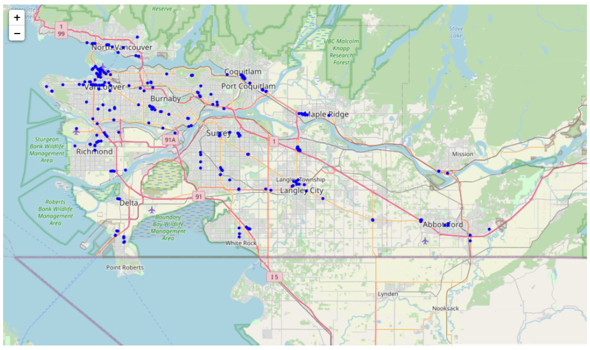
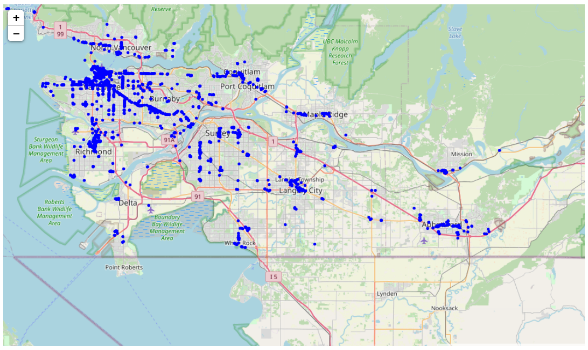
Part 2: Chain and Non-chain Restaurant Identification and Analysis(Siyuan Wu)

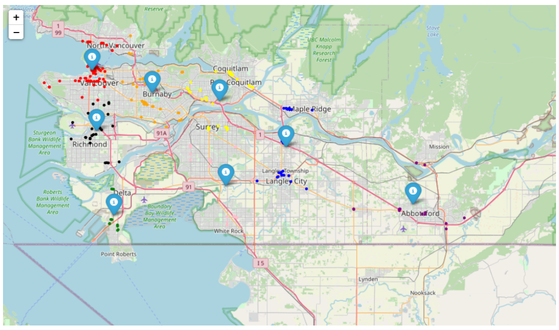
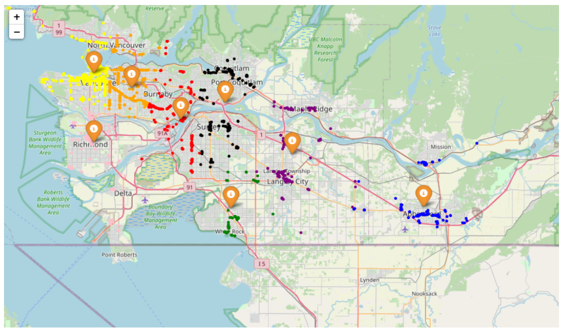
- Problem addressing:
    - ```
      The problem: I feel like there are some parts of the city with more chain
      restaurants: is that true? Is there some way to find the chain places automatically
      and visualize their density relative to non-chains?
      ```
    - For this topic, we are only looking into data where attribute equals to 'restaurant'.
    - A chain restaurant is defined to be having more than 3 data entries in the restaurant category or have the name listed on Wikipedia' chain restaurant' page.
- Data cleaning (data_forming.py)
    - All data cleanings are achieved through manipulating pandas libraries, so all data cleanings are performed on the spot.
    - Remove data entries where attribute 'name' is equal to Nan.
        - If the name is Nan, it would probably be a false entry. In addition, users cannot find the restaurant from the data, so it should be excluded.
    - Remove duplicate data entries.
        - Some restaurants have more than 3 data entries in the data and have their name listed on the Wikipedia page.
        - Duplicate data would reduce the accuracy of clusters, so duplicate data entries are being removed using `.drop_duplicates()`.
    - Manual cleaning
        - Some scraped data have branch names included in their names, preventing them from matching the OSM data.
        - These names don't have a general pattern that can be clean with regular expression. Thus, manual cleaning is needed.
        - Examples of manual cleaning include: 'A&W(Canada)' to 'A&W', 'Baton Rouge (restaurant)' to 'Baton Rouge', 'BeaverTails restaurant' to 'BeaverTails'
- Data gathering(data_ forming.py)
    - The definition of a chain restaurant is to have more than 3 data entries in the dataset. However, some chain restaurants are having less or equal to 3 in size, so data from Wikipedia are being used to chain restaurants from this category.
    - The first list of restaurant names is generated by retrieving `.count()` larger than 3 on the restaurant name.
    - The second list of restaurant names is generated from data scraping from Wikipedia using `BeautifulSoup`. Since the OSM data is based on Vancouver, only USA and Canada chain restaurant pages are being used. The data source page is listed below.
        - The U.S chain restaurant: https://en.wikipedia.org/wiki/List_of_restaurant_chains_in_the_United_States
        - Canada chain restaurant: https://en.wikipedia.org/wiki/List_of_Canadian_restaurant_chains#Major_chains
    - Two lists then concatenate to form a final list of names, then performing a join operation on the list and OSM data to obtain the chain restaurant data. All other restaurants that are not chain restaurants are classified into non-chain restaurants.
      ```

- Finally, chain restaurant data is written into 'data/chain.csv', and non-chain restaurant data is written into 'data/non_chain.csv.'

- Analysing techniques and Visualization (map_cluster.ipynb, cluster.py, stat_test.py)
    - The map_cluster.ipynb and its helper code cluster.py are adopted from the groupmate, Yifan Zuo's work. The code employed python `folium` package to present the map.
        - The visualization of chain and non-chain restaurant data points on map are below.

Chain Restaurant Distribution	Non-chain Restaurant Distribution
	

    - Cluster is done by using `sklearn.cluster Kmean`
        - The number of clusters is tuned to 8 to provide the best result.
        - The large blue and orange indicator indicates the center of each cluster and is calculated by the mean of lon and lat of that cluster.
        - The visualization of chain and non-chain restaurant data clusters and their centers on map are below.

Chain Restaurant Cluster	Non-chain Restaurant Distribution
	

    - After data points are grouped into clusters, lat, lon and their designated cluster number are written into 'data/cl_data' and 'data/ncl_data' for chain and non-chain restaurants.
    - In stat_test.py, data from 'cl_data' and' ncl_data' are converted to the contingency table for `scipy`'s stats package. Finally, the code will perform a chi-square test on the relationship between locations and chain/non-chain restaurants.

- Findings
    - The null hypothesis for the chi-square test is that chain/non-chain restaurant density has is not affected by geographic location.
    - The outcome of the p-value for chi-square is `p-value = 0.000002172` which is less than the after-correction p-value of `0.00625`, so we can conclude that geographic location has some effect on

chain/non-chain restaurant density.

- Limitations
    - Some chain restaurant names are still not being classified into chain restaurants because of their slightly different name. Therefore, more comprehensive manual data cleaning is needed.
    - Drawing out the boundary for each cluster may be helpful to see which area has more or fewer chain restaurants.
    - A density map instead of a cluster map may be more straightforward to show their density.