Josh R Powell
(860) 977-2006

713 NW 1st Ave
Fort Lauderdale, FL 33311

OpenStreetMap Project

Map Area: Miami, FL, United States - mapzen.com/data/metro-extracts

TABLE OF CONTENTS

OVERVIEW

This project aims to assess the data quality for Miami, FL on OpenStreetMap for validity, accuracy, completeness, consistency and uniformity. Throughout the Wrangle OpenStreetMap Data project I have learned how to gather and parse data from popular file formats (json, xml, csv, html). I have also learned to store, query and aggregate data using MongoDB.

I will demonstrate what I've learned in the following sections.

GOALS

1. Process the Miami, FL OpenStreetMap data acquired on MapZen.com
    a. audit and clean the dataset
    b. convert it from XML to JSON
    c. import clean JSON file into a MongoDB database
2. Document the problems encountered along the way.

PROBLEMS ENCOUNTERED

Tag Review - mapparser.py & tags.py

For my initial review of the dataset I parsed the map file and built a dictionary of the XML tags to get an idea of what I could expect to work with in subsequent steps.

```
{'bounds': 1,
 'member': 42426,
 'nd': 1807642,
 'node': 1516787,
 'osm': 1,
 'relation': 1523,
 'tag': 1477841,
 'way': 198244}
```

In looking closer at the key tags I initially try and identify patterns within the data that would easily parse into a clean address for a dictionary to import into MongoDB. ~97% of the keys in the tags met the criteria of containing lower case or lower case with a single colon which would allow us to nicely parse the dictionary. Some of the other common differences were upper case and upper case with a single colon.

```
{'lower': 602295,
  'lower_colon': 829057,
  'lower_upper_colon': 2406,
  'other': 28768,
  'problemchars': 2,
  'upper': 15198,
  'upper_colon': 115}
```

Digging deeper I built a dictionary of the keys and found that we only had address information on ~20,000 locations. This seems like a small number of locations for such a large county. Something to investigate further.

```
{'addr:city': 20011,
 'addr:country': 18304,
 'addr:full': 18244,
 'addr:housename': 82,
 'addr:housenumber': 20311,
 'addr:postcode': 19615,
 'addr:state': 20683,
 'addr:street': 20445,
 'addr:suburb': 17254}
```

Audit - audit.py

A fair amount of auditing was involved in cleaning up the addresses. The majority of the address cleanup focused on the following (standard = action taken to normalize):

- Approve unique street types to this area, such as Causeway, Trace and Point.

FIELD	BEFORE	STANDARD	AFTER
street	1020 E Broward Blvd	convert abbreviations to full use full street types	1020 East Broward Boulevard
city	Coconut grove coconutgrove	titleized city names	Coconut Grove
state	Florida	abbreviated	FL
postalcode	33311-2018	+4 digits were sparsely populated, so remove	33311
building	Data Center	lowcase, '_' = space	data_center
URL	downtown.com	http://	http://downtown.com
phone	(561) 793-9980 954-463-7588 (954) 792-8181	varied - used phonenumbers library	(561) 793-9980 (954) 463-7588 (954) 792-8181
(ALL fields formatting)	parking Forestry Field_Office	spelling lowercase keys replace space w/ "_"	parkgin forestry_field_office
To be completed on next iteration ():			
source	Bing, bing, 'survey;Bing' Local Knowledge	varied	bing local_knowledge
power	substation, station		sub_station
surface	concrete:plates		concrete_plates

DATA OVERVIEW

File	Type	File Size (MB)
miami_florida	OSM	362.3
maimi_florida	JSON	399.3
miami_florida_sample	OSM	18.3

MongoDB Analysis

After importing the scrubbed OSM file into MongoDB I came across the following findings:

- # of Documents
  - `db.master1.find().count()`
  - `1,715,031`
- # of Nodes
  - `db.master1.find({"type":"node"}).count()`
  - `1,515,916`
- # of Unique Users
  - `len(db.master1.distinct("created.user"))`
  - `1,048`
- # of Users with 1 post - ~20% of of the users only have one post

```
{"$group": {"_id": "$created.user", "count": {"$sum": 1}}},
{"$group": {"_id": "$count", "num_users": {"$sum": 1}}},
{"$sort": {"_id": 1}},
{"$limit": 1}
```

  - `218`
- Top 10 Users - bots make up 2 of 10 largest contributors

```
{"$group": {"_id": "$created.user", "count": {"$sum": 1}}},
{"$sort": {"count": -1}},
{"$limit": 10}
```

```
{u'_id': u'grouper', u'count': 302908},
{u'_id': u'woodpeck_fixbot', u'count': 235504},
{u'_id': u'Latze', u'count': 137309},
{u'_id': u'carciofo', u'count': 92536},
{u'_id': u'freebeer', u'count': 78338},
{u'_id': u'bot-mode', u'count': 61612},
{u'_id': u'NE2', u'count': 58715},
{u'_id': u'westendguy', u'count': 49373},
{u'_id': u'Seandebasti', u'count': 48176},
{u'_id': u'georafa', u'count': 39763}
```

- Postal Codes by User - 1 user has provided the bulk of the postal code contributions for zip codes in Weston, FL

```
{"$match": {"address.postcode": {"$exists":1}}},
{"$group": {"_id": {"postcode":"$address.postcode",
                    "user":"$created.user"
                   }, "postcode_count": {"$sum": 1}}},
{"$sort": {"postcode_count": -1}},
{"$limit": 10}
```

```
{u'_id': {u'postcode': u'33327', u'user': u'maggot27'}, u'postcode_count': 6769},
 {u'_id': {u'postcode': u'33326', u'user': u'maggot27'}, u'postcode_count': 6462},
 {u'_id': {u'postcode': u'33331', u'user': u'maggot27'}, u'postcode_count': 3112},
 {u'_id': {u'postcode': u'33332', u'user': u'maggot27'}, u'postcode_count': 1888},
 {u'_id': {u'postcode': u'33414', u'user': u'Seandebasti'}, u'postcode_count': 65},
 {u'_id': {u'postcode': u'33304', u'user': u'thetornado76'}, u'postcode_count': 60},
 {u'_id': {u'postcode': u'33131', u'user': u'georafa'}, u'postcode_count': 31},
 {u'_id': {u'postcode': u'33140', u'user': u'thetornado76'}, u'postcode_count': 22},
 {u'_id': {u'postcode': u'33138', u'user': u'thetornado76'}, u'postcode_count': 21},
 {u'_id': {u'postcode': u'33178', u'user': u'williehlh'}, u'postcode_count': 20}
```

- Key overview
  - I found a nice tool for analyzing a Mongo DB schema here, https://github.com/variety/variety
  - I used this to choose additional categories to analyze
  - I noticed the “tiger:[x]” categories make up a large majority of the contributions
    - more info on tiger data can be found here: http://www.census.gov/geo/maps-data/data/tiger.html

key	types	occurrences	percents
_id	ObjectId	1715031	100.00000000000000000000
created	Object	1715031	100.00000000000000000000
created.changeset	String	1715031	100.00000000000000000000
created.timestamp	String	1715031	100.00000000000000000000
created.uid	String	1715031	100.00000000000000000000
created.user	String	1715031	100.00000000000000000000
created.version	String	1715031	100.00000000000000000000
id	String	1715031	100.00000000000000000000
type	String	1715031	100.00000000000000000000
pos	Array	1516787	88.44079203233060582079
node_refs	Array	198244	11.559207967669388850 14
highway	String	161288	9.40437811328191664018
name	String	116369	6.78524178280159340204
tiger:county	String	88356	5.15186022876554439165
tiger:cfcc	String	88176	5.14136479165682747094
tiger:reviewed	String	87305	5.09057853764742418434
tiger:name_base	String	81827	4.77116740163880415082
tiger:name_type	String	79221	4.61921679549815689114
tiger:zip_left	String	65459	3.81678232055280641788
tiger:zip_right	String	63253	3.68815490798708589537
tiger:name_direction_prefix	String	52746	3.07551292075770055234
building	String	34769	2.02731029351656033555
source	String	34449	2.00865173865661894581
address	Object (22133),String (1)	22134	1.29058891646856532809
power	String	21542	1.25607058997767384589
oneway	String	21162	1.23391355608149355660

- Top 10 Waterway Categories - unique to coastal area like South Florida

```
{"$group": {"_id": "$waterway", "count": {"$sum": 1}}},
{"$sort": {"count": -1}},
{"$limit": 10}
```

```
{u'_id': None, u'count': 1711186},
 {u'_id': u'canal', u'count': 2952},
 {u'_id': u'weir', u'count': 397},
 {u'_id': u'riverbank', u'count': 122},
 {u'_id': u'drain', u'count': 118},
 {u'_id': u'stream', u'count': 90},
 {u'_id': u'ditch', u'count': 62},
 {u'_id': u'river', u'count': 42},
 {u'_id': u'yes', u'count': 38},
 {u'_id': u'dock', u'count': 12}
```

- Top 10 ‘Natural’ categories - give you a sense of the landscape

```
{"$group": {"_id": "$natural", "count": {"$sum": 1}}},
{"$sort": {"count": -1}},
{"$limit": 10}
```

```
{u'_id': None, u'count': 1705241},
 {u'_id': u'water', u'count': 5714},
 {u'_id': u'tree', u'count': 2479},
 {u'_id': u'sand', u'count': 433},
 {u'_id': u'coastline', u'count': 358},
 {u'_id': u'wood', u'count': 266},
 {u'_id': u'tree_row', u'count': 230},
 {u'_id': u'wetland', u'count': 150},
 {u'_id': u'scrub', u'count': 63},
 {u'_id': u'beach', u'count': 47
```

- Top 10 ‘Leisure’ categories - south florida is a big tourist destination!

```
{"$group": {"_id": "$leisure", "count": {"$sum": 1}}},
{"$sort": {"count": -1}},
{"$limit": 10}
```

```
{u'_id': None, u'count': 1710726},
 {u'_id': u'pitch', u'count': 1969},
 {u'_id': u'park', u'count': 1013},
 {u'_id': u'swimming_pool', u'count': 713},
 {u'_id': u'playground', u'count': 104},
 {u'_id': u'golf_course', u'count': 100},
 {u'_id': u'sports_centre', u'count': 79},
 {u'_id': u'stadium', u'count': 56},
 {u'_id': u'marina', u'count': 45},
 {u'_id': u'track', u'count': 43}
```

ADDITIONAL IDEAS

After exploring the data set some additional ways of improving and analyzing the data are implementing quality assurance tools such as the following:

- **Error detection tools** that check data for potential errors, inaccuracies or missing data. For example, amenities without types: a Thai restaurant missing the value Thai, etc.
    - Benefits: automatic detection can weed out clean data from data that has characteristics that may need further review. The would allow contributors to focus pockets that need more focus than others.
    - Challenges: automatic detection is not uniform and would likely have to be established at a local level. As the data quality improves over time the detection tools need to evolve. This would require active development of the tools over time. Perhaps this concern could be alleviated some with Machine Learning.
- **Monitoring tools** that spot changes / edit. For example, users who have thoroughly mapped and area might want to follow changes that occur to their data set.
    - Benefits: active users familiar with a particular area can keep track of any changes that may be applied by others.
    - Challenges: the number of updates could be too much to manage. There would have to be some kind of rollback mechanism in place if the change had a negative impact on the data.
- **Visualization tools** that provide third party overlays that provide a representation of activity in an area.
    - Benefits: Something like cell phone tracking data could visually point contributors to areas that need more attention and contributions.
    - Challenges: Is this data publicly available? Who would have this data? Phone companies / services like Yelp / Uber? Will contributing the data expose a competitive advantage? The incentives here may not be aligned.