Mark Ciganovic
Project 4

The list of steps involved in this project

1. Select a personalized osm file for data analysis
2. Wrangling the data by standardizing the geographical values, and then preparing the file for conversion into csv files for creation of a SQL database for further analysis
3. Create queries to answer basic questions about the data
4. Create queries to answer specific questions of interest about the database
5. Create queries to suggest ways to improve the data for further analyses

Map Area Selected

I selected the area around where I live which is Northwest Indiana, with a small amount of Illinois that borders this area of Indiana. This area is mainly Lake and Porter County in Indiana. The reason that I chose this area is because I live here.
https://www.openstreetmap.org/export#map=10/41.5065/-87.1909
overpass-api.de/api/map?bbox=-87.5212,41.3479,-86.9060,41.6975

Data Wrangling

As far as the data wrangling was concerned, my goal was to standardize the geographical values in the data, which included street, state, postcode, and city. As far as street values were concerned, the desired format was the full name with the first letter capitalized, with values like Street and Place. For state values, the desired format was the standard two letter abbreviation where both letters are capitalized like IN, and IL. For postcode it was the standard five digit postcode with the four digits after a hyphen, and for city names the format with the name with the first letter capitalized with the rest in lower case. For the most part, the values fit this format with a few deviations, which were corrected in the Project4.py. My corrections were specific to the dataset, and the mapping variable shows all the corrections for the geographical values.

```
mapping = { "St": "Street",
        "PLACE": "Place",
        "Ave" : "Avenue",
        "in" : "IN",
        "Indiana" : "IN",
        "46410-5468" : "46410",
        "portage" : "Portage",
        }
```

Queries that answer basic questions about the database

Size of the file

```
sqlite> PRAGMA PAGE_SIZE;
```

```
1024
sqlite> PRAGMA PAGE_COUNT;
51916
```

The size is 53,161,984 bytes, which is the product of the first two queries

Number of unique users

```
sqlite> SELECT COUNT(*)
   ...> FROM (SELECT DISTINCT(uid) FROM nodes UNION SELECT DISTINCT(uid) FROM ways);
445
```

Number of nodes

```
sqlite> SELECT COUNT(*) FROM nodes;
318345
```

Number of ways

```
sqlite> SELECT COUNT(*) FROM ways;
35151
```

Queries that answer specific questions of interest from the database

Step 1 – Find the different types of keys in nodes tags to find area of interest

```
 sqlite> SELECT key, COUNT(*)
   ...> FROM nodes_tags
   ...> GROUP BY key
   ...> ORDER BY COUNT(*) DESC
   ...> LIMIT 25;
power|3022
highway|2590
source|2017
railway|1772
name|1587
amenity|1113
ele|1069
feature_id|958
created|882
county_id|879
state_id|878
religion|520
denomination|290
fixme|290
place|247
import_uuid|187
is_in|122
Class|119
County|119
ST_alpha|119
ST_num|119
id|119
County_num|118
```

```
leisure|111
ref|104
```

Step 2 – Find the different types of values in leisure to select area of interest

```
sqlite> SELECT value, COUNT(*)
   ...> FROM nodes_tags
   ...> WHERE key = "leisure"
   ...> GROUP BY value
   ...> ORDER BY COUNT(*) DESC;
park|69
pitch|15
playground|12
sports_centre|6
slipway|4
marina|2
picnic_table|2
swimming_pool|1
```

Step 3 – Find the name and number for the different sports centers in the area

```
sqlite> SELECT value, COUNT(*)
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "sports_centre")
   ...> AND key = "name"
   ...> GROUP BY value
   ...> ORDER BY COUNT(*) DESC;
YMCA|2
Planet Fitness|1
The Courts|1
World Gym|1
```

From my personal knowledge of the area, this information is not correct. There are other gyms (sports_centre) besides the ones listed here. First of all, I can think of anytime fitness, which is the gym that I go to, and there are many in the area. Also, there are two World Gyms that I can think of, one in Schererville, and another in Highland. The data is definitely incomplete. The step that follows is to create queries that could help in further auditing of the data.

Queries that suggest ways to improve the data for further analysis

Step 1 – Find the different types of values in amenity to select area of interest

```
sqlite> SELECT value, COUNT(*)
   ...> FROM nodes_tags
   ...> WHERE key = "amenity"
   ...> GROUP BY value
   ...> ORDER BY COUNT(*) DESC
   ...> LIMIT 20;
place_of_worship|533
school|222
fuel|55
restaurant|38
grave_yard|37
```

```
fast_food|35
fire_station|30
parking|19
library|17
post_office|17
bank|14
pharmacy|9
townhall|7
bar|6
atm|5
cafe|5
fountain|5
bench|4
clinic|4
police|4
```

Step 2 – Find the count of the distinct names of values in restaurant and fast food

```
sqlite> SELECT DISTINCT(value)
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "restaurant" OR value = "fast_food")
   ...> AND key = "name"
   ...> ORDER BY value;
3 Floyd's Brewpub
444 Grill
5 Guys Burgers and Fries
Aj's Pizza
Arby's
Baker's Square
Beggar's Pizza
Bob Evans
Buffalo Wild Wings
Burger King
Burgerking
Cedar Lake Kitchen
Chipotle
Culver's
Dairy Queen
Dino's Pizza
Edwardo's Natural Pizza
El Salto
Fifth Avenue Gyros and Mexican Foods
Georges Gyros
Industrial Revolution
Jade Garden
Jimmie's Coney Island No 1
Jimmy John's
Jonathan's Pancake House
Long John Silver's
Lucrezia's Cafe & Catering
McDonald's
McDonald's
```

```
Miller Bakery Cafe
Miller Pizza
Panera Bread
Papa John's Pizza
PePe's
Pizza Hut
Pizza Hut / Taco Bell
Ponderosa Steakhouse
Portillo's Hot Dogs
Rosewood
Round the Clock
Schererville lounge
Subway
Subway & Dairy Queen
Taco Bell
Texas Corral
The Sandbar Grill
The Scrambled Diner
Toast and Jam
Valpo Vienna
Viking Chili Bowl
Wagner's Ribs
Wendy's
```

The one problem that I notice is that McDonald’s is listed twice under the same spelling, and Burger King is listed twice under different spellings. The problem with Burger King is one that can be corrected during auditing, the problem with McDonald’s is one that requires further analysis.

Step 3 – Find the cuisine values for the different restaurants and fast food establishments

```
sqlite> SELECT DISTINCT(n.value) as name, c.value as cuisine
   ...> FROM
   ...> (SELECT id, key, value
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "restaurant" OR value = "fast_food")
   ...> AND key = "name") as n
   ...> INNER JOIN
   ...> (SELECT id, key, value
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "restaurant" OR value = "fast_food")
   ...> AND key = "cuisine") as c
   ...> ON n.id = c.id
   ...> ORDER BY name;
444 Grill|caribbean
5 Guys Burgers and Fries|burger
Beggar's Pizza|pizza
Bob Evans|american
Buffalo Wild Wings|wings
Burger King|american
Burger King|burger
```

```
Burgerking|burger
Chipotle|mexican
Culver's|burger
Dairy Queen|ice_cream
Dino's Pizza|pizza
Edwardo's Natural Pizza|pizza
El Salto|mexican
Fifth Avenue Gyros and Mexican Foods|mexican
Industrial Revolution|burger
Jade Garden|chinese
Jimmy John's|sandwich
McDonald's|american
McDonald's|burger
Miller Bakery Cafe|american
Miller Pizza|pizza
Panera Bread|american
PePe's|mexican
Pizza Hut|pizza
Subway|sandwich
Taco Bell|mexican
Texas Corral|steak_house
Viking Chili Bowl|American
```

I see some problems in the way this data is structured. There should be only one type of cuisine for a specific restaurant. Both Burger King and McDonalds have two types of cuisines, and that is “american” and “burger”. This should be changed to just one type of cuisine. This could also explain why McDonald’s was listed twice under the same spelling.

Step 4 – Find the restaurants that don’t have a cuisine listed for them

```
sqlite> SELECT l.name
   ...> FROM
   ...> (SELECT DISTINCT(n1.value) as name
   ...> FROM
   ...> (SELECT id, key, value
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "restaurant" OR value = "fast_food")
   ...> AND key = "name") as n1
   ...> INNER JOIN
   ...> (SELECT id, key, value
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "restaurant" OR value = "fast_food"))
   ...> as c1
   ...> ON n1.id = c1.id) as l
   ...> LEFT JOIN
   ...> (SELECT DISTINCT(n2.value) as name
   ...> FROM
   ...> (SELECT id, key, value
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "restaurant" OR value = "fast_food")
   ...> AND key = "name") as n2
   ...> INNER JOIN
   ...> (SELECT id, key, value
   ...> FROM nodes_tags
   ...> WHERE id IN (SELECT id FROM nodes_tags WHERE value = "restaurant" OR value = "fast_food")
   ...> AND key = "cuisine") as c2
   ...> ON n2.id = c2.id) as r
   ...> ON l.name = r.name
   ...> WHERE r.name IS NULL
   ...> ORDER BY l.name;
3 Floyd's Brewpub
Aj's Pizza
Arby's
Baker's Square
Cedar Lake Kitchen
Georges Gyros
Jimmie's Coney Island No 1
Jonathan's Pancake House
Long John Silver's
Lucrezia's Cafe & Catering
McDonald's
Papa John's Pizza
Pizza Hut / Taco Bell
Ponderosa Steakhouse
Portillo's Hot Dogs
Rosewood
Round the Clock
Schererville lounge
Subway & Dairy Queen
The Sandbar Grill
The Scrambled Diner
Toast and Jam
Valpo Vienna
Wagner's Ribs
Wendy's
```

This is definitely incomplete, types of cuisines could be added for many of these restaurants and fast food establishments. For example, Wendy’s would be under “burger”, and Papa John’s Pizza under “pizza”. There also seems to be the problem that a certain McDonald’s does not have a cuisine along with McDonald’s having “burger” and “american” as well.

Summary

I have found the information gathered from the osm data file to not be very informative. First of all, when I wanted to search for gyms / fitness centers, I found the listings not representing the area that well. There are many gyms and fitness centers that were left out, like the one that I go to. When it came to the analysis of the restaurants and fast foods, there is definitely the need for some auditing the correct the data. First of all, there needs to be one spelling for Burger King, and not two. Also, the classification of Burger King into both “american” and “burger” is something that needs to be corrected. There should be only one cuisine that it is listed under. Also, the same holds true for McDonald's, it is listed under both "american" and "burger", along with having a value that has no cuisine. The next step would be to go through the file and clean this up, probably in the mapping variable. As far as the restaurants and fast food restaurants that do not have a cuisine listed for them, I just see that as a problem with the incompleteness of the data. Sometimes that data that one gets is not perfect, but you have to make the best with what you have. The benefits of my suggestions would make sure that there is only one name for each restaurants or fast food establishments, so that any type of summary data on that restaurant or fast food establishment, or the count of distinct restaurants or fast food establishments would yield correct information. If Burger King is spelled multiple ways then the count of Burger Kings in the area would not be correct, since that count would be spread over multiple names, also if there are multiple spellings for restaurants then the count of distinct restaurants would not be accurate as well. There is a definite need for standardization when it comes to spelling to extract better information from the data set. Also, I do believe it would be a better practice to make one cuisine value per restaurant. This would help when it comes to summary information, when listing the restaurant and its cuisine type, we should want the restaurant, and its cuisine type listed, not duplicates, and duplicates would come from restaurants having multiple cuisine values. It also shows a standardization that is taking place as far as having more of a controlled vocabulary, and not just whatever term someone chooses arbitrarily. As far as problems are concerned, I think it would be very easy to standardize the spellings of the names. The mapping variable would be extended beyond geographical values to restaurant and fast food values. The standardization of cuisines would be a little more difficult, since there would have to be an agreed upon controlled vocabulary, and that would be an involved project to find out how to initiate that for the Open Street Map. It would be a very intense project to create a controlled vocabulary for all the keys and values in the Open Street Map. The positive is that standardizing data like that will make the data richer, so that more meaningful insights can be gleaned from it.