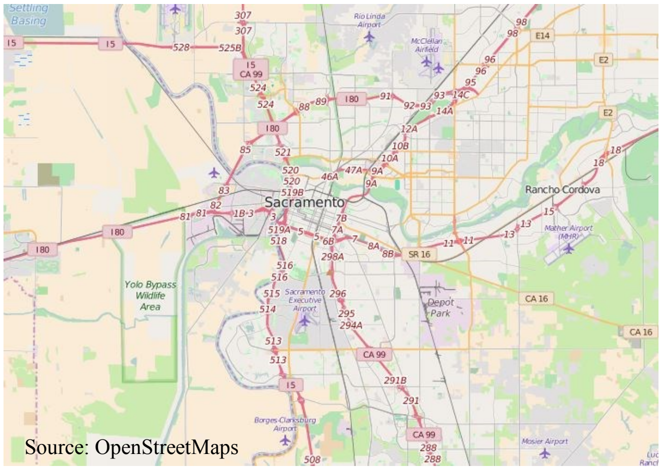
Exploring Sacramento, California, with MongoDB
Author: Stas Sajin



Source: OpenStreetMaps

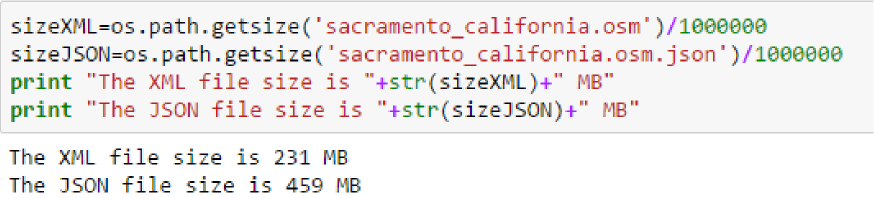
General Purpose:
The purpose of this document is to wrangle OpenStreetMaps data for the City of Sacramento, California.

Sacramento is the capital of California, with an estimated population of about half a million. In 2002, the Time magazine nicknamed Sacramento as one of the most diverse city in the United States. Although I will not be performing a cross-city comparison for a relative measure of diversity, I will look at how diverse is Sacramento when it comes to its cuisine and places of worship.

Source: Wikipedia Commons

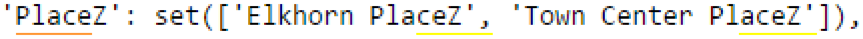
Panoramic view of downtown Sacramento from West Sacramento

Data Source:
I downloaded the data from Mapzen. The XML file can be found here. The structure of the XML tree can be found here.
The XML file size was 231MB and JSON file ended up being 459MB.

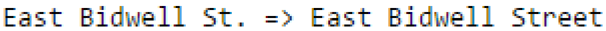

```
sizeXML=os.path.getsize('sacramento_california.osm')/1000000
sizeJSON=os.path.getsize('sacramento_california.osm.json')/1000000
print "The XML file size is "+str(sizeXML)+" MB"
print "The JSON file size is "+str(sizeJSON)+" MB"

The XML file size is 231 MB
The JSON file size is 459 MB
```

Problems encountered in the map:
When I examined the user contributions, I noticed that the top five users were automated systems. Because of automation, most of the data was relatively clean. Nonetheless I found the following issues:

- There were several typos (e.g., Town Center Placez should be Town Center Plaza)


```
'PlaceZ': set(['Elkhorn PlaceZ', 'Town Center PlaceZ']),
```

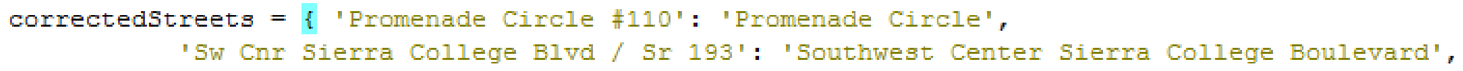
- A lot of streets type names were abbreviated, so I fixed this by providing the correct street name mapping (e.g., St. was changed into Street)


```
East Bidwell St. => East Bidwell Street
```

- Postal codes for Sacramento range from 94000-96000. During data auditing, I noticed that one address had the code 96816, which corresponds to Honollulu, Hawai rather than Sacramento. I changed the postal code to none when the postal code was outside of the permitted range.
- I removed the tags with some nonsense address names (e.g., 9311,9321, PO Box 5259, Industrial)

Data Issues Continued:

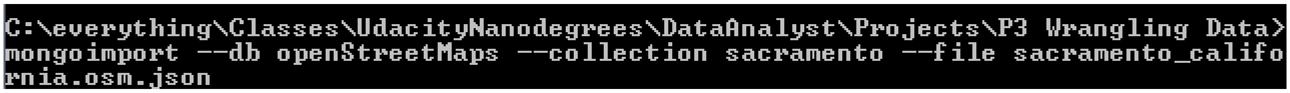
- I corrected some street addresses. See below:



```
correctedStreets = { 'Promenade Circle #110': 'Promenade Circle',
            'Sw Cnr Sierra College Blvd / Sr 193': 'Southwest Center Sierra College Boulevard',
```

Data Importing:

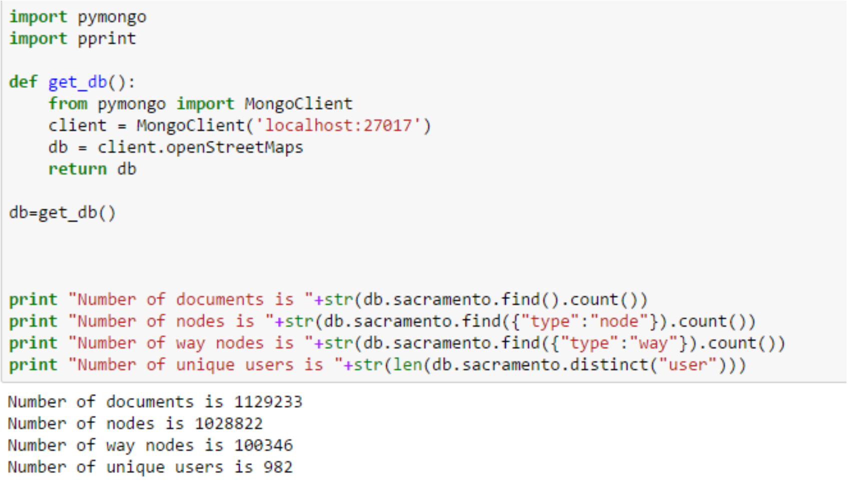
The json file has been imported in MongoDB using mongoimport through the command line:



```
C:\everything\Classes\UdacityNanodegrees\DataAnalyst\Projects\P3 Wrangling Data>
mongoimport --db openStreetMaps --collection sacramento --file sacramento_califo
rnia.osm.json
```

Data Summary and MongoDB queries

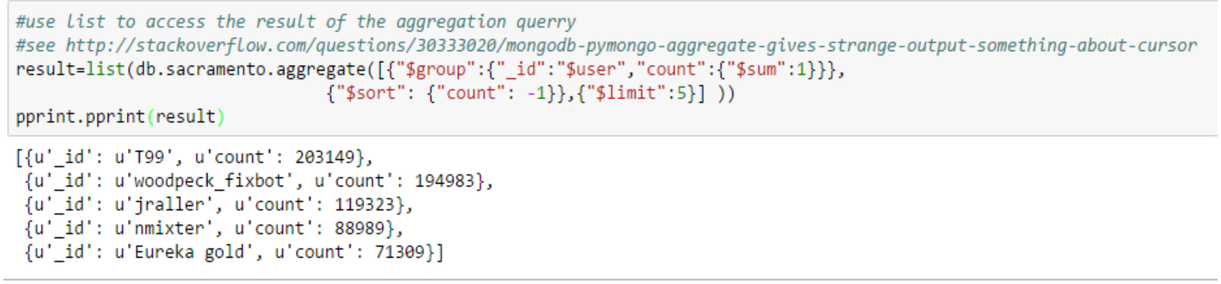
- Number of documents in the database, the nodes, way nodes, and unique user:



```
import pymongo
import pprint

def get_db():
    from pymongo import MongoClient
    client = MongoClient('localhost:27017')
    db = client.openStreetMaps
    return db

db=get_db()



print "Number of documents is "+str(db.sacramento.find().count())
print "Number of nodes is "+str(db.sacramento.find({"type":"node"}).count())
print "Number of way nodes is "+str(db.sacramento.find({"type":"way"}).count())
print "Number of unique users is "+str(len(db.sacramento.distinct("user")))
```

```
Number of documents is 1129233
Number of nodes is 1028822
Number of way nodes is 100346
Number of unique users is 982
```

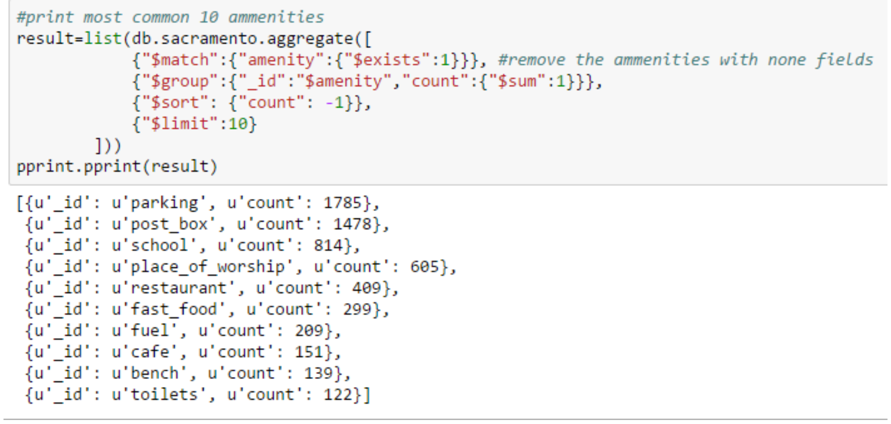
- The counts for top 5 contributing users:



```
#use list to access the result of the aggregation querry
#see http://stackoverflow.com/questions/30333020/mongodb-pymongo-aggregate-gives-strange-output-something-about-cursor
result=list(db.sacramento.aggregate([{"$group":{"_id":"$user","count":{"$sum":1}}},
                                {"$sort": {"count": -1}},{"$limit":5}] ))
pprint.pprint(result)
```

```
[{u'_id': u'T99', u'count': 203149},
 {u'_id': u'woodpeck_fixbot', u'count': 194983},
 {u'_id': u'jraller', u'count': 119323},
 {u'_id': u'nmixter', u'count': 88989},
 {u'_id': u'Eureka gold', u'count': 71309}]
```

- Let's look at the top 10 most common amenities:

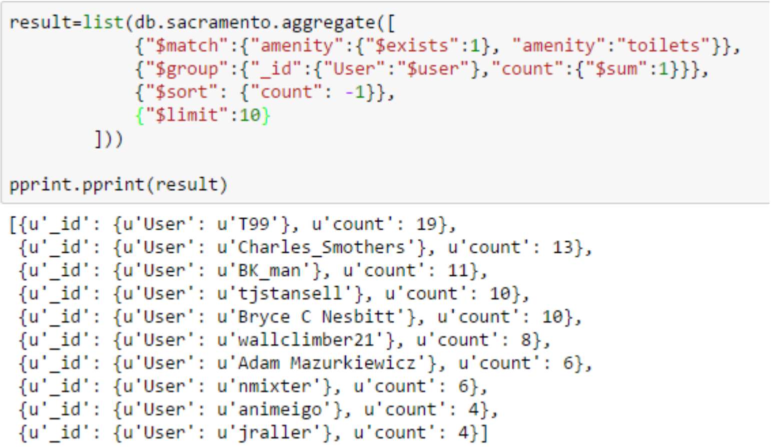


```
#print most common 10 ammenities
result=list(db.sacramento.aggregate([
            {"$match":{"amenity":{"$exists":1}}}, #remove the ammenities with none fields
            {"$group":{"_id":"$amenity","count":{"$sum":1}}},
            {"$sort": {"count": -1}},
            {"$limit":10}
        ]))
pprint.pprint(result)
```

```
[{u'_id': u'parking', u'count': 1785},
 {u'_id': u'post_box', u'count': 1478},
 {u'_id': u'school', u'count': 814},
 {u'_id': u'place_of_worship', u'count': 605},
 {u'_id': u'restaurant', u'count': 409},
 {u'_id': u'fast_food', u'count': 299},
 {u'_id': u'fuel', u'count': 209},
 {u'_id': u'cafe', u'count': 151},
 {u'_id': u'bench', u'count': 139},
 {u'_id': u'toilets', u'count': 122}]
```

Data Summary and MongoDB queries continued:

It seems pretty strange to have so many toilets and benches listed as amenities. Let’s see if there is any particular user who has an obsessions for tagging toilets on the map.

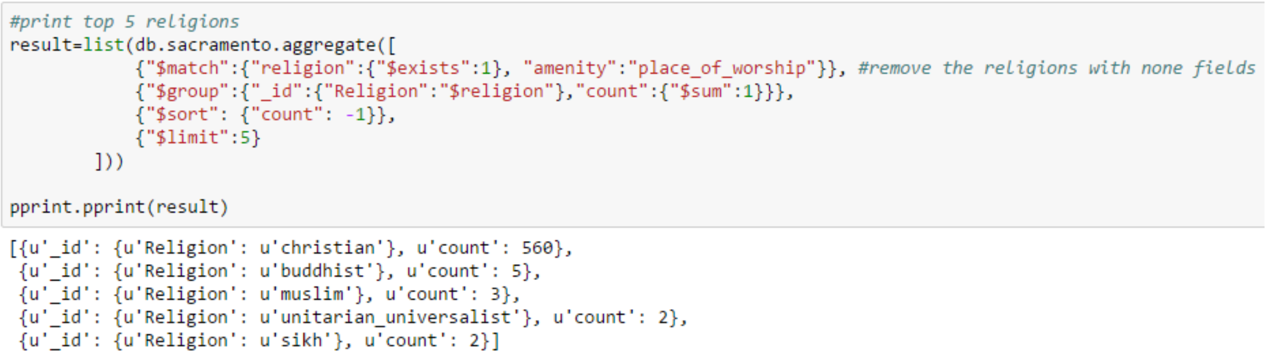


```
result=list(db.sacramento.aggregate([
            {"$match":{"amenity":{"$exists":1}, "amenity":"toilets"}},
            {"$group":{"_id":{"User":"$user"},"count":{"$sum":1}}},
            {"$sort": {"count": -1}},
            {"$limit":10}
        ]))

pprint.pprint(result)
```

```
[{u'_id': {u'User': u'T99'}, u'count': 19},
 {u'_id': {u'User': u'Charles_Smothers'}, u'count': 13},
 {u'_id': {u'User': u'BK_man'}, u'count': 11},
 {u'_id': {u'User': u'tjstansell'}, u'count': 10},
 {u'_id': {u'User': u'Bryce C Nesbitt'}, u'count': 10},
 {u'_id': {u'User': u'wallclimber21'}, u'count': 8},
 {u'_id': {u'User': u'Adam Mazurkiewicz'}, u'count': 6},
 {u'_id': {u'User': u'nmixter'}, u'count': 6},
 {u'_id': {u'User': u'animeigo'}, u'count': 4},
 {u'_id': {u'User': u'jraller'}, u'count': 4}]
```

Well, it seems like T99 and Charles_Smothers have helped a lot of people in need of a toilet.

- Is Sacramento Religiously Diverse?

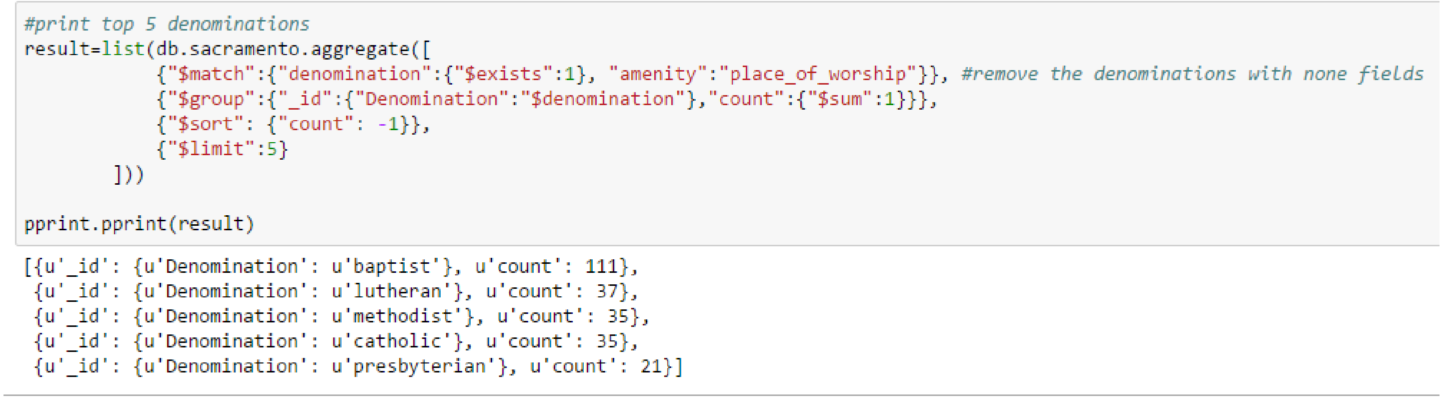


```
#print top 5 religions
result=list(db.sacramento.aggregate([
            {"$match":{"religion":{"$exists":1}, "amenity":"place_of_worship"}}, #remove the religions with none fields
            {"$group":{"_id":{"Religion":"$religion"},"count":{"$sum":1}}},
            {"$sort": {"count": -1}},
            {"$limit":5}
        ]))

pprint.pprint(result)
```

```
[{u'_id': {u'Religion': u'christian'}, u'count': 560},
 {u'_id': {u'Religion': u'buddhist'}, u'count': 5},
 {u'_id': {u'Religion': u'muslim'}, u'count': 3},
 {u'_id': {u'Religion': u'unitarian_universalist'}, u'count': 2},
 {u'_id': {u'Religion': u'sikh'}, u'count': 2}]
```

A huge majority of places of workshop in Sacramento are Christian.

- What are the top 5 religious denominations?



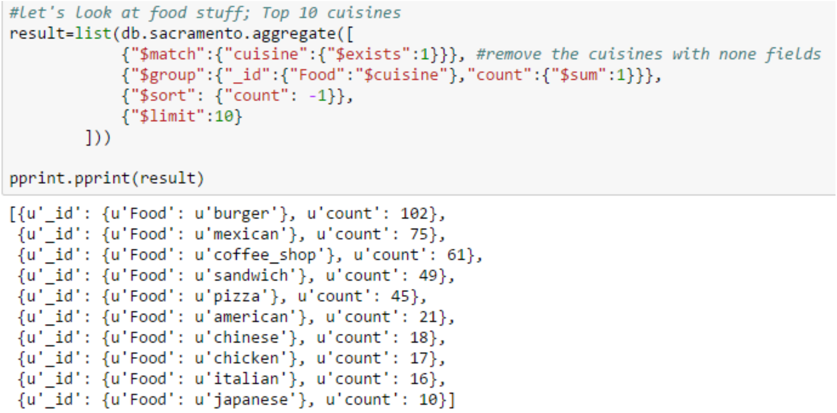
```
#print top 5 denominations
result=list(db.sacramento.aggregate([
            {"$match":{"denomination":{"$exists":1}, "amenity":"place_of_worship"}}, #remove the denominations with none fields
            {"$group":{"_id":{"Denomination":"$denomination"},"count":{"$sum":1}}},
            {"$sort": {"count": -1}},
            {"$limit":5}
        ]))

pprint.pprint(result)
```

```
[{u'_id': {u'Denomination': u'baptist'}, u'count': 111},
 {u'_id': {u'Denomination': u'lutheran'}, u'count': 37},
 {u'_id': {u'Denomination': u'methodist'}, u'count': 35},
 {u'_id': {u'Denomination': u'catholic'}, u'count': 35},
 {u'_id': {u'Denomination': u'presbyterian'}, u'count': 21}]
```

The top religious denomination seems to be Baptist.

Data Summary and MongoDB queries continued:

- What are the top 10 cuisines in Sacramento?



```
#let's look at food stuff; Top 10 cuisines
result=list(db.sacramento.aggregate([
            {"$match":{"cuisine":{"$exists":1}}}, #remove the cuisines with none fields
            {"$group":{"_id":{"Food":"$cuisine"},"count":{"$sum":1}}},
            {"$sort": {"count": -1}},
            {"$limit":10}
        ]))

pprint.pprint(result)
```

```
[{u'_id': {u'Food': u'burger'}, u'count': 102},
 {u'_id': {u'Food': u'mexican'}, u'count': 75},
 {u'_id': {u'Food': u'coffee_shop'}, u'count': 61},
 {u'_id': {u'Food': u'sandwich'}, u'count': 49},
 {u'_id': {u'Food': u'pizza'}, u'count': 45},
 {u'_id': {u'Food': u'american'}, u'count': 21},
 {u'_id': {u'Food': u'chinese'}, u'count': 18},
 {u'_id': {u'Food': u'chicken'}, u'count': 17},
 {u'_id': {u'Food': u'italian'}, u'count': 16},
 {u'_id': {u'Food': u'japanese'}, u'count': 10}]
```

Sac has quite a bit of variety when it comes to cuisines. These numbers correspond pretty well with the proportion of different races based on the Cenus Data for Sacramento County.

For the list of other queries, see the end portion of the WranglingCode.ipynb

Other ideas about the dataset

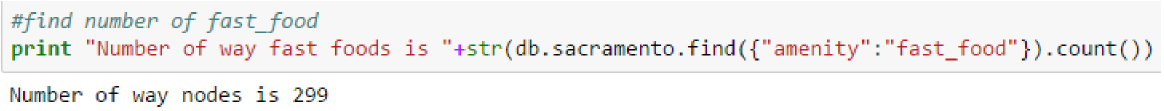
One issue that I observed in this dataset is that the postal codes don't have a standardized format. For instance, some postal codes are 5 digit numbers (e.g., 95832), some are in the format of 5digits followed by 4 digits (95832-1447), and some have non-digit strings (CA 95382). Homogeneity of the data could be further improved by changing the latter two formats into the former one. One way of improving the postal code data is to ask user to include only 5-digit integer numbers when it comes to regions in the US. If they include more than five digits or characters, they would get a notification telling them that the inputted format is incorrect. A potential issue with this solution is that some volunteer mappers might feel like we are putting too much burden on how the postal codes should be inputted.

The other issue is that the data that we have has not been cross-validated. In other words, without some domain knowledge and familiarity about a city, we can't be sure if someone included the right address or not. One possible way of cross-validating our data is to use google api. The main issues with this is that there is usually a limit when it comes to the number of queries that can be made through the google api, so it is best that we make queries selectively on data that is more likely to contain errors.

-For instance, we could cross-validate only data that comes from users that make few submissions, under the assumption that data that comes from user bots or experienced mappers is usually more accurate than human inserted data from new, less experienced volunteers.

-We can also identify users who often make mistakes during mapping, and decide that the data imputed by those users should receive more scrutiny during cross-validation.

One thing that surprised me was the small number of fast foods in the area relative to population size. There are a total of 299 fast foods in the dataset. For a population of about 450k, it comes to 1505 people per fast food restaurant. I moved recently from NYC, and the ratio there comes more to 935 people per fast food restaurant. I think that the smaller number of fast food restaurants might be due to the fact that Sacramento has a similar median annual income to NYC (about 51k in each city), even though NYC is much more expensive to live in. In other words, inhabitants in NYC are poorer, in relative terms, to people living in Sacramento. Smaller relative wages are likely to push people to make less healthy, and more cheap food consumption choices.



```
#find number of fast_food
print "Number of way fast foods is "+str(db.sacramento.find({"amenity":"fast_food"}).count())
```

```
Number of way nodes is 299
```