$exists" : 1}, "address.street”: {"$exists" : 0}}},
{"$group": {"_id": "Addresses", "count" : {"$sum" : 1}}},
]
result = db.map.aggregate(pipeline)
```

```
3519
```

Number of addresses without postcodes

```
pipeline = [
{"$match": {"address" : {"$exists" : 1}, "address.postcode”: {"$exists" : 0}}},
{"$group": {"_id": "Addresses", "count" : {"$sum" : 1}}},
]
result = db.map.aggregate(pipeline)
```

```
11809
```

Conclusion and Additional Ideas

Based on the number of addresses and the missing address fields in this dataset it is obvious that the data is incomplete. It doesn't have enough data to be useful when there are superior map applications such as Google Maps.

The Brisbane City Council have a public data store (http://data.brisbane.qld.gov.au/). The data available is very diverse, examples include: bikeways (dedicated roads for cyclists), bus stops, parking zones, garbage collection days for different streets and public toilet locations. If all of this data was cleaned and contributed openstreetmap.org then it would have a very rich set of data for Brisbane and perhaps a developer can build a suite of apps based on this data for people who live or travel to Brisbane to help them discover new things in the city.

Some examples of apps that could use this data:

- After adding the waste collection days dataset to the map data. Remind you to take out the trash on days of collection based on address inputted into the app. In Brisbane, every second week the recycling rubbish is collected and twice a year there is a special collection for bulky items. The app would also track these and remind the user.
- After adding the wifi hotspots dataset to the map data. Based on the GPS location of the user's smartphone, it will be able to tell the user where the nearest public wifi hotspot is.