# OpenStreetMap Data Case Study

## Map Area

Tampa, FL, United States

- https://mapzen.com/data/metro-extracts/metro/tampa_florida/

I chose this area, because I am unfamiliar with it, and wanted to see what the data for this area looked like.

## Problems Encountered in the Map

Use `audit.py` to check and clean for inconsistencies in city, street, and zip codes. Here are some examples of problems found:

### City name inconsistencies

- Capitalization:

```
- SPRING HILL -> Spring Hill
- port richey -> Port Richey
```

- Spelling

```
- Clearwarer Beach -> Clearwater Beach
- St Petersbug -> St. Petersburg
```

- Punctuation
  - `Palm Harbor, Fl. -> Palm Harbor`
  - `Land O Lakes -> Land O' Lakes`

### Street name inconsistencies

Some streets are listed with more information than the street address. For example:

```
- 8492 Manatee Bay Dr Tampa, FL 33635
- 6010 US-301, Ellenton, FL 34222, Vereinigte Staaten
```

Some streets have a `#` symbol in their name, for example:

```
- Starkey Rd #G
- E Fletcher Ave #131
```

Some streets have abbreviated directions. For example:

```
- E -> East
- NW -> Northwest
```

Additionally, sometimes the direction is listed at the end of the street, rather than at the beginning. For example:

```
- 37th Ave Northeast
- 77th Drive West
- San Martin Blvd NE
```

Some street names have `Suite` in the name. For example:

```
- 66th Street North Suite 135
- W Cypress St Suite
```

After these fixes, there are still a few inconsistent street names. These are streets that are mostly US Highways, such as

```
- State Road 52
- SR 52
- FL 52
- U.S. 19
- US-301
```

### State inconsistencies

Use `audit.py` to clean state names: The majority of the data have `FL` as the state in `addr:state`. Otherwise, the state is listed as:

```
Florida 24
GA      3
Fl      3
fl      16
```

```
 florida 1
 FLq     1
```

### Zip code inconsistencies

There are a few inconsistent zip codes, all of which have a length longer than 5. For example:

```
 - 33548:33556
 - 34669; 34667; 34667
```

## Data Overview

### File sizes

```
tampa_florida.osm.... 355 MB
nodes_csv............ 131 MB
nodes_tags.csv....... 6.5 MB
ways.csv............. 11 MB
ways_nodes.csv....... 44 MB
ways_tags.csv........ 32 MB
tampa.db............. 204 MB
```

### Number of nodes

```
SELECT COUNT(*) FROM nodes: 1655566
```

### Number of ways

```
SELECT COUNT(*) FROM ways: 182866
```

### Number of unique users

```
SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e: 1448
```

### Top 10 contributing users

```
SELECT e.user, COUNT(*) as num FROM (SELECT user FROM nodes UNION ALL
SELECT user FROM ways) e
GROUP BY e.user
```

```
ORDER BY num DESC
LIMIT 10
```

```
coleman          258302
woodpeck_fixbot  235013
grouper          187215
EdHillsman       106677
NE2              72924
David Hey        60918
LnxNoob          58364
KalininOV        48825
westampa         42145
bot-mode         37656
```

### Number of users contributing once

```
SELECT COUNT(*) FROM (SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
HAVING num=1) u: 330
```

### Top 10 amenities

```
SELECT value, COUNT(*) as num FROM nodes_tags WHERE key="amenity"
GROUP BY value
ORDER BY num DESC
LIMIT 10
```

```
restaurant        852
place_of_worship  771
school            553
fast_food         396
bicycle_parking   353
bench             279
fuel              235
fountain          201
bank              170
toilets           148
```

### Top 5 places of worship

```
SELECT nodes_tags.value, COUNT(*) as num FROM nodes_tags
JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value="place_of_worship")
i ON nodes_tags.id=i.id
WHERE nodes_tags.key="religion"
GROUP BY nodes_tags.value ORDER BY num DESC LIMIT 5
```

```
christian  724
jewish     4
bahai      3
buddhist   3
unitarian_universalist 3
```

### Top 5 cuisines

```
SELECT nodes_tags.value, COUNT(*) as num FROM nodes_tags
JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value="restaurant") i
ON nodes_tags.id=i.id WHERE nodes_tags.key="cuisine"
GROUP BY nodes_tags.value ORDER BY num DESC LIMIT 5
```

```
american 93
pizza    70
mexican  41
italian  28
seafood  25
```

### Top 10 restaurants

```
SELECT value, COUNT(*) as num FROM nodes_tags
JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value="restaurant") i
ON nodes_tags.id=i.id WHERE key="name"
GROUP BY value ORDER BY num DESC LIMIT 10
```

```
Tijuana Flats      8
Applebee's         6
Bob Evans          6
Denny's            6
IHOP               6
Outback Steakhouse 6
Panera Bread       6
Chili's            5
Golden Corral      5
```

```
Pizza Hut          5
```

# Other Ideas

## Further fix the errors encountered in the street names

The street names that are now considered inconsistent are mostly due to US Highway names that have numbers. Therefore, streets that are US Highways should be taken into account when deciding whether or not a street name is consistent.

### Benefits

- The dataset is further cleaned

### Anticipated Issues

- Need to make sure that a street that has the format of a highway name (ends in a number) is actually a highway, and is not a mistake/typo in the street name.

## Validate zip codes

A few states were listed as `GA`. The addresses that had these listed should be verified with external data to see if GA is a typo and the address is indeed in FL, or if the address is in GA and is included in the dataset by mistake. The zip code fields that have multiple zip codes listed with semicolons also need to be validated. The data can be validated with external data sources, such as Google Maps.

### Benefits

- Improvement in accuracy for data queries.

### Anticipated Issues

- The external database could have incorrect info.
- The external database could be missing the needed information.
- Need users to perform the cleaning.

## Check consistency of other data fields

The consistency of other fields, like phone numbers, also needs to be checked. As with the zip codes, this can be done by cross-referencing an external data source, and has the same benefits and anticipated issues.

## Ensure new data is consistent

As this analysis has shown, this dataset is not without errors. Instead of cleaning the dataset after data has been entered, I think a better way would be to have a more structured way for users to input data. For example, the user could only select a zip code from zip codes that were validated to

be in the area.

#### Benefits

- Less cleaning of data set needed

#### Anticipated Issues

- Need users who are dedicated to implementing the solution, could implement gamification to encourage users
- Initially it would require a lot of time to implement and validate the structured input form

### Add more data for restuarant delivery

From querying the dataset, there are 852 restaurants:

```
SELECT value, count(*) FROM nodes_tags
WHERE value="restaurant"
```

91 of these restaurants have information on delivery, with 72 having no delivery and 19 providing delivery:

```
SELECT value, count(*)
FROM nodes_tags JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value="
restaurant")
ON nodes_tags.id=i.id
WHERE key="delivery"
GROUP BY value
```

I think that people using the database would be interested in whether or not a restaurant provides delivery, so the database could be improved by adding delivery information for more restaurants.

#### Benefits

- Enhanced user experience

#### Anticipated Issues

- Need people to find delivery information
- Requires time to implement
- Need to find external data source, perhaps Yelp data

## Files

All of the analysis is done with the `osm.ipynb` file. The cells were exported in python scripts as:

```
- audit.py: audit street names, city names, and zip codes
- data.py: from OSM file, create CSV file
- database.py: from CSV file, create SQL database
- mapparser.py: count unique tags
- query.py: SQL queries used
- sample.py: extract 25 MB sample of the OSM file
- users.py: get contributing users
- tags.py: count patterns in the tags
```

## References

- https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md

Jump to Line
Go



```
<script crossorigin="anonymous"
integrity="sha256-0/jjywXESVr8eMN68y6Hf5nUfLBqgVjvr6Kuv1VnyDA="
src="https://assets-cdn.github.com/assets/frameworks-d3f8e3cb05c4495afc78c37af32e87
7f99d47cb06a8158efafa2aebf5567c830.js"></script>

<script async="async" crossorigin="anonymous"
integrity="sha256-imYlhu1Qzl0SLt7A7KTC2wUrY1iKs/wZK0GPJh0tC1M="
src="https://assets-cdn.github.com/assets/github-8a662586ed50ce5d122edec0eca4c2db05
2b63588ab3fc192b418f261d2d0b53.js"></script>
```