OpenStreetMap Project – Peter Heimann

Map Area:

The area around my home town of New Brunswick, NJ, USA:
http://overpass-api.de/query_form.html
(node(40.4,-74.6,40.6,-74.2);<;);out;
File size: 56.8MB

I also downloaded a subset (40.49,-74.41,40.51,-74.39), for initial examination. All of the problems described below were found and fixed using this subset, unless otherwise stated.

Problems Encountered:

Most of the nodes did not have any information on "created," so the original code from Lesson 6 yielded an empty dictionary. For these nodes, I decided to explicitly show the "created" as "None" rather than leaving it blank.

Under the child element "tag", I noticed keys that begin with "gnis:", which is the USGS Geographic Names Information System, a database that contains millions of names for geographic features in the United States and Antarctica. I decided to map the gnis keys into the "addr:" keys from the Lesson 6 scripts.

I also noticed keys that begin with "tiger:". According to the OSM Wiki, "The Topologically Integrated Geographic Encoding and Referencing system (TIGER) data, produced by the US Census Bureau, is a public domain data source which has many geographic features. The TIGER/Line files are extracts of selected geographic information, including roads, boundaries, and hydrography features. All of the roads were imported into OSM in 2007 and 2008, populating the nearly empty map of the United States." Again, I mapped the tiger keys into the existing "addr:" keys.

Finally, I noticed that there are many different keys for the places (nodes), such as "amenity - post_office," "amenity - place_of_worship," "shop – supermarket," "gnis:Class – Populated Area," etc. I mapped all of these keys to a single key, "place_type."

In summary, here is the mapping, which in some cases had to be done separately for nodes and ways:

Node/Way	OSM Key	Output Key
Node	addr	sub-keys: city, state, street, housenumber, zipcode

Either	created_by	created:user
Either	Source	created:user
Node	amenity	place_type
Node	gnis:Class	place_type
Node	gnis:county_name	addr:county
Node	gnis:County	addr:county
Node	gnis:ST_alpha	addr:state
Node	name	name
Way	name	street
Node	addr:postcode	addr:zipcode
Node	shop	place_type
Way	tiger:county	county
Way	tiger:zip_left	zipcode

I added all of these mappings to the Lesson 6 script. Although it would have been more elegant to create a separate dictionary with these mappings, I decided to write explicit code for this small number of mappings, as shown in the cleanup.py file used for this project.

I ran the python code in the cleanup.py file, and then loaded the resulting output, new_brunswick.osm.json, into MongoDB, with an explicit collection name "nb":

```
Owners-MacBook-Air:Project Owner$ ./bin/mongoimport --file
new_brunswick.osm.json --collection nb --db osm
```

Overview of the Data:

Number of documents in the database:

```
from pymongo import MongoClient
import pprint
client = MongoClient('localhost:27017')
db = client.osm
result = db.nb.find().count()
pprint.pprint(result)

447637
```

I ran various queries on this database, not only to get statistics, but also to find data issues that I did not notice when perusing the json files.

Count of first-level types:

```
from pymongo import MongoClient
import pprint
client = MongoClient('localhost:27017')
db = client.osm
pipeline = [
    { "$group" : {"_id" : "$type", "count" : {"$sum":1}}},
    { "$sort" : {"count":-1}}
    ]
result = db.nb.aggregate(pipeline)
```

```
pprint.pprint(result)

{u'ok': 1.0,
 u'result': [{u'_id': u'node', u'count': 397934},
             {u'_id': u'way', u'count': 49613},
             {u'_id': u'multipolygon', u'count': 86},
             {u'_id': u'gas', u'count': 3},
             {u'_id': u'Public', u'count': 1}]}
```

I see that 89% of the documents are nodes, and nearly all of the remaining 11% are way, with no relations. But I see that there is still more cleanup to be done. I looked at the contents of a few of these non-node, non-way documents, and I saw things like:

```
u'type': u'multipolygon', u'natural': u'wood'
u'type': u'multipolygon', u'landuse': u'recreation_ground'
u'type': u'gas', u'location': u'underground', u'man_made': u'pipeline'
u'type': u'Public', u'aeroway': u'aerodrome'
```

I suggest that the "type:multipolygon" should be changed to "type:relation," while the "type:gas" and "type:Public" should be changed to "type:way." I did not do this as part of this project, but a real-world data-wrangling project would most likely have several iterations of cleaning, database loading, and data examination, so this cleanup would be included in the next iteration.

A few more statistics of interest: there are 49 unique creators and 89 unique place names. More on this in the next section.

```
pipeline = [
    { "$group" : {"_id" : "$created", "count" : {"$sum":1}}},
    { "$group" : {"_id" : "unique users", "count" : {"$sum":1}}}
    ]

{u'ok': 1.0, u'result': [{u'_id': u'unique users', u'count': 49}]}


pipeline = [
    { "$group" : {"_id" : "$place_type", "count" : {"$sum":1}}},
    { "$group" : {"_id" : "unique users", "count" : {"$sum":1}}}
    ]

{u'ok': 1.0, u'result': [{u'_id': u'unique users', u'count': 89}]}
```

Additional Ideas:

Next, I looked at the distribution of creators, because I had noticed so many nodes/ways with no creator when I was doing the original cleanup:

```
pipeline = [
    { "$group" : {"_id" : "$created", "count" : {"$sum":1}}},
    { "$sort" : {"count":-1}}
    ]

u'ok': 1.0,
 u'result': [{u'_id': u'None', u'count': 400472},
```

```
             {u'_id': {u'user': u'ArcGIS Exporter'}, u'count': 41266},
             {u'_id': {u'user': u'NJ2002LULC'}, u'count': 2500},
             {u'_id': {u'user': u'Bing'}, u'count': 2124},
             {u'_id': {u'user': u'Rutgers'}, u'count': 616},
             {u'_id': {u'user': u'TIGER/Line\xae 2008 Place Shapefiles
(http://www.census.gov/geo/www/tiger/)'},
              u'count': 140},
             {u'_id': {u'user': u'bing'}, u'count': 131},
             {u'_id': {u'user': u'USGS Geonames'}, u'count': 107},
             etc.
```

I see that 89% of the documents had no creator identified, while the bulk of the remainder were created by "ArcGIS Exporter." According to their web site, "ArcGIS Online is a collaborative, cloud-based platform that allows members of an organization to use, create, and share maps, scenes, apps, and data, and access authoritative basemaps and ready-to-use apps."

I also see that there are values "Bing" and "bing" for created:user, so this should be cleaned up on the next iteration.

I then looked at place_types, to see the result of the mapping of keys shown in the table above:

```
pipeline = [
    { "$group" : {"_id" : "$place_type", "count" : {"$sum":1}}},
    { "$sort" : {"count":-1}}
    ]

{u'ok': 1.0,
 u'result': [{u'_id': None, u'count': 444908},
             {u'_id': u'parking', u'count': 972},
             {u'_id': u'school', u'count': 346},
             {u'_id': u'place_of_worship', u'count': 320},
             {u'_id': u'Populated Place', u'count': 155},
             {u'_id': u'restaurant', u'count': 115},
             {u'_id': u'fast_food', u'count': 62},
             {u'_id': u'grave_yard', u'count': 59},
             {u'_id': u'bank', u'count': 50},
             {u'_id': u'supermarket', u'count': 47},
             {u'_id': u'fire_station', u'count': 45},
             etc.
```

I see many different values for this place_type key, but nothing out of the ordinary, other than that these map data contain more graveyards than banks or supermarkets!

Finally, I went back to looking at creators, but this time only for the documents that had a non-null value for place_type:

```
pipeline = [
    { "$match" : { "place_type": {"$ne":None} } },
    { "$group" : {"_id" : "$created", "count" : {"$sum":1}}},
    { "$sort" : {"count":-1}}
    ]
```

```
{u'ok': 1.0,
 u'result': [{u'_id': u'None', u'count': 2501},
             {u'_id': {u'user': u'Bing'}, u'count': 133},
             {u'_id': {u'user': u'USGS Geonames'}, u'count': 53},
             {u'_id': {u'user': u'local_knowledge'}, u'count': 18},
             {u'_id': {u'user': u'local knowledge'}, u'count': 6},
             {u'_id': {u'user': u'bing, personal location survey'},
              u'count': 4},
             {u'_id': {u'user': u'Rutgers'}, u'count': 3},
             {u'_id': {u'user': u'osmsync:dero'}, u'count': 3},
             {u'_id': {u'user': u'Mapbox Satellite'}, u'count': 2},
             {u'_id': {u'user': u'NJ2002LULC'}, u'count': 2},
             {u'_id': {u'user': u'osmsync:dero; anonymous reader
comment'},
              u'count': 1},
             {u'_id': {u'user': u'Merkaartor 0.12'}, u'count': 1},
             {u'_id': {u'user': u'Local Knowledge'}, u'count': 1},
             {u'_id': {u'user': u'Potlatch 0.10f'}, u'count': 1}]}
```

The list in now quite different, in that ArcGIS is absent, and NJ2002LULC is much further down on the list. It is not surprising that the identified place names come from different sources than the simple nodes.

Conclusion:

This project showed how messy the OSM data can be, and it showed two different methods for cleaning the data: visual examination of the json file (which is more readable than the initial xml file), and queries of the data after they are loaded into MongoDB. This should be an iterative process, until the data are clean enough for their intended use.