Udacity OSM Project - Irvine, CA - Randy Crane

Introduction

In this project I will be exploring OpenStreetMap data for Irvine, California. I chose this map data because Irvine is my current city of residence. I am already pretty familiar with the area, which will potentially help me recognize some of the bad data in the set. At the same time, I'm curious about some aspects of my city and I will use this data set to glean some new insights about it.

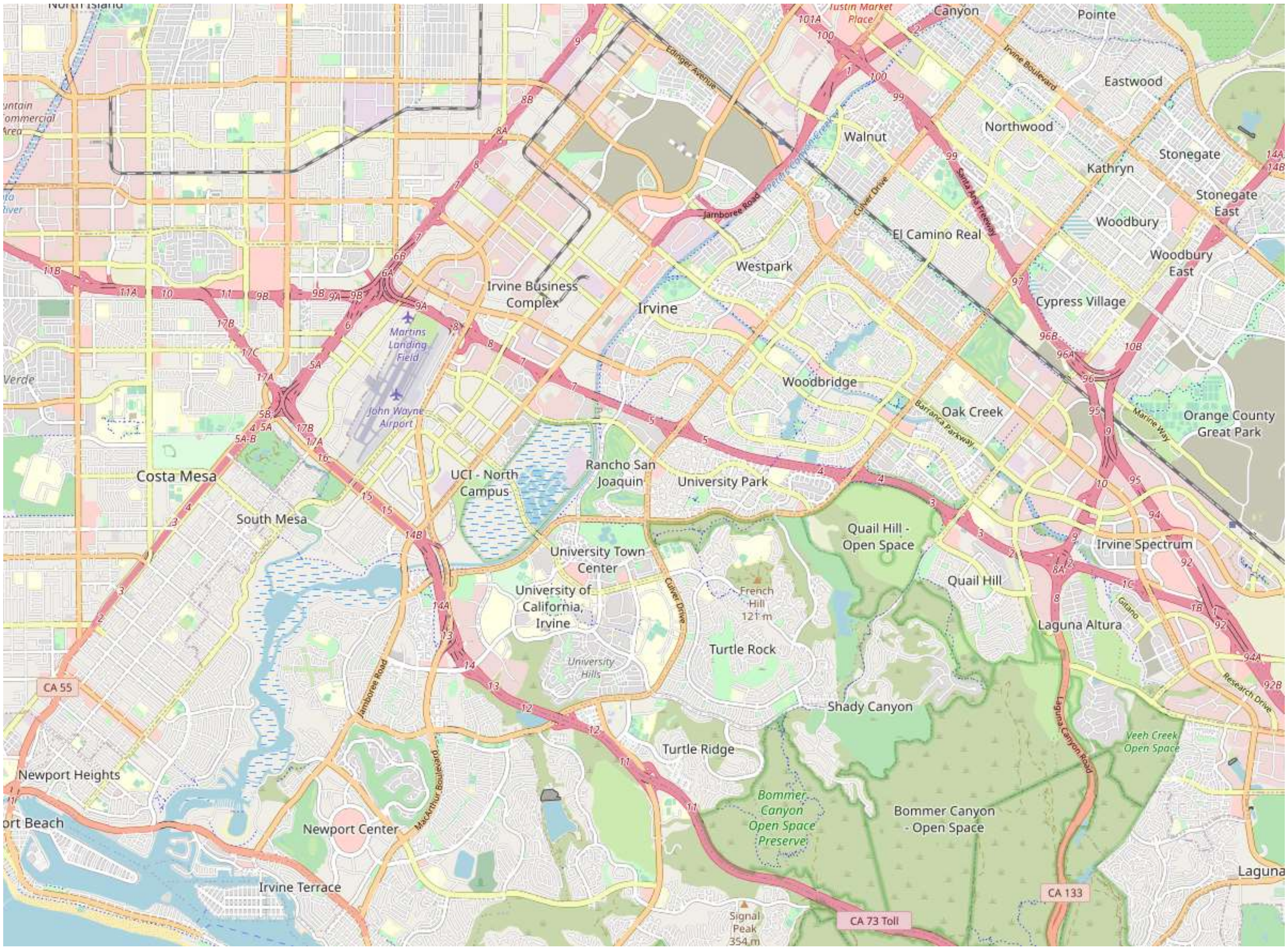
The data consists of XML elements called "nodes" (points of interest) and "ways" (linear features and area boundaries). Each element can have one or more tags associated with it. These tags provide more details and information about the element. Additional information about the data structure can be found here: https://wiki.openstreetmap.org/wiki/OSM_XML (https://wiki.openstreetmap.org/wiki/OSM_XML)

I will first explore the raw XML data using Python and audit tags that are appear to have errors, and tags that I'm interested in analyzing in more detail. Due to the scope of this project, I will not be cleaning every error I find, but will focus on two or three to meet the requirements of the project. After identifying and cleaning the problem areas, I will export the data to CSV files, then import it into a SQLite database. Finally, I will query the database to gain greater insights.

Map Area

On OpenStreetMap.org, the area I selected to include as much of Irvine I could without too much data from other cities is bounded within the following coordinates:

Longitude: -117.8792, -117.7281 </p> Latitude: 33.7301, 33.6135


Canyon
Pointe
Tustin Market Place
Eastwood
Northwood
Walnut
Stonegate
Kathryn
Stonegate East
Woodbury
Woodbury East
El Camino Real
Westpark
Irvine Business Complex
Irvine
Cypress Village
Martins Landing Field
John Wayne Airport
Woodbridge
Oak Creek
Orange County Great Park
Costa Mesa
UCI - North Campus
Rancho San Joaquin
University Park
South Mesa
Quail Hill - Open Space
Irvine Spectrum
University Town Center
Quail Hill
University of California, Irvine
French Hill 121 m
Laguna Altura
Turtle Rock
University Hills
Shady Canyon
CA 55
Turtle Ridge
Veeh Creek Open Space
Newport Heights
Bommer Canyon Open Space Preserve
Bommer Canyon - Open Space
ort Beach
Newport Center
Laguna
Irvine Terrace
CA 133
Signal Peak 354 m
CA 73 Toll
Edinger Avenue
Jamboree Road
Culver Drive
Santa Ana Freeway
Irvine Boulevard
Barranca Parkway
Marine Way
Laguna Canyon Road
Research Drive
MacArthur Boulevard
Gitano
Verde
North Island

The map can be found here: https://www.openstreetmap.org/search?query=Irvine%2C%20CA#map=12/33.7299/-117.8232 (https://www.openstreetmap.org/search?query=Irvine%2C%20CA#map=12/33.7299/-117.8232).

The underlying data was downloaded using the Overpass API, at this link: https://overpass-api.de/api/map?bbox=-117.8792,33.6135,-117.7281,33.7301 (https://overpass-api.de/api/map?bbox=-117.8792,33.6135,-117.7281,33.7301).

```
'C:\\Users\\randy'
```

```
'D:\\Documents\\School - WGU\\Term 4\\C750\\Project'
```

Data Exploration and Cleaning

Using the 'OSM_sampling.py' script provided by Udacity, I will create a sample file of every 10th element from the original file map data.

Next, we will parse the sample dataset to count the unique element types.

```
{'member': 2608,
 'nd': 38943,
 'node': 33409,
 'osm': 1,
 'relation': 145,
 'tag': 35851,
 'way': 4695}
```

By using the count_tags function, I got the number of unique tags shown. The ones that I expect to be of the most interest to me are:

- Nodes
- Tags
- Ways
- Members

To begin, I will check the "k" value for each "" to see if there are any potential problems. There are three regular expressions:

1. "lower", for tags that contain only lowercase letters and are valid,
2. "lower_colon", for otherwise valid tags with a colon in their names,
3. "problemchars", for tags with problematic characters, and
4. "other", for other tags that do not fall into the other three categories.

We need a count of each of these four tag categories in a dictionary.

```
{'lower': 17335, 'lower_colon': 18152, 'other': 364, 'problemchars': 0}
```

As we continue to explore the data, let's find out how many unique users have contributed to the data in this area.

```
769
```

Now, we need to check the validity and consistency of the street names.

I know from personal experience that these will be a challenge and result in a long list of streets outside the "expected" values. Many streets in Irvine consist of names that do not have street suffixes. Thus, I would expect to see names like Cardiff, Apache, Lexington, and Paseo Westpark.

The street name data is remarkably clean, though not perfect. Let's clean it up.

This is better, though it does introduce a new problem: names that start with "St" (e.g. "Stanza") are changed to "Street," which is obviously not what was intended. If this project's purpose was to fully clean the data 100%, I would add code to exclude those those exceptions. However, as a proof of concept for the methodology this was successful, so we will move on.

Let's look at the zip codes in the dataset.

```
Postal codes in data set that are not in Irvine
{'92520': {'92520'},
 '92625': {'92625'},
 '92626': {'92626'},
 '92630': {'92630'},
 '92637': {'92637'},
 '92653': {'92653'},
 '92657': {'92657'},
 '92660': {'92660'},
 '92663': {'92663'},
 '92705': {'92705'},
 '92707': {'92707'},
 '92714': {'92714'},
 '92780': {'92780'},
 '92782': {'92782'}}
```

Let's see what unique keys there are, and how many.

Total number of unique keys (tag attrib['k'])is 324.

There are a few "Fix Me" tags. I won't be cleaning those today, but they definitely indicate a problem with the data set that it would help users of OpenStreetMaps to resolve.

Let's also take a look at phone number formats and state names before we do the analysis and cleaning of the full data set.

```
{'(XXX) XXX-XXXX': 3,
 '+X (XXX) XXX-XXXX': 2,
 '+X XXX XXX XXXX': 3,
 '+X-XXX-XXX-XXXX': 3,
 'XXX-XXX-XXXX': 2}
```

```
California
ca
ca
```

The last step I'm going to take before reshaping the data and exporting it to CSV files is to clean up the street names for the whole data set, not just the same set.

Data Reshaping and Exporting to CSV Files

OK, it's time to actually reshape/clean the data and get it exported to CSV files.

```
Reshaping and export complete.
```

```
Opened database successfully
```

Data Analysis

File Sizes

map.osm	81 MB
irvine_osm_database.db	101 MB
nodes.csv	27 MB
nodes_tags.csv	5 MB
ways.csv	3 MB
ways_nodes.csv	9 MB
ways_tags.csv	6 MB

Let's do some analysis of the data.

First, we need to import some Python libraries so we can work with the data.

	Table	Number
0	nodes_tags	170339
1	ways	46950
2	ways_tags	182972
3	ways_nodes	378505
4	nodes	334088

We have quite a variety in the number of tags within each table, from about 47,000 in ways to over 378,000 in ways_nodes.

How many unique user contributors are there? Let's see ...

```
[(1017,)]
```

A little over 1,000 contributors. I haven't cpompared to other areas to see this number relative to the same metric in other areas, but this seems like a good number of contributors to me.

Let's find out who the top 10 are.

	User ID	User Name	Count
0	2709708	SJFriedl	122169
1	10165657	Fluffy89502	37379
2	53073	Aaron Lidman	19286
3	6123527	d5f40e1632	9592
4	11567017	n76_oc_import	9365
5	101657	ponzu	9058
6	9812385	leandrok	7843
7	6869539	Grayfox88	6703
8	10544027	Avidgolfer	5759
9	6203853	Bill Sellin	5000

Even within this top tier of contributors, there is a clear division. The top 3 are all above 10,000 (well above), while the ones that make up the remainder of the top 10 are below. The top contributor has more than 3 times as many contributions as the second highest, over 122,000.

Now, what are the most common node tags?

	Node Tags	Count
0	city	28553
1	postcode	28513
2	street	28503
3	housenumber	28485
4	building	27278
5	height	8696
6	highway	5092
7	state	1827
8	barrier	1388
9	traffic_signals	1091

Addresses seem to be the most common node tags, and in general they appear to be fairly complete--housenumber, street, city, and postcode all have counts very close to each other, with only 68 separating the most from the fewest. Interestingly, state is left out of many of them, as there are more than 15 times as many address without the state as there are with it.

Now, what are the most common amenities?

	Amenity	Count
0	parking	727
1	restaurant	165
2	fast_food	110
3	bench	96
4	shelter	75
5	school	70
6	toilets	66
7	bicycle_parking	64
8	drinking_water	53
9	cafe	53
10	parking_entrance	51
11	waste_basket	49
12	bank	42
13	fuel	36
14	fountain	36
15	place_of_worship	29
16	bbq	27
17	vending_machine	15
18	fire_station	14
19	ice_cream	11

I think this might tell us more about what contributors prioritize than it does about what amenities are most common. Parking is top of mind for many urban dwellers, so perhaps they notice and contribute parking locations. After parking (and well behind it) comes 5 different designations for places to eat (restaurant, fast food, cafe, bbq, and ice cream). We do like our food here in Irvine!

Next, I would like to see what religion appears to be most popular/common. I did not analyze and clean this data, so it is entirely possible that this will not be completely accurate, but it at least gives us initial idea.

```
[('christian', 7)]
```

On the surface, it appears that Christian is the most popular religion. However, I ran this code again (not included here) with a slight alteration--I changed the limit from 1 to 3. The result was exactly the same, which means there are no other religions identified by the religion tag. This seems extraordinarily unlikely, so some additional data cleaning would need to be done on this tag before running it again to provide any meaningful result.

One thing that makes Irivne unique is the identification of individual "villages." Most other cities just call them "neighborhoods" and don't name them--or if they do, they are fairly generic names. In Irvine, however, each neighborhood has its own name and its own unique identity. Even though they are all Irivne, Woodbridge is distinct from Westpark, which is distinct from Stonegate East, and so on. It's one of the things that makes living in Irvine fascinating to me.

So, let's see which villages have the most contributions. This won't tell us much about the villages themselves, but it could be an interesting first glance.

	Villages Tagged Most Frequently
0	Northwood Pointe
1	Woodbury
2	Woodbury East
3	Stonegate East
4	Laguna Altura
5	Northwood
6	Walnut
7	El Camino Real
8	Lower Peters Canyon
9	Turtle Rock

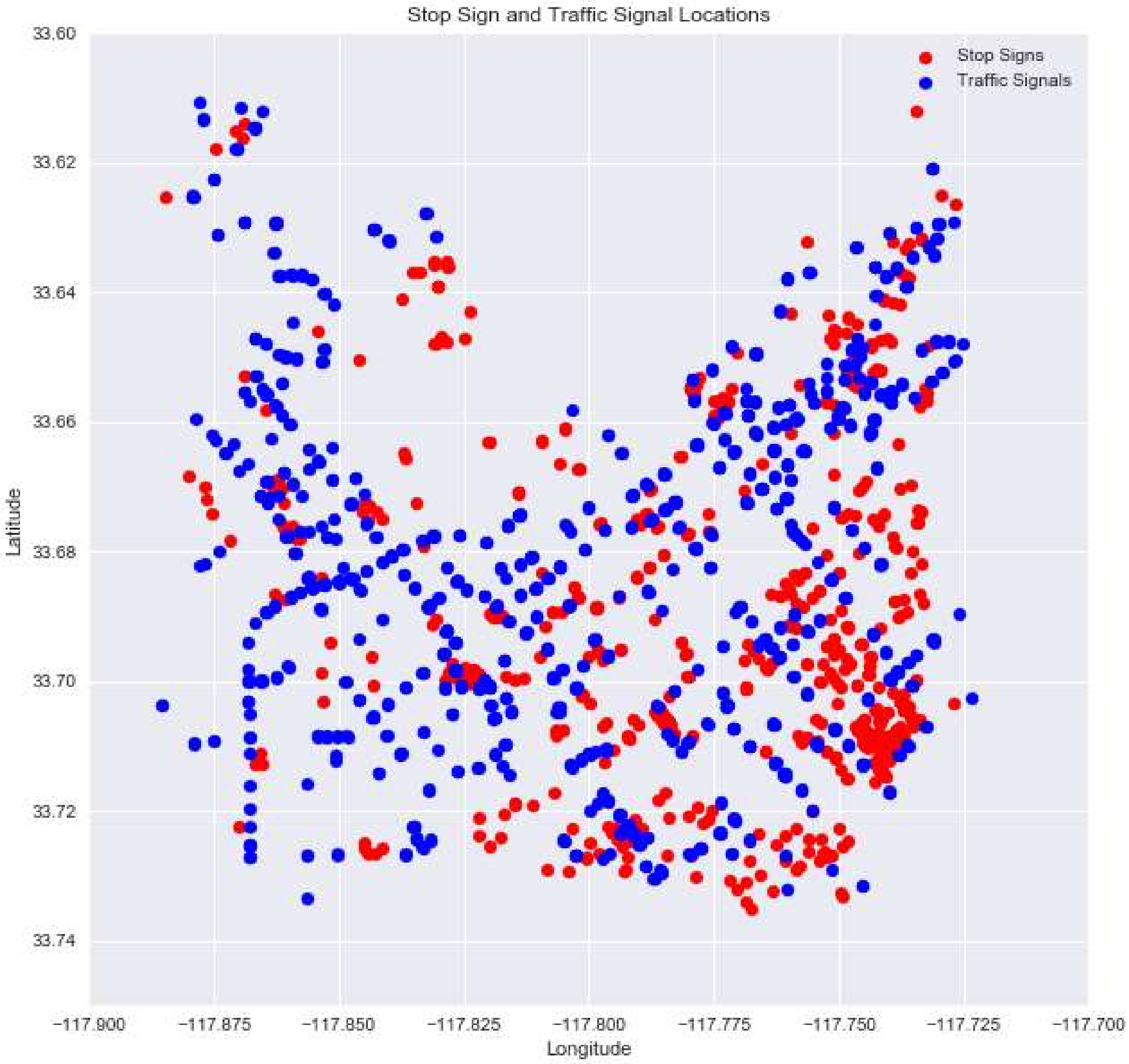
Nothing particularly revealing here. This would be fun to explore sometimes.

Additional Ideas About the Data - Graphical Representations

We've looked at a lot of text and tables, so let's do a couple of graphical representations.

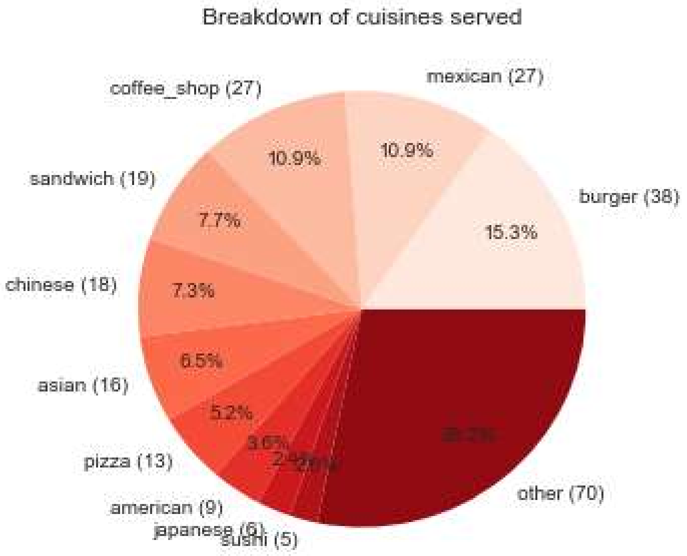
Living in Irvine, it seems like there are a lot of stop signs--almost eveywhere I go. I wonder how those compare to traffic signals?

To find out, I am going to create a visualization that plots all stop sign and all traffic signals by their latitude and longitude.



They see to be appximately evenly distributed between the two, byt there are definitely some sections/stretches with no stop signs and a couple of high concentrations.

Let's do one more visualization. As I said earlier, we Irvine residents and visitors do like to eat. So let's take a look at the cuisine served. To be someehat ironic about it, I will use a pie chart to visualize the results.



Other Thoughts About the Data

In looking at the top 10 contributors, I would like to know more about the top one, and why they have contributed so much. Unfortunately, that data is not available. I could probably getr some clues if I were to analyze what their specific contributions were, but that is beyond the scope I have chosen for this project.

Looking at the stop signs vs. traffic signals visualization, I would like to overlay that onto a map sometime and see exactly where these are.

There is clearly some additional data cleaning that needs to be done to make this data set truly useful--the religion example above illustrates this well.

Benefits & Problems of Implementing Changes

Benefits:

- One of the greatest and most obvious benefits of implementing data wrangling changes is that it makes the data more useful. Clean, consistent data can be used for analysis to gain insights, make decisions, and take action.
- A less obvious but still important benfit of implenting these changes is that it provides a good example for future contributors. When I am doing something new, I like to see if there are standards in place and see how others have implemented those standards. I know that not everyone is like me, but for those who are, the more clean and consistent data there is, the more they have to follow to keep their entries in line with the "good data" that's already there, thus increasing the amount of "good data."

Anticipated Problems:

- The biggest anticipated problem I see with attempting to implement these changes is that for several of the tags--street names comes to mind right away--there are a lot of edge cases. Trying to deal logically, consistently, and accurately with those could be even more time consuming and challenging than usual.
- Another problem with implementing these changes, especially if they were to become data quality standards, is that they may be too restrictive to be flexible enough to accommodate all od the possible legitimate variations on the data that could belong to each tag. If they are too restrictive, they may discourage users from contributing due to frustration. But if they are not restrictive enough, then data quality remains an issue and we may be no better off than we were before we started.

Conclusion

This data set was generally in good shape--except for the notable exceptions called out earlier. It did present some unique challenges, particularly in the street names. Even in the partially cleaned conditions for this project, it did yield some interesting insights.

I would need considerably more skill and experience to deal with some of these problems, and until I can, I would say these improvements I have identified are not ready to be implemented. Perehaps the most useful change would be to more thoroghly parse compound k-values throughout the data set. While this would take a lot more time, it may be the change that would provide the most useful changes for the time spent and could make other issues easier to identify and correct.